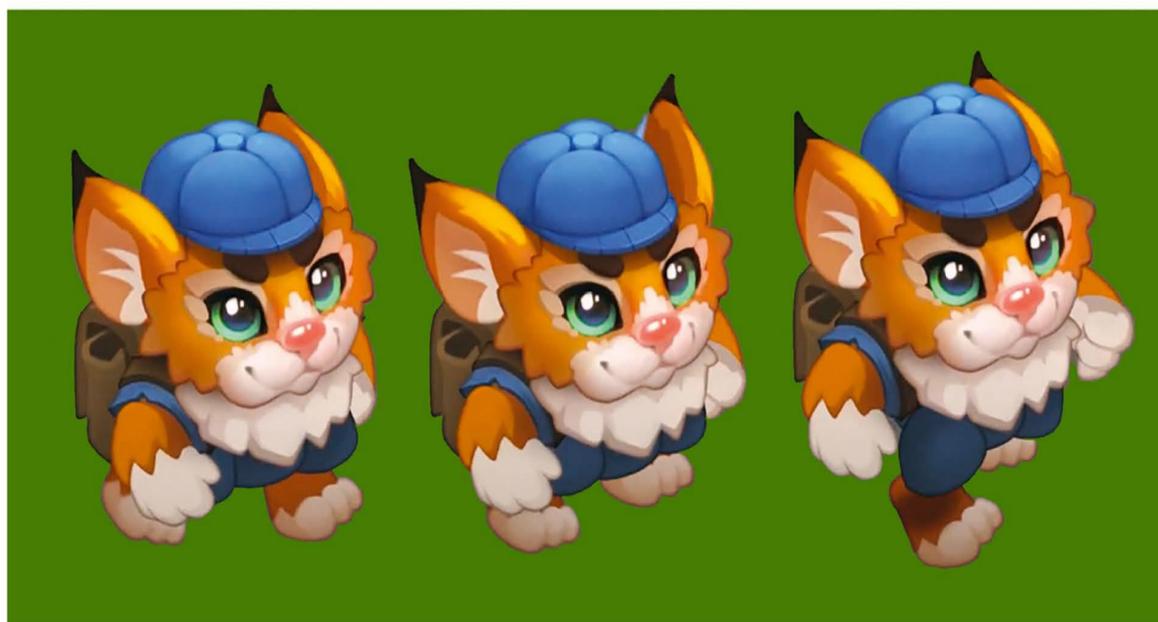


АННА ЛЕПЕШКИНА

Основы анимации в
Spine 2D
для начинающих



КАК СОЗДАВАТЬ КРУТЫЕ ИГРЫ
**БЕЗ ГИГАНТСКИХ
БЮДЖЕТОВ**

АННА ЛЕПЕШКИНА

Основы анимации в
Spine 2D
для начинающих

КАК СОЗДАВАТЬ КРУТЫЕ ИГРЫ

БЕЗ ГИГАНТСКИХ
БЮДЖЕТОВ

Лепешкина, Анна Викторовна.

Л48 Основы анимации в Spine 2D для начинающих : как создавать крутые игры без гигантских бюджетов / Анна Лепешкина. — Москва : Эксмо, 2024. — 288 с. : ил. — (Российский компьютерный бестселлер. Гейм-дизайн).

ISBN 978-5-04-190811-9

Программа Spine 2D представляет собой идеальный инструмент для создания анимации персонажей в 2D. Она широко используется в игровой индустрии, а также в различных проектах для мобильных приложений, веб-сайтов и рекламных материалов, где требуется создание анимации. Книга поможет читателям овладеть навыками работы в программе Spine 2D, а также научиться использовать различные инструменты для редактирования и настройки анимации, управлять движениями и эмоциями персонажей, создавать анимационные ролики с помощью Spine 2D и Unity, экспортировать готовую анимацию в различные форматы и многому другому.

Эта книга предназначена для художников, дизайнеров и программистов, которые хотят создавать высококачественную анимацию персонажей в играх и других проектах, сокращая время на ее создание и улучшая качество работы благодаря широкому набору инструментов и функций Spine 2D.

УДК 004.4
ББК 32.973.26-018

Оглавление

Вступление	6
Разрешите представиться	8
Как я пришла в сферу анимации	8
Должен ли аниматор уметь рисовать	9
Какие навыки необходимы аниматору	10
Чем занимается 2D-аниматор	13
Глава 1. Виды анимации	16
Покадровая анимация	17
Анимация перекладкой	19
Деформация	20
Скелетная анимация	20
3D-анимация	21
Сочетание нескольких видов анимации в Spine 2D	22
Плюсы и минусы скелетной анимации	23
Глава 2. Почему именно Spine 2D	26
Глава 3. Spine 2D: знакомство с программой	32
Интерфейс	33
Общий алгоритм анимирования в Spine 2D	34
Основные инструменты	35
График характеристик движения	38
Шкала времени	41
Инструменты шкалы времени	43
Предпросмотр	45
Подсказки от разработчиков	47
Глава 4. 12 принципов анимации	48
Глава 5. Топ-5 упражнений для начинающего аниматора	58
Движение по прямой	59
Прыжок мяча	69
Скачок мяча в сторону	73
Маятник	76
Дополнительный элемент. Добавляем хвост движению	85
Глава 6. Подготовка материалов для анимации в Spine 2D	92
Выбор концепта для работы. Какие исходники лучше подходят для скелетной анимации	93
Обводка и тени	95
Прогноз анимации на этапе подготовки исходника	98
Проблемы, которые можно решить на этапе подготовки исходника	99

Глава 7. Нарезка концепта для анимации. Практические примеры	102
Нарезка простейшего персонажа	103
Персонаж для игры в жанре платформер	104
Персонаж для игры в изометрии	108
Нарезка интерфейса. Поп-ап	109
Глава 8. Импорт исходника в программу Spine 2D	112
Глава 9. Возможности Spine 2D. Углубляемся в функции и инструменты программы	116
Маска	117
Габаритный контейнер	121
График характера движения. Целостный формат	123
Меш. Ручная настройка весов	127
Меш. Free form deformation	130
Глава 10. Ограничители	132
Инверсная инематика	133
Ограничитель траектории	139
Ограничитель траектории. Практические примеры	142
Ограничитель трансформы	156
Ограничитель трансформы. Практические примеры	158
Глава 11. Эффект псевдо-3D	162
Настройка	163
Анимация	166
Изменение формы	167
Изменение ракурса во время анимации	168
Изменение положения и поворот объекта	169
Глава 12. Работа с покадровой анимацией	170
Секвенции	172
Глава 13. Просмотр кадров и дуг. Окно «Ореол» (Ghosting)	174
Отслеживаем дуги	178
Глава 14. Скины	180
Добавление костей в оболочки	182
Создание оболочек при экспорте из Photoshop	183
Конструктор оболочек	184
Глава 15. Анимация персонажей	186
Передача характера и настроения героя	187
Создание скелета персонажа	189
Псевдо-3D для туловища	194
Глава 16. Основные анимации персонажа для игры	196
Ходьба	197
Анимация бега. Основные отличия от ходьбы	202
Анимация покоя	204
Анимация атаки	208
Анимация прыжка	213

Глава 17. Лицевая анимация	218
Как показать эмоцию	219
Настройка для лицевой анимации	221
Псевдо-3D лица	230
Особенности анимации персонажа в изометрии	234
Глава 18. Эффекты в Spine. Пример нестандартного использования программы	238
Глава 19. Экспорт анимаций. Форматы и настройки	242
Глава 20. Создание анимационного ролика с помощью Spine 2D и Unity	254
Основной алгоритм работы	255
Подготовка исходных кадров	256
Нарезка кадров под анимацию	258
Настройка скелета	261
Анимация	263
Сборка	264
Работа с камерой	265
Видеоиз движка	267
Глава 21. Что дальше? Как развиваться аниматору	268
Прокачиваем навыки	269
Портфолио. Советы	270
Широкий кругозор или конкретное направление: что выбрать?	274
Где учиться? Самообразование, университет, курсы	276
Заключение	278

ВСТУПЛЕНИЕ

В детстве многие хотели стать мультипликаторами: создавать истории, оживлять статичные картинки. «Хочу делать мультики» — желание, которое каждый хотя бы раз испытывал в детстве, да и не только. Анимация вызывает интерес, дает возможность реализовать творческий потенциал. Сложно представить себе сферу развлечений, где не присутствует анимация.

Помимо непосредственно мультипликации, анимация давно стала частью игровой индустрии. Каждый сундучок, каждая иконка требуют внимания аниматора для оживления изображения. Анимируется все, от персонажей до интерфейса и элементов в нем. Огромный пласт задач ждет аниматора в сфере геймдева. Анимация нужна и в мобильных проектах, и в сегменте инди-игр, и даже в крупнобюджетных играх. Вспомним *Atomic Heart* и видеоролики о навыках, необходимых в этой игре, созданные в стиле классической мультипликации. В 3D-играх есть 2D-интерфейс, где также нужны анимация или анимационные вставки.

Анимация используется для создания рекламных роликов, разных типов видеоконтента, в приложениях, в киноиндустрии. Аниматору открывается множество различных возможностей для реализации своих навыков.

В этой книге мы ознакомимся с анимацией в целом, с ее областями применения, основными принципами создания. А главное — углубимся в скелетную анимацию, которая набирает все большую популярность в игровой индустрии и других сферах.

Эта книга — путеводитель по миру скелетной анимации для новичка. В ней мы рассмотрим основные виды и принципы анимации. Разберем самую популярную программу для скелетной анимации Spine 2D в теории и на практике. Я расскажу, как проходит цикл разработки анимации для игр, и поделюсь внутренней кухней своих проектов. После прочтения этой книги и выполнения практических заданий вы сможете использовать скелетную анимацию в своих работах и улучшить свое профессиональное портфолио.

Книга подойдет для начинающих аниматоров, художников, игровых разработчиков, которые хотят изучать игровую анимацию и работать в сфере геймдева или мультипликации. Также она будет полезна для инди-разработчиков, желающих использовать скелетную анимацию в своих проектах.

Разрешите представиться

Но сначала давайте я немного расскажу о себе и о том, как я пришла к работе с анимацией. Меня зовут Аня, и я художник-аниматор с десятилетним опытом работы в игровой индустрии. Я занимаюсь концепт-артом, иллюстрациями, игровой анимацией и даже иногда пишу музыку для игр. А еще я гейм-дизайнер и сценарист. В этой книге я поделюсь с вами опытом и расскажу, как освоить скелетную анимацию. Я помогу освоить программу Spine 2D и подскажу, как обучаться максимально эффективно.

Как я пришла в сферу анимации

Истории о том, как я рисовала с самого детства, не выпуская карандаш из рук, не будет. Я всегда была творческим ребенком, занималась музыкой, танцами и также ходила в кружок рисования. Но рисование и тем более анимация тогда были для меня просто занятиями после школы. Главным моим увлечением были игры. Я скупала все печатные издания, посвященные новинкам игровой индустрии. У меня есть полное собрание журналов «РС-ИГРЫ» от первого до последнего номера. Также я счастливый обладатель огромной коллекции дисковых изданий игр и картриджей. А в подростковом возрасте у меня родилась идея не просто играть, но и создавать игры. Я начала делать простенькие проекты в 14 лет. Тогда не было столь огромного количества средств разработки, как сейчас. Не было ни Unity, ни Unreal Engine. Даже простые задачи давались с трудом. Образование я получила именно в сфере программирования. Но в университете меня увлекло рисование в Photoshop. Я купила свой первый графический планшет, и понеслось. После университета я сразу ушла из сферы программирования и обратила внимание на художественную часть игровой разработки.

Я начала свой путь как 2D-художник и в сферу анимации пришла не сразу. За более чем семь лет в игровой индустрии я успела поработать и над ПК-играми, и в мобильном геймдеве, и в мультипликации. Но я всегда хотела делать что-то свое, самовыражаться, воплощать собственные идеи в реальность, создавать целые миры, наполнять их жизнью, писать истории.

И вот я пришла к разработке собственных игр. Первый проект *Potata: fairy flower* начался с концептов локаций и персонажей. Изначально не было планов создать игру, я делала лишь работы для портфолио в едином стиле. Нарисовав огромное количество скетчей и готовых локаций, я задумалась, что же с ними делать дальше. Посмотрев на объем материала, я приняла решение собрать это все в единый проект. Однако статичные локации не смогли бы раскрыть всю красоту придуманного мной мира.

Именно с того момента я начала заниматься анимацией, так как поняла, что в собственных проектах без анимации никуда. Изучив несколько программ, я остановилась на Spine 2D как на самой удобной и понятной среде для скелетной анимации. На сегодняшний день мы с мужем выпустили уже три проекта, вся анимация в которых сделана собственными силами. А также я выполняю заказы на фрилансе как аниматор в сфере геймдева и являюсь автором курса для начинающих аниматоров в школе Smirnovschool.

В основном я использую в проектах именно скелетную анимацию. Но и для покадровой нашлось место. Некоторые из наших игр нарисованы в стиле пиксель-арт. Вся анимация в проектах Gravifire и Lord Ambermaze выполнена покадрово.

Должен ли аниматор уметь рисовать

Мой опыт художника, безусловно, очень помог мне в работе аниматором. С самого начала я уже понимала принципы динамичных поз, знала, что такое линия движения, умела создавать персонажей для анимации. Многие художественные принципы применимы и в анимации. Законы композиции, постановки кадра, зрелищности движения. Художественная база очень полезна, но не необходима для скелетной анимации.

Само собой, если вы хотите заниматься классической покадровой анимацией, то уметь рисовать для вас обязательно. Кадры анимации сами собой не нарисуются, тут нужен художник.

В скелетной же анимации больший упор делается на владение программой и техниками анимирования, нежели на художественные навыки. Да, вам нужно знать основы Photoshop и желательно уметь подправлять исходники, например, слегка подмазать кистью или переместить определенные части рисунка. Зачастую аниматор полностью подготавливает исходник для анимации. Вам поступает цельная картинка, а ваша задача нарезать ее на сегменты так, чтобы эти кусочки позволили создать требуемое движение. Соответственно, нужно уметь использовать Photoshop или другой аналогичный графический редактор для нарезки исходника. И конечно, вам нужна минимальная художественная база, чтобы не создать невозможных в природе движений.

Какие навыки необходимы аниматору

Умение анимировать складывается из множества навыков, комбинация которых создает профессию аниматора. Аниматор не просто человек, который двигает картинки из точки А в точку Б. Аниматор должен понимать принципы этого движения, делать его естественным и зрелищным. С помощью анимации можно показать характер персонажа, создать определенное впечатление о герое. Именно анимация оживляет целые игровые сцены и локации. Что же нужно уметь крутому аниматору? Давайте разбираться. Будем рассматривать именно скелетную анимацию, так как она является основой для этой книги.

ОСНОВНЫЕ ПРИНЦИПЫ АНИМАЦИИ

12 принципов анимации — это библия аниматора. Это база навыков для любого аниматора, где бы он ни работал и какую бы анимацию ни делал. Реализацию некоторых принципов упрощает Spine 2D. Но все же программа не делает крутую анимацию по нажатию одной кнопки. Поэтому базовые принципы нужно знать и понимать, как они реализуются в рамках скелетной анимации в программе Spine 2D.

НАВЫКИ РАБОТЫ В PHOTOSHOP ИЛИ ДРУГОМ АНАЛОГИЧНОМ РЕДАКТОРЕ

Для работы в Spine 2D необходима подготовка исходников. Например, если мы анимируем персонажа, мы должны разделить его на фрагменты по суставам и отделить побочные объекты. При этом разделении порой нужно дорисовывать недостающие части, которые не видны в статике, но будут видны в процессе движения. А значит, требуются минимальные художественные навыки. Когда мы создаем эффекты, нужно понимать, как работают режимы наложения слоев в Photoshop и как их же использовать в Spine 2D. В любом случае нам придется обратиться к Photoshop, от него никуда не деться.

АНАТОМИЯ, ПЕРСПЕКТИВА, КОМПОЗИЦИЯ

Чтобы правильно нарезать персонажа, нужно понимать основы анатомии. Где расположены суставы и как они двигаются. Иначе есть риск создать неестественное движение, выгиб конечности в сторону, противоположную возможному движению, или вовсе сломать персонажа. Анатомия также важна для создания правдоподобной

анимации. Если мы знаем, какова биомеханика суставов в теле человека, какая мышца тянет за собой другие, мы можем перенести все это в нашу анимацию. Иначе персонаж будет двигаться несбалансированно и нереалистично. Зритель не поверит в то, что герой настоящий.

Для анимации объектов, например сундука, нужно понимать законы перспективы. При движении крышки сундука она будет искажаться согласно перспективе. И это нужно правильно изобразить в анимации. Также при любой анимации в пространстве придется изменять размер объекта согласно перспективным искажениям. Иначе наш объект не будет казаться объемным и вписываться в окружающую его среду.

При анимации сцен или просто нескольких объектов в связке понадобится знание основ композиции. Какой объект главный, какое движение основное, как это подчеркнуть и донести до зрителя идею? Как привлечь внимание на нужный нам элемент? В решении этих вопросов помогут принципы композиции.

Причем композиция может работать не только в рамках крупной сцены, но и в рамках одного объекта. На персонаже могут быть главные и второстепенные элементы. В анимации персонажа могут быть главные и второстепенные движения. У анимации может быть основная задача, на которую нужно обратить все внимание зрителя. А значит, нам нужно использовать композицию, чтобы достичь поставленных целей.

ЗНАНИЕ СРЕДСТВ И ВОЗМОЖНОСТЕЙ SPINE 2D, ТЕХНИЧЕСКИЕ НАВЫКИ

За редким исключением анимация уже давно не создается карандашом на бумаге. Аниматору нужно обладать навыками работы с программой для анимации. А в идеале освоить несколько программ, чтобы комбинировать их для получения лучшего результата. Spine 2D покрывает все задачи скелетной анимации и позволяет осуществлять полный цикл разработки движения. Но навыки, например, в Adobe After Effects для создания эффектов не повредят.

Нужно знать основные форматы для экспорта анимации, понимать, в каком виде она будет использована после вашей работы, и, соответственно, уметь предоставлять нужный формат на выходе.

ПОНИМАНИЕ ИГРОВОЙ ЛОГИКИ, ПРИНЦИПОВ ПОСТРОЕНИЯ ЛОКАЦИЙ И ПРИНЦИПОВ ИГРОВОЙ АНИМАЦИИ В ЦЕЛОМ

Если вы хотите работать в сфере разработки, то понимание игровой логики и того, как анимация используется в играх, обязательно.

Возьмем самый распространенный пример. Анимация прыжка персонажа. Если создавать ее без понимания игровой механики прыжка, то аниматор сделает персонажа, прыгающего на заданную высоту или длину, единой анимацией. Но в игре персонаж будет прыгать на разные расстояния и высоту. Или он будет падать с разной высоты. Мы не знаем точную траекторию прыжка. Чтобы сделать анимацию прыжка гибкой, ее нужно разбить на части. Момент подпрыгивания — зависание в воздухе — падение — приземление. А значит, аниматор должен сделать не одну анимацию, а четыре сегмента. Да, скорее всего, изначально все равно будет создана единая анимация, а потом разбита на части. Но аниматор должен понимать, как это сделать и как подкорректировать сегменты, чтобы они соединялись в игровом движении.

В играх для комфортного прохождения важна производительность. А значит, остро стоит вопрос оптимизации. Анимации в 2D-играх являются важной статьей расходов ресурсов устройства. Аниматор всегда должен следить за тем, чтобы его анимации были оптимизированы.

БИБЛИОТЕКА ВИЗУАЛЬНЫХ ОБРАЗОВ, ВООБРАЖЕНИЕ

В вакансии аниматора часто можно встретить абстрактные требования, например, такие как креативность и воображение. Но эти навыки нельзя измерить. Мы можем понять, насколько хорошо человек разбирается в компьютерной программе. Можем проверить его на знание законов анимации и правил композиции. Но как понять, креативный он или нет? И насколько хорошее у него воображение?

Я считаю, что эти навыки развиваются с насмотренностью. Обладая богатой библиотекой образов в голове, аниматор во время работы каждый раз обращается к своим чертогам разума и ищет там подходящие примеры. А затем обращается к чертогам Google и там тоже ищет примеры, чтобы расширить свою библиотеку образов. При помощи комбинации внутренних и внешних референсов создается образ требуемой анимации. Аниматор должен следить за новинками в своей сфере деятельности, играть в интересные с точки зрения анимации игры, смотреть анимационные картины, фильмы. Разбирая готовые работы, подмечая приемы и техники анимации, мы развиваем воображение и креативность. Мы наполняем мозг данными, на основе которых он будет строить новые образы. Нельзя создать образ из пустоты. К работе воображения всегда подключается то, что мы когда-либо видели. Ну и к каждой задаче нужно подбирать референсы извне. Не стоит надеяться только на свою насмотренность.

Чем занимается 2D-аниматор

Спектр задач 2D-аниматора очень широк. В этой книге я делаю упор на сферу гейм-дева, так как в этой области у меня больше всего опыта. Но спектр задач аниматора не ограничивается игровой индустрией. Вы можете найти себя в мультипликации, рекламе или сфере разработки приложений.

Первое, что приходит на ум при упоминании 2D-анимации, это мультипликация. Например, классические мультфильмы Disney или «Союзмультфильма», нарисованные с любовью к каждому кадру. В современной мультипликации покадровая анимация далеко не всегда является чисто покадровой. Зачастую там, где это возможно, используется скелетная, перекладная мультипликация или даже подключается 3D. В мультипликации 2D-аниматор, который занимается скелетной анимацией, может быть занят созданием скелетов для объектов и персонажей. Он может дополнять покадровые анимации промежуточными кадрами или даже полностью анимировать отдельные элементы или сцены. Задачи по созданию визуальных эффектов также являются важной частью работы 2D-аниматора.

В сфере **рекламы** часто используются анимационные вставки или отдельные персонажи. А значит, и тут много работы для 2D-аниматора. А также есть спектр задач по анимированию элементов рекламного ролика, например, текста. 2D-аниматор может создавать эффекты в ролике для большего вовлечения зрителя.

Аналогичная ситуация и в **разработке приложений**. Персонаж-помощник, анимированный интерфейс, текст и иконки, эффекты, обучающие анимационные вставки — работа для 2D-аниматора здесь обязательно найдется.

Ну и, конечно же, **игровая индустрия**. Это просто кладезь интересных задач для 2D-аниматора, в частности, для скелетного. Именно в играх постоянно встречается скелетная анимация всего и вся. Да и анимаций в играх очень много, а значит, работа у аниматора будет всегда.

Один персонаж может ходить, бегать, прыгать, делать подкат или рывок, атаковать несколькими разными способами, присесть... А персонажей может быть несколько, и у каждого — ряд визуальных вариаций. Анимаций на одно движение тоже может быть несколько — это нужно для того, чтобы разнообразить персонажа, сделать его более живым. Многие анимации сопровождаются эффектами. А сколько у главного героя может быть врагов! И каждому нужен свой набор анимаций и эффектов!

Локация также не строится только за счет статичных элементов. Даже если вся природа не анимируется, движения нужны элементам, с которыми взаимодействует игрок: дверям, кнопкам, рычагам, лифтам, уничтожаемым объектам. А если бюджет проекта чуть больше стоимости батона хлеба, то природные объекты также будут приведены в движение. Цветы, лианы, листья деревьев, кусты, да и вообще любая растительность — все это работа для аниматора.

Интерфейс в играх очень объемен и богат на движущиеся элементы. Бесчисленное количество иконок, всплывающих окон, анимированного текста и плашек. Для крупного проекта нужен целый отдел аниматоров, чтобы справиться со всеми задачами.

Скелетная анимация в игровой индустрии в 2D-проектах приоритетна. Поскольку задач очень много, то остро стоит вопрос соотношения цены, качества и времени. И именно скелетная анимация может его обеспечить. Покадровая слишком долгая и дорогая. 3D может не подходить по стилю и быть слишком затратной по времени и ресурсам.

Spine 2D — основная программа для игрового аниматора. Как Photoshop для художника. Эта программа обеспечивает хорошую связку с игровым движком и удобный экспорт анимации в проект. В наших проектах 90% анимаций сделано именно в Spine 2D. Остальные десять — во встроенной программе для анимации в Unity. Встроенную программу мы используем для простейших движений на фоне или среднем плане локации. Работа со встроенной системой может дать чуть лучшую оптимизацию, но для сложных комплексных анимаций ее использовать неудобно. Spine 2D намного лучше справляется со сложными задачами.

Также Spine 2D постоянно обновляется и пополняется новыми фишками, упрощающими работу. Остальные программы обычно сильно отстают от нее, заимствуя и добавляя функции намного позднее.

ГЛАВА 1

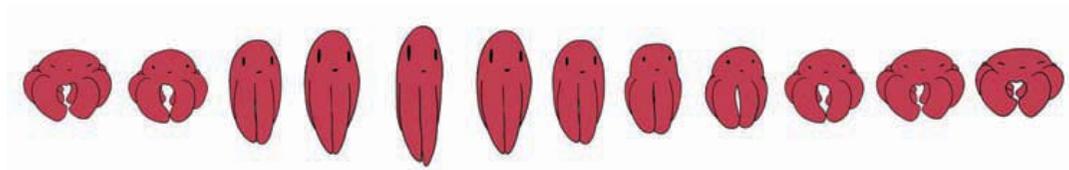
Виды анимации

Покадровая анимация

Первой на ум всегда приходит классическая покадровая анимация. Именно она знакома всем нам с детства по работам студии Disney или советским мультфильмам. Каждый кадр в такой анимации нарисован вручную. Анимация происходит за счет быстрой смены статичных кадров, что для наших глаз создает иллюзию движения на экране. Зрительный аппарат не способен различить отдельные статичные кадры при большой скорости их замены. И нам кажется, что объект движется, хотя фактически мы видим смену статичных картинок.

Плавность движения достигается увеличением количества кадров в секунду. Но чем больше кадров в секунду, тем больше задач у аниматора. Даже если учесть, что иногда кадры можно создавать за счет движения или небольшого изменения предыдущего кадра, эта работа все равно очень долгая, сложная и кропотливая.

В классической анимации допустимо использовать 12 кадров в секунду, дублируя каждый, чтобы получить заветный стандарт в 24 кадра.



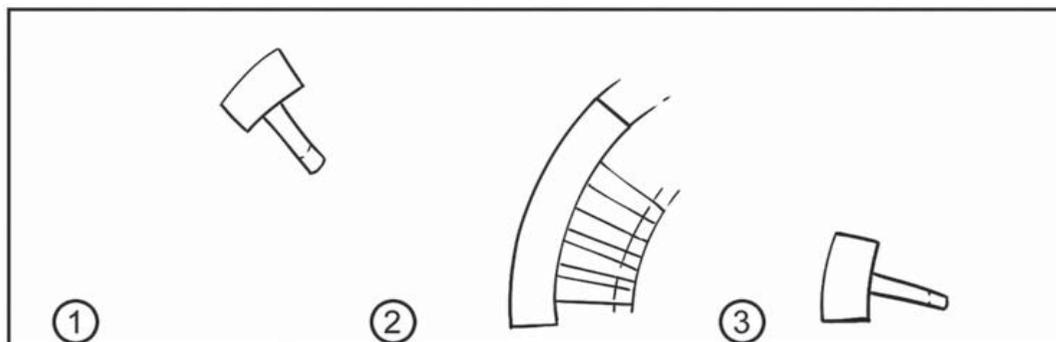
Пример секунды анимации простого персонажа с количеством кадров 12

В отдельных случаях используется и меньшее количество кадров, но тогда эффект отдельных слайдов уже может быть заметен. Но такой эффект применяется для стилизации определенного движения и проекта в целом.

С помощью других видов анимации и современных компьютерных программ можно дотягивать количество кадров до 60, что дает очень плавную картинку.

С помощью количества кадров на анимацию мы можем управлять резкостью движения. Для резкой смены положения используется прием размытия в одном или нескольких кадрах. По сути, этот прием представляет собой объединение нескольких положений элемента в одно.

Если смещение элемента происходит очень быстро, то его можно передать практически одним кадром. В этом кадре будет содержаться весь путь элемента в растянутом виде.



Второй кадр совмещает в себе несколько кадров для передачи скорости удара молотка

Сейчас для покадровой анимации существует множество программ, упрощающих ее создание, таких как TV Paint, Toon Boom Harmony или Adobe Animate. Но этот вид анимации все еще считается самым сложным из 2D-видов и самым затратным по времени и стоимости. В основном классическая покадровая анимация сейчас используется в сочетании с другими видами и способами анимирования объектов. Например, в сочетании с 3D, как в мультфильме «Klaus». Сочетание скелетной анимации, анимации деформацией и покадровой анимации можно увидеть в таких популярных мультфильмах, как «Рик и Морти» или «Время приключений».

В программе Spine 2D, которую мы рассматриваем в этой книге, также можно использовать покадровую анимацию, сменяя картинки с движением времени. А также программа позволяет увеличивать количество кадров в секунду за счет подключения других средств анимации и упрощать задачу создания промежуточных кадров. Это снижает затраты времени аниматора и добавляет плавности вашей работе.

Плюсом данного вида анимации является полный контроль над видом отдельных кадров. Мы не ограничены ничем, кроме умений художника-аниматора. И это в общем-то как плюс, так и минус, потому что все ложится на плечи художника. Только он ответственен за качество анимации. Классическая анимация всегда будет актуальна, как минимум в сочетании с современными средствами ее создания. Главный же минус в том, что нужно создавать каждый кадр (в чистом классическом варианте покадровой анимации), а это долго и дорого. Поэтому все чаще приоритет отдается скелетной анимации, анимации перекладкой или деформацией.

Деформация

По аналогии с анимацией перекладкой мы можем не двигать, а деформировать элементы и таким образом создавать новые кадры. Этот тип анимации можно использовать для движения тканей, растительности, волос и других подобных элементов. Также деформацию применяют для анимации неразделенных элементов картинки. Например, если мы не нарезали руку персонажа на 2–3 части, разделив ее по суставам, то мы сможем согнуть целостную руку с помощью деформации.

Для деформации элементов в современных программах есть множество инструментов. Иногда суть деформации скрыта во внутренних алгоритмах программы, а нам даются лишь ключевые точки для искажения. Иногда мы можем создать специальную сетку деформации на картинке (меш) и управлять ее отдельными точками, менять эти точки, задавать их влияние на изображение.

Сочетание деформации и перекладки открывает довольно широкие возможности для анимации. Деформация компенсирует основной недостаток перекладки — статичность отдельных элементов.

Современные программы для анимации, в частности и Spine 2D, позволяют использовать эти два способа анимации не как стоп-моушен (создание каждого отдельного кадра перекладкой и деформацией), а как автоматическое заполнение промежуточных кадров. Например, мы хотим анимировать сгиб руки. По принципу стоп-моушен нам нужно в каждом кадре поворачивать руку на определенный угол и каждый раз фиксировать это положение для кадра. В Spine 2D мы можем повернуть руку из начального положения в конечное и задать лишь два ключевых кадра. Промежуток программа заполнит сама. Скажем так, просчитает вращение элемента и заполнит недостающими кадрами.

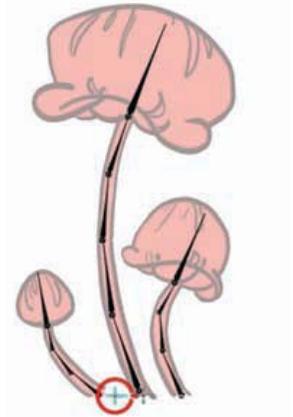
Скелетная анимация

Наконец мы подошли к главному виду анимации, который описывается в этой книге. Скелетная анимация, как следует из названия, подразумевает анимацию за счет скелета, построенного в программе.

Мы создаем группу взаимосвязанных костей и прикрепляем их к картинкам.

Анимировать мы будем именно кости, а картинки последуют за ними, согласно установленным нами правилам. Кости можно создавать с определенной иерархией, по аналогии с деревом. У нас всегда есть первоначальная кость, корень нашего дерева. В Spine 2D она так и называется — *root*. От нее отходят дочерние кости, от дочерних — уже их дочерние кости, и т. д. Изменяя положение родительской кости, мы изменяем положение всех ее дочерних элементов.

Мы можем прикрепить несколько костей к одной картинке, создать для нее меш и затем, анимируя кости, деформировать элемент. А можем, наоборот, к одной кости прикрепить сразу несколько элементов. С помощью костей создаются волновые движения, эффект псевдо-3D. Возможности этого типа анимации очень широки, особенно в сочетании с деформацией и перекладкой. А если для некоторых состояний объекта подключить еще и покадровую анимацию, получится отличный плавный результат. А главное — мы можем настроить количество кадров больше 24 в секунду автоматически с помощью программы. Это дает весомый аргумент в пользу покадровой анимации.



Пример простейшего скелета

Но поскольку для анимации мы все-таки используем мозаику из отдельных кусочков, их статичность все еще остается главным минусом данного подхода. Особенно в сравнении с покадровой анимацией. От некоторой «роботизированности» скелетной анимации избавиться довольно тяжело.

3D-анимация

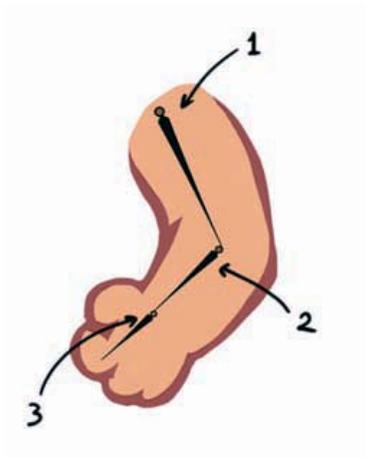
Отдельно стоит упомянуть 3D-анимацию. Это абсолютно другой мир, и в рамках этой книги мы не будем в него погружаться. 3D-анимация очень популярна в индустрии кино, мультипликации, 3D-играх. Но она требует огромных затрат времени в сравнении с 2D-анимацией. Это проявляется уже на этапе создания персонажа или объекта, еще до анимации. Сначала нужно сделать 2D-концепт, затем 3D-модель, текстуры, подготовку модели к анимации и только после всего этого приступить к созданию движения. Также для 3D-визуализации и итогового рендера нужны машинные мощности, в которых нет необходимости в 2D. И поскольку работы проводится больше, специалистов задействуется больше — затраты на 3D-анимацию в разы больше. Это делает качественную 3D-анимацию прерогативой крупных студий и больших проектов за редким исключением.

Но не стоит забывать, что можно комбинировать 2D и 3D, использовать 3D для улучшения 2D-анимации или анимации отдельных простых для моделинга элементов. Например, бывает сочетание 3D-фонов и анимации на них с 2D-персонажами. Или даже 3D-пушка в руках у 2D-персонажа.

Сочетание нескольких видов анимации в Spine 2D

Как я уже говорила, в современной индустрии анимации в большинстве случаев для более быстрого и качественного результата используется сразу несколько видов анимации. В Spine основным видом является скелетная анимация, но есть возможность использовать покадровую, перекладки и деформацию. Но и перекладки, и деформация, и смена кадров все равно осуществляются с помощью прикрепленных к картинкам костей. Поэтому скелетная анимация главенствует над остальными типами в программе Spine.

Скелетная анимация осуществляется следующим образом. У нас есть рука персонажа, разделенная по суставам на три части: кисть, предплечье и плечо. Чтобы иметь возможность двигать этот объект, мы должны создать соответствующие кости. Главная кость начинается в плечевом суставе и идет по плечу до локтя. Следующая — от локтя до запястья, причем она является дочерней по отношению к главной. И последняя кость проходит от запястья до конца кисти, являясь «дочкой» предыдущей локтевой кости и «внучкой» главной кости. Теперь, если мы будем двигать плечевую кость, то за ней придут в движение «дочка» и «внучка». Если будем двигать локтевую кость, за ней последует запястная. Ну а крайняя кость на запястье не имеет последующих костей. Длина кости нужна в основном для визуального удобства, для простых анимаций можно создавать кости только точками. Но для продвинутых фишек программы, таких как распределение влияния кости на точки меша или инверсная кинематика, длина кости будет иметь значение. Но это тема будущих глав книги.



1. Главная родительская кость
2. Дочерняя кость
3. Дочерняя к дочерней

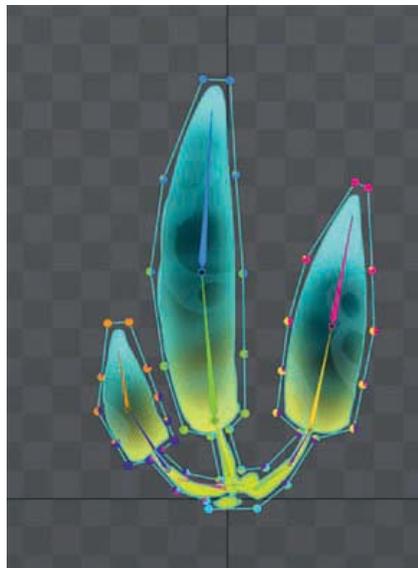
Для деформации элементов необходимо использовать меш. **Меш** — один из ключевых инструментов в Spine. Если мы хотим деформировать, исказить или создать эффект псевдо-3D, нам нужно создать меш и совместить его с картинкой. Меш — это набор вершин и многоугольников, определяющих форму объекта.

Мы можем деформировать меш, а значит, и картинку, двигая определенные его точки. Но намного удобнее это делать с помощью костей.

Мы можем настроить влияние костей на точки меша и таким образом исказить объект. Чем больше точек на меше мы создадим, тем более плавное искажение получим. При условии правильного положения точек, разумеется. Но с увеличением количества точек усложняется контроль над анимацией.

Покадровая анимация осуществляется сменой видимости картинок, заранее загруженных в проект. В любой момент анимации мы можем включить или выключить определенный сегмент. А значит, можем последовательно менять кадры. По сути, в Spine 2D используется заранее нарисованная покадровая анимация с возможностью подтянуть недостающие кадры за счет деформации или смены видимости через прозрачность. В самой программе возможности рисовать кадры нет.

Обычно переключение кадров используется для создания эффектов, изменения выражения лица персонажа, смены ракурса определенного фрагмента.



Плюсы и минусы скелетной анимации

У скелетной анимации есть свои плюсы и минусы, и уже исходя из них определяется спектр задач, для которых она подходит больше всего.

Главным плюсом является **переиспользование скелета и настроек**. Мы создаем скелет и настраиваем все дополнительные параметры один раз. А затем можем делать сколько угодно анимаций на основе наших настроек. Можем даже менять картинки, привязанные к этому скелету. Таким образом мы получаем множество одинаковых анимаций для разных персонажей, которые двигаются по правилам одного скелета. Это существенно упрощает работу с большим количеством похожих задач, что отлично подходит для игровой индустрии. Даже если нельзя напрямую использовать тот же самый скелет для новой анимации или персонажа, мы можем создать новый проект на основе старого и отредактировать скелет. Соответственно, нам не нужно каждый раз делать работу с нуля.

Из этого плюса вытекает и минус — **длительная подготовка к анимации**. Сначала нам нужен исходник, который будет правильно нарезан и удобен в использовании для анимации. А значит, необходимо не только нарисовать работу, но и разобраться с ее делением на части. Сгладить этот момент помогает сочетание мешей и скелета. Такая комбинация позволяет избежать мелкой нарезки и оставить слитными некоторые элементы в исходнике и анимировать их деформацией с помощью меша.

Далее мы должны ответственно подойти к созданию скелета и настройкам, так как это наш фундамент. Если допустить ошибку на этом этапе, рухнет весь дом.

К плюсам можно отнести **варьируемое количество кадров**. Мы только задаем само движение, его правила и законы. Заморачиваться с каждым кадром нам не нужно. Мы работаем с движением в целом, а программа решает вопрос количества кадров и плавности анимации. Поскольку современные игроки хотят даже не 30, а 60 и более кадров в секунду, этот момент очень важен при анимации. Мы уже не можем позволить себе 12 кадров в секунду, как в классической анимации. Разве что в стилизованных под мультипликацию проектах.

Скелетную анимацию можно **применять для различных задач**, несмотря на то что обычно она ассоциируется с персонажной анимацией, так как именно у персонажей очевиден скелет. Мы можем использовать скелетную анимацию и для природных элементов, и даже для анимации видеороликов и постановки сцен. Схема «кость — картинка — движение» применима практически к любым задачам. А значит, все анимации в проекте могут быть выполнены в едином стиле с использованием скелетной анимации. На протяжении всей работы можно не скакать из одной программы в другую, а делать анимацию только в Spine 2D.

Скелетная анимация **хорошо интегрируется в игровой движок**. В Unity есть набор средств для работы со скелетной анимацией и с анимацией, импортированной из Spine 2D. В самом Spine также богатый выбор вариантов экспорта и представления анимации.

Главным минусом скелетной анимации является **работа в заданной плоскости**. Если нам нужен поворот героя на зрителя, выпад, атака — все что угодно, то сделать это движение в рамках скелетной анимации довольно тяжело. У нас есть набор заданных картинок в плоскости с определенным ракурсом. Двигать, поворачивать и масштабировать мы можем также в рамках этой плоскости и ракурса. Для выхода за пределы плоскости нам нужно использовать эффект псевдо-3D. Но он довольно сложен в реализации и требует дополнительных исходников. Или мы можем прибегнуть к покадровой анимации и добавить отрисовку новых элементов в нужном ракурсе. Но это приведет к затратам ресурсов художника и тому, что исходник станет более сложным и массивным, с большим количеством разных ракурсов, которые аниматору будет трудно контролировать. Дополнительные материалы подразумевают дорисовку отдельных ракурсов, элементов, искажений, а это требует дополнительных затрат времени и сил художника.

ГЛАВА 2

Почему именно Spine 2D

В собственных играх для всей анимации, кроме игр в стиле пиксель-арта, мы используем Spine 2D. Мы пришли к нему не сразу. Отпугивала цена лицензионной версии и потребность изучать еще одну программу. Скелетную анимацию мы выбрали изначально как основную. На проектах я единственный художник и, кроме того, занимаюсь многими другими вещами. Возлагать на себя еще и огромное количество кадров анимации в такой ситуации не лучшее решение. В самом начале мы пробовали делать анимации с помощью аниматора Unity и даже создали пару локаций только этим инструментом. Но он хорошо подошел лишь для простейших анимаций среды, делать в нем персонажей было долго и неудобно. Аниматор, встроенный в Unity, развивается, в него добавляются новые функции, и интерфейс становится более удобным. Но за Spine 2D ему все еще не угнаться. И, понимая, что мы только тратим время зря, мы решились на покупку Spine 2D, о чем еще ни разу не пожалели. Делая анимацию в Unity, вы сможете использовать ее только в Unity. Если бы мы захотели делать наши игры на Unreal engine, то нам бы пришлось изучать другую программу и переделывать анимации в ней. Создавая анимации в Spine, мы нацелены на любой движок или даже видеоролики и мультипликацию. Мы не ограничены средством разработки.

Следующий важный момент — это возможность экспорта анимаций в виде json-файла и атласа изображений. Эти форматы позволяют использовать анимации в игровом движке без танцев с бубном. Просто и удобно. Подробнее мы рассмотрим эту фишку программы в разделе, посвященном игровым движкам. Остальные популярные программы для анимаций выдают на экспорт либо видео, либо покадровую последовательность, поэтому Spine так популярен в игровой индустрии.

Также стоит сказать про размер файлов после экспорта из Spine. Если мы используем покадровую анимацию, то придется хранить все кадры подряд, а в случае со Spine 2D только атлас и json-файл. Условно наш персонаж в игре имеет сотню анимаций. Если их все показывать покадрово, это займет много места, а со Spine 2D мы будем хранить только его атлас и json-файл. Но есть минус — потом, уже в самой игре, на движение точек меша, просчет масок, движение костей, будет затрачено больше вычислительной мощности и ресурсов компьютера.

У Spine есть бесплатный аналог Dragonbones, который пытается клонировать все функции Spine, при этом не требуя затрат на лицензию. Его основная проблема в том, что разработчики программы давно не поддерживают ее, не добавляют новые функции или добавляют очень редко и в разы меньше, чем в Spine. Программа изначально

была аутсайдером на фоне Spine, а спустя 3–4 года разница между ними стала слишком велика. Также в бесплатном аналоге не такой удобный интерфейс и больше танцев с бубном при экспорте анимаций. Поэтому Dragonbones в основном используется как ознакомительная программа для изучения анимации, а в игровой индустрии практически не применяется.

У программы Spine 2D есть бесплатная версия, в которой люди могут спокойно обучаться всему, кроме возможности сохранения и экспорта. Функциональность программы никак не ограничена. Вы можете попробовать все ее возможности и решить, подходит ли вам программа для работы, чтобы затем купить полную версию с возможностью сохранения файлов.

У Spine 2D низкий порог входа. Чтобы начать делать в Spine простейшие анимации, нужна буквально пара недель изучения интерфейса. Да, за это время крутым аниматором не стать, но первые плоды своего труда можно увидеть. Когда проект уже в работе, это очень важный фактор, потому что разработка не останавливается в ожидании, пока вы поладите с новым инструментом. Если вы уже практикующий специалист, быстрый результат в программе также имеет вес, так как вам не нужно брать творческий отпуск на изучение программы. Вполне достаточно каждый вечер по часику уделять обучению. Если вы только начинаете свой путь, то сможете довольно быстро перейти в разряд junior-аниматоров и уже искать первые заказы. В общем, возможность быстро получить результат от программы — это огромный плюс. А после первых успехов можно уже наращивать уровень своего мастерства и постепенно углубляться в мир скелетной анимации.

Также этот момент очень важен для поддержания мотивации. Сложно не сдать на старте, когда не видишь результата. На то, чтобы изучить окна, кнопки, функции, тратишь много времени, при этом не создаешь ничего, пусть даже минимально интересного. Это что-то вроде рисования кубиков и простых примитивов для начинающего художника. На этом этапе легко потерять интерес к сфере деятельности. Spine же очень быстро увлекает в создание анимации и сразу позволяет делать что-то интересное.

Вторым важным моментом при выборе программы стала игровая направленность. Для игровой анимации Spine — лучший выбор, так как программа заточена именно под нужные задачи. Это не значит, что Spine нельзя использовать для мультипликации или создания видеороликов, но основная сфера применения все же геймдев. Для мультипликации есть Adobe animate или Toon Boom Harmony. В этих программах заложена возможность и создавать покадровую анимацию, и дополнять ее деформацией или перекладкой. Также вы наверняка слышали про программное обеспечение Adobe After Effects, которое часто применяется для моушен-дизайна и видеороликов. Но в этих программах огромное количество функций и разделов под разные задачи, что усложняет старт работы. Программа для всего и сразу не может раскрыть определенный сегмент сферы анимации так же круто, как это делает специализированная программа. Такая как Spine 2D.

Для создания скелетной анимации Spine, на мой взгляд, самый удобный инструмент. Упор на конкретный вид анимации и конкретные задачи позволяет лучше всего раскрыть их в программе. Нацеленность на конкретный тип задач также обеспечивает неперегруженный интерфейс, в котором сложно запутаться. Если вы хотите заниматься скелетной анимацией, особенно если ваша цель — геймдев, то Spine лучший выбор для работы.

Несмотря на четкую нацеленность на скелетную анимацию, Spine обеспечивает широкий охват задач в проекте. Можно все анимации делать с помощью именно этой программы. Покадровая анимация также не забыта в Spine, и там есть функции для работы с ней. И набор инструментов постоянно дополняется. С помощью Spine можно анимировать персонажей, объекты, интерфейс и даже делать сцены для видеороликов. Монтировать сам видеоролик нужно в другой программе, но отдельные вставки можно создавать в Spine. А значит, в работе над игрой не нужно постоянно переключаться с одного инструмента на другой, а можно сконцентрироваться только на Spine.

Команда разработчиков активно развивает программу, а также добавляет большое количество бесплатных обучающих материалов. Добавляя новую функцию, создатели сразу же прикрепляют к ней небольшой урок по использованию. Также у программы отличное руководство пользователя, в котором есть описание инструментов, функций и приемов. На форуме вы сможете получить поддержку и помощь в работе. Сообщество Spine очень приятное и активное. Это отличный бонус к самой программе.

Чтобы определиться с программой для работы, необходимо также учитывать ее распространенность в индустрии. При выборе среды для рисования, например, многие художники и компании предпочитают Photoshop другим аналогичным программам. Так как Photoshop является стандартом и самой используемой средой для создания концепт-арта и иллюстраций. Умение работать в Photoshop — обязательный пункт в резюме каждого художника. Даже если он предпочитает работать в аналоге, при устройстве на работу будет требоваться Photoshop в 90% случаев. Так же и с программой Spine. Если вы ищете работу игровым скелетным аниматором, то знание Spine 2D нужно обязательно. Да, некоторые компании могут работать в собственном анимационном движке, написанном под конкретные задачи компании. Или делать анимации в Unity. Но в 90% случаев требуется Spine. А если даже работа осуществляется не в ней, то знание Spine позволяет с легкостью перейти на любую другую программу для скелетной анимации. В вакансиях скелетного аниматора практически всегда первым требованием указано знание Spine. Рядом с знанием принципов анимации, разумеется. Спрос на специалистов в области скелетной анимации очень высок, многие компании используют для анимации Spine 2D и скелетную анимацию. Spine 2D можно назвать стандартом игровой индустрии.

Возможность использовать повторно материал, созданный в программе, также стала важным пунктом при выборе именно Spine для наших игр. В Spine можно создавать множество анимаций, базирующихся на одной настройке. Иногда даже для разных проектов можно использовать одну настройку бега или прыжка, заменив лишь

визуальную часть. Это очень важно для нашей команды, так как ресурсы ограничены и в сутках все еще 24 часа.

Начать анимировать в Spine под силу каждому, кто мечтает оживлять статичные картинки. С помощью Spine 2D можно в разумные сроки создавать качественные анимации и добавлять их в игры или другие проекты.

Spine стала неотъемлемой частью нашей работы и очень помогла оживить игры и персонажей. И в этой книге я постараюсь привить вам такую же любовь к этой программе в частности и к скелетной анимации в целом.

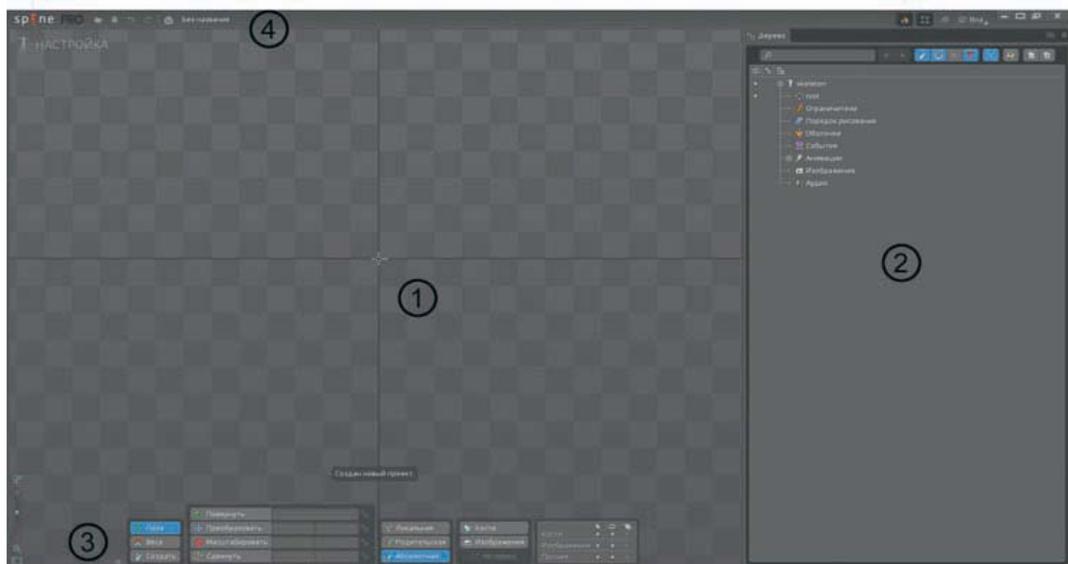
ГЛАВА 3

Spine 2D: знакомство с программой

Давайте наконец познакомимся с программой. Обсуждать, какая же она крутая, это, конечно, хорошо. Но только оценив все преимущества Spine на практике, вы сможете причислить ее к основным средствам своей работы.

Интерфейс

Давайте же кликнем на заветный ярлык, который будет нашим порталом в мир скелетной анимации, и запустим программу. При открытии программы мы видим перед собой четыре области интерфейса.



Создан новый проект

1. Самая большая и заметная область — окно с шахматным паттерном, на котором изначально стоит одна кость, там же будет отображаться наш объект анимации. Это основное окно, и на нем мы будем настраивать кости, меши и другие необходимые нам вещи. Колесом мыши мы можем менять масштаб, а с зажатой правой кнопкой мыши перемещать область видимости.

2. В правой части мы видим своеобразные папки с иерархической структурой. Там будут отображаться слои нашего исходника в последовательности наложения. В этом окне мы найдем и кости, привязанные к слоям, упорядоченные согласно иерархии. Во вкладке оболочки хранятся скины персонажа. Скины — это различные вариации картинок на один и тот же скелет. Там же будут сгруппированы все наши анимации, которых в одном проекте может быть много.
3. В нижней части находятся панель создания костей, весов (влияния костей на точки меша), инверсной кинематики и стандартные средства трансформации: перемещение, масштаб, поворот. Тут мы можем настроить вид осей трансформирования и включить фильтры отображения в окне видимости.
4. В самом верху мы видим стандартное меню с выпадающими плашками. С помощью него мы можем открыть новый проект, сохранить проект, экспортировать анимацию, отменить и повторить действие.

Обратите внимание на надпись в левом верхнем углу экрана «Настройка». Если на нее нажать, окно программы перейдет в режим анимации и интерфейс слегка изменится. Надпись сменится на «Анимировать», и в нижней части экрана появится шкала времени, на которой мы будем задавать ключевые кадры.

Изменения, сделанные в окне «Настройка», меняют скелет и влияют на анимацию. Изменения в окне анимации влияют только на текущую анимацию.

Некоторые дополнительные окна можно вызвать в выпадающем меню «Вид» в правом верхнем углу. Чтобы закрыть окно, нужно щелкнуть правой кнопкой по иконке в верхнем правом углу окна. Потянув за верхнюю плашку с названием окна, вы можете переместить его в удобное для вас место. Программа позволяет настроить рабочую область так, как вам хочется. Когда вы наберетесь опыта, то поймете, какие функции чаще всего используются и какие окна нужны вам в постоянном доступе.

Общий алгоритм анимирования в Spine 2D

Весь процесс создания анимации в Spine можно разделить на несколько этапов.

1. Самый первый пункт в работе над анимацией — это **подготовка исходника**. Нарезка цельной картинкой на части. На данном этапе важно заранее спрогнозировать движение персонажа насколько это возможно, определить, какие части будут двигаться, а какие нет. Необходимо решить, какие элементы мы будем анимировать деформацией, а какие нарежем на части. Также важно не забыть дорисовать недостающие элементы, которые становятся видны при движении.

Некоторым объектам нужно добавить тень на отдельном слое, чтобы движение было более естественным, а тень двигалась за объектом. Если на элементах сгиба есть обводка или орнамент, эти моменты нужно проработать отдельно. Необходимо проверять движение элемента в Photoshop и следить, чтобы орнамент не терялся или обводка не нарушалась. Этап очень объемный и важный. Далее этому будет посвящена отдельная глава, в которой мы рассмотрим основные проблемы подготовки к анимации.

2. Импортировав наш исходник в программу с помощью скрипта Photoshop to Spine или вручную, мы приступаем к **созданию скелета**. Этот процесс называется **риггинг**. На этом этапе мы проектируем связи костей, привязываем их к картинкам. Здесь важно правильно сделать структуру частей персонажа. Особое внимание нужно уделить главным и второстепенным элементам и их связи друг с другом. Например, чтобы голова и руки двигались вместе с туловищем, а волосы и глаза — вместе с головой. Проектируя поэтапно, от общего к частному, мы должны получить скелет, способный обеспечить требуемые движения.
3. Далее нужно **настроить меши**. Этот этап называется **скиннинг**. В тех случаях, где нам нужна деформация или эффект псевдо-3D, а также если мы по каким-то причинам не разрезали фрагмент на части, мы должны создать меш. Затем необходимо настроить влияние костей на его точки, проверить, что все гнется так, как нам нужно, и меш максимально оптимизирован. Оптимизация меша подразумевает наименьшее количество точек, обеспечивающее плавный сгиб и требуемую деформацию.
4. И только после всего этого мы переходим к **анимации**. К анимации лучше подходить от общего к частному, как и в случае со скелетом. Обычно мелкие второстепенные элементы следуют за более крупными и важными. Например, анимация волос должна быть сделана после анимации головы и туловища. Волосы зависят от головы, голова от туловища, а значит, сначала анимируем туловище, потом голову, потом волосы и другие аналогичные второстепенные части. У любого движения есть основа, а есть вспомогательные анимации. Например, ходьба или прыжок — это в первую очередь движение ног и туловища, а движение рук уже второстепенно, не говоря о волосах или одежде.

Основные инструменты

Прежде чем перейти к практическим упражнениям, остановимся на основных инструментах, которые понадобятся вам при первом знакомстве со Spine 2D. Не будем сразу углубляться во все функции программы. Их довольно много, и, если окунуться в это море сразу с головой, можно не усвоить информацию. Поэтому начнем с базы и будем постепенно, по мере необходимости, расширять спектр изучаемых панелей и инструментов.

Начнем с **создания костей**. Для этого используется первая из нижних панелей под рабочей областью. С помощью пункта «Создать» можно добавлять новые кости в скелет.

Создаваемая кость автоматически становится дочерней к выделенной кости. Выделенная кость подсвечивается голубым цветом на рабочей области. Чтобы привязать кость к другой родительской кости, нужно сначала кликнуть левой кнопкой мышки на нужную кость, а затем создавать новую. При наведении на кости красными пунктирными стрелками отображаются привязки костей к родительским, что очень удобно для отслеживания иерархии костей. Кость создается кликом мышки и вытягиванием в нужную сторону. При простом клике кость создается в виде точки.

Чтобы при создании кости сразу привязать ее к конкретной картинке, нужно выделить эту картинку щелчком мыши с зажатой клавишей Ctrl. Картинка подсветится голубым, так же как и кость.

Инструментом «По́за» мы можем корректировать положение и поворот кости.

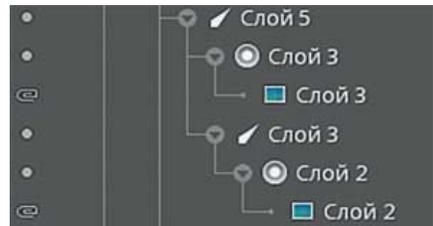
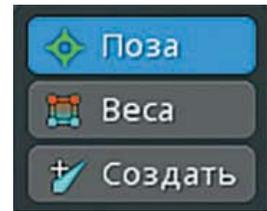
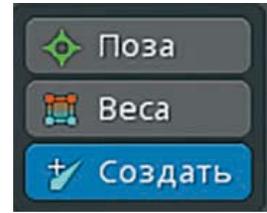
Причем вместе с костью будут менять положение и все ее дочерние элементы. Можно корректировать области скелета целиком. Взаимодействие с основанием кости перемещает ее, а работа с кончиком кости поворачивает саму кость и дочерние к ней.

В процессе создания скелета мы также можем управлять положением костей с помощью инструментов поворота, перемещения и масштаба, которые находятся на второй нижней панели. Но эти инструменты корректируют каждую кость отдельно, не затрагивая дочерние элементы.

Также обратите внимание на третью нижнюю панель с вариантами осей. Когда наш скелет разрастется до десятка костей, эта панель очень пригодится. Локальное и глобальное положение костей отличается, и для разных операций может быть удобно менять это положение.

Посмотрим подробнее на **структуру проекта** на панели справа. Если проект состоит из нескольких картинок и костей, то образуется структура следующего вида.

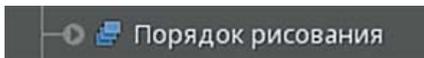
В этой структуре мы можем видеть кости с отношениями, которые показаны линиями перед наименованиями. Например, кость «Слой 3» является дочерней по отношению к кости «Слой 5». Названия костей аналогичны названию картинок в проекте. Внутри каждой кости у нас слот, в котором содержится изображение. Кликая на серые кружочки справа,



мы можем отключать слот или кость вместе со слотом. В этой панели мы можем определять структуру проекта не только при создании костей, но и после. Перетаскивая слоты, кости и картинки в разные места структурного дерева, мы можем менять привязки костей к картинкам и друг к другу. Это очень удобно для корректировки скелета на позднем этапе, когда все кости созданы и привязаны к картинкам.

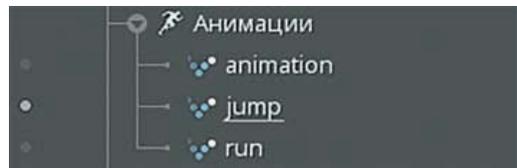
В слоте может быть не одна картинка, а несколько. И мы можем кость привязать к нескольким картинкам. Это используется для контроля сразу нескольких изображений либо для покадровой анимации. Все кадры подгружаются в один слот, и переключением видимости изображений мы можем создавать покадровую анимацию в Spine. Или добавлять несколько ракурсов одному элементу и переключать их в процессе анимации. Намного удобнее хранить кадры в одном слоте, а не в разных. Это позволяет избежать расширения структуры проекта без надобности, а также группировать кадры для простоты поиска и работы с ними.

Чуть ниже в этой панели есть вкладка «Порядок рисования».



Открыв ее, мы увидим все наши слоты. Порядок отображения слотов в этой вкладке совпадает с их положением на рабочей области. Чем ниже слот, тем ниже он и на рабочей области. А вышестоящие в этой вкладке слоты перекрывают нижние. Перетаскиванием слотов в этом выпадающем меню мы можем менять перекрытие элементов в проекте.

Еще ниже во вкладке «Анимации» находятся все анимации для скелета в данном проекте. Когда мы выберем анимацию, она отобразится в рабочей области, и мы будем работать именно с ней.



Также мы можем продублировать анимацию, переименовать или удалить с помощью соответствующих иконок ниже.



И последнее, на что стоит обратить внимание на этой панели в начале работы с программой, — пункт «Изображения». Там находятся все картинки проекта, даже те, которых нет на рабочей области. Мы можем добавить картинки в папку проекта, и они появятся в этом меню. А далее мы можем перетащить их на рабочую область и использовать в анимации. Для анимации не всегда нужен весь набор картинок. Их может быть больше, и в зависимости от ситуации мы будем использовать те или иные.

График характера движения

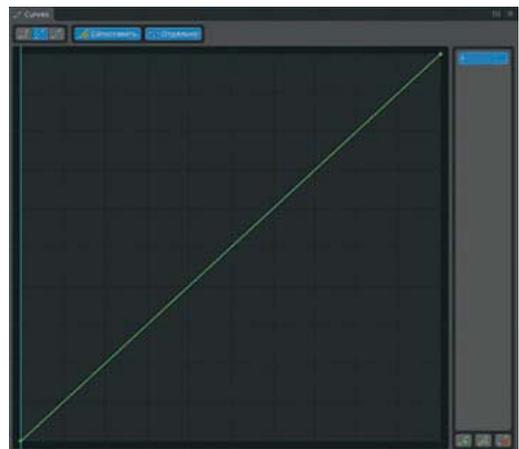
Обязательно нужно остановиться на теории о графике движения в программе Spine 2D. Это очень важный и довольно сложный для понимания инструмент программы. Именно с помощью графика мы можем контролировать движение между заданными ключами. В покадровой анимации мы регулируем скорость движения с помощью плотности кадров. Например, для замедления в начале и конце движения рисуется больше кадров около стартового и конечного положений объекта. В середине движения кадры проскакивают быстрее, их меньше, а иногда вообще используется один кадр с размытым движением. В скелетной анимации мы задаем ключевые положения, а остальное решаем с помощью графика.

График можно настраивать для любого вида ключей. Для перемещения, поворота, масштаба и сдвига. Это добавляет возможности контроля над различными состояниями объекта. Мы можем назначить разные графики для разного вида ключей. Например, объект будет двигаться с замедлением, а расширяться с ускорением в начале. Также мы можем буквально парой кликов усилить или ослабить заданный эффект, в отличие от покадровой анимации, где корректировка требует перерисовки кадров.

А теперь нам нужно вспомнить школьные уроки алгебры и построение графиков функций. У нас есть две оси. По горизонтальной оси расположено время между ключевыми кадрами, которое в Spine можно соотнести с промежуточными кадрами. Начало графика — первый ключ, конец графика — второй. По вертикальной оси расположено изменение параметра (перемещение, масштаб, поворот, сдвиг). В начальной точке — значение параметра в первом ключе, в конечной точке — значение параметра во втором ключе.

У нас есть три варианта графика. Ступенчатая кривая, линейный график и кривая Безье. Эти три вида соответствуют трем иконкам в верхнем левом углу окна графика.

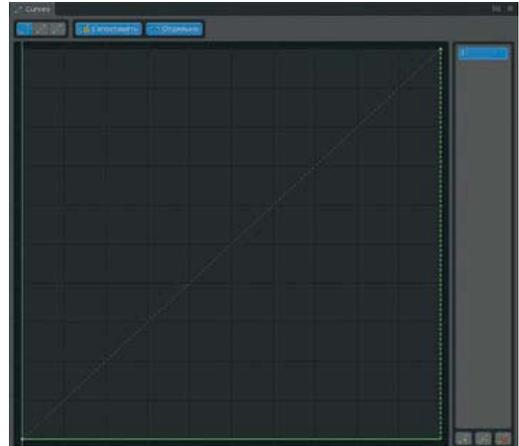
Самый простой вариант (вторая иконка) — равномерное движение, прямой график изменения параметра. Этот вариант графика означает, что изменение параметра происходит одинаково на каждом участке движения. Нет замедления или ускорения. Но это одновременно и самый редко используемый вид движения. Равномерное движение редко встречается в анимации. Оно может использоваться, если в анимации нет начала и конца движения. Например, машинка едет с одинаковой скоростью на протяжении всего времени анимации, а старта и торможения в анимации



Равномерное изменение

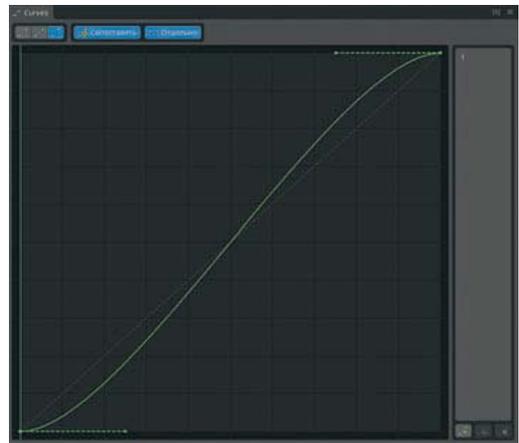
нет. Либо для движения не по прямой, а по кругу, или для вращения. При этих движениях также нет момента старта и окончания. Примерами служат вращение колеса или лопастей мельницы. Хотя и в этих случаях можно разнообразить движение скачком колеса на кочке или ускорением-замедлением лопастей в моменты порывов ветра.

Теперь посмотрим на график мгновенного изменения параметра — ступенчатую кривую. Это первая иконка в верхнем левом углу окна графика. График показывает нам, что вплоть до финального кадра параметр не изменяется, оставаясь в начальном положении, а затем резко меняет значение на финальное. Такой тип графика дает нам, по сути, эффект покадровой анимации. Один кадр мгновенно сменяется другим без промежуточного эффекта.



Ступенчатая кривая

И самый интересный и часто используемый график — кривая Безье. Именно с его помощью мы можем задавать замедление в начале и конце. Редактируя кривую, мы можем убирать замедление в одной из точек, настраивать степень этого замедления и ускорения соответственно. График может приобретать самые причудливые формы и задавать абсолютно разный характер движения.



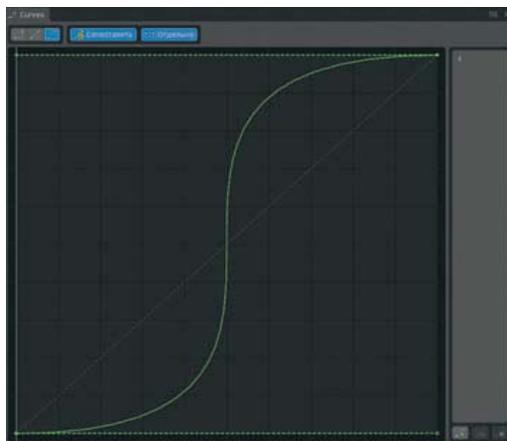
Базовый вид кривой с замедлением в начале и конце

В своем базовом формате, который создается при щелчке на третью иконку в верхнем левом углу окна, график лишь слегка отличается от равномерного. Добавляется небольшое замедление в начале и такое же в конце. Если сравнить этот график с прямой, то мы увидим, что в начале при прямолинейном движении изменение параметра происходило быстрее. За счет изгиба графика вниз в начале параметр изменяется медленнее, к середине графика догоняя значение равномерного движения. После замедления всегда следует ускорение, график довольно быстро растет вверх. А изгиб графика вверх в конце снова замедляет изменение параметра.

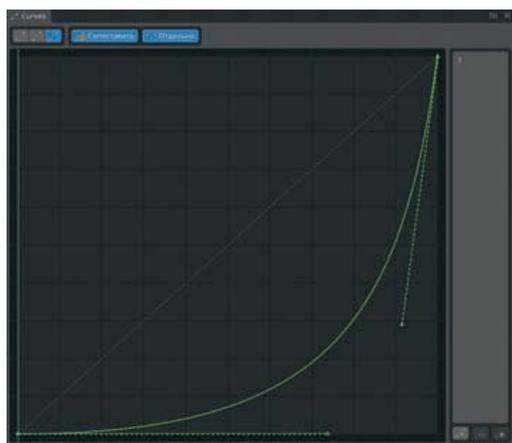
Самый утрированный вид этого графика будет близок к ступенчатой кривой в середине движения. Если мы потянем за опорные точки и преобразуем график в сильное замедление в начале и конце, то в середине движения график станет вертикальной

прямой. А это означает мгновенное изменение параметра в этой области. Такой график будет соответствовать движению, при котором объект очень медленно стартует, а потом резко меняет положение и долго тормозит.

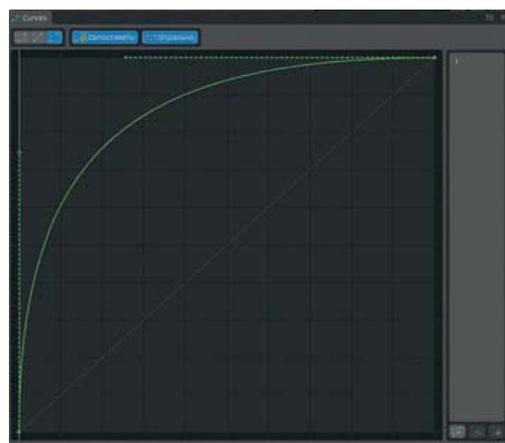
Также часто используется вариант графика без замедления в одной из точек, когда нам нужно, чтобы объект разогнался и выскочил из кадра. Например, чтобы снаряд резко вылетел из пушки на определенное расстояние, а затем затормозил. Такой график нам также пригодится при создании прыжка мячика в одном из последующих упражнений.



Сильное замедление в начале и конце с мгновенным скачком параметра в центре

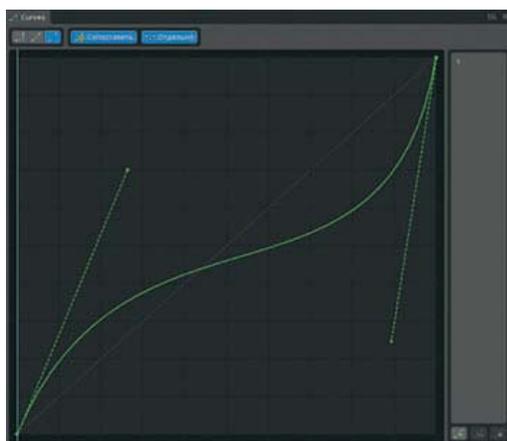


Графики замедления в начале и ускорения в конце и, наоборот, ускорения в начале и замедления в конце



С помощью кривой Безье можно задать довольно причудливые формы графика движения для различных специфических ситуаций в анимации. С помощью графика мы можем контролировать движение и искажать его как нам угодно.

В окне графика есть еще несколько важных кнопочек, значение которых нужно понимать для работы с графиками. Лучше оставить включенными обе кнопки,



Нетипичная форма графика движения

потому что тогда график дает более понятный и логичный результат, а по мере необходимости отключать их.



«Сопоставить» при включенном состоянии позволяет настраивать график сразу для всех выделенных ключей. Если кнопка отключена, то редактироваться будет только первый выделенный ключ. Возможность редактирования сразу нескольких ключей очень полезна, так как часто нам нужно задать одинаковый график для целой группы движений.

Кнопка «Отдельно» позволяет редактировать только фрагмент между выделенным ключом и последующим за ним, не затрагивая соседние графики. По умолчанию график всего движения представляет собой единый график, который делится ключами на сегменты. Меняя один сегмент, мы меняем и сегменты рядом, так как график непрерывен. Включив кнопку отдельно, мы будем работать только с нужным нам графиком.

Обратим внимание на боковую панель окна графиков. Это окно шаблонов. В анимации довольно часто используется один и тот же график. Нам не нужно каждый раз настраивать его и запоминать показатели, программа сделает это за нас. Мы можем задать шаблон на основе редактируемого графика. Это делается нажатием иконки с плюсиком внизу этой панели. И шаблон графика появится в списке с заданным названием. Таких шаблонов можно создавать сколько угодно и использовать их одним нажатием в дальнейшем.



Шкала времени

Познакомимся поближе с основным окном анимации, помимо непосредственно рабочей области, разумеется. В этом окне отображаются все заданные нами ключи для каждой кости. Когда у вас будет сложная составная анимация с множеством костей, то шкала вытянется вниз. Используя колесико мышки, вы можете прокрутить список костей и ключей вниз. Можно менять порядок этих строк, перенося выше наиболее важные и используемые кости. Просто потянув одну кость и отпустив в нужном месте, вы измените порядок по вертикали.

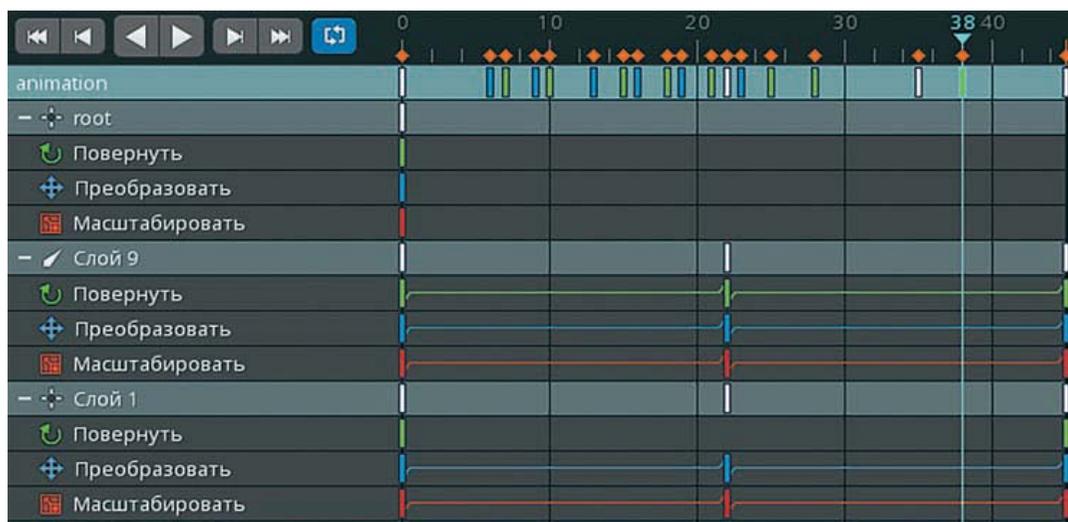
Если не выделена ни одна кость, будут показаны все ключи. Если выделена одна или несколько костей, то в окне шкалы времени будут показаны только ключи для них. Также, если кликнуть на название кости или инструмента в строке шкалы времени, мы выделим кость и оставим в окне только информацию о ней. Снять выделение можно клавишей Esc.

Щелкая на пустом месте шкалы, мы выделяем конкретный кадр анимации. Если кликнуть по уже заданному ключу, мы выделим его в данном кадре. Также можно выделять ключи рамкой. Проигрывание анимации можно запустить и остановить двойным щелчком мыши по верхней части шкалы либо нажатием на кнопку «Play» в левом верхнем углу.



Обратите внимание, что кнопка проигрывания две — в одну и другую сторону. Таким образом, мы можем проигрывать анимацию в прямом и обратном порядке кадров.

Рядом с кнопками проигрывания находятся стандартные для любого видеоплеера иконки. Эти кнопки перемещают анимацию на один ключ вперед (или назад) по верхней строке «animation». А крайние кнопки перемещают анимацию на первый и последний кадр соответственно направлению иконок на кнопках.



Пример заполненной ключами шкалы времени

Самая верхняя строчка содержит все ключи анимации. Если где-то внизу в какой-либо кости этот кадр занят ключом, то он будет отображаться в верхней строке. Она удобна тем, что мы всегда видим все назначенные где-либо ключи, но неудобна плотностью ключей. Если анимация сложная, то эта строчка может быть хаотично заполнена множеством ключей. Ориентироваться по ней в анимации бывает затруднительно.

Между ключами отображается вид графика, который задействован на этом промежутке. Если линия прямая — на интервале равномерное движение. Если линия с закруглениями в начале и конце — кривая Безье. Ступенчатая кривая отображается пунктиром.

Инструменты шкалы времени

Над самой шкалой с кадрами есть панель инструментов. Эти инструменты облегчают работу со шкалой, ключами и помогают с некоторыми задачами.



Начнем с первой группы кнопок. Кнопка «Синхронизировать» совмещает шкалу времени и график движения цельного вида. Его мы рассмотрим позже, так как он довольно сложный, и сначала мы будем работать с упрощенной версией графика движения. Но когда вы решите использовать график всего движения, кнопка «Синхронизировать» пригодится для совместной работы со шкалой и графиком одновременно.

Кнопки «минус» и «плюс» показывают и скрывают подробные ключи для кости. Для каждой кости у вас может быть несколько строчек ключей. Для каждого инструмента и его изменения предусмотрена своя строка. Нажав на «минус», вы скроете эти подробные ключи, и тогда будет отображаться лишь одна строка с общими ключами кости.

При нажатии на иконку воронки открывается меню фильтров, с помощью которого можно настроить отображение определенных элементов по одному критерию или по нескольким.

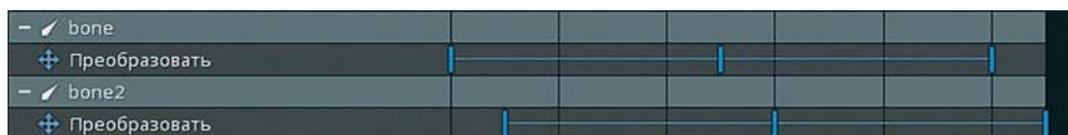
«Замок» предназначен для блокировки переключения костей на шкале времени. Если у вас была выделена определенная кость и вы нажали на «замок», то ее ключи фиксируются на шкале. И выбирая другие кости, вы не будете вызывать отображение их ключей на шкале времени. При нажатии на значок «обновить» закрепляются ключи новой выбранной кости. При помощи иконки кости можно выбрать отображаемую кость в структуре проекта. Иконка ключа предназначена для установки нового ключа в выбранном кадре. Таким образом устанавливаются ключи для всех отображаемых костей на шкале времени и только для тех значений, которые уже изменяются в анимации.

Далее идет панель со стандартными инструментами копирования и вставки и инструментами работы с графиком характера движения. А вот панель с тремя большими

кнопками «Переместить», «Сместить», «Корректировать» рассмотрим подробнее. Используя эти кнопки, мы активируем особые режимы работы с ключами на шкале времени.



Кнопка **«Переместить»** позволяет передвигать выбранный ключ и все ключи за ним в ту сторону, в которую мы его потянем. Так мы можем перемещать по шкале времени всю строку ключей, сохраняя расстояния между ними. Также можно выбрать сразу несколько ключей и двигать несколько строк одновременно. Длина анимации автоматически увеличится при выходе ключей за последний кадр.

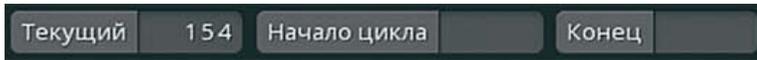


Применение инструмента «Переместить» к строке ключей преобразования кости bone2

Кнопка **«Сместить»** также позволяет двигать ключи, сохраняя расстояния между ними, но при этом сохраняя и длину анимации. По сути, этот инструмент смещает анимацию целиком, и ее начало будет не в нулевом кадре, а в том, куда мы подвинем ключ. Сместить можно только строку, а не отдельные ключи. Разберем действие этого инструмента на этапы. Сначала на определенное количество кадров сдвигаются все ключи. Затем в последнем кадре анимации (в том, который был последним до перемещения ключей) создается ключ. Этот ключ копируется в нулевой кадр для цикличности анимации. А тот ключ, который вышел за границы длины анимации, удаляется. Важный момент — инструмент работает только при условии, что нулевой и финальный кадр анимации идентичны. Если это не так, смещения происходить не будет.

Кнопка **«Корректировать»** позволяет скорректировать положение всех выделенных ключей. С помощью этого инструмента мы можем перемещать всю анимацию или ее сегмент, поворачивать или масштабировать. Это полезно в случае, если у нас сложная анимация, которая должна происходить левее или правее по оси, или если ее нужно повернуть на определенный угол. Мы должны выбрать анимацию целиком или ее область. Далее с включенным инструментом «Корректировать» на рабочей области нужно изменить анимацию. Все ключи скорректируются под новое положение анимации. Лучше всего выбирать начальный, конечный кадр или любой, в котором установлены все ключи. Если же ключей нет, инструмент поставит их сам, а это не всегда удобно. Инструмент работает только с однотипными ключами, то есть только преобразования или только поворота. Разнотипные ключи с помощью этого инструмента не скорректируются.

И последняя панель работы с кадрами.



Поле «Текущий» позволяет вписать нужный кадр вручную и выбрать его. Тогда не придется искать его на шкале. Это полезно, если анимация длинная и поиск кадров бегунком уже не удобен. Если нажать на кнопку «Текущий», то при проигрывании анимации окно шкалы будет двигаться за текущим кадром.

Поля «Начало цикла» и «Конец» позволяют задать проигрывание сегмента анимации. Мы можем или вписать значения вручную в текстовое поле, или выбрать бегунком нужный кадр и нажать кнопку.

Предпросмотр

Для того чтобы вызвать окно предпросмотра, нужно кликнуть на меню «Вид» в правом верхнем углу окна программы. Окно предпросмотра позволяет просматривать все наши анимации. Мы, конечно, можем проигрывать анимацию и в самой рабочей области, нажав на кнопку «Play». Но в окне предпросмотра есть несколько удобных функций для оценки своей анимации и удобного просмотра.



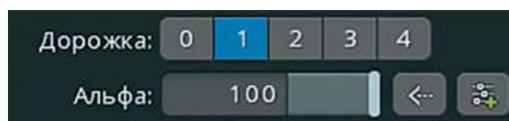
Окно предпросмотра анимаций

Окно состоит из непосредственно окна просмотра анимации и панели справа. На панели отображаются все созданные в проекте анимации. Переключаясь между ними, мы можем в этом окне просматривать каждую из них. Это удобнее, чем переключаться между анимациями в проекте и проигрывать каждую через кнопку «Play». В нижней части панели мы можем менять скорость просмотра анимации и стрелочкой сбрасывать эту скорость к 100%. Это значение соответствует той скорости, что задана в проекте на шкале времени. Значение в процентах меньше 100 будет замедлять анимацию, больше 100 — ускорять. Нужно помнить, что это не фактическое изменение скорости анимации, а лишь скорость предпросмотра. Но это полезно для общего понимания нужной нам скорости, так как данный параметр мы можем настроить уже после импорта анимации в игровой движок. Поэтому у нас нет задачи идеально точно попасть в нужную скорость воспроизведения, ее можно довести до идеала уже в движке. А предпросмотр поможет понять, нужно ли будет в движке ускорение или замедление нашей анимации.

Конечно, не стоит специально отклоняться от требуемой скорости и делать анимацию нарочито медленной, а потом ускорять в движке. В таком случае, скорее всего, анимация потеряет свою привлекательность. Изначально мы создавали ее с упором на определенную скорость и оценивали детали движения именно в этой скорости, поэтому, ускорив нашу анимацию в движке, мы можем получить не тот результат, который видели в медленном варианте. Но слегка довести скорость уже в процессе работы в движке можно.

Пункт «Смешанное» помогает оценить автоматические переходы между анимациями. Если персонаж бежал, а затем решил прыгнуть, то между этими анимациями нужен переход. Можно делать отдельную анимацию на переход, если этого требует проект. Но в основном используются автоматические переходы. Значение в этом поле задает длительность перехода в секундах.

Ниже располагается панель «Дорожка», которая содержит несколько дорожек для анимаций. На каждую можно назначить свою анимацию и проигрывать их все одновременно.



Это удобно для случаев, когда мы анимируем различные части персонажа или нескольких персонажей в одной сцене, а длительность цикла у этих анимаций разная. Например, персонаж бежит, стреляет и моргает. Длительность анимации моргания явно значительно короче остальных, за цикл бега персонаж может моргнуть много раз. Не очень удобно копировать ключи этой анимации в более длительную, чтобы соединить их. Поэтому мы делаем эту анимацию отдельно и просматриваем ее на дополнительной дорожке. Так же и со стрельбой. Персонаж может стрелять реже, чем передвигать ногами при беге. Чтобы посмотреть, как персонаж бежит и стреляет одновременно, нужно использовать дорожки.

Подсказки от разработчиков

В программе Spine 2D при наведении на кнопки всплывает подсказка. Эта подсказка поможет вам вспомнить назначение какой-либо панели или кнопки. Также во всплывающей подсказке может быть представлено сочетание горячих клавиш для данной функции. Со временем при работе в программе вы начнете использовать все больше горячих клавиш. Зачастую так намного быстрее и удобнее, чем вызывать команды щелчком мыши.

Для начинающих пользователей это очень удобная и полезная функция. Также если нажать на иконку конверта в верхнем левом углу окна программы, появится окно с примерами от команды разработчиков Spine, новостями и обновлениями программы. И здесь же можно найти даже некоторые обучающие материалы от команды разработчиков. У них очень полезный, наполненный нужными материалами сайт. Советую ознакомиться с предлагаемыми уроками и примерами от разработчиков. Вы найдете для себя много интересного и полезного.

Также на сайте есть форум, где можно задать любой вопрос и получить на него ответ. Правда, общение происходит на английском языке, так как форум не региональный.

ГЛАВА 4

12 принципов анимации

Прежде чем перейти к первым практическим упражнениям по анимации в Spine 2D, я бы хотела рассказать об основных принципах анимации. 12 принципов анимации — это первое, что вам выдаст поисковик в интернете на запрос об анимации и о том, как ее создавать. Ни один курс и ни одна книга не обходятся без перечисления принципов анимации. Я не буду исключением и расскажу вам о базе создания анимации. Не все 12 принципов одинаково часто используются при работе, особенно в случае скелетной анимации. Я буду делать упор на те пункты, которые наиболее важны именно в Spine и скелетной анимации. Большинство из этих базовых принципов изначально разрабатывались на примере покадровой анимации, и для других видов есть некоторые нюансы их использования. Многие делается в программе несколькими кликами. Также некоторые принципы анимации более применимы для работы именно с мультфильмами, видеороликами, а в игровой анимации уже не так актуальны. Или есть особенности их использования.

1. Сжатие и растяжение

Самый важный и часто используемый закон анимации. Именно с помощью сжатия и растяжения мы показываем гибкость и вес предмета. Растяжение при быстром движении или прыжке, изменение формы при столкновении с поверхностью — подобные эффекты иллюстрируют этот принцип. В примерах с анимацией мячика, описанных далее, мы сразу же будем его использовать. Главное — быть осторожным и не перестараться с растяжением. В работе над игрой *Potata* я допустила ошибку, связанную с этим принципом. В анимации главного персонажа я слишком явно использовала сжатие и растяжение. Причем не на отдельные элементы персонажа, а на всю героиню целиком. Этот прием неплохо смотрелся в медленном и крупном варианте анимации, но при импорте в игру в итоговом размере создается ощущение резинового человечка. Человек на самом деле не растягивается как резина при каждом движении, поэтому, если буквально применить данный принцип анимации, можно испортить персонажа. Нужно всегда учитывать характеристики объекта, который мы анимируем, и степень стилизации его изображения и движения. Чем реалистичнее у нас сеттинг, тем более аккуратно и выборочно нужно применять данный закон анимации. В итоговом варианте анимации персонажа я стала использовать растяжение не персонажа целиком, а только волос и юбки героини. Это элементы, которые обладают наибольшей пластичностью и реально могут искажаться при движении. При таком подходе эффект резинового человечка уже не наблюдается.

2. Подготовка и отказное движение

Этот принцип говорит о том, что практически всегда перед движением и после него есть дополнительное движение. Перед броском мяча персонаж сначала замахнется, а после броска выставит ногу вперед для удержания равновесия. Перед прыжком персонаж обязательно присядет. Даже при движении глаз мы должны задать предварительное изменение формы века, а не сразу двигать зрачок в нужную сторону. Двигаться без соблюдения этого принципа могут только механизмы, но даже в этих случаях можно придумать какие-то предварительные или последующие движения. Например, выпускание пара или мигание индикатора, движение мелких элементов перед крупными.

В игровой анимации этот принцип иногда может проявляться в минимальной степени по той причине, что любая анимация занимает время. И если это анимация главного персонажа, которым мы управляем, такие предварительные и последующие за основным движения могут вызывать задержку после нажатия кнопки. Например, возьмем анимацию прыжка и сделаем явное приседание до и после основного движения. Это будет значить, что после нажатия клавиши персонаж сначала присядет, а потом прыгнет. Между нажатием и прыжком будет промежуток времени, который нужен для подготовки к движению. У игрока не появится ощущения, что он с точностью управляет персонажем, так как нажатие клавиши не сразу вызовет прыжок. Для точного контроля это недопустимо. Также если после прыжка персонаж будет приседать, то он не сможет сразу побежать после приземления или совершить атаку. Игроку очень сложно просчитывать эти задержки, поэтому такой подход создаст неудобное и неотзывчивое управление. В играх, в которых требуется контроль, подготовку и отказное движение делают минимальными или не делают вовсе. А вот для объектов, которыми игрок не управляет, принцип остается актуальным.

3. Смягчение начала и окончания движения

Я изменю классический порядок принципов анимации, который вы можете встретить в других источниках. Поскольку мы в этой книге в основном изучаем скелетную анимацию и использование ее в играх, то в первую очередь я расскажу о принципах, наиболее актуальных для такого использования.

Движение не начинается и не заканчивается мгновенно. Для набора скорости, так же как и для остановки, нужно время. У машины всегда есть время разгона и торможения. И так в целом для всех движений. В анимации мы будем очень часто использовать замедление в начале и конце вместо равномерного движения. В покадровой анимации для создания эффекта замедления художник всегда должен рисовать больше кадров именно возле начальной и конечной позиции. В Spine этот принцип реализуется с помощью графика характера движения, который мы меняем с прямой линии на кривую. Равномерное движение очень редко встречается в природе, поэтому привыкайте постоянно менять график. Также можно начало и конец движения

дополнять побочными элементами, эффектами и движениями. Например, эффектом пыли из-под ног при разгоне или с помощью рук при прыжке.

Опять же, если персонаж в игре контролируется игроком, то этот принцип используется минимально, так как создает задержку между основным движением и нажатием кнопки. Но, в отличие от подготовительных движений, мы не можем полностью убрать замедление в начале и конце, иначе движение будет неестественным. Поэтому персонаж просто довольно быстро разгоняется и замедляется, если его нужно контролировать управлением.

4. Движение по дугам

Практически любое движение в природе, за исключением механизмов, осуществляется по криволинейной (дуговой) траектории. В покадровой анимации художники всегда рисовали эти дуги движений в процессе отрисовки кадров анимации. Движение по дугам распространено так же, как и скелетная структура объектов. Сустав в такой структуре является центром окружности, по которой будет происходить дуговое движение. Движение рук, ног, головы — все это привязано к определенным суставам. Анимация человекоподобных персонажей состоит из множества дуговых траекторий движения с различными точками радиуса. Растительность также строится по скелетной структуре. Каждый листик крепится кончиком стебля к стеблю покрупнее. А значит, движение листьев, ветвей, травинок — все будет происходить по дуговым траекториям.

В Spine дуги можно задавать либо движением по двум осям, либо специальными инструментами для создания пути. В Spine мы можем создавать любые траектории для движения с помощью кривых Безье. Помимо дуг в создании движений часто используется траектория восьмерки, или символа бесконечности, как вам больше нравится. Вместо движения вперед-назад по прямой линии движение по траектории восьмерки всегда будет более естественным и приятным глазу. Полет бабочки, движение глаз, различные эффекты — все это намного лучше смотрится при движении по траектории восьмерки. Более упрощенным вариантом траектории является овал. Если восьмерка дает слишком много изгибов, то овальная траектория для движения вперед-назад является, по сути, комбинацией двух дуг. Она дает визуально более прямолинейное движение, но с учетом принципа анимации движения по дугам.

5. Захлест действия, сквозное движение

Этот принцип говорит о том, что не все элементы персонажа или объекта двигаются одновременно. Есть часть объекта, которая задает движение, а есть подчиняющиеся элементы, которые следуют за основой. Эти подчиняющиеся элементы слегка запаздывают за основным движением, так как их приводит в движение основа,

а сами они двигаться не могут. Такими элементами будут части одежды (пояс, длинная юбка), различные украшения (подвески), волосы персонажа. А иногда двигаться с запаздыванием могут и конечности, особенно в утрированной стилистике. Наверняка в мультфильмах вы видели прием, когда ноги персонажа бегут впереди туловища. Допустим, персонаж бежит вперед. Основой движения в данном примере будут ноги. В большинстве случаев ноги будут соединены с туловищем и головой, и только в очень утрированной мультяшной стилистике мы можем эти части персонажа разделять запаздыванием движения. Все-таки тело персонажа — это довольно цельный объект, и запаздывания между ногами, туловищем и головой мы практически не наблюдаем. Но всегда учитывайте сеттинг и степень утрированности проекта, в котором будет ваша анимация. Иногда можно даже нарушать определенные правила в угоду зрелищности и сопоставимости с проектом. Если у персонажа волосы заплетены в косу, то коса сама по себе не может двигаться. Она может двигаться только за головой, в точке крепления. А значит, движение персонажа будет тянуть косу за эту точку. То же самое с хвостом, плащом, юбкой и шлейфом от платья. Эти элементы не могут двигаться сами. Мы тянем их за точки крепления. А значит, они будут двигаться с запаздыванием.

В Spine есть специальные инструменты смещения ключей анимации. Сначала лучше задать движение всем элементам одновременно, так будет быстрее и удобнее, чем сразу задавать запаздывание. А уже после этого нужно сместить анимацию некоторых частей персонажа на определенное количество кадров на шкале времени. Причем иногда у нас могут образовываться целые цепочки элементов, запаздывающих друг за другом. Туловище персонажа, рука персонажа, рукав одежды. Тогда мы должны поочередно сместить каждую строчку, которая относится к элементу. Если в покадровой анимации такие вещи нужно контролировать в рисовании кадров, что довольно сложно, в Spine мы можем легко настраивать степень запаздывания в любое время работы с анимацией.

6. Утрирование

Я уже много раз говорила об утрировании анимации и о его связанности с проектом и стилем. Этот пункт является отдельным принципом анимации — вот насколько важно помнить об утрировании движения. В реальной жизни мы можем не обращать внимания на некоторые детали движения и выраженность принципов анимации. Но при создании анимации нам нужно донести до зрителя характер движения, а для этого нужна утрированность. Да, степень утрированности будет тем больше, чем мультяшнее стиль. Но даже в реалистичном сеттинге нам нужно максимально ярко показывать детали движения.

В играх, кино мы видим картинку в самом выгодном ракурсе, не так, как в жизни. Кадр выставлен более художественно. Объекты на локациях всегда расставлены как для фотосъемки — красиво и в соответствии с художественными правилами. Дизайн

персонажей и окружения более детален и колоритен. В любой сфере мы ограничены определенными факторами, временем повествования, спецификой. А поэтому всегда должны выжимать максимум из каждого кадра. То же самое и с анимацией. Каждое движение должно быть максимально говорящим. Движения могут раскрывать персонажа и его характер. Рассказывать нам о мире игры. Именно поэтому мы и должны утрировать максимально, насколько это возможно в рамках данного проекта и сеттинга.

Утрировать мы можем не только саму анимацию, но и характер персонажа и его эмоции. Если персонаж злится, то он может злиться напоказ. Тогда мы будем утрировать анимацию в сторону раскрытия. Персонаж будет кричать, махать руками, его глаза будут максимально выпучены. Персонаж покраснеет и будет брызгать слюной при разговоре. Или же герой может злиться скрыто, внутри себя. В таком случае мы будем утрировать анимацию в другую сторону. Персонаж будет надуваться, сжимать кулаки, на виске может биться жилка, глаза вообще могут быть закрыты. А брови и рот будут максимально сжаты. И в том и в другом случае мы раскрываем персонажа через эмоцию, максимально утрируя ее в анимации.

7. Второстепенные движения и элементы

Чтобы подчеркнуть основное движение, можно использовать дополнительные. В самом дизайне персонажа уже на начальном этапе можно задать части для того, чтобы сильнее подчеркнуть движения. Зачастую это те самые элементы, с помощью которых создаются запаздывающие движения. Косы, хвосты, ленты. Если персонаж бежит, а за ним развеивается плащ или коса, то такое движение будет более заметным и интересным. Также движение часто подчеркивается визуальными эффектами. Если персонаж увидел что-то важное, он не просто щурит глаза. В глаза можно добавить блеск, который акцентирует внимание зрителя именно на них в этот момент. Если персонаж подобрал какой-то ценный объект, то его можно подсветить сиянием. От прыжка и приземления часто разлетаются клубы пыли и дыма, что довольно утрировано, ведь в жизни мы не видим таких эффектов. Если персонаж стреляет или атакует мечом, то в момент атаки анимируется и лицо персонажа. Это добавляет движению эмоциональности.

Также этот принцип можно интерпретировать с точки зрения логики анимации, а не ее техники. К примеру, если персонаж со злостью стучит в дверь, он в качестве второстепенного движения может еще и топнуть ногой, что добавит эмоциональности его анимации. Мы часто делаем несколько движений одновременно, сами того не замечая, и основное движение будто усиливается побочными. Если мы хотим анимировать задумчивость, то направим взгляд персонажа вверх и приложим руку к лицу. Но дополнительно можно добавить движение постукивания ступней или нервного переминания с ноги на ногу. Это также окрасит анимацию в более яркие эмоциональные оттенки.

8. Расчет времени

Как и в покадровой анимации, в скелетной, созданной в Spine, время анимации определяется количеством кадров. Чем меньше кадров потрачено на движение, тем более резким и быстрым оно будет. Важно сразу определять скорость анимации хотя бы примерно, так как детализация движения и подход к его анимации определяются скоростью. Чем медленнее движение, тем больше мелких деталей мы можем в нем показать. Работая в программе Spine 2D, мы располагаем большей свободой в регулировании скорости анимации. В случае покадровой анимации, если мы нарисовали определенное количество кадров, изменить скорость будет уже довольно затруднительно, так как эти кадры актуальны для заданной скорости движения. В случае со скелетной анимацией мы можем регулировать скорость, меняя количество кадров на шкале времени в любой момент. Также мы можем регулировать скорость анимации в отдельном окне просмотра анимации. Так мы не затрагиваем саму анимацию, а лишь оцениваем различную скорость и то, как выглядит анимация с данной скоростью. Таким образом мы можем проверить разные варианты и выбрать лучший. Также мы можем регулировать скорость анимации уже в движке.

Несмотря на свободу регулирования скорости, нам все равно важно определять ее заранее, просто не настолько точно, как в случае с покадровой анимацией. Особенно это важно, когда в одной анимации есть несколько движений с разной скоростью. В таком варианте нам нужен контраст между движениями, чтобы одно выглядело медленным на фоне другого. А значит, нам важно задать разницу скоростей.

9. Четкие позы

В анимации важно создавать четкие, понятные позы для передачи движения. Позы должны быть максимально выраженными и иногда даже утрированными (возвращаясь к принципу анимации «утрирование»). Причем именно позами во время движения мы можем задавать различные варианты анимации. Например, персонаж может подгибать одну ногу в колене во время прыжка. Это создаст некоторую непосредственность в движении. Но если при этом еще и вытянуть руку в направлении движения, то получится уже супергеройская поза, которая добавит персонажу целеустремленности. Любое движение можно разделить на несколько основных говорящих поз. И каждую из них нужно продумать и сделать выразительной. Также всегда обращайтесь внимание на силуэт. Силуэт — это первое, что читается при взгляде на анимацию и персонажа. Поэтому лучше всего делать движение так, чтобы части персонажа выступали на его силуэт, так они будут более читаемы и заметны. А значит, и наше движение будет выглядеть более понятно.

Помимо понятности, важна еще и реалистичность позы в плане возможности ее осуществления с точки зрения анатомии. Само собой, есть утрирование и стилизация, которые могут развязать нам руки и позволить этой самой анатомией местами слегка пренебречь. Но восприятие персонажа искажаться не должно. Возможности

суставов, изгибов элементов — это нужно учитывать при создании анимации. В движении довольно легко поломать персонажа или противоестественно вывернуть суставы. Нужно всегда помнить об анатомическом строении героя. Хотя, безусловно, порой искажения анатомии даже играют на руку зрелищности анимации. Например, когда мы делаем анимацию рук при создании походки, можно выгибать кисти в обе стороны, чтобы придать герою пластичности. Или даже локоть. В случае с локтем это будет нарушением анатомии, так как локоть выгнуть в обратную сторону невозможно. Но если стилизация позволяет, для большей пластичности можно добавить движение, противоположное обычному.

10. От позы к позе или прямо вперед

Движение можно создавать, используя два подхода. Первый — от позы к позе. Сначала мы задаем основные кадры движения, а потом уже промежуточные. Именно такой подход в большинстве случаев используется в работе со скелетной анимацией в Spine. Ведь промежуточные кадры может создать сама программа, а мы при этом можем только направить ее и задать для промежутков определенные условия. А вот ключевые позы лучше как раз создавать вручную, делая положение персонажа или объекта максимально выразительным. Такой подход используется практически всегда, за исключением эффектов и некоторых специфичных анимаций. Причем анимацию мы можем строить не по логике совершения движения, а по логике ключевых поз. В этом случае сначала создаются начальная и конечная точки, а затем пара промежуточных. После этого добавляются позы для подготовки и последствия движения.

Анимацию эффектов лучше выстраивать по второму принципу «прямо вперед», или от кадра к кадру. Эффекты не содержат каких-то ключевых поз, нам важно контролировать изменение формы от одного кадра к другому. Мы, конечно, все равно не будем контролировать буквально каждый кадр, так как все-таки нужно полагаться на программу в построении промежутков, раз есть такая возможность. Это сильно экономит время. Но создавать кадры через каждые 5–10 кадров, двигаясь от начала до конца по логике совершения движения, — вполне применимое решение.

11. Сценичность

В анимации важно следовать законам композиции, построения кадра и создавать максимально сценичные образы. Законы композиции применимы не только для сцены, но и для отдельных объектов и персонажей. Мы должны управлять вниманием зрителя, выделяя основное в концепте персонажа и подчеркивая это анимацией. У героя может быть какая-то узнаваемая деталь, и в анимации нужно будет использовать ее по максимуму. При этом нельзя перенаправлять внимание на неважный элемент. Если в концепте у персонажа очень объемная шевелюра, то именно ей в анимации стоит уделить большее внимание, а не ленточке на ноге, например. Саму анимацию также нужно делить на основные и побочные движения и направлять внимание зрителя на основные.

12. Привлекательность

Этот параметр невозможно измерить или описать четкими характеристиками. Но в целом нужно помнить, что задача аниматора — преподнести зрителю персонажа или целую сцену. Зрителю должно нравиться то, что он видит. А задача аниматора — добиться этого впечатления. Все средства должны быть направлены на то, чтобы создать максимально привлекательный образ. Чтобы произвести впечатление на зрителя, привлекательность в первую очередь закладывается на уровне концепта персонажа, а дальше дело уже за созданием движений. И порой именно движения делают персонажа привлекательным. Если посмотреть на персонажей для анимации в статике, далеко не все из них выделяются дизайном и сразу запоминаются. А вот образ в целом, мимика, характер движений — все это уже намного сильнее влияет на привлекательность.

ГЛАВА 5

Топ-5 упражнений для начинающего аниматора

Прежде чем перейти к подробным урокам и фишкам программы, мы должны усвоить азы. Путаница в интерфейсе не должна мешать заниматься анимацией. Изучать базу лучше на простых примерах, которые помогут освоить и принципы анимации, и интерфейс программы. Нельзя сразу погружаться с головой в сложные комплексные задачи. Это как если вы только начали рисовать, но уже пытаетесь сделать иллюстрации в стиле Blizzard. Сначала нужно освоить отдельные объекты и базовые движения.

Я представлю вам пять упражнений для аниматора, которые, на мой взгляд, дадут отличный старт в увлечении анимацией. Мы будем делать пять разных движений для мяча. «Да, всего лишь мяч, что тут сложного?» — скажете вы. Но даже в таких, казалось бы, простых движениях нужно соблюдать принципы анимации и использовать практически все базовые функции Spine 2D. А значит, эти упражнения отлично подойдут для первой практики! Давайте начнем!

Движение по прямой

Начнем мы с **движения по прямой**. В этом упражнении, несмотря на кажущуюся простоту, есть пара нюансов. Мы не просто переносим объект из точки А в точку Б. Это лишь основа нашего движения. Если мы не применим принципы анимации, оно получится безликим и роботизированным. Да, если у нас на проекте поставлена задача ни в коем случае не утрировать анимации и не добавлять даже минимальной мультяшной яркости движению, мы просто подвинем круг. Но такое бывает очень редко. Даже если мы анимируем робота, мы должны соблюдать принципы анимации и слегка утрировать движение для большей выразительности.

Первый принцип анимации, который мы должны учесть при этом движении, — замедление в начале и конце. Объект не может мгновенно набрать скорость и также мгновенно затормозить. Есть определенное количество времени на разгон и торможение, которое мы можем регулировать и таким образом задавать резкость движению.

Второй момент — это сам эффект торможения: bounce effect. Это своего рода передача инерции в анимации. Вы наверняка замечали, что, когда вы пытаетесь резко затормозить после быстрого бега, ваше тело подается вперед, а затем возвращается в прямое положение. Ноги уже затормозили, а верхняя часть тела продолжает движение.

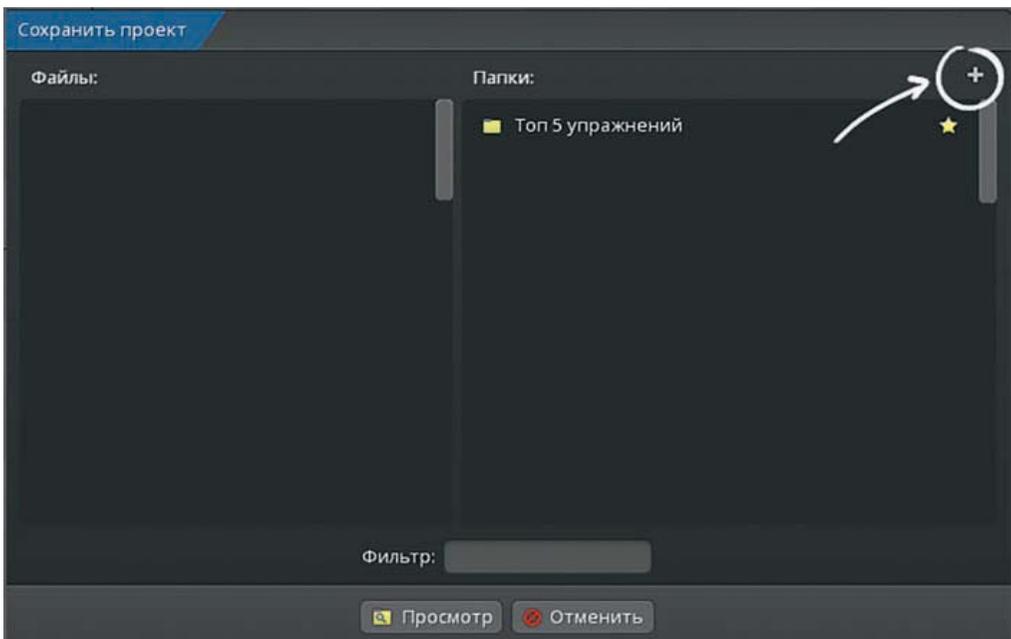
В анимации этот эффект часто утрируют и даже цельным объектам добавляют эффект «пружины» при торможении. Кстати, в мультфильмах можно наблюдать подобное утрирование и на старте движения, когда ноги персонажа стартуют раньше тела.

Третий нюанс — размытие в движении. Чтобы показать очень быстрое движение или утрировать анимацию, используется один кадр вместо нескольких. Когда движение происходит очень быстро, то наш глаз не фиксирует отдельные кадры. Попробуйте быстро махать рукой перед глазами. Вы увидите размытую картинку. Этот эффект и применяется в анимации.

А теперь поэтапно реализуем движение по прямой в Spine 2D, попутно изучая интерфейс программы.

Создадим проект. По умолчанию при открытии Spine у вас уже открывается пустой проект. Но я советую сразу же сохранить его в нужном вам месте, чтобы он не потерялся и вы точно знали, где он размещен на диске.

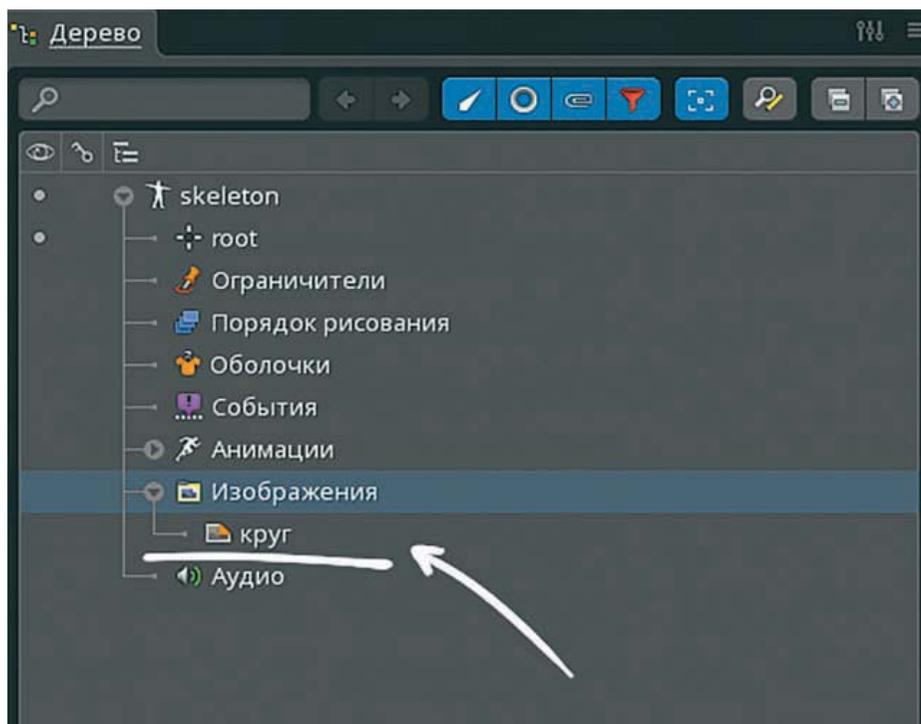
1. Нажимаем сочетание клавиш Ctrl + S или в выпадающем меню выбираем пункт «Сохранить проект как». У нас высвечивается окно с последними использованными файлами и папками, в котором пусто, если мы первый раз открыли программу. Нажимаем плюсики в правом верхнем углу, и у нас появляется окно выбора расположения проекта.



Выбираем нужную папку, и она появляется в окне папок. Затем выбираем ее в списке, вбиваем название проекта и нажимаем на кнопку «Сохранить».

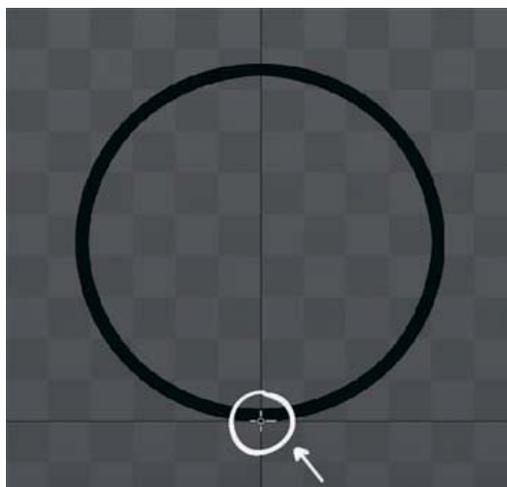
2. Начнем с того, что вставим нашу картинку в программу. Картинка у нас одна, поэтому сделаем это вручную. Для более сложных проектов мы будем использовать скрипт, который автоматически импортирует наш psd-файл в Spine, сохраняя положение элементов.

Нам нужно положить png-файл в ту же самую папку, куда мы сохранили проект. Тогда в меню справа в «Изображениях» автоматически появится наша картинка.

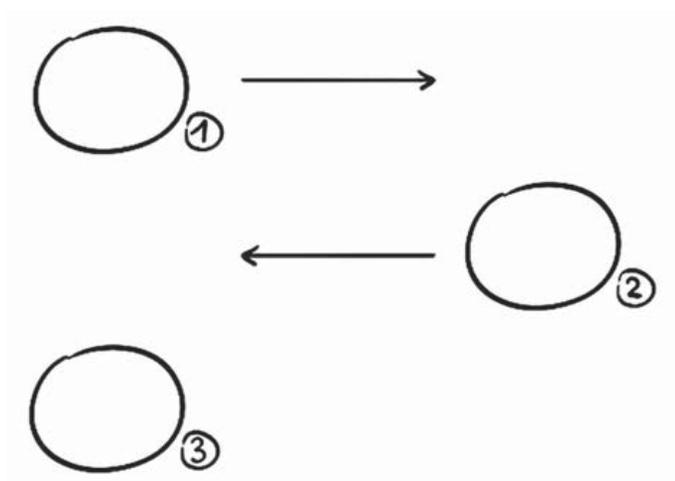


После этого просто переносим картинку из этого меню в главное окно программы. Перенести нужно так, чтобы низ кружочка совпал с началом координат.

3. В случае с одной картинкой кость привяжется к ней автоматически. Та корневая кость, *root*, которая создается в любом проекте, будет костью нашего мячика. С ее помощью мы сможем его анимировать. По сути, мы пока пропускаем этап рига (он выполнен автоматически) и сразу переходим к анимации.



4. Для того чтобы переключиться в окно анимации, нажимаем на надпись «Настройка» в левом верхнем углу. Она меняется на надпись «Анимировать», и появляется шкала времени.
5. Нам нужно задать основу движения — перемещение по оси X влево-вправо.

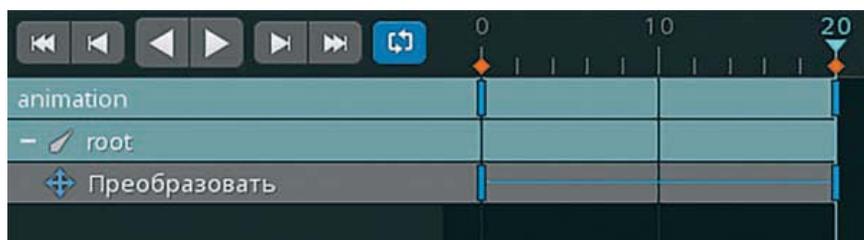


Зафиксируем начальное положение объекта. Это делается нажатием иконки ключа, расположенной напротив инструмента «Преобразовать».

Повернуть		0.0	
Преобразовать	0.0	0.0	
Масштабировать	1.0	1.0	
Сдвинуть	0.0	0.0	

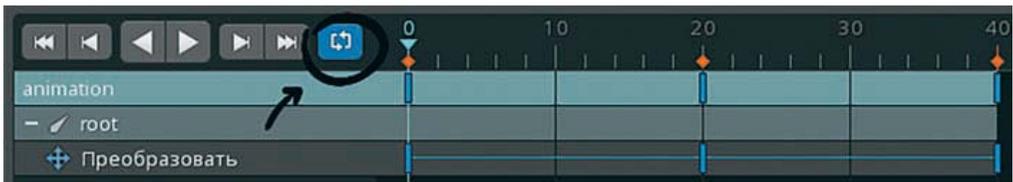
Затем зафиксируем конечное положение. На шкале времени нужно переместить указатель в 20-й кадр. Для этого щелкаем левой кнопкой мыши по шкале времени в этом месте.

Затем с помощью инструмента «Преобразовать» передвигаем нашу кость (а значит, и картинку) по оси X вправо. Нажимаем на красную стрелку оси левой кнопкой мыши и тянем вправо на требуемое расстояние.



Теперь нам нужно скопировать начальный кадр в 40-й. Нажимаем на ключ 1-го кадра. Он должен выделиться голубым цветом. Нажимаем сочетание клавиш Ctrl + C. Нажимаем на 40-й кадр и клавиши Ctrl + V.

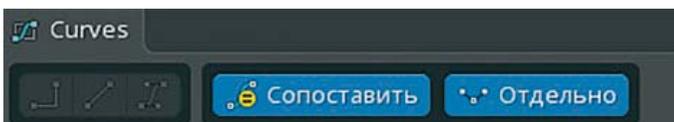
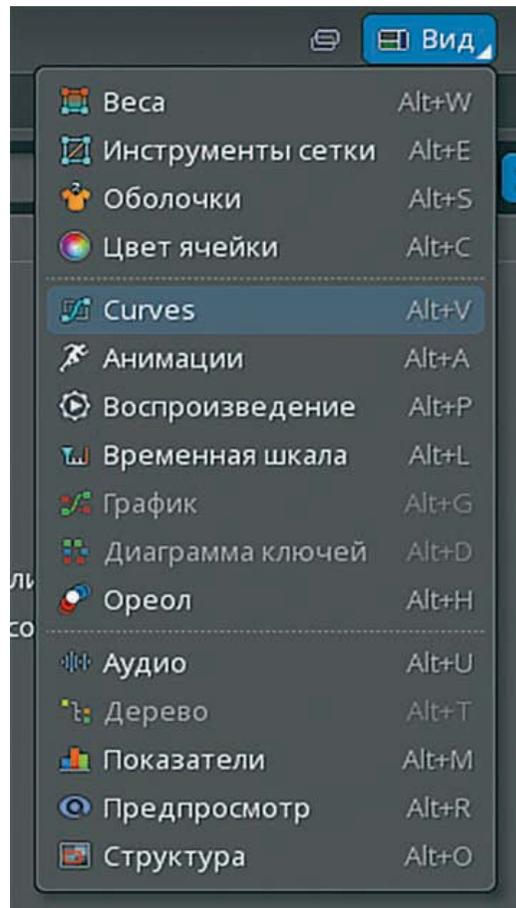
Теперь можно нажать на «Play» и увидеть, как двигается наш круг. Чтобы анимация проигрывалась зациклено, нажимаем иконку круговых стрелок возле шкалы кадров.



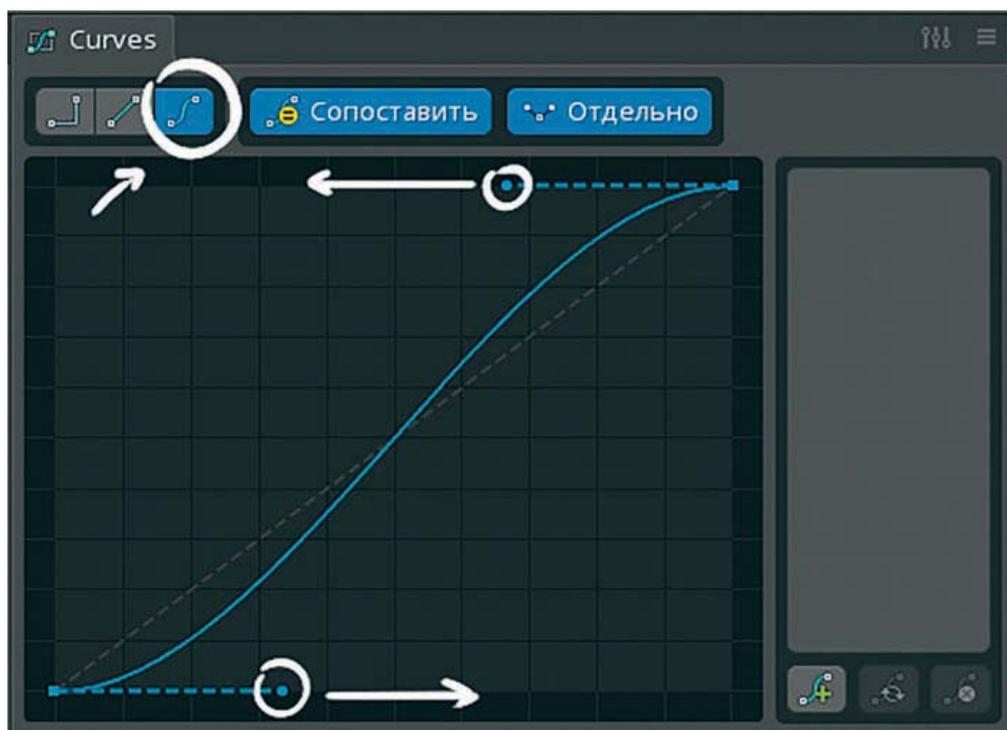
6. Обратите внимание на неестественность движения. Давайте исправим это и добавим замедления в начале и конце перемещения. В Spine это делается с помощью графика характера движения. Это автоматический инструмент, который позволяет нам с помощью кривой контролировать скорость во время движения от одного ключевого кадра к другому.

Для начала нам нужно вызвать окно графика, так как оно может не отображаться у нас в рабочем пространстве по умолчанию. Нажимаем на кнопку «Вид» и в выпавшем меню выбираем «Curves».

Важно в появившемся окне активировать кнопку «Отдельно», чтобы иметь возможность независимо контролировать части общего движения.



Далее выбираем два ключа в 0-м и 20-м кадре. Можно делать это с зажатой клавишей мыши, выделяя их рамкой на шкале времени, или кликать на каждый ключ мышкой с зажатой клавишей Ctrl. Вы увидите график движения в виде прямой линии, что означает равномерность нашего движения. Нужно нажать на иконку «кривая Безье» (третья иконка над графиком), чтобы график сменился на кривую с замедлением в начале и конце. Интенсивность этих искривлений, а значит, и замедления, мы можем контролировать, меняя кривую с помощью опорных точек.

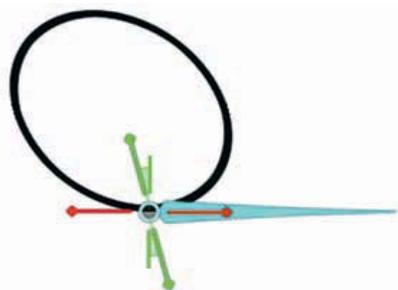


7. Добавим эффект «вперед бегущих ног». Для этого будем использовать инструмент «Сдвинуть».

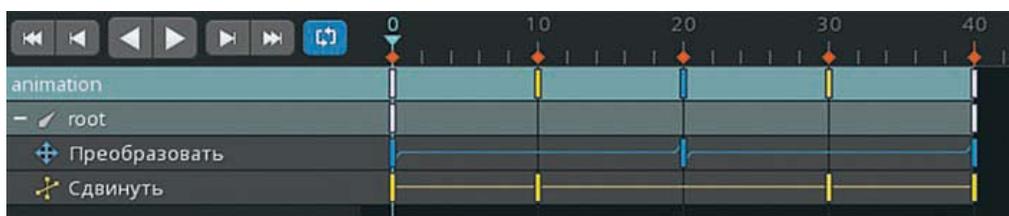
 Повернуть	0.0	
 Преобразовать	0.0	
 Масштабировать	1.0	
 Сдвинуть	0.0	

Нам нужно в центральный кадр добавить искажение круга. Зафиксируем положение ключа сдвига в начальном кадре. Для этого нажимаем на иконку ключа около инструмента при нажатом 0-м кадре. Нажимаем на 10-й кадр, выбираем инструмент «Сдвинуть». И, двигая верхнюю ось на кости, наклоняем наш круг в противоположную движению сторону.

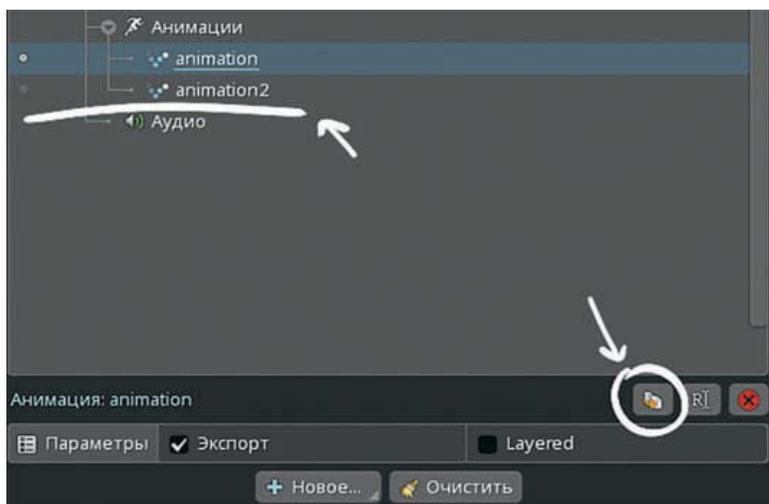
То есть при движении вправо наклон будет влево. То же самое проделываем с 30-м кадром, только в противоположную сторону. Значения наклона должны совпадать, но иметь противоположный знак. Например, в 10-м кадре у нас 15, а в 30-м должно быть минус 15. Это значение мы можем менять в текстовых полях рядом с инструментом «Сдвинуть».



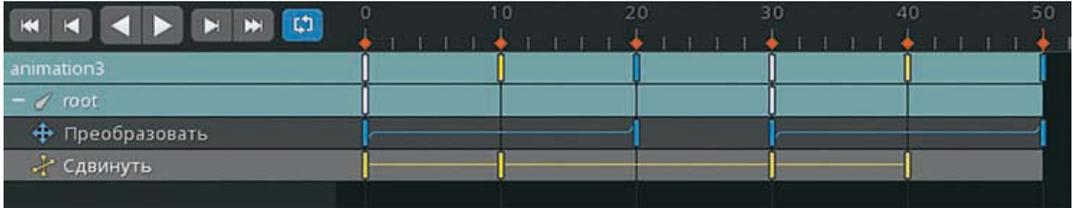
И не забываем скопировать 0-й ключ в 40-й.



- В качестве финального штриха добавим bounce-эффект в начальной и конечной точке движения. В нашем случае этот эффект будет представлять собой возвращение в исходное положение из искажения, которое мы добавили на предыдущем этапе. Итак, для начала нам нужно несколько изменить положение ключей. Чтобы сохранить существующую анимацию, а изменения делать в новой, продублируем ее. Выделяем анимацию в правом меню и нажимаем Ctrl + D или иконку «Дублировать» в правом нижнем углу этого окна. Теперь выделим дублированную анимацию и будем делать изменения в ней.



Нам нужно создать промежутки между движениями туда и обратно. Для bounce-эффекта нужно некоторое время, а значит, нужны и дополнительные кадры. Выделяем синие ключи (от инструмента преобразования) 20-го и 40-го кадра. Чтобы это сделать, нужно щелкать только по синим ключам. С помощью щелчка по верхнему ключу выбираются все ключи, а по остальным — отдельные. Нажимаем Ctrl + C. Затем удаляем ключ преобразования в 40-м кадре, так как финальное положение сместится. Выделяем 30-й кадр и нажимаем Ctrl + V.

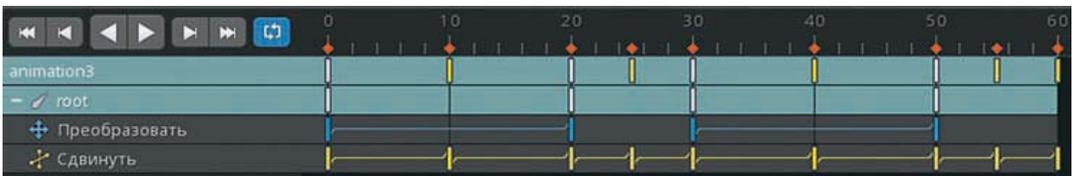


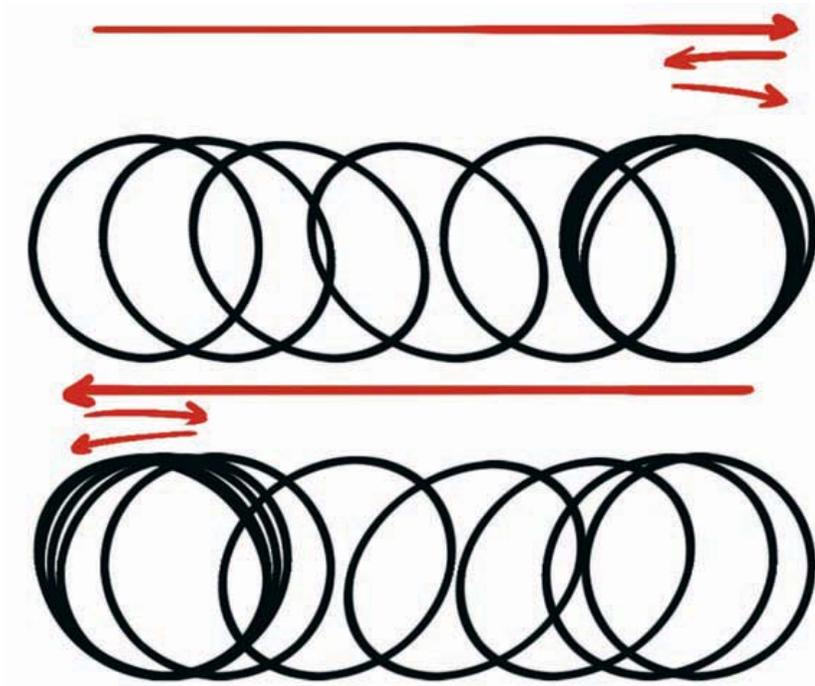
Теперь будем работать с кадрами 20, 25, 30. Нам нужно зафиксировать ключи сдвига в этих кадрах. Лучше это делать значениями в текстовом поле, чтобы все было точно, а не «на глазок». В 20-м кадре задаем значение минус 10. Тем самым в этом кадре кружок исказится в противоположную 10-му кадру сторону. Наше «туловище» продолжит движение, когда «ноги» уже встали. Далее в 25-м кадре устанавливаем значение 5. Это значение меньше, а значит, кружок постепенно приходит к изначальному ровному положению. Из 30-го кадра значение ключа сдвига переносим в 40-й, просто потянув за ключ и переместив его туда. И в 30-м кадре устанавливаем значение 0, возвращаем кружок в ровное положение.

Теперь делаем то же самое, но с противоположными знаками для кадров 50, 55, 60.

50	10
55	5
60	0

Настроим график движения для сдвига. У всех ключей ставим замедление в начале и конце.

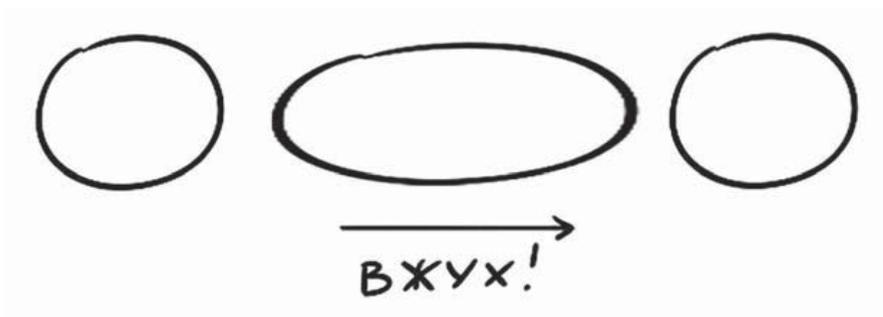




Покадровая схема анимации движения по прямой. Стрелками показано направление движения. В области bounce-эффекта видим уплотнение кадров

БЫСТРОЕ ДВИЖЕНИЕ ПО ПРЯМОЙ

Сделаем отдельную вариацию движения для большой скорости. Именно в этом примере мы добавим центральный растянутый кадр, чтобы показать быстрое изменение положения объекта.



Возьмем нашу анимацию до применения bounce-эффекта (вы ведь следовали моим указаниям и продублировали анимацию, не так ли?). При быстром движении у нас просто нет времени на этот эффект. Хотя это зависит от анимации, бывают разные вариации. Итак, нам нужно сжать анимацию в два раза, то есть длительность новой анимации должна быть двадцать кадров вместо сорока. Затем нужно сохранить

положение всех ключей относительно новой длительности. Сжав длительность в кадрах, мы сделаем наше движение быстрее.

Для этого необходимо рамкой выделить верхние ключи, таким образом мы выделим все ключи нашей анимации. Наводим на правый край рамку выделения. Курсор сменяется на две стрелки в разные стороны. Нажимаем и тянем влево, пока 40-й кадр не встанет в 20-й.

Теперь нужно добавить масштаб. Фиксируем исходный масштаб в 0-м кадре нажав на иконку ключика рядом с инструментом масштаба при выделенном 0-м кадре.

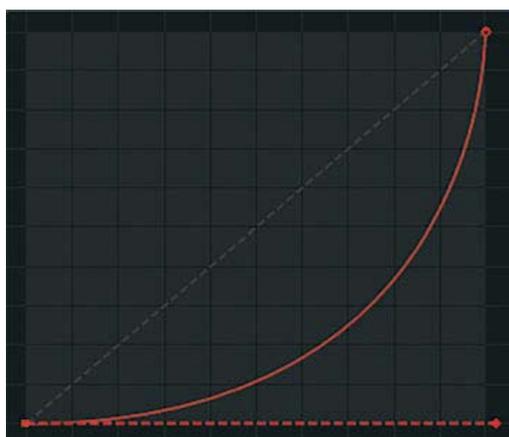


В 5-м и 15-м кадрах устанавливаем масштаб 2,5, а в 10-м и 20-м кадрах по единице.



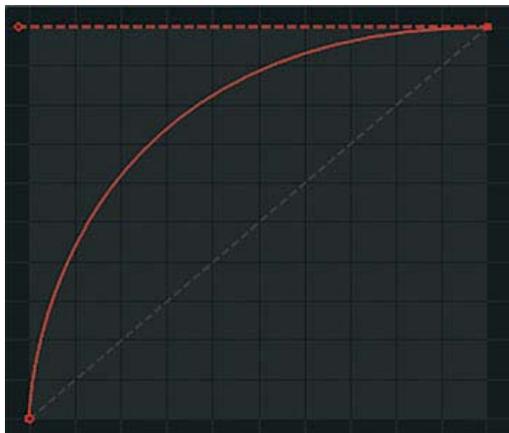
Для большей резкости движения нам нужно поменять график, усилив замедление в начале и конце и рывок в основе движения. Назовем этот эффект «вжух». Выделяем ключи преобразования (синие). Потянув за опорные точки, делаем график более изогнутым.

Также можно изменить график отдельно для масштабирования. Нам нужно добиться того, чтобы масштабирование начиналось не сразу после начала движения. Это нужно для того, чтобы эффект размытия был максимальным только по центру, а не плавным по всей длине движения. В начальном и 10-м кадре



убираем замедление в конце и делаем сильное замедление в начале.

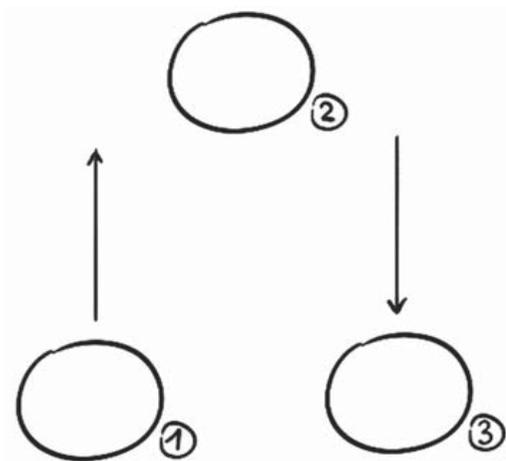
А в 5-м и 15-м кадрах, наоборот, делаем сильное замедление в конце.



Прыжок мяча

В анимации прыжка, помимо соблюдения принципов из предыдущего примера, есть важный нюанс. Особое внимание нужно уделить моменту подготовки к движению и моменту окончания движения. Если бы у нас был персонаж, то перед прыжком он бы приседал и только затем прыгал. А после приземления опять же приседал, смягчая свое падение. В нашем случае, поскольку мячик будет прыгать зациклено и постоянно, то эти две фазы движения сольются в одну — фазу взаимодействия с поверхностью.

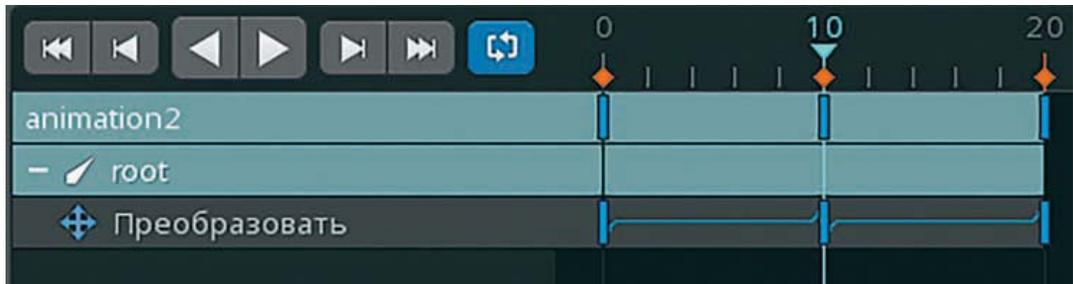
Первое, что мы сделаем, — по аналогии с предыдущим упражнением **зададим основу движения**. Фиксируем нулевой ключ, двигаем наш мяч по оси Y вверх и фиксируем второй ключевой кадр, затем копируем нулевой кадр в финальный.



Длина прыжка может быть разной, тут можно поэкспериментировать и сделать пару-тройку вариаций, которые будут отличаться степенью подпрыгивания вверх. Проще начать с более высокого прыжка, так как тогда будет где развернуться в применении принципов анимации.

Я поставлю средние значения: длительность анимации — 20 кадров, в верхней точке мячик в 10-м кадре на высоте 1500 пунктов.





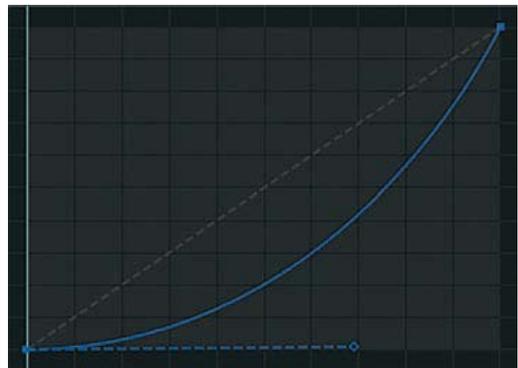
Настроим **характер движения** с помощью графика для движения вверх-вниз.

Нам нужно передать эффект «зависания в воздухе» нашего мячика. Скорость к верхней точке уменьшается до 0, а затем постепенно нарастает с движением вниз. Значит, от 0-го до 10-го кадра нам необходимо добавить замедление в конце, причем утрировав его на графике. Замедления в начале движения не будет.

А от 10-го до 20-го кадра нам нужно добавить замедление в начале без замедления в конце.

Добавим **растяжение во время движения**. Не забываем о законе сохранения объема. Этот закон подразумевает, что, на сколько мы растянули объект по одной оси, на столько мы должны его сжать по другой, чтобы общий объем сохранился. Вычисляем значения по формуле: $1 / X = Y$. Где X — значение по одной оси, Y — значение по второй. Например, мы сжали объект до значения 0,6 по оси X . Получаем: $1 / 0,6 = 1,66$. Это значение мы должны поставить по оси Y .

Десятый кадр мы возьмем за единицу масштаба. Это момент, когда мячик находится в верхней точке в основном своем объеме и масштабе. К 20-му кадру мячик максимально растягивается перед соприкосновением с землей. Степень растяжения зависит от мягкости

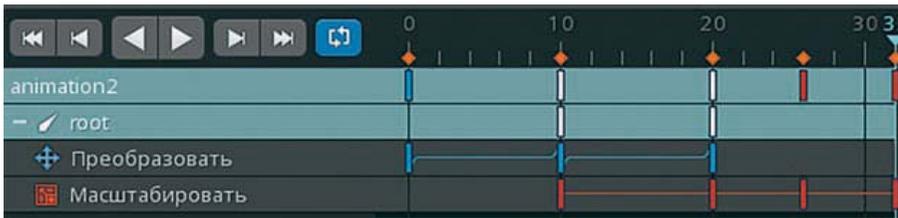


Повернуть	0.0	0.0	
Преобразовать	0.0	0.0	
Масштабировать	0.8	1.25	
Сдвинуть	0.0	0.0	

и пластичности объекта, скорости его движения и мультяшности нашей анимации. Если анимация утрированная и мультяшная, то растяжения добавляются практически всегда, даже при довольно медленном движении и анимации твердых элементов.

Я поставлю значения 0,8 и 1,25 соответственно по осям X и Y.

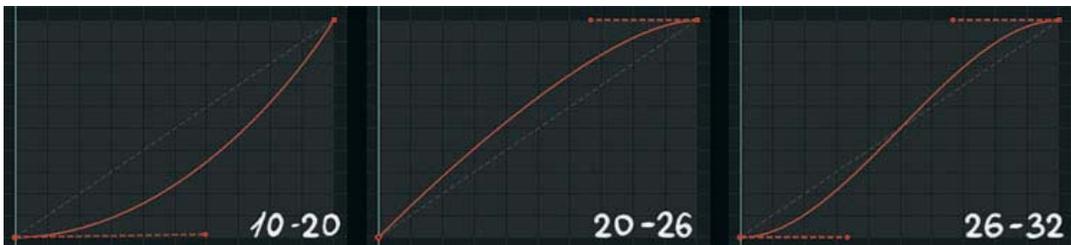
Далее наш мяч на полной скорости **сталкивается с землей и сжимается**, так как часть объекта продолжает движение. Торможение начинается в точке соприкосновения и постепенно распространяется на весь объект. Возьмем шесть кадров на сжатие и столько же на растяжение. От количества кадров будет зависеть время нахождения мячика у земли. Чем меньше кадров, тем быстрее мячик отскочит обратно вверх. В 26-м кадре ставим значения 1,25 и 0,8. В 32-м кадре меняем эти значения местами.



Вот так выглядят ключи на данном этапе

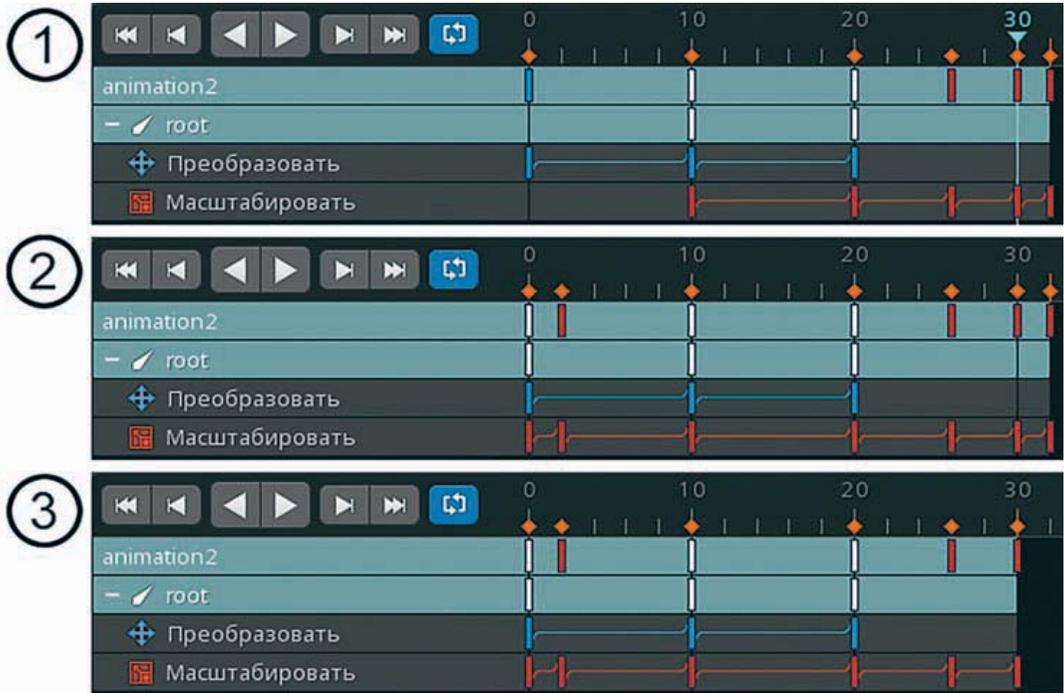
Закон сохранения объема можно не применять только в том случае, когда у нас очень быстрое движение и мы задаем не пластичность элемента, а размытие движения. Размытый кадр вообще может быть не буквальным масштабированием элемента, ведь движение не всегда происходит по прямой, а объект не всегда такой же простой. Размытый кадр — объединение нескольких кадров движения в один. Растяжение по закону сохранения объема не объединяет кадры, а добавляет пластичности элементу во время движения.

Использовать этот прием стоит, только если у нашего объекта есть возможность к пластичности. Камень не будет растягиваться при движении или расплющиваться при соприкосновении с поверхностью, а резиновый мяч будет. А если в качестве объекта у нас капля воды — эффект будет максимально заметен.



Отредактируем **графики характера движения для изменения масштаба**. От 10-го до 20-го кадра график будет аналогичен графику перемещения. Замедление

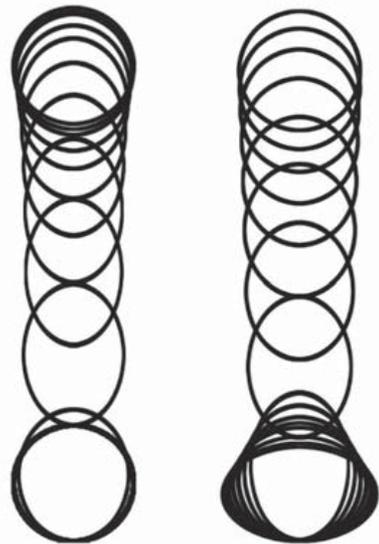
в начале без замедления в конце. От 20-го до 26-го кадра добавляем замедление в конце. К 20-му кадру и от 20-го кадра нет замедления, так как он не делит движение мячика на фазы. Если мы добавим замедление в этом кадре, мы нарушим логику движения и разобьем его на сегменты там, где это не нужно. С 26-го по 32-й кадр добавляем график с замедлением в начале и конце.



Посмотрим на нашу анимацию в медленном формате. Обратите внимание, что мячик при движении вверх полностью растягивается, находясь еще на земле, что выглядит странно. Будто его что-то держит за нижнюю точку. Нам нужно сделать так, чтобы это растяжение происходило позже.

Выделим 30-й кадр. Кадр нужно взять тот, где масштаб близок к единице. Мы возьмем 30-й, чтобы округлить количество кадров анимации. Можно взять и 29-й кадр, в котором масштаб даже ближе к единице, чем в 30-м. Но с некруглым количеством кадров неудобно работать.

Нажав на иконку ключика, фиксируем ключ масштаба в этом кадре. И выделяем кадры 30 и 32, копируем их и вставляем в кадры 0 и 2. Удаляем кадр 32, так как он уже принадлежит к следующему циклу анимации.



Покадровая интерпретация сделанной нами анимации прыжка

Теперь нам нужно подкорректировать график характера движения для масштаба. Со 2-го по 10-й кадр добавляем замедление в начале и конце. И наша анимация прыгающего мячика готова!

Скачок мяча в сторону

На основе прыжка мяча сделаем скачок в сторону. По сути, мы добавим к предыдущему примеру движение в сторону, а также затухание прыжка мяча со временем. В этом примере мы обратим внимание на создание движения по дуге и изменение эффекта сжатия-растяжения с затуханием силы движения.

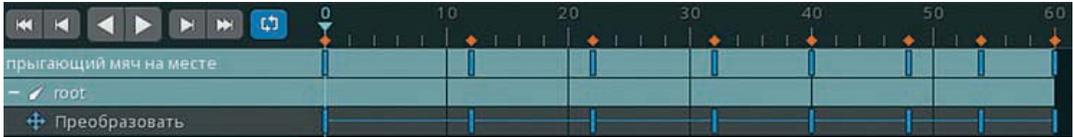
Разобьем это упражнение на два. Сначала сделаем затухающий прыжок на месте, а затем добавим движение по горизонтали.

Задаем основу движения. Этот этап актуален для любой анимации. Мы всегда начинаем с базы. Анимация будет длиться 60 кадров. Мы начинаем анимацию с верхней точки, а не с нижней, как в предыдущем примере. То есть сначала мячик будет падать, а затем еще три раза подпрыгнет на поверхности, постепенно замедляясь. Каждый раз высота прыжка будет все меньше, и количество кадров на этот прыжок тоже будет меньше. Мы не будем четко рассчитывать значения по формулам, для анимации хватит визуально верного восприятия. А значит, нам достаточно соблюсти соотношение: чем меньше высота прыжка, тем меньше расстояние, пройденное по горизонтали, и тем меньше количество кадров на прыжок. Если нам захочется более точной анимации — на финальном этапе всегда можно будет подправить дуги движения.

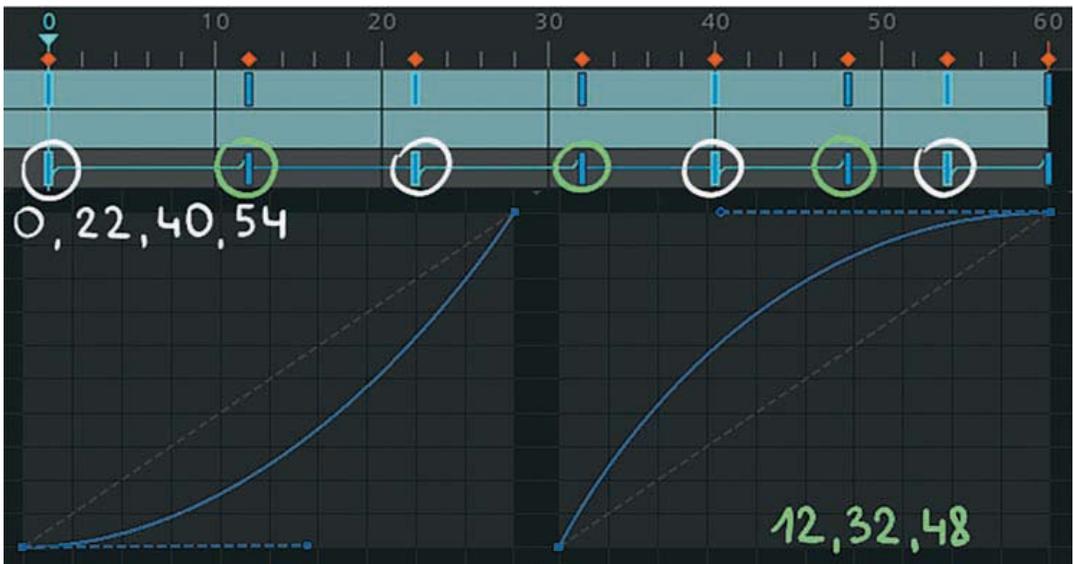
Итого мы имеем следующие значения высоты для кадров.

0	2000
12	0
22	1500
32	0
40	700
48	0
54	250
60	0

Высота прыжка у нас меняется от 2000 до 1500, а затем до 700, 250 и уходит в 0. Количество кадров на прыжок, соответственно, также уменьшается. Изначально у нас было 24 кадра на прыжок, но поскольку мы начинаем с верхней точки, то задействуем только 12 кадров. Далее на прыжок уйдет всего 20 кадров, затем 16 и 12.

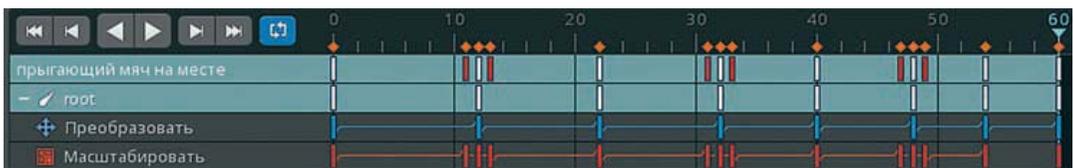


Далее аналогично предыдущему упражнению настраиваем график характера движения. Когда мячик падает, происходит замедление движения в начале (перед столкновением с поверхностью мячик не замедляется!). Когда он подлетает, происходит замедление в конце.



Теперь добавляем растяжение при движении и взаимодействии с поверхностью. Отличие от предыдущего примера в мгновенности анимации столкновения. У нас нет достаточного количества времени на то, чтобы в деталях показывать этот момент анимации. А значит, нам нужно сосредоточиться на основных ключевых кадрах.

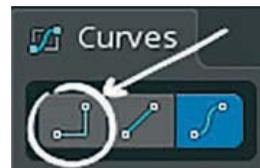
Мы выделим всего по одному кадру до и после каждого столкновения с поверхностью. С каждым прыжком эффект масштаба будет все менее значительным и в конце придет к единице. Степень пластичности зависит от скорости, с которой мячик ударяется о землю. С каждым новым прыжком скорость будет уменьшаться, а значит, и сжиматься мяч будет меньше.



После работы со столкновениями ключи будут выглядеть так

Итак, в 11-м кадре ставим значения сжатия 0,7 и 1,42. В 12-м кадре, наоборот, меняем значения местами. В 13-м кадре уменьшаем эффект масштаба и ставим значения 0,8 и 1,25.

Далее нам нужно задать мгновенную смену кадра с 11-го на 12-й. Это делается с помощью нажатия на первую иконку «ступенчатой кривой» в верхнем левом углу окна графика. Мы делаем это для того, чтобы мячик в максимально растянутом состоянии достиг пола и затем сжался от столкновения с землей.



Характер движения с 0-го по 11-й кадр и с 13-го по 22-й аналогичен характеру перемещения (синие ключи). С 12-го по 13-й кадр оставляем равномерное движение, прямую линию графика.

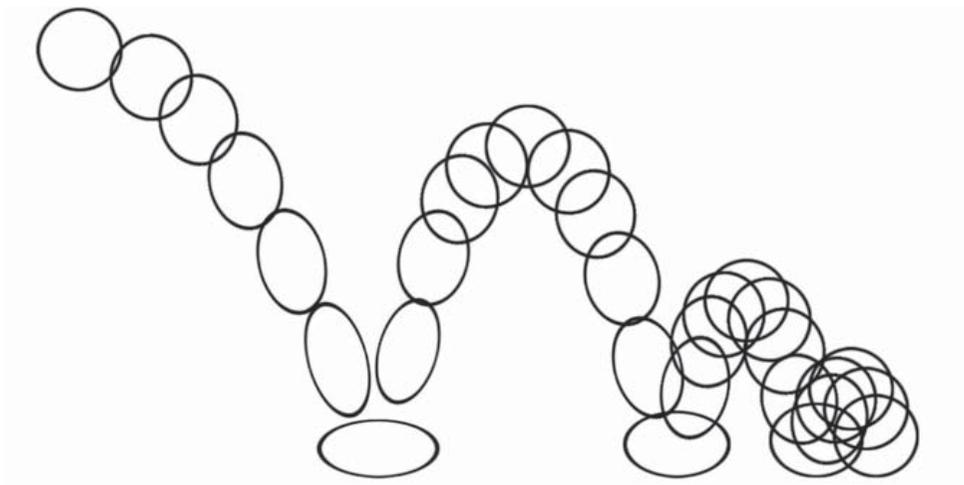
Для остальных моментов столкновения с поверхностью делаем то же самое, только с другими значениями масштаба. Значения масштаба 31-го кадра совпадают с 13-м. В 32-м кадре мы меняем их на противоположные. Также мы будем менять значения на противоположные предыдущему кадру в остальных кадрах столкновения. В 33-м кадре ставим значения 0,9 и 1,1, и в 49-м кадре — 0,95 и 1,05, почти единичка. Анимация затухающего прыжка мячика на месте готова!

Перейдем ко второму этапу упражнения. Нам осталось только добавить **движение по горизонтальной оси**. Первое, что нужно сделать, это нажать галочку «Преобразовать» в строке «Отдельно» в правой части экрана. Если галочка будет выключена, то значения по обеим осям запишутся в один ключ. Если поставим галочку, мы будем контролировать значения по оси X и по оси Y отдельно. На шкале времени появится дополнительная строчка синих ключей. Сейчас мы будем работать со строчкой «Преобразовать по оси X».

В кадрах столкновения с землей по очереди выставляем следующие значения: 1500, 3000, 3700, 4000.

Финальный штрих — **добавление поворота мячику**. Это важный шаг, так как у мячика есть направление движения по дуге, а значит, растяжение в масштабе должно происходить вдоль этого направления. Работать будем опять же с кадрами столкновения и рядом с ним. Нам нужно повернуть наш мячик в кадре, предшествующем столкновению, и последующем кадре, согласно направлению движения. А в самих кадрах столкновения и кадрах верхней позиции прыжка поворот равен нулю. Итак, выставляем следующие значения (используя инструмент «Повернуть»).

11	15
13	10
31	10
33	8
47	8
49	5



Так выглядит наша итоговая анимация

Маятник

В примере с маятником мы возьмем за основу пластичную веревку, чтобы показать деформацию при анимации. Для создания анимации этого элемента мы будем использовать меш. При движении влево-вправо веревка будет искажаться, а значит, нам понадобится меш, чтобы деформировать исходную картинку.

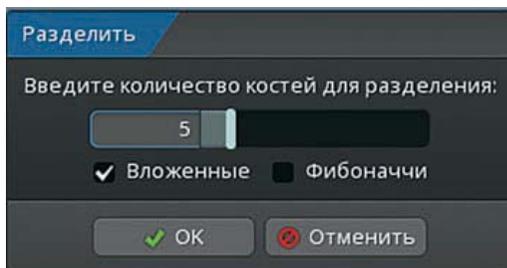
Создадим скелет.

В этот раз скелет уже будет более сложным. Составим его из нескольких костей, привязанных к разным картинкам. От начала до конца веревки создаем одну длинную кость. У мячика мы также создаем одну кость в месте стыка с веревкой. Кость мячика должна быть дочерней к кости веревки. Чтобы привязать кость к определенной картинке, нужно выделить ее, одновременно нажав клавишу Ctrl.

Для создания изгиба мы должны поделить кость веревки на несколько одинаковых костей. Чем больше будет количество звеньев цепочки, тем плавнее получится сгиб. Но для оптимизации и удобства контроля анимации необходимо всегда стараться добиться нужного результата минимальными средствами.



Выделяем кость и нажимаем кнопку «Разделить» в правом нижнем углу окна программы. Высвечивается окно настроек разделения.

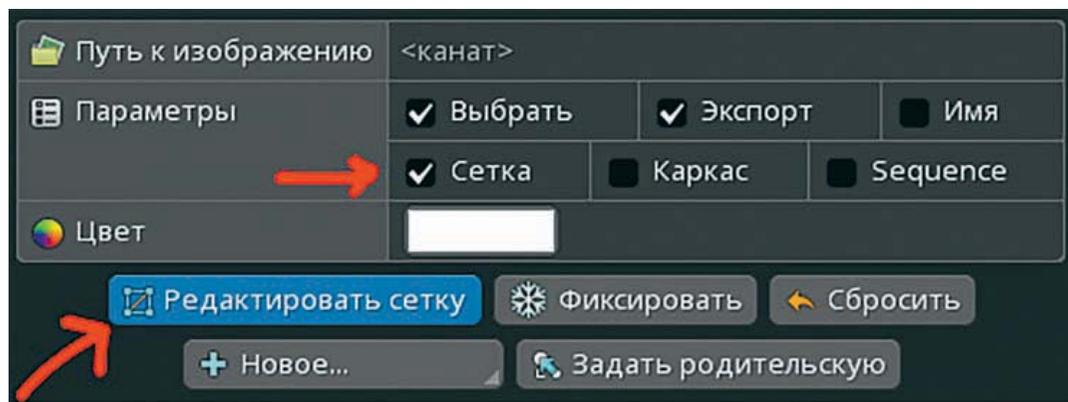


В этом окне мы можем задать количество костей, полученных в результате деления. Мы зададим 5. Поставив галочку возле настройки «Вложенные», мы создадим иерархическую структуру, в которой каждая следующая кость будет дочерней к предыдущей. Поставив галочку возле настройки «Фибоначчи», мы зададим изменение размера костей. Каждая следующая кость будет меньше предыдущей. В данном упражнении мы эту галочку ставить не будем.

После применения разделения проверим структуру костей. Кость мячика должна быть дочерней к последней кости веревки. Теперь в панели справа необходимо перенести кость мяча в нужное положение в панели иерархии.

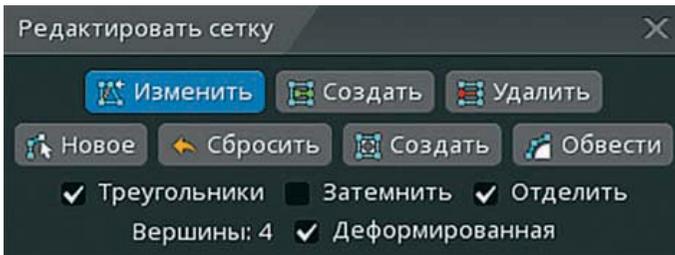
СОЗДАНИЕ МЕША

В панели иерархии проекта выбираем картинку веревки. Это должно быть именно изображение, а не слот, будьте внимательны. В нижней части панели ставим флажок в поле «Сетка». Нажимаем кнопку «Редактировать сетку».



Высвечивается окно с настройками нашего меша. Сейчас мы не будем углубляться в значение каждой кнопки этой панели, так как нам нужен простейший прямоугольный меш с определенным количеством точек для сгиба. Схема построения подобного меша следующая. Нам необходимо создать по две точки меша с каждой стороны веревки в месте стыка костей и посередине каждой кости. Почему именно столько точек? Поскольку мы стараемся добиться цели минимальными средствами, то большее количество точек будет избыточным. Увеличение детализации меша не даст

существенного прироста в плавности. А вот уменьшение сразу даст углы при деформации. Мы выбираем наилучший вариант из соотношения количества точек и плавности сгиба.



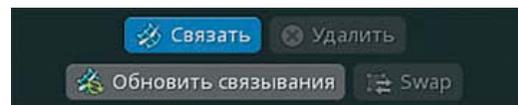
Чтобы вручную создать точки меша, нажимаем кнопку «Создать» в верхней строке панели и кликаем мышкой поочередно в нужных местах. Такой меш можно создать и автоматически. Для этого необходимо нажимать кнопку «Создать» в нижней строке окна до тех пор, пока точки не окажутся в нужных нам местах. Эта кнопка постепенно ставит новые точки на меш, соответственно добавляя детализации. Но учтите, что это работает только в описанном конкретном случае. В общем эта кнопка просто добавляет деталей в меш. Эта функция может работать непредсказуемо и даже менять существующий меш.

Если вас не устраивает положение какой-либо точки, то ее всегда можно удалить. Для этого нужно нажать кнопку «Удалить», навести курсор на ненужную точку и кликнуть на нее. Или можно передвинуть точку с помощью кнопки «Изменить».

Далее нужно привязать меш к костям и настроить их влияние на его точки. Суть деформации в Spine с помощью меша состоит в том, что к мешу привязывается группа костей. Каждая из этих костей влияет только на определенную область изображения. При изменении положения кости меняется положение только части изображения, а значит, изображение искажается. Также можно назначить степень влияния кости на определенные области. Этим мы сейчас и займемся.

Сначала вызовем панель настройки влияния костей на точки меша. Она называется «Вес». Степень влияния кости на точку в Spine называется «вес» и отображается определенным цветом. Выделяем изображение веревки в рабочей области или панели иерархии. Нажимаем кнопку «Вес» в левой нижней панели. Высвечивается окно настройки весов, нажимаем кнопку «Связать» в самом низу окна.

Выделяем все кости веревки. Они появляются в списке, и каждая обозначена своим цветом.

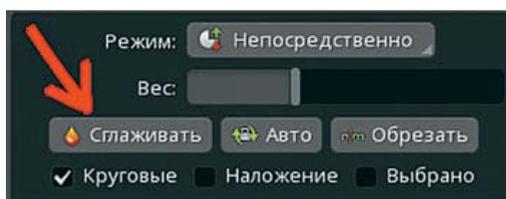


И еще раз нажимаем кнопку «Связать», чтобы привязать кости к мешу. У каждой точки появляется кружок определенного цвета. На некоторых кружках может быть несколько цветов. Это и есть отображение весов костей в пропорциях круга. Если круг окрашен в один цвет, значит на него влияет только одна кость этого цвета. Если сегментов круга разного цвета несколько, значит на эту точку влияет несколько костей соответствующих цветов. Степень влияния пропорциональна размеру сегмента круга. Если нажать на круг, то в списке костей появятся точные значения влияния в процентах.

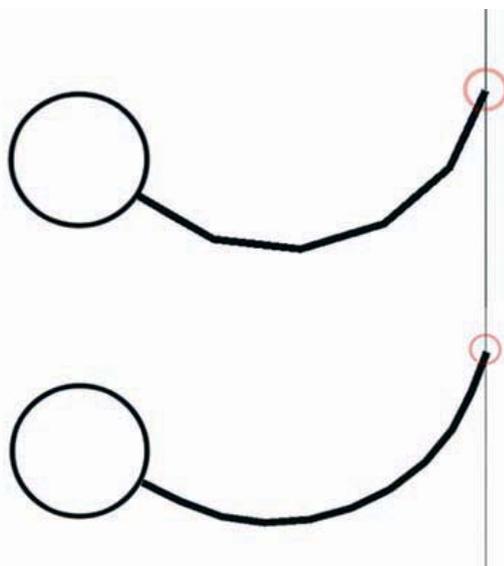


Автоматическая настройка весов не всегда сразу дает нам нужный результат. Давайте посмотрим, как искажается веревка. Выделяем все кости и пробуем повернуть ее. Мы видим, что на веревке появились изломы, которые мешают плавности изгиба. Давайте это исправим.

Переходим в настройки весов. Нажимаем два раза на кнопку «Сглаживать» в верхней части окна.



И снова пробуем повернуть веревку. Идеальной плавности мы не получили, но изломы стали в разы менее заметными. Мы не будем заворачивать в рулет наш маятник, поэтому данный результат подойдет для этого примера. Кнопка «Сгладить» как бы «размывает» влияние костей, добавляя небольшие сегменты в соседние точки.

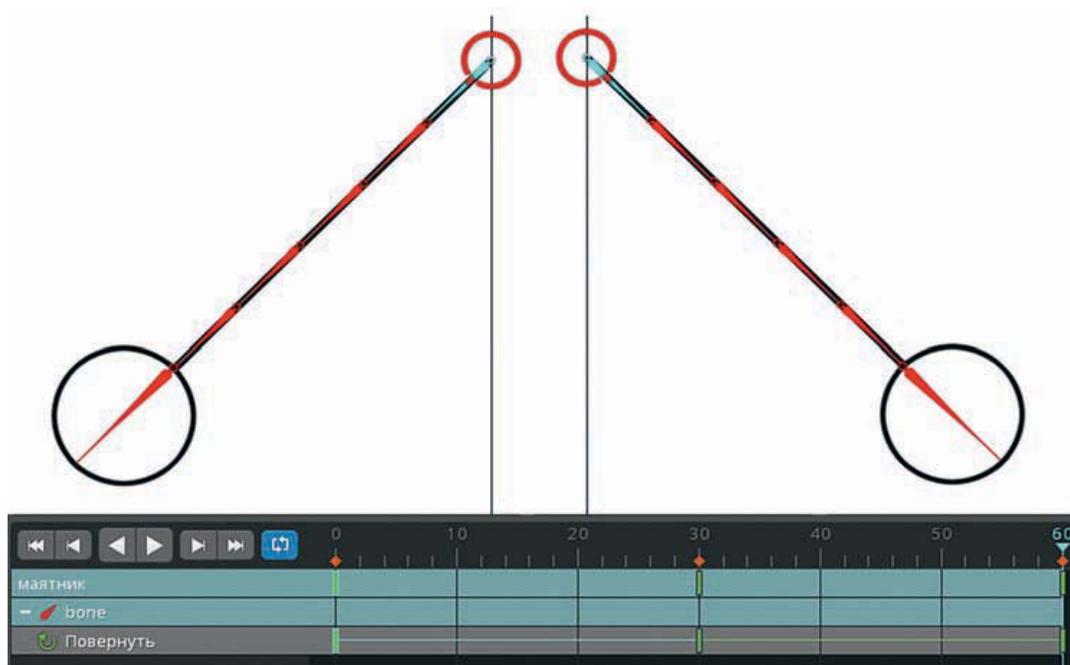


До и после сглаживания

АНИМАЦИЯ

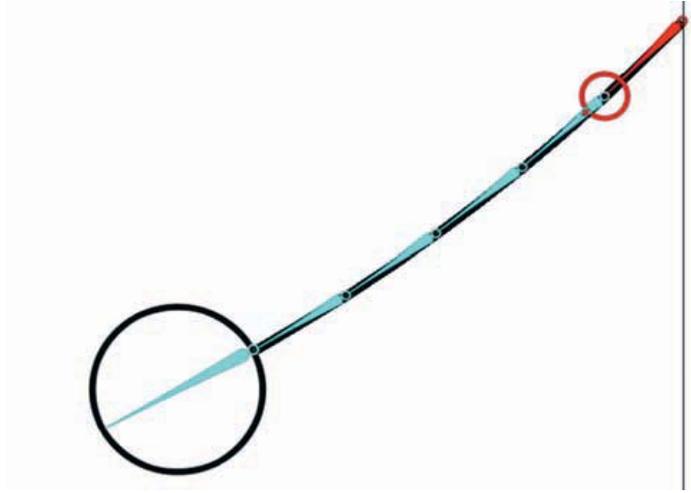
Вы уже ознакомились с интерфейсом программы на предыдущих примерах. Замечаете, как с каждым новым упражнением вы все лучше ориентируетесь в окнах и кнопках программы? Общий смысл работы с анимацией в этом примере остается тем же. Мы должны задать основу движения, закрепляя ключи в нужных нам кадрах, а затем добавить изгиба веревке и мультяшности анимации.

Выделяем самую верхнюю кость и в 0-м кадре поворачиваем ее влево на 45 градусов. Значение угла в поле «Повернуть» будет минус 135 градусов. В 30-м кадре ставим значение поворота минус 45 градусов. Таким образом, маятник отклоняется от начального положения на 45 градусов в обе стороны. Копируем 0-й кадр в 60-й, чтобы зациклить анимацию. Основа движения задана.



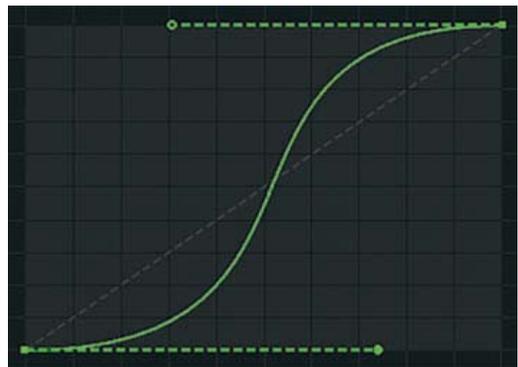
Теперь зададим пластичность веревки. Выделяем все остальные кости, кроме верхней. Выделяем 0-й кадр, фиксируем ключ поворота и перетаскиваем ключи в 30-й кадр. Мы делаем так для сохранения прямого положения и более удобного поворота в этом кадре.

В 0-м кадре, кадре крайнего левого положения, поворачиваем все кости влево. Тем самым создаем изгиб. Степень изгиба будет соответствовать степени пластичности веревки. Советую не задавать слишком сильный поворот и ограничиться лишь несколькими градусами. Все-таки веревка натянута под грузом мячика, и мы не можем сильно ее скручивать. Это не соответствует физике движения. Ту же самую операцию, только в правую сторону, делаем в 30-м кадре. И копируем 0-й кадр в 60-й.



Добавляем изгиб веревке

Снимаем все выделения клавишей Esc, чтобы на шкале времени отобразились все кости. И приступаем к настройке характера движения. Выделяем все ключи рамкой. Вызываем панель графика характера движения, если она у нас не открыта. Ставим график «кривая Безье» и усиливаем этот эффект.



ЭФФЕКТ РАЗМЫТИЯ В ДВИЖЕНИИ

В целом анимация маятника на этом этапе завершена, но я хочу показать вам еще один прием на этом примере. Мы добавим эффект размытия мячику. Это делается для того, чтобы добавить мультяшности движению, приблизиться к виду покадровой анимации. В анимации движения по прямой уже был использован данный прием. Но тогда все было максимально просто, так как для создания размытия достаточно было масштабировать объект. В данном примере мы не можем просто использовать масштаб. Мяч привязан к веревке, и эта точка не может смещаться. Плюс к тому размытие должно происходить в определенную, противоположную движению, сторону. Если мы будем использовать масштаб, то шарик будет расширяться в обе стороны сразу относительно веревки, поэтому нам снова нужен меш.

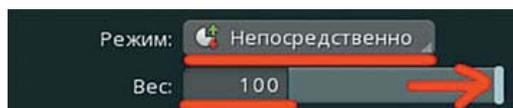
Меш будет простейшим, но даже для него нам понадобится небольшая дополнительная настройка скелета. Для контроля деформации добавляем две кости слева

и справа от мяча. Обе новые кости являются дочерними по отношению к центральной кости мяча.

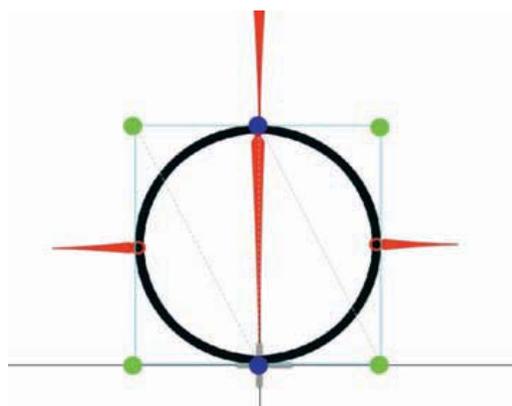
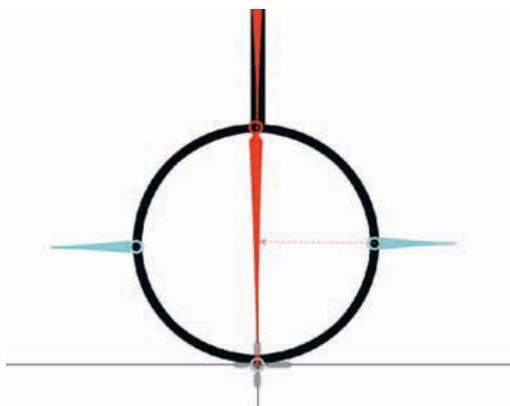
Создаем меш для изображения мяча. По умолчанию уже создан квадратный меш, в который вписан мяч. Добавляем две точки, которые будут делить этот квадрат пополам по вертикали. Первая дополнительная точка соответствует месту крепления веревки, а вторая располагается на противоположной грани квадрата в том же положении.

Переходим в панель настройки весов и связываем меш с тремя костями — с основной костью мяча и двумя дополнительными, созданными нами. Веса распределяются автоматически, но нам это распределение не подходит. Наша цель — левой костью исказить только левую часть мячика относительно центра. Правой — соответственно, только правую. А центр должен контролироваться центральной костью, оставаться привязанным к веревке и не искажаться.

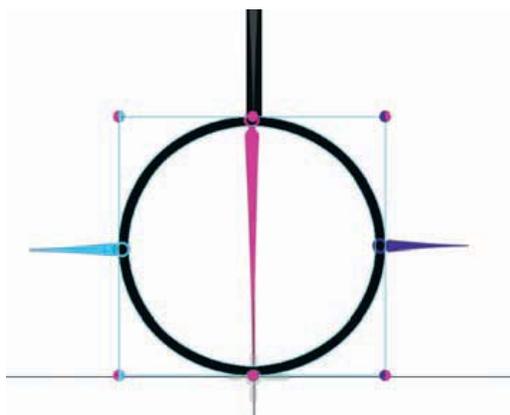
Нам нужно убрать дополнительное влияние костей на соседние точки. Левая кость должна влиять только на две левые точки, центральная — на центральные и правая — на правые. Кружки должны быть окрашены лишь в один цвет, без деления на сегменты. Выделяем кость в режиме «Непосредственно». Нажимаем на соответствующий круг и тянем бегунок «Вес» вправо до 100%.



Делаем так же для всех костей и соответствующих им узлов.

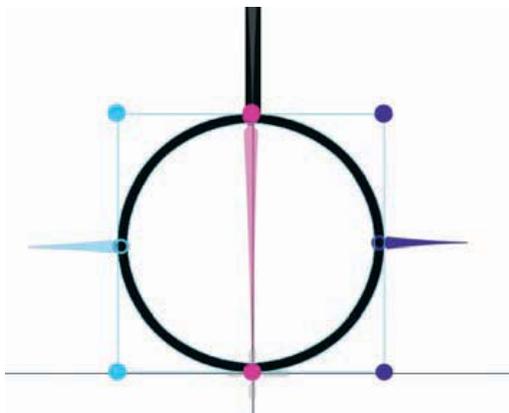


Центральные точки, отмеченные синим, мы создаем вручную. Зеленые создаются автоматически

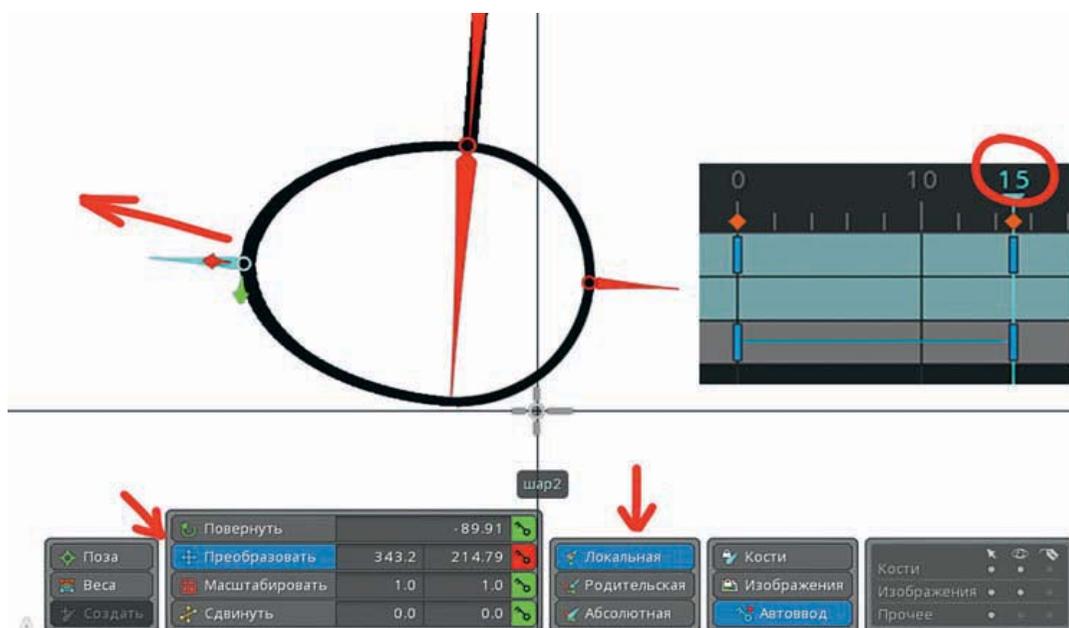


Первоначальное автоматическое распределение весов

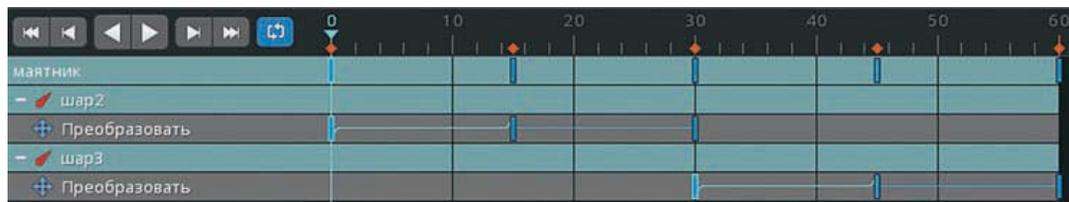
Переходим в анимацию. Фиксируем ключ преобразования для левой кости мячика. Выделяем 15-й кадр и тянем за эту кость. Растягиваем мяч в левую сторону, тем самым добавляя размытие в движении. А 0-й кадр копируем в 30-й, чтобы убрать эффект размытия. Все преобразования делаем в локальной оси объекта, так как нам нужна деформация под углом движения. В абсолютной же шкале мяч будет деформироваться строго влево и вправо.



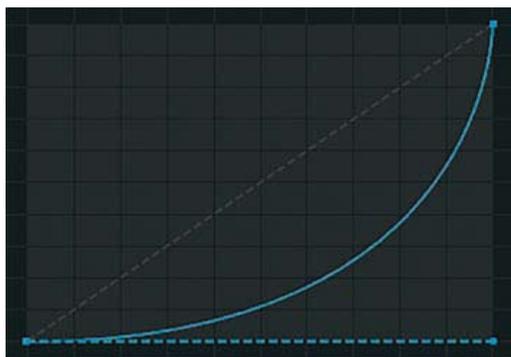
Итоговое распределение весов



В 30-м кадре фиксируем положение правой кости мяча. В 45-м кадре тянем кость вправо на столько же, на сколько вытянули левую влево. Размытие должно быть одинаковым в обоих направлениях. А 30-й кадр копируем в 60-й.



Чтобы размытие происходило не сразу по началу движения, настраиваем графики. В 0-м кадре для левой кости и в 30-м для правой делаем максимальное замедление в начале без замедления в конце. В 15-м и 45-м — максимальное замедление в конце без замедления в начале для каждой кости соответственно. С такой настройкой размытие будет происходить только близко к нижней точке движения.

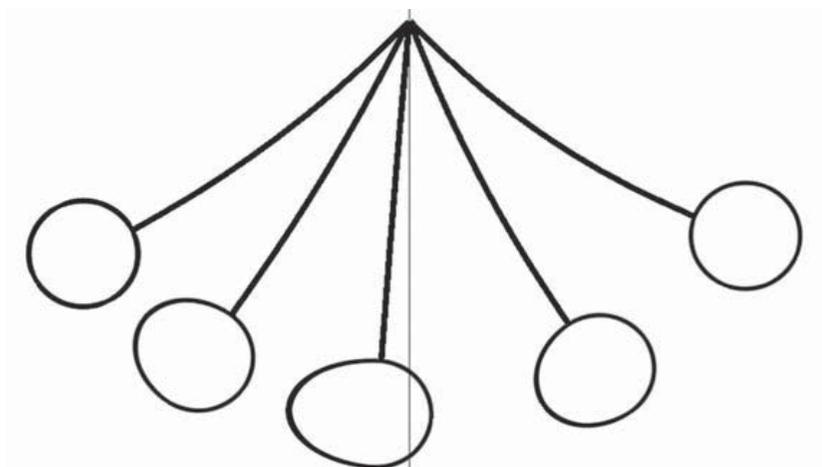


0-й и 30-й кадры



15-й и 45-й кадры

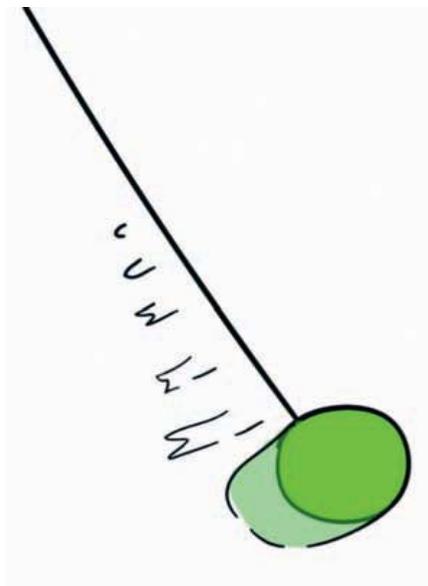
И вот наша анимация готова. Обратите внимание, как размытие в движении добавляет мультяшности и характера анимации. Максимальное размытие в нижней точке движения подчеркивает максимальную скорость в этом месте. За счет графиков мы также обеспечили это размытие только на интервале возле нижней точки. Как видите, на промежуточных средних кадрах это размытие едва заметно. Чтобы подчеркнуть скорость объекта, размытие нужно добавлять только в том интервале, где она максимальна.



Последовательные кадры анимации

В покадровой анимации можно встретить дополнительные эффекты для движения. Например, эффект прохождения сквозь воздух в виде пыли или просто линий по направлению движения элемента. В Spine также можно добавить подобный эффект с помощью заранее нарисованных кадров.

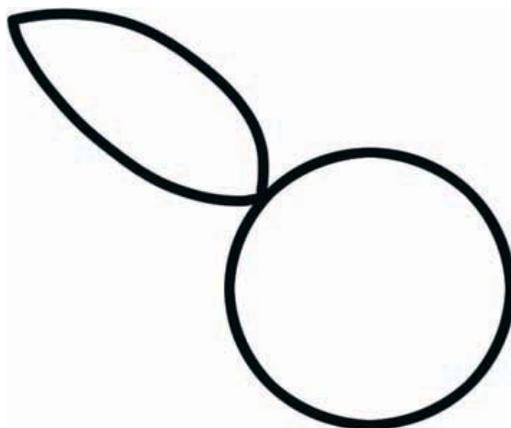
Но подобные эффекты добавляют в анимацию визуальный шум и даже чрезмерную мультяшность. Поэтому нужно соотносить количество средств выразительности со стилем проекта, в котором работаете. Подобные эффекты размытия не всегда будут смотреться органично. Например, на мелких элементах это только перегрузит анимацию деталями. Обилие деталей и эффектов на второстепенных элементах может отвлечь зрителя от важных движений и сегментов анимации.



Пример эффекта быстрого движения из покадровой анимации

Дополнительный элемент. Добавляем хвост движению

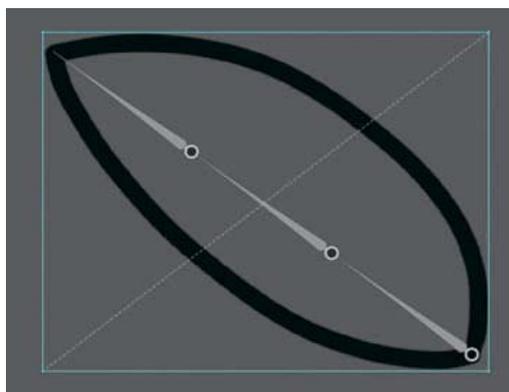
При выполнении этого упражнения мы добавим к уже существующему движению мяча хвост. Работать будем в том же самом проекте. Дополнительное изображение загрузим в папку проекта и добавим на рабочую область. Этот элемент позволит нам познакомиться с «запаздывающим» движением, так как хвост будет все время повторять движение шара, только с задержкой времени. Деформация хвоста будет создана с помощью меша, что позволит познакомиться с этим инструментом поближе. Подобный элемент можно добавить к каждому предыдущему упражнению. Мы рассмотрим движение дополнительного элемента на примере маятника.



Хвост для новой анимации

Создаем кость по центру шарика, чтобы контролировать поворот хвоста. Получаем кость *root* в нижней точке мяча. Дочерняя к ней кость находится в центре мяча. Далее переходим к настройке костей и меша хвоста.

Меши уже были применены на базовом уровне в примерах с мячиком. Но нам еще не приходилось подробно настраивать меш для различных фигур и задач. Мы не меняли грани меша и не разбирались, как делать сильные изгибы элементов без артефактов. Поскольку меш — один из основных инструментов программы, наша задача — изучить его более подробно. Рассмотрим новые функции на примере создания меша для хвоста.



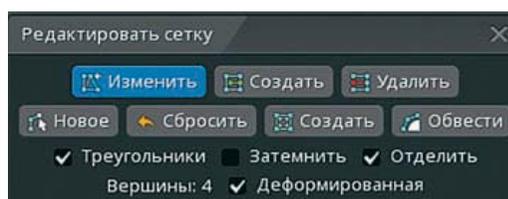
Меш, созданный по умолчанию

Нам нужно деформировать хвост поворотом, примерно как нить маятника из предыдущего примера. Но теперь у элемента есть толщина, что добавляет проблем при деформации. Начиная с основания хвоста, создадим три кости, каждая из которых будет дочерней по отношению к предыдущей. И создадим базовый меш. Меш создается в виде прямоугольника с вписанным в него по диагонали хвостом. В этот раз нас уже такой результат не устроит, потому что форма элемента более произвольная. А значит, мы будем тщательнее и детальнее настраивать меш.

МЕШИ. ЗНАКОМИМСЯ ПОБЛИЖЕ

Давайте подробнее посмотрим на окно создания меша. Мы изучили не все его кнопки и функции. Настало время разобрать каждую кнопку.

Верхняя строчка уже использовалась в предыдущем примере. «Создать», «Изменить», «Удалить» — инструменты работы с вершинами меша. С их помощью мы можем создавать и удалять вершины, а также менять их положение на меше.

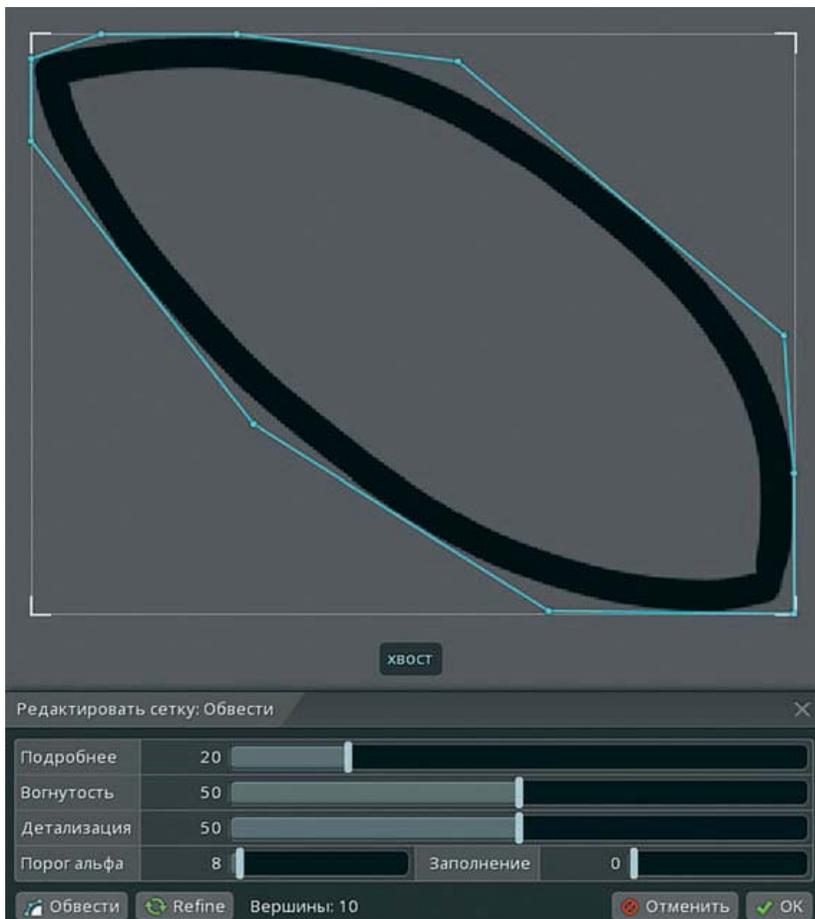


Кнопка «Новое» используется для удаления всех существующих вершин и граней и создания меша заново. После нажатия на эту кнопку нужно проставить точки меша одну за другой, создав замкнутый контур. Точки можно перемещать вручную прямо в процессе создания меша. Чтобы завершить создание меша, нужно нажать на первоначальную вершину повторно, тем самым замкнув контур.

Кнопка «Сбросить» возвращает меш по умолчанию. Что бы вы ни создали в процессе работы, эта кнопка удалит все и вернет меш из четырех точек по краям изображения.

Кнопка «Создать», находящаяся во второй строке, добавляет детализации созданному мешу. С каждым следующим кликом программа ставит новые точки там, где, по ее мнению, их не хватает. Это автоматическая функция, которая хорошо работает для геометрически правильного меша, особенно прямоугольного. Но с произвольным мешем она будет добавлять случайную детализацию, которая далеко не всегда помогает добиться лучшего результата. Позже мы проверим это утверждение, создав два меша с разной детализацией и сравнив результаты изгиба.

Функция «Обвести» автоматически создает точки меша по контуру объекта. При нажатии этой кнопки создается базовый меш по контуру и появляется окно настроек этого инструмента.



«Подробнее» определяет количество вершин. Число в этом пункте не отображает настоящее количество вершин, это скорее степень детализации. Программа каждый раз просчитывает алгоритм и добавляет определенное количество вершин. Это автоматическая функция, результат которой нужно контролировать визуально. С помощью

минимального количества точек необходимо добиться максимально близкого контура меша к объекту, с минимумом пустот.

«Вогнутость» помогает автоматически обводить вогнутые сегменты объекта. При низком значении программа, скорее всего, не заметит вогнутость и проставит точки поверх нее. Как будто этой «ямки» на объекте нет. Более высокое значение повышает вероятность того, что вогнутость будет замечена алгоритмом автоматического создания меша. Но мелкие впадинки программа, скорее всего, все равно не заметит, и их придется настраивать вручную.

«Детализация» определяет время и усилия программы на подсчет положения точек. На сложных формах этот параметр можно увеличить, тогда ваш меш получится более аккуратным. На простых формах особой разницы заметно не будет.

«Порог альфа» позволяет контролировать входение прозрачных пикселей в меш. Пиксели со значением прозрачности ниже установленного в этом пункте порога не учитываются. Это помогает избавиться от артефактов источника. Бывает, что в изображении есть почти невидимые пиксели с прозрачностью. И если не подкорректировать параметр «Порог альфа», то они будут обводиться мешем. И даже при приближении, казалось бы, четкой линии вы все равно увидите некоторое размытие на краях. Также этот параметр важен при создании меша для эффектов, которые часто используют прозрачность. Но в этом случае нам, наоборот, нужно, чтобы все области эффекта входили в меш.

«Заполнение» позволяет отдалиться от контура объекта при большом положительном значении этого показателя. Если увести показатель в минус, то программа, наоборот, будет заходить мешем внутрь контура при подсчете положения точек. Значение нуля дает максимально четкое обведение контура объекта.

В нижней части окна вы можете заново обвести и уточнить положение точек. Рядом отображается количество вершин меша и кнопка отмены или применения автоматического меша.

Вернемся к разбору основного окна создания меша. Под кнопками в этом окне есть еще набор флажков настроек.

Пункт «Треугольники» отражает автоматически просчитанные грани меша, которые показываются пунктирной линией между вершинами.



«Затемнить» затемняет изображение, чтобы на нем были лучше видны точки и грани меша.

«Отделить» скрывает ненужные изображения и показывает только то, для которого сейчас настраивается меш. Таким образом, большое количество элементов не мешает при настройке.

Пункт «Деформированная» всегда остается включенным. Отключают его очень редко, в специфических ситуациях.

СОЗДАЕМ МЕШ

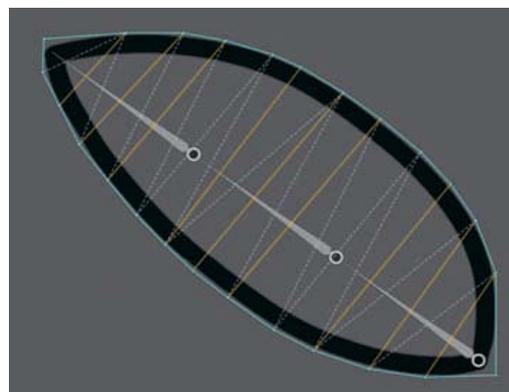
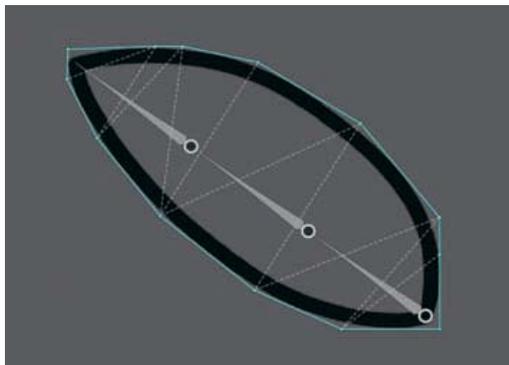
Сначала используем функцию «Обвести» с базовыми настройками. Делаем это просто для того, чтобы меш подошел ближе к контуру фигуры. Так гораздо удобнее для редактирования.

Теперь мы должны отредактировать вершины и, главное, грани меша. Нужно подвинуть точки так, чтобы через стыки костей обязательно проходило ребро меша. Пунктирная линия должна проходить через стыки костей. Через каждую кость также должно проходить определенное количество ребер. Точное значение можно выяснить опытным путем. Сначала нужно ребром поделить кость пополам, потом проверить сгиб. Также количество ребер зависит от размера изображения в итоговом виде и степени искажения, которая нам потребуется в анимации. Для большего искажения понадобится большая детализация. Мы остановимся на трех ребрах. Будем считать только основные ребра, а не дополнительные диагональные. Необходимые ребра отмечены на картинке желтым цветом.

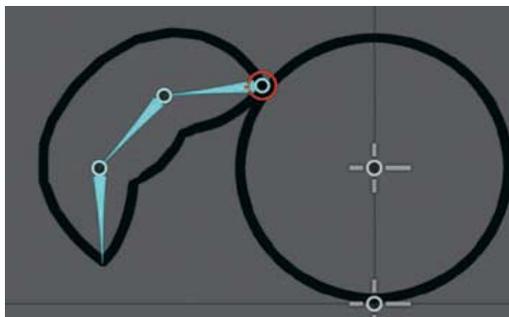
Положение точек также нужно скорректировать. Постараемся минимизировать прозрачную область за контуром. Но при этом мы не должны обрезать мешем изображение.

Привязываем меш к костям и назначаем веса автоматически. После этого необходимо проверить объект на сгибаемость. Проверять нужно в анимации, а не в настройке, чтобы не изменять исходник. Обратите внимание, что на внутренней стороне хвоста есть артефакты сгиба. Это значит, что значения весов недостаточно размыты.

Для того чтобы убрать этот эффект, нам нужно сгладить веса костей. Нажимаем на кнопку «Сглаживать» до тех пор, пока результат сгиба нас не устроит. В данном

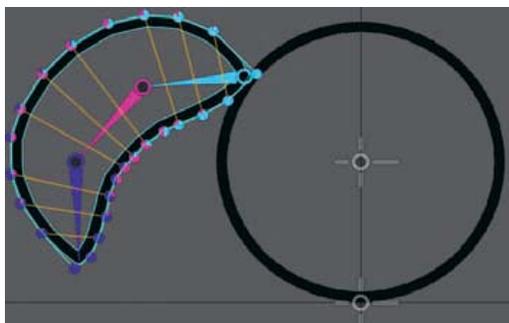


Итоговый меш для хвоста



Неплавный сгиб объекта

примере достаточно нажать четыре раза. После этого мы получаем довольно плавный сгиб с обеих сторон хвоста. Нужно учитывать степень скручивания элемента. Мы настраиваем меш для максимально возможного сгиба. С такими настройками мы не сможем скрутить хвост сильнее. Но для данной анимации настройки и детализации меша будет достаточно. На этом настройка под анимацию завершена, и мы переходим к созданию движения.



Итоговый меш хвоста с весами костей

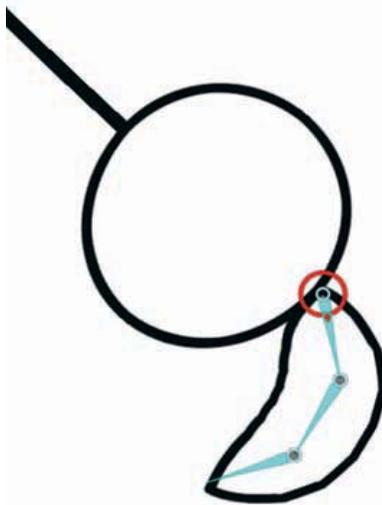
АНИМАЦИЯ

Мы будем использовать уже готовую анимацию маятника, дополняя ее анимацией хвоста и редактируя под новый элемент. Когда мячик движется вниз, хвост вытягивается кончиком вверх, а когда мяч движется вверх, хвост заворачивается кончиком вниз. Причем хвост несколько запаздывает за движением основы, к которой прикреплен. Вот такую схему движения мы создадим.

Хвост необходимо прикрепить к нижней точке маятника. Для этого выделяем первую кость хвоста и переносим ее в нужное положение. Поворачиваем хвост вниз.

В 20-м кадре нужно задать поворот хвоста в сторону, противоположную движению маятника. Когда маятник летит вправо, хвост должен поворачивать влево. Для этого выделяем все три кости хвоста и используем инструмент поворота. Мы берем именно 20-й кадр, потому что в нем начинается замедление маятника. Выбираем примерно, ориентируясь на глаз и на удобство работы по шкале времени. Круглые значения всегда предпочтительнее, чем случайные.

Далее выбираем 40-й кадр, это финальная точка движения маятника. В этой точке нужно развернуть хвост вправо. Хвост движется в сторону движения маятника по инерции, хотя сам маятник уже завис в этой точке. Добавляем между 20-м и 40-м кадром замедление в начале и конце. Для быстрого задания стандартного вида графика с замедлением в начале и конце

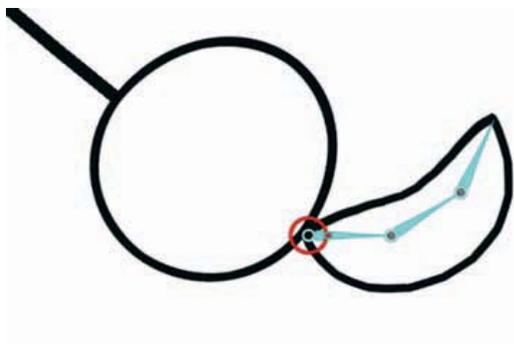


20-й кадр

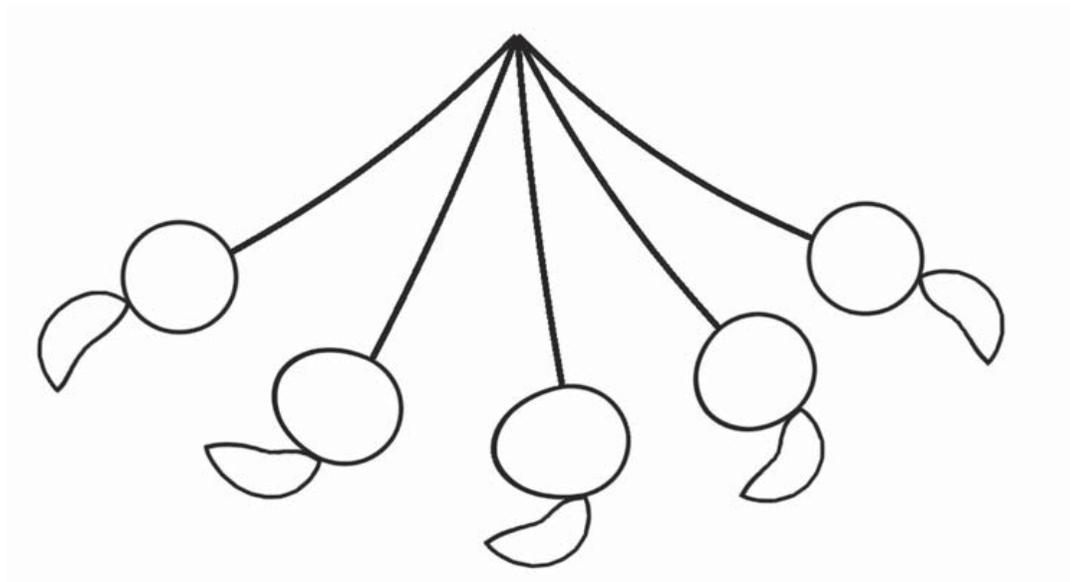
можно просто нажать на соответствующую иконку графика в панели над шкалой времени.

Теперь создадим аналогичное движение хвоста для движения маятника влево. Мы копируем 40-й кадр в 50, 20 и 70-й. И также делаем замедление в начале и конце между 50-м и 70-м кадрами. Мы вышли за границы анимации маятника, которая заканчивается в 60-м кадре. Нам нужно эту выступающую часть скопировать в начало анимации. Выбираем 60-й кадр и фиксируем ключ поворота. Копируем 60-й и 70-й кадры и вставляем в 0-й и 10-й соответственно. Удаляем 70-й кадр.

Финальный штрих. Нам нужно добавить запаздывание движения внутри самого хвоста. Выбираем инструмент «Сместить» и смещаем анимацию поворота для второй и третьей кости хвоста. Вторую (центральную) кость смещаем на 5 кадров вперед. А третью (конечную) — на 10 кадров вперед. И анимация хвоста маятника готова.



40-й кадр



ГЛАВА 6

**Подготовка
материалов
для анимации
в Spine 2D**

Выбор концепта для работы. Какие исходники лучше подходят для скелетной анимации

Если рассмотреть и сравнить примеры игр и мультфильмов, в которых используется скелетная анимация, то можно найти схожие моменты в визуальном стиле. Общие черты обусловлены спецификой и ограничениями скелетной анимации. В основном ограничения на стиль обусловлены тем, что при нарезке на части и последующем движении вместо цельной картинке мы имеем дело с отдельными частями. Детализация, которая была на изначальном элементе, тоже делится на части. И при этом разделении нарушается смысл деталей или они перестают стыковаться друг с другом. Именно поэтому концепты под анимацию зачастую сильно упрощены и стилизованы.

Самым удобным для скелетной анимации является стиль «аппликации». Это означает, что наш объект сделан из практически однотонных элементов, без обводки и выраженных теней. В таком случае нам не нужно контролировать стыки элементов, так как однотонный сегмент можно делить на части без опасений за сохранность деталей.



Пример 1: стиль «аппликации»

Мы можем добавлять детализацию в данном стиле. Но делается это только на элементах, которые не подвержены сильным сгибам и анимируются деформацией с помощью меша. Например, на юбках героинь на картинке есть узоры, которые будут искажаться вместе с юбками. А руки и ноги выполнены единой заливкой, так как они будут сильно гнуться в суставах.



Рассмотрим еще один пример в данном стиле, но уже с чуть большей детализацией. Обратите внимание, что отдельные элементы прорисованы довольно детально. Например, шляпа, которая будет только подсакивать при ходьбе, слегка деформируясь. А значит, мы можем смело добавлять детали, так как нам не нужно ничего резать и сильно сгибать. У частей нет падающих теней или обводки. Падающие тени в анимации должны двигаться вместе с анимированным элементом. Это означает, что нам нужно постоянно контролировать дополнительный спрайт и изменять его в зависимости от изменения основного элемента. Но это полбеда, тень может меняться в зависимости от того, куда она падает, и рельефа в этом месте. А это значит, что нам нужно еще и исказить тень мешем. Это слишком трудозатратно. Даже если тени в анимации есть, они обычно статично следуют за анимированным объектом, что выглядит неестественно. Руки на сгибах все еще выполняются единой заливкой без детализации. Детализация сгибов суставов — это вообще главная проблема исходников для анимации, так как при любой дополнительной детализации в этих местах добавляется множество нюансов. Поэтому при подготовке и нарезке исходника, поступившего от художника, может потребоваться подчистить сгибы от лишних деталей.

Делать весь дизайн с помощью заливок, конечно, здорово для анимации, но тут особо не разгуляешься. Поэтому давайте разбираться, как добавлять детали и элементы стиля в концепт.

Обводка и тени

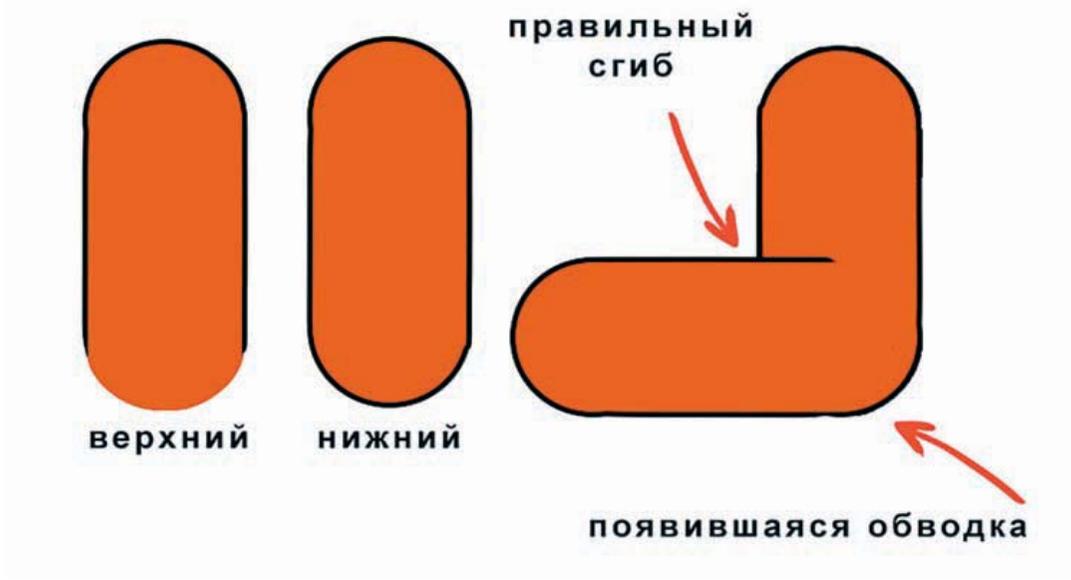
Чаще всего в скелетной анимации используются заливки с обводками. Иногда к этому сочетанию добавляются еще и жесткие тени на элементах. Опять же, на сгибах суставов в любом случае лучше обойтись без теней. Тени не стоят того, чтобы из-за них так сильно усложнять себе задачу при анимации. Принцип максимальной детализации на несгибаемых элементах также сохраняется.



Персонажи для анимации с обводкой на элементах общего концепта

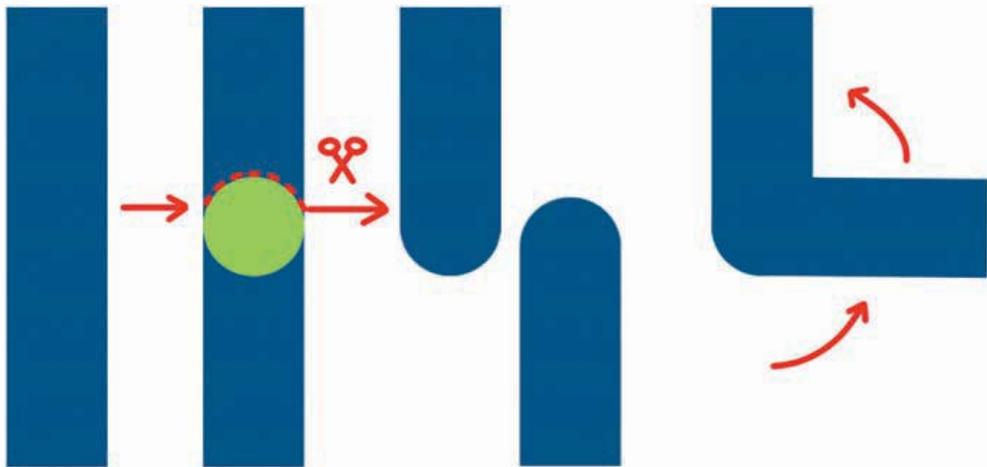
При добавлении обводки появляется нюанс при сгибе. Мы должны продолжить обводку на нижнем кусочке, так как при сгибе часть нижнего сегмента становится видимой, и обводка должна сохранить непрерывность и добавиться на этом кусочке. А на верхнем отрезанном сегменте обводка должна оборваться.

Таким образом, мы получаем непрерывную обводку на внешней стороне сгиба и правильное положение обводки на внутренней части сгиба. Такой прием отлично подходит для анимации конечностей и хорошо передает структуру сгиба на суставе. Как видите, уже появляются определенные сложности с теряющейся обводкой и обводкой, выступающей при повороте на внутренний сгиб. Если нам нужно, чтобы обводка была только по контуру, без внутреннего наложения на сгибе, то придется применять меш. Но в этом случае будет возникать проблема искажения самой обводки. Чтобы обводка не меняла толщину при сгибе, придется постараться.



Такой пример актуален не только для обводки, но и в целом для создания суставов. Давайте выясним, как единую конечность поделить на части. Причем поделить так, чтобы суставы сгибались наиболее удобно и практически без ограничений.

Рассмотрим это на примере единого полотна без обводки и детализации. Я уже говорила, что любая детализация усложняет задачу сгиба. На месте сустава нам нужно создать круг. Центр круга будет точкой вращения элементов конечности. Сам круг будет разделителем. Мы обрезаем одну часть конечности по верхней части этого круга, а другую по нижней. Таким образом, круг-сустав будет входить в состав обоих элементов. И теперь по точке центра круга мы можем совершать вращение под любым углом.



В реальной анимационной задаче конечности практически никогда не двигаются на все 360 градусов. Поэтому можно сосредоточиться только на требуемом движении. Но суть деления с помощью круга все равно соблюдается. Просто детали на конечности и неидеальная форма могут мешать крутить объект на полный оборот. Но в большинстве случаев это и не нужно.

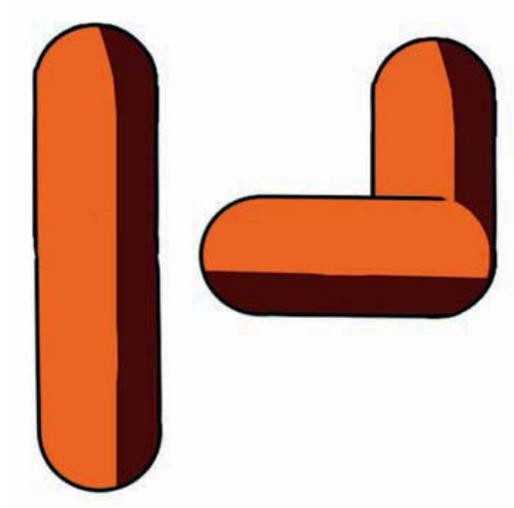
ТЕНИ

Рассмотрим проблему теней на сгибе. Как я уже сказала, вне сгибов можно добавлять практически любые детали. А вот сгибы разберем отдельно. Допустим, мы нарисовали тень в выпрямленном положении. Разрежем этот элемент на две части и согнем их под углом 90 градусов. Что мы видим? Тень не преобразовалась под новое положение объекта. На сгибе виден артефакт, который делает движение неестественным. В некоторых ситуациях этот артефакт некритичен, например, при небольших изгибах или допущениях в стиле проекта. Но целостность картинки в любом случае теряется.

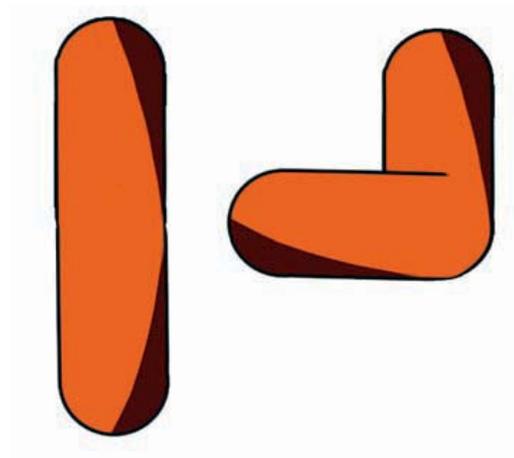
Решением этого вопроса может стать тень, не затрагивающая сгиб. В выпрямленном положении это будет выглядеть не стопроцентно натурально, но зато даст свободу в анимации элемента.

В итоге можно выделить основное правило концепта и нарезки под анимацию: в месте стыка делается максимальное упрощение, а вся детализация фокусируется в несгибаемых элементах.

Стоит сказать, что проблемное место стыка порой можно просто чем-то скрыть, например, объемным рукавом или перчаткой, высоким воротником, волосами. Тогда место стыка не будет привлекать много внимания и требовать такого тонкого подхода при анимации. Также скрыть



Проблема сгиба объекта с тенью



Вариант с чистым сгибом

стыки помогают сжатые пропорции персонажа. В таком случае элементы очень утрированы и накладываются друг на друга со значительным перекрытием. А значит, у нас появляется возможность упростить и анимацию. Иногда при сжатых пропорциях даже игнорируются определенные суставы, так как конечность слишком короткая, чтобы делить ее на три части. Она делится только на две — в районе локтя или колена. А иногда вообще остается единым элементом, как рука на приведенном примере. Она закрыта рукавом и настолько короткая, что ее можно лишь слегка деформировать мешем.

У такого подхода есть свои плюсы и минусы. Плюсы в том, что мы скроем точки сгиба, упростим себе задачу. Также упрощится и скелет, потому что тело будет сильно укорочено и количество костей максимально сократится. Персонаж в таком стиле будет выглядеть мило и отлично подойдет для Casual-игр. Но есть и минусы. Упрощение во всем не позволит нам сделать выраженные движения и сложную, интересную анимацию. Поэтому тут важен баланс. Стилизация — это хорошо, но не стоит превращать персонажа в мячик на ножках.

Для создания интересной анимации можно добавить объект, который будет эту анимацию подчеркивать. Например, коса или плащ будут развеиваться при беге, а широкие рукава подчеркнут движение рук. Такие элементы забирают на себя основное внимание, тем самым позволяя нам упростить само движение и сделать концепт движущегося элемента проще. А детализацию можно оставить как раз на вспомогательный элемент: узор на рукаве, ленты в косе, рваные края плаща. При деформации мешем эти детали не пострадают и привлекут внимание игрока.



Персонаж, выполненный в сжатых пропорциях

Прогноз анимации на этапе подготовки исходника

Уже на этапе нарезки исходника нам важно задуматься о предстоящей анимации. Необходимо определить, какие элементы и насколько сильно будут двигаться. Ведь именно от этого зависит, будем ли мы подробно вдаваться в нарезку этого элемента или нет. На этапе подготовки исходника необходимо составить список требуемых

анимаций. В каждой анимации могут быть свои движущиеся и статичные элементы, а нарезка у нас стандартная. И она должна подходить для всех требуемых анимаций. Нужно избегать ситуаций, когда мы не стали нарезать ноги персонажу, потому что для ходьбы достаточно меша, а потом выяснилось, что у героя должна быть анимация приседания, для которой одного меша мало.

Также мы должны подумать о материале каждого элемента. Чем пластичнее материал, тем больше вероятность того, что нарезка для этого элемента не подойдет и понадобится меш. А более жесткие элементы в деформации не нуждаются, и для анимации их точно нужно разрезать на части.

Само собой, предсказать всю анимацию мы не можем. И не исключены ситуации, когда уже в процессе анимации выясняется, что в исходнике чего-то не хватает. В таких случаях приходится возвращаться к началу и заполнять пробелы в нарезке. Но если у нас уже имеется набор анимаций и мы при этом станем менять исходник, то сделанные анимации могут начать работать неправильно. Это значит, что нам потребуется их скорректировать. Так как анимации опираются на исходник, то при его изменении они тоже изменятся. Поэтому, учитывая сложность исправления исходника на позднем этапе, лучше максимально продумать анимации на начальном. Например, можно записать все возможные варианты и оценить, как будут двигаться определенные элементы.

Проблемы, которые можно решить на этапе подготовки исходника

Этап между концептом и анимацией должен закрыть максимум проблем. Иначе эти недостатки потянутся в анимацию и усложнят задачу аниматору. Во-первых, нужно очистить концепт от лишних элементов. Как определить, лишний элемент или нет? Если объект или часть объекта никак не подчеркивают основное движение или характер персонажа, скорее всего, этот элемент можно убрать из концепта. Перегруженность не приветствуется в анимации. Лучше укрупнить важный элемент, удалив лишнее, чем нагрузить концепт деталями, которые ничего не привносят в анимацию.

Во-вторых, нужно очистить стыки от деталей. Если на сгибах присутствует орнамент, градиент, тень или любая другая дополнительная деталь, то лучше ее убрать, для того чтобы движение в этой области было свободным. Если детали важны, можно сместить их в области до или после сгиба.

Также можно удалить элементы, которые задают сложную иерархию слоев. Или переработать эти места так, чтобы минимизировать перекрытия. Иначе для сохранения порядка наложения элементов может понадобиться дополнительная нарезка.

Поскольку, используя скелетную анимацию, мы работаем в основном в одной плоскости, мы должны проверить правильность ракурса каждого элемента. Нужно попробовать покрутить и подвигать элементы в графическом редакторе, чтобы удостовериться в правильности их ракурса. Иногда бывает так, что в статике художник рисует более выгодный и естественный ракурс для некоторых элементов. Но при анимации мы не можем двигать элемент, потому что он не в нужном ракурсе. Распространенный пример — выставленная вперед нога при боковом ракурсе. Для того чтобы сделать бег, нам нужно будет эту ногу заменить на ногу в ракурсе сбоку. Можно сохранить оба положения, чтобы менять их в разных анимациях.



Большая часть успеха вашей анимации задается на этапе подготовки исходника. На исходнике базируются удобный скелет и настройка, а только потом уже подключается анимация. Поэтому уделите внимание этому этапу, чтобы не хвататься за голову во время анимации и не жалеть о зря потраченных часах из-за внезапно вскрывшихся проблем в фундаменте.

ГЛАВА 7

Нарезка концепта для анимации. Практические примеры

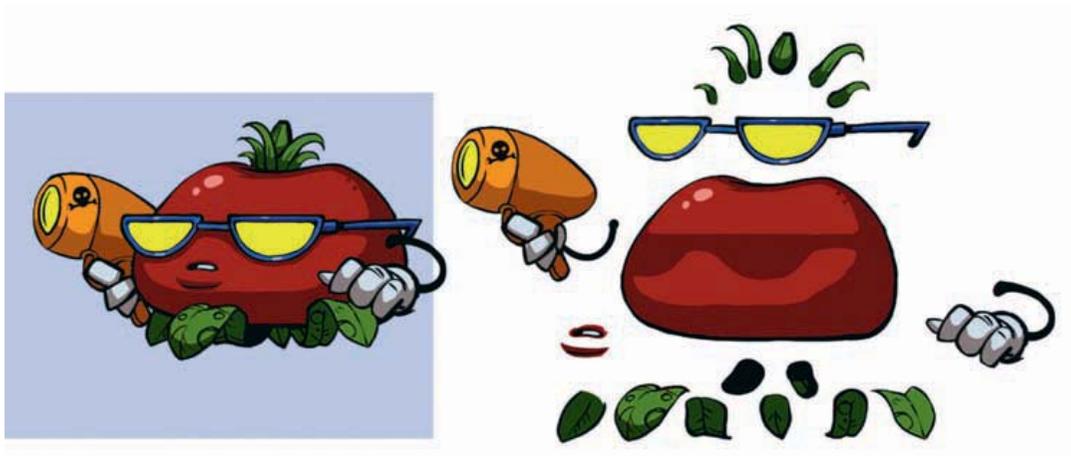
Нарезка простейшего персонажа

Чем меньше элементов на концепте, тем проще нарезка и анимация. Не всегда персонаж состоит из множества деталей. В большинстве игровых проектов, особенно в мобильном сегменте, используются максимально упрощенные персонажи и объекты. Для понимания принципа подготовки персонажа к анимации рассмотрим пару базовых примеров.

Обычно у простых персонажей движения тоже простые и их немного. **Первый пример — рыба.** Она может только плыть с различными вариациями: ускоряться, выпрыгивать из воды. Но в основе этого всего одно и то же движение. Когда рыба плывет, она двигает плавниками, хвостом и туловищем. А также она моргает своим большим глазом и меняет направление взгляда. Отрезать нам нужно только веки, зрачок и плавники. Анимировать хвост и туловище предпочтительнее мешем, так как эти элементы очень пластичны и лучше анимируются с помощью деформации. На каждый плавник также будут сделаны меши для дополнительной пластичности. Если в природе элемент «бескостный», «резиновый», то есть подвержен искажениям и произвольной деформации, меш для него обязателен в анимации, так как статичность будет сразу бросаться в глаза.



И рассмотрим еще один подобный пример, но уже с наземным существом. Это стрелок-помидор, который может передвигаться, прыгать и стрелять. В основе у него единое тело без суставов, которое мы будем искажать, как мячик, во время анимации. Основная сложность этого персонажа в ногах-листиках. Тут прослеживается аналогия с пауком или многоножкой. При ходьбе персонаж будет перебирать ножками и за счет этого двигаться. Поэтому каждый листик-ножку нам нужно отрезать друг от друга и от тела, а также поделить на части для изгиба внутри одного листика. Еще от тела нужно отделить очки, так как они двигаются относительно него: подсакиваются. Если мы их неотрежем, то получим эффект склеенности с основой. Само собой, отделим руки и верхние листики, которые у этого персонажа играют роль прически, а значит, и анимируются как копна волос на голове.



Персонаж для игры в жанре платформер

Перейдем к более сложным работам. Я покажу еще один пример из своего проекта. Нарезка персонажа сделана для игрового проекта *Funny Bunny Gun*. Это игра-платформер, метроидвания с механикой стрельбы по врагам, а значит, основными движениями персонажа будут бег, прыжок, стрельба. Также персонаж будет получать урон, следовательно, нам понадобится анимация на этот случай. Хотя анимацию получения урона можно создать при помощи минимального движения или только выражения лица, иногда даже просто окрашивания в красный цвет или моргания в прозрачности. Поэтому сосредоточимся на беге, прыжке и стрельбе.

При создании подобного концепта нужно сразу учесть возможные изменения персонажа и нарисовать для этого отдельные спрайты. Стрелять герой будет из разных пушек, а значит, и держать их будет по-разному. Нам понадобится несколько вариаций рук для разного типа оружия: одноручного, автоматов, тяжелых пушек. Кроме

того, герой не будет бегать по миру с постоянно каменным лицом. Реакция персонажа на происходящее вокруг добавит ему естественности и жизни. Нам понадобится создать несколько разных эмоций на лице, и мы также будем использовать для этого отдельные спрайты. Что-то можно сделать и с помощью мешей, например, добавить легкую улыбку или изменить положение век. Но для создания кардинально разных эмоций, например испуга и злости, вам наверняка потребуются разные спрайты. Спрайты эмоций могут включать не целое лицо, а набор изменяющихся частей лица. Из этих частей потом можно будет собрать несколько разных вариаций лица. Рисуем в первую очередь рот, глаза и брови. Нос на эмоции не влияет, так как практически не двигается. У некоторых персонажей на эмоции могут влиять волосы или уши. Например, при испуге волосы встанут дыбом. При грусти уши опустятся вниз, если персонаж — кот, собака или, как в нашем случае, заяц.

Итак, вот перед нами главный герой — заяц с крутыми пушками. Нам нужно спрогнозировать его движения и решить, какие части мы будем нарезать, а какие делать через меши деформацией. Начнем с отдельного от тела элемента — плаща. Он, очевидно, будет создан с помощью меша, поэтому его нужно отрезать отдельным куском. Важно не забыть нарисовать плащ целиком, так, чтобы он выходил за пределы тела, а пустые области при этом были не видны.



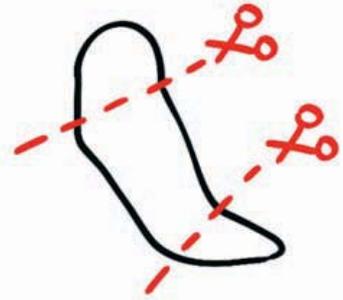
Изначальный концепт

После этого у нас остается довольно типичный человеческий скелет с туловищем, руками, ногами и головой. Специфика данного концепта в том, что, согласно игровой логике проекта, персонаж всегда будет держать пушку. Это накладывает определенные ограничения на движение. Нам не нужно сильно двигать руки и выкручивать на максимум суставы. У нас будет только анимация компенсации отдачи при стрельбе и анимация тяжести пушки при ходьбе и прыжке. В случае с одноручным оружием

движение рук будет чуть более свободным. Во время бега и прыжков руки будут двигаться стандартно, при этом постоянно находясь в полусогнутом положении. К чему я это все? К тому, что руки можно не нарезать, а оставить единым элементом. Движения небольшой амплитуды мы сделаем с помощью меша. Я бы только подумала над тем, чтобы отделить перчатки, так как это позволит показывать движение тонкой руки персонажа относительно толщины перчатки.

В остальном мы должны применить стандартную для человекоподобного скелета нарезку. Голова, ноги и руки отделены от туловища. Ноги разрезаем в суставах. Почему ноги разрезаем, а руки нет, спросите вы? Потому что ноги будут очень сильно гнуться при беге и прыжке. В этих анимациях ноги — основной элемент движения. А значит, нам нужно разрезать концепт и в области таза, и в коленях, и даже отделить ступни. Иногда внутри ступни еще отрезают и носочек, но для нашего зайца так делать необязательно. Ступня у персонажа очень маленькая и простая по дизайну, тут можно обойтись и деформацией. А вот если у героя длинный ботинок с деталями и в анимации идет упор на ступни — лучше разрезать ее на две части. А то и на три, если есть потребность анимировать еще и область с пальцами.

Отрезав голову от тела, разберемся с нарезкой внутри нее. Всегда стоит обдумывать, что у нас будет двигаться и как. Это позволит нарезать только нужные элементы. Ничего хорошего в излишней нарезке нет. Вам придется таскать за костями в анимации лишние привязанные спрайты, а это усложнит контроль над движением.



Очевидно, нам нужно отрезать уши от головы. Уши будут двигаться при любой анимации. Даже при анимации покоя уши будут слегка колыхаться на ветру (по аналогии с волосами) или пружинить при легком движении тела зайца. Сами уши мы будем дополнительно деформировать мешем. В данном случае из-за сложности рисунка уха нарезать его на части не получится, даже если мы сильно захотим. Так как у нас сложный рисунок внутри, при нарезке он рассыплется на кусочки. А значит, выбор в сторону деформации мешем очевиден. Ну и сама структура уха подразумевает деформацию и плавное движение по дуге, а не сгиб в какой-то отдельной части или суставе.

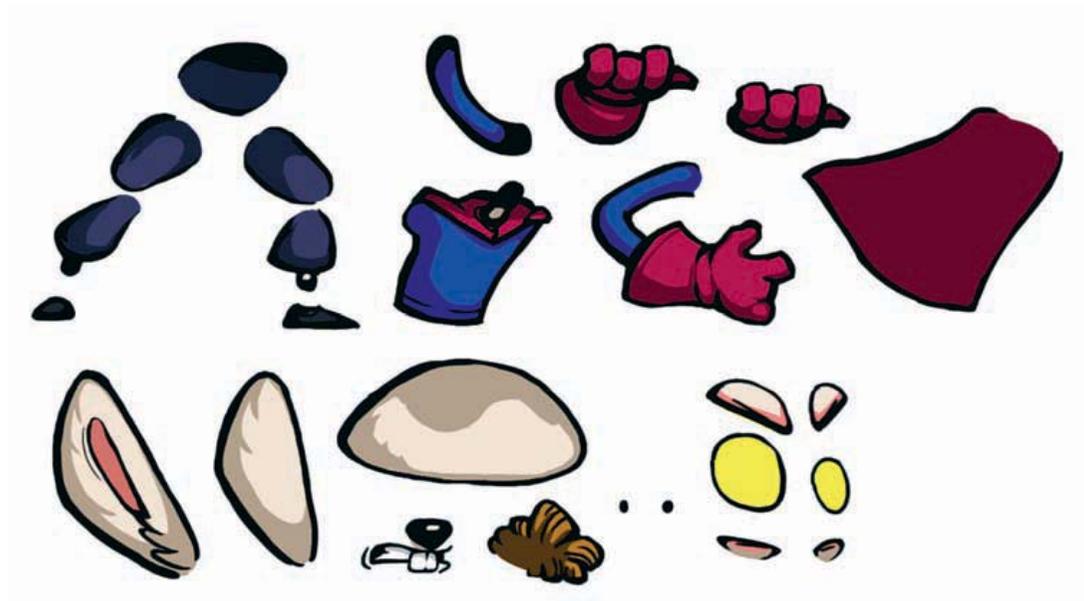
Также нам нужно отрезать глаза и рот, чтобы обеспечить возможность создания мимики персонажа. Голову представим в виде пустого овала, на который мы будем накладывать черты лица. На нарезку глаз обратим отдельное внимание. Отделим каждое веко, чтобы можно было добавлять в анимации моргание или движение век при создании эмоций. Зрачок также отделим для анимации направления взгляда.

Челку тоже нужно отделить от головы, так как она является пластичным элементом и будет двигаться при любом движении. К тому же, если ее не отделить, мы не сможем

анимировать дальний глаз. Челка перекрывает его, а значит, она должна быть в отдельном слое. Это еще один пунктик, на который нужно обращать внимание при нарезке, — иерархия слоев и перекрытия. Бывают случаи, когда мы должны отрезать статичный элемент только для того, чтобы сохранить наложение элементов друг на друга в анимации.



Мы отрезали от туловища все, что могли. Нужна ли нарезка самого торса? Иногда для задания подвижности позвоночника туловище нарезается еще на две части с разделением в районе живота под грудью. Это позволяет показать скручивание персонажа или наклон только верхней части туловища. Но зачастую нам не нужно сворачивать персонажа в клубок, а значит, можно обойтись мешем для задания легкой сутулости или наклона. В пользу нарезки также выступают длина персонажа и выраженность наклонов туловища. Если у нас персонаж выполнен в сжатых пропорциях, торс закрыт другими элементами и движений скручивания или сильного наклона не требуется — лучше обойтись мешем.



Итоговая нарезка зайца

Персонаж для игры в изометрии

Рассмотрим еще один пример нарезки персонажа, но для другого жанра со своей спецификой. В целом у гнома из этого примера такой же человеческий скелет, как и у зайца из предыдущего. Но пропорции персонажа намного более сжатые, концепт еще более упрощенный и ракурс отображения другой.



Пример игрового ракурса в изометрии

Обратите внимание на отдаленность игрового ракурса. Наш персонаж очень маленький, и основная его часть — это голова. В таком удалении анимация должна быть максимально утрированной, чтобы ее можно было прочесть в процессе игры. Особенно учитывая мультяшный стиль персонажа и окружения. Даже в состоянии покоя должно быть заметно, как персонаж дышит и переминается с ноги на ногу. В данном случае мы сконцентрируемся не на реалистичности и детализации нашей анимации, а на ее читаемости издалека. Это означает, что нам нужно нарезать только самые основные крупные элементы. Мелкие детали все равно будут не видны в таком ракурсе.



Итоговая нарезка гнома

Мы не будем нарезать волосы и бороду на отдельные пряди или отделять зрачки глаз и веки. В маленьком размере такая детализация будет избыточной. На голове мы выделим три основных блока, которые нужно отрезать: брови, уши и хвост. Уши и хвост мы отделим, потому что они выделяются на силуэте персонажа, а значит, довольно хорошо читаются на удалении от камеры. В анимации эти элементы будут заметны. Коричневые брови контрастируют с зеленым лицом, а значит, их тоже будет отлично видно издалека. Мы можем использовать для создания эмоций персонажа только брови, и в данном ракурсе этого будет достаточно.

Тело у гнома довольно условное. Пузо закрывает ноги практически полностью, из-под него торчат только кончики ботинок. Остальной части ног на концепте нет, а значит, их анимация будет состоять из одного элемента, который нам нужно отрезать. Руку персонажа мы отделим от туловища, но разрезать на составные части не станем. Если у героя будут сложные анимации с руками, их, скорее всего, придется рисовать в нескольких вариациях и ракурсах. Но для анимации покоя, бега и прыжка нам будет достаточно движения плеча и деформации руки в локте с помощью меша.

Нарезка интерфейса. Поп-ап

Анимация интерфейса в основном представляет собой анимацию появления или привлечения внимания к определенной части интерфейса. Раскрывающиеся свитки, выпадающие плашки, выпрыгивающие иконки, эффекты — все это помогает сделать интерфейс интереснее и привлечь внимание пользователя к нужным его зонам. Анимация — один из главных инструментов управления вниманием.

Анимация появления зачастую задает постепенное, поэтапное появление элементов цельного окна. Анимацией даже можно задать характер поп-апа. Появление окна выигрыша будет отличаться от появления окна поражения. Вряд ли иконки в шапке окна проигрыша будут прыгать и подсвечиваться, создавая атмосферу радости и веселья.

А вот вяло качающиеся флаги подойдут намного лучше. Поэтому не стоит думать, что раз у нас неживой персонаж, то анимация будет без характера и настроения. В анимации интерфейса также важно подчеркивать с помощью продуманных движений характер, цель и функции определенного окна.

Для того чтобы нарезать цельный поп-ап на части для анимации, нам нужно разобраться, какие элементы будут появляться по отдельности, какой будет последовательность их появления и какие движения мы можем создать для этого окна. Рассмотрим пример довольно стандартного окна победы в виде свитка.

Основным движением у нас будет раскрытие свитка. А значит, нам нужно отделить верхнюю скрученную часть от развернутой части. Также отделяем нижний подворот для создания bounce-эффекта и возможности отдельного контроля этой части. Свиток довольно пластичен, поэтому нам важно сделать меш для нижней части и слегка деформировать ее в процессе анимации. Иначе свиток будет казаться твердым и пластиковым.



Пример нарезки игрового поп-апа для анимации появления на экране

Верхняя часть свитка держится на стержне, на который накручен пергамент. Пергамент не приклеен к стержню, а значит, может двигаться относительно него при анимации. Отделяем боковые части от свитка, чтобы иметь возможность двигать эти элементы относительно друг друга. Обратите внимание, что в интерфейсе часто используется симметрия. Элементы повторяются, только отражаются по одной из осей. При нарезке подобных симметричных объектов нам не нужны одинаковые спрайты.

В исходнике должен быть только один, а его копии мы сделаем уже в процессе настройки скелета. Если у этих элементов и анимации будут одинаковые, то мы и их продублируем в Spine, не создавая с нуля. На нашем примере дублируются также канат, труба, лента снизу, колосок. Все эти части в исходнике представлены в единственном экземпляре.

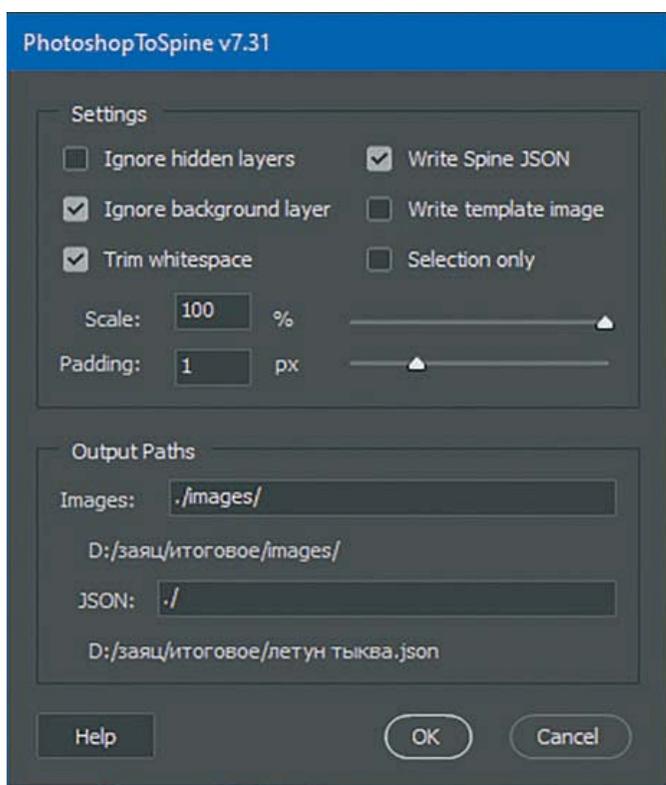
Анимация появления подразумевает наложение элементов друг на друга. В нашем случае подложка под надпись будет появляться на свитке. Сзади будут выскакивать корона, подсвеченная эффектом, и трубы. Снизу аналогично появится кнопка и из-за свитка выскочат две ленточки. Все эти элементы должны быть отдельными спрайтами.

Текст в шапке окна часто анимируется отдельно, например, появлением каждой буквы поверх уже развернувшейся подложки. Если предполагается такая анимация, каждая буква должна быть отделена.

ГЛАВА 8

Импорт исходника в программу Spine 2D

После того как мы нарезали все на сегменты, правильно назвали каждый слой, проверили расположение и иерархию слоев в psd, нужно перенести этот файл в Spine. Сначала необходимо убрать все группы слоев и слить со слоем все эффекты. В Spine переносятся только «чистые» слои, без групп и эффектов. Также нужно убрать смарт-объекты, маски и прикрепленные друг к другу слои. Не забудьте проверить исходник на артефакты перед экспортом. Это существенно упростит вам работу в будущем. Прозрачные пиксели вокруг контура элемента, мусорные пиксели вдали от основного объекта — все это лучше почистить самостоятельно на этапе подготовки исходника к анимации, а не оставлять на потом. Экспорт может произойти не так, как вам хотелось бы, если вы оставите артефакты на картинках.



Окно скриншота
PhotoshopToSpine

Перенести файл из Photoshop в Spine нам поможет скрипт Photoshop to Spine. Скачать его можно на github. Если просто вбить в строке поиска название скрипта,

первой же появится ссылка на скачивание на github. Нужный нам файл называется PhotoshopToSpine.jsx. Сохраните его в удобное место, чтобы не забыть расположение. Мы будем вызывать этот файл из программы Photoshop. Нажатие в меню пунктов «файл — сценарии — обзор» вызовет каталог папок системы, в котором нужно выбрать файл скрипта и открыть его двойным кликом мыши.

Скрипт запустится, только если ваш файл сохранен. Поэтому не забудьте сохранить проект Photoshop, в котором работаете. У вас высветится следующее окно, где можно задавать настройки экспорта.

Пункт Ignore hidden layers позволяет экспортировать только слои с активной видимостью. Это может потребоваться, если у вас в исходнике, например, сразу два персонажа. Скрыв слои одного, можно экспортировать второго. Или, наоборот, вам могут понадобиться и скрытые слои тоже, если в них хранятся варианты выражения лица или других элементов, а все слои не видны в файле Photoshop. Функция Ignore background layers не экспортирует закрепленный фон-подложку.

Trim whitespace позволяет кадрировать каждый фрагмент по его границам в прямоугольном формате. Если проставить галочку возле этого пункта, то при экспорте изображения разного размера будут соответствовать размеру сегмента. В противном случае программа создаст полноразмерное изображение, соответствующее размеру файла, а все пустое пространство будет прозрачным.

Write Spine JSON создает специальный файл экспорта для Spine 2D. В этот файл записывается вся информация по положению и наложению слоев. По сути, таким образом информация psd-файла переносится в Spine. Если не поставить галочку возле этого пункта, вы получите только последовательность png-изображений.

Write template image сохраняет цельную картинку, соответствующую виду вашего исходника в Photoshop. Ее нужно сохранять для проверки положения элементов во время работы, чтобы можно было посмотреть, не сдвинули ли вы что-то в исходнике при настройке в Spine.

Selection only позволяет экспортировать только выделенные слои в меню слоев в Photoshop. Это удобно, если нужно экспортировать только часть объекта, но с видимостью слоев возиться не хочется. Мы просто выделяем нужные слои и ставим эту галочку.

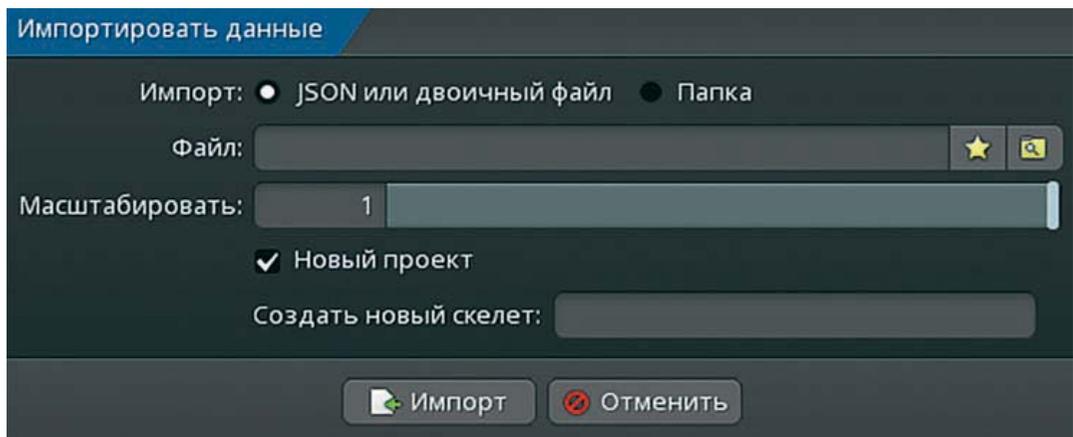
Scale используется для настройки масштаба всех экспортируемых изображений. Порой удобно работать в Photoshop с более крупными исходниками, чем те, что будут представлены в игре. А анимацию лучше делать на итоговом размере изображений. Мы можем уменьшить сам исходник в Photoshop, а можем настроить масштаб экспорта. Настройка при экспорте удобнее тем, что нам не нужно менять исходник. И по мере необходимости мы можем создавать новые файлы для Spine или других программ в ином масштабе.

Padding создает дополнительные пустые пиксели по краям экспортируемых изображений. Это делается для уменьшения артефактов на краях изображения. А также

полезно при запаковке в атласы в движке. Лучше, чтобы при создании единого полотна спрайтов между ними были расстояния. Про то, что такое атлас, мы узнаем в главе, посвященной переносу анимации в игровой движок.

Далее показаны два пути, куда можно сохранить наши файлы. Обратите внимание, что нет выбора каталога папок. Путь определяется относительно расположения файла Photoshop. Поэтому сохранять файл Photoshop нужно туда, где вы хотите видеть исходники для Spine. Картинки лучше сохранять в дополнительную папку, которая по умолчанию уже задана как images. Можете придумать свое название.

Для того чтобы открыть исходник в программе Spine 2D, нужно создать новый пустой проект. Затем в основном меню программы, которое вызывается нажатием на логотип в верхнем левом углу экрана, нужно выбрать строчку «Импортировать данные».



Выбираем экспорт JSON. Это как раз тот файл, что мы создали с помощью скрипта в Photoshop. Выбираем его в строке «Файл». Можем еще раз масштабировать наш исходник в строке «Масштабировать». Отмечаем галочкой пункт «Новый проект», который вместе с импортом данных создаст под них новый проект. Если галочку не поставить, эта функция импортирует данные в существующий проект. Таким образом мы можем импортировать несколько файлов JSON и соответствующих им изображений в один проект Spine.

Создание нового скелета с заданным именем добавит его в панель справа в программе Spine. Теоретически мы можем для каждого новых данных создавать свой скелет и свой набор папок и пунктов в Spine. Но на практике эта возможность практически не используется.

ГЛАВА 9

Возможности Spine 2D. Углубляемся в функции и инструменты программы

Базовый набор функций программы мы уже рассмотрели в этой книге. И даже закрепили эти навыки на практике. Этим возможностей уже достаточно для выполнения множества задач. Например, для анимации простых объектов или даже интерфейса без большого количества дополнительных деталей. Но стоит обратиться к примеру чуть посложнее, и вы обязательно упретесь в стену, так как в простых примерах мы не затрагивали многие функции. Перемещать, масштабировать, искажать объекты — это еще далеко не все, что требуется в анимации. На следующем этапе мы рассмотрим более сложные примеры и разберем новые инструменты программы Spine 2D.

Маска

Иногда нам нужно ограничить видимость определенного элемента. Или привязать один элемент к форме другого, чтобы объект не мог выйти за пределы этой формы. Возможно, вы уже сталкивались с масками в графических редакторах, например в Photoshop, когда вам нужно было привязать один слой к другому и рисовать только в его пределах. В Spine маска работает аналогичным образом. Мы создаем форму, которая ограничивает видимость определенных изображений и анимаций. Маску не стоит использовать без необходимости, так как это является неоптимизированным решением. Особенно если точек создания маски получается много. Чем сложнее форма объекта, тем больше точек требуется для создания маски (аналогично созданию меша). И таких решений нужно избегать, если без маски можно обойтись. Четыре точки — оптимальное значение для маски. Шесть — только для особых случаев. Больше — уже считается неоптимизированным решением. Но для создания многих эффектов и приемов маска все же используется. В частности, если мы делаем анимацию не для игры, то оптимизация неважна, а значит, мы можем создавать маску с любым количеством точек, лишь бы компьютер выдержал. А вот для игр, особенно для мобильных, вопрос оптимизации стоит довольно остро. Поэтому каждый вклад в затраты ресурсов для создания игры нужно разбирать подробно.

Давайте рассмотрим, как создать маску и работать с ней. Также разберем примеры, для которых маска будет наиболее подходящим решением.

ИСПОЛЬЗОВАНИЕ МАСКИ. АНИМАЦИЯ МЕРЦАНИЯ ИКОНКИ

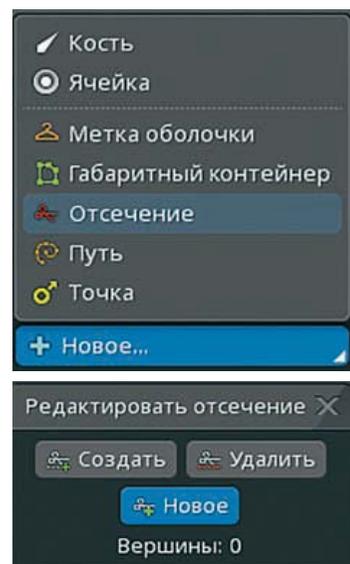
Для начала рассмотрим простой пример, который очень часто используется в анимации интерфейса и иконок. Для привлечения внимания к определенной плашке или элементу создается эффект мерцания. Светлая полоса проходит по объекту, придавая ему блеск. Часто этот прием используется в упрощенном виде, а форма объекта игнорируется. То есть по всему силуэту мы пускаем светлую полосу. Если учитывать форму элемента, то пускать полосу нужно по отдельным граням. В рамках нашей книги мы будем делать анимацию мерцания кристалла в упрощенном виде, не учитывая его форму. В большинстве случаев этого достаточно для интерфейса.



Итак, изначально у нас есть два изображения. Сам кристалл и белая полоска-блик. Полоска должна быть больше кристалла, выходить за его пределы, чтобы полностью покрывать его силуэт. Настройка скелета будет очень простой. Нужно создать две кости: одну — для кристалла, по его центру, и одну — для полоски.

Нам нужно ограничить видимость полоски формой кристалла. Создадим маску по силуэту. Итак, выбираем кость кристалла для создания маски. И в панели «Новое» выбираем пункт «Отсечение». Появится окно редактирования маски.

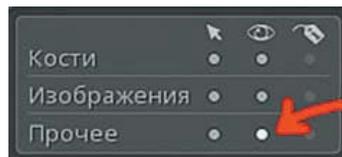
С помощью нажатия на кнопку «Новое» мы можем вручную, точка за точкой, создать маску. Форма маски не корректируется кривыми Безье в Spine. А значит, создание плавных форм сильно затруднено и требует множества точек. Как я говорила выше, большое количество точек не приветствуется из соображений оптимизации анимации. Поэтому маску в Spine в основном применяют для объектов рубленой формы, без изгибов. Кристалл у нас в форме ромба, поэтому маска для него делается просто. Кликая по вершинам ромба, мы



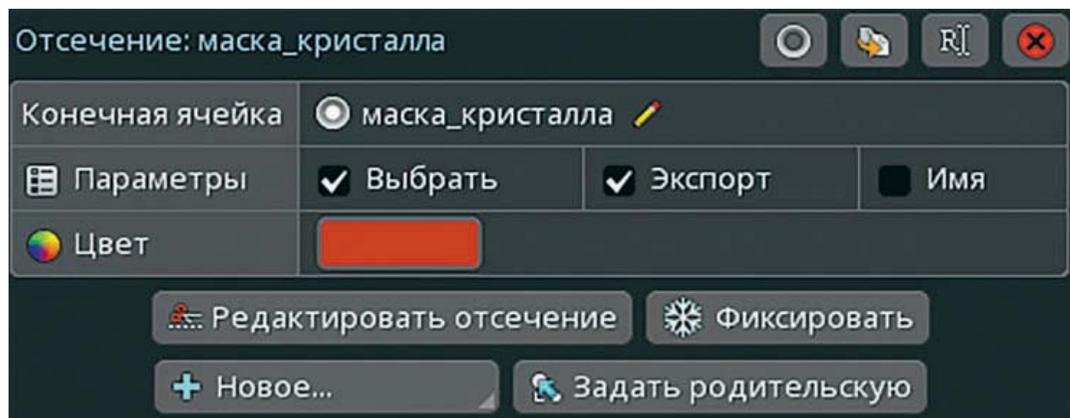
создаем контур по силуэту и замыкаем его повторным щелчком по начальной точке. Маска отображается красной линией.

В окне редактирования также есть кнопки «Создать» и «Удалить», которые позволяют добавлять новые точки и удалять их. Также мы можем перемещать точку, кликнув и потянув за нее в нужном направлении. Таким образом мы можем редактировать форму маски.

После создания маски весь экран становится красноватым. Это может мешать в работе. Если вы хотите скрыть этот эффект, нужно отключить в столбце видимости элементов нижний флажок «Прочее».



У созданной маски, как и у всех элементов, есть свое окно настроек под панелью структуры проекта. Выделяем маску и тем самым открываем это окно.

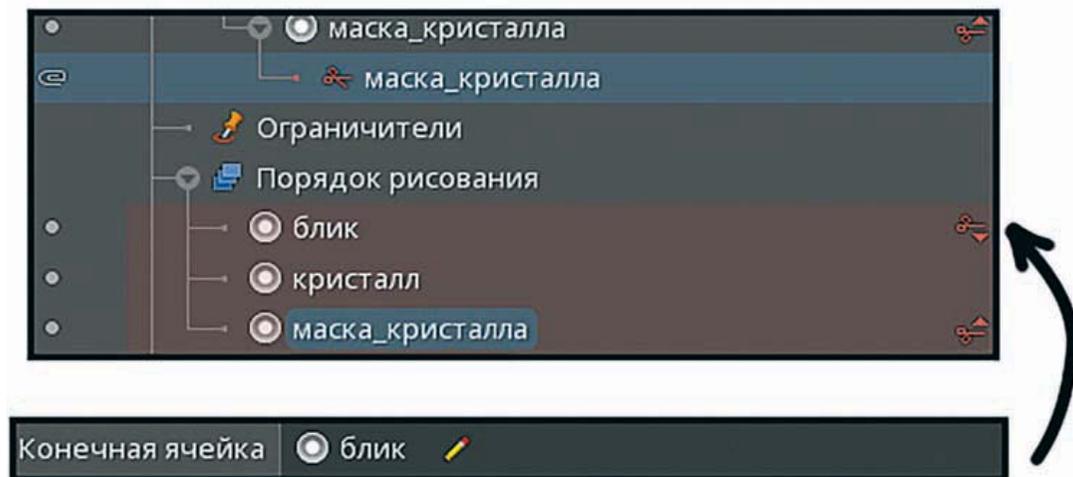


Первая строка **«Конечная ячейка»** определяет влияние маски по слоям отрисовки. Нужно открыть в структуре проекта выпадающее меню «Порядок рисования», в котором у нас поочередно располагаются слоты с изображениями. Там мы увидим слот с маской. В этой ячейке нет изображения, только наша маска. Маска влияет на то, что находится выше этого слота. И мы должны ограничить это влияние, чтобы маска работала только на нужные нам слои. Конечная ячейка определяет, на чем закончится влияние маски. Учитывайте, что на один слот может влиять только одна маска. Мы не можем сделать несколько масок и задать им влияние на одни и те же слоты. Для корректной работы диапазоны влияния масок не должны пересекаться. В нашем случае маска должна влиять на кристалл и блик. Значит, конечной ячейкой нужно поставить блик, так как он выше кристалла.

Проверим действие маски, попробуем подвигать блик относительно кристалла и убедимся, что теперь блик не выходит за его силуэт.

В меню выбора цвета в окне настроек маски можно выбрать цвет, который будет использоваться для окрашивания пространства вне маски. Также с помощью этого

меню можно вообще убрать окрашивание, понизив прозрачность цвета в нуль. Это полезно в том случае, если у вас в проекте присутствует много элементов, не относящихся к костям и изображениям. Таким образом, отключив видимость флажком «Прочее», мы убираем все элементы, кроме костей и картинок. А если нам нужно отключить только окрашивание маски, то делаем это через выбор цвета и понижение прозрачности.



Кнопка «Фиксировать» задаст новому положению маски значение по умолчанию. Изначально, когда мы создали маску, ей были заданы значения масштаба, поворота и перемещения. Вы можете видеть их в соответствующих полях рядом с этими инструментами. Если мы исказим маску каким-либо образом, то эти значения изменятся. Но если мы нажмем кнопку «Фиксировать», то заданное положение маски сохранится, а значения станут теми, что были изначально. То есть мы как бы обнулим значения, сохранив новое положение, масштаб и поворот. Теперь это будет положением и формой маски по умолчанию.

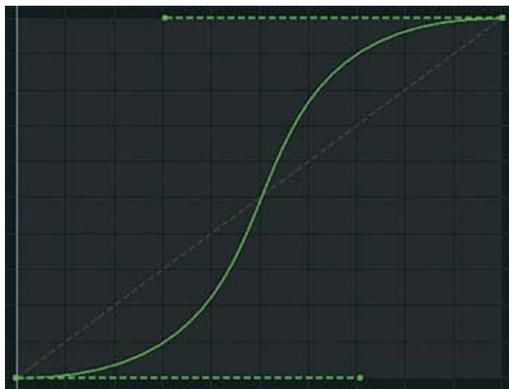
Кнопка «Задать родительскую» позволяет перекинуть маску в новый слот. Эта кнопка является общей для многих элементов и действует аналогично.

АНИМАЦИЯ БЛИКА

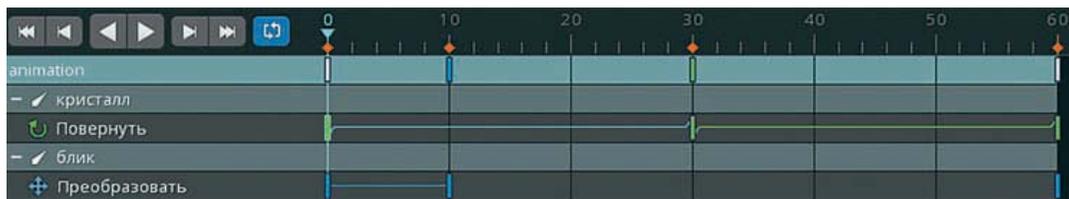
Теперь сделаем анимацию блика. Выбираем кость блика. Перетаскиваем ее влево, чуть заступая за границу кристалла, чтобы в анимации блеск исходил из-за границ маски. Фиксируем положение кости в нулевом кадре. В десятом кадре (или в другом, на ваше усмотрение) мы должны переместить полосу вправо за границу кристалла. Перемещаем строго по оси X. Или можно сделать это, просто работая с цифровыми значениями в поле «Преобразовать» кости блика. Итак, у нас с 0-го по 10-й кадр блик движется по оси X из-за левой границы кристалла за правую.

Если оставить анимацию так, то блеск будет постоянным. Чтобы это исправить, нужно сделать задержку между проходами блика. Для этого копируем 10-й кадр в 60-й. Таким образом, с 10-го по 60-й кадр анимации кристалл блеснуть не будет. Тут можно экспериментировать. Например, сделать два блика подряд, потом задержку. Или два блика, потом задержку, затем один блик, а далее снова задержку.

Давайте добавим кристаллу динамики, чтобы анимация была готова к вставке в интерфейс. Нам нужно сделать анимацию поворота. Выделяем кость кристалла. В 0-м кадре поворачиваем кристалл на 5 градусов влево, а в 30-м на 5 градусов вправо. А 0-й кадр копируем в 60-й. Добавляем выраженное замедление в начале и конце для поворота кристалла. Настраиваем график для 0-го и 30-го кадров следующим образом.



Анимация кристалла для интерфейса готова. Такой эффект можно применять для любых объектов. Главное — помнить про оптимизацию, если вы делаете анимацию для игры.



Ключи анимации кристалла

Габаритный контейнер

В движке нам часто нужно обрабатывать определенные события. Например, удар персонажа должен производить определенный эффект, когда кулак или оружие соприкасаются с врагом или объектом. Нам нужно определить столкновение кулака с врагом. Это делается следующим образом. У кулака и врага есть специально настроенные области — коллайдеры. Они и определяют столкновение, когда пересекаются или соприкасаются гранями. Так, например, задается взаимодействие персонажа с землей. У персонажа есть коллайдер, который своей нижней гранью не может проходить сквозь коллайдер земли. В движке коллайдеры задаются специальным инструментом и прикрепляются к элементу. Не будем углубляться дальше, главное — понимать логику работы коллайдеров.

Но в анимации положение кулака будет постоянно меняться, поэтому задать область влияния кулака в движке нельзя. Мы ведь не будем для каждого кадра анимации рисовать новую область в движке. А вот Spine может нам в этом помочь. Для этого есть инструмент «габаритный контейнер».

Допустим, у нас уже есть анимация удара мечом для персонажа. Чтобы работать с этим ударом в движке, нам понадобится коллайдер для меча. С его помощью мы будем обрабатывать удар в игре. Нам нужно выбрать кость меча и в панели «Новое» кликнуть на «габаритный контейнер». Окно создания контейнера аналогично окну создания маски. Теперь поочередно точками задаем форму. Можем добавлять, удалять и перемещать существующие точки. Созданная форма подсветится цветом и будет сопровождать объект на протяжении всей анимации. Не обязательно создавать контейнер очень близко к форме объекта. Обычно столкновение фиксируется не пиксель в пиксель, а примерно. И какие-то вогнутые формы на мече, к примеру, абсолютно не влияют на работу с его атакой, так как она все равно распознается по лезвию в целом. Но бывают случаи, требующие и более точной настройки.



Пример габаритного контейнера на мече персонажа

Каких-то специфических настроек у контейнера нет. Выделив его, мы можем менять цвет подсветки, редактировать контейнер. И, как и в случае с маской, у контейнера есть кнопка «Фиксировать».

Этот элемент удобен тем, что мы можем включать и выключать его в процессе анимации. В нашем случае коллайдер удара должен быть активен только в момент атаки,

а не в состоянии покоя и не в момент замаха. А значит, мы должны включить контейнер приблизительно после замаха при прохождении над головой и оставить включенным до удара о землю. В остальное время контейнер должен быть выключен. Включить и выключить контейнер можно нажатием на иконку скрепки в начале строки структуры проекта габаритного контейнера.



Также в Spine есть полезная фишка — возможность привязывать точки маски и точки габаритного контейнера к костям. Это делается аналогично работе с точками меша и с помощью того же алгоритма и инструмента «Весы». Разница лишь в том, что автоматическая настройка весов не работает, все придется делать вручную. Но учитывая, что точек в маске и контейнере обычно меньше, чем в меше, ручная настройка не так страшна. Зато мы можем исказить маску или контейнер в анимации по аналогии с мешем.

ТОЧКА

В Spine есть еще один инструмент для правильной и удобной интеграции анимации в движок. Он называется «Точка» и позволяет зафиксировать определенную позицию, которую мы потом сможем отслеживать в движке. Находится «Точка» все там же, в панели «Новое».

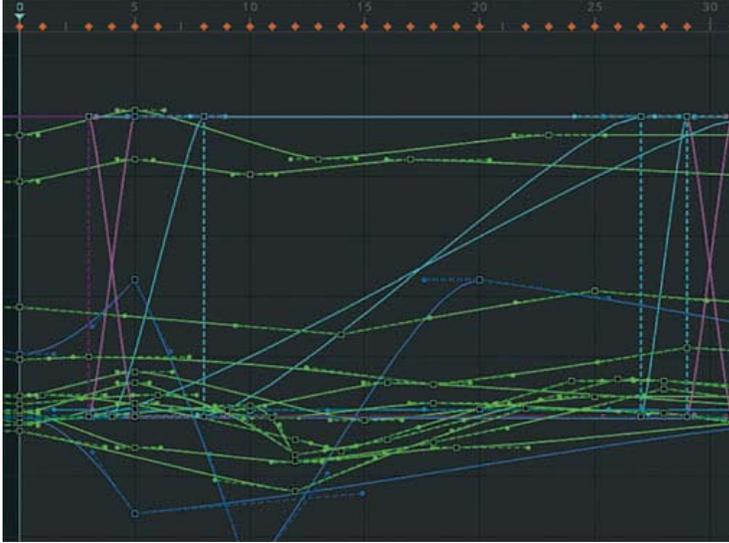
Например, у персонажа есть оружие и из дула этого оружия вылетают пули. Опять же, эта позиция в анимации изменяется, и отследить ее нужно в первую очередь именно в анимации. Плюс к тому у нас может быть множество разного оружия и место вылета пуль из каждого будет свое. А значит, нам нужно назначить эту точку для пуль в анимации для каждого оружия.

График характера движения. Целостный формат

В четвертой версии программы Spine появился новый тип отображения графика движения. Ранее мы рассматривали график для каждого ключа и промежутка между двумя ключами. Но для длинной комплексной анимации порой важно оценивать картину в целом, видеть общий характер движения и все графики сразу. Эту возможность

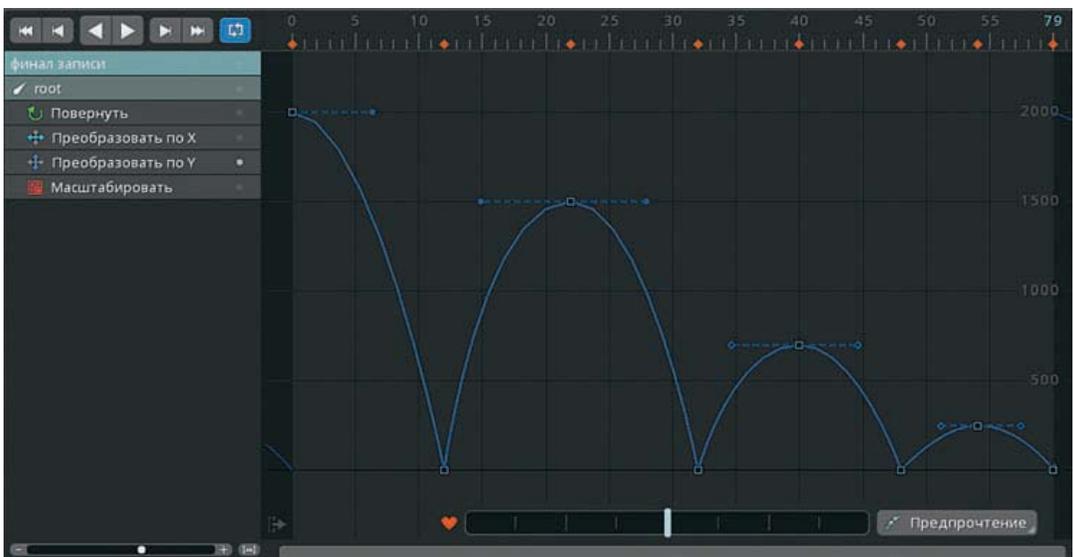
и дает нам целостный формат графика. Он содержит в себе все отдельные графики, которые мы рассматривали ранее.

Находится окно графика в выпадающем меню «Вид» и так и называется — график.



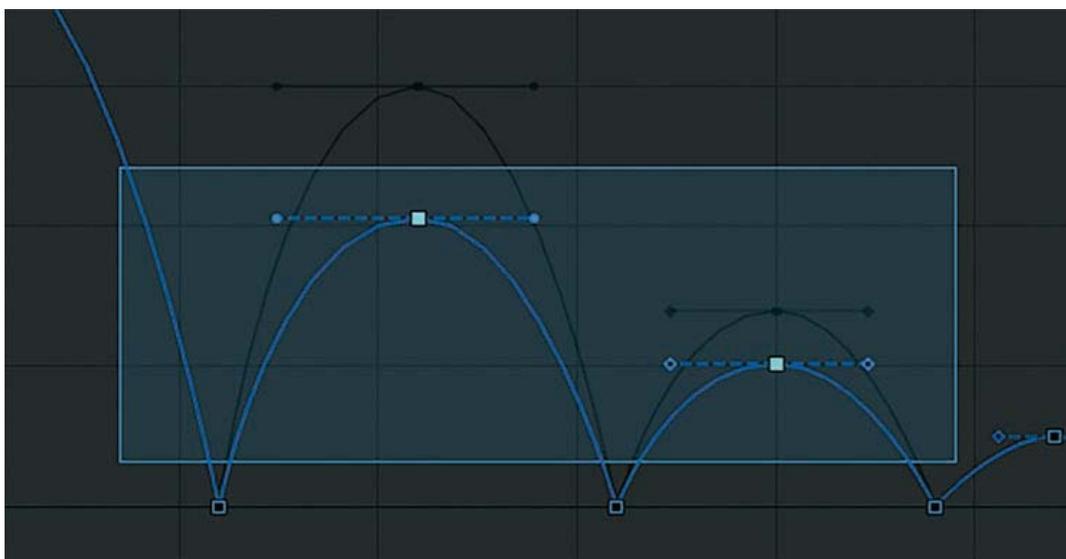
Пример множества целостных графиков сложной анимации

Есть определенная сложность в работе с таким большим количеством информации. Может быть трудно вычленив нужную кривую из общей паутины. Но для того чтобы корректировать общий характер анимации, это очень полезная функция Spine. Мы можем корректировать не только отдельные точки, но и целые области, тем самым добиваясь нужного движения.



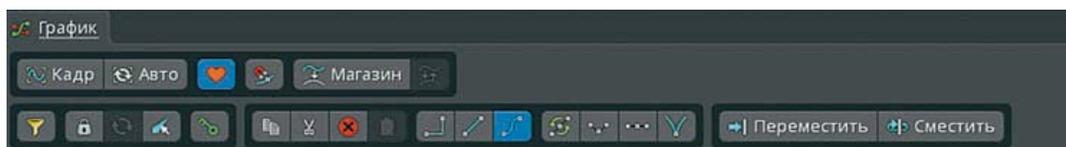
Откроем сделанное упражнение с прыгающим в сторону мячиком. На этом примере наглядно видно, как форма общего графика показывает характер движения в целом. Чтобы увидеть определенный график, а не огромное их количество, как на примере выше, в правой части окна графика необходимо выбрать строку. Для этого нужно выбрать кость и нажать на кругляшок рядом с названием инструмента. Далее выбираем строку «Преобразовать по Y». Получаем следующий график. Обратите внимание, что график наглядно показывает все движение и по его форме можно понять общий характер.

Корректируя опорные точки, мы можем менять сразу два прилегающих к ним сегмента графика. А можем двигать только один график, но это мы рассмотрим ниже, когда будем подробно разбирать интерфейс окна. Также мы можем выбрать рамкой несколько точек и контролировать сразу целые области графика. Причем будет видно и исходное положение графика, и новое, что позволит оценить масштаб изменений. Поскольку мы видим весь график, то можем оценить и разницу между сегментами. В данном случае мы видим, как происходит затухание в каждом прыжке. И мы можем контролировать степень этого затухания, меняя амплитуду волн. Также мы можем легко сделать из прямых линий кривые (добавить замедления в начале и конце) на всех точках сразу, вместо того чтобы выбирать каждый ключ по очереди.



Теперь подробнее рассмотрим **интерфейс окна работы с графиком**. Как я уже упомянула, слева находится список всех задействованных к выбранной кости инструментов.

Сверху графика находится основная панель инструментов.



Кнопка **«Кадр»** влияет на отображение графика в окне. При нажатии на нее график будет соответствовать размеру окна. То есть в окне мы будем видеть весь график сразу, если у нас ничего не выделено. Если выделить часть графика и нажать «Кадр», то на все окно распространится выделенный сегмент.

Кнопка **«Авто»** помогает при переключении с графика на график. Если ее нажать, то при каждом переключении новый график будет разворачиваться на весь кадр. Если вы часто переключаетесь между графиками в меню слева от основной области, держите эту кнопку включенной.

Далее мы видим иконку **сердечка**. За такой маленькой иконкой кроется довольно важный инструмент. Нажав на кнопку в нижней части окна, вы увидите бегунок и еще одну кнопку. По умолчанию кнопка стоит в значении «предпочтение». В основном используется именно это значение, но нажав на кнопку, вы можете выбрать нужный вам тип смещения.

На шкале времени нужно выбрать промежуточное значение между ключами. В окне графика появится соответствующая этому кадру голубая линия. Двигая бегунок, вы увидите, как точки графика, который пересекается с голубой линией (а значит, соответствует выбранному кадру), перемещаются. В случае с выбранным значением «предпочтение» точка будет смещаться к значению левого или правого ключа, в зависимости от того, в какую сторону вы двигаете бегунок в нижней части окна.

С помощью иконки **магнитика** включается привязка к значениям графика. Если мы будем двигать точку с включенным магнитом, то при приближении к другим значениям на графике она будет к ним примагничиваться.

Кнопка **«Магазин»** позволяет сохранить значения графика. В английском варианте это кнопка «Store», что означает хранить. В данном случае имеет место неточный перевод. Нажав на эту кнопку, мы зафиксируем значения точек графика. Изменив значения, мы будем видеть и старые, и новые (при выделенных точках). Нажимая на иконку справа от кнопки «Магазин», мы можем переключаться между старым и новым графиками для оценки изменений. Мы можем даже запустить созданную анимацию и менять график с включенным воспроизведением анимации. Тогда мы наглядно увидим, как изменение графика влияет на анимацию, и сможем сравнить старый и новый варианты.

Перейдем к следующей строчке инструментов. По направлению слева направо располагаются инструменты «фильтр», «блокировка», «выбор кости». «Фильтр» позволяет отображать только определенные элементы, чтобы не замусоривать окно графика множественными кривыми. Инструмент блокировки фиксирует график для выбранной кости, и даже если вы выберете другую кость, это никак не повлияет на окно графика. Если замочек не нажат, то отображается график кости, на которую мы кликнули в структуре проекта или на рабочей области. Рядом с замочком находится инструмент «обновить». Эта кнопка перезакрепляет график на новую выбранную кость. Эти два инструмента полезны тем, что мы можем видеть определенный график и работать с костями, а также в любой момент имеем возможность, выбрав новую кость,

закрепить уже ее график. Нажатие на иконку кости позволяет выбрать кость, график которой отображается в окне. А нажатие на иконку ключа — зафиксировать новый ключ в выбранном кадре в окне графика.

Правее располагается панель с иконками стандартных инструментов «вырезать», «копировать», «вставить», а также уже знакомых нам видов графиков. Рассмотрим следующую панель.



«1» — Восстанавливает значения по умолчанию для точки кривой. Если выбрана точка и положение усиков изменено, нажав на эту кнопку, мы вернем стандартное замедление в точке в виде изгиба. Если выбран усик, то при нажатии этой кнопки образуется прямая линия.

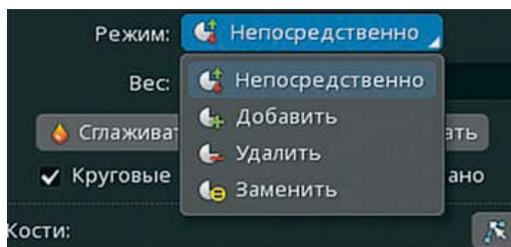
«2» — Позволяет корректировать сегменты графика по отдельности. Настраивая точку кривой, мы по умолчанию меняем график и слева и справа от нее. Если нажать кнопку «2», то мы сможем корректировать только одну сторону графика относительно редактируемой точки.

«3» — Выставляет усики редактируемой точки кривой на прямую линию параллельно оси X.

«4» — Создает резкий скачок графика в выбранной точке. На графике образуется острый угол.

Меш. Ручная настройка весов

Для простых примеров достаточно автоматического распределения весов костей на меше. Иногда можно слегка корректировать их вручную. Но для создания сложных мешей, например для эффекта псевдо-3D, необходимо работать с влиянием костей самостоятельно, не полагаясь на программу. Само собой, настраивать каждую точку неудобно, нужно работать сразу с областями и несколькими точками. Для этого в Spine есть специальный набор инструментов. Нажав на кнопку «Непосредственно» в меню «Режим», мы увидим выпадающий список с возможными вариантами работы с весами.



При выборе каждого из пунктов списка под кнопкой появятся настройки, относящиеся к этому типу работы с весами. Базовый режим «Непосредственно» мы уже рассматривали в упражнениях с мячиком. Перейдем к режиму «Добавить».

В этом режиме на рабочей области у нас появляются две окружности. Внутренняя окружность определяет размер кисти, которая будет задавать вес. Область от внутренней до внешней окружности является градиентом, от полного влияния до нуля. Это что-то вроде мягкой кисти в растровом редакторе с возможностью задавать область мягкости. Для этого инструмента у нас есть три бегунка настроек: «интенсивность», «размер» и «перо». «Интенсивность» показывает степень максимального влияния, которое мы задаем кистью. «Размер» задает диаметр внешней окружности. «Перо» задает диаметр внутренней окружности. Водя по мешу кистью, настроенной определенным образом, мы добавляем вес выбранной кости точкам, которые попадают под эту кисть. Можно даже выбрать несколько костей в списке и добавлять влияние сразу всех выбранных костей на точки меша. В случае работы с несколькими костями сразу заданное значение интенсивности делится поровну на количество костей. То есть, если мы зададим значение 30, то, выбрав одну кость, мы будем добавлять вес 30% на точки, попадающие под кисть. Если в списке выбрано две кости, то значение интенсивности разделится пополам и кость будет добавлять по 15% на точку. А если выберем три кости сразу, оставив значение интенсивности 30%, то кисть будет задавать вес по 10% каждой кости на точки под кистью.

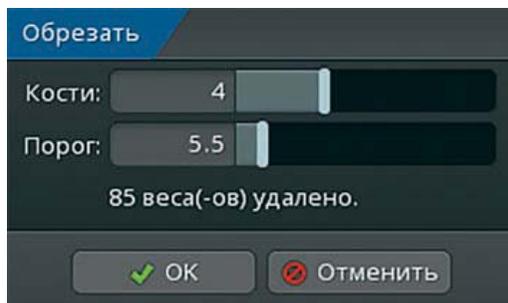
Полезная функция такого режима — возможность работать только с выбранной областью или точками. Прежде чем красить кистью весов, мы можем выбрать одну или несколько точек. Также можно выделить область с точками рамкой. Инструменты будут работать только с выбранной областью, не внося изменений в остальные точки. Если у нас включен режим «Непосредственно», то изменения будут производиться сразу для всех выбранных точек.

Пункт «Удалить» в списке является обратной операцией для добавления. Настройки кисти те же самые. Но теперь мы отнимаем заданное значение интенсивности у точек. То есть если у нас было влияние кости на точку 60% и мы задали значение интенсивности 20%, то при каждом щелчке кистью мы будем отнимать по 20% влияния кости.

Пункт «Заменить» позволяет устанавливать вес, как добавив, так и убавив значение на определенный показатель. Допустим, мы поставили значение интенсивности 20%. Опустим работу с плавностью и будем считать, что кисть у нас жесткая. Поставим значение «перо», равное нулю. Тогда на всей области кисти будет установлено влияние выбранной кости в 20%, несмотря на то, какой вес был на точках под кистью у этой кости до этого.

Далее рассмотрим настройки бегунков параметров, расположенных ниже. Кнопку «Сглаживать» мы уже разобрали на предыдущих примерах. Кнопка «Авто» позволяет в любой момент времени автоматически рассчитать настройки весов из значений для выбранных точек и костей, то есть только для части меша. При первом запуске для всех точек и костей устанавливается как раз эта автоматическая настройка. А мы можем выбрать только нужные и пересчитать их при надобности.

Кнопка «Обрезать» нужна для чистки точек от весов с минимальными значениями. В процессе настройки, особенно с помощью кисти, могут появиться небольшие значения весов в несколько процентов, которые не нужны и не влияют на общую картину движения. Убирать такие мелочи вручную было бы муторно и неудобно. При нажатии на эту кнопку появляется дополнительная панель настроек.



С помощью этой панели можно указать максимальное количество костей, влияющих на точку меша. Если костей больше, то кости с наименьшим влиянием будут отрезаны и потеряют влияние на эту точку. Например, у нас на точку влияет 6 костей, а мы поставим значение 4 в настройках обрезки. Значения влияния: 30, 20, 20, 15, 10, 5. Кости с влиянием 10 и 5 будут отрезаны, а их влияние равномерно распределится по остальным костям. Если влияние костей одинаковое, будет отрезана первая в списке кость.

Бегунок «Порог» отсекает все кости с влиянием меньше заданного значения.

И ниже располагаются три флажка, которые влияют на визуальное отображение точек меша и весов.



Флажок «Круговые» добавляет кружки на точки с сегментами, соответствующими весам костей. Флажок «Наложение» окрашивает весь меш в градиенты цветов весов костей. Если отключить и то и другое, то отображаться будут только голубые точки. Галочка, проставленная рядом с пунктом «Выбрано», окрасит весь меш в черный и оставит цветными только выбранные кости.

Ручная настройка меша в большинстве случаев происходит во вкладке анимации, так как мы не можем менять положение костей во вкладке настройки скелета каждый раз, когда нам нужно определенным образом настроить меш. Во вкладке анимации мы можем в определенном кадре обозначить требуемое положение меша и подогнать веса так, чтобы добиться нужной формы. На общую настройку костей это не влияет, а меш будет обладать нужными нам весами в точках.

Обратите внимание, что при окрашивании точек меша в цвета весов положение точек меняться не должно. Если положение точек меняется, это означает, что в настройке скелета вы подвинули кости, привязанные к мешу. Лучше всего в таком случае восстановить верное положение костей. Но если такой возможности нет, то нужно нажать кнопку «Обновить связывания» в нижней части окна настройки весов. Кости свяжутся с мешом заново, и точки меша перестанут двигаться.

В работе с мешем может случиться ситуация, когда нужно перепривязать меш к другой кости. А веса уже настроены, и значит, что-то менять затруднительно. В Spine для замены кости в настройке меша есть специальная функция «Swap». Чтобы заменить кость, нам нужно сначала добавить новую кость в список костей меша. Затем выделить новую кость и кость, веса которой мы хотим назначить на новую. Нажав кнопку «Swap», мы заменим кости. Теперь можно удалить старую кость из списка или назначить на нее новые веса. Это очень полезная функция для редактирования настроек меша в процессе работы с анимацией. Иногда, уже анимируя, мы понимаем, что структура костей не идеальна. И функция «Swap» поможет не тратить много времени на перенастройку костей и мешей.

Меш. Free form deformation

Мы уже использовали меш в анимации, привязывая его к костям и настраивая веса. Но есть еще один способ использования меша. Мы можем управлять точками напрямую, без костей, и таким образом создавать произвольные формы для анимации.

Настройка меша с костями имеет заметные ограничения. Мы не можем подвинуть любую точку меша в любом направлении. Мы оперируем только областями, привязанными к костям. Причем и менять эти области в процессе анимации мы тоже не можем. А если анимировать точки меша напрямую, то у нас появится полный контроль над формой объекта. Мы сможем двигать любое количество точек и постоянно выбирать разные области меша.

Делается это в анимации следующим образом: необходимо выбрать меш и после этого рамкой или кликом выбрать определенные его точки. Выбранные точки можно двигать стандартными инструментами, тем самым фиксируя ключи на шкале времени.

Такой детальный контроль вызывает свои сложности. Для создания нужной формы нам необходимо произвести детальную настройку точек. В случае с костями мы создаем движение намного проще, буквально парой инструментов. Свободное трансформирование меша — кропотливая работа.

К тому же такое трансформирование меша — неоптимизированное решение, так как для каждой точки меша задаются свои параметры и положение.

Деформация меша может использоваться для создания сложных произвольных форм, когда костями не добиться нужного результата, или для помощи в анимации костями. Если в процессе возникли определенные артефакты, мы можем вручную подправить меш и избавиться от них. Но все же используйте этот прием только в крайнем случае.

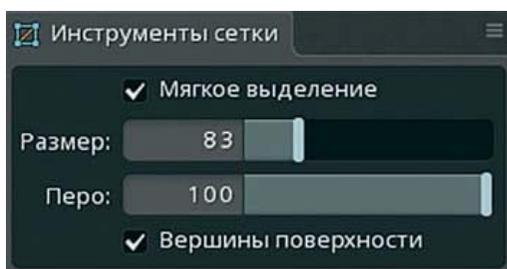
Также сложности с трансформацией меша напрямую возникают при желании скорректировать промежутки движения. Допустим, мы задали искажение 1 и искажение 2.

И задали анимацию из 1 в 2. Если мы захотим сделать промежуточную настройку меша, это будет затруднительно.

Для свободного трансформирования меша есть дополнительный инструмент. Когда мы выделяем часть точек и двигаем их, то часто возникают артефакты изображения на стыке выделенных точек и невыделенных, так как в этом месте изображение будет сильно искажаться. Для того чтобы искажение происходило плавно, есть инструмент «Мягкое выделение». Находится он в панели «Инструменты сетки» в меню «Вид». Итак, у нас появилось окно инструмента, в котором мы можем настроить его параметры. Для начала проставляем галочку возле инструмента «Мягкое выделение».

Бегунок «Размер» будет менять размер области мягкого выделения. Эта область отображается серым кругом в рабочей области. Выбрав конкретную точку меша и меняя положение бегунка, вы будете видеть размер круга относительно этой точки. Также будут подсвечиваться соседние точки меша, которые попадают в круг. Чем ближе к центру круга, тем больше влияние деформации. Это отображается цветом. Светло-голубой — сильное влияние. Темно-синий — слабое влияние.

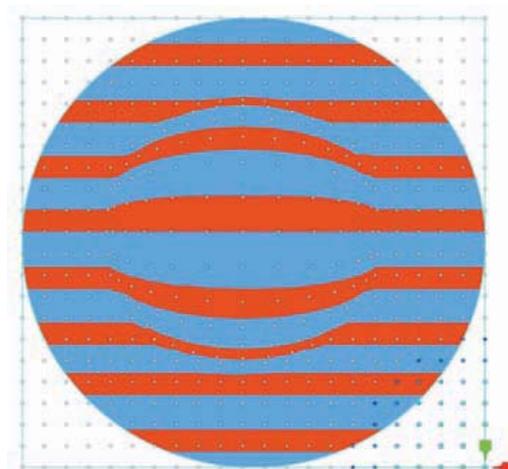
Параметр «Перо» задает размер кисти, внутри которой точки перемещаются одинаково. Двигая бегунок, вы увидите второй серый круг внутри первого. Максимальный его диаметр равен размеру первого круга. Все внутри этого дополнительного круга будет искажаться одинаково. А в области между внутренним и внешним кругами создастся градиент влияния на точки меша.



Галочка, проставленная возле пункта «Вершины поверхности», позволяет деформировать границы меша. Если ее отключить, то деформация будет происходить только внутри контура.

С помощью этого инструмента можно легко добиться эффекта сферизации меша. Эдакое вздутие и сжатие текстуры.

В целом я бы рекомендовала эту функцию Spine только для уникальных задач на крайний случай. Нужно понимать, что такая возможность в программе есть, но использовать ее только при острой необходимости, так как у этой техники анимации много минусов и полный контроль над деформацией того не стоит.



ГЛАВА 10

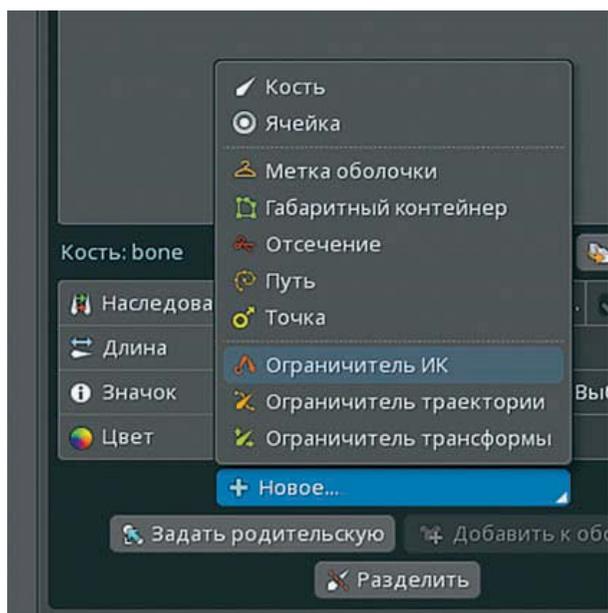
Ограничители

Инверсная кинематика

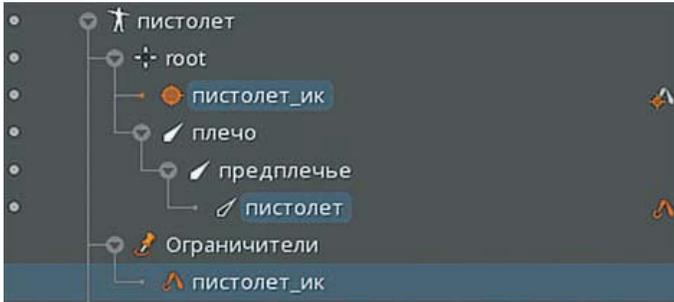
Инверсная кинематика — это отдельная точка влияния на кость. Когда мы работаем с костями напрямую, через инструменты преобразования, изменения происходят по иерархии. Дочерние кости следуют за родительскими. Иногда нам нужно настроить обратное влияние, чтобы изменение дочерней кости влияло на родительские. Это и позволяет сделать инверсная кинематика. Самый простой и понятный для нас пример — это человеческая рука. Движения кисти тянут за собой предплечье и плечо. В анимации намного удобнее управлять кистями, чтобы оказать влияние на остальную часть руки. Обойтись без инверсной кинематики в анимации рук практически невозможно.

Необходимый нам инструмент находится во вкладке «Новое». В выпадающем окне нужно выбрать «Ограничитель ИК». Чтобы эта функция была доступна, у нас должна быть выделена кость, которой мы будем управлять с помощью инверсной кинематики. В случае с рукой это кость кисти. Выбрав этот пункт, мы должны кликнуть на рабочей области там, где хотим создать точку инверсной кинематики. Если нажать на самую кость, то точка создастся на кончике кости. Лучше делать именно так, потому что, если мы создадим инверсную кинематику в произвольном месте, кость повернется кончиком по направлению к точке ИК. Наша начальная настройка скелета изменится. И если мы будем менять степень воздействия инверсной кинематики на кость или отключать ее в определенный момент анимации, то будет происходить движение. Это может стать проблемой при работе с анимацией.

Итак, у нас создалась точкой красная кость. Но, по сути, это обычная кость. Она, кстати, появилась в списке костей в панели справа, обратите внимание. Но также

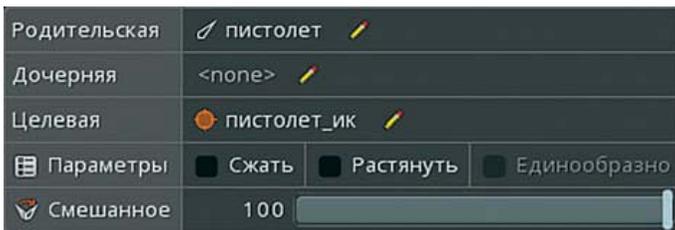


в панели справа созданся и другой элемент. В меню ограничителей появился элемент с таким же названием, как у кости. В нем мы как раз и будем задавать настройки инверсной кинематики.



Попробуйте подвигать кость инверсной кинематики. Кость, к которой мы ее прикрепили, будет следовать кончиком за ней. Сейчас инверсной кинематике подвержена только одна кость. Это первый вид инверсной кинематики. Второй вид — когда влиянию подвержены две кости. Для трех костей инверсную кинематику сделать нельзя. Позже мы рассмотрим почему.

В нижней части панели справа при выделении инверсной кинематики появляется панель ее настроек. Давайте посмотрим, какие пункты мы можем там настраивать.



Первые три строки занимают как раз две возможные кости, подверженные влиянию, и сама кость инверсной кинематики (целевая). В следующей строке находятся «Параметры». «Сжать» и «Растянуть» соответственно сжимают и растягивают кость, на которую влияет инверсная кинематика, а также изображение, привязанное к кости. Происходит масштабирование кости, что оказывает влияние на прикрепленные к ней изображения.

Если оба флажка выключены, при перемещении инверсной кинематики кость будет следовать за ней поворотом, но при этом никак не исказится. Если мы нажмем «Растянуть», то кость будет растягиваться настолько, что ее кончик всегда будет у инверсной кинематики. «Сжатие» работает аналогично, только в обратную сторону. То есть, если мы включим оба флажка, кончик кости будет всегда на месте инверсной кинематики. Но мы можем включать только один флажок и тем самым направлять влияние инверсной кинематики в какую-либо сторону. Все изменения будут происходить только по одной локальной оси. Если мы хотим, чтобы изменения транслировались

и на вторую ось, то есть масштабирование всего объекта происходило равномерно, то должны нажать флажок «Единообразно».

«Смешанное» контролирует степень влияния инверсной кинематики на кость. Нулевое значение соответствует отключенной ИК. Значение 100% означает, что ИК включена полностью. Промежуточные значения, которые оказывают определенное влияние на кость, используются намного реже, чем крайние.

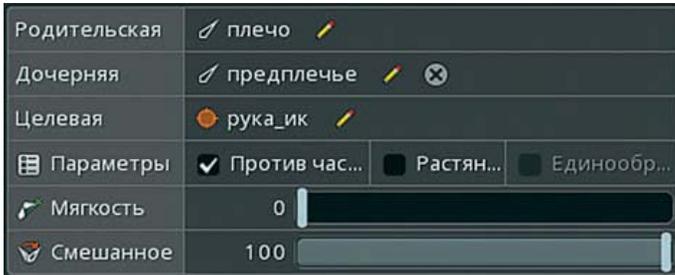
Теперь рассмотрим второй вид инверсной кинематики — влияние на две кости сразу. При создании ограничителя ИК нужно просто выбрать две кости (выделять кости можно с зажатой клавишей Ctrl) и на кончике второй создать ограничитель.

Этот вид ИК можно успешно применить к руке. Поставим инверсную кинематику на запястье, а предплечье и плечо будут подвержены его влиянию. Получается, что как раз две кости участвуют в инверсной кинематике. Таким образом, двигая кость инверсной кинематики на запястье, мы можем управлять рукой и создавать различные движения для нее. Теперь персонаж сможет потянуться к определенной точке, взять предмет руками или атаковать ударом кулака. Такая настройка используется для множества вариантов анимации рук.



Пример инверсной кинематики на запястье с подключением плеча и предплечья

Окно настроек инверсной кинематики в нижней части панели справа немного изменилось. Теперь мы видим назначенную дочернюю кость, а ранее у нас в этом поле было значение none. Нажав на крестик рядом с костью, мы удалим ее из инверсной кинематики и вернемся к виду с одной костью, а нажав на карандаш — заменим кость в инверсной кинематике. Обратите внимание, что на панели появились новые параметры. Давайте обсудим их подробнее.



Пример настроек ИК для двух костей

Параметр «Против часовой» настраивает угол сгиба дочерней кости. С выключенным или включенным флажком кость будет менять направление сгиба на противоположное.

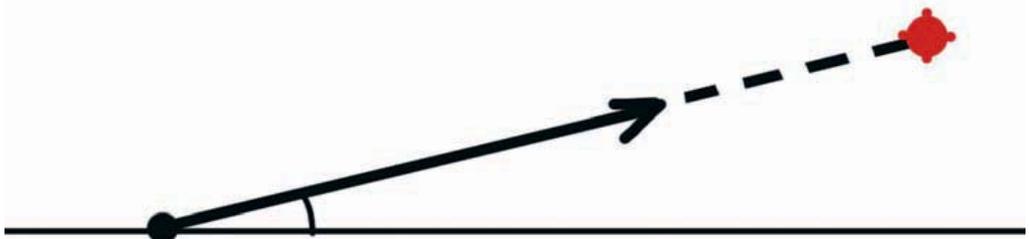
Параметр «Растянуть» теперь добавит растяжение для двух костей сразу.

«Мягкость» используется для устранения артефакта, возникающего при начале сгиба костей. Если этот параметр находится на нуле, то из полностью выпрямленного состояния в слегка согнутое кости переходят через небольшой скачок. Чтобы его убрать, мы настраиваем бегунок. Но в таком случае возникает некоторая сложность при контроле ИК. Чем выше «Мягкость», тем дальше нам нужно оттянуть точку ИК для полного выпрямления. Этот параметр настраивается опытным путем. Или, если у вас изначально кости в согнутом положении и полностью выпрямлять их не нужно, оставляем «Мягкость» на нуле.

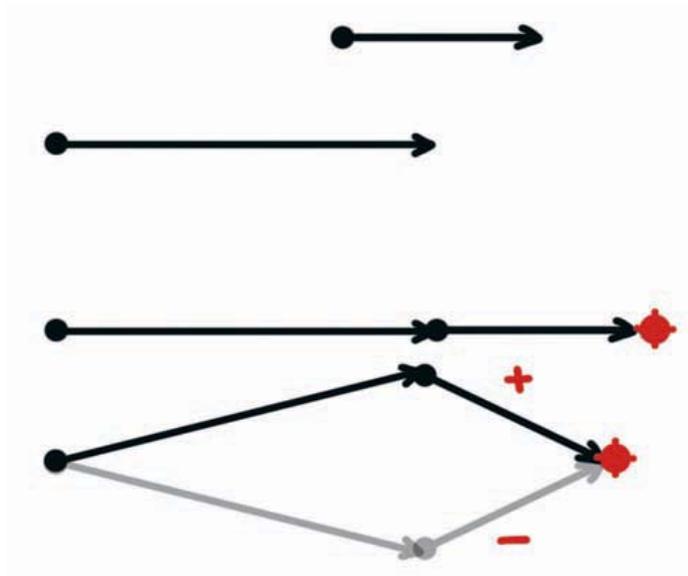
ПОЧЕМУ НЕТ ИНВЕРСНОЙ КИНЕМАТИКИ ДЛЯ ТРЕХ КОСТЕЙ?

Как я уже сказала, инверсную кинематику можно создать только для одной или двух костей. Так устроено по причине строго определенного положения этих костей, возможности расчета их положения относительно инверсной кинематики.

Если кость одна, то все просто. У нас есть кость и есть точка, куда кость «смотрит». Мы всегда строго задаем угол ее поворота. Различных вариаций одного и того же положения кости относительно ИК быть не может.



Аналогичная ситуация и с двумя костями. Допустим, у нас есть две кости определенной длины и ИК, которая управляет ими. С помощью инверсной кинематики мы задаем расстояние между крайними точками костей, не меняя их длину (или меняя длину сразу обеих, что сохраняет пропорции их размеров). Эти две кости могут подстроиться под заданное расстояние с помощью угла поворота, который также будет в одном и том же варианте для каждой из них. Разве что знак угла мы можем менять и выгибать кости в одну или другую сторону.



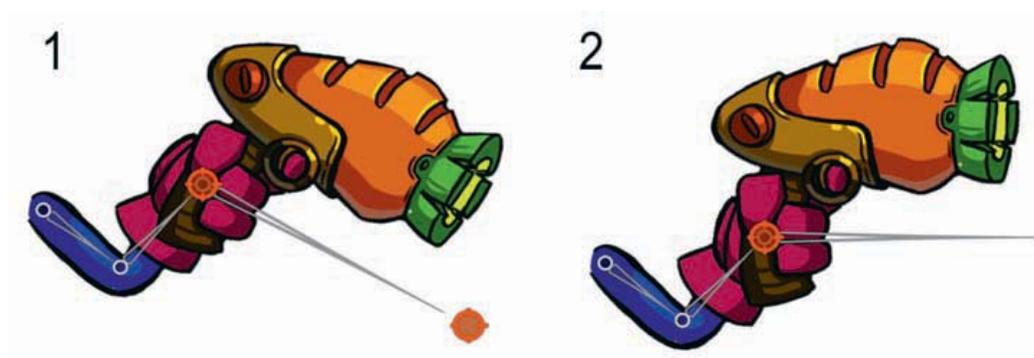
В случае с тремя костями мы не сможем предугадать, как они себя поведут. Образуется множество вариаций их положения и углов. А значит, смысл инверсной кинематики потеряется. Как мы сможем контролировать движение, если положение костей будет случайным? И техническая реализация такой функции со стороны разработчиков программы тоже не ясна. Можно, конечно, сделать так, чтобы программа каждый раз вычисляла определенное положение костей из всех возможных вариантов их поворота и положения. Но смысла в этом нет, так как движение мы контролировать не сможем.

ПРИМЕР С ПИСТОЛЕТОМ. КАК ИСПОЛЬЗОВАТЬ НЕСКОЛЬКО ОГРАНИЧИТЕЛЕЙ

Мы настроили инверсную кинематику для руки с пистолетом. Но таким образом движется только сама рука. А нам нужна анимация стрельбы и имитация отдачи. Стрельба происходит в определенную точку, которая будет задаваться в игре. А значит, мы точно не знаем, куда стреляет персонаж. Нам нужно задать точку, за которой будет следить пистолет. Но рука также должна участвовать в этом движении. В таком случае можно применить настройку с двумя точками инверсной кинематики.

Для кисти руки, которая держит пистолет, создаем новую кость. Начало кости находится в точке инверсной кинематики. Для удобства советую вытянуть кость вперед. Для этой новой кости создаем свою инверсную кинематику на кончике.

Теперь, двигая инверсную кинематику руки, что была у нас первой, мы можем отслеживать, как пистолет следит за целью. В подобной игровой механике часто требуется, чтобы оружие персонажа смотрело вперед, вне зависимости от того, как он сейчас двигается. Для создания такого эффекта нужно оттянуть инверсную кинематику кисти далеко вправо от пистолета.



В первом случае пистолет следит за ближней точкой, во втором — смотрит вперед

Таким приемом мы добиваемся отключения наследования вращения для кости кисти. Да, мы можем отключить наследование вращения, убрав соответствующую галочку в настройках кости. Но в этом случае в анимации нельзя будет включить наследование снова. Даже в разных анимациях мы не сможем регулировать данный параметр. Галочка в настройках актуальна для всех анимаций в проекте. При создании инверсной кинематики в нашем примере мы всегда сможем регулировать параметр «Смешанное» и настраивать степень влияния ИК на кость, а значит, включать и отключать наследование вращения в любой момент анимации.

ПРИМЕР ИСПОЛЬЗОВАНИЯ ИНВЕРСНОЙ КИНЕМАТИКИ. ГЛАЗА ПЕРСОНАЖА

У нас есть круглый зрачок глаза и область белка глаза около овальной формы. Нужно создать две кости: одну по центру белка и одну по центру зрачка. Кость зрачка является дочерней к кости белка. Кость белка для удобства лучше вытянуть, а кость зрачка

поставить точкой. Инверсная кинематика ставится на кончике кости белка. Кость зрачка не наследует вращения и масштабирования от родительской. Также ставим галочку возле параметра «Сжать» и оставляем пустым поле в параметре «Растянуть». Важный момент — кость зрачка должна стоять строго на линии родительской кости, иначе будут скачки при движении. Нужно поставить кость в максимальном приближении в окне программы. Теперь, когда мы потянем инверсную кинематику в сторону, глаз будет следовать за ней. Таким образом, мы можем задавать направление взгляда персонажа и даже заставлять его следить за каким-либо объектом.

Если мы подвинем все кости в центр глаза, то они наложатся друг на друга. Обратите внимание, что зрачок не выходит за пределы глаза. Это фишка данной настройки костей. Движение зрачка происходит за счет сжатия родительской кости. Зрачок двигается благодаря тому, что мы отключили наследование масштабирования и поворота, и осталось только движение элемента. А сжатие кости глаза происходит за счет движения инверсной кинематики. Поскольку параметр «Растянуть» мы отключили, кость зрачка не сможет выйти за пределы размера родительской кости. Длина кости глаза соответствует максимальному расстоянию, на которое может двигаться зрачок. Таким образом, мы задали предел для движения зрачка.



Вытянутая кость — кость по центру белка глаза. Кость точкой — кость по центру зрачка

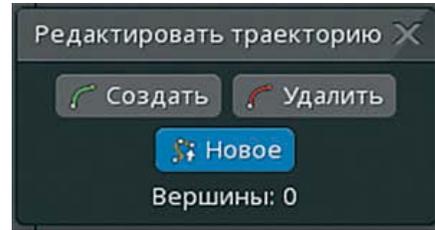
Можно регулировать радиус движения зрачка, перемещая кость зрачка по родительской. Радиус движения зрачка зависит от двух факторов: длины кости белка глаза и положения кости зрачка на родительской кости. Если мы переместим кость зрачка ближе к основанию родительской, радиус будет меньше. Оптимальная длина кости глаза — от центра до края. А положение кости зрачка можно отрегулировать в зависимости от того, насколько сильное движение глаз будет в анимации.

Ограничитель траектории

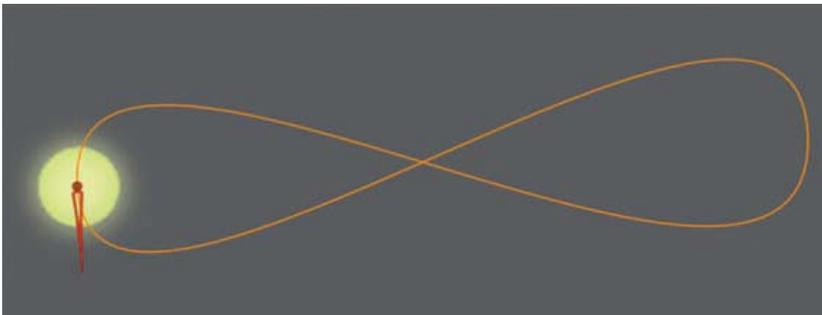
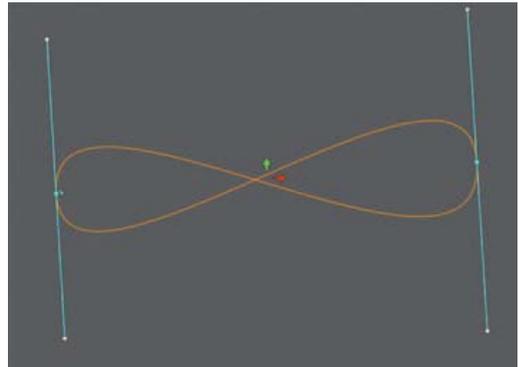
Следующий в меню ограничителей — ограничитель траектории. С его помощью мы можем задавать произвольную траекторию движения объекта в виде кривой линии и запускать движение по этой траектории. Выделив кость и нажав во вкладке «Новое» пункт «Путь», создаем основу для линии движения. Появляется окно создания

в нижней части рабочей области. Путь создается с помощью точек и кривых Безье. Щелчком на экране создаем точки, после чего редактируем кривую.

Кнопки окна говорят сами за себя. В верхней строке — «Создать» и «Удалить» точки. Нажатие на кнопку «Новое» запускает создание пути с нуля. Чтобы закончить создание пути, можно щелкнуть на первоначальную точку и тем самым создать замкнутый контур. Если просто щелкнуть правой кнопкой мыши, тогда контур будет незамкнутый. Путь, привязанный к слоту кости, появится в правой панели структуры проекта.



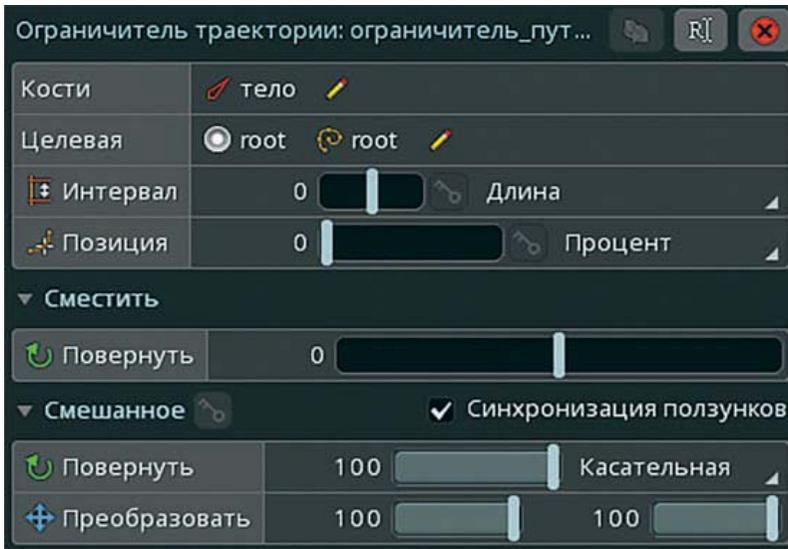
А теперь запустим объект по этой траектории. Сначала берем одно простейшее изображение круга и создаем для него кость. Выделив кость изображения, выбираем в меню «Новое» ограничитель траектории. После этого программа предлагает выбрать путь для данного объекта. Просто щелкаем по созданному ранее пути левой кнопкой мыши и называем наш ограничитель. Таким образом кость круга привязывается к пути.



Выбрав ограничитель в структуре проекта, мы вызовем окно его настроек. С помощью этих настроек можно управлять движением по траектории.

В поле «Кости» отображаются все кости, привязанные к этому ограничителю. Кость может быть как одна, так и несколько. Пока в нашем примере кость одна.

В строке «Целевая» указывается слот и путь в этом слоте, к которому мы привязали ограничитель. Путь в слоте также может быть несколько. Это удобно для переключения различных траекторий. Меняя видимость пути в структуре проекта, мы активируем один из путей для объекта.



«Интервал» задает интервал между костями. В случае с одной костью он ни на что не влияет. Подробнее разберем этот пункт на примере с несколькими костями.

«Позиция» задает положение кости на траектории. Именно этим показателем мы и двигаем объект по пути. Обратите внимание, что при перемещении объекта кость смотрит в направлении движения вдоль пути. Если нам нужно изменить этот момент, мы должны воспользоваться бегунком в строке «Повернуть» в поле «Сместить». Этим бегунком мы задаем постоянный поворот кости в градусах.

Чтобы кость смотрела по направлению пути, необходимо воспользоваться бегунком «Повернуть» в панели «Смешанное». По умолчанию он выкручен на 100. Если мы поставим нуль, то кость всегда будет смотреть туда, куда она смотрела в изначальной настройке при ее создании.

Пункт «Преобразовать» разделен на два бегунка. Первый задает влияние ограничителя по оси X, второй — по оси Y. Соответственно, мы можем устанавливать траекторию только по одной из осей, выставив один бегунок в значение 100, а другой — в 0, или задавать определенную пропорцию влияния по каждой из осей. Галочка, проставленная возле пункта «Синхронизация ползунков», позволяет редактировать все три бегунка в панели «Смешанное» одновременно.

Обратите внимание, что в панели «Смешанное» мы можем только уменьшать влияние ограничителя. То есть мы можем уменьшить траекторию за счет уменьшения значений ползунков. Если поставить одинаковые значения по обеим осям, то объект будет двигаться по аналогичной траектории, только в уменьшенном виде. Увеличить влияние, то есть поставить значение больше 100, мы не можем.

Я бы не советовала устанавливать бегунок поворота в промежуточные значения. Или опускаем полностью до 0, или выкручиваем до 100. На промежуточных значениях движение может быть неплавным.

Ограничитель траектории. Практические примеры

ПРИМЕР ДЛЯ ОДНОЙ КОСТИ. СВЕТЛЯЧОК

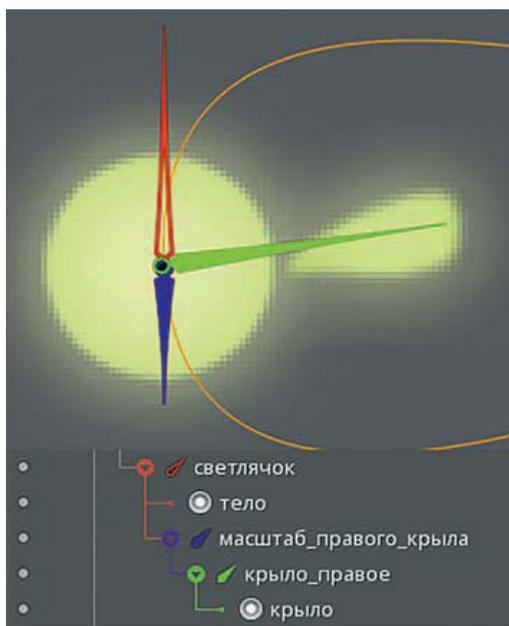
А теперь создадим анимацию светлячка, летающего по траектории. Для произвольного движения лучше создавать траекторию, близкую к восьмерке, а не траекторию круга или прямой. В этом случае движение будет выглядеть максимально естественно. Плюс к тому восьмерка создается довольно легко по сравнению с реалистичной произвольной траекторией.

У нас уже созданы путь, ограничитель и кость тела светлячка. Теперь нам нужно настроить движение и встроить в проект анимацию крылышек. Поскольку анимация крылышек симметрична, мы создадим одну анимацию и скопируем ее, отразив по горизонтали. Итак, нам нужна кость по центру тела, которая будет являться дочерней к основной. Назовем ее «масштаб_правого_крыла» и к этой кости сделаем дочерней кость крыла. Кость масштаба нам пригодится позже для копирования анимации на второе симметричное крыло.

Кость крыла может находиться как в центре тела, так и в месте стыка крыла с телом. В зависимости от этого получатся слегка разные движения. Центр вращения соответствует положению основания кости, а значит, и анимация крыла будет сопоставляться с этой точкой. Тут важно обратить внимание на то, что не обязательно кость должна находиться на изображении. Кость может быть привязана к изображению и находиться в любом нужном нам месте.

В этом же примере мы рассмотрим создание двух анимаций и комбинирование их с помощью дорожек в предпросмотре. За цикл движения по траектории светлячок взмахнет крыльями очень много раз, но не будем же мы создавать десятки копий этой анимации на шкале времени.

Для анимации крыльев нам нужно создать отдельную анимацию в структуре проекта. У нас будут две анимации: одна — для движения по пути, другая — для движения крыльев. Выделим анимацию крыла и будем работать в ней.



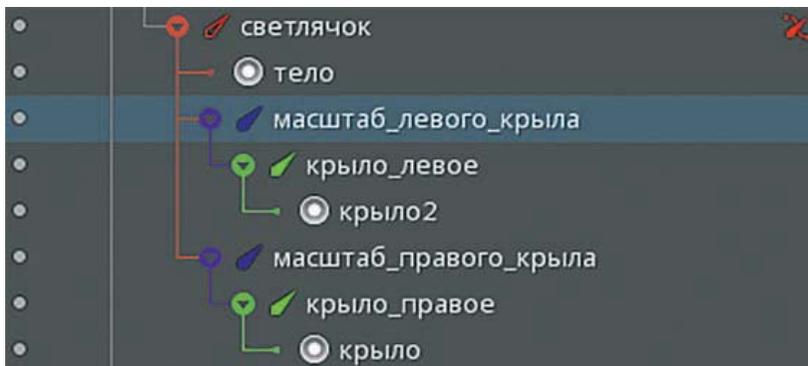
Анимация крыла будет простейшей. Поворачиваем крыло вверх на нужный угол, фиксируем ключ. В третьем кадре поворачиваем крыло на тот же угол в другую сторону и также фиксируем ключ поворота. И копируем 0-й кадр в 6-й. Анимация длится всего шесть кадров. Для такой скорости не обязательно даже настраивать график, так как разницы вы особо не увидите. Если вы хотите сделать медленные взмахи крыльями, то расширьте анимацию настолько, насколько вам нужно. И тогда будет смысл добавить замедление в начале и конце движения.

Далее нужно выбрать кость «масштаб_правого_крыла» и продублировать ее. Для этого нажимаем иконку в нижнем правом углу панели структуры проекта или сочетание клавиш Ctrl + D.

И теперь нужно выставить для новой кости значение -1 в поле масштаба по оси Y.



Структура проекта выглядит следующим образом. Советую сразу давать костям говорящие названия, чтобы удобнее было ориентироваться в структуре проекта.



Запустив анимацию, мы увидим, что светлячок синхронно машет двумя крыльями одновременно. И нам не пришлось для каждого крыла проделывать одни и те же действия.

Для отражения анимации обязательно должна быть создана отдельная кость. Если вы попытаетесь убрать ее и сделать отражение, то увидите, что синхронно крылья не двигаются. В простейших случаях еще можно исправить отраженную анимацию, изменив фазу, но в более сложных примерах отражение у вас пойдет абсолютно не по плану. Анимация не получится такой же.

Теперь осталось запустить светлячка по пути. Переключаемся обратно на первую анимацию с анимации крыла. Выбираем 0-й кадр и выделяем ограничитель пути. Фиксируем ключ параметра «Позиция» в нуле, нажав на иконку ключика рядом.

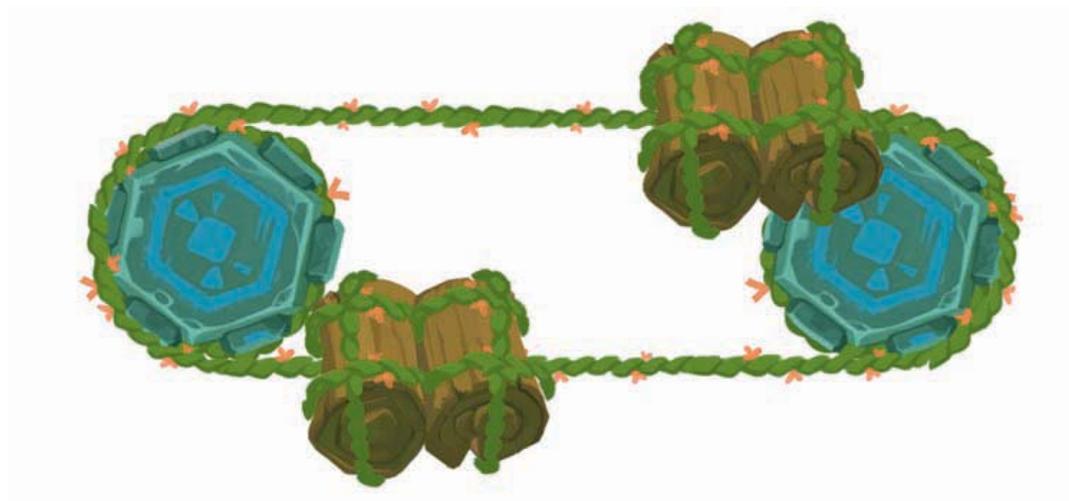


И в кадре окончания анимации, допустим в 60-м, делаем то же самое, но для значения 100 в параметре «Позиция».

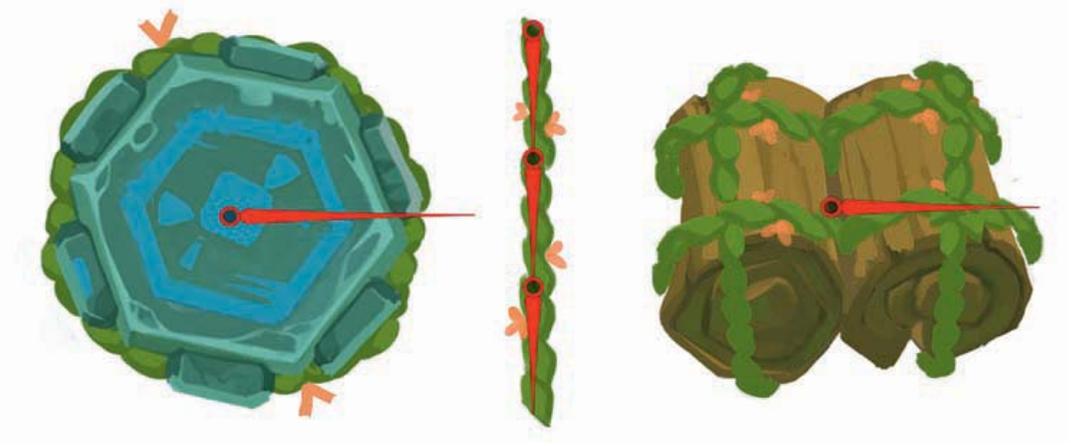
Теперь, чтобы увидеть всю анимацию целиком, нужно открыть окно предпросмотра. В нем на нулевую дорожку назначаем первую анимацию — выделяем дорожку и щелкаем по анимации. Выбираем дорожку с номером 1 и щелкаем по второй анимации. Подобные анимации экспортируются в движок как две отдельные анимации. В движке мы соединяем их в одну аналогично соединению в окне предпросмотра. Но если вам нужен экспорт в виде видео или gif, то вам придется все же копировать анимации на одну шкалу времени или делать запись с экрана окна предпросмотра. Это может понадобиться для портфолио.

ПРИМЕР ДЛЯ НЕСКОЛЬКИХ КОСТЕЙ. МЕХАНИЗМ

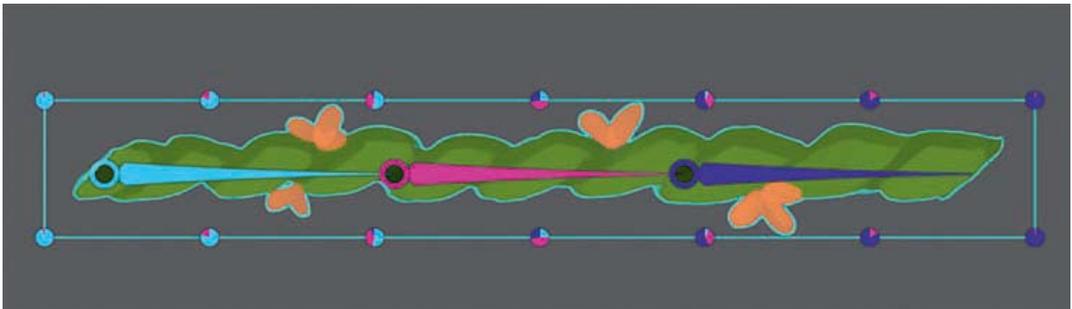
Сделаем пример с применением ограничителя траектории для нескольких костей. Возьмем для разбора анимацию механизма лифта из нашей игры *Potata*. Объект состоит из двух шестерней, платформ и каната, по которому эти платформы передвигаются.



Начнем, как всегда, с настройки скелета. Изначально проект в Spine выглядит иначе. У нас две одинаковые шестерни, а значит, только одно изображение. Платформа также в единственном экземпляре. А канат мы будем составлять из небольших кусочков, копируя их друг за другом. Итого — у нас три изображения.



Расположены они пока произвольно. Для шестерни и платформы создаем по одной кости. С их помощью мы будем управлять положением, поворотом и движением этих элементов. Для сегмента каната нужны три кости, так как мы будем делать меш и гнуть канат в месте прохода по шестерне, в крайних точках механизма. Эти кости создаем через разделение одной на три равные части. Для каната нам нужен стандартный меш прямоугольной формы с ребрами, проходящими через стыки костей и их середины. Веса задаем автоматически и два раза нажимаем кнопку «Сгладить».



Чуть изменяем структуру проекта и кладем слот с канатом в основную кость каната. Таким образом, канат становится дочерним к первой кости. Это мы делаем для того, чтобы при дублировании кусочка каната слот также дублировался и двигался вместе с новыми костями. Иначе дублироваться будут только кости, дочерние к основной.

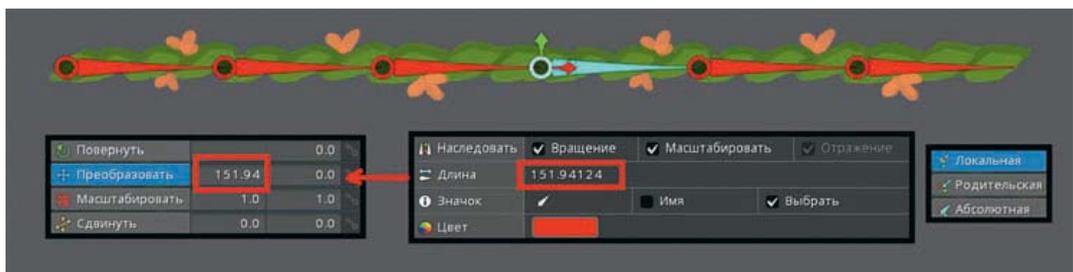


Создаем копию шестерни и отодвигаем вправо на определенное расстояние. Это будет длина механизма. Под это положение шестерней создаем путь, по которому будут двигаться канат и платформы. Сначала добавляем четыре точки в крайних верхних и нижних точках шестеренок. Затем вытягиваем усики кривых Безье этих точек

параллельно оси X. Немного доводим положение усиков и настраиваем путь так, чтобы он обходил шестеренки по радиусу.



Перейдем к настройке каната. Выделяем основную кость. Дублируем ее. Новый кусок каната вместе со всей его настройкой делаем дочерним от предыдущего. Для этого нужно опять же поменять структуру проекта в панели справа, перетащить кость одного каната в структуру другого. Далее нужно подвинуть основание второго кусочка скелета к концу первого. Совмещаем мы именно кости, а не изображения. То есть мы должны создать непрерывную структуру костей из наших сегментов. Чтобы это делать не на глаз, а точно, нужно скопировать значение длины кости в поле «Преобразовать» первой кости нового каната. Обязательно выбираем «Локальная» в меню отображения осей под рабочей областью.



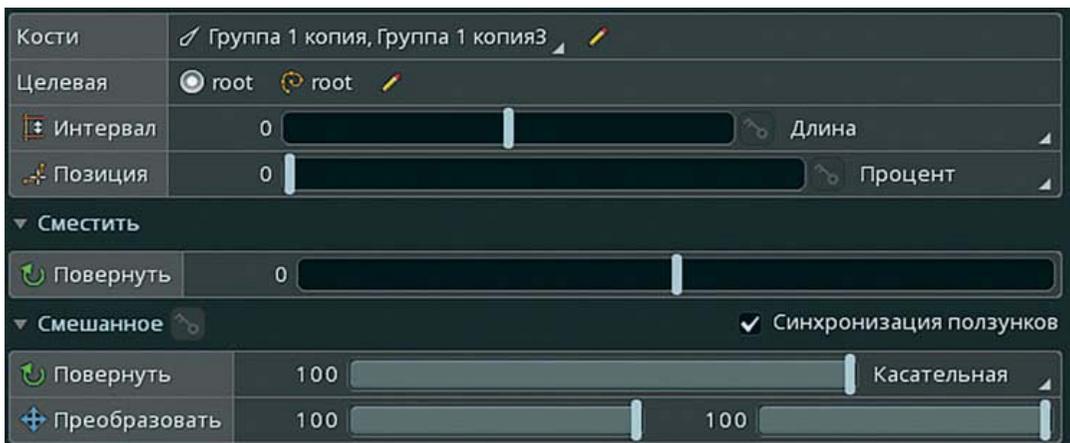
Далее повторяем этот шаг несколько раз. Когда мы выберем путь, то увидим под структурой проекта параметр его длины. Эту длину делим на длину кости. Получаем количество костей, требующихся для того, чтобы покрыть путь канатом. Поскольку у каждого сегмента каната по три кости, мы должны получившееся число увеличить в три раза и округлить до большего. Итого у нас получается 27 костей, а значит, скопировать нужно 9 раз. Но это наш пример, ваши значения будут зависеть от ваших параметров длины костей и пути. Можно округлить и в меньшую сторону. Тогда при финальной настройке положения костей на пути мы доведем канат до целостности настройкой интервалов.

Выделяем все кости каната. Это удобнее делать в структуре проекта, так как костей много. В верхней части фильтра включаем видимость только костей и выделяем их все, как файлы в окне Windows, зажав клавишу Shift и кликнув на начальную и конечную кости. Теперь для этого выделения создадим ограничитель траектории. Выбираем наш путь между двумя шестернями, даем ему название и нажимаем о'кей. Канат автоматически привязался к пути, но не совсем идеально лег на траекторию. Особенно в месте сцепления с шестеренками. Дальше мы будем доводить это положение до идеала.



ОКНО НАСТРОЕК ДЛЯ НЕСКОЛЬКИХ КОСТЕЙ

В окне настроек появилось несколько дополнительных функций.



Например, в списке костей, при нажатии на который выпадает меню со всеми костями по очереди, мы можем менять порядок следования этих костей. В нашем случае это не нужно, так как кости являются дочерними по отношению друг к другу. Но такая функция есть, и она может понадобиться, особенно если у костей нет родительно-дочерних отношений в структуре.

Теперь мы можем менять значение в поле «Интервал», так как костей несколько и между ними появился этот самый интервал. Справа есть меню выбора, в котором можно задать вариации для этого параметра.

«Длина» ставит каждую следующую кость, используя длину предыдущей. По сути, получаем формулу «длина + N», где N — это значение параметра. Если значение равно нулю, то следующая кость стоит ровно на конце предыдущей.

При нажатии на поле «Длина» выпадает меню с дополнительными функциями. «Фиксированное» задает фиксированный интервал между костями. Длина кости уже никак не влияет на расчет параметра. Параметр строго задается показателем.

«Процент» выставляет значение интервала, исходя из общей длины пути. Чтобы следующая кость начиналась строго из предыдущей, нам нужно 100% (длину всего пути) поделить на количество костей.

«Сохранять пропорции» располагает кости так, чтобы они занимали весь путь, когда значение интервала равно 100. Эта функция не дает идеального результата на точке замыкания для кругового пути, так как положения первой и последней костей будут совпадать.

Мы используем значение в процентах, поделив 100 на количество костей. Это даст самый точный результат. Кости создадут идеальный замкнутый контур, где конец предыдущей кости будет совпадать с началом последующей.

В нижней части рядом с полем «Повернуть» активируется выпадающее меню. В нем мы можем выбрать следующие пункты.



Нажатие на «Касательная» позволяет поворачивать кости по касательной к окружности. Также можно раскрыть меню с дополнительными пунктами.

«Цепь» — если активирован этот пункт, кость всегда смотрит на основание следующей кости в списке и длина костей не меняется. Кость всегда смотрит на основание следующей кости в списке (при этом длина костей остается неизменной) и затем указывает на точку пути, определяемую интервалом. Это может затруднить настройку движения по контурам с поворотами меньше размера кости и резкими изгибами, так как в этих случаях точки костей могут выходить за контур.

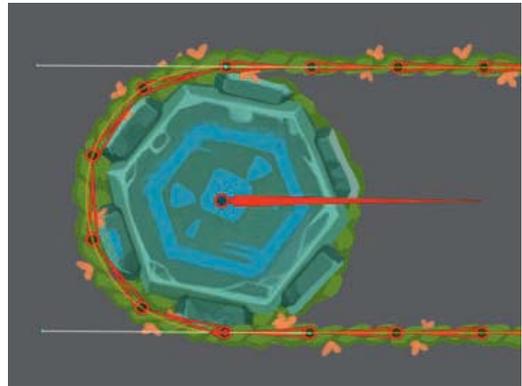
Наиболее удобным параметром в этом меню я считаю «Масштаб цепи». Когда этот параметр включен, кость указывает на определенный нами интервал и затем масштабируется так, чтобы кончик находился на траектории в той же позиции. Таким образом, все основания и кончики костей гарантированно находятся на пути.

Эти параметры сложно понимать по определениям. Лучше всего проверять их на различных примерах и визуально смотреть, как работает каждый параметр.

Поскольку мы выбрали пункт «Масштаб цепи», Spine требует от нас изменить структуру проекта. Обратите внимание, что в правом верхнем углу пункта появилась желтая точка с восклицательным знаком. Программа предупреждает, что все кости должны иметь общую родительскую кость или быть дочерними от первой кости последовательности. Иначе вычисления и результат будут неточными. Мы выделяем в структуре проекта все кости последовательности, кроме первой. Нажимаем клавишу P (parent) и нажимаем первую кость.

Настроив интервал, выбрав параметр «Масштаб цепи» и позицию через «Процент», мы добились лучшего прилегания костей к закруглению вокруг шестерни.

Выбираем кость платформы. В окне настроек под структурой проекта убираем для нее флажки наследования вращения и масштабирования. Нам нужно, чтобы она в исходном виде двигалась по траектории, поэтому оставляем только перемещение. Перемещаем эту кость на траекторию в точку начала одной из костей каната. Если мы поставим ее в другую точку, то платформа может выйти за пределы траектории. Делаем кость платформы дочерней по отношению к кости каната, в начало которой мы поставили платформу. Дублируем платформу и проделываем те же шаги. Иногда при добавлении родительской кости платформе меняется поворот. Если это произошло, просто ставим нулевое значение обратно.



На этом настройка механизма завершена, можно переходить к анимации.

АНИМАЦИЯ МЕХАНИЗМА

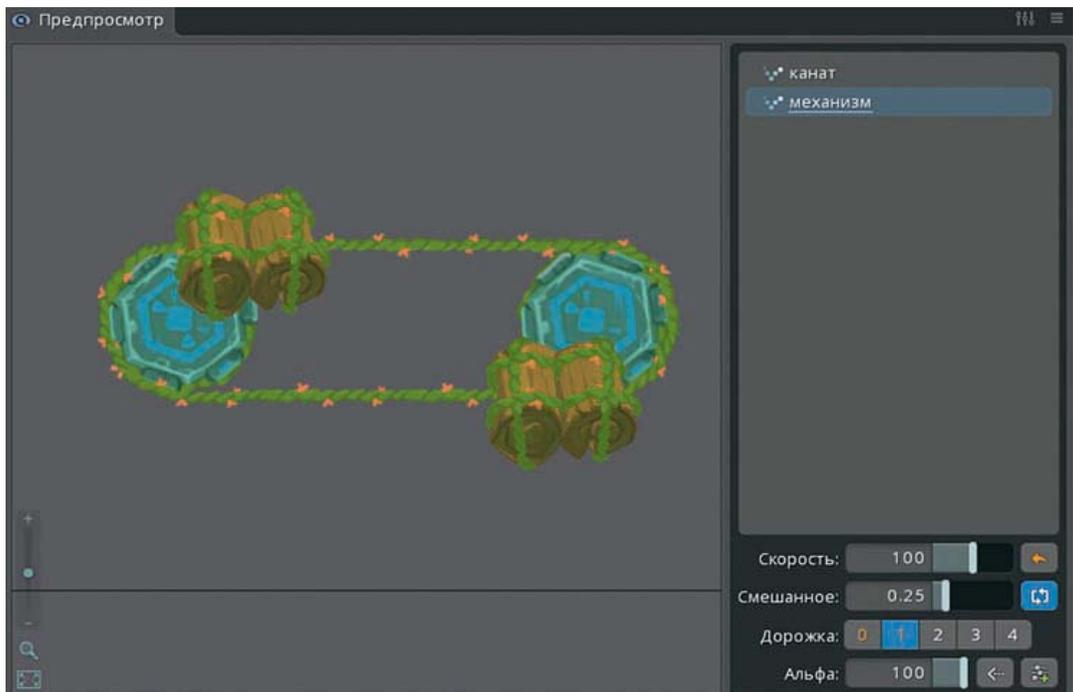
Разделим анимацию на две дорожки: анимацию шестерней и анимацию пути.

Сначала сделаем анимацию шестерней. Выделяем обе кости шестерней и фиксируем в 0-м кадре ключ поворота в нуле. В 60-м кадре ставим значение 360. Если нужно вращать в другую сторону, ставим значение -360 . Вот и все, это очень простая анимация. Переходим к анимации основной части. Канат будем делать в отдельной анимации.

Выделяем ограничитель траектории. В 0-м кадре фиксируем значение «Позиция» на нуле. В 60-м кадре ставим 100. Переходим в предпросмотр, ставим на две дорожки наши анимации и оцениваем результат. Получилось, что канат двигается быстрее, чем крутится шестеренка. А должно создаваться ощущение, что канат крутит шестеренку. Это значит, что скорость должна быть одинаковой.

Теперь настроим длину анимации каната с включенным предпросмотром. Необходимо перемещать последний кадр дальше по шкале времени, пока скорости не сравняются. В нашем случае получилось значение, равное 160. То есть анимация каната вместо 60 кадров длится 160.

Итак, анимация готова. В данном примере этап настройки намного больше и сложнее, чем этап анимации. Это еще раз показывает его важность. Ведь создав один раз хорошую базу, мы сможем легко делать и корректировать анимации.



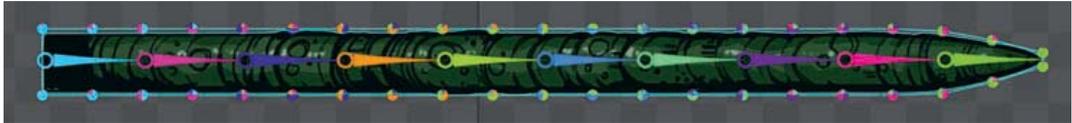
ПРИМЕР ДЛЯ ОГРАНИЧИТЕЛЯ ПУТИ. ЗАРОСЛИ. НАСТРОЙКА

В этом примере мы создадим стебель, который будет расти по определенной траектории. Эту траекторию мы зададим с помощью пути.

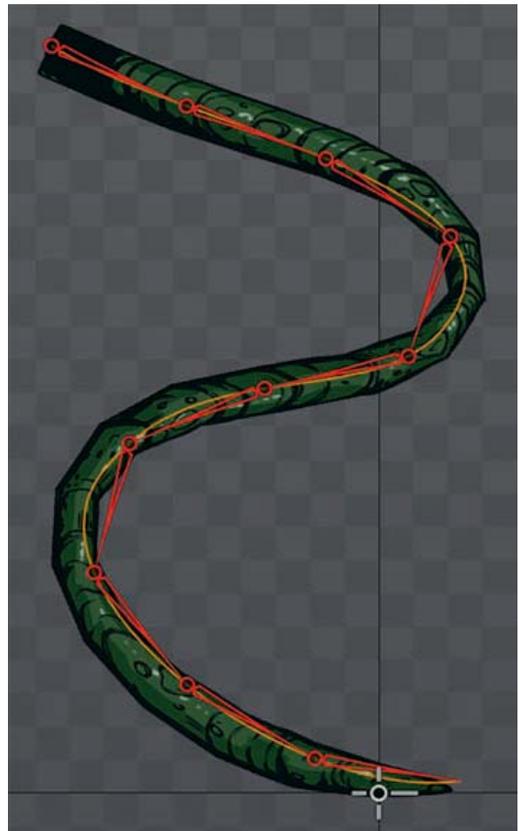
Исходник для этого примера — вытянутый стебель растения. Для лучшего изгиба в анимации изначально он должен быть прямым. Для работы с изогнутыми элементами в Spine лучший вариант — это прямой исходник. Если у нас исходник уже изначально завернут в какое-либо положение, то сделать противоположный изгиб будет практически невозможно. А из прямого положения мы можем создавать изгибы в разные стороны.

Создаем кость от одного конца стебля до другого и делим ее на части. Количество костей зависит от длины стебля и требуемой детализации. Точного значения я назвать

не могу, так как это делается на глаз и корректируется в дальнейшем, если детализации не хватает. В данном примере я разделю на 10 костей. Меш создаем прямоугольный, с ребрами на стыках и серединах костей. Если на исходном изображении есть какие-то неровности формы, то их нужно обвести точками меша. В моем примере такой неровностью является суженный носик стебля. И точки меша также сужаем в этом месте, чтобы не было лишних пустот. Веса привязываем автоматически и сглаживаем два раза. Все кости делаем дочерними от первой. Или просто не ставим галочку в пункте «Вложенные» при делении одной большой кости на несколько маленьких.



Создаем путь. Форма пути определяет траекторию роста стебля. Для наглядности примера задаем несколько изгибов в форме пути. Вся нашу цепь костей привязываем к пути с помощью ограничителя траектории. Стебель будет повторять форму пути. Уже на этом этапе можно оценить, достаточно ли нам детализации и точек меша, чтобы повторять форму пути; нужно ли попробовать сгладить веса костей или создать дополнительную детализацию на меше и добавить кости в цепь. На этом этапе настройка завершена, можно переходить к созданию анимации роста стебля.



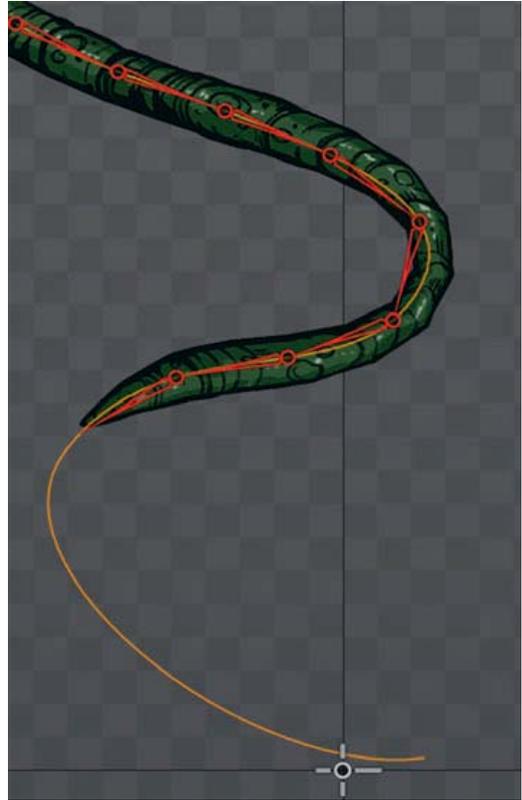
АНИМАЦИЯ

Создать подобную анимацию мы можем за счет двух параметров в настройке ограничителя траектории. Это «Интервал» и «Позиция». «Интервал» будет масштабировать кости, и за счет увеличения масштаба они разрастутся по созданной траектории. Но изменение интервала исказит исходник. Если на исходнике есть детализация, то при уменьшении интервала стебель «сморщится», а при увеличении детализация растянется. Но если у вас просто однотонная линия, то артефактов заметно не будет и можно смело работать с «Интервалом».

Если использовать «Позицию», то кости в исходном масштабе будут перемещаться по заданной траектории. Но в таком случае нам понадобится такая длина стебля, чтобы ее хватило на всю траекторию. А начало этой траектории нужно будет прикрывать маской, чтобы спрятать излишки длины стебля. Ведь он не вырастает, а только перемещается. В анимации мы используем и ту и другую настройку. Маску нам в любом случае придется использовать, если начало анимации ничем не прикрыто в самом исходнике. Например, куском земли, как в случае со стеблем. Или границами кадра.

В 0-м кадре вместо 10-го поставим интервал 4. Позицию уберем в отрицательное значение так, чтобы кончик стебля был в начале траектории. В кадре конца анимации вернем значения, которые были изначально. Это интервал 10 и позиция 0.

Если мы хотим получить заросли из таких стеблей, мы можем копировать анимацию. Чтобы растительность была разнообразной, мы меняем траектории роста. Меняем скорость анимации, а значит, и скорость роста. Можем задавать характер движения, замедляя или ускоряя рост на определенных участках анимации. Вариативность задается довольно легко, и мы можем простыми средствами получить целую группу растений с различными вариантами движения.



ПРИМЕР ОБЪЕКТА С НЕСКОЛЬКИМИ РОДИТЕЛЬСКИМИ КОСТЯМИ

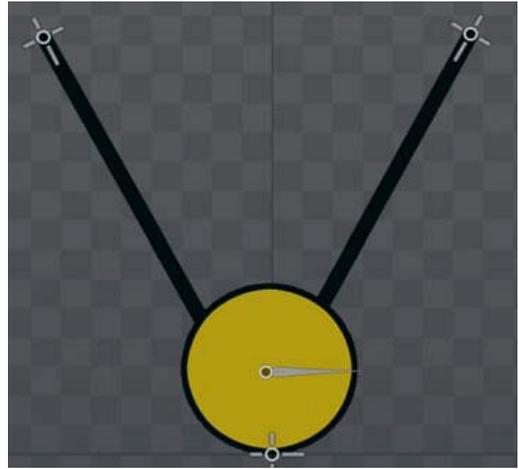
Рассмотрим пример, в котором у нас есть один элемент, привязанный к двум родительским. Например, медальон на двух цепочках. Когда двигается медальон, определенным образом должны двигаться и цепочки. Реализовать такой пример можно не одним способом. Рассмотрим несколько вариантов настройки.

В качестве исходника у нас два изображения: круг, который мы будем считать медальоном, и прямая линия.

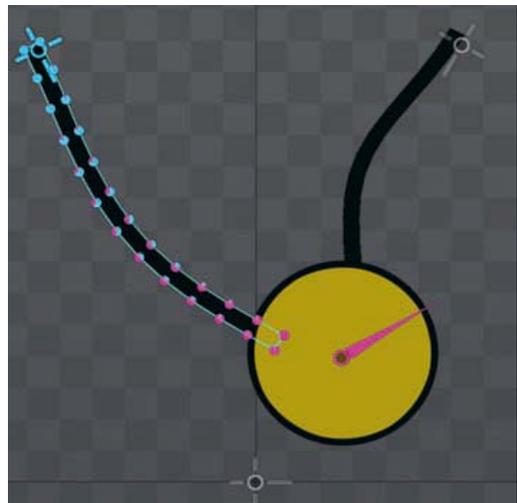
ПЕРВЫЙ ВАРИАНТ НАСТРОЙКИ. БЕЗ ИК, БЕЗ ПУТИ

Переносим две картинки в проект. Для круга создаем кость в его центре, а для линии одну кость в ее начале. Также для линии создаем прямоугольный меш. После этого дублируем линию и располагаем линии для медальона в положении цепочек.

Привязываем точки меша к костям. Верхние две точки — к кости линии, а нижние две точки — к кости круга. Прodelываем это для обеих линий. Теперь, двигая медальон, мы двигаем и меши, а верхние кости удерживают верхние точки цепочек в статичном положении. Этот вариант настройки очень оптимизированный. В нем мало костей, простейшие меши и понятная логика.



Чтобы сделать цепочки медальона гнущимися, необходимо усложнить меш. Нам нужно добавить ребер меша на линию. Для добавления детализации мешу нажимаем кнопку «Создать» несколько раз. Теперь создаем градиент весов от одного конца к другому. Делаем дополнительную кость, которую ставим внутри меша на втором конце цепочки возле круга. Эта кость должна быть дочерней по отношению к первой кости веревки. Это мы делаем для того, чтобы программа автоматически просчитала нам веса. Привязываем эти две кости к мешу (сбросив старую настройку меша). Если хотите сохранить и старую настройку, то нужно создать новый проект для новой настройки. После этого пару раз нажимаем кнопку «Сглаживать» для еще более плавного распределения весов. Дублируем линию с настройкой весов и меша и располагаем в позицию с другой стороны круга.



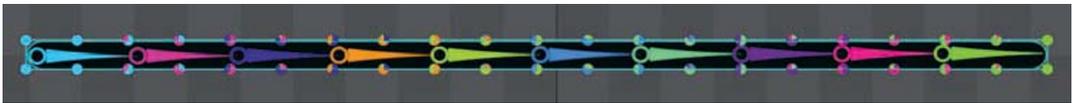
Теперь нужно убрать лишние кости из настройки скелета. Привязываем кость круга к мешу одной из линий. Выбираем кость круга и нижнюю кость линии и меняем их влияние местами, нажав кнопку «Swap». После этого нижнюю кость линии можно удалить. То же самое делаем для второй линии.

Настройка скелета и меша для медальона

Теперь при движении круга веревки, к которому он привязан, будут двигаться за ним, но уже с изгибами. Это более применимо к анимации реального предмета, а не схемы. Но оптимизация уже слегка пострадала, так как мы добавили значительное усложнение меша в настройку.

ВТОРОЙ ВАРИАНТ НАСТРОЙКИ. С ПОМОЩЬЮ ПУТИ

Начинаем с того, что создаем кость по центру круга и кость, идущую от одного конца линии к другому. После этого делим кость линии на цепь костей одного размера, дочерних по отношению друг к другу. Количеством костей задаем степень детализации. Редактируем меш по уже известному нам принципу. На каждом стыке костей и посередине каждой кости должно быть ребро меша. Привязываем меш к линии костей. Корректируем автоматическое распределение весов двойным нажатием на кнопку «Сглаживать». Такая настройка дает больше контроля над изгибом линии. Сама интенсивность и плавность изгиба определяются количеством точек меша, которое может быть одинаковым в любом виде настройки. Но линия костей позволяет этот изгиб контролировать лучше.

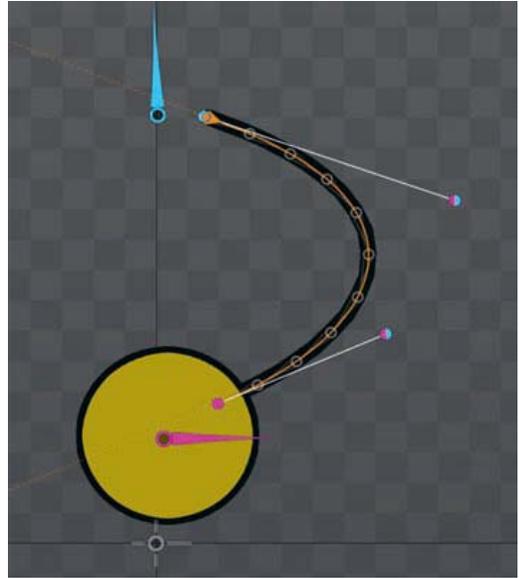


Настройка меша для цепочки медальона

Теперь нужно создать путь в корне структуры. Выделяем кость *root* и создаем путь. Проводим путь от одного конца линии к другому на всю длину. Затем все кости линии привязываем к этому пути с помощью ограничителя траектории. Настраиваем ограничитель. Ставим интервал, равный числу костей (в нашем примере их 10), и тип интервала «Процент». Это равномерно распределит кости по пути. Если масштабировать путь, кости будут масштабироваться вместе с ним. Также выбираем пункт «Масштаб цепи» и, согласно его требованию, делаем все кости линии дочерними по отношению к первой.

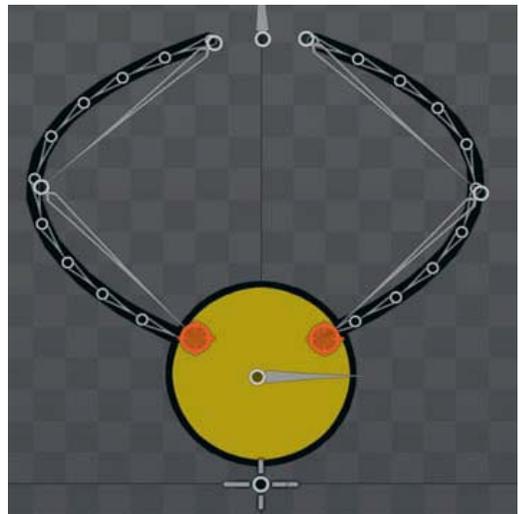
Теперь, управляя путем, мы можем расположить веревку в нужном нам положении. А главное, изменяя форму пути, мы можем задавать форму веревки. Это было сложно сделать с помощью предыдущей настройки. А в настоящем примере вместо прямых линий мы можем уже изначально задать изгиб веревок, имитирующий охват шеи, к примеру. Путь мы должны привязать к двум костям: кости медальона и новой кости сверху, которую мы создадим. Путь привязываем к костям аналогично мешу. Веса распределяем следующим образом. Точки по краям пути полностью подчиняем соответствующим костям. А точки на усиках кривой пути делим пополам на две кости, по 50% влияния на каждую. Точки, которые подчинены соответствующим костям, полностью последуют за ними, точно копируя их передвижение. А изгиб будет создаваться влиянием обеих костей.

Теперь нам нужна такая же веревка с другой стороны медальона. Ограничители мы скопировать не сможем, поэтому некоторые действия придется повторить заново. Делаем новую кость там же, где корневая кость. В нее нужно положить путь, меш и цепь костей. Дублируем новую кость и заново настраиваем ограничитель траектории с теми же параметрами, что настраивали для первой веревки. У нового пути необходимо убрать привязку костей и их веса. Масштаб новой кости, а значит и всего содержимого, ставим -1 по оси X. И привязываем путь к кости медальона и верхней кости заново. Не забудьте снять галочку с пункта «Тыльная отрисовка», если установили ее, делая предыдущий пример. Иначе у вас не будет отображаться меш при отрицательном масштабе. Веса костей настраиваем аналогично первой веревке. Теперь, двигая медальон, мы двигаем и веревки, на которых он держится. С помощью путей мы можем задавать форму веревок и их изгиб. В анимации мы будем использовать лишь одну кость медальона. Если нам понадобится дополнительный контроль над изгибом веревок, то мы можем создать еще одну кость для контроля усиков кривой пути и изменять изгиб в процессе анимации.



НАСТРОЙКА С ПОМОЩЬЮ ИНВЕРСНОЙ КИНЕМАТИКИ

Для еще большего контроля над изгибом можно добавить к предыдущей настройке инверсную кинематику. В таком случае изгиб будет более понятен. Главной точкой изгиба будет точка стыка костей. Если в случае с использованием только пути изгиб происходит в большей степени случайным образом и в разных местах веревки, то с ИК мы будем точно знать, как согнется наша линия. Это отлично подойдет для гнущихся конечностей персонажей или различного вида тентаклей и даже растительности. Но есть минус. Основной изгиб всего один, и он будет



довольно заметен. А значит, для случайных, постоянно гнущихся элементов такая настройка не лучший вариант.

Создаем по две кости (дочерние по отношению к корневой кости) на веревку и инверсную кинематику в конце веревки. Точкой стыка двух костей обозначаем основной сгиб. В нашем случае с медальоном это примерно середина веревки. Инверсную кинематику делаем дочерней по отношению к кости медальона.

Пути нужно отвязать от костей, к которым мы привязали их в предыдущем примере. В этой настройке пути должны быть привязаны к костям с инверсной кинематикой. Верхняя кость в этом примере станет ненужной, ее можно и удалить. Настройка весов будет аналогична предыдущему примеру — по 100% на краях и по 50% на усиках кривой пути.

Ограничитель трансформы

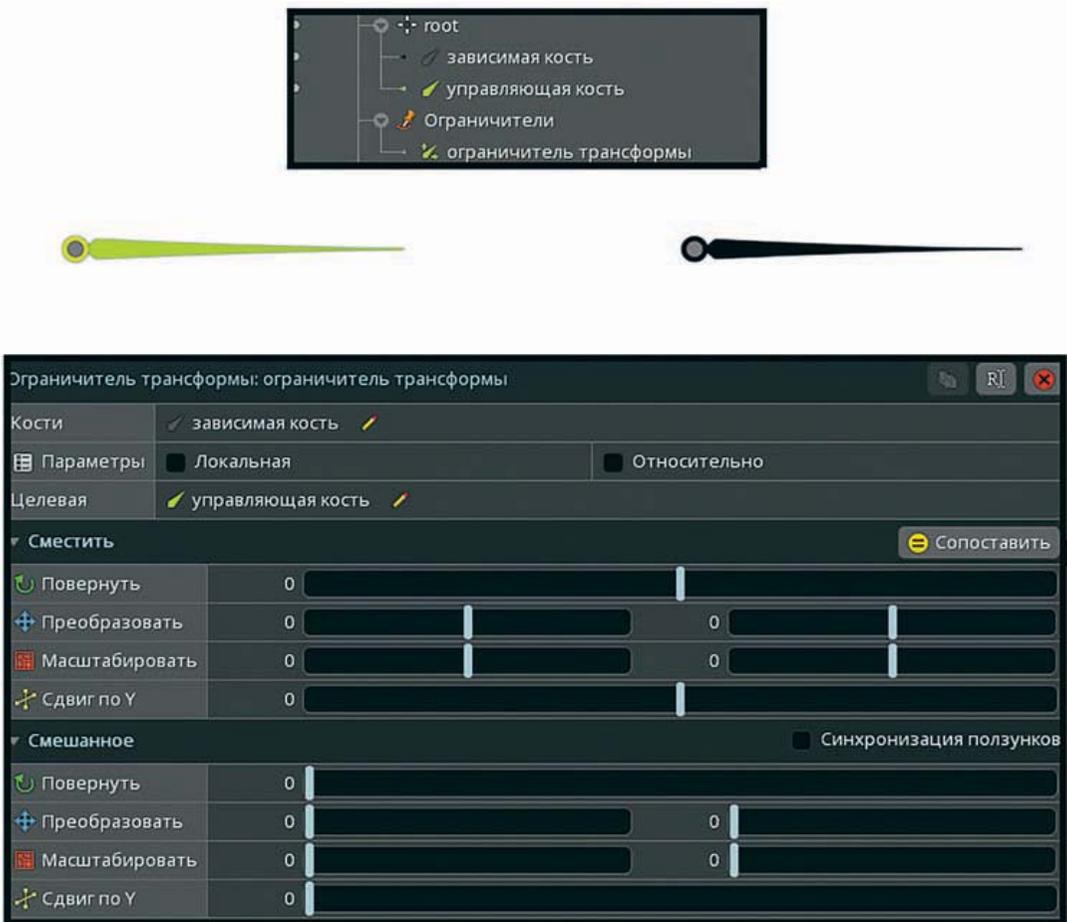
Этот ограничитель создает своеобразный аналог родительской кости, но с отдельной панелью настроек и множеством настраиваемых показателей. Это позволяет создавать довольно своеобразные отношения между костями за счет комбинации настроек.

Создаем две кости. Выделяем одну из них (правую) и в панели «Новое» создаем ограничитель трансформы. После выбора ограничителя кликаем на вторую (левую) кость. Таким образом, мы привязали первую выбранную кость ко второй и теперь можем настроить показатели влияния левой кости на правую. Называем правую кость «управляющей», левую — «зависимой», чтобы не путаться в определениях. Выделяем ограничитель в структуре проекта и видим довольно объемное окно настроек. Давайте разбираться, что нам может предоставить ограничитель трансформы.

На первой сверху плашке мы видим назначенные кости и два флажка. «Локальная» означает передачу всех параметров ниже относительно локальной оси. Если флажок отключен, то используется абсолютная шкала. Значения в разных системах осей могут отличаться, особенно в сложной структуре проекта.

Флажок «Относительно» нужно разобрать на конкретном примере. Допустим, у нас сейчас есть две кости. Координаты одной 100, 100, а второй — 500, 100. Когда флажок выключен, при настройке преобразования зависимая кость будет двигаться к управляющей. И в значении 100 они окажутся на одном месте. А если включить флажок «Относительно», то за основу будет браться положение управляющей кости относительно нуля. И преобразование подвинет зависимую кость на эти же значения. То есть координаты в случае преобразования суммируются.

Если подытожить, то при включенном флажке разница между нулем и управляющей костью добавляется к зависимой кости.



Далее спускаемся ниже и обращаем внимание на важную кнопочку «Сопоставить». Заметьте, что эта кнопка не активна при включенном флажке «Относительно». Эта кнопка вычисляет разницу между двумя костями по всем показателям и задает ее в значениях показателей ниже. Если этого не сделать, то при настройке влияния зависимая кость будет стремиться к управляющей. А зачастую нам нужно, чтобы показатели менялись синхронно, но с определенной настройкой, а зависимая кость осталась там, где она была. Для этого мы и задаем первоначальный сдвиг.

Ниже во вкладке «Сместить» мы можем настроить этот сдвиг вручную для каждого параметра отдельно. Это может понадобиться, если автоматическая настройка не дала идеального результата в анимации. Параметры преобразования и масштаба контролируются по каждой оси отдельно.

И последняя вкладка в окне — «Смешанное». Именно в ней мы можем настраивать степень влияния тех или иных изменений, производимых с управляющей костью. Настройка производится в процентах для всех показателей. Например, выставляем значение в пункте «Поворот», равное 50. Значит, при повороте управляющей кости

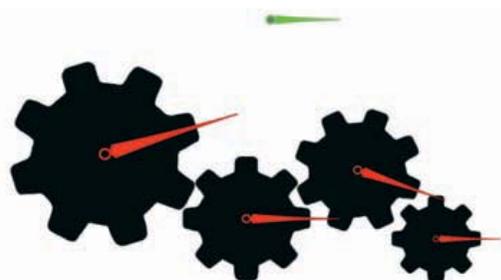
на 90 градусов зависима повернется только на 45 градусов. Флажок «Синхронизация ползунков» позволяет настроить одно и то же значение для всех ползунков сразу.

Ограничитель трансформы. Практические примеры

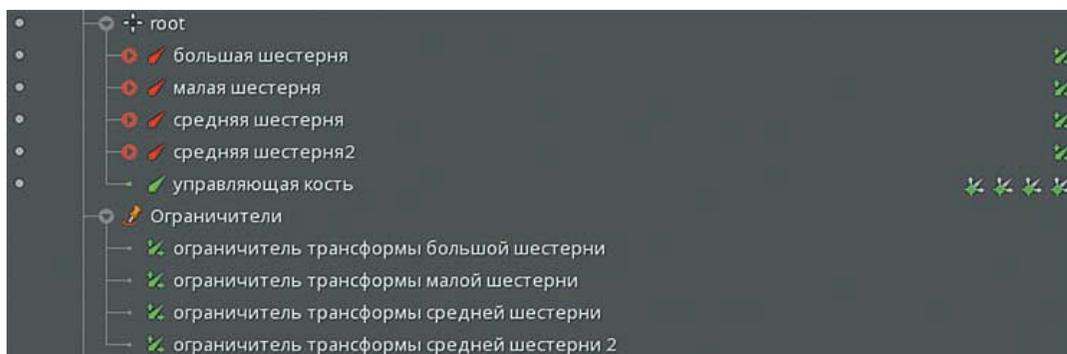
ПРИМЕР. МНОЖЕСТВО ПОВТОРЯЮЩИХСЯ ОБЪЕКТОВ

В примере лифта с ограничителем пути мы просто скопировали анимацию шестерни. Но если у нас таких шестерней будет много, применить копирование мы не сможем. В таком случае нас как раз спасет ограничитель трансформы.

Итак, создаем несколько шестеренок разного размера. К каждой из них прикрепляем кость с основанием в центре вращения шестерни. Или можно создать одну шестеренку и копировать ее, меняя масштаб, если изображение одинаковое. Располагаем их на рабочей области в произвольном порядке. Если они будут стыковаться друг с другом — это здорово. Так мы сделаем задачу еще интереснее. И скорость будет лучше читаться.



Ограничители трансформы создадим по одному для каждой шестеренки. Контролирующая кость будет одна на все шестеренки. И эта кость будет располагаться отдельно от всех шестерней. Выделяем кость шестеренки, создаем ограничитель через меню «Новое» и выделяем нашу отдельную кость в качестве управляющей. Так же делаем для всех остальных костей шестеренок.

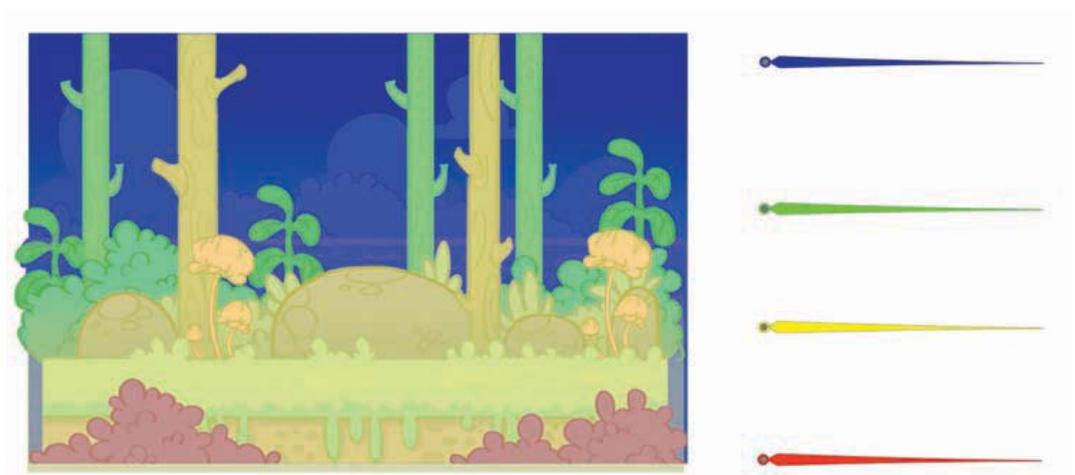


Выделяем каждый ограничитель по очереди и нажимаем кнопку «Сопоставить» в окне настроек. Далее выбираем по очереди те кости, шестерни которых будут крутиться по часовой стрелке. Рядом стоящие шестерни будут крутиться в обратной фазе. Значит, выбираем через одну. Выбираем первую и третью. В значении поля «Повернуть» во вкладке «Смешанное» ставим 100. Для остальных костей в этом же поле мы должны поставить значение -100 . Мы не можем сделать это бегунком, так как он дает выставить значения только от 0 до 100%. Но вписав значение вручную, мы можем сменить знак.

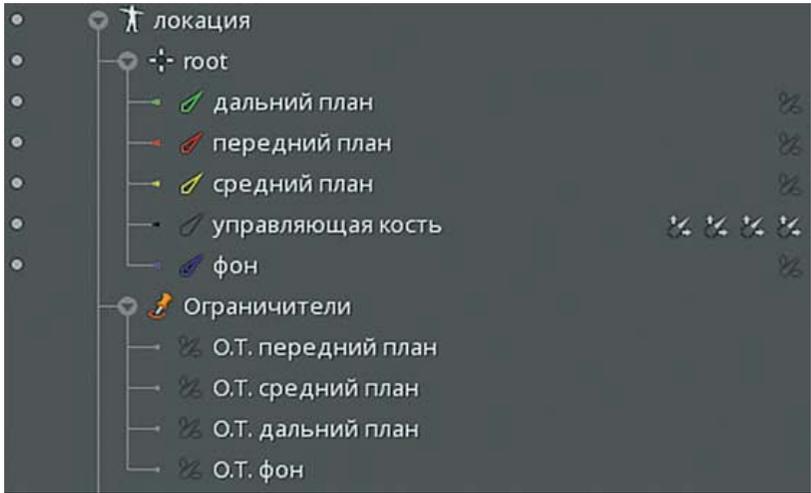
Теперь проверяем настройку, покрутив управляющую кость. Все шестеренки должны крутиться с одной скоростью, через одну в разной фазе. Таким образом, мы удобно управляем поворотом всех шестеренок с помощью всего одной кости. А значит, и анимацию будем делать всего лишь ключом поворота одной управляющей кости. Чтобы создать анимацию, нужно зафиксировать ключ поворота в нулевом кадре и значение 360 градусов в кадре конца анимации на ваш выбор. Мы можем настраивать график всего для одной кости и перемещать ее, а также изменять как нам угодно, тем самым создавая сложную анимацию минимумом ключей.

ПРИМЕР. ПАРАЛЛАКС ОКРУЖЕНИЯ

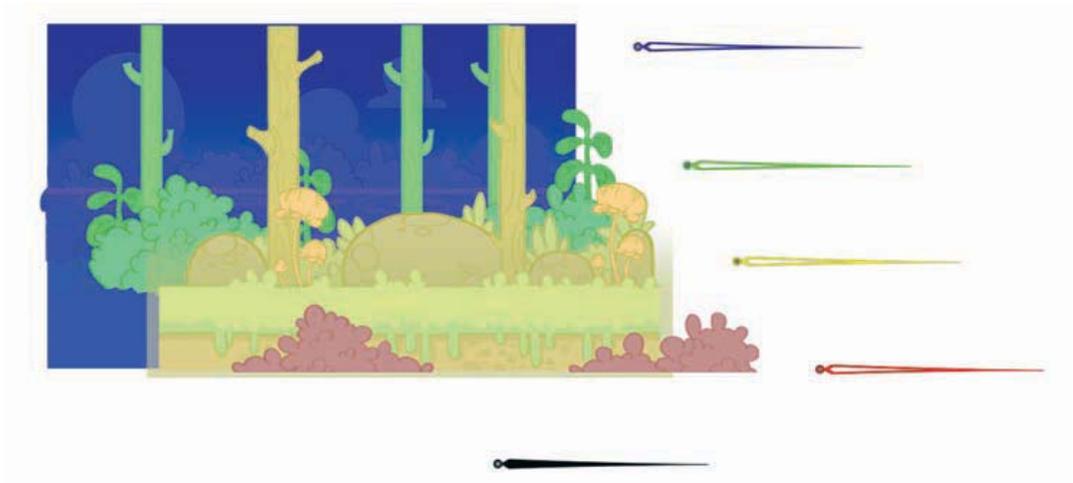
Ограничитель трансформы отлично подойдет для создания анимации фона. С помощью этого инструмента мы сможем задавать различную скорость движения планов кадра. А управлять смещением планов будем все так же всего одной костью, что в разы упростит данную анимацию. Для примера возьмем локацию с передним, средним, дальним планами фонов и элементами на них. Всего у нас четыре плана, которые будут двигаться. Группируем изображения и каждой группе назначаем свою кость, которая будет отвечать за движение. Выделяем их разным цветом и так же окрашиваем планы, чтобы было понятно, какая кость за что отвечает.



Создаем отдельную управляющую кость и ограничитель трансформы для каждой кости планов. Также для каждой кости не забываем нажать в настройках ограничителя кнопку «Сопоставить».



Теперь для задания разной скорости движения планов настраиваем значения в поле «Преобразовать». Чем ближе план, тем быстрее скорость движения, а значит, тем выше значение в этом поле. Если нужно задать значение больше 100%, то вписываем его вручную, а не настраиваем бегунком. Для планов, близких к зрителю, как раз лучше выставить больше 100. Я поставлю 120 для переднего плана. Далее визуально оцениваем разницу скоростей. Разница еще зависит от того, на каком расстоянии находятся планы логически на иллюстрации. В комнате разница скоростей будет намного меньше, чем на горном пейзаже. Потому что расстояние между объектами в помещении намного меньше, чем в природе. Я выставила значения 120, 70, 40, 10, соответственно, от переднего плана к дальнему. Теперь, двигая управляющую кость, мы будем наблюдать эффект параллакса.



Чтобы картинка не прерывалась, нам нужно определить границы кадра маской и задать постоянное копирование картинок в планах. Нужно добиться того, чтобы движение создавало единый цикл. Это можно делать уже в движке с помощью простого скрипта. Но для этого нужны минимальные навыки программирования. Такой эффект можно сделать и полностью в Spine, но это довольно неудобно и долго. Нам придется создавать копии изображений и включать-выключать их при выходе за кадр. Нам сложно будет контролировать этот эффект и менять его параметры. Поэтому этот момент все-таки лучше реализовывать в игровом движке или программе, заточенной под работу с камерой.

ГЛАВА 11

Эффект псевдо-3D

Настройка

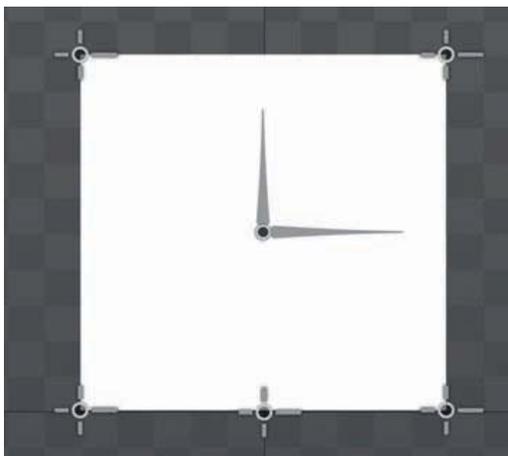
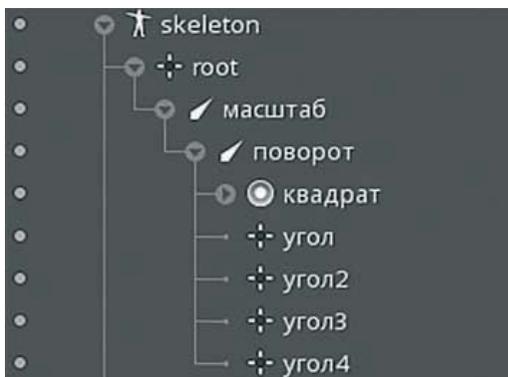
Исходное изображение нам потребуется всего одно. Нужен белый квадрат, который мы перенесем в проект. Вся суть приема псевдо-3D раскроется в настройке.

По центру квадрата создаем кость, которая будет отвечать за масштаб. Изначально создаем кость примерно по центру. Затем дополнительно выбираем слот с изображением и выставляем нулевые значения в полях инструмента преобразования. Для удобства эту кость лучше повернуть вверх на 90 градусов, чтобы кости хорошо читались на изображении.

Создаем кость, дочернюю по отношению к кости масштаба, также привязанную к изображению квадрата. Нам нужно, чтобы угол поворота кости был нулевой. Для этого нажимаем на кнопки «кости» и «изображения» в нижней части окна программы. И затем выбираем кость и выставляем значение 0 в поле поворота кости.

Создаем еще четыре кости по углам квадрата. Все четыре кости должны быть дочерними от центральной кости. Каждый раз при создании новой кости выбираем центральную в качестве родительской.

Теперь попробуйте изменить масштаб квадрата через кость масштаба, а после поворачивать кость поворота. Обратите внимание, что поворот работает корректно для искаженного изображения. Этот прием можно использовать не только для псевдо-3D. Например, он подойдет для создания колеса в перспективе или лопастей мельницы в том случае, если

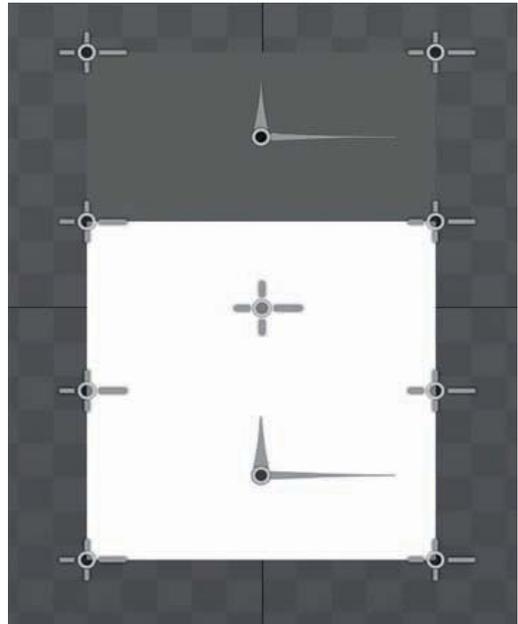


нам понадобится вращение с искажением. Выставляем масштаб по оси Y 0,5. Это будет верхняя грань куба. Сразу окрашиваем эту грань в определенный цвет. Выделяем слот и поле выбора цвета под структурой проекта внизу. Там можно настроить любое окрашивание грани. Лучше окрасить грани в разные цвета для наглядности 3D-эффекта.

Теперь создаем новую грань. Выделяем кость *root* и вытаскиваем квадрат из папки «Изображения» в структуре проекта. Просто перетягиваем квадрат снова на рабочую область. Это будет передняя грань куба. Совмещаем верхнюю часть новой грани с нижней частью верхней. Они должны состыковаться между собой и образовать ребро куба.

Выбираем всю верхнюю грань в структуре проекта нажатием на кость масштаба. Делаем ее копию сочетанием клавиш Ctrl + D. И перемещаем вниз, чтобы нижняя часть совместилась с нижней линией переднего квадрата. Это будет нижняя грань куба. Передняя грань должна находиться ближе всех к зрителю, чтобы нижняя грань по умолчанию была не видна. Регулируем порядок ячеек в структуре проекта.

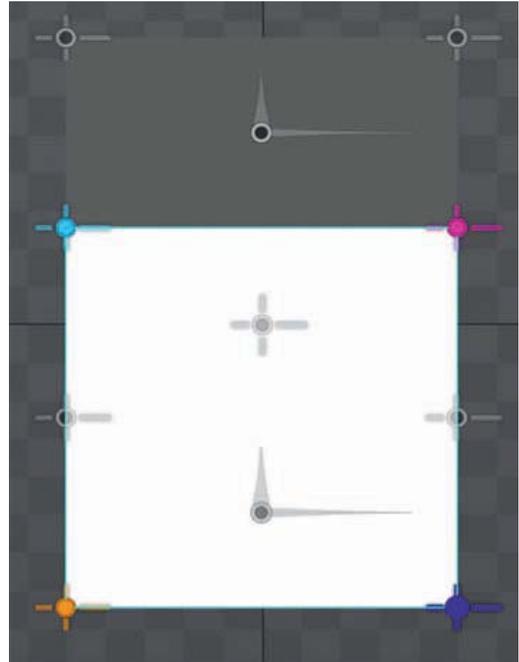
Далее нам нужно поставить масштаб нижнего слота (не кости!) –1. Выбираем слот нижней грани в структуре проекта и изменяем значение масштаба. Для работы с отрицательным значением масштаба необходимо поставить одну важную галочку в настройках программы. Нам нужно, чтобы при отрицательном значении масштаба спрайты не отображались в анимации. Если мы не поставим галочку, то нам придется включать и отключать грани куба вручную в моменты анимации, когда они будут появляться или исчезать из зоны видимости. Открываем настройки программы (меню Spine в верхнем левом углу) и нажимаем нижний пункт «Настройки». В окне настроек выбираем вкладку «Окно проекции» из списка слева. В самом низу настроек окна проекции находим опцию «Тыльная сортировка» и ставим возле нее галочку. Эта опция должна быть включена.



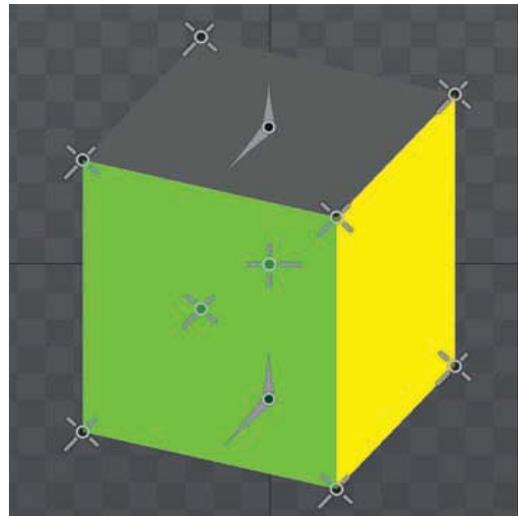
Теперь наша задача — связать переднюю грань с остальными. Для этого создаем стандартный меш для переднего квадрата. Ничего не редактируем, просто создаем меш с четырьмя точками на углах квадрата. Создаем четыре копии этого меша. Лучше, как и в случае с гранями куба, задать разные цвета для слотов. Так будет удобнее ориентироваться в структуре. Цвет для слота задается при выборе слота внизу под структурой проекта.

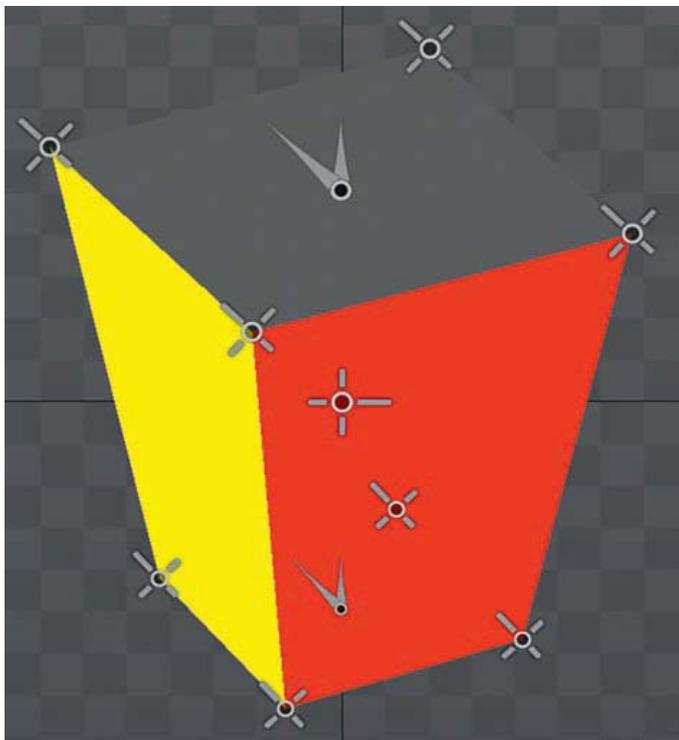
Привязываем первую копию меша к костям, которые располагаются на углах белого квадрата (эти кости привязаны к другим спрайтам). Веса распределятся автоматически, главное, чтобы на каждом углу было стопроцентное влияние кости. Если автоматически получилось не так и в настройке есть небольшое влияние на угол меша соседних костей, уберите его. Чтобы уже на этом этапе проверить эффект псевдо-3D, попробуйте выделить обе кости поворота (те, что вытянуты горизонтально) и покрутить их. Куб должен крутиться влево-вправо с эффектом 3D, но пока без боковых граней.

Теперь сделаем боковые грани куба. Выбираем обе кости поворота и в значении поворота указываем 90 градусов. По сути, мы разворачиваем куб боковой гранью вперед к зрителю. Но там у нас пусто. Включаем любой из скопированных ранее слотов с передней гранью квадрата. И теперь также привязываем меш этого квадрата к костям, которые в этом положении находятся по углам фигуры. Не забываем сделать вес кости на углу 100% без дополнительного влияния соседних костей. Таким образом мы задаем одну из боковых граней. Далее проделываем то же самое для значений поворота 180 и 270 градусов. То есть мы назначаем меш на грани по очереди. Теперь опять попробуйте покрутить кости поворота, куб должен крутиться уже со всеми существующими гранями.



При управлении масштабом мы можем регулировать ракурс куба, открывая верхнюю или нижнюю грани. А еще с данной настройкой мы можем сделать из куба параллелепипед, перемещая верхнюю или нижнюю кость масштаба. Эффект 3D будет работать и для измененного куба тоже. Мы даже можем изменить перспективу куба или сделать из него пирамиду. Для этого нужно отрегулировать масштаб одной из костей масштаба на верхней или нижней грани по одной или обеим осям.





Уменьшение масштаба кости масштаба нижней грани куба и перемещение одной из костей

Анимация

Начнем с анимации поворота куба. В целом с такой настройкой можно делать любую анимацию с эффектом 3D. Так как мы сделали все грани куба, мы можем сделать и прыжок куба, и перемещение с поворотом. В целом для более серьезных объектов эффект 3D выполняется по подобной логике, но его настройка гораздо сложнее. Мы всегда должны иметь под рукой появляющиеся грани объекта при изменении ракурса, например, бок, спину и грудь персонажа, и периодически показывать их в анимации. Правда, в случае с персонажем и любым сложным объектом мы вряд ли сможем поворачивать его на все 360 градусов. Это все-таки уже не куб, кроме того, у объекта может быть множество деталей и сложная форма граней и ребер. Поэтому эффект псевдо-3D на сложных объектах обычно ограничивается небольшим поворотом.

Выбираем кости поворота и локальную ось координат. Мы выбираем локальную ось, так как в абсолютной шкале поворот ограничен значением 360, а 0 и 360 считаются одним и тем же положением объекта. Это значит, что при задании поворота от 0 до 360 градусов анимация происходить не будет. А в локальной шкале можно задавать значения и 360 градусов, и более 360 градусов, и анимация осуществляться

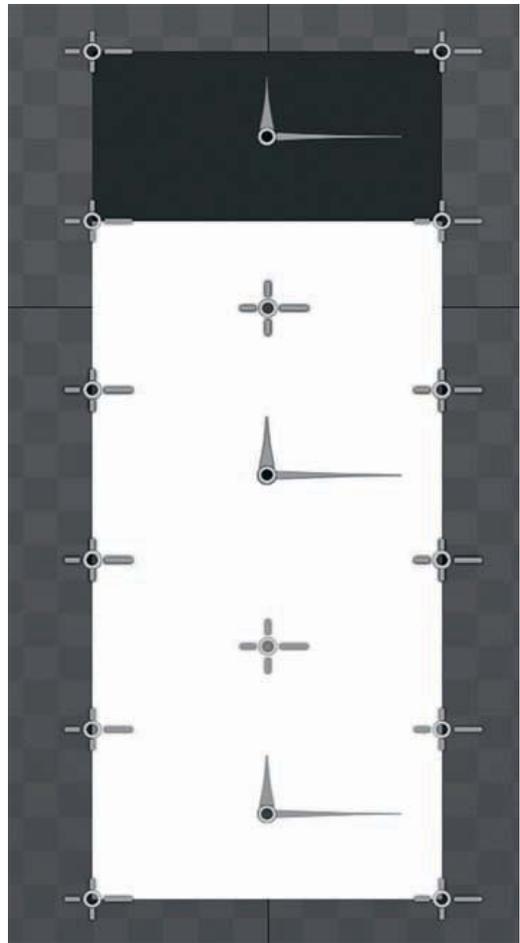
будет. Поворот по часовой стрелке задается углом со знаком минус, а против часовой — со знаком плюс. В нулевом ключе фиксируем значение 0 в поле поворота. А в 80-м кадре ставим 360 градусов. Таким образом мы создаем постоянное зацикленное вращение куба. Можете поэкспериментировать с изменением масштаба верхней и нижней грани или вытягиванием куба по осям. Это даст интересные форматы фигур. А также вы увидите, что вращение даже при искажении сохраняется. Главное — не забывайте каждый раз создавать дубликат анимации, когда собираетесь что-то радикально изменить.

Изменение формы

Подскажу довольно интересный формат эксперимента с данной настройкой. Итак, создаем кость, дочернюю от кости root, и помещаем в нее всю нашу структуру куба. Дублируем получившуюся структуру. Программа спрашивает, сохранить ли ключи для кости при дублировании? Нажимаем «да». Этот вопрос задается в случае, если вы уже сделали какую-либо анимацию, а значит, для дублируемой кости в анимации есть заданные ключи. И при копировании эти ключи можно сохранить или скопировать с пустой шкалой времени.

Далее опускаем вниз новый куб до того момента, пока его верхние кости поворота и масштаба не совпадут с нижними костями старого куба. Мы, по сути, совмещаем верхнюю грань нового куба с нижней гранью старого.

Если запустить анимацию, два куба будут крутиться синхронно. Такой вытянутый параллелепипед мы могли получить и простым растягиванием куба по вертикали. Но в данном случае у нас появились новые кости и грани для настройки формы. Попробуем их изменить, чтобы создать более сложную фигуру.



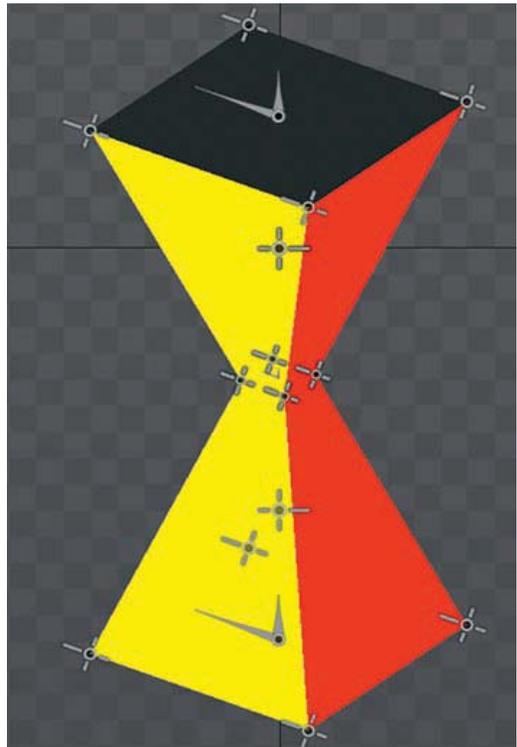
Мы должны выделить кости масштаба нашей средней грани. Поскольку у нас их две друг на друге, выделить щелчком их будет затруднительно. Поэтому отключим возможность выбирать картинки в нижней части рабочей области. Рамкой выделим кости. Увеличим масштаб этих костей по горизонтальной оси. При увеличении масштаба все пройдет гладко, и наш параллелепипед распухнет посередине. А вот при сужении появится некоторый артефакт. Грани нижнего куба выйдут на передний план, а так быть не должно. В данном случае нам нужно поменять порядок слоев. В пункте «Порядок отрисовки» в структуре проекта просто перетащим изображения граней нижнего куба под изображения верхнего.

Также в случае изменения масштаба по горизонтали нужно скорректировать масштаб по вертикали, чтобы промежуточная грань была правильной формы. Для этого необходимо посмотреть на соотношение сторон у верхней или нижней грани и выставить для средней такое же. Иначе фигура получится кривой, а не симметричной.

Изменение ракурса во время анимации

Чтобы поменять ракурс, нам нужно выбрать все кости масштаба (те, что вытянуты вертикально на примере). При изменении масштаба вертикальной оси верхняя грань будет уменьшаться, а нижняя увеличиваться пропорционально уменьшению верхней. Это создаст иллюзию смены ракурса в 3D-пространстве.

В анимации поворота устанавливаем значения масштаба трех костей. В нашем примере мы устанавливаем значения в 0, 80 и 160-м кадрах. Предварительно увеличиваем длину анимации поворота вдвое до 160 кадров, чтобы поворот и изменение ракурса происходили с разной скоростью. Итак, в 0-м кадре оставляем исходный масштаб, у нас это 0,5 и 1. В 80-м кадре мы устанавливаем 0,5 и 1. И в 160-й кадр копируем значения масштаба из 0-го. Если мы работаем с измененной формой, то для



измененных граней устанавливаем пропорциональные значения. Также при работе с измененной формой (в нашем случае с формой песочных часов) может возникнуть ситуация, когда при смене ракурса грани неверно располагаются по сортировке. И снова мы получаем грани, которые выходят на передний план, хотя должны быть на заднем с точки зрения логики ракурса. В таком случае мы должны менять порядок отрисовки прямо во время анимации. Делается это следующим образом. Выбираем кадр анимации, в котором происходит смена ракурса. В структуре проекта в пункте «Порядок отрисовки» перетаскиваем слои в нужное нам положение. Ключ в выбранном кадре, соответствующий новому порядку отрисовки, создается автоматически. Затем при повторной смене ракурса нужно будет снова перетащить слои в структуре проекта в обратное положение. Ключ будет называться «Порядок рисования». Как видите, на шкале времени мы можем создавать ключи не только для инструментов, но и для некоторых других параметров.



Изменение положения и поворот объекта

Создаем еще одну новую кость, дочернюю от корневой *root*. Всю структуру помещаем в эту кость. Мы могли бы работать и с корневой костью, но нужно привыкать этого не делать, так как для игровой анимации мы не можем менять корневую кость. Если вы планируете использовать анимацию в игре, то должны выполнять это правило.

Теперь для этой новой кости сделаем также анимацию поворота. Выставляем в 0-м кадре значение 0. А в 160-м кадре значение 360. Если вы не меняете длительность анимации и оставляете те же 160 кадров, то можете работать всегда в одной анимации. А если меняете, то должны создать разные анимации и назначить их на разные дорожки в предпросмотре.

Также для этой кости можно менять значения перемещения и масштаба, тем самым меняя объект целиком во время анимации. Вообще для данной настройки имеется очень много различных вариантов использования. Мы можем создавать сложные формы. Можем менять масштаб отдельных граней во время анимации. Можем добавлять побочные анимации в эффект 3D, создавая комбинации движений. Когда мы работаем с простыми геометрическими формами, 3D-эффект очень вариативен и гибок в настройке и анимировании.

ГЛАВА 12

**Работа
с покадровой
анимацией**

Для работы с кадрами в Spine есть свой набор инструментов. Рисовать кадры в программе нельзя, но использовать уже подготовленные исходники можно. Это делается для того, чтобы добавлять в анимацию покадровые эффекты или даже целые сегменты анимации, нарисованные покадрово. Возьмем для примера анимацию осьминога, состоящую из 26 кадров. У нас есть 26 отдельных изображений, которые мы перенесем в проект. Сначала разберем, как работать с переключением кадров напрямую, без специальных инструментов.

Все изображения, которые мы планируем переключать, должны лежать в одном слоте в структуре проекта. Сначала перенесем их на рабочую область. Выделяем все строки в пункте «Изображения» и переносим в центр окна программы. Затем выделяем их уже в окне и ставим значения преобразования, равные нулю. В этом случае все они становятся друг на друга в одно и то же место на рабочей области. Это работает, если на самих изображениях персонаж или объект находятся на одном месте. То есть у нас картинки одного размера, и объект на них расположен в нужном положении. Если это условие не выполняется, нам придется каждый кадр поправлять и подгонять вручную.

При перемещении каждое изображение кладется в собственный слот. Нам нужно выделить их и положить в один. Чтобы выделить все изображения, не кликая по каждому из них, воспользуемся фильтрами отображения. Отобразив в структуре только изображения, мы сможем выбрать их все разом, не затрагивая их слоты. Лишние оставшиеся слоты можем удалить, они нам не пригодятся.

Теперь, чтобы создать покадровую анимацию, нам нужно поочередно в каждом кадре включать нужное изображение. В первом — первое, во втором — второе и т. д. Учитывая, что у нас 26 кадров, мы проделаем это действие 26 раз. Это долгая и муторная работа. А кадров может быть и больше. Да и ключей получится много, по ключу на каждый кадр как минимум. А если переключений больше, чем кадров, то и ключей будет больше. В таком проекте сложно ориентироваться. Но для небольшого количества изображений такой способ подходит. Или если изображений мало, а переключать их нужно часто. Таким образом, мы получаем обычную покадровую анимацию в Spine. Но плюс в том, что мы можем сочетать ее со скелетной в одном проекте. Это позволит использовать плюсы и скелетной, и покадровой анимации.

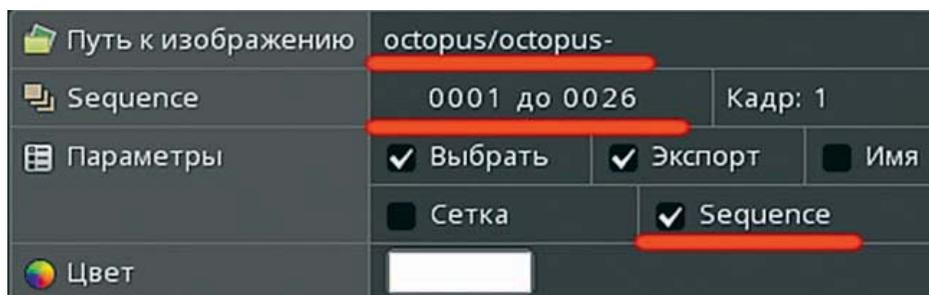
Созданную покадровую анимацию мы можем ускорять и замедлять. Это делается за счет увеличения промежутка между кадрами. А значит, каждый кадр будет виден дольше. Само собой, таким образом нельзя бесконечно замедлять анимацию, так как

для плавности движения в определенный момент потребуются новые промежуточные кадры. Иначе стоящий на месте кадр будет бросаться в глаза. Но все-таки мы не ограничены изначально заданной скоростью анимации. Для того чтобы изменить длину анимации, нужно выделить рамкой все ключи и потянуть за край рамки. Чтобы ключи могли встать в промежуточные значения между кадрами, нужно зажать клавишу Shift. Тогда ключи смогут находиться в дробных значениях кадров. В покадровой анимации это может пригодиться для того, чтобы сохранять расстояние между ключами, меняя только общую длительность анимации.

Секвенции

Для более удобной работы с покадровой анимацией в Spine добавлен инструмент секвенции. Переносим в рабочую область только одно изображение. Выделив само изображение в структуре проекта (не слот), ставим галочку возле пункта «Sequence» в настройках в нижней части окна под структурой проекта. В поле «Путь к изображению» мы должны вписать путь относительно проекта Spine. Лучше всего, чтобы изображения лежали в той же папке или в папке проекта, сгруппированные в отдельную папку. Изображения должны быть с одинаковым названием, но с припиской к основному названию цифры или последовательности цифр. В поле «Путь к изображению» мы как раз вписываем именно основное название — все, что присутствует в имени файлов до цифр. В нашем случае изображения лежат в папке проекта, сгруппированные в папку «octopus», и называются «octopus-(и далее цифры)». В поле пути мы вписываем «octopus/octopus-» и таким образом задаем путь сразу ко всей последовательности кадров. В поле «Sequence» указываем цифровые значения кадров. В нашем случае после названия стоят по четыре цифры. От 0001 до 0026. Если у вас просто цифры без нулей, то указываем их. Мы можем указать не все кадры, а создать несколько секвенций с разными включенными в них кадрами.

В поле «Кадр» мы выбираем кадр по умолчанию.



Переходим к анимации. При переходе в панель анимации окно настройки секвенции меняется. Появляется строчка для работы именно с анимацией.



Первое поле содержит в себе выпадающее меню. Оно отвечает за способ проигрывания анимации. «Hold» — анимация не проигрывается, показывается выбранный кадр. «Once» — анимация проигрывается один раз. «Цикл» — анимация постоянно повторяется. «Pingpong» — анимация проигрывается от первого кадра к последнему, а затем, наоборот, от последнего к первому. И последовательность всегда будет такой — как цикл, но не от первого к последнему и снова от первого к последнему, а туда-обратно. «Once reverse» — анимация проигрывается один раз от последнего кадра до первого. «Loop reverse» — цикл, но от последнего до первого кадра. И «Pingpong reverse» работает аналогично простому «Pingpong», но меняет направление проигрывания.

Второй пункт содержит значение кадров в секунду. Это скорость нашей анимации. И последний пункт — кадр, с которого начинается анимация. Выставляем значения: цикл, 30 кадров в секунду, кадр начала 1.

Теперь выбираем 0-й кадр анимации и ставим ключ секвенции, нажав на иконку ключика в настройках. Затем в 25-м кадре еще раз ставим ключ секвенции. Делаем это в 25-м кадре, так как у нас всего 26 кадров, но начинаем мы с 0-го, поэтому последним будет 25-й. Поставив первый ключ в 0-м кадре, мы уже задали анимацию по умолчанию.

Поэтому можно просто вести бегунок анимации по кадрам и смотреть в настройках секвенции номер кадра в поле «Кадр». Когда появится последний кадр, в этом месте анимации поставим второй ключ секвенции. И на этом все! Мы задали всего лишь два ключа, а покадровая анимация проигрывается сама с заданными настройками. Так намного удобнее, чем создавать множество ключей для переключения изображений. Проект не захламлен ключами, и мы можем легко менять тип воспроизведения анимации. Между двумя ключами у нас на шкале образуется линия. Маленькие штришки показывают переключения между кадрами. А большие штрихи показывают циклы анимации.



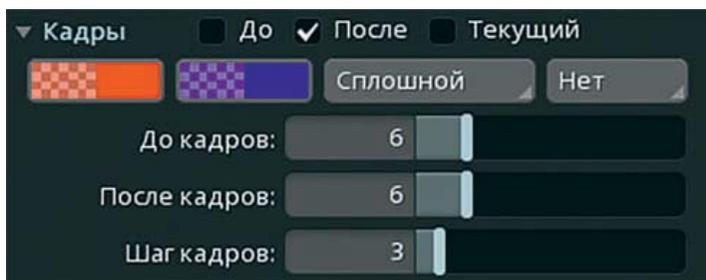
ГЛАВА 13

Просмотр кадров и дуг. Окно «Ореол» (Ghosting)

Для оценки правильности анимации и ее корректировки нам желательно видеть несколько кадров сразу и дуги, по которым двигаются элементы. Глядя на несколько кадров одновременно, мы видим динамику движения и можем скорректировать анимацию, наблюдая изменения. При стандартном просмотре мы можем отслеживать, как происходит изменение формы и положения объекта только кадр за кадром. А с помощью окна «Ореол» можно отобразить на рабочей области сразу несколько кадров движения.

В таком формате просмотра мы также можем изучать траектории движения, даже если делали движение не через пути, а просто задавали ключи. Такой тип предпросмотра поможет и при создании дополнительных изображений для портфолио, для иллюстрации покадрового изменения объекта.

При работе с анимацией на вкладке «Анимировать» в меню «Вид» нужно выбрать пункт «Ореол». Откроется специальное окно, добавляющее элементы на область просмотра в окне программы.



Галочки, проставленные возле «До» и «После», как раз позволяют показывать на рабочей области определенное количество кадров до и/или после текущего выбранного. Мы можем задавать цвет, которым они будут отображаться. Для удобства лучше выбирать разные цвета, чтобы разделять кадры по областям. Ниже мы можем задавать количество кадров, которые будут отображаться. Поскольку Spine создает очень много промежуточных кадров, нет необходимости показывать их все. В таком случае никакой наглядности не будет, все превратится в кашу из кадров, накладывающихся друг на друга. Для того чтобы отображать кадры с определенным интервалом, мы должны задать «Шаг кадров». Число в этом поле означает, что через каждые N кадров кадр будет показываться на рабочей области. Мы можем задавать как целые числа, так и дробные. Spine позволяет нам отображать хоть каждый 0,1-й кадр. Это все равно что создать практически цельную траекторию из изображений.

При использовании окна «Ореол» на шкале времени появляется полоска в верхней части, которая показывает, сколько кадров мы отображаем ореолом.

Галочка, проставленная в поле «Текущий», используется довольно редко. С ее помощью можно отобразить ореол от текущего кадра, если сам текущий кадр будет изменен. Рядом с выбором цвета также есть тип отображения, при нажатии на кнопку которого выпадает меню с несколькими пунктами. «Сплошной» означает заливку силуэтов заданным цветом. Детализация изображения в таком случае не сохраняется, виден только силуэт. Функция «Изображения», наоборот, сохраняет детализацию, лишь подкрашивая исходную картинку в заданный цвет. Лучше всего картинка подкрашивается, если она светлая. На темных изображениях окрашивание будет видно слабо, а на черном не видно совсем. «Нет» — отображение заливки отсутствует. Это меню относится к отображению заливки и изображения.

Правее есть еще одно выпадающее меню, с помощью которого можно настраивать отображение силуэта элемента и его обводки. Иногда это выглядит нагляднее, чем отображение полностью окрашенного изображения. «Силуэт» отображает обводку заданного цвета по силуэту объекта. Настройки толщины обводки в программе нет, а по умолчанию обводка довольно тонкая. Отображение «Силуэт» обводит силуэт целиком, объединяя изображения в одну плашку. А вот второй тип отображения обводки «Рентген» обводит каждый сегмент отдельно.

При таком отображении отлично видно наложение элементов исходника друг на друга. Мы наглядно видим места пересечений изображений.



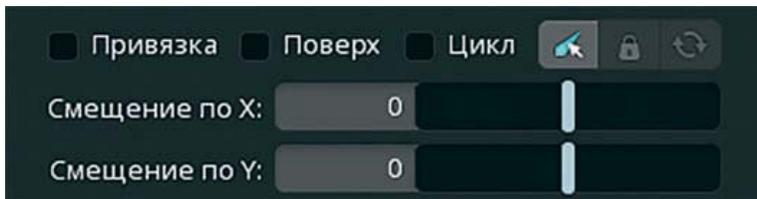
Разница отображения «Рентген» и «Силуэт»

Окно «Ореол» делится на несколько окон. Первое, которое мы рассмотрели ранее, это отображение кадров. Ниже располагается аналогичное окно под названием «Ключи». Настройки в нем практически полностью совпадают с предыдущим верхним окном

«Кадры». Разница в том, что окно «Ключи» отображает только те кадры, в которых мы задали ключи на шкале времени.

Ниже находится окно аналогичного вида «Вектор движения». В этом окне мы можем настраивать отображение направления движения. На рабочей области появятся стрелки, отображающие направление перемещения объектов. Стрелки рисуются для каждой точки меша и изображения. Количеством кадров мы задаем длину стрелки. Стрелка будет отображать положение точки, в которой она окажется через N кадров, или кадр, в котором она была N кадров назад. Пункт «Порог» задает минимальную длину стрелки, которая будет отображаться. С помощью этого мы можем отсекать отображение слабо движущихся изображений.

Рассмотрим самое нижнее окно с настройками, добавляющее дополнительные параметры к предыдущим настройкам.



Включенная галочкой настройка «Привязка» считает отображаемые кадры по заданным настройкам от начала анимации в 0-м кадре. Даже если мы меняем текущий кадр, заданные кадры для отображения остаются в изначальной позиции. Например, мы задали отображение каждого 5-го кадра после текущего. Без включенной галочки при перемещении текущего кадра из 0-го в 1-й все кадры отображения ореолов также сдвинутся на один. То есть будут отображаться кадры 1, 6, 11, 16 и т. д. А при включенной галочке кадры останутся на позициях 5, 10, 15 и т. д.

Пункт «Поверх» позволяет настраивать перекрытия наших обозначений. Нажав галочку возле него, вы сделаете так, что все обозначения будут отображаться поверх изображений.

Если не поставить галочку возле настройки «Цикл», то отображение ореолов закончится с окончанием анимации. Если галочку поставить, то отображение при приближении к концу анимации переместится на начало анимации по циклу.

Три кнопки правее позволяют отобразить ореолы только для выбранной кости и изображения, прикрепленного к ней. Первая кнопка с иконкой кости включает отображение только для выбранной кости. Кнопка с изображением замочка закрепляет выбранную кость, и что бы мы ни выбирали, отображаться будет заблокированная кость. Кнопка со значком «обновить» обновляет закрепление.

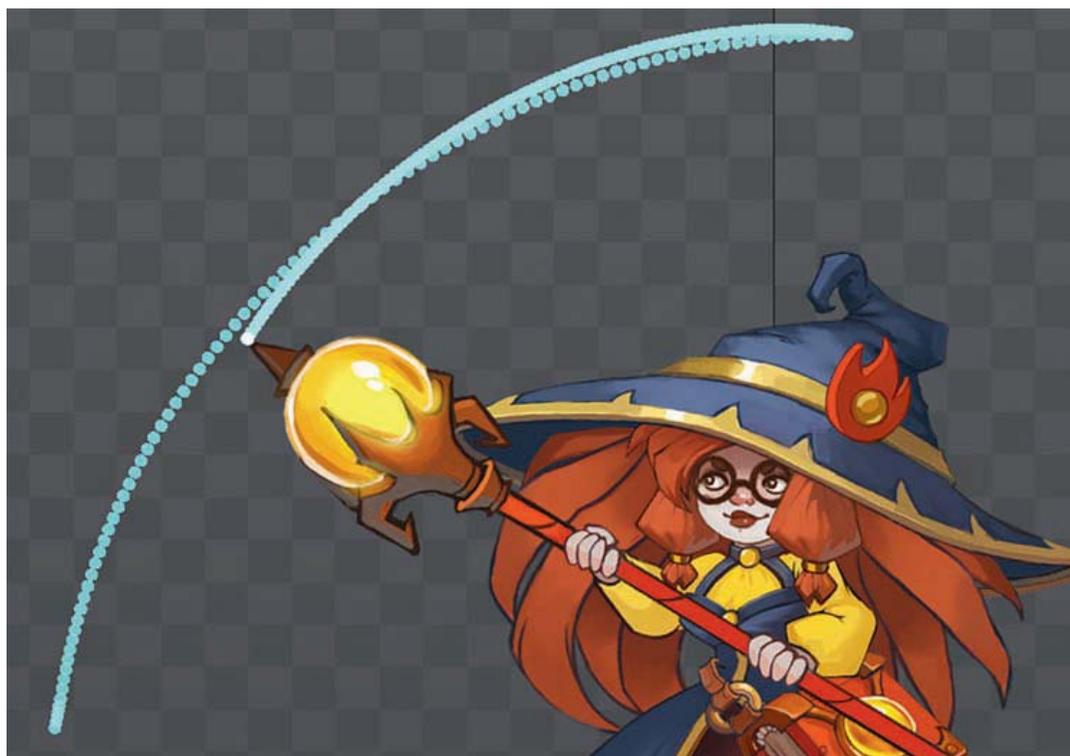
Кнопки «Смещение по оси X» и «Смещение по оси Y» позволяют смещать отображение ореолов по осям. Без смещения все будет отображаться в области движения объекта. И в таком случае может возникать множество наложений кадров друг на друга.

Так мы потеряем наглядность и информативность ореолов кадров. Сместив эти ореолы по осям, мы разделим их и добавим промежутки. А значит, наблюдать за ними будет удобнее.

Отслеживаем дуги

При стандартной работе с ореолами мы видим кадры, но они сливаются друг с другом. Траекторию конкретной точки проследить взглядом довольно сложно. Для того чтобы отследить конкретную точку движения, нужно немного поколдовать с настройкой проекта и окном «Ореол».

Для отслеживания дуг через окно «Ореол» нам понадобится дополнительное изображение. Лучше всего использовать простую геометрическую форму: круг или квадрат. Поместим это изображение в точку, траекторию которой мы хотим наблюдать. Также в структуре проекта это изображение должно быть прикреплено к кости, которая задает траекторию. Причем если уже есть прикрепленные к этой кости изображения, то нужно создать новую пустую кость и туда уже положить наш кружочек для отслеживания траектории.



Настроим отображение только отдельной кости, к которой прикреплен наш кружочек. Интервал отображения кадров должен быть меньше 1, я ставлю 0,1. Это нужно, чтобы траектория отображалась линией, а не отдельными кружочками с пробелами. Количество отображаемых кадров также ставим побольше, чтобы покрыть всю траекторию отслеживаемого движения.

Возьмем для примера мага с посохом, удар которого мы хотим отследить. На кончике посоха создаем кость, дочернюю от крайней кости посоха. Новая кость нужна, так как у нас нет пустой кости, все кости заняты изображениями. К новой кости прикрепляем кружочек, который размещаем на кончике посоха. Именно это положение и будет задавать траекторию для отслеживания. Кость лучше закрепить, чтобы не сбить траекторию переключением между костями. После установки настроек, озвученных ранее, мы получаем траекторию взмаха посоха, отображаемую с помощью окна «Ореол».

ГЛАВА 14

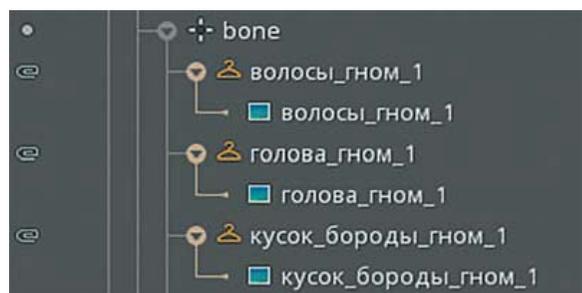
СКИНЫ

Довольно распространена ситуация, когда объект или персонаж обладают одинаковыми пропорциями или даже содержат одинаковые сегменты. Скелет у таких объектов одинаковый, меняются только изображения. Иногда нам вообще нужна группа одинаковых персонажей в разной одежде или просто разного цвета. В таких случаях, само собой, мы не будем каждому персонажу делать скелет с нуля. Да и копировать существующий в разные проекты тоже будет муторно. В таких задачах необходимо использовать скины. Это наборы изображений, которые мы можем настроить на один скелет. Их можно переключать, когда понадобится сменить персонажа или его часть. В оболочки можно даже добавлять кости, ограничители и все, что имеется в слоте. Это пригодится, если у одного персонажа есть определенный элемент, а у другого нет. А значит, у одного будут кости этого элемента, а у другого нет. В таком случае нужно добавить эти кости в оболочку персонажа. Есть еще вариант — конструктор. Мы можем не переключать цельные скины персонажей полностью, а создавать скин для каждого элемента персонажа и менять ему прически, сумки и другие элементы. Таким образом можно составлять персонажа из конструктора.

Рассмотрим для примера двух гномов в разной одежде. Лица и некоторые элементы полностью совпадают, но они окрашены в разные цвета. Импортируем нарезку обоих гномов в Spine. Привязываем группы изображений к разным костям, создав по кости на каждого гнома и положив все слоты с изображениями в соответствующие кости в структуре проекта.



Скины создаются в пункте «Оболочки» в структуре проекта. Нажимаем на пункт, и под структурой проекта появляется кнопка «Новое». Нажав на нее, мы можем создать новую оболочку. Создаем две оболочки, по одной на каждого гнома. Теперь мы должны все изображения разложить по соответствующим оболочкам. Для этого выбираем конкретное изображение и в меню «Новое» выбираем «Метка оболочки». Создается метка с символом вешалки, и изображение уже автоматически оказывается в ней. Или выбираем слот и также создаем метку оболочки. Но тогда необходимо содержимое слота вручную перетащить в метку. Этот вариант подойдет, если нам нужно положить в метку оболочки только определенное содержимое слота. Таким образом мы можем выбрать содержимое вручную.



Но кликать по каждому изображению долго и неудобно. Поэтому выделим все изображения первого гнома. Для удобства выбора картинок включаем фильтр отображения изображений и костей. Затем выбираем изображения и в меню «Новое» создаем метку оболочки. Для каждого изображения автоматически создается своя метка.

Мы можем такой же алгоритм использовать и для второго гнома. Но в таком случае нам придется создавать много меток оболочки, и у нас будет много слотов. Лучше сделать так, чтобы слот был один и метка оболочки тоже была одна, а все изображения конкретного элемента лежали внутри этого слота и метки. Переключаться они будут при переключении самих оболочек. То есть мы создадим слот «рука», в нем метку оболочки, и все руки разных персонажей будут лежать уже внутри. Так структура проекта станет более лаконичной и понятной.

Для создания такой структуры нам нужно перенести изображения второго гнома в соответствующие метки и слоты. Сначала в обязательном порядке переключаемся на вторую оболочку в пункте «Оболочки». Только с включенной второй оболочкой переносим соответствующие изображения. Руку — в слот руки, лицо — в слот лица, и так далее для всех изображений. Если имеются изображения, которых нет в новом скине персонажа, то для них создаем новые метки оболочки. В нашем случае ноги первого гнома отсутствуют у второго гнома. Ноги второго закрыты юбкой, а значит, они не нужны нам в анимации. Для ног делаем свою метку оболочки, она создается при выбранной второй оболочке. Первая оболочка у нас для седого гнома, вторая — для рыжего. А значит, ноги будут только во второй оболочке. Теперь при переключении оболочек в пункте «Оболочки» картинки в рабочей области будут меняться согласно настройке меток.

Добавление костей в оболочки

Как я уже упоминала выше, в оболочки можно добавлять не только изображения. Давайте разберемся, как добавить дополнительные кости скелета в оболочку. В нашем случае у одного из гномов есть длинная борода, которой для анимации нужны несколько дополнительных костей. У рыжего гнома этих костей в скелете не будет, они будут только у седого. А значит, нам нужно добавить эти кости в структуру и оболочку только одного персонажа.

Для добавления в оболочку мы выделяем в иерархии нужную нам кость. После этого в нижней части под структурой нажимаем кнопку «Добавить к оболочке». Таким образом, мы добавляем кость к выделенной в данный момент оболочке. Но проблема может возникнуть, если у кости есть еще дочерние. Если мы их не выделим, они не включатся в оболочку автоматически и программа выдаст предупреждение в виде желтых иконок костей в иерархии. Чтобы все работало корректно, нам нужно выделить все

дочерние кости и также добавить их к оболочке. Добавленные к оболочке кости отобразятся дополнительно в иерархии самой оболочки, а не только в структуре костей.

Если ваши кости привязаны к пути с помощью ограничителя, нужно также добавить в оболочку все элементы этой конструкции. То есть кости, путь и ограничитель. Иначе программа снова выдаст предупреждение. Добавить ограничитель к оболочке можно, просто перетащив его в структуру в нужную оболочку. Если мы что-то упустим, программа будет работать некорректно.

Также кости и ограничители можно добавить следующим образом. Выделяем саму оболочку и внизу под структурой нажимаем кнопку «Добавить к оболочке». После этого в структуре проекта мы можем выбрать все нужные элементы, которые хотим добавить к данной оболочке.



Создание оболочек при экспорте из Photoshop

Если мы знаем заранее о необходимости создания набора оболочек, то можем сразу при экспорте из Photoshop сделать необходимые настройки. Применим все тот же скрипт PhotoshopToSpine. В скрипте заложена возможность использования тегов, с помощью которых мы автоматически создадим две оболочки для наших гномов и метки оболочки, необходимые для работы. Теги — это специальные команды в квадратных скобках, которые мы будем дописывать к названию слоев в начале. А скрипт автоматически распознает их и сделает настройку в Spine. Мы будем использовать тег `[skin: name]`. Полное описание всех тегов можно найти в описании скрипта на github, откуда вы его и скачаете. Дописывать тег к названиям слоев можно вручную, но если слоев очень много, то лучше также использовать специальный скрипт. Если слоев немного, можно обойтись и ручной работой. Главное — даже после использования скрипта проверить названия слоев, чтобы экспорт был верным. Скрипт пробивается по запросу Group Layer Renaming.

Итак, мы должны этот тег дописать к началу имен слоев. Сами слои, которые будут переключаться в оболочках, должны называться одинаково. Например, у нас есть два вида бороды и оба этих вида должны называться «борода». А в квадратных скобках

после двоеточия мы должны указать название оболочки. У одного слоя название оболочки будет «гном1», у другого — «гном2». Имена слоев выглядят следующим образом: [skin: гном1]борода, [skin: гном2]борода. Нужно пройтись по всем слоям и задать им одинаковые названия и теги перед ними с указанием нужной оболочки. Слои с одинаковым названием в результате экспорта попадут в один слот с данным названием. Чтобы Photoshop не ругался на одинаковые имена, можно сгруппировать слои. Когда слои в разных группах, одинаковые названия Photoshop не смущают.

После этого запустим стандартный экспорт в Spine. При открытии проекта в самом Spine у вас не будет отображаться никаких картинок на рабочей области. Не пугайтесь, просто по умолчанию оболочки отключены. Нужно включить одну из них, чтобы на рабочей области появились соответствующие изображения.

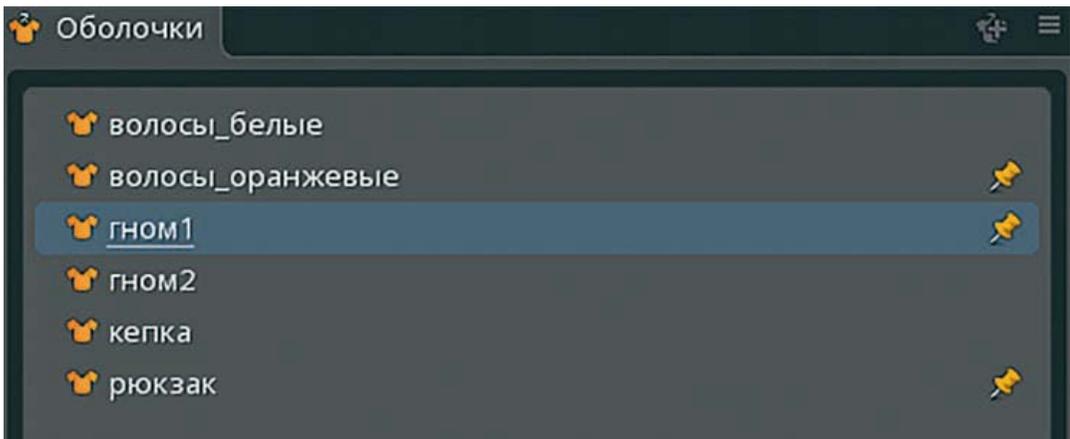
Конструктор оболочек

Допустим ситуацию, когда у нашего гнома есть кепка, волосы, собранные в хвост, и рюкзак. И нам нужно создать различные вариации гнома с этими элементами: только с хвостом, с хвостом и в кепке, без хвоста и в кепке, с рюкзаком и в кепке, но без хвоста, и т. д. Для такого конструктора в Spine есть отдельное окно работы с оболочками, в котором мы можем включать несколько оболочек сразу и тем самым создавать комбинации. Что-то вроде предпросмотра дорожек анимаций, только для оболочек.



Для каждого такого элемента конструктора мы должны создать свою персональную оболочку. Для кепки создаем оболочку «кепка», для рюкзака — «рюкзак». В общих оболочках персонажей этих элементов быть не должно.

Далее нам нужно открыть окно через меню «Вид — Оболочки». Откроется окно со списком всех созданных нами оболочек, которые мы можем включать и выключать.



Выделим основную оболочку персонажа и нажмем иконку булавки справа от ее названия. Тем самым мы закрепим эту оболочку. Далее мы можем также закрепить оболочки для создания конструктора. Эти булавки при включенном положении означают, что, какую бы оболочку мы ни выбрали, закрепленные оболочки останутся на месте. Мы можем закреплять определенные детали и, переключаясь между основными оболочками, смотреть, как деталь вклеивается в основную оболочку и анимацию.

Этот прием позволяет нам оценить корректность анимации для разных оболочек персонажа, проверить, правильно ли стыкуется анимация отдельных элементов с основной. Также, включив определенный набор деталей, мы можем экспортировать данный вариант персонажа в виде gif или видео. А вот в игру эта настройка конструктора не переходит. Мы экспортируем весь проект, а уже в движке снова настраиваем конструктор. Примерно так же, как и в этом окне.

ГЛАВА 15

Анимация персонажей

Анимация персонажей является обязательным пунктом при изучении скелетной анимации. Персонажная анимация составляет значительную часть от общих задач в игровых проектах. Анимировать героя, безусловно, сложнее, чем объекты или растительность, так как для создания достоверного движения важно понимание анатомии. Скелет персонажей зачастую сложнее скелета объектов интерфейса или природы. А помимо самого движения, нам важно создать характер персонажа с помощью анимации. Также для персонажей, представленных в проекте в крупном размере, нужна лицевая анимация. А это тот случай, для которого сложнее всего делать меши и эффект псевдо-3D.

Для того чтобы упростить изучение и понимание персонажной анимации, мы разделим всю тему на несколько пунктов. В каждом из них мы разберем основной алгоритм для задач персонажной анимации. Сначала рассмотрим принцип построения скелета для человекоподобного героя. От персонажа к персонажу скелет будет изменяться незначительно. У нас есть туловище, руки, ноги, голова. Да, может добавиться хвост или количество рук. Может измениться длина конечностей и пропорции скелета, но основная логика построения будет сохраняться. Затем мы рассмотрим основные моменты при создании типичных анимаций для персонажа в игре. Это бег, ходьба, прыжок, атака и анимация покоя. Умение создавать эти анимации уже обеспечит вам уверенное знание персонажной анимации и возможность работы в игровой индустрии. Отдельно поговорим о лицевой анимации и ее принципах. Другие задачи уже не требуют персонального внимания, так как вытекают из базовых. Умея делать базовые задачи и анимации, вы всегда сможете создать что-то дополнительное для вашего героя. И перед практическими примерами я расскажу о теории персонажной анимации, так как, в отличие от объекта, герой обладает характером, настроением и каждое его движение будет рассказывать целую историю.

Передача характера и настроения героя

В анимации персонажей важно всегда отталкиваться от описания героя и его характера. А также необходимо учитывать состояние и настроение персонажа в момент конкретного движения. У любой анимации нет четкого вида. Да, есть законы

и анатомические принципы. Но для разных персонажей анимация одного и того же движения может быть абсолютно разной. Например, анимация походки. Есть основа, без которой ходьбу сделать не получится. Персонаж двигает ногами и руками, переносит тело вперед. Но для толстого и худого героя скорость ходьбы будет разной, и шаги будут создаваться по разным принципам. Походка может быть семенящей, например, у женщины в узкой юбке и на каблуках. Герой может быть увлечен собственными мыслями, и тогда он будет идти широкими шагами, наклонив голову и даже корпус вперед. Анимацией походки можно показать настроение персонажа, подчеркнуть его визуальные особенности и создать характер, поэтому мы не можем ограничиться технической частью анимации. Просто двигая ноги и руки в нужное время и в нужном направлении, мы не создадим хорошей анимации. Только учитывая особенности персонажа, можно создать подходящую ему анимацию и сделать ее максимально выразительной.

Рассмотрим на примере, как особенности героя и его настроение влияют на анимацию. Допустим, у нас есть крупный, неповоротливый герой в доспехах и с тяжелым оружием. И в контраст ему — очень худой и быстрый почтальон с сумкой, наполненной письмами. Уже на этапе постановки задачи будет видна разница. В первом случае ходьба дается герою тяжело. Он тащит на себе и доспехи, и меч. Сам он, таская тяжести, стал крупным и уже не таким быстрым. Да и в целом управляться с тяжелым оружием на скорости невозможно. Нужно более детально показывать подготовку к замаху и момент после удара в анимации. До удара мы должны показать, как тяжело воину поднимать меч, иначе создастся впечатление, что меч из пенопласта. А после удара возвращать меч в исходное положение тоже довольно тяжело. Уделяя внимание подготовке к замаху и тому, что последует за ударом, мы показываем тяжесть оружия и одновременно силу героя. Двигаться такой персонаж будет также с большими задержками. При прыжке нужно уделить особое внимание приземлению. Персонаж тяжелый, а значит, довольно сильно ударится о землю. Можно добавить эффект пыли и показать, как после приземления герой упирается ногами в землю, чтобы снова встать. Так же и при ходьбе: герой не может бегать без труда в тяжелых доспехах и с огромным мечом. Каждый стук ноги о землю должен быть проработан для того, чтобы передать тяжесть движения и веса героя.

А теперь опишем походку почтальона. Он не воин, тяжесть сумки не сопоставима с тяжестью оружия. Его работа — быстро разносить письма. Он ходит, чуть-чуть подсакивая, практически переходя на бег. Быстро перебирая ногами, почтальон стремится вперед. Даже уже на уровне описания отличия в анимации этих двух персонажей заметны невооруженным глазом. Можно еще привести третий пример персонажа. Женщина на каблуках в вечернем узком наряде. Ее походка будет более элегантно. Особое внимание можно уделить пластике бедер. Широко расставлять ноги будет мешать одежда. Шажки будут короче, но если нам понадобится увеличить скорость ходьбы, то на единицу времени их будет больше. И это еще одна походка, которая не похожа на предыдущие две. Вспоминая разные типажи персонажей, можно составить целый список вариантов походки. Эти варианты схожи только основным движением. Сложность персонажной анимации состоит в том, что каждый раз мы делаем новую анимацию. Даже если это один и тот же тип движения.

Анимация требует особого подхода не только если это разные герои, но и если это один и тот же персонаж в разных состояниях. Грустная походка и веселая будут отличаться довольно сильно. Если персонаж устал, то это также отразится на его движениях. То есть, помимо описания персонажа, мы должны учитывать еще и ситуацию, в которой происходит действие. Необходимо создавать движения согласно эмоциям персонажа в этой ситуации.

Если мы создаем анимацию для игры, то подход все же больше стандартизируется. Подключаются уже требования жанра игры и вида игрового процесса. Нам все еще важно передать характер героя, но игровые ограничения выходят на передний план. Каким бы тяжелым и неповоротливым персонаж ни был, он должен хорошо управляться игроком. А значит, мы в первую очередь минимизируем задержки перед движением. Тут важно соблюдать баланс. Тяжелый персонаж не должен порхать как пушинка, но и в полной мере показать, насколько тяжело ему даются движения, мы не сможем. Ведь если этим персонажем управляет игрок, то после нажатия на кнопку персонаж должен незамедлительно реагировать.

Также в создании персонажной анимации для игры важно учитывать роль этого персонажа. Его игровое время и размер. По сути, мы должны определить, насколько данный персонаж важен. Для каждого персонажа из массовки мы не можем позволить себе делать детальные анимации по несколько штук. Это оптимизировано с точки зрения затрат на проект. А значит, им нужно создать такую анимацию, которая будет максимально универсальной для использования в разных ситуациях. Для более важных персонажей как раз включаются характер и эмоции в движения.

Создание скелета персонажа

Рассмотрим создание скелета на примере человекоподобного персонажа. Мы хорошо знаем, как работают наши кости. А если не знаем, то всегда можем проверить, посмотреть, насколько и каким образом выгибаются суставы и поворачиваются определенные части тела. Не стесняйтесь использовать зеркало при создании анимации персонажа. Ведь у вас всегда под рукой отличный референс. При создании скелета в Spine нам важно задать удобный контроль над знакомыми нам движениями. Также очень важно минимальными средствами обеспечить создание всех возможных для человеческого скелета анимаций.

СТРУКТУРА

Начнем с основы. На каждую крупную кость нашего тела нам нужно создать аналогичную в Spine. Пока мы не будем углубляться в более детальную настройку ступней

или кистей, она потребуется, только если персонаж довольно крупный и детализированный. В большинстве случаев для кисти и ступни хватает одной кости.

Для рук у нас по три кости: от плеча до локтя, от локтя до запястья и кисть. Аналогичная система и для ног: кость от таза до колена, от колена до голеностопа и отдельная кость для ступни. Ступню, как и кисть, даже без учета пальцев порой делят на две части — до пальцев и после. Это делается для того, чтобы показать сгиб носочка при ходьбе или добавить возможность сомкнуть кисть в кулак.

Количество костей для туловища зависит от длины тела персонажа и его стилизации. При сжатых пропорциях туловище персонажа зачастую можно передать одной-двумя костями максимум, добавив персонажу возможность небольшого изгиба спины. Но если пропорции реалистичны, то нам может понадобиться несколько костей для передачи движения позвоночника. Позвоночник не обладает одной или двумя точками сгиба. Это равномерно изгибающаяся структура. А значит, для ее реалистичной передачи мы будем делать цепочку костей. Но для маленьких персонажей в игре или для стилизованных чибби-героев детализация спины не нужна, так как анимация этой части практически не делается. А если нет движения, то и дополнительные кости нам в этом месте не нужны.

Так же и с шеей. Если шея персонажа длинная и требуется точная передача ее движений, то может потребоваться несколько костей в этом месте. Но в случае с шеей это будет максимум три кости, если мы не анимируем лебедя или жирафа.

От шеи мы создаем одну кость для головы. Голова — монолитный череп, вся анимация головы ограничивается лицевой анимацией. Ну, и в некоторых случаях делается анимация ушей, если персонаж является фэнтезийным героем с большими ушами или звероподобным, который может этими ушами шевелить.

От шеи до плеча иногда также нужна кость, например, для создания поднятия руки в анимации. Эта часть тела также будет подниматься, что необходимо показать. Это можно сделать или с помощью кости, или деформацией меша.



Базовый скелет для анимации человека

Стыки костей соответствуют положению суставов нашего тела. В случае позвоночника и шеи стыки располагаем равномерно, согласно количеству костей, так как выраженных суставов в этих местах нет.

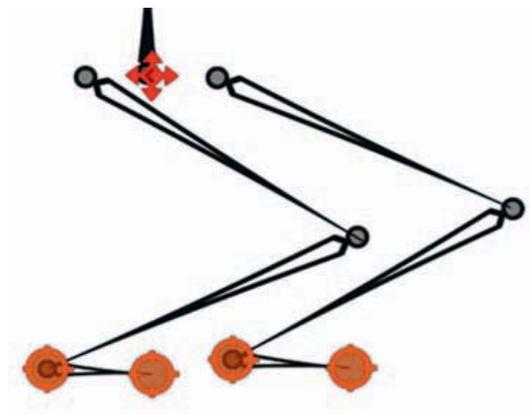
ИЕРАРХИЯ

Самая основная кость, от которой уже будут отходить остальные дочерние структуры, расположена в тазовой области ближе к паху. Назовем ее «центральной». Вся верхняя и вся нижняя части тела являются дочерними по отношению к ней, но между собой верх и низ тела не связаны. Мы можем двигать только ногами или только верхней частью тела обособленно. Вниз идут две дочерние структуры для ног, каждая последующая часть ноги является дочерней от предыдущей. От центральной кости вверх идет структура позвоночника. Каждая последующая кость является дочерней от предыдущей. От последней кости, которая заканчивается в начале шеи, влево и вправо создаем две независимые друг от друга дочерние структуры. Независимые, так как, когда мы двигаем одной рукой, вторая в этот момент может не двигаться. И еще одну независимую дочернюю структуру, обозначающую голову, создаем вверху. Каждая последующая кость руки является дочерней от предыдущей, а кость головы — дочерней от кости шеи. Вся структура костей прослеживается на нашем теле, и если вы в чем-то сомневаетесь, то всегда можете проверить на себе. Совершая определенное движение, смотрите, какие части тела это движение тянет за собой. Например, наклоняясь в сторону, мы сгибаем область позвоночника и полностью перемещаем всю часть выше сгиба, то есть руки, шею и голову. Поднимая ногу вверх, мы поворачиваем тазобедренный сустав, и за поворотом поднимается вся нога.

Центральная кость нам нужна для перемещения и вращения персонажа целиком. А также она понадобится при использовании инверсной кинематики, которую мы сейчас и настроим.

Контроль определенных частей тела осуществляется не родительскими, а дочерними костями. Например, для создания удара определенной конечностью удобнее двигать ступню или кисть. Поэтому нам нужно создать один или несколько контроллеров инверсной кинематики для каждой конечности.

Начнем с ног. Создадим ИК в точке пятки, и она будет контролировать две кости ноги, от пятки до колена и от колена до таза. Уже после создания двух таких ИК для каждой ноги создавать для



Инверсная кинематика для ног

персонажа некоторые действия будет намного проще. Например, если подвинуть центральную кость вниз, персонаж автоматически присядет, благодаря инверсной кинематике. Но если у героя есть ступни и кости для них, то для ступней нам также понадобится инверсная кинематика. Создадим еще по одной инверсной кинематике на носочках ступней. И привяжем ИК к кости ступни. Теперь ступня будет оставаться на месте при приседании. Также мы можем двигать саму инверсную кинематику, а за ней будет двигаться и нога.

Инверсная кинематика на руках создается аналогично ногам. Но использовать ее нужно только для тех анимаций, где она требуется. Если ИК рук не используется в анимации, ее лучше отключать. Не забывайте, что, согласно одному из основных принципов анимации, движения конечностей происходят по дугам. А значит, если мы задаем движение с помощью ИК, то лучше дополнительно использовать траекторию движения или контролировать дуги движений рук через окно «Ореол». При работе же с инверсной кинематикой напрямую можно совершить ошибку и создать прямолинейные движения вместо дуговых. Для управления ногами с помощью ИК это остается актуальным. Также инверсная кинематика может помочь, если конечность должна оставаться на месте при движении других элементов. Например, в позе «руки на поясе» у персонажа.

Рассмотрим настройку скелета на конкретном примере героя из нашей игры Almost My Floor.

Размещать кости нужно согласно конкретным изображениям в исходнике, так как у персонажа могут быть покатые плечи, или может быть не видно начала ног, или ступни могут быть в объемных ботинках. Такие моменты затрудняют определение расположения суставов. Плюс к тому иногда положение реального сустава слегка нарушается для того, чтобы картинка лучше двигалась и поворачивалась относительно друг друга. Поэтому необходимо постоянно проверять на практике расположение суставов при настройке, вращая элементы сразу при расстановке костей. Обычно точка вращения сустава, например плечевого, располагается примерно посередине окружности, вписанной в изображение в месте сустава. Не с самого края, а внутри. Это довольно распространенная ошибка — размещать точки вращения по краям изображений.

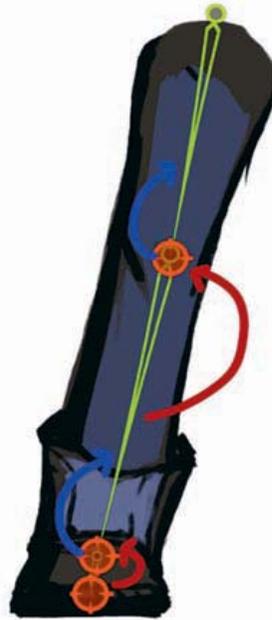


Но опять же, иногда нужно подогнать сустав под конкретные картинки, и может случиться так, что при точке, расположенной с краю, вращение будет лучше. Но в первую очередь мы должны ориентироваться на центр окружности, это стандартная ситуация. На нашем примере как раз плечи располагаются по принципу окружности.

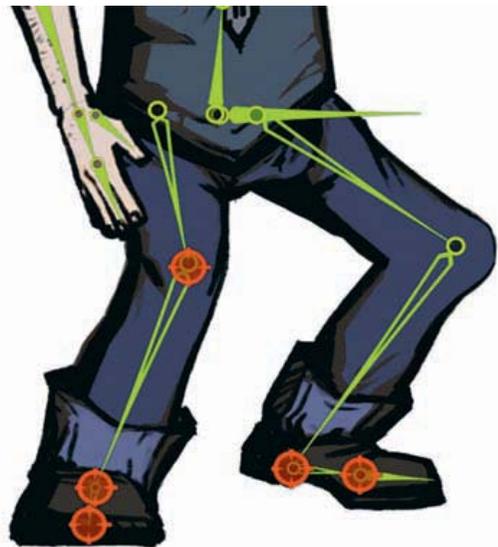
Инверсная кинематика руки будет удерживать ее в кармане при любом движении. Вторая рука в данном примере без ИК, так как произвольную анимацию дугами проще делать с помощью прямых поворотов.

На обеих ступнях у нас классический пример настройки инверсной кинематики. А вот на ноге, которая смотрит на зрителя, в области колена есть еще одна инверсная кинематика. Структура костей этой ноги также изменена относительно классической настройки. Этот прием используется для конечностей, которые смотрят на зрителя. Колено при анимации должно также выгибаться в сторону зрителя. А при классической настройке мы можем задать только изгиб вбок. При этом ступня все еще должна быть прикреплена к земле.

Структура родительских отношений между костями в данном случае изменена. Кость ступни является дочерней от кости инверсной кинематики в области пятки. Эта инверсная кинематика контролирует кость от колена до пятки. Кость от колена до пятки является дочерней по отношению к ИК колена. А ИК колена контролирует верхнюю кость ноги. Инверсные кинематики этой ноги не дочерние от скелета, они — дочерние от корневой кости проекта. Так как при движении



Красными стрелками показаны направления от дочерней кости к родительской. Синими стрелками показан контроль костей инверсной кинематикой



Пример приседания со сгибанием колена в сторону зрителя

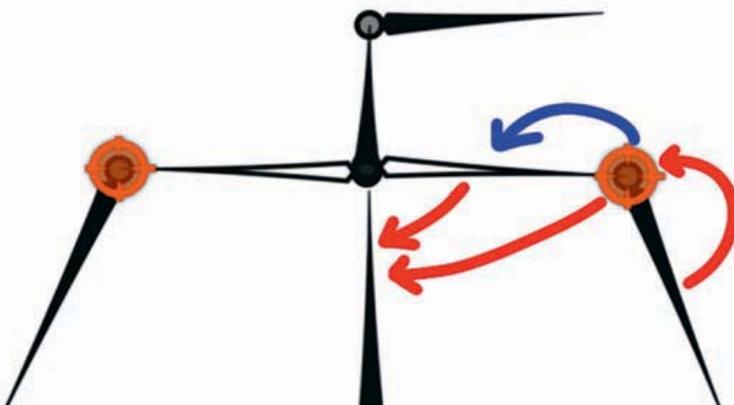
таза они двигаться не должны, мы будем настраивать их вручную. Как видите, небольшое отклонение позы героя от классического скелета уже дает определенные сложности и изменения в настройке.

Итак, сначала настраиваем приседание героя движением центральной тазовой кости вниз. Согнется только одна нога, направленная вбок. Ногу, направленную на зрителя, вручную подгоняем под положение скелета, двигая инверсную кинематику колена.

Псевдо-3D для туловища

Для более реалистичной анимации ходьбы или бега нам может потребоваться настройка для 3D-эффекта тела. Само собой, полный поворот туловища без дополнительных исходных материалов мы не сделаем, но слегка изменить ракурс сможем. При ходьбе или беге человек поочередно двигает плечами вперед-назад, и даже частью торса. А значит, при ракурсе сбоку или в 3/4 мы будем то в большей, то в меньшей степени видеть дальнее плечо, а ближнее будет периодически перекрывать тело. Аналогичная ситуация происходит и в области таза при движении ног. Для задания такого эффекта нам и нужна настройка 3D-эффекта.

Нам необходимо снова изменить структуру базового скелета и добавить инверсную кинематику. Инверсная кинематика ставится в точку плеча. Эта ИК контролирует кость от шеи до плеча (синяя стрелка на примере). Иерархия также немного меняется. Кость от плеча до локтя становится дочерней от ИК, а ИК и кость от шеи до плеча — дочерними от туловища. Аналогичную настройку нужно сделать и для второго плеча.



Изменение иерархии костей плеча

Две новые инверсные кинематики привязываем с помощью ограничителя трансформы к новой кости. Этой новой костью мы будем контролировать 3D-эффект в области

плеч. Далее нужно настроить параметры привязки. Нас интересует окно «Смешанное» в настройках ограничителя.



Дальнее плечо привязываем со значениями преобразования по обеим осям, равными 50. А ближнее со значениями -50 . В остальных полях ставим значение, равное нулю. Масштаб, поворот и сдвиг для привязки нам не нужны. Аналогичную настройку нужно создать и для области таза.



ГЛАВА 16

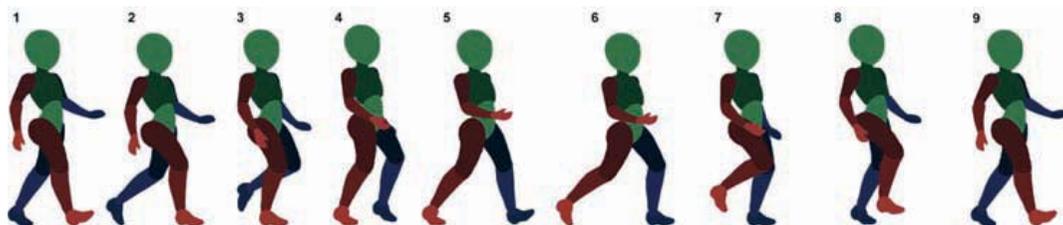
Основные анимации персонажа для игры

Ходьба

Основными анимациями для передвижения персонажа в большинстве игр являются ходьба и бег. Причем почти всегда есть и то и другое. При игре с геймпада используется прием, когда при малом смещении стика герой идет медленно, а при полном смещении — бежит. Если использовать только одну анимацию бега, то в замедленном виде это будет выглядеть странно. Поэтому, даже если герой по умолчанию всегда бежит в игре, ходьба создается как дополнительная анимация для медленного движения или постановочных сцен в игре. Ну а если герой по умолчанию ходит, а бежит при нажатии на определенную кнопку, то ходьба будет основной анимацией и ей нужно уделить максимум внимания.

Ходьба в некотором смысле создается даже сложнее, чем бег, из-за своей медленной скорости. При беге мы довольно быстро меняем положение тела и в промежуточных состояниях между крайними можно допускать определенные погрешности и недочеты. А ходьба требует проработки на всей своей продолжительности, так как мы отчетливо видим каждую стадию движения.

Для создания анимации ходьбы лучше использовать метод работы от ключевых кадров. Сначала нужно создать основные кадры, затем главные из промежуточных, а уже потом дорабатывать всю линию движения. Крайними точками в ходьбе будут противоположные положения рук и ног. Одна нога должна быть выставлена вперед, рука с этой же стороны отведена назад, а другая, наоборот, выставлена вперед. Также сначала нужно проработать именно руки и ноги, а уже потом анимировать побочные элементы на туловище, само туловище, одежду и голову. Первостепенная задача — создать основу движения. Причем сначала можно ограничиться только ногами, а затем подключить руки в противовес движению ног.



Разберем цикл ходьбы. 1, 5 и 9-й кадры как раз являются стартовыми положениями анимации. Одна нога касается поверхности пяткой, другая носочком, руки находятся в противофазе. 2-й и 6-й кадры, следующие за ними, показывают момент максимального сцепления с поверхностью. Персонаж максимально переносит свой вес на одну ногу. Ступня одной ноги полностью соприкасается с землей, а вторая нога практически отрывается от поверхности и уже не участвует в удержании веса героя. Это устойчивое, максимально приземленное положение героя. Эту позу реально удержать. Также эти кадры важны тем, что в них тело персонажа находится в самой нижней точке за весь цикл ходьбы. Можете обратить внимание на положение головы в цикле целиком. Голова не создает прямую линию траектории за цикл анимации. Во время ходьбы положение по вертикали меняется по траектории волны.

Кадры до первоначальных — 4-й и 8-й, наоборот, задают самое верхнее положение тела. Это кадры момента, который происходит перед переносом веса героя на одну ногу. Персонаж готовится навалиться вперед и поставить перед собой ногу. Это положение неустойчивое. Герой уже поднял одну ногу для переноса веса тела, и при этом вторая нога больше не удерживает вес полностью и касается поверхности только передней частью стопы. Возникает некое подвешенное состояние, стоять в таком положении затруднительно.

Оставшиеся кадры 3-й и 7-й — кадры смены позиции. Именно в них происходит максимальный захлест ног и переход от шага одной ногой к шагу второй. Эти позы максимально сжаты по горизонтали. Ноги даны внахлест, руки близки к положению «по швам». Эти кадры важно дорабатывать вручную, чтобы сохранить читаемость движения. Так как в этих кадрах много перекрытий, нужно постараться сохранить читаемый силуэт и позу. Автоматически программа может этого не учесть.

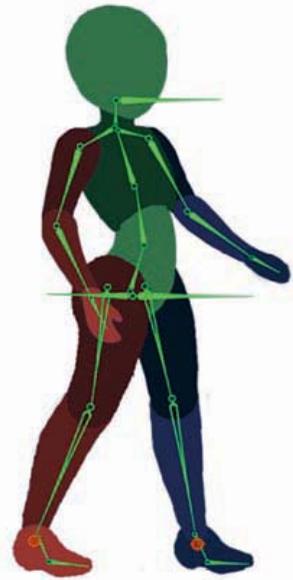
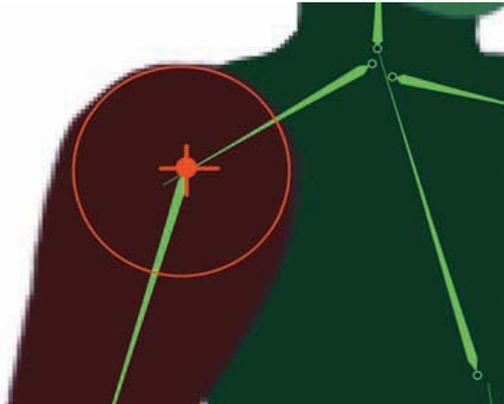
В процессе ходьбы мы двигаем плечами и областью таза в соответствии с движениями рук и ног. Этот эффект можно добавить с помощью псевдо-3D на теле, выдвигая одно плечо вперед, а другое пряча назад. И то же самое нужно проделать с областью таза. Также в процессе ходьбы персонаж может слегка покачивать головой в сторону наклона тела. Но такие нюансы уже нужно обрабатывать после основного движения.

РАБОТА С ХОДЬБОЙ В ПРОГРАММЕ SPINE 2D

Настройка скелета для нашего примера остается аналогичной рассмотренной ранее. Пропорции в примере не реалистичные, но и не сильно сжатые, поэтому в туловище мы сделаем три кости, чтобы иметь возможность менять изгиб спины. Для ходьбы это не так важно, но как я уже говорила, настройка персонажа должна быть универсальной.

Также на этом примере хорошо просматриваются места крепления костей и логика их расположения. Поскольку персонаж голый, силуэт не искажен одеждой, и мы можем поближе посмотреть на суставы. Мы должны визуальнo представить окружность в месте сустава и в центре этой окружности прикрепить кость для этого сустава. Такой

логики стоит придерживаться, даже если место сустава скрыто объемной одеждой или другими предметами. При приближении мы можем видеть некоторые неточности стыковки костей. Нет идеальной симметрии, и кончик одной кости не точно заходит на начало другой. В данном случае это не страшно и в исходном ракурсе никак не повлияет на анимацию.



Настройка скелета для базового примера персонажа

В Spine на шкале времени выставляем кадры по представленному выше циклу, начиная с основных. Промежутки между кадрами для начала можно оставлять одинаковыми, фиксированными. Таким образом, мы задаем новые позы через равные промежутки и у нас получается довольно плавная, равномерная ходьба. Это хорошо для тренировки и отработки навыков. И в целом зачастую обычной равномерной ходьбы бывает достаточно. Но в такой ходьбе нет характера персонажа, нет характера движения. Например, момент между зависанием в воздухе и переносом веса на одну ногу происходит быстрее, чем остальные, так как в это время персонаж будто падает вперед. Также в цикле ходьбы в областях столкновения с поверхностью (там, где основные кадры — 1, 5 и 9-й) движение происходит медленнее, чем смена ног и рук в промежуточных состояниях. Экспериментируя с различными промежутками времени между позами, можно получить походку разного характера.

Обратимся к покадровой анимации. Чтобы задавать скорость, мы оперируем количеством кадров и их плотностью. При создании ходьбы больше кадров рисуется как раз в месте соприкосновения стопы с поверхностью. А между этими положениями кадры проходят быстрее. На примере вы можете видеть этот эффект. Красным цветом показаны силуэты кадров, в месте соприкосновения площадь наложения и перекрытия больше, чем в промежутке. В Spine мы также можем регулировать плотность кадров и делать позы возле соприкосновения ноги и земли плотнее друг к другу на шкале времени. Плюс к тому в Spine для задания характера движения у нас есть графики, которыми мы также должны воспользоваться.



Пример распределения кадров из покадровой анимации

Нам нужно выставить замедление в начале и конце между 1-м и 5-м кадром, чтобы добиться той самой плотности кадров около моментов соприкосновения с землей. Затем, уже добавляя новые кадры, мы будем разбивать этот график на части. В Spine есть возможность регулировать график целиком. Поэтому мы всегда сможем скорректировать замедление в процессе анимации. Аналогично для второй части цикла анимации ходьбы с 5-го по 9-й кадр.

Есть еще второй способ работы с графиком. Он заключается в том, чтобы сначала выставить все в линейном формате, а уже по окончании работы с кадрами корректировать замедление в нужных моментах. Такой способ по-

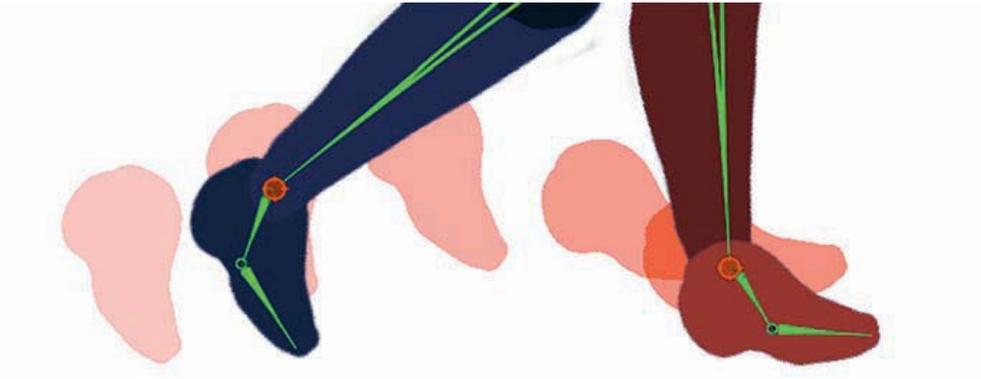
дойдет, если вы все-таки решите делать анимацию по принципу от кадра к кадру поочередно, вместо того чтобы сначала создать ключевые.

При работе с ходьбой и положением ног нужно выставлять галочку в пункте «Преобразовать» параметра «Отдельно» в настройках кости инверсной кинематики. Нам важно иметь возможность редактировать дугу движения, а значит, работать отдельно с перемещением по каждой оси.

После того как мы выставили положения рук и ног, важно не забывать поднимать и опускать персонажа в целом с помощью основной кости. Общий график для перемещения этой кости должен выглядеть волнообразно. Если график недостаточно плавный, то можно корректировать значения ключей.

В процессе создания анимации по принципу от кадра к кадру поочередно может возникнуть проблема неравномерного движения или скачков. Мы задаем положения, но не можем четко контролировать расстояния между ними и скорость перемещения определенных точек (например, точек ИК на ногах). Поэтому и важно сначала делать ключевые кадры, а потом промежуточные. Если мы будем создавать кадры поочередно от первого до последнего, то потом может потребоваться корректировка. Также при выставлении кадров рекомендую использовать меню «Ореол» для того, чтобы видеть уже созданные кадры и выставлять новые относительно них. В покадровой анимации кадры рисуются таким же образом. Мы видим уже нарисованные и между ними вставляем новые. В Spine же мы не рисуем кадры, а задаем положения костей, мешей и других элементов настройки. Но логика создания новых кадров остается такой же. С помощью меню «Ореол» мы можем наблюдать движение отдельных частей,

это очень крутая фишка Spine в сравнении с программами для покадровой анимации и аналогичной функцией отображения кадров в них. Там мы можем видеть только целиком нарисованный кадр, если он, конечно, не разделен на отдельные слои и элементы не нарисованы отдельно. А в Spine мы всегда можем отследить каждый отдельный кусочек нашего персонажа. Например, можно следить за кадрами ступни и корректировать ее положение.



Отображение кадров ступни

Когда я обучалась покадровой анимации, то для создания ходьбы использовала два способа. Можно создавать персонажа, идущего на одном месте, будто по беговой дорожке. А можно задавать движение персонажа вперед и делать ходьбу с движением вперед, а не на месте. Я советую попробовать оба варианта. Когда мы создаем анимацию ходьбы на месте, мы хорошо видим наложения кадров, можем легко отслеживать мелкие изменения частей. Создавать новые кадры проще, так как мы видим предыдущие прямо под нашими новыми. Но в таком случае мы недостаточно хорошо ощущаем сцепление персонажа с поверхностью. Он проскальзывает по земле ногами. Это может помешать созданию правильной работы ног. Если задавать движение вперед, мы будем четко ощущать вес персонажа и то, как движение ног влияет на шаг. Особенно при работе со ступнями для понимания их функционирования стоит попробовать сделать ходьбу вперед, а не на месте. Для игр нам понадобится именно анимация на месте, так как мы не знаем точной скорости движения героя. Но для обучения важно стараться делать анимацию с перемещением вперед. Да и в Spine анимацию с движением вперед мы всегда сможем превратить в анимацию движения на месте, и наоборот.

Анимация бега. Основные отличия от ходьбы

Анимация бега во многом строится по той же логике, что и ходьба. Персонаж все так же перемещает ноги и руки вперед и назад поочередно. Бег часто используется в играх в качестве основного вида передвижения. В большинстве случаев игроку нужно быстрее перемещаться по локации. Ходьба подойдет для пристального осмотра элементов локации, но в основном геймплее чаще всего используется бег. Поэтому можно сказать, что именно бег для нас, как для игровых аниматоров, является основным видом анимации для персонажа.

Основные отличия бега от ходьбы вытекают из скорости движения. Персонаж двигается быстрее за счет большего количества прикладываемых сил. Перемещение конечностей происходит быстрее, а сила отталкивания от земли больше, чем при ходьбе. Из-за скорости движения также добавляется определенное упрощение. Поскольку конечности быстро меняют положение, в первую очередь необходимо уделить внимание крайним точкам анимации бега — точкам столкновения ног с землей. В промежутках для многих задач можно не добавлять особой детализации, так как глаз все равно будет фокусироваться только на крайних кадрах. В самом упрощенном варианте вообще можно сделать только два кадра бега, которые соответствуют крайним положениям цикла. И при быстром их переключении уже будет создаваться ощущение, что персонаж бежит. В некоторых играх, где персонаж очень маленький или упрощенный по стилю, можно встретить и такой вариант анимации бега.

При создании анимации бега нужно уделить особое внимание моменту толчка ногой от земли из-за его силы. Еще большую важность приобретают замедление в месте столкновения и плотность создаваемых кадров в этой части анимации. Персонаж глубже приседает и работает не только ступней, но и коленом, и даже тазом. Особенно если бег очень быстрый, как в спортивном формате. Чтобы показать такие изменения конечностей, нужно выделить время на эту область анимации. Амплитуда волны движения персонажа по вертикали увеличивается не только вниз, за счет приседания, но и вверх, за счет сильного отталкивания. В процессе ходьбы персонаж всегда связан с землей, а в процессе бега герой может оторваться от поверхности и на некоторое время зависнуть в воздухе. При беге намного сильнее ощущаются движения персонажа вверх-вниз, поэтому не забывайте двигать героя целиком по вертикали основной костью движения. Правда, этот момент всегда нужно проверять в самой игре. Бег с сильными подскоками и приседаниями не всегда хорошо смотрится в игровом процессе. У игрока есть четкая ассоциация, что, когда он нажимает кнопку, отправляя персонажа вперед, тот движется вперед. Сильная волна по вертикали может нарушать восприятие контроля персонажа. Также эта волна должна соответствовать длине шага. Если персонаж бежит, быстро перебирая ногами, то и волна будет меньше. А вот если длина одного шага больше, то и подскок тоже

будет больше. Также учитывается длина самих ног. На маленьких, утрированно коротких ножках подскок будет меньше.

Руки во время бега согнуты в локтях, в отличие от ходьбы. Нам не нужно задавать волновое изменение положения костей для каждой руки. Важно сосредоточиться на движении плеч и рук целиком. Пластичность бега в основном достигается за счет ног, руки только отвечают за баланс персонажа при движении. Смещение плеч вперед-назад будет больше, чем при ходьбе, поэтому эффект псевдо-3D для бега более важен и применяется чаще.



Пример бега из игры *Potata: fairy flower*

Рассмотрим пример бега из нашей игры. Обратите внимание, насколько выразительна волна движения по вертикали. Наша героиня бежит подскоками, в игре нет задачи максимально быстро передвигаться вперед спортивным бегом. Поэтому волна по вертикали в таком случае обязательна. Более того, она подчеркивает некоторую легкость и воздушность бега. У бега, как и у любой анимации, есть характер. Для каждого персонажа и каждой ситуации он свой. И бег на скорость будет отличаться от легкой пробежки или бега подскоками.

Часть ног героини скрыта юбкой. Для бега это усложняющий читаемость момент. Ноги — основная область анимации бега. А если часть их скрыта, то нужно максимально утрировать положение открытых частей, чтобы силуэт в данном кадре читался. Поэтому в верхней точке у героини ноги максимально вытянуты в разные стороны. А в момент отталкивания одна нога вытянута, а вторая сильно поджата в колене. Эти положения довольно утрированы, и в реальности так бегать может быть сложно. Но в мультяшном стиле и сжатых пропорциях нужно всеми силами добиваться читаемости движения. Сжатые пропорции тоже могут усложнить жизнь аниматору. Короткими ногами создавать анимацию бега может быть сложнее. Руки также утрированно разводятся шире, чтобы силуэт лучше читался. С одной стороны, элементы, закрывающие конечности, мешают создавать читаемый силуэт. Но с другой стороны, позволяют нам допускать погрешности в анимации конечностей при их перекрытии. Например, в нижней точке, где ноги максимально скрыты юбкой. Нам не всегда важно полностью видеть силуэт, читаемости бега можно достигать и за счет движения персонажа целиком, а не отдельных его частей. Например, за счет утрирования волны по вертикали или пластики побочных элементов: волос, лент, одежды.

Настройки псевдо-3D в классическом понимании в данной анимации нет. Но этот эффект можно задавать и обычным перемещением элементов. Обратите внимание, что

ближние к зрителю рука и рукав интенсивно двигаются вперед и назад. Таким образом мы задаем движение плеча. Дальняя рука не так заметна в этом эффекте, но она тоже участвует в нем и прячется за торсом тогда, когда ближняя рука выставляется вперед. Такая возможность работы с руками достигается за счет ракурса и нарезки. У нас есть три изображения: рукав ближний, торс, рукав дальний. И в исходнике они расположены один за другим. При анимации мы просто перемещаем ближний и дальний рукава относительно торса и за счет перекрытий изображений задаем эффект 3D. Можно даже усилить эффект, если нарисовать отдельно переднюю часть торса героини и ее спину. Тогда можно будет еще сильнее выставлять ближний рукав вперед, подставляя изображение спины в моменты, когда она должна открываться зрителю.

Для создания читаемого движения, особенно при работе со сжатыми пропорциями и маленьким размером, важны побочные элементы. На нашем персонаже это юбка и волосы. Это пластичные элементы, которые будут повторять траекторию движения героини в целом. При подъеме вверх волосы прилипают к телу, а при падении спускаются шлейфом за персонажем. Этот эффект можно утрировать, чтобы подчеркнуть волнообразность движения. Также волосы могут помочь показать скорость бега, так как при быстром движении вперед они будут следовать шлейфом за героиней. То же самое и с юбкой. Оба элемента мы деформируем мешами.

Бег можно назвать максимально утрированной ходьбой. Так как логика движения схожа, скелет остается тот же самый, без дополнений. А вот амплитуда и искажения при движении становятся больше и заметнее. По сути, мы максимально утрируем заданные в ходьбе движения. Плюс к тому делаем поправки на скорость перемещения персонажа.

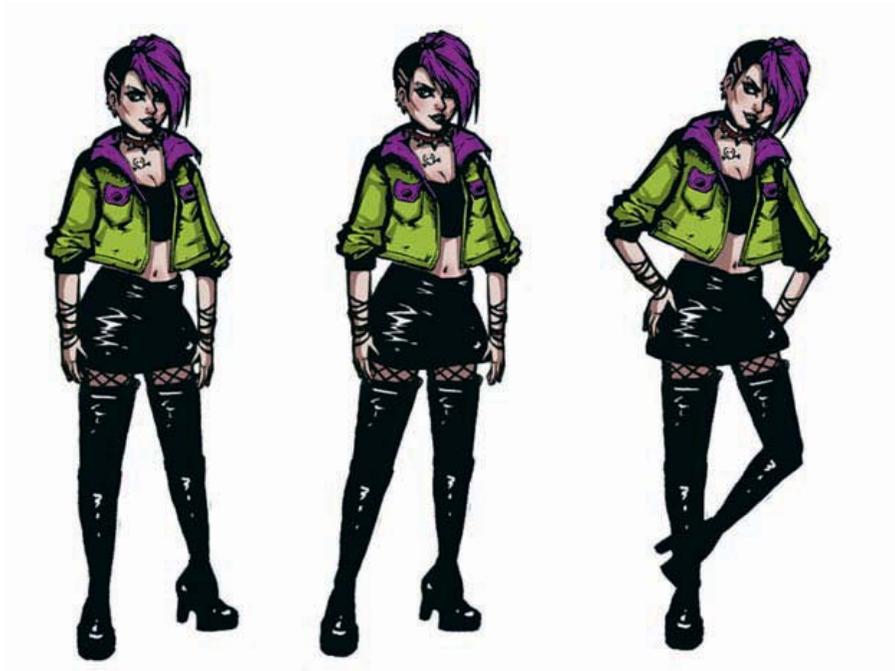
Анимация покоя

Классикой для анимации покоя в игре будет анимация дыхания персонажа и ветра вокруг него. Эта анимация не должна привлекать много внимания. Она создается для того, чтобы избежать эффекта картонной декорации. Если мы оставим спрайт персонажа статичным, то это как раз и будет привлекать излишнее внимание, так как персонажи и растительность практически не бывают полностью статичными. Ощущение дыхания персонажа не анимируется только поднятием груди или живота от вдоха. Я бы даже сказала, что эти движения, наоборот, игнорируются. Чаще используется легкое поднятие плеч, чтобы показать вдох. Лайфхак для создания анимации покоя через дыхание. Дыхание можно показать через увеличение грудного отдела или живота. Для этого мы должны масштабировать центральную часть тела целиком или только ее фрагмент с помощью меша. Но дочерние кости рук и головы масштабироваться вслед за родительской не должны. Если мы создадим отдельную кость для масштабирования, то руки вообще никак не будут реагировать на изменение тела. А вот если отключить костям, являющимся дочерними от туловища, наследование масштабирования,

то руки не будут увеличиваться и уменьшаться в размерах, но будут подчиняться анимации тела, оставаясь его частью.

Также персонаж в покое может слегка приседать. Это похоже на боевую стойку и состояние постоянной готовности к действию. Такой вид анимации покоя применим к активным персонажам: воинам, бегунам, ниндзя и т. д. Такие персонажи не расслабляются, останавливаясь на месте, поэтому им можно добавлять больше движений в покое. Также можно добавлять не просто приседание, а перенос веса с ноги на ногу. Герои могут поправлять меч на плече или в руке, оглядываться так, словно ищут противника взглядом. Можно даже добавлять минимальную разминку. Герой может потереть плечо или размять кисти рук.

Если мы не можем утрировать анимацию дыхания и персонаж не является воином, то для анимации покоя остается набор более спокойных движений. Одежду персонажа можно искажать мешем, имитируя ветер. Персонажу можно добавлять моргание и легкое изменение эмоций на лице. Также, если мы долго стоим, то из-за усталости невольно переносим вес с ноги на ногу. Но это уже не эффект боевой стойки, а более расслабленное движение, и оно производится реже. Еще персонаж может поменять положение рук, поправить волосы, почесаться.



Пример кадров для анимации покоя из игры *Almost my floor*

К примеру, для персонажа из нашей игры *Almost my floor* я добавила перенос веса с одной ноги на другую в анимации покоя. А также периодически героиня разминает ногу и меняет положение рук, похлопывая по юбке. Поскольку персонаж выполнен

в пропорциях, близких к реалистичным, я не могу добавить утрированное дыхание. Действие игры в основном происходит в помещении, а значит, эффект ветра, колышущего одежду на героине, тоже не подойдет. Остается только добавлять более реалистичные движения. Тут нужно вспомнить, что мы делаем, когда долго стоим. Перенос веса с одной ноги на другую, разминка, взгляд на часы, кручение чего-либо в руках, прикосновение к волосам — все эти движения можно наблюдать, глядя на себя в зеркало.

Поскольку я перечислила довольно много движений, расскажу о логике создания анимации покоя, которая содержит вариации. Вариации необходимы для естественного движения. Постоянно чешущийся персонаж будет выглядеть странно. Или герой, который одинаково поправляет волосы каждую секунду. Нам нужно смешать эти движения между собой. Изначально мы должны создать основу простейшей анимации покоя. Она будет содержать регулярно повторяющееся простое движение героя целиком. Покачивание, дыхание, приседание, ветер — эти движения можно использовать для основы анимации покоя.

Далее нам нужно создать несколько движений, которые будут происходить случайным образом. Мы уже в движении назначим время между анимациями для побочных движений покоя и создадим этакую солянку из анимаций. Причем какие-то движения будут производиться чаще, какие-то реже — таким образом мы сделаем персонажа более живым. Например, мы можем добавить порывы ветра, колыхая одежду сильнее. То самое разминание суставов или изменение положения частей тела пополнят список побочных анимаций. И не забудем про моргание, которое, кстати, тоже не происходит строго каждые три секунды. Время между морганиями может колебаться. И если мы зададим это в анимации персонажа, наш герой будет более естественным, а не похожим на робота, моргающего строго по часам.

Во многих играх есть специальная анимация покоя на случай долгого простоя персонажа. Герой может повернуться к игроку с вопросительным видом. Может присесть, сделав вид, что устал. Может топтать ногой в ожидании и поглядывать на часы. Это ломает стену между игроком и миром игры, но в определенном сеттинге часто используется и вполне актуально. Вы вряд ли встретите такой прием в реалистичной серьезной игре, но порой и там герой начинает жить своей жизнью, если его надолго оставить без управления. Он может начать осматривать окружение или даже что-то читать или писать.

В создании анимации покоя главное — контролировать амплитуду движений. Не забываем, что все-таки это анимация статичного положения. Да, в мультяшном стиле мы можем позволить себе больше. Вплоть до очень заметных приседаний в такт музыкальному сопровождению в игре, но это отдельные случаи. Такое возможно только в мультяшном стиле. Обычно пропорции персонажей там тоже не реалистичные. В остальных случаях герой должен все же стоять на месте, а не плясать вприсядку, если игрок не нажимает никаких кнопок пару секунд. Ощущение того, что персонаж остановился, не должно исчезать даже при добавлении множества побочных движений.

Рассмотрим пример анимации покоя для мультяшного чиби-персонажа в игре. Стиль игры очень утрированный, пропорции тоже. А значит, мы можем позволить себе кукольную анимацию покоя с сильным покачиванием, приседанием и дыханием.

Настройка скелета персонажа будет классической с небольшими отклонениями, в связи с особенностями дизайна героя. Например, при таком формате ног нам нужны кости для штанин отдельно, так как при анимации объемной одежды нам часто приходится поправлять ее форму и положение. При стандартных поворотах костей ног штаны также будут поворачиваться, но пластика ткани и форма учитываться не будут. Чтобы настраивать такие элементы более точно и обособленно от основного движения, нужно использовать отдельные кости.

Добавим инверсную кинематику не только на ноги, но и на голову. Герой должен следить за врагами или событиями вокруг. Управлять положением головы и направлением взгляда будет удобно с помощью ИК.

После настройки переходим к созданию анимации покоя. Условно разделим все тело персонажа на части: голову, туловище до границы футболки, штаны и ноги. Пока не будем обращать внимание на побочные элементы, их анимируем уже после основных. Поскольку персонаж мультяшный, мы можем смело анимировать его части, не боясь нереалистичности. Представим, что части тела закреплены пружинами и при каждом приседании колеблются вверх-вниз. Делать базу анимации можно просто перемещением частей тела сначала вниз, потом вверх с задержкой относительно друг друга. Так мы зададим эффект приседания и запаздывающих за основным движением частей тела. Можно усилить эффект, сжав и растянув персонажа целиком в нужных кадрах. С этим эффектом нужно быть аккуратнее, чтобы персонаж не стал резиновым. Но слегка добавить амплитуды приседанию через общий масштаб персонажа мы можем. Особенно учитывая стиль нашего исходника.



Настройка персонажа

После основного движения обратим внимание на побочные элементы. В первую очередь на челку. Она выступает на силуэт и является максимально эластичной. С ее помощью мы можем простыми средствами подчеркнуть движение. Вытягиваем ее при движении персонажа вниз и скручиваем при движении вверх. У персонажа есть еще побочные элементы: фонарь и топор. Фонарь максимально спрятан и слабо читается

в анимации. В анимации покоя его можно оставить без отдельного движения, просто совместив с телом. Топор выступает на силуэт, что добавляет ему читаемости. Но пластики в нем нет. Мы можем только слегка покачивать его вверх-вниз в добавление к основному движению. Когда персонаж движется вверх, топор долго остается внизу и с запаздыванием движется за телом. Аналогично при движении вниз.



Пример анимации покоя мультяшного персонажа

Анимация атаки

Анимация атаки бывает очень разной. Герой может бить мечом или кулаком, стрелять или наносить удары при помощи магии. Движения очень разнообразны, но у этих анимаций есть общие черты, которые мы и постараемся выделить.

Движение атаки зависит от того, что у персонажа в руке и как он этот объект держит. Оружие может быть одноручным или двуручным, и это сильно влияет на анимацию. При двуручном оружии — что бы это ни было: посох, пушка или меч — герой при движении больше задействует тело, так как хват двумя руками показывает тяжесть оружия. Чтобы его держать и двигать, нужно усилие. А руки и так максимально заняты удержанием оружия. Значит, для движения, атаки или компенсации отдачи в случае с пушкой мы будем больше двигать тело, чем сами руки. Если же оружие в одной руке, то большая часть движения совершается непосредственно рукой. Тело только добавляет амплитуды движению. Или телом мы удерживаем баланс, компенсируя силу удара рукой. Например, при ударе кинжалом герой может выставить руку вперед, а вместе с ней плечо и торс, чтобы добавить удару силы и амплитуды выпада.

Согласно принципам анимации, в анимации атаки очень важно разделение на стадии движения. Нужно уделить время замаху и окончанию удара, а также возвращению персонажа в исходное положение. Если атака особенно сильная, например, персонаж наносит суперудар, то он может даже какое-то время его заряжать, максимально растягивая момент перед нанесением урона. При ударе герой может выставить ногу вперед для компенсации сильного наклона тела в сторону атаки. Можно

добавить эффект удара о поверхность, если удар пришелся не по воздуху, а по твердому объекту.

Главное — помнить о принципах анимации и игровых ограничениях. Мы не можем позволить себе долгие задержки между ударами и между нажатием на кнопку и ударом. Важную роль играют темп игрового процесса и суть боевой системы игры. Если боевка очень быстрая и при этом стиль игры утрированный и упрощенный, то удар может быть показан только непосредственно движением. Иногда даже используется только эффект удара, шлейф от оружия, некий эффект рассекания воздуха. В играх особо активных в плане боевой системы этот эффект может перекрывать часть персонажа, и анимация самого оружия или конечностей создается очень условная или вообще отсутствует. Но если жанр игры файтинг или боевая система похожа, например, на *Dark Souls*, то задержки между ударами являются частью игрового процесса. Игрок должен понимать эти задержки и уметь учитывать их при атаке.

Если персонаж стреляет, то подготовки к движению как таковой нет, но зато есть заметный этап после выстрела — компенсация отдачи оружия. Этот этап нельзя опускать, так как именно он показывает вес оружия и импульс выстрела. Если не задать отдачу оружия, то оно будет выглядеть игрушечным. А выстрел не будет ощущаться опасным.



Для анимации атаки обычно создается минимум 2–3 варианта, чтобы разнообразить движение. Удары игрок будет наносить часто, и они будут идти подряд. Если не добавить вариации, то одинаковость движения будет бросаться в глаза. Создастся эффект «закликивания» врага. Будет казаться, что персонаж не дерется по-настоящему. Единичная анимация удара возможна при редких и длительных атаках, при мощных суперударах. Это актуально для жанров, геймплей которых не подразумевает

постоянную, повторяющуюся атаку. В нашем примере, который мы будем разбирать, персонажи атакуют по очереди. А значит, между каждой атакой каждого персонажа проходит достаточно времени. Можно ограничиться одним вариантом анимации. Плюс к тому некоторое время между атаками затрачивается на то, чтобы показать эффекты атаки и урона, полученного врагом.

В примере у нас два персонажа, оба с оружием. Но анимации атаки будут отличаться. Не только по принципу нанесения удара, но и по своему характеру. Несмотря на то что меч одноручный, а посох двуручный — посох легче. Маги в целом не славятся развитой мускулатурой. Даже если визуальное магическое оружие тяжелое, обычно считается, что волшебник держит его с помощью магических сил. Накачанный маг с огромным посохом — довольно редкое явление в игровой индустрии. А значит, анимация с посохом также более легкая и воздушная. Взмахи производятся по любой траектории и выполняются легко. Нет ощущения тяжести оружия и ограничений на движение. Персонаж крутит им вокруг себя, перемещает из руки в руку, создает различные формы траекторий взмахом. В конце движения маг делает выпад вперед и запускает заклинание. А вот с помощью энергии заклинания уже можно показать эффект тяжести управления железом. Когда посох наполняется энергией, его направление сложно удержать, и это можно показать анимацией. Персонаж может слегка трястись, пытаясь удержать посох. Его может обдуть ветром, словно от заклинания идет волна воздуха. Также персонаж может с помощью тела добавлять заклинанию импульс, словно стараясь усилить магический удар. Посоху не требуется сила, вложенная в удар, так как он всегда бьет в воздух, а по врагу бьет заклинание. Если маг бьет врага посохом словно палкой — это уже отдельная история. Если не нужно пробивать тело врага или любую прочную поверхность непосредственно оружием, то и силу в удар вкладывать незачем. Сила в энергии из железа, а не в ударе. Это еще одна причина легкости обращения с магическим оружием.



Атака мага поэтапно

У воина хоть и одноручное оружие, но ощущаться оно должно как более тяжелое, чем двуручный посох. Это не кинжал, которым легко махать из стороны в сторону. Одноручное оружие все еще относится к категории тяжелого. Поэтому в данном случае мы как раз наблюдаем максимально долгий замах, так как поднимать меч тяжело. Затем следует быстрый сильный удар вперед. Под собственной тяжестью меч быстро совершает движение удара. Плюс к этому воин задает направление и дополнительную

силу этого удара. Столкновение с врагами в играх с упрощенной стилистикой обычно не прорабатывается. Персонаж все еще бьет в воздух, но с силой. Столкновения происходят случайным образом, мы не можем точно знать, как соприкоснутся области оружия и врага. Это означает, что анимация удара может в любое время прерваться анимацией столкновения оружия с врагом. Это создает трудность для создания цельного движения, и такие ситуации прорабатываются только в жанрах, заточенных на сложную боевую систему.



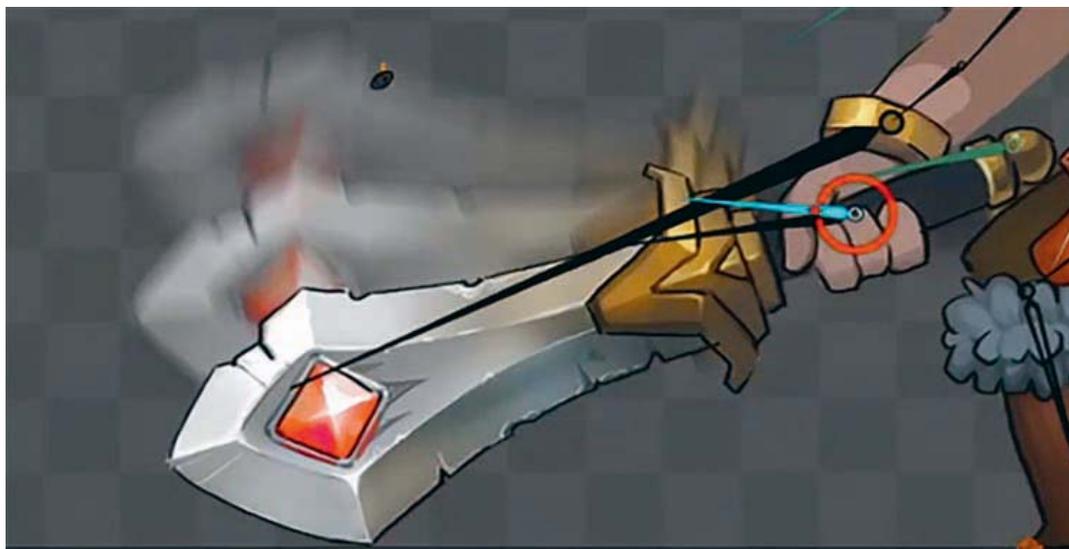
Атака воина поэтапно

Обратите внимание, как сильно отклоняется назад и вперед персонаж на примере. Это помогает увеличить амплитуду удара и акцентировать внимание на замахе и окончании атаки. Но нужно быть осторожными и не забывать о балансе стойки. Если персонаж слишком сильно замахнется, то под тяжестью оружия он упадет назад. В таком случае нужно поменять положение ног, поставив одну ногу назад, чтобы сохранить баланс персонажа и не дать ему упасть. То же самое и при выпадении вперед. Подчеркнуть скорость и силу атаки можно анимацией побочных пластичных элементов. Плащ, табард, волосы — все эти элементы максимально утрированно анимированы, чтобы за счет них на движении создавался большой акцент, особенно на местах, влияющих на силуэт персонажа.

Для создания эффекта большой скорости движения мы можем добавить размытый кадр оружия или эффект шлейфа от движения. Добавлять подобные штуки допускается как в Spine, так и в движке. В Spine в саму анимацию можно добавить размытые кадры оружия, которые будут следовать за основным. В каждом последующем размытом кадре оружия можно уменьшить прозрачность и увеличить размытие. Для создания такого эффекта существуют два способа. Первый — задействовать тот же самый спрайт, который у нас используется как основной, только размыть его и уменьшить прозрачность. И второй — нарисовать новые изображения для размытых кадров. Если мы будем рисовать новые картинки, то сможем настраивать форму размытия так, что оно будет происходить по траектории движения. Но и без новых кадров тоже получится интересный эффект. На примере с воином я добавила два кадра меча, просто размыв и понизив прозрачность спрайта.

В движке шлейф от движения можно делать через систему партиклов. Это симулятор движения частиц. Вид каждой частицы и принцип ее движения настраиваются в движке. Таким образом создаются различные эффекты: пыль, светлячки, молния. Но стиль

этих эффектов будет отличаться от покадрово отрисованных. Мы можем нарисовать в заданном стиле одну частицу (например, дыма или пыли), задать ей вариации и использовать в движке. Но даже так это не будет аналогом покадрового эффекта или эффекта, сделанного с помощью деформации в Spine.



Пример эффекта размытия при ударе оружием

Поэтому в данном вопросе лучше отталкиваться от стиля проекта. Если эффекты должны быть максимально похожими на классическую мультипликацию, то тут подойдет только покадровая анимация, и каждый кадр нам придется рисовать вручную. Эти кадры можно использовать с помощью секвенций в Spine, встроив в основную анимацию удара, либо добавить их уже в движке отдельной анимацией. Также система партиклов в движке больше подходит для создания случайных эффектов, таких как пыль, дым, туман, когда нам нужно произвольное, постоянно меняющееся движение. Для эффекта удара также можно использовать эту систему, но нужно строго задать область распространения частиц. В примере с воином я так и поступила: задала область шлейфа, а внутри нее запустила движущиеся партиклы.

Плюс такого подхода в том, что наш эффект настраиваемый. Используя один эффект, мы можем получить



Пример эффекта удара с помощью партиклов в движке

разные варианты, а также применить его для разного оружия, меняя ширину области. Если мы рисуем покадровый эффект или создаем эффект деформацией в Spine, менять его под другие задачи будет не так просто.

Анимация прыжка

Анимация прыжка встает в ряд необходимых анимаций сразу после ходьбы и бега. Есть игры, где персонаж может ходить, но не может прыгать. Это, к примеру, *point&click*-квесты. В таких играх персонаж не прыгает, следовательно, анимация прыжка в них не нужна, но зато нужно много анимаций реакции на события и взаимодействия с предметами. Также персонажи могут стоять на месте и только атаковать. Но в жанрах, близких к платформеру или файтингу, прыжок необходим. Освоить анимацию прыжка — важная задача для аниматора персонажей в играх.

У прыжка в игре есть особенность: мы не знаем его высоту и длину. Даже если персонаж просто прыгает с места вверх и у нас есть определенная заданная высота, то обычно у этой высоты есть вариации. В платформерах есть механика усиленного прыжка, сила которого зависит от длительности нажатия на кнопку. Если мы нажмем и сразу отпустим, то персонаж едва оторвется от земли. А вот удерживание даст более высокий прыжок. Это создаст ощущение усилия, прикладываемого для более высокого прыжка. Длина прыжка также может варьироваться. Помимо усиленного нажатия, на длину прыжка еще влияет разбег. В общем, мы пришли к тому, что оба параметра вариативны. А значит, в анимации нужно это учитывать.

Прыжок строго вверх и прыжок вбок отличаются друг от друга. Вполне может быть, что понадобится сделать вариации для разного вида прыжков. Прыгая в сторону, персонаж поджимает ногу, больше вытягивается вперед и руками работает в стороны. А при прыжке вверх персонаж вытягивается вверх, а руками работает по вертикали. Попробуйте прыгнуть вперед и вверх. Это будут абсолютно разные движения. Но поскольку нужно всегда стараться добиться хорошего результата минимальными усилиями и затратами, можно создать некую среднюю версию прыжка. Эта версия подойдет для обоих вариантов. В таком случае мы должны делать движения максимально нейтральными, не создавать сильных растяжений персонажа в определенную сторону. Движения конечностей будут универсальными. Например, можно сделать прыжок с подскоком, когда одна нога персонажа поджата, а вторая — вытянута. Таким образом мы можем прыгнуть и вверх, и вперед. Возможность создания общей анимации прыжка зависит от дизайнера и степени упрощенности персонажа, а также от его характера и описания. Старичок с палочкой подскоком вверх не прыгнет. А вот маленькая ведьмочка, да еще и в сжатых пропорциях, вполне. В своей игре *Potata: fairy flower* я как раз делала прыжок с небольшим поджатием одной ноги. И разных вариантов прыжка не потребовалось.

Несмотря на то что точную длину и высоту прыжка мы не знаем, эти параметры нам нужно понимать до создания анимации хотя бы примерно. В целом необходимо оценить прыгучесть персонажа для передачи его характера через анимацию. Также интенсивность усилия для прыжка зависит от требуемой высоты и длины. Если прыгать нужно дальше, то постараться придется побольше. Поэтому если в игре, к примеру, присутствует механика паркура и персонаж прыгает на максимально возможные для него расстояния, то и анимация будет соответствующей. Сильный рывок и усилие перед прыжком плюс смягчение приземления после. А если персонаж прыгает расслабленно, не на пике своих возможностей, то и особого усердия в анимации можно не задавать. Для понимания интенсивности прыжка нам нужно видеть игру, в которой мы делаем анимацию. Необходимо посмотреть на расстояния между платформами и понять основные механики игры и то, насколько сильно игра завязана на платформинг. Разделение анимации на части даст нам возможность регулировать расстояние между платформами в дизайне локаций, если понадобится. Но не в разы. Так как анимация все же по своей сути нацелена на определенные расстояния и высоты.

Создавать анимацию также может быть удобнее, задав в Spine длину прыжка. Потом мы ее уберем, просто удалив ключ перемещения перед делением анимации на части. Но работать над движением удобнее, когда видны все его параметры. В этом случае не придется представлять, что персонаж прыгает вперед, а на экране видеть только прыжок вверх и пытаться работать только с ним. А еще, добавив перемещение вперед, можно будет оценить дугу движения и то, как анимация выглядит в итоговом виде.

ПРИМЕР АНИМАЦИИ ПРЫЖКА

В качестве примера я покажу анимацию главной героини нашей игры *Potata: fairy flower*. В итоговом формате в игре анимация разбита на три части, но изначально она создавалась единым полотном. Начнем с обсуждения настройки героини. Настройка общая для бега, атаки, прыжка и других анимаций.

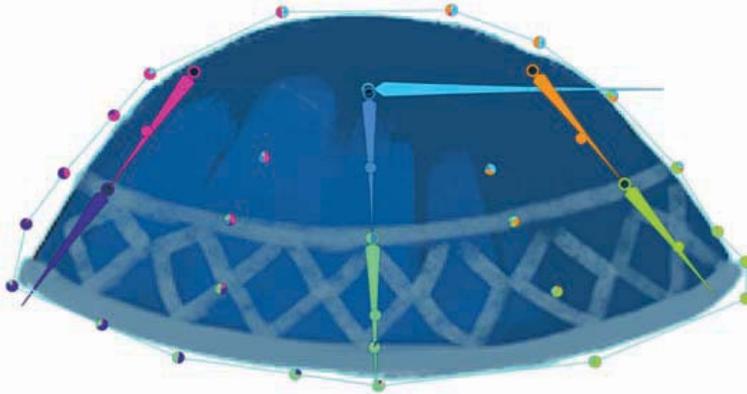
Как и в случае с бегом, читаемость анимации будет достигаться во многом за счет анимации волос и юбки. Ноги частично спрятаны под юбку, поэтому не так сильно выступают на силуэт, как могли бы без нее. Волосы у этого персонажа — основной маркер. За счет них мы считываем персонажа на игровой локацией.



Настройка персонажа из игры *Potata*

Движение персонажа мы также считываем, глядя на искажение прядей волос. Вообще у любого персонажа в концепте есть основные маркеры дизайна. И именно они привлекают взгляд зрителя. Поэтому в анимации порой стоит уделять больше внимания крупным и важным элементам дизайна, а не конечностям. Конечности могут быть очень утрированными, маленькими и частично скрытыми под одеждой. Поэтому их не так сильно видно в игре. Это не значит, что можно делать анимацию конечностей как попало, но детализация движения нужна не всегда.

В настройке скелета нашей героини обеспечим контроль над юбкой и волосами. Создадим несколько цепочек костей для изменения отдельных частей объекта. Также для юбки и волос мы сделаем меши, опять же, чтобы контроль был максимально точным. А волосы еще и разделим на части, как минимум на переднюю и заднюю области. Но можно отделить и каждую прядь, для того чтобы создать перекрытие одной поверх другой. В этом случае волосы будут выглядеть более реалистично, несмотря на мультяшный стиль.



Меш для юбки героини

Если требуется общий прыжок для направления вверх и вперед, лучше сразу задать направление вперед тоже, а не делать прыжок вверх и потом трансформировать его в прыжок вперед. Проще будет убрать перемещение по оси X. В таком случае, скорее всего, никакой дополнительной коррекции анимации не потребуется. А вот создавая прыжок только вверх, вы рискуете столкнуться с тем, что при добавлении движения вперед анимация не будет смотреться. Особенно это видно при наличии пластичных элементов на персонаже. В нашем случае это юбка и волосы. При движении вперед они будут оттягиваться назад. А при прыжке вверх они будут оттягиваться вниз. Поэтому, если прыжок общий для обоих случаев, нам важно найти некоторую диагональ движения пластичных элементов, при которой оба варианта движения будут смотреться хорошо. Иногда для строгого соответствия направлению движения персонажа такие элементы делаются отдельно в движке. В таком случае анимация создается без них, а в движке добавляется новый элемент. Например, так можно сделать с длинной косой персонажа, которая будет следовать за ним и искажаться согласно законам физики.



Общий вид прыжка персонажа

На примере я представила вам кадры прыжка героини. Конечно, я увеличила расстояние между кадрами, чтобы отделить их друг от друга. Фактически прыжок не такой дальний. Начнем с первого этапа прыжка — приседания. Перед отталкиванием мы должны присесть и с усилием оттолкнуть тело вверх. В платформере из-за задержек в управлении для игрока мы не можем позволить себе долгий этап подготовки к прыжку. Поэтому можно создать его буквально одним кадром легкого приседания и сплющивания персонажа по вертикали. После этого последует резкий толчок вверх без замедления. Это позволит минимизировать задержку и создаст ощущение приложенной силы. Волосы в этот момент, как и юбка, максимально облепляют персонажа под действием воздуха вокруг. Одну ногу я решила поджать, так конечности лучше читаются на силуэте. Особенно это хорошо работает с короткими конечностями. Одна вытянутая нога и одна поджатая видны на силуэте даже в случае чибби-пропорций. К верхней точке добавляем замедление. Уже обе ноги поджимаются и максимально прячутся под юбку. Юбка же начинает расширяться, волосы также приходят в первоначальное состояние. После этого, с замедлением в начале, героиня начинает падать вниз. Ноги мы меняем местами, вытянув уже другую ногу вперед, а вторую снова поджав. В финале прыжка героиня сталкивается с землей и все пластичные элементы приходят в исходное положение. Приседание после прыжка нужно добавлять с осторожностью и проверять, как оно смотрится в игре. Игрок может прыгать постоянно, без перерыва. И тогда, поскольку у нас есть приседание до прыжка и после прыжка, в промежутках между прыжками героиня будет приседать по два раза. Также после прыжка игрок может захотеть сразу побежать вперед, и момент окончания прыжка не должен этому мешать. Аналогично и с началом прыжка. Желая прыгнуть на полной скорости, игрок не должен ощущать, что персонаж замедлился и приседает.

Частой ошибкой в создании анимации прыжка является перемещение персонажа с помощью кости таза (центра тяжести). В таком случае затруднено управление ногами. Кости ног и кости инверсной кинематики являются дочерними от тазовой, а значит, будут вытягиваться вслед за движением вверх. Но если вы захотите поджать

персонажу ногу в прыжке или еще что-то поменять, то столкнетесь со сложностями управления скелетом, так как влияние движения тазовой кости будет мешать свободно контролировать ноги. Для перемещения персонажа в пространстве нужно создать общую родительскую кость для всего скелета и двигать именно ее. *Root* использовать нельзя, так как изменение корневой кости приведет к проблемам при работе с анимацией в движке. Поэтому всегда нужно создавать отдельную кость, если нужен аналог корневой. А корневая кость пусть остается в своем изначальном положении.

ГЛАВА 17

Лицевая анимация

Как показать эмоцию

Для персонажей, имеющих в игре достаточный размер, чтобы можно было рассмотреть лицо, важно создание лицевой анимации. Лицо в статичном виде будет заметно выделяться на анимированном персонаже. Мы не можем оставить без внимания такой важный для персонажа элемент. Лицо позволяет показать реакцию героя на определенное событие. Мимика персонажа раскрывает его характер и добавляет жизни. У героя должно меняться выражение лица при атаке и получении урона, мимику можно добавлять даже при прыжке. В анимации покоя герой может задуматься или устать, и это также важно показать с помощью лицевой анимации.

Всю лицевую анимацию можно разделить на несколько областей, которые могут двигаться. В первую очередь это глаза. Добавив персонажу моргание, мы сразу сделаем его более живым. Также можно акцентировать внимание зрителя на определенных элементах окружения с помощью направления взгляда героя. Игрок посмотрит туда же, куда смотрит персонаж, так как во время игры мы ассоциируем себя с управляемым героем, и направление его взгляда не останется незамеченным. А значит, мы можем управлять взглядом игрока и указывать ему на что-то важное и интересное.

Дополнительно подчеркнуть мимику глаз можно бровями. Многие эмоции вообще можно показывать при помощи именно бровей и их положения. Нахмурить брови — при обиде и злости. Поднять — при испуге и удивлении. В концептах персонажей для анимации и игр брови часто делают нарочито выразительными для лучшей передачи эмоции. Брови — важный инструмент для создания мимики, поэтому их часто рисуют довольно крупными, темными и заметными.

Помимо глаз, эмоции можно показывать с помощью рта. Особенно если стиль проекта мультяшный и мы можем позволить себе утрировать эмоции. Например, при злости можно показать звериный оскал, а при удивлении — широко открытый рот. Это дополнительно подчеркнет эмоции, созданные с помощью глаз. Но в реалистичном стиле такие утрированные эмоции не всегда могут быть к месту. Анимировав рот персонажа, не забывайте, что при движении рта двигается и челюсть. Распространенная ошибка — не двигать челюсть вместе с губами. В Spine мы работаем с отдельными изображениями. И рот зачастую также задан отдельной картинкой. Поэтому движение рта и движение челюсти — две разные части задачи. Нужно не забывать про их совместное использование.

Также утрировать эмоции важно, если персонаж небольшого размера, а его лицо нельзя разглядеть в деталях. В таком случае нам нужно даже на значительном удалении добиться читаемости эмоции. А значит, необходимо задействовать все доступные мимические средства.

Иногда для усиления эмоций можно использовать и уши. Человек ушами особо не шевелит, по крайней мере настолько, чтобы это было заметно при эмоциях. Но вот если у нас зверь или фэнтезийный герой с большими ушами, то их можно задействовать в мимике: прижать при испуге, вытянуть вверх при удивлении или в стороны при злости. Это отлично покажет эмоцию даже в маленьком формате персонажа, так как уши выходят на силуэт, а значит, хорошо читаются.



По аналогии с ушами для усиления эмоций можно использовать и волосы. Пригладить волосы, когда герой пугается, или взъерошить, когда злится. Такой прием часто применяется в классической мультипликации.

Помимо самого лица и головы, эмоцию можно подчеркивать позой. При удивлении персонаж вытягивается вверх и выпрямляет шею. При испуге, наоборот, старается сжаться и отодвинуться от пугающего объекта. Можно создавать дополнительные движения руками для лучшего понимания эмоции. Персонаж может сжать кулаки от злости или попытаться закрыть лицо при испуге. Не всегда есть возможность использовать все тело для создания эмоции. Например, если персонаж держит в руках оружие, то задействовать руки у нас не получится. Тогда можно придумать движения тела целиком, допустим, при испуге и удивлении добавить подскок, это даст дополнительный акцент на эмоции. Также усилить эмоцию можно эффектами, это максимально акцентный способ, но используется он только в утрированном стиле. Герой может гореть от злости буквально или пускать пар от нетерпения. Вспомним аниме и эмоции в нем. Они могут быть максимально утрированными в этом стиле, и их можно использовать в качестве референса для анимации. В аниме эмоции еще показывают с помощью символов. К примеру, когда персонаж стыдится, на его голове появляется капля.

Настройка для лицевой анимации

Всю лицевую анимацию мы разделим на несколько частей. Моргание и анимация глаз, анимация рта и эффект 3D на все лицо. К каждому пункту нужно подходить отдельно, учитывая конкретную задачу. Во всей настройке лица важно разобраться с настройкой мешей, именно они являются основными элементами анимации лица. В самой же анимации нужно все эти выделенные мной пункты сочетать. Мы будем также делать отдельные анимации для глаз, рта и лица, а потом совмещать их на различных дорожках и в движке. Но всегда нужно помнить, что анимация лица комплексная и все ее части должны создавать единую картину. Если у персонажа двигаются только глаза, а другие элементы лица статичны, то это может выглядеть неестественно.

МОРГАНИЕ

Для анимации моргания нам нужны отдельно веки для обоих глаз. Изначально веки рисуются в закрытом состоянии, когда они покрывают глазное яблоко. В большинстве случаев моргание делается только верхним веком. Нижнее тоже участвует в моргании, но настолько незначительно, что этим можно пренебречь в анимации. Только у некоторых животных или придуманных существ могут задействоваться два века. Также это оправданно в случае определенной мультяшной стилистики, в которой нижнее веко используется для эмоций персонажа. Например, в аниме, где глаза персонажей очень крупные и именно на них строится эмоция. В аниме вы можете увидеть довольно явное использование нижнего века при моргании.

Разные веки нужны, если глаза в ракурсе, так как в перспективе форма глаза меняется, и будет лучше сделать два разных века, а не подгонять одно изображение под оба ракурса. Если персонаж показан строго в фас, то можно использовать одинаковые веки, отразив по горизонтали одно изображение. На изображении века должно быть минимум детализации. Это поможет лучше и проще исказить веко при моргании. Анимация моргания происходит очень быстро, поэтому детали нам на веке не нужны, они будут только мешать при анимации.

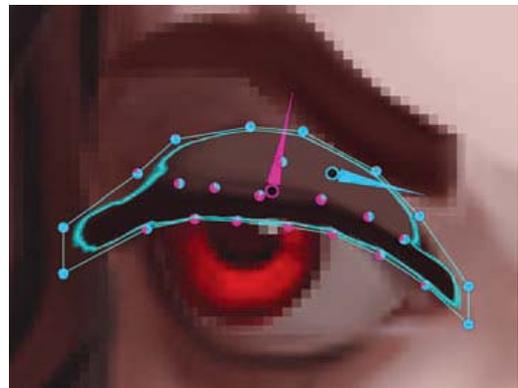
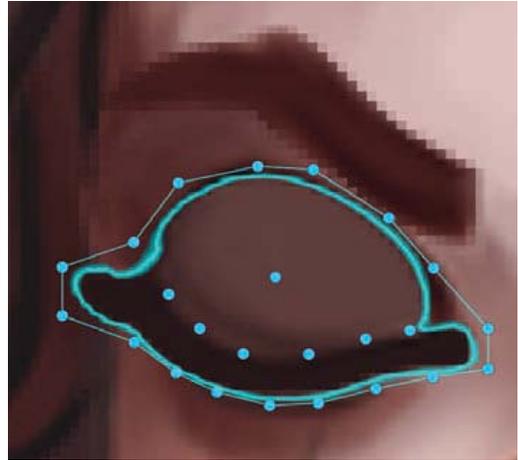
В первую очередь нам нужно сделать меш для века. Необходимо максимально близко к контуру обвести его точками меша, минимизируя прозрачные области. Но помните, что нужно обходиться наименьшим количеством точек, это тоже важно для оптимизации. Основную детализацию лучше сосредоточить на нижней границе века. Именно ее мы будем больше всего искажать. Поэтому по нижней части лучше пройти максимально точно. Если на веке есть ресничная линия, то ее нужно дополнительно обвести точками меша. В процессе искажения века нам нужно будет контролировать изменение ширины этой линии. Иногда, если нужна детализация, ресничный край рисуется отдельно и анимируется также отдельно с помощью собственного меша.

Для управления веком нам потребуются две кости. Одна для управления морганием. Другая — для движения века в целом и участия в 3D-эффекте. Кость моргания является дочерней от кости движения.

Начинаем настройку весов с того, что привязываем весь меш века к кости движения. Все точки на 100% подчиняются одной кости движения. Привязываем и вторую кость, но пока не добавляем ей веса. Если привязывать по очереди, то веса как раз автоматически и распределяются на одну кость, привязанную первой. Если вы привяжете сразу обе, то веса сначала распределяются по двум костям.

Далее создадим новую анимацию моргания. В локальной системе координат для кости моргания делаем два ключа для закрытого и открытого глаза. Мы делаем это без настройки весов, поэтому с веком пока ничего не происходит. Просто кость в нулевом кадре стоит внизу глаза, а в конечном кадре анимации вверху глаза. Мы делаем анимацию для последующей настройки весов таким образом, чтобы веко открылось к конечному кадру. Выделяем конечный кадр анимации и подкрашиваем точки меша в цвет кости моргания, чтобы веко было открытым. Основное влияние кости моргания будет на нижней линии века посередине и по ресничному краю. К краям глаза влияние уменьшается, а верхняя граница века остается под влиянием кости движения.

Также можно посмотреть промежуточные кадры между нулевым и финальным. Нужно убедиться, что в этих кадрах все хорошо с искажением века. Если что-то не так, необходимо поменять веса костей. В данном примере мы настраиваем веса не заранее, а под конкретное движение. Так мы можем работать не гипотетически, не зная, как поведет себя изображение, а настраивать веса костей по ходу работы, учитывая возникающие артефакты. Если округлая линия века не получается, то можно добавить больше точек на край века или отрезать ресничный край отдельно. Если персонаж не очень крупный, то анимацию моргания можно не делать строго без углов и артефактов. В быстром формате анимации и на удалении от зрителя некоторые неточности заметны не будут.



Розовым цветом отмечена кость моргания. Голубым — кость движения

ДВИЖЕНИЕ ГЛАЗ

Если нам потребуется настройка движения глаза, то исходник должен быть соответствующим. Главный момент — прорези в лице под глаза. Как бы странно это ни звучало, нам нужно вырезать части, которые отведены под глаза. Это делается для того, чтобы свободно перемещать зрачок под таким исходником лица. Далее под это лицо мы подкладываем изображения белков глаз. Между лицом и белками помещаем зрачок. Зрачок совмещен с радужкой в одно изображение. Отдельно они не двигаются, поэтому разделять их не нужно. Исключением является анимация самого зрачка при реакции на свет. Это, конечно, очень редкая ситуация, когда нам требуется такая анимация. Глаза должны быть очень крупными, чтобы это было заметно. И сама задача должна быть оправдана. Такая анимация может быть в анимационном ролике с кадром, в котором очень близко лицо персонажа. Если зрачок должен менять размер при усилении света, тогда мы отделим его от радужки. В остальных случаях это нам не нужно и мы будем работать с цельной окружностью радужки и зрачка.

Зрачок с радужкой должен быть круглым в исходном виде. Если нам понадобится его искажение в перспективе, сделаем это уже в программе. С таким положением изображений нам не понадобится маска для глаза. Лицо будет перекрывать все, что происходит под ним, оставляя только области глаз видимыми. А значит, мы можем свободно перемещать зрачки, не переживая о вылезавших за границы глаза элементах.

Зрачку с радужкой добавляем кость по центру. Ставим его так, чтобы глаз смотрел максимально вверх. Учитывайте, что сам зрачок (черный круг) не может уходить за пределы видимой части глаза. Если так сделать, глаза будут выглядеть закотившимися. Поэтому не нужно уводить глаз в стороны слишком сильно. Зрачок всегда должен быть виден.



Исходник лица для анимации глаз



Настройка костей глаза

Далее создаем кость из центра глаза в край, куда смотрит зрачок. Эта кость пересекает кость зрачка. Кость зрачка является дочерней от длинной кости из центра глаза. Обязательно приблизьте глаз в рабочей области программы и убедитесь в том, что кость глаза пересекает кость зрачка. Это обязательно для дальнейшей работы. Если есть расхождение, нужно это поправить.

Для кости из центра глаза нам нужно создать инверсную кинематику. ИК должна быть на кончике кости, а значит, при создании мы должны нажать на кость. Тогда ИК автоматически создастся на кончике кости. У этой инверсной кинематики мы ставим в настройках параметр «Сжать». У кости зрачка мы убираем наследование вращения и масштабирования (в окне настроек под структурой проекта). Кость наследует только перемещение.

Теперь попробуйте двигать инверсную кинематику. Глаз не будет уходить за пределы видимой части белка. По сути, длиной кости из центра глаза до его края мы задали область возможного движения зрачка. Движение происходит лишь при сжатии и растяжении кости. Растянуться она может только обратно до исходного размера и не более, так как параметр «Растянуть» мы не включили в настройках. Включен только параметр «Сжать». Движением ИК мы сжимаем кость глаза, и кость зрачка движется следом как дочерняя.

Для второго глаза настройку повторяем полностью аналогично первому. Нужно только учитывать форму глаза, которая может слегка отличаться. Также, чтобы глаза двигались синхронно, максимальная амплитуда движения должна совпадать. Максимальная амплитуда в данной настройке у нас задается длиной кости глаза. А значит, длина кости глаза у обоих глаз должна быть одной и той же. Просто скопируйте числовое значение из одного поля в другое. Если длина будет отличаться, то возникнет ситуация, когда один глаз уже достиг предела движения и не двигается, а второй продолжает двигаться. Если нам нужно немного донастроить амплитуду движения зрачка, то длину кости глаза менять не нужно. Необходимо кость зрачка переместить вдоль кости глаза.



Теперь настроим контроллер для движения двух глаз. Нам вообще редко когда нужно анимировать глаза отдельно. Если персонаж следит за предметом, то обоими глазами сразу. Проще управлять ими совместно, а не по отдельности. Создаем новую кость-контроллер. Привязываем кости инверсной кинематики обоих глаз к контроллеру с помощью ограничителя трансформы. Нажимаем в настройках ограничителя кнопку «Сопоставить» и задаем числовое значение в поле «Преобразовать» панели

«Смешанное». Остальные значения оставляем нулевыми. Значение «Преобразовать» не должно быть 100%, потому что тогда мы лишимся возможности двигать глаза отдельно. А эту возможность лучше сохранить, даже если она кажется ненужной. Настройка должна быть максимально функциональной. И в целом, если поставить большое значение, близкое к 100, то для того, чтобы двигать глаза, кость-контроллер придется двигать очень мало. Иначе будет просто неудобно, потребуется слишком небольшая амплитуда движения. Лучше поставить значение 10–30%, чтобы можно было свободно двигать кость-контроллер.

В финальной части настройки может случиться так, что у вас не двигаются изображения. Причем могут двигаться даже ИК вместе с контроллером, а изображения оставаться на месте. В таком случае проверьте структуру проекта в меню ограничителей. Ограничители трансформы должны в иерархии стоять выше, чем инверсные кинематики.

АНИМАЦИЯ РТА. ЛИПСИНГ

Движение рта необходимо в двух случаях. При разговоре и для создания эмоций персонажа. Создание анимации разговора называется липсингом. Эта анимация подразумевает изменение формы рта в зависимости от слов и звуков, которые произносит герой. На каждый звук, конечно, анимацию создавать не нужно, это слишком трудозатратно, да и избыточно. Многие звуки мы визуальным образом произносим довольно похоже. Нужно разделить звуки на группы и сделать несколько видов анимаций. Иногда для каждого вида звуков рисуют отдельный рот, чтобы более точно передать его форму. Менять изображения можно просто переключением. А если требуется большая плавность, то с помощью меша можно деформацией подтягивать одну форму рта к другой. Так мы создадим промежуточные кадры, но при этом сохраним уникальные, отдельно нарисованные формы рта.



Пример форм рта из игры *Almost my floor*

Можно менять форму рта только с помощью меша и правильной нарезки, не создавая отдельных изображений. Нарезка требуется, если у нас есть детализация губ или внутренней части рта. Губы можно разделить на верхнюю и нижнюю. Отдельно

оставляем внутреннюю подложку рта, на которой будут зубы и язык. Зубы и язык также можно отрезать отдельно для большего контроля. Например, это нужно для того, чтобы сомкнуть зубы при злости или, наоборот, открыть больше внутренней части рта при пении или крике героя. Нарезка зависит от конкретного дизайнера и задачи.

Звуки делятся на гласные и согласные, среди которых выделяются особенные звуки. Гласные требуют отдельного подхода, так как почти каждый из них произносится с разным раскрытием рта. Звук «а» и звук «и» одной анимацией не передать. Можно показать одной формой рта звуки «и», «ы», «э», «е». Все они произносятся с помощью расширения рта по горизонтали. Если нет задачи передавать идеально точный липсинг, можно сделать одну форму рта под все эти гласные. Также одной формой можно задать гласные «а» и «я» — растянуть рот по вертикали, довольно сильно открыв его. Остаются звуки «у» и «ю», которые также создаются одной формой рта. Для этого нужно вытянуть губы в трубочку. Для создания гласных «о» и «ё» тоже нужно вытянуть рот, но не смыкать губы так плотно, как в случае с «у» и «ю». Гласные звуки задают основу липсинга. В целом именно за счет изменения формы рта при произношении гласных мы и считываем анимацию разговора. Рот больше всего изменяется именно на этих звуках. Гласные более заметны в речи и анимации, и для них лучше создавать именно несколько форм рта. Не важно, с помощью меша или отдельных изображений.

С согласными звуками проще. Нам необходимо создать всего несколько самых заметных форм. Например, для звуков «м», «п» нужно подтянуть губы внутрь рта. Для шипящих звуков — слегка приоткрыть рот и сомкнуть зубы. В остальном нужно ориентироваться уже на конкретного персонажа и детализацию его лица, на стиль, в котором он нарисован. Согласные произносятся быстро и похоже, поэтому можно ограничиться буквально парой-тройкой форм рта для создания липсинга. Но при реалистичной детализации может потребоваться и более серьезная проработка.

АНИМАЦИЯ РТА. ЭМОЦИИ

Для создания эмоций также можно использовать различные формы рта, как и при липсинге. Но в случае создания эмоции форма будет более утрированная и выразительная. Создавать форму рта нам нужно для крайнего состояния выражения эмоции. Промежуточные состояния мы можем сделать мешем. Если эмоция длительная по времени, то нам потребуются анимация внутри самой эмоции, а не только изменение одной формы рта на другую. Если сравнить с липсингом, то в нем слова произносятся быстро и определенная форма рта на экране долго не задерживается. Но вот если персонаж поет и тянет какую-либо гласную, то из-за длительности произнесения может дополнительно понадобится анимация тянущейся гласной. В любом случае мы будем делать подобные анимации мешем или подключим нарезку, как я уже говорила ранее.

Рассмотрим пример персонажа-воина. Во время атаки мечом персонаж должен менять лицо с улыбающегося на злое. Воин, размахивающий мечом направо и налево

с улыбкой на лице, выглядит не очень правдоподобно. Поскольку у нас два состояния, противоположных друг другу, мы не можем просто взять и изменить одну картинку на другую. Сначала улыбка должна сойти с лица, а затем перерасти в злость. Изменение должно происходить поэтапно. Просто поменять картинку можно, только если эмоция происходит очень быстро или скрыта каким-то предметом или эффектом в момент смены выражения лица.



Поэтапное изменение лица персонажа от улыбки к злости

Для того чтобы создать поэтапность эмоции, нам нужно настроить меш для обоих состояний рта персонажа. В качестве исходников у нас есть два изображения: улыбка и злобный оскал.

Начнем с улыбки. Это изображение рта базовое для персонажа. Именно с ним мы видим героиню большую часть времени. И настройку базового варианта лучше провести с заделом на самостоятельное изменение, в том числе вне эмоции. То есть мы должны иметь возможность контролировать степень улыбки и наклон уголков рта, и даже форму рта целиком. Это может понадобиться для того, чтобы персонаж не ходил все время с одним и тем же выражением лица. Постоянно улыбающийся воин выглядит странно. Анимацию рта можно добавить в анимацию покоя, меняя степень улыбки героини.

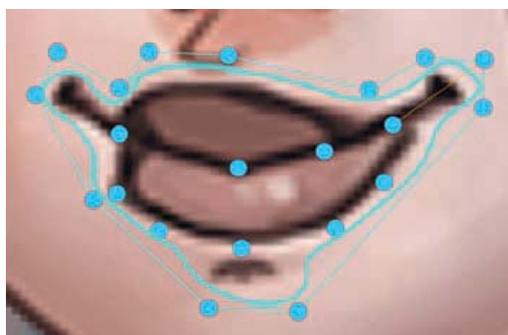
Для таких изменений нам нужен меш. Точки меша должны располагаться по контуру изображения и по форме прорези рта. Учтите, что иногда исходник рта содержит еще и область вокруг губ. Тогда контур меша не будет точно совпадать с контуром губ. Часть вокруг губ может быть добавлена в изображение рта для создания тени или морщин. В моем примере под нижней губой есть складочка, которая прикреплена к исходнику. Такие детали двигаются вместе с губами, поэтому я объединяю их в одно изображение. Можно разделять такие детали, но тогда в анимации необходимо следить за тем, чтобы они двигались вместе с движением рта.

Также, если вам необходимы положения рта, в которых уголки слишком сильно направлены вверх или вниз (то есть персонаж очень веселый или, наоборот, угрюмый), исходное положение должно быть прямым. Выгнуть губы из одного состояния в другое будет сложнее, чем из прямого положения. Количество точек зависит от формы рта и размера персонажа. Если персонаж маленький, то рот еще меньше. В таком случае мы можем не обращать внимания на некоторые изломы рта при деформации.

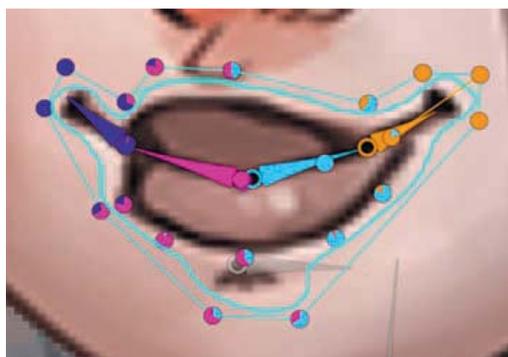
Для управления мешем мы создаем четыре кости. Две для уголков рта и две для центральной части. Одну цепочку из двух костей направляем влево, другую — вправо. Уголки рта делаем дочерними от центральных костей. Этими костями мы будем задавать основное движение рта. Но для лучшей формы губ при различных положениях рта лучше дополнительно еще двигать отдельные точки меша в анимации. Так мы добьемся наилучшего результата. Веса распределяем автоматически с корректировкой уже в процессе анимации, если что-то пойдет не так. В целом нам нужно распределение согласно положению костей с градиентом веса от одной кости к другой, чтобы изгиб был более плавным. Такое распределение Spine дает практически автоматическим методом.

Теперь в анимации изгибаем кости вниз, тем самым убирая улыбку с лица героини. Дополнительно корректируем форму губ, двигая отдельные точки меша.

А также с помощью изменения масштаба крайних костей мы можем изменять длину уголков рта. Уменьшим кость, и уголки станут короче. Это даст лучший переход в положение злого оскала, так как на втором изображении рта, показывающем злость, длинных уголков нет. Небольшие артефакты видны только при максимальном увеличении. В таком формате смотреть работу никто не будет, поэтому главное — это то, как выглядит форма рта в целом. Всегда нужно издали смотреть на формат персонажа целиком и с этой позиции оценивать работу.



Настройка меша для искажения рта



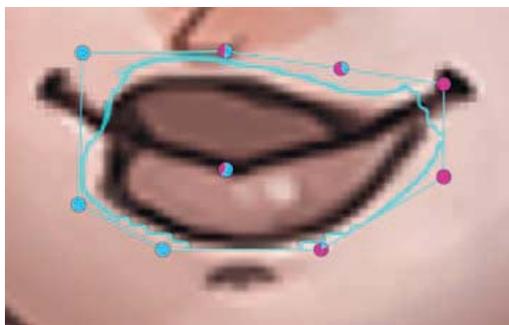
Настройка костей и весов для искажения рта



Этапы деформации улыбки

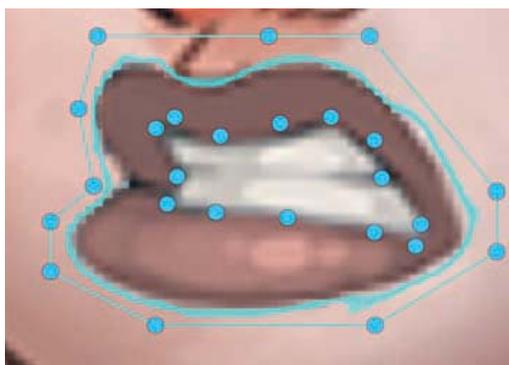
Минусом такой настройки рта является большое количество точек меша на довольно маленьком объекте. Такую настройку можно применять для крупного плана лица. Если черты лица героя на экране представлены в мелком формате, то лучше еще

больше пренебречь плавностью деформации, но оптимизировать количество точек меша. Можно даже обойтись без костей и деформировать меш напрямую. Когда точек мало, прямой деформацией пользоваться проще, так как одна точка меша контролирует больший его объем. Также можно отрезать все побочные детали и уголки рта, чтобы контролировать их отдельно, и использовать меш только непосредственно для губ.



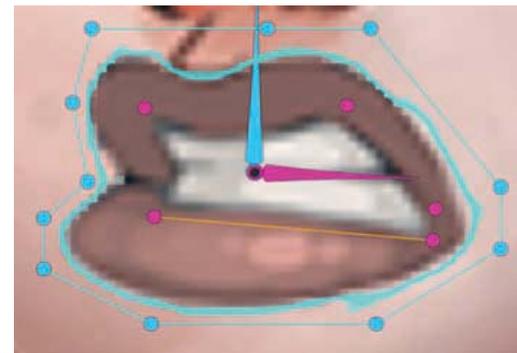
Вариант оптимизированной настройки меша

Первую часть перехода от улыбки к злости мы разобрали. Мы постепенно меняем эмоцию, убирая улыбку и делая выражение лица более нейтральным. Теперь нужно, наоборот, из крайне злого выражения рта получить более нейтральное. Необходимо добиться того, чтобы посередине перехода эти два изображения были максимально похожи и мы могли незаметно поменять одно на другое.



Настройка меша для злого варианта рта

Логика настройки остается такой же. Мы должны обвести точками меша контур формы губ и внутреннюю часть рта. Нам нужно будет менять ширину открытия рта и степень оскала зубов. Контроль над этой частью очень важен. Мы можем обойтись свободной деформацией и просто подогнать точки меша в нескольких промежуточных кадрах между нейтральным и злым выражениями рта. Плюс такого подхода в полном контроле над каждой точкой. А минус — это трудоемкость процесса. Если нам нужно много анимаций с использованием деформации рта, то лучше настроить кости. Каждый раз подгонять позиции точек меша слишком трудозатратно.



Настройка для злого варианта рта с костями

Создаем две кости. Одну для контроля над внутренней частью рта, а вторую для контроля над внешней частью и над ртом в целом, она же будет основной. Обе кости берут начало в центре изображения рта. Точки меша внутренней части рта мы оптимизируем и уменьшаем их количество, грани меша создаем вручную. Мы все еще обводим внутреннюю часть рта,

но уже не так точно, главное — задать общую форму. Веса распределяем следующим образом. Внутренняя часть полностью контролируется одной костью, а внешняя часть другой. Окрашиваем все точки меша внутренней и внешней части полностью в разный цвет. С такой настройкой мы можем, меняя масштаб костей, управлять степенью открытости зубов и масштабом рта в целом.

Псевдо-3D лица

Помимо движений частей лица, нам нужно работать с лицом в целом. И часто для этого необходимо менять ракурс. Учитывая специфику работы со скелетной графикой, подобные вещи в анимации довольно сложны и требуют отдельного подхода. Все-таки в основном мы работаем в плоскости, и для выхода из заданного ракурса нужны определенные настройки и техники.

Вид лица и его отдельных частей меняется с изменением ракурса. Сильный 3D-эффект получается только тогда, когда изменение ракурса достигается за счет искажений. А это возможно лишь в упрощенном стиле с утрированием частей лица в сторону геометрических форм. Если у нас детальные веки, нос или губы, то менять их ракурс только искажениями будет затруднительно.

На примере изображено лицо, для которого можно легко сделать 3D-эффект. Отсутствие плавных теней позволяет нам двигать объекты по лицу вместе с их тенями, не нарушая объема. Если добавля-



Пример простого лица для 3D-эффекта

ются плавные градиенты, то это накладывает ограничение на 3D-эффект. Черты лица сделаны по типу аппликации. То есть они не встроены в форму лица градиентами или обводкой. Поэтому мы свободно можем их двигать. Глаза простой овальной формы, как и зрачки. И при этом даже нет век, которые меняют свой ракурс при повороте. Уши, если их отделить от головы, не добавляя обводку на стыке с ней, тоже можно будет свободно перемещать. Эффект 3D будет создаваться сужением и перемещением всех элементов лица в определенную сторону с коррекцией расстояния между ними согласно перспективе. Даже волосы можно нарезать отдельно на пряди и анимировать таким же образом. Для более детальной деформации на отдельные части лица можно добавить простые меши. Это позволит нам создать эффект расположения элементов по сфере головы, и поворот будет выглядеть естественнее.

Для создания эффекта 3D для сложных форм лица используются дополнительные спрайты. Спрайты рисуются для различных ракурсов элемента. Техника работы с такими спрайтами схожа с техникой смены эмоции. У нас есть два крайних положения с отдельными картинками. Создавая 3D-эффект деформациями и перемещением, мы тянем один ракурс к другому и в середине анимации меняем изображения. Такая техника подойдет для изменения положения носа снизу вверх с постепенным появлением ноздрей в ракурсе. Или, к примеру, для перехода из фаса в профиль нам понадобятся два спрайта для губ и глаз при их сильной детализации, так как вид глаза и рта при повороте головы значительно меняется. Иногда даже создается промежуточный кадр отдельным изображением, а промежутки между ракурсами дополняются деформацией. Все зависит от детализации.

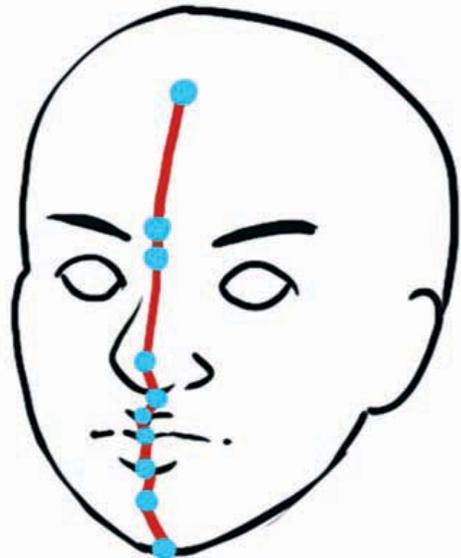


Пример поворота простого лица смещением и деформацией его частей

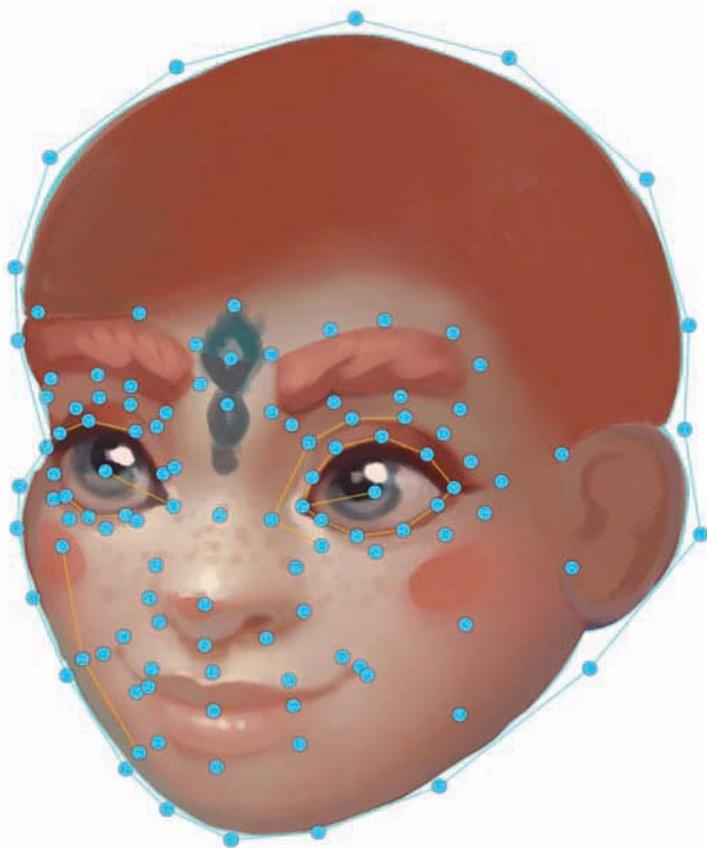
В большинстве задач эффект псевдо-3D не используется на 360 или даже 180 градусов. Часто нам просто нужно слегка поворачивать персонажа и его лицо, менять ракурс плеч при ударе оружием или ракурс лица, когда персонаж следит взглядом за каким-либо предметом. Такой эффект можно сделать для исходника любой сложности.

Эффект 3D осложняют дополнительные элементы в дизайне персонажа. Все-таки лицо героя находится не в вакууме, а в окружении волос и других деталей. И эффект 3D должен быть на всех элементах персонажа. А значит, чем больше дополнительных деталей, тем сложнее будет создать 3D-эффект. Шляпа, серьги, волосы — все это также придется настраивать для работы с ракурсом.

Мы рассмотрим лицо отдельно. В 3D-эффекте лица главное — правильная настройка костей и меша. 3D-эффект основан на искажении и перемещении частей лица. Поэтому точками меша мы должны обвести все элементы на лице. Также нужны точки меша по центральной оси лица. Если исходник предусматривает отдельные изображения для частей лица, то мы создаем отдельные меши для каждой



части, также обводим их точками и задаем центральную часть у носа и рта. Кроме того, важно точками меша отметить ключевые точки лица: уголки губ, глаз и ноздрей, начало и конец бровей. Если на коже лица присутствует детализация, то и там добавляем точки меша, чтобы детализация также перемещалась со сменой ракурса. Но этот вариант используется скорее для реалистичных лиц с деталями на коже или для персонажа с татуировкой на лице. Если у нас лицо однотонное или градиент, то точки располагаются только возле частей лица. Если у персонажа выраженный рельеф лица, например на щеках или подбородке, то этот рельеф также нужно передать в меше. Проводим линию сверху вниз через основной рельеф и размещаем несколько точек меша на этой линии.



Настройка меша для 3D-эффекта лица

В примере из игры *Potata* персонаж средней детализации и стилизованности. На лице есть определенная детализация и рельеф. Выраженные щеки, краска на них и градиенты объема. Исходник представляет собой цельное изображение лица без разделения на части. Обводим точками меша контур изображения и части лица. Также, поскольку есть рельеф и детализация на коже, добавляем на щеки линию из точек. Желтым цветом обозначены линии, которые нарисованы вручную. Автоматически программа не всегда настраивает нужные нам грани. Также обратите внимание на обводку глаз

двумя линиями из точек. Это сделано из-за выраженных век. Если мы захотим менять наклон лица вверх-вниз, то нам понадобится контроль над веками. Так как при подобной смене ракурса глаз верхние или нижние веки будут больше видны.

Меш изначально должен быть привязан к кости, ответственной за голову. Также создаем новую кость-контроллер для смены ракурса и привязываем меш и к ней. Кость-контроллер является дочерней от кости головы. Иногда нам могут потребоваться две кости-контроллеры, дочерние от кости головы. В эффекте 3D бывают ракурсы, когда некоторые точки меша должны идти в обратном направлении относительно движения основного контроллера. Например, когда нужно спрятать определенную часть лица при смене ракурса. В таком случае мы делаем дополнительную кость-контроллер и ограничителем трансформы привязываем ее к основной. И в настройке «Преобразовать» ставим значения -100 по обеим осям. Так кости будут двигаться в противоположном друг другу направлении, а значит, и привязанные к ним точки меша тоже.

Распределение весов мы делаем в зависимости от выпуклости точки на лице. Чем больше конкретная точка выступает вперед, тем больше веса управляющей кости будет в этом узле. Самая выступающая точка лица — это нос, а значит, и больше всего влияния кости-контроллера будет на нее.

Такую сложную настройку весов необходимо проводить в меню анимации, постоянно проверяя, как работает эффект 3D. В анимации нужно зафиксировать среднее начальное положение головы и оба крайних ракурса поворота максимально, насколько позволяет исходник. После этого необходимо, поворачивая управляющую кость, проследить за появлением изломов и артефактов. Кроме того, может происходить неравномерное искажение и движение элементов лица. Шаг за шагом, корректируя веса, вы придете к нужному результату.

Для понимания выступающих точек лица можно обратиться к референсу, который всегда под рукой, и подойти к зеркалу. Это позволит оценить лицо в ракурсе в профиль и посмотреть, какие точки наиболее выдаются вперед. В основном это нос, щеки (скулы), брови и подбородок. Но разные лица могут давать разные соотношения. У персонажа могут быть очень выступающий подбородок или выразительные брови. Нужно всегда отталкиваться от исходного дизайнера персонажа.

Если у нас какие-то части лица присутствуют по отдельности, мы также отдельно создаем для них меши, а значит, и кости для перемещения. Это относится и к побочным элементам вокруг лица. Создав кость для части лица, мы привязываем ее ограничителем трансформы к основной управляющей. Для того чтобы настроить веса, нужно посмотреть настройку основного меша и веса точек рядом с элементом. Рассчитываем среднее значение веса управляющей кости и такое же ставим на отдельный элемент. Изначально весь меш элемента привязан к его кости и вес всех точек — 100% . А после настройки веса будут поделены между управляющей костью и костью элемента.

Особенности анимации персонажа в изометрии

Отдельно хочу остановиться на изометрии, так как эта проекция очень часто используется в играх с 2D-графикой. Изометрия помогает имитировать 3D без фактического его применения. Для построения изометрии используется сетка из квадратов, повернутых на 45 градусов и сжатых по вертикали в два раза. Если вы посмотрите характеристики изометрии в курсе начертательной геометрии или даже в 3D-программе, то увидите, что они отличаются от упрощенного 2D-формата. Но для удобства построения сетки в качестве основы обычно используют именно сжатый по вертикали ромб. Все объекты и персонажи для соответствия ракурсу рисуются по данной сетке.



Пример изометрической сетки и персонажа на ней

Такая проекция часто встречается в играх, причем в самых популярных. Обратите внимание на жанр *builder*. Очень много проектов в нем задействуют изометрию, при этом используя 2D-анимацию. А значит, аниматору, желающему попасть в геймдев, нужно знать базу работы в изометрии.

Все особенности подхода к анимации в изометрии вытекают из специфики ракурса. Мы уже не можем работать в плоскости, не обращая особого внимания на перспективу. В изометрическом ракурсе есть строгая иерархия перекрытий элементов, что также нужно учитывать. Изометрия подразумевает ракурс сверху. А значит, нижние части перекрываются всем, что находится выше по логике персонажа. При этом нужно ориентироваться именно на логику героя, а не на фактическое расположение. Дальняя рука персонажа (та, что дальше от зрителя) тоже будет находиться в 2D-плоскости выше, чем нога, но дальше по анатомии персонажа. А вот шляпа

всегда будет перекрывать остальные элементы. По сути, нам нужно всегда держать в голове 3D-формат изображения и опираться на него при анимации.



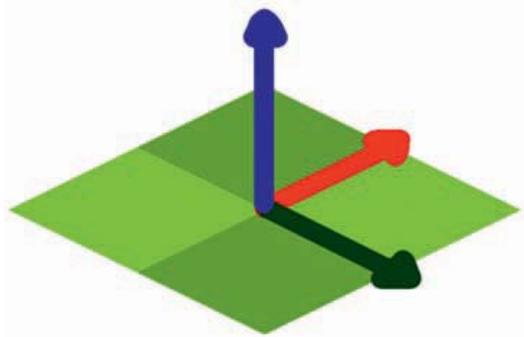
Чтобы сохранять ракурс и в движении, нам нужно само движение также выполнять по изометрической сетке. Передвижение рук и ног в анимации ходьбы яркий тому пример. Мы задаем направляющие по сетке и уже по ним двигаем элементы. Это сложнее, чем в плоскости, так как нельзя нарушать ракурс. А главное — при движении должны сохраняться расстояния между некоторыми элементами. Те же руки и ноги опять хороший пример. Двигая ноги, мы не можем нарушить изначальное расстояние между ними в ракурсе. Иначе ракурс перестанет читаться.

Если нам нужно сжатие или растяжение элемента или объекта в целом, мы не можем делать эти манипуляции



Пример направляющих для походки персонажа. Черными стрелками показаны сами направляющие, а белыми расстояние, которое должно сохраняться

в стандартных осях. Оси в данной проекции смещены по ее направляющим. Ось Y в плоскости представляет собой перпендикуляр. Исходя из этого, нам нужно или использовать меш, или задавать объекту искажения по смещенным осям. В данном ракурсе персонажи в большинстве случаев небольшого размера, так как изометрия в основном используется на достаточном удалении от зрителя. Поэтому сжатие и растяжение лучше не делать слишком выраженными, а, может, не стоит их делать вообще. Искажения занимают достаточное количество времени, а в итоговой анимации их может быть не видно или они даже ухудшат общее впечатление.



Частой проблемой при анимации изометрического персонажа становится силуэт. Из-за перекрытий в ракурсе многие объекты оказываются скрытыми. Например, за большой шляпой, животом или объемными штанами могут быть скрыты ноги. А для читаемости ходьбы важно, чтобы их части было видно. Дальняя рука персонажа также большей частью будет скрыта от зрителя, если не утрировать ее положение и силуэт. Иногда можно даже слегка нарушить правила и вытащить на силуэт ногу или руку там, где их быть не должно. Главное, чтобы от этого не пострадала читаемость ракурса. На примере с котом я стараюсь не прятать дальние руку и ногу полностью, так как тогда их совсем не видно в мелком формате на общем экране. Также в этом примере максимально утрированное положение ног, опять же, чтобы движение при ходьбе было видно в итоговом размере. А вот подгонять ступни под направляющие изометрической сетки я не стала, потому что маленький размер позволяет не замечать некоторые шероховатости. Но если бы персонаж был крупнее, то нужно было бы мешем подгонять положение пальцев под сетку изометрии. Обратите внимание, что при повороте ног пальцы смещаются вместе с ногами и не соответствуют сетке.



Основные кадры ходьбы персонажа в изометрии

Если персонаж сделан с помощью 2D-графики в изометрии, то его ходьба ограничена несколькими направлениями. Обычно это восемь направлений. Четыре — по сетке. И еще четыре — строго влево, вправо, вверх, вниз по диагоналям сегментов сетки изометрии. Для каждого направления нужен свой ракурс персонажа. Часть ракурсов можно получить, отразив героя по оси X. Но остальные требуют отдельных исходников.

Для каждого ракурса нужна своя анимация. Сложность в том, что это один и тот же персонаж. Движение также идентично. А значит, нам нужно создавать идентичные по параметрам анимации.

Хорошим подходом будет сохранять числовые значения, расстояния, углы поворотов. Тогда, используя их в анимации другого ракурса, мы добьемся максимальной схожести. Добиться идентичности только за счет чисел не получится, так как на числа влияет сам ракурс, и иногда они все же должны отличаться. В этом случае подключается метод «на глазок». Но, пользуясь этим методом, обязательно ставьте анимацию разных ракурсов рядом, чтобы иметь возможность оценить сходство.

По причине сложности реализации 2D-анимации в изометрии персонажи в подобных проектах могут создаваться с помощью 3D-болванок. Но 2D-анимация все же не списывается со счетов полностью. Она дает большее ощущение мультфильма, и в определенных типах стилизации используется именно 2D. Задача анимации существ и персонажей в изометрии одна из самых сложных для аниматора. А вот механизмы, здания, элементы природы в 2D делать удобнее и порой быстрее, чем в 3D. Все зависит от сложности объекта. 3D требует отдельных подготовительных этапов, таких как создание модели, текстурирование, риг. И для каждого минимального элемента на экране делать такие манипуляции долго и затратно. Поэтому 3D может быть использовано для основного героя, который понадобится во всех ракурсах поворота. А для более статичных персонажей и объектов подойдет 2D-анимация в Spine.



Исходник персонажа, вид сзади

ГЛАВА 18

Эффекты в Spine. Пример нестандартного использования программы

Я уже много раз упоминала создание эффектов в Spine. Но в основном это были просто покадровые последовательности, встроенные в основную анимацию, или деформации, которые сами по себе дают довольно скромный потенциал в создании эффектов. Но если комбинировать средства программы, можно создавать довольно интересные и комплексные эффекты.

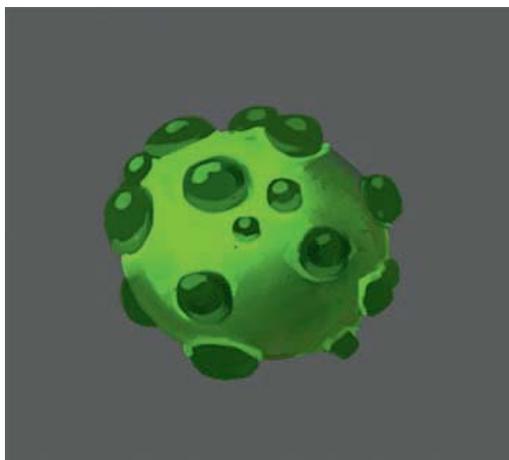
Spine часто воспринимается как средство для анимации персонажей, ну, или других объектов, у которых есть скелет. Но программу можно использовать вообще для чего угодно. Давайте посмотрим на пример бомбы из нашей игры *Potata: fairy flower*. Эффект ее движения и взрыва создан с помощью мешей, отдельных кадров и их движения и деформации.

В качестве основного исходника мы используем цельное изображение. Нарезки именно основы бомбы нет никакой. Все разделения внутри одного изображения мы зададим с помощью меша.

Логика построения меша такова, что основные выпуклости на бомбе мы должны обвести точками, чтобы иметь возможность их трансформировать. Выберем самые крупные и заметные из них и внутри меша добавим точки вокруг. Также, само собой, меш должен очерчивать контур бомбы. Там, где на контур выходят вздутия, их основания тоже можно очертить точками меша. Для основных вздутий назначим кости, каждому свою.

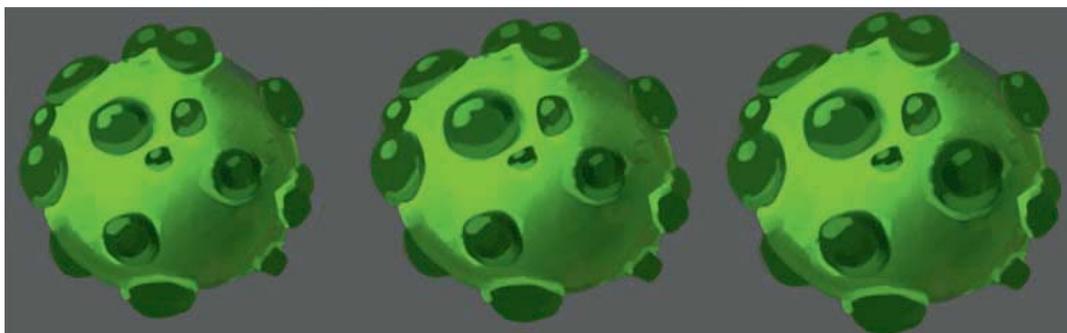
Или даже одну отдельную на все, но тогда мы не сможем вздуть один шарик больше, чем другой. Зато управление мешем сильно упростится. Также несколько костей для контроля над вздутиями полезны тем, что мы сможем вздувать части нашей бомбы не одновременно, а создавая случайный эффект. Еще можно использовать и свободное трансформирование. Особенно если требуемых искажений немного.

На примере у бомбы три основных шарика внутри, которые мы будем раздувать по очереди. Сначала увеличится верхний, а затем подтянутся два нижних. Таким образом, у нас получится эдакая анимация покоя для бомбы. В игре она так и используется.



Исходник бомбы

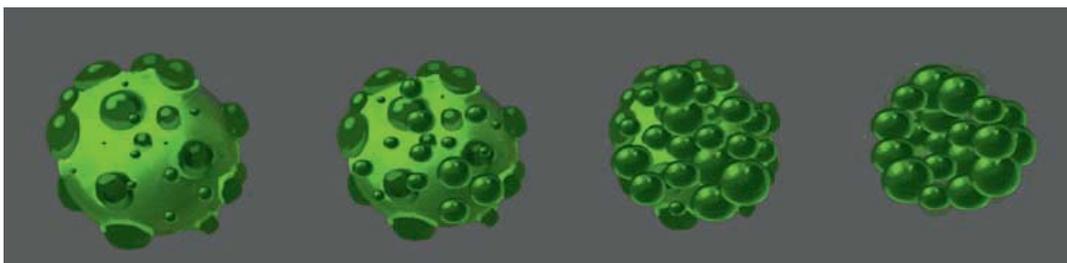
Пока героиня не подойдет к бомбе на определенное расстояние, бомба будет слегка раздуваться и сдуваться. А вот когда героиня пересечет черту, бомба приготовится к взрыву и взорвется. Далее добавим эту анимацию к нашей бомбе.



Пример постепенного раздувания бомбы

Идея анимации такова: бомба наполняется темными шариками, их становится все больше на ее поверхности. Когда она вся покрывается вздутиями, происходит взрыв. Эта анимация хорошо встроена в игровой процесс. Пока бомба готовится к взрыву, у игрока есть время отойти на безопасное расстояние. Если бы бомба взрывалась сразу же, то игрок не смог бы избежать урона.

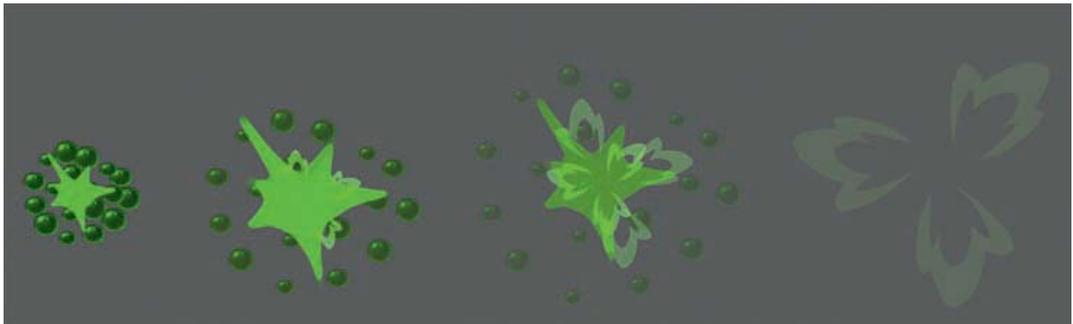
Для заполнения бомбы взрывающимися шарами нам понадобится такой шарик отдельным изображением. Можно просто вырезать любой приглянувшийся с основного источника. Мы будем постепенно добавлять новые шарики через расширение. То есть сначала будет много маленьких точек на бомбе, и затем они расширятся до видимых размеров. Добавлять точки нужно не все сразу, а по группам через определенное количество кадров. Помимо нарастания шариков, добавим самой бомбе целиком тряски. А также слегка уменьшим размер всей бомбы к финальному кадру подготовки взрыва. Это создаст ощущение сжатия перед рывком массы. Словно бомба уже не может держать в себе все эти надувающиеся пузыри.



Подготовка бомбы к взрыву

Следующим этапом создадим сам взрыв. Для этого нам понадобится пара новых изображений. Мы не будем делать покадровую анимацию в классическом понимании,

но для основных стадий взрыва нам нужны дополнительные исходники. Одно изображение для центра взрыва и одно для эффекта дыма от разрыва бомбы.



Взрыв бомбы

Внутреннюю часть можно делать мешем даже из круга заданного цвета, вытягивая из него части в стороны по направлению к радиусу окружности. Или можно нарисовать изначально произвольную форму с торчащими выступами и ее уже исказить мешем и увеличивать в масштабе. Взрыв начинается с анимации этой самой части. Этим эффектом мы создаем ощущение вырвавшейся энергии. Одновременно с расширением внутренней формы взрыва мы добавляем разлетающиеся в стороны пузыри. Предварительно не забываем отключить основной исходник, если он не был отключен на финальном этапе подготовки к взрыву. Уже тогда степень заполненности бомбы пузырями была такой, что исходник не просвечивал, и теперь его лучше выключить.

Второе изображение, которое нам понадобится, это образец дыма. Мы будем копировать несколько таких образцов и выводить их из центра взрыва с увеличением масштаба, а затем с уменьшением прозрачности. Дополнительно для создания вариативности можно наложить меш на формы и исказить их в процессе анимации. Это снизит видимость копирования одного и того же изображения. Но поскольку взрыв происходит очень быстро, не стоит слишком закапываться в детали. Главное — проработать начало и финал взрыва.

И вот, довольно простыми и стандартными средствами мы создали эффект взрыва для бомбы в игре. Его можно будет дополнить уже в движке с помощью системы частиц, добавить вокруг партиклов, свечения от взрыва и рассеивающегося дыма после уничтожения бомбы.

ГЛАВА 19

Экспорт анимаций. Форматы и настройки

В Spine очень обширные возможности экспорта и его настроек. Выбрав в меню Spine (в верхнем правом углу экрана) экспорт, вы откроете окно со всеми возможными форматами экспорта. Каждому формату соответствует своя панель настроек. Начнем с самых распространенных форматов.

GIF

Анимированное изображение представляет собой проигрываемую последовательность кадров. Оно используется в веб-среде, для портфолио или социальных сетей.

Не пугайтесь количества бегунков и кнопочек. В основном нам нужны лишь несколько из них для настроек экспорта. Остальное можно оставить в значениях по умолчанию или настроить только один раз. Плюс к тому при наведении на показатели Spine всегда даст подсказку и напомнит, что это за кнопка или ползунок.

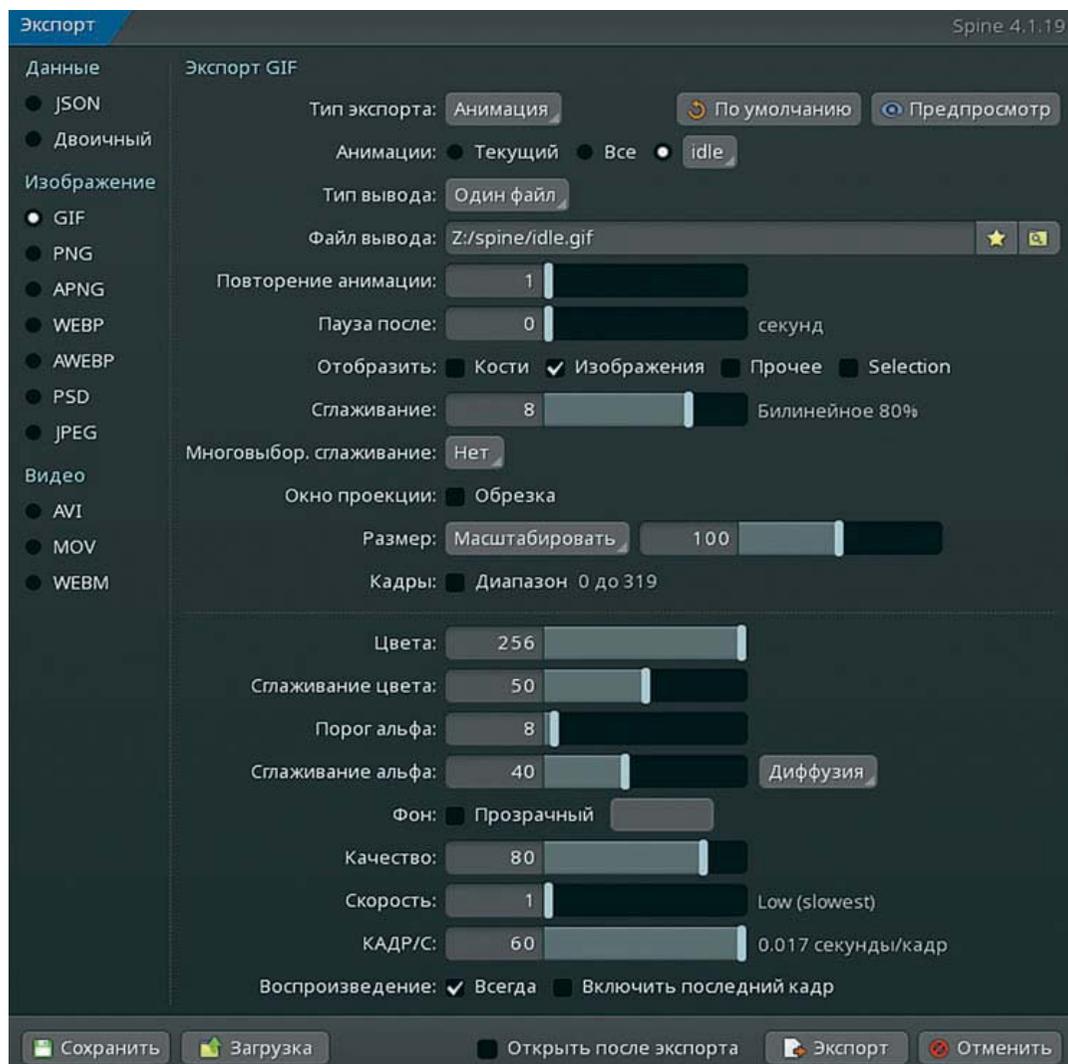
Можно включить предпросмотр результата экспорта, нажав на кнопку в верхнем правом углу окна. При этом откроется дополнительное окошко с анимацией справа. В нижней его части можно отслеживать размер и вес файла. Вес gif важен, так как многие площадки ограничивают вес файла для публикации. Обычно это 10–15 мегабайт.

Для начала в поле «Анимация» выбираем анимацию для экспорта. Можно экспортировать текущую или выбрать из списка. Если выбрать «Все», то в один файл экспортируются все анимации подряд.

Тип вывода ставим «Один файл», если нам нужна анимация в gif-формате, а не набор отдельных кадров. Второй пункт в этом меню как раз позволяет экспортировать анимацию в виде множества кадров.

В поле «Файл вывода» задаем путь для файла экспорта. В поле «Повторение анимации» ставим единицу, так как после публикации gif в интернете она зациклится автоматически. Поле «Пауза» создаст задержку между повторениями анимации.

Строка «Отобразить» очень полезна для публикации gif в портфолио. Поставив нужный флажок, вы можете отображать скелет анимации и все элементы настройки. Это используется, чтобы показать процесс работы и навыки аниматора по созданию скелета.



Настройки экспорта в gif, которые я применяю

Далее находится бегунок настройки сглаживания. Если у вас анимация пиксель-арта в Spine, то для экспорта нужно поставить значение сглаживания, равное нулю. Тогда будет использоваться метод сглаживания «По соседним», который означает отсутствие размытия, а значит, и искажения картинки. Этот же метод сглаживания используется для рисования пиксель-арта в Photoshop, к примеру.

В остальных случаях я ставлю значение, равное восьми.

Следующая настройка сглаживания «Многовыбор. сглаживание» помогает в борьбе с лесенками на изображении на его границах. Лесенки появляются в основном при поворотах изображений в анимации или обрезке границ изображения мешем.

Обрезка поможет нам кадрировать изображение на экспорт. После нажатия на обрезку в окне предпросмотра появится рамка, которая определяет область экспорта. Мы можем настроить положение анимации на холсте или срезать ненужные части анимации, которые зритель видеть не должен. По умолчанию рамка будет такой, чтобы все элементы анимации на всем протяжении движения вошли в кадр.

Строка «Размер» позволяет настраивать размер изображения. При экспорте в gif нам не подходит исходный размер, если он большой. Как я уже говорила, нам важен вес файла, а вес зависит от размера. Лучше создавать gif точно под публикацию на определенных площадках. Обычно размер больше 800 пикселей по ширине не требуется и будет избыточным. Размер можно задавать двумя способами: масштабировать исходный размер в процентах или выбрать пункт «Подогнать» и вручную задать размер по одной из сторон. Вторую сторону программа подгонит по пропорциям автоматически.

Если мы хотим увеличить размер, то должны вписать значения вручную. Рядом находятся два пункта «Увеличить» и «Заполнить».



Если поставить галочку рядом с «Увеличить», то изображение увеличится. А галочка, проставленная рядом с «Заполнить», оставит изображение в исходном размере, добавив вокруг него пустую область для достижения заданного размера.



Разница между «Увеличить» и «Заполнить»

Поставив галочку возле пункта «Кадры», мы можем задать экспорт диапазона кадров, а не всей анимации. Ниже начинается настройка максимального количества цветов

гифки и сглаживания цвета для более плавных градиентов. «Порог альфа» позволяет игнорировать прозрачные пиксели ниже значения этого параметра. Таким образом можно отсечь мусорные пиксели или ненужную прозрачную часть анимации. Далее располагается настройка сглаживания прозрачности. Тут нужно просто визуально оценивать степень сглаженности краев прозрачности анимации. Это важный параметр, и он пригодится, если у вас есть полупрозрачные эффекты или края изображений.

Сразу при экспорте мы можем выбрать цвет фона или сделать его прозрачным. Для публикации в сети лучше сразу задать фон. И ниже находятся настройки качества файла. Чем лучше качество файла, тем больше его вес.

«Скорость» влияет на скорость экспорта. Можно экспортировать быстрее, но при этом потеряется качество цветов и увеличится размер файла. Эта настройка важна при работе на слабых компьютерах, которые способны затыгивать экспорт. Или в том случае, если нужно очень быстро оценить результат экспорта.

Настройка кадров в секунду определяет плавность анимации. Для сложной, качественной анимации на детализированном исходнике лучше поставить максимальное значение, чтобы избежать размытия в движении и выжать максимум качества из формата. Если анимация простая, без деталей, или вам нужно, опять же, уменьшить вес файла, то можно поставить значение 30 или даже 24 кадра.

Галочка в самом низу в строке «Воспроизведение» определяет, будет ли анимация зацикленной в gif. Но обычно, когда вы выкладываете в интернете гифку, то она все равно зациклится после обработки площадкой. Но в просмотре в браузере или в редакторе gif будет воспроизводиться один раз.

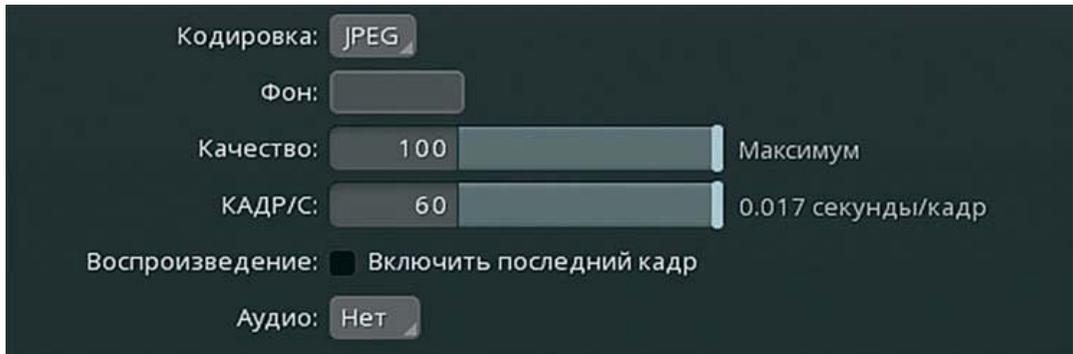
AVI И MOV

Это видеоформаты экспорта. Формат mov в основном используется в продукции компании Apple. А также этот формат часто задействуется при работе с альфа-каналом. Часто можно встретить различные эффекты для видео или игр в данном формате с прозрачностью. Это удобно для видеомонтажа, так как можно просто вставить видео на отдельную дорожку и подложить под него любой фон.

Формат avi является самым распространенным видеоформатом. А значит, он распознается большинством проигрывателей и также легко импортируется в программы видеомонтажа.

Окно настроек экспорта видео делится на две части. Верхняя содержит уже рассмотренные нами пункты в формате gif. Стоит отметить, что при экспорте видео число повторений анимации уже имеет большее значение, так как видео может использоваться в незацикленном виде. И интернет-площадки не всегда автоматически его зацикливают, поэтому необходимо задать значение, соответствующее нужной длительности видео, чтобы зритель успел оценить анимацию за время просмотра ролика.

Нижняя часть настроек уже более специфична и относится именно к видеоформату.



«Кодировка» — тип сжатия по аналогии с кодировкой изображений. Выбрав png, мы получим возможность поставить прозрачный фон. В остальных случаях мы можем только выбрать цвет фона. Кодировка в png значительно увеличивает вес файла.

Также мы можем настроить качество кодирования, тем самым определив степень сжатия. В jpeg мы настраиваем бегунок «Качество», и чем больше значение, тем лучше качество и больше размер файла. А при работе с png мы настраиваем параметр «Сжатие». И чем больше значение параметра, тем хуже качество, меньше размер файла и больше сжатие.

Ниже настраивается разное количество кадров в секунду в итоговом видео. Мы можем выставить любое количество, но лучше ориентироваться на стандарты в 24, 30 и 60 кадров. Если экспортируемое видео пойдет в монтаж, то будет учитываться количество кадров, заданное при монтаже, и ваше нестандартное значение все равно будет приведено к общему знаменателю. Плюс к тому это может произойти с размытием или потерей качества.

Галочка, проставленная рядом с пунктом «Аудио», позволяет экспортировать звук из анимации, если вы с ним работали.

ФОРМАТ WEBM

Формат для публикации в интернете. Отличается от предыдущих двух видеоформатов меньшим весом файла. Настройка качества у этого формата уникальная, отличная от avi и mov. Мы можем настроить битрейт и сжатие файла. Чем больше битрейт, тем лучше качество.

ФОРМАТЫ PNG, APNG, PSD, JPEG, WEBP

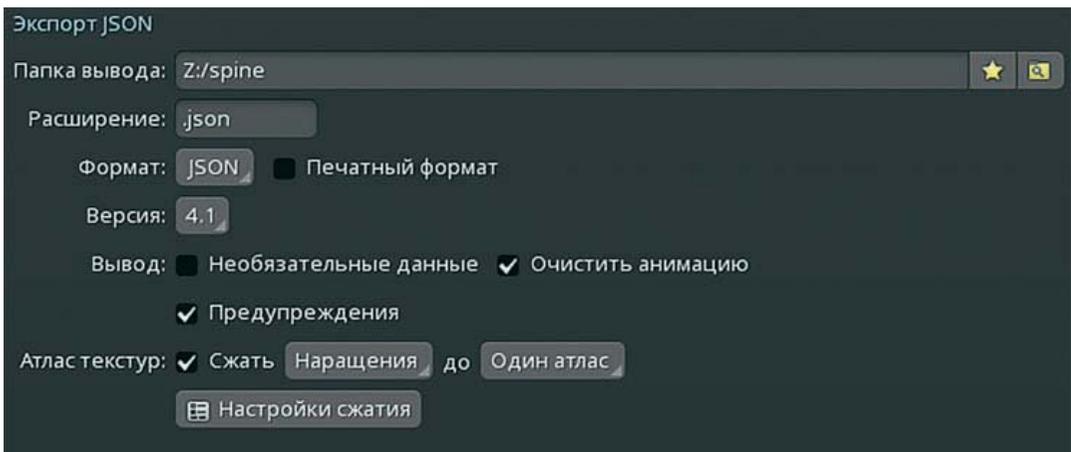
Эти форматы используются для экспорта последовательности изображений. То есть на выходе мы получаем множество картинок определенного формата,

соответствующих кадрам анимации. В случае с PSD создается файл, в котором содержится множество слоев с кадрами нашей анимации. WEBP — формат с наибольшим сжатием и наименьшим весом файла для публикаций в интернете.

APNG — аналог GIF, он весит больше, зато дает лучшее качество цветов и прозрачности.

JSON

Этот формат самый важный в дальнейшей работе с анимацией в играх. Именно он используется для импорта в движок. На выходе мы получаем файл для использования в движке. Файл можно открыть в текстовом редакторе и увидеть там описание анимации. Понять его довольно сложно, но иногда там можно отыскать нужное цифровое значение или другую информацию. Проставив галочку возле пункта «Печатный формат», мы сделаем текстовое описание читаемым, добавив пробелы и переходы на новую строку.



«Версия» указывает, для какой версии Spine у нас происходит экспорт. В Unity есть средство для работы с анимациями из Spine и поддерживается определенная версия программы. Поэтому мы должны знать, какая версия использовалась в проекте, в который мы будем вставлять новую анимацию. В проекте версия всегда одна и меняться не может. Это что-то вроде обратной совместимости. Если мы вставляем новую анимацию в старый проект, то, скорее всего, в нем использовалась старая версия Spine. Это значит, что экспорт нам нужно делать под старую версию, а не под новую.

Вывод необязательных данных увеличит размер файла. Но при необходимости мы сможем импортировать данные обратно в Spine. Это полезно, если проект был утерян, зато остался экспорт в движок.

Флажок «Очистить анимацию» уберет из вашей анимации все ключи, которые ни на что не влияют. В работе с анимацией они могут появляться, если вы не удаляли

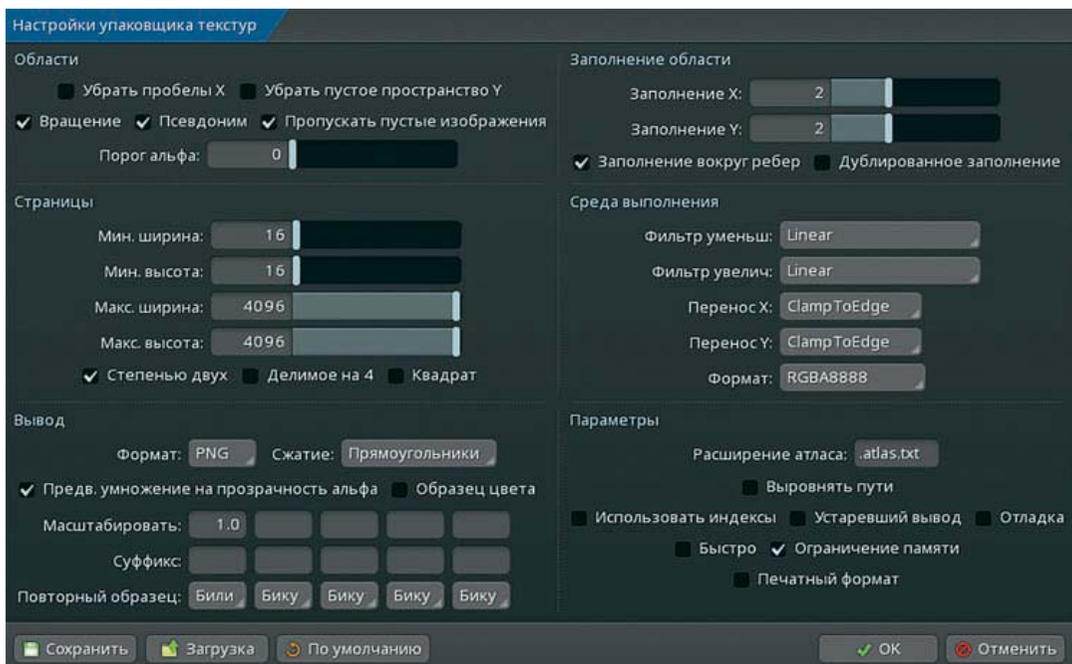
все неиспользуемые или зафиксированные заодно с другим инструментом ключи. Это действие можно делать не только при экспорте, но и во время работы. Выделив анимацию и нажав кнопку «Очистить» под окном структуры проекта, вы уберете все не влияющие на анимацию ключи.

Флажок «Предупреждения» позволяет видеть сообщения о неполадках экспорта, например, если отсутствуют какие-либо изображения в проекте.

«Атлас текстур» — важная вещь для экспорта в движок. Это изображение, которое будет содержать все кусочки исходника анимации. По сути, таким образом мы собираем все элементы анимированного изображения в одно. Нажатием флажка «Сжать» мы активируем экспорт атласа.

У нас в проекте может быть не один скелет. Мы могли собрать несколько проектов в один или создать структуру проекта так, что скелетов несколько. При экспорте атласа на каждый скелет можно сделать отдельный атлас или собрать все в одну кучу. Это регулируется выпадающим списком с пунктами «Один атлас» и «Атлас на скелет». Соответственно, «Один атлас» соберет все изображения проекта в одно, несмотря на разные скелеты. А «Атлас на скелет» создаст несколько атласов по количеству скелетов проекта.

В случае нескольких скелетов в одном проекте есть два варианта настроек экспорта атласа. Можно использовать одни настройки для всего или создавать разные виды настроек для разных папок с изображениями. Выбрав пункт «Нарращения», вы примените настройки сжатия (кнопка с которыми находится ниже) для всех изображений. «Папка изображений» позволит создавать разные настройки сжатия для папок.



Нажмем кнопку «Настройки сжатия». Перед вами появится окно с настройками, похожее на панель управления самолетом. Но не пугайтесь, сейчас мы во всем разберемся. Опять же, при каждом экспорте вам не нужно будет настраивать каждый пункт.

В первой строке располагаются два пункта — «Убрать пробелы X» и «Убрать пустое пространство Y», они кадрируют изображения по соответствующим осям, если есть пустоты. «Порог альфа» определяет, какую прозрачность учитывать для кадрирования. Галочка, проставленная возле пункта «Вращение», позволяет вращать изображения на 90 градусов для более плотного их расположения. Галочка рядом с пунктом «Псевдоним» делает так, что повторяющиеся изображения будут присутствовать в атласе только в одном экземпляре. Галочка возле «Пропускать пустые изображения» позволяет экспортировать только те изображения, в которых есть хотя бы один пиксель, равный порогу альфа или выше него.

Важными параметрами в этом окне являются «Максимальная высота» и «Максимальная ширина». Уменьшая эти параметры, мы будем вынуждать алгоритм программы собирать изображения в атласе плотнее. Если места под изображения не хватит, то программа выдаст несколько атласов. Флажки снизу задают ограничения для значения высоты и ширины. Алгоритмы движка могут работать как раз с тремя видами атласов: «Степенью двух», «Делимое на 4» и «Квадрат». Галочку нужно проставлять в зависимости от того, какой алгоритм используется в движке для работы с атласами.

Ниже мы выбираем формат изображения и тип сжатия. В списке параметра сжатия у нас три вида.

Пункт «Прямоугольники» просто складывает изображения в исходном виде. В атласе может остаться много пустых областей.



Пункт «Полигоны» кадрирует изображение максимально близко к контуру. В этом случае атлас получится очень сжатым и плотным. Это смотрится аккуратно и красиво, но не всегда удобно для работы в движке, а иногда даже и невозможно. Из-за

произвольного плотного расположения вырезать нужную часть в движке из такого атласа будет затруднительно.



«Сетка» активирует создание универсальной сетки на основе всех изображений и затем каждое изображение кладет в ячейку. Обратите внимание, что размер ячейки выбирается по максимальным значениям изображений по осям. Шляпка гриба имеет максимальную ширину и задает ширину ячейки. Ножка гриба имеет максимальную высоту и задает высоту ячейки. А остальные изображения, несмотря на меньший размер, помещаются в ячейку с максимальной шириной и высотой.

Переходим в следующий правый столбец настроек. Функция «Заполнение области» добавляет пробелы между изображениями. Значения мы указываем по разным осям в пикселях с помощью бегунка. Галочка, проставленная возле пункта «Заполнение вокруг ребер», задает пробелы рядом с границами атласа. Например, между изображениями у нас пробел в 2 пикселя, а между изображениями и границей холста будет пробел в 1 пиксель.



Галочку возле пункта «Предварительное умножение на альфа» нужно ставить, если аналогичный параметр включен в движке, в котором вы будете далее работать с атласом.

Строка «Масштабировать» позволяет экспортировать атласы разного масштаба по отношению к исходному. Это может понадобиться в том случае, если мы работали с большим форматом изображений, но в игре такой размер не нужен. А значит, и экспортировать огромные изображения не нужно. В игре мы будем работать с уменьшенной версией. Но при этом у нас останется проект с максимальным качеством и размером изображений. Несколько атласов разного масштаба может понадобиться, если персонаж в игре показан и в отдалении, и в приближении. Строка «Суффикс» позволяет каждый файл с определенным масштабом назвать по-своему, чтобы не путаться в атласах после экспорта и не переименовывать их вручную. Программа сама допишет к основному названию проекта заданный суффикс. И последняя строчка позволяет выбрать алгоритм масштабирования для каждого атласа.

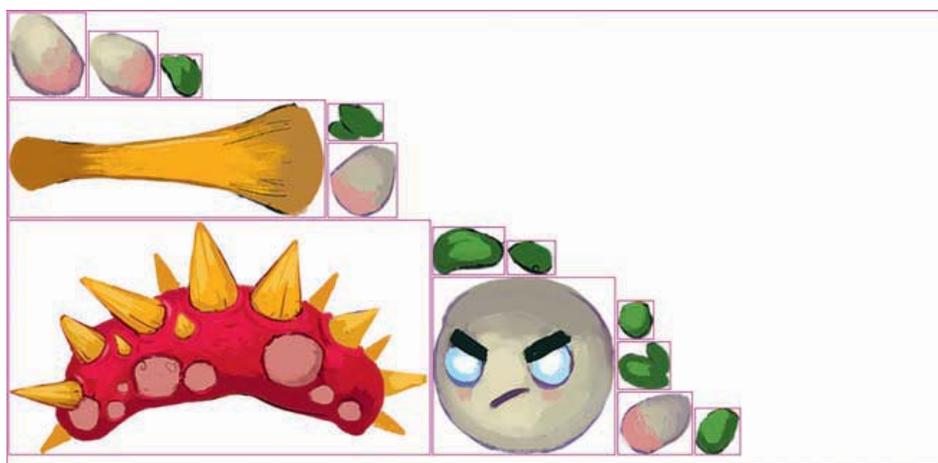
Далее в панели «Среда выполнения» находятся технические характеристики экспорта, которые можно оставить по умолчанию.

И последняя панель — «Параметры».

Расширение атласа по умолчанию — .atlas. Для экспорта в Unity нам нужно дописать расширение .atlas.txt.

В остальных пунктах галочки можно не проставлять в большинстве случаев. Разве что нужно обратить внимание на пункт «Ограничение памяти». Поставив галочку возле него, вы обезопасите себя от вылета программы или зависания в процессе экспорта из-за нехватки оперативной памяти.

Галочка в пункте «Отладка» выдает технические сообщения и показывает границы изображений в атласе, что бывает полезно для оценки качества сборки атласа.



ГЛАВА 20

**Создание
анимационного
ролика
с помощью
Spine 2D и Unity**

Основной алгоритм работы

Мы много говорили о том, как Spine применяется в играх. Но программе можно найти применение и в других сферах. Например, в создании видеороликов. Если снова затронуть тему игровой индустрии, то нужно сказать, что есть сфера околоигровых задач. В основном это создание различных промоматериалов. Рекламные ролики требуют анимации, порой сделанной отдельно. За счет этого они приобретают максимальную привлекательность для зрителя. Это могут быть смешные вставки с героями из игры или даже отдельные мини-истории. Самой важной частью рекламных материалов являются трейлеры. И вот там как раз и встречаются анимационные вставки или даже целые отдельные анимационные ролики по теме игры. Само собой, анимационные видеоролики применимы не только для рекламы игр, но и в целом для рекламы. Они могут существовать и как отдельные творческие проекты.

В Spine можно создать ролик, практически не выходя за пределы программы. Если можно создать анимацию иллюстрации, значит можно сделать и ролик. За исключением монтажа, конечно. Когда вы сделали несколько сцен, вам нужно смонтировать их в специальной программе для монтажа. Все-таки не стоит делать задачи в программе, которая для этого не приспособлена, даже если такая возможность в ней есть. Теоретически мы можем соединить все сцены в Spine на временной шкале, но это очень неудобный вариант работы. Поэтому для монтажа готовых сцен лучше использовать такие программы, как Adobe premier или Sony Vegas. Плюс к тому вы можете делать в Spine не все сцены, вероятно, что-то у вас снято на видеокамеру или сделано в 3D. Также анимационные вставки в трейлерах сочетаются с видео из игры. Так что монтаж — отдельная сфера, которую не стоит делать в Spine.

Второй задачей, которую нужно делать вне Spine при создании роликов, является работа с камерой. В видео не приветствуются статичные кадры в плане ракурса. Да, мы можем анимировать сам кадр, но задать движение камеры в пространстве в Spine практически невозможно. Лучше для этой задачи использовать программы, где камера присутствует как отдельный объект, или настраивать камеру при монтаже. Выбор программы для движения камеры относительно кадра может быть любым, главное, чтобы была возможность управлять камерой. Я предпочитаю работать с камерой в игровом движке Unity. Это позволяет управлять не только камерой, но и созданными анимациями, настраивать слои параллакса, свободно двигать персонажей

Созданию самого ролика предшествует пара этапов разработки. Необходимо придумать персонажей и локации для ролика, сделать концепты и продумать анимации для них. Также ролик нужно продумать в виде текстового описания. Мы должны написать общий сюжет, а также подробнее расписать, что хотели бы видеть в каждой сцене: какие движения и анимации, какие эмоции должна вызывать каждая сцена и ролик в целом, какая у него задача и какое впечатление мы хотим произвести на зрителя. По сути, нам нужно написать техническое задание и создать базу из концептов под наш ролик.

Далее нужно нарисовать все кадры в виде раскадровки ролика. Сначала это делается в виде очень быстрого скетча, без детализации. Это позволяет на раннем этапе оценить весь ролик и то, как кадры передают динамику повествования и сюжет в целом. Поскольку концепты персонажей и локаций у нас уже были, то мы сразу перешли к раскадровке.

Дальше уже идет техническая реализация задуманного. Поэтому важно проработать техническое задание окончательно, не переходя к дальнейшим пунктам. Иначе придется менять что-то уже в сделанном материале или отказываться от проделанной работы. После итоговой коррекции идеи и кадров ролика я перешла к отрисовке каждого кадра. Стиль отрисовки подобной анимации может быть довольно разным. Это зависит от требуемой анимации, возможных затрат времени и бюджета ролика. Если кадры будут представлять собой анимацию покоя или небольшого движения персонажа, то стиль изображений может быть довольно реалистичным и детализированным. Если же движения значительны, то используются все те же ограничения стиля, как и для анимации объектов в Spine. Плюс к тому нужно учитывать, что чем детальнее исходник, тем сложнее создается анимация и тем более аккуратный и кропотливый подход требуется к ее проработке. А значит, уйдет больше времени. Исходя из этого, я выбрала довольно мультяшный, но красочный стиль для ролика, чтобы уложиться в разумные сроки. На создание видео ушло чуть больше месяца. В условиях маленькой команды это довольно значимый срок. После первого этапа у нас получился набор кадров для будущей анимации, последовательное изменение которых расскажет историю. В создании финальной раскадровки также важно уделить внимание общей цветовой палитре. Необходимо смотреть не на каждый кадр по отдельности, а оценивать их единым полотном, чтобы понимать общую атмосферу происходящей истории. Нельзя менять цвет в каждом кадре как минимум потому, что действие происходит порой в одном и том же антураже. При частой смене палитры не будет ощущения целостности происходящего. Поэтому у меня кадры делятся на две части. Первая, светлая и радужная, палитра добавляет ощущение сказки, а темная гамма синих оттенков — ощущение опасности.



Нарезка кадров под анимацию

Как и в случае с отдельным объектом, кадр нужно нарезать для анимации. И как и при работе над анимацией иллюстрации, нужно поделить кадр на отдельные объекты. Сначала анимируем отдельно, затем собираем в Spine или в Unity. В Spine будем совмещать элементы, когда потребуется их взаимодействие друг с другом. Специфика нарезки заключается в том, что нам нужно обязательно разделить кадр на планы. Иллюстрация в плане работы с камерой в большинстве случаев остается статичной. А кадр ролика, скорее всего, будет с пролетами камеры, а значит, нам понадобится эффект параллакса. Это эффект разницы скорости движения переднего и дальнего плана. Чем ближе план к зрителю, тем быстрее он движется. Сразу стоит обратить внимание на одинаковые или очень похожие элементы в кадре. Одинаковые элементы оставляем в единственном экземпляре. Очень похожие могут быть сделаны с одинаковым скелетом. В моем примере одинаковые цветочки будут сделаны копированием одного анимированного цветка со смещением фазы анимации. Кусты на заднем плане могут покачиваться с помощью одной и той же анимации, несмотря на то что сами изображения кустов слегка отличаются. Также множество одинаковых листьев на переднем плане лучше делать копированием одного-двух вариантов анимаций.

Несколько вариантов требуется в случае большого числа копий, чтобы копирование не бросалось в глаза. Одинаковая анимация может быть заметна даже с учетом смещения фазы движения.

Каждый отдельный объект также подлежит нарезке. Тут важно не нарезать избыточно. У нас очень много отдельных изображений в сцене. В таком случае порой лучше понадеяться на меш, а не на нарезку, чтобы контролировать всю сцену было чуть проще.



Пример нарезки кадра на планы и элементы

Поскольку предстоит работа с камерой, для ее движения нужно дополнительное пространство. Кадр всегда должен быть несколько шире, чем итоговый ракурс. Вдруг мы решим сузить или расширить кадр или переместить камеру в сторону. В таком случае утыкаться в жесткие границы, за пределами которых нет изображения или оно обрезано, будет неприятно. Это очень ограничит работу с ракурсом. Если сам фон можно особо не расширять без необходимости сильного движения камеры, то объекты за пределами ракурса должны быть максимально продолжены. Они не могут обрываться на границе камеры, поскольку при анимации, скорее всего, определенные фрагменты элемента будут двигаться так, что часть из-за границ будет видна. В моем примере нарезки обратите внимание на лепестки цветка. Они прорисованы полностью, несмотря на то что скрыты кадрированием.

На исходном кадре есть эффект тумана. Его можно сделать несколькими способами, например, с помощью системы частиц в движке. В этом случае нам понадобится образец частицы, который мы будем копировать множество раз и пускать в виде тумана. Обычно этот образец аналогичен отпечатку кисти, которой вы пользовались для рисования тумана в Photoshop. Главное, чтобы у образца не было обрезанных границ и все изображение плавно уходило в прозрачность по краям. Недостаток такого

подхода в том, что прозрачные образцы могут накладываться друг на друга и в месте их наложения будут видны границы образцов. Можно использовать существующие в движке партиклы и за счет их настройки создавать похожий эффект. Но тогда он будет меньше похож на то, что у нас в исходном кадре. И для создания интересных эффектов в системе партиклов движка нужно очень долго разбираться. Это сложная система с множеством настроек.

Можно делать эффекты в Spine с помощью меша и движения образца по заданному пути. В этом случае для вариативности эффекта в качестве исходника нам понадобится несколько образцов тумана.

В любом кадре есть основной элемент. Это может быть персонаж или любой другой объект, который больше всего двигается и является базовым в кадре. В нашем примере это фея в цветке. Именно основному элементу нужно уделить максимальное внимание, а значит, и нарезать более детально. Для центра композиции нужна самая проработанная анимация, чтобы привлечь внимание зрителя к нужному элементу.



Пример нарезки персонажа для кадра анимации

Я не стала выделять каждый локон волос или лепесток воротника или юбки отдельным спрайтом. Отрезаны только основные части, которые выделяются на силуэте или создают заметные перекрытия. Для остальных групп мы будем делать меш и анимировать с его помощью. То есть будем сочетать нарезку и меш, чтобы получить

наилучший результат с минимумом усилий. Да и контролировать такое количество элементов в анимации попросту было бы неудобно.

Настройка скелета

Весь исходник должен быть разделен на части как минимум по количеству планов, для того чтобы создать ощущение объема пространства. К плановости отдельно добавляются персонажи и некоторые дополнительные элементы. Все планы и персонажи настраиваются и анимируются по отдельности, а потом собираются в движке, так как именно в движке мы будем работать с камерой. Не обязательно каждый кустик выделять на отдельный план, зачастую достаточно группы элементов, которые будут значительно удалены от другой группы. Рассмотрим пример кадра с цветком и феей.

Этот кадр поделен на шесть сегментов. Персонаж, цветок и три плана растительности сзади плюс заливка голубого неба на самом дальнем плане. Но небо считать планом не будем, так как там нет никакой анимации. Мы просто в движке на задний план вставим прямоугольник нужного цвета.

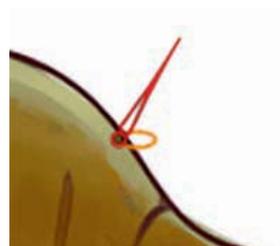
Самым сложным элементом кадра является персонаж. Именно персонаж содержит наибольшее количество изображений, а значит, и костей, и мешей. И больше всего времени занимает его настройка и анимация. Анимация феи представляет собой парение над цветком. Все отдельные части совершают волновые движения, покачиваясь в такт основному с определенным запаздыванием. Таких элементов много: пряди волос, усики, уши, руки, лепестки воротника и юбки. Для каждого такого элемента, в зависимости от его длины, мы делаем группу костей от одной до трех. И этими костями мы будем создавать волну в анимации. Поскольку мы не нарезали каждую прядь волос и каждый лепесток, то меши будут общими, согласно разделению на изображения. Но кости нам все равно нужны для каждой пряди и лепестка. В целом этой настройки более чем достаточно, и она повторяется от элемента к элементу. Поэтому при визуальной сложности и загроможденности костями настройка довольно проста в создании и понимании ее структуры. Вся сложность только в количестве элементов и костей, что означает появление большого количества ключей и графиков, в которых главное — не запутаться. В таких случаях очень важно поддерживать структуру проекта, называть кости понятными именами и группировать их цветом. Также не забывайте использовать фильтры отображения, чтобы видеть только то, что нужно, а не кашу из костей и ключей.

Следующий отдельный элемент — это цветок. Причем его может понадобиться разделить на две части из-за плановости. Фея находится посередине цветка. Это значит, что задние лепестки она будет перекрывать, а передние лепестки, наоборот, будут перекрывать фею. Из этого следует, что для создания анимации появления феи из цветка или ее захода на передние лепестки в процессе парения и покачивания цветок должен быть поделен на два плана: план перед феей и за ней.



Настройка скелета феи

Что касается настройки костей, ничего уникального в цветке нет. В целом схема схожа с настройкой феи. Мы создаем линии из костей для ближних лепестков и торчащих усиков для волновой анимации. Для дальних лепестков достаточно добавить по одной кости, так как они уже частично перекрыты передними лепестками и феей, а значит, их анимация будет не видна. Движение дальних лепестков можно делать довольно условным, так как они скрыты другими элементами. Для анимации покачивания лепестков лучше добавить небольшую траекторию движения, чтобы они покачивались не просто вверх-вниз, а по окружности или восьмерке. Так движение будет намного более естественным и похожим на реальное покачивание растения на ветру.



Пример круговой траектории для лепестка

Всю остальную растительность за феей делим на три плана, которые отделены друг от друга воздушной перспективой. Дополнительно между этими планами уже в движке можно добавить тумана или свечения. Для каждого из этих планов также делаем круговую траекторию и один общий меш. Делить на отдельные кустики дальний план не нужно, можно обойтись единым мешем и деформировать его с помощью кости (или нескольких) и траектории. Основное внимание зрителя будет сосредоточено на персонаже и переднем плане. Не нужно детализировать задник. Не только из-за экономии времени, но и из-за того, что чрезмерная детализация анимации может отвлечь внимание на себя. А нам нужно максимально удерживать взгляд зрителя на композиционном центре. В данном случае это персонаж.

Анимация

После довольно утомительной настройки скелета для всех частей кадра сама анимация уже не кажется сложной. Мы не зря создавали одинаковые линии костей для каждой пряди волос. Теперь, чтобы создать анимацию, нам хватит буквально нескольких шагов.

Вся анимация как персонажа, так и растительности строится на алгоритме волновой анимации в данном кадре. Мы должны задать основное движение персонажа вверх-вниз. Затем необходимо настроить движение всех элементов персонажа, согласно основной траектории и гравитации, а далее добавить запаздывание элементов. Даже если бы у нас присутствовало основное движение, отличное от простого покачивания, мы бы все равно сначала делали его, а затем все остальное по типу запаздывающего движения элементов. В анимации иллюстрации и кадров ролика важно создать ощущение воздуха. Мы практически не можем позволить себе оставить статичными какие-либо части. За исключением реально не двигающихся объектов. Волосы, ткань, растения — все покачивается на ветру и в такт движениям.

В кадре с феей есть две области, которые будут двигаться слегка по-разному. Персонаж двигается вверх-вниз, паря над цветком. А все остальное колыхается на ветру. Нам нужно выбрать направление ветра и периодичность его дуновения. Лучше не делать эти дуновения одинаковыми и повторяющимися через одно и то же время. Но этим мы можем заняться уже в движке, когда будем собирать все части в одно целое. Изначально хватит даже минимальной анимации в один цикл. А периодичность настроим уже в Unity. Правда, лучше сделать несколько вариаций сильно и слабо покачивающихся растений, чтобы было что чередовать при сборке. Для создания промежуточных состояний покоя тоже нужна анимация, чтобы картинки между дуновениями ветра не были статичными. Вы ведь наверняка замечали, что травинки или листья на деревьях очень сложно заставить без движения. Вот и мы в анимации не можем отклоняться от этого правила. Даже для состояния покоя нужна анимация.

Поскольку мы анимируем части изображения по отдельности, то нужно учитывать длину анимации всех частей, чтобы не создать слишком разные по скорости анимации. Да, мы можем корректировать скорость при сборке, но это скажется на качестве анимации. Если вы делаете анимацию в медленном формате, то, скорее всего, у вас будет множество мелких движений и нюансов анимации. А при ускорении эти детали в лучшем случае просто потеряются, а в худшем еще и превратят анимацию в кашу и дадут скачки при движении. Если же сначала сделать анимацию очень быстрой, а затем попытаться растянуть ее по длительности, то новых деталей движения, которые нужны при медленном формате, не появится. Поэтому нужно стараться хотя бы примерно попадать в требуемую скорость. Также необходимо учитывать скорость других частей кадра, чтобы они проще собирались в единое целое.

С анимацией окружения разобрались. Вернемся к персонажу. Помимо покачивания над цветком, у персонажа есть дополнительные анимации: моргание, движения руками и ногами, подергивания ушами. Эти анимации не встраиваются в цикл парения, не подчиняются основной анимации. Они не синхронизируются ни друг с другом, ни с базовым движением. Поэтому их нужно сделать отдельными анимациями, которые мы сможем просматривать на разных дорожках. А в движке мы уже асинхронно вставим их в общую канву анимации кадра.

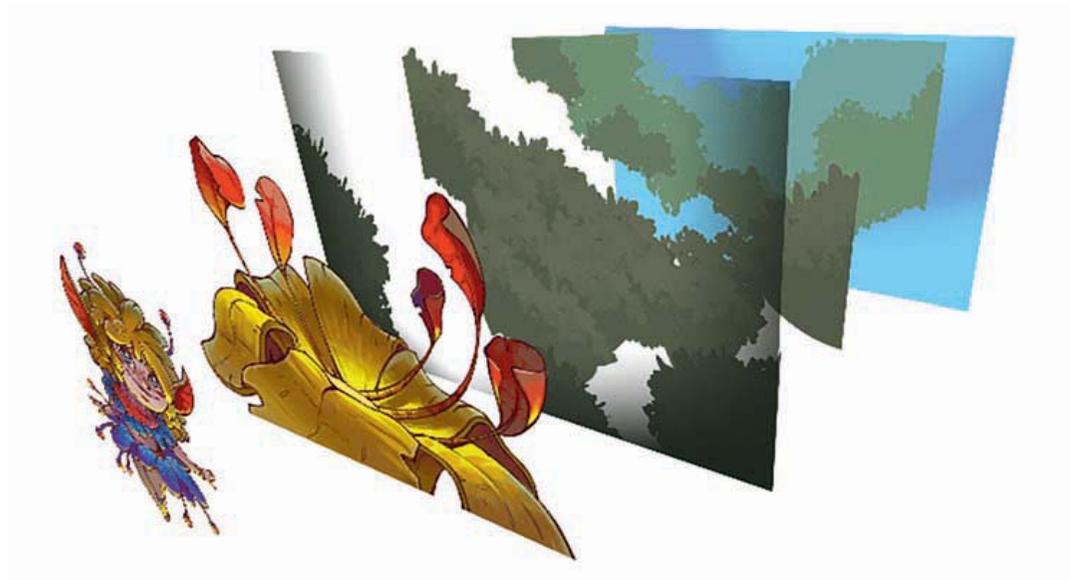
Сборка

Поскольку в рамках этой книги мы не изучаем работу с конкретным движком и его интерфейс, я лишь в общих чертах объясню, как мы собираем кадр и какую работу проделываем на данном этапе.

Сначала нам нужно собрать все планы и элементы в одну картинку. В Unity у нас есть два варианта сборки — с использованием 2D-формата сцены или без. Отличие в том, будем ли мы видеть 3D-вид сцены или нет. В случае использования 2D-формата все картинки схлопнутся в одну плоскость. За расположение картинок относительно друг друга по слоям будет отвечать специальный числовой параметр. А в случае с 3D-видом мы просто распределяем плоскости по разным значениям оси Z, которая направлена на зрителя. Если сегментов немного, то этот способ более наглядный и удобный, так как позволяет видеть слоистость сцены. Если элементов мало, то эта 3D-слоистость не мешает в работе. А при работе с видео, скорее всего, слоев не очень много в кадре. А вот при разработке 2D-игры уже предпочтительнее использовать 2D-формат, потому что элементов много и 3D-вид просто перестает быть информативным из-за количества планов.

Анимации собираются воедино в специальной части движка для работы с анимациями. В аниматоре Unity мы можем регулировать скорость и выставлять анимации вручную друг за другом. Если нам нужно случайное воспроизведение анимаций,

например, для постоянно меняющегося колыхания на ветру, то необходимо писать скрипт. Скрипт — это программа, которая будет управлять воспроизведением анимаций и случайным образом воспроизводить варианты. Но для этого нужно уметь программировать. Даже если скачать готовое решение, все равно нужно понимать, как его использовать. Чтобы обойтись без программирования, нужно вручную выставить наши вариации движения в длинную последовательность. Но в отличие от полной случайности, выставленная последовательность все равно будет повторяться. Когда кадр на экране задерживается долго, даже такое повторение бывает заметно.



При создании вариаций движения важно, чтобы все анимации обладали идентичным началом и концом не только внутри каждой анимации, но и между собой. Они должны бесшовно стыковаться в одно целое. Если этот момент нарушен, то в движке можно создавать автоматические переходы. Но иногда автоматика нарушает целостность и плавность анимации. Поэтому лучше делать так, чтобы переходы не требовались.

Работа с камерой

Камера очень важный инструмент в работе с роликом. С ее помощью можно значительно улучшить видеоряд, но можно и испортить. Нужно подходить к анимации камеры со всей серьезностью и не игнорировать этот этап работы с видеороликом. Камера в движке — отдельный объект, который мы можем двигать и поворачивать, тем самым меняя ракурс. Этот инструмент применяется для создания динамики ролика.

Камеру можно использовать для акцентирования внимания на определенном элементе кадра. Причем скорость движения камеры к акценту будет определять степень акцентирования. Чем быстрее мы «посмотрим» камерой на объект, тем больше внимания это привлечет.

Также можно добавлять камере эффектов от происходящего на экране, например, трясти камеру при взрыве. Поскольку зритель ассоциирует камеру со своим взглядом и самим собой, такие эффекты способствуют погружению в события ролика.

Движением камеры можно просто разбавить статичность кадра. Мы можем добавлять минимальные движения вбок или вперед для того, чтобы ракурс кадра не был статичным. Главное — не переборщить и не создать для зрителя «буэ-эффект» из-за постоянного движения камеры.

Мы можем даже добавить эффект съемки и наличия оператора, например, создать эффект бега при быстром перемещении или заставить несуществующего оператора оглядываться по сторонам от страха.

В нашем ролике мы в основном добавляли медленное движение камеры к главному элементу кадра. Это создало акцент и вместе с тем придало ролику динамики. Но поскольку движение медленное, зрителю ничего не мешает воспринимать информацию из видео.



Только в одном случае мы сделали акцент на важном моменте с помощью резкого движения камеры ближе к композиционному центру, так как нам нужен был эффект неожиданности и даже страха. Это кадр, в котором ведьма Потата срывает цветок. С точки зрения сюжета этот момент является ключевым во всем ролике. Поэтому логично именно на нем максимально акцентировать внимание быстрым движением камеры.

И тут можно обратить внимание, что композиционный центр есть не только в каждом кадре, но и в ролике в целом. Поэтому средства выразительности нужно распределять так, чтобы они работали для целого ролика. Необходимо учитывать сюжет ролика

и его смысл и максимальными средствами подчеркивать главное во всем видео.

С камерой можно работать и в программе монтажа видео. Но в движке мы имеем больше контроля над камерой. Можно создать несколько камер и переключаться между ними во время пролета, а также накладывать различные эффекты, которые будут влиять на отображение кадра, на саму камеру. Если у нас исходник большого разрешения, то при приближении к нему не будет сразу теряться качество. Таким образом, мы сможем управлять разными ракурсами без проблем с размытием картинки. Даже если мы имеем дело с пикселями, в движке мы можем воспользоваться специальным режимом камеры «Pixel Perfect», который позволяет приближаться к пиксель-арту, не размывая его. Если у нас исходник видеоформата, то возможности работы с камерой при монтаже довольно ограничены. Видео вряд ли будет разрешением больше 4К, а значит, возможности приближения у нас ограничены буквально двойным увеличением.



Потата срывает цветок, и на лес обрушивается гнев злой ведьмы. Это главный момент всего ролика

Видео из движка

В движке мы создали сцену, но нужно получить видео для дальнейшего использования в программе монтажа. В Unity есть специальные плагины для записи и экспорта видео с камеры сцены. Но даже без них мы всегда можем просто создать билд-сцены. Запустив сцену с .exe-файла, мы запишем видео с экрана. Однако в ряде случаев это приведет к потере качества при записи с экрана или нарушению передачи цветов. Причина в том, что при записи делаются свои настройки качества, а в самой сцене отображение может быть другим. Продельвая такой алгоритм для каждого кадра, мы создаем набор видеороликов, которые корректируем уже в программе монтажа. Там мы создаем максимально целостный видеоряд и работаем с постэффектами картинки. В движке тоже можно заниматься различной коррекцией цвета экспозиции и других параметров. Но при монтаже мы можем легко набрасывать подобную коррекцию на все фрагменты разом. Это удобнее, если нужно корректировать все фрагменты одновременно. В движке у нас все еще нет целостного ролика, а есть только его части.

ГЛАВА 21

Что дальше? Как развиваться аниматору

Вот вы ознакомились с программой Spine 2D, выполнили базовые и более сложные упражнения. Вы чувствуете в себе прилив новых сил и думаете, что же дальше?

Прокачиваем навыки

После того как вы прочитали книгу, выполнили упражнения и даже пару работ для портфолио, нельзя останавливаться. Если мы откладываем определенное занятие на долгое время, то навыки притупляются и забываются. Вспомним трудности с письмом после летних каникул в школе. Каждый раз после долгого перерыва вам придется тратить какое-то время, чтобы освежить знания и умения. Регулярность занятий — главный фактор постоянного продвижения в профессии. Для начала я советую закрепить материал этой книги и спустя недельку еще раз проделать основные практические примеры, но с другими исходниками. Это выведет полученную информацию из кратковременной памяти. Так же лучше поступать и с любыми курсами и задачами: повторять через какое-то время заново. Заодно накопятся и работы для портфолио.

Для оттачивания навыков очень важна практика. Многие бесконечно сохраняют видеоуроки в плейлист, а потом не смотрят их. А если все-таки смотрят, то вряд ли делают что-то самостоятельно по прослушанной теории. Даже если вам кажется, что, посмотрев ролик, вы все поняли и практика вам не нужна, это не так. Это ложное впечатление, которое развеется, если вы приступите к практической работе по данному уроку. А уж если решите слегка отступить от урока, то проблем станет еще больше. Просмотр теории важен и нужен, но всегда должен сопровождаться практикой. Времени на практику должно быть затрачено больше, чем на теоретический материал. Нужно набить руку, довести до автоматизма определенные действия. Вы должны чувствовать себя в интерфейсе программы как рыба в воде. В любом случае какие-то неиспользуемые фишки будут забываться, но постоянная практика не даст забыть основы. Запоминание будет происходить глубже, на уровне рефлексов. Сложно разучиться кататься на велосипеде, если в детстве каждый день объезжал весь двор.

В обучении необходимо выстроить четкий план и иерархию задач. Не нужно сразу браться за сложные составные анимации. Начните с простых мультяшных персонажей и объектов, шаг за шагом переходя к исходникам с большим количеством элементов. Постепенное увеличение количества сегментов в анимации поможет вам

научиться контролировать любое количество объектов и создавать целостное движение из множества составляющих. Затем можно перейти к более сложным ракурсам и 3D-эффекту в разных его представлениях, сделать поворот персонажа, удар мечом с замахом, эмоции. После оттачивания приемов на отдельном объекте можно перейти к анимации сцен. Хаотичность обучения создаст пробелы в знаниях, которые будут тормозить вас и постоянно напоминать о себе при более сложных задачах.

Портфолио. Советы

Первое, что нужно аниматору для начала работы, это портфолио. В творческой среде редко требуют красный диплом об окончании университета по нужной специальности. Основа успеха для специалиста — это его портфель работ. Ну и резюме с опытом, конечно, но не все сразу. Нужно с чего-то начинать.

Портфолио должно состоять из нескольких анимаций определенной тематики. Обязательно иметь огромное количество работ, вряд ли вашу страничку будут просматривать от и до, заглядывая в каждую вкладку и оценивая каждую анимацию. Обычно по первым 3–5 работам становится понятен ваш уровень и стиль работы. Поэтому советую не напирать на количество, а сосредоточиться на качестве. В то же время единичные работы могут отпугнуть заказчика, так как создастся впечатление, что вы только-только начали изучать анимацию. Остановимся на 8–10 работах в портфолио — этого должно быть достаточно для того, чтобы произвести впечатление на работодателя.

При создании портфолио лучше сразу определиться, в какую сферу и даже в какую компанию вы хотите попасть. Исходя из этого, нужно выбирать стиль и тип работ, которые вы положите в портфель. Если нужно получить работу аниматором в компании, занимающейся разработкой мобильных игр, то и работы должны попадать в стиль подобных игр. И задачи, выполненные в анимациях, должны совпадать с основными рабочими задачами в подобных компаниях. В портфолио не стоит класть учебные работы и упражнения. Нужно посмотреть, какие анимации есть в необходимых вам проектах, и делать подобные работы для портфеля. Если специалист по подбору кадров увидит на вашей странице реалистичные анимации пейзажных сцен, он не пригласит вас анимировать милых персонажей в казуальной ферме.

Желательно, чтобы портфолио было нацелено на определенные задачи и не было разношерстным. Если у вас присутствуют разные стили и виды анимаций в портфолио, нужно сгруппировать работы. Тогда работодатель сразу выберет нужную для него плашку и посмотрит работы только этого типа.

Итак, допустим, мы решили попасть в мобильный геймдев и хотим подготовить хорошее портфолио для устройства на работу. Необходимо проанализировать, какие

анимации нужны в данной сфере, какие стили наиболее распространены и популярны. В мобильных проектах есть три большие группы анимационных задач: анимация персонажей, анимация интерфейса и иконок, анимация объектов и среды. Также во всех видах анимации нужны эффекты. Вот такие группы работ и должны быть в нашем портфолио. Основным казуальным стилем можно выделить мультяшный яркий вид объектов с пухлыми формами и сжатыми утрированными пропорциями. Значит, нам нужно подбирать подобные исходники.

Рассмотрим отдельные задачи, которые смогут выгодно представить вас как аниматора.

АНИМАЦИЯ ПЕРСОНАЖЕЙ

Этот вид задач делится на несколько типов. Первый — это **передвижение персонажа**: ходьба, бег, прыжок, подкат. Эти анимации являются основными для жанров квеста, платформера или раннера. Везде, где герой виден, двигается и создается с помощью 2D-графики, нужно много анимаций, чтобы это передвижение показать. Анимация данных базовых движений персонажа должна быть в портфеле у каждого игрового аниматора.

Эмоции персонажа. Практически во всех жанрах встречается персонаж-помощник или просто сопровождающий герой. Портреты и фигуры таких персонажей могут появляться в интерфейсе и дополнять их содержание. Основные эмоции — это радость, грусть, удивление. Например, в окне победы может быть герой, который явно радуется нашему выигрышу. В окне поражения или в случае, если игрок ничего не купил в магазине, персонаж может грустить. Для акцентирования внимания на каком-то важном элементе персонаж будет показывать на него и удивляться. Могут быть и другие эмоции, особенно если в игре есть диалоги. Тогда спектр задач по созданию эмоций еще больше расширится. В создании эмоций аниматор показывает свои навыки работы с мешем и эффектом псевдо-3D.

Атакующие приемы. Для многих жанров нужно создавать атакующие анимации: для экшенов, файтингов. Да и на ферме порой придется взмахнуть топором. Причем герой может и стрелять, и драться врукопашную, и использовать магический посох. Зачастую нужно несколько анимаций на каждый вид атаки, чтобы постоянно не повторять в игре одно и то же. Разнообразие частых движений персонажа важно для создания естественной анимации.

В портфолио отлично будет смотреться проект, где один персонаж будет анимирован по всем вышеперечисленным пунктам. Нужно сделать герою анимацию бега, атаки и отдельно выделить портрет для пары эмоций. Такая подборка будет выглядеть реальной рабочей задачей. Данный набор анимаций вполне можно использовать в игре. Подобным проектом вы покажете все навыки на данную тему и заинтересуете заказчика.

АНИМАЦИЯ ИНТЕРФЕЙСА И ИКОНОК

Анимация интерфейса состоит из передвижения окон, плашек и других частей интерфейса. Отдельно можно рассмотреть анимацию кнопок и иконок, входящих в состав окон.

Окна редко появляются и изменяются только через прозрачность, и тем более это не происходит мгновенно. Анимация появления и изменения добавляет эффектности окнам и привлекает внимание игрока к определенным их частям. В таких анимациях основной задачей является контроль большого количества элементов и их поэтапное анимирование. Из отдельных простых мини-анимаций мы должны создать композицию для определенного окна. В работе над такой задачей всегда нужно учитывать тип окна и иерархию важности его элементов. Больше всего анимаций должно быть сосредоточено около основных смысловых элементов окна. Например, вокруг его названия или содержимого. Если аниматор увлечется и потратит много времени на анимацию ленточек на дальнем плане где-то в пустом месте окна, то, возможно, оттянет внимание игрока от важных кнопок или плашек.

Каждая кнопка, на которую можно нажать, скорее всего, будет обладать собственной анимацией. Особенно это касается основных функциональных кнопок окна, в частности кнопки «Купить», куда же без нее. Эта кнопка подпрыгивает и трясется при нажатии или наведении будто от щекотки. Привлекая ваше внимание, она словно просит нажать на нее. То же самое, только с большей сложностью, происходит и при анимации иконок.

В портфолио отлично будет смотреться проект с анимацией одного из стандартных окон игры. Магазин, выигрыш, проигрыш, рейтинг. Анимация появления и исчезновения с экрана покажет ваши навыки грамотного управления вниманием игрока и умение контролировать несколько частей интерфейса сразу. Анимация иконок и кнопок покажет, как вы умеете создавать привлекающие внимание анимации.

Эти задачи являются одними из основных в геймдеве. В каждой игре, независимо от жанра и платформы, есть интерфейс. И практически всегда ему нужна анимация. Элементов интерфейса очень много, а значит, много и работы для аниматора. С таким проектом в портфолио вы сразу попадете в поле зрения многих заказчиков.

АНИМАЦИЯ ОБЪЕКТОВ И СРЕДЫ

Тут можно привести в пример довольно разные задачи. Нужно отталкиваться от жанра. Один из самых популярных мобильных жанров — это матч-3 с элементами строительства. В таком проекте в сегменте строительства много элементов ландшафта и зданий, обычно в изометрическом ракурсе. Эти элементы часто эффектно появляются на поле, собираются по частям или появляются через визуальный эффект. Это может быть как небольшой уличный фонарь из нескольких составных элементов, так

и большое здание или даже группа. Анимацию сборки-появления такого объекта желательно добавить в портфель.

Подобная анимация представляет собой что-то вроде сборки конструктора. Стены разбиваются на бревна, крыша на черепицу или доски. Дерево сначала вырастает стволом, потом появляются ветки, и сверху уже добавляются листья. Делить крышу на множество плиток черепицы, если они мелкие, может быть избыточно, но и разделение на отдельные блоки будет выглядеть эффектно в анимации. Нам нужно понять структуру объекта и разбить ее на части. Сборка обычно осуществляется просто по слоям, с самого нижнего и выше, или по логике строительства. Например, сначала собираем каркас дома, а потом добавляем обшивку и побочные элементы.



Пример объектов, анимация появления которых отлично дополнит портфель игрового аниматора

Это, конечно, не единственный вариант анимации объектов. Ракурс может быть и сбоку, и сверху. Прошерстите популярные проекты в топках магазинов и выделите из них основные анимации. Затем возьмите их как референс и создавайте работы в портфолио. Тогда они будут отражать требования индустрии, а ваш шанс на получение предложения о работе увеличится.

АНИМАЦИЯ ИЛЛЮСТРАЦИЙ

В играх, да и в мультипликации и видеороликах, частая задача — анимация иллюстраций. Это может быть загрузочный экран или сюжетная вставка в игре. Данный тип задач — один из самых сложных для аниматора. Сложнее только собрать видеоролик.

В иллюстрации много взаимосвязанных элементов разного типа. К каждому нужен свой подход и своя анимация. Но поскольку у нас целостная картинка, анимация также должна быть целостной. Нам необходимо создать композицию из множества движений.

К тому же иллюстрации часто менее стилизованы и упрощены, чем отдельные объекты. А значит, нас ждет множество градиентов, различных элементов и детализация на них. Аниматору нужно контролировать очень большой объем изображений. Да и нарезать сам исходник тоже будет не так просто, как в случае одного объекта. В иллюстрации на 100% проверяются все навыки аниматора. Из-за большого объема иллюстрацию иногда даже анимируют по частям, а потом соединяют в единое полотно.

Настройка исходника под анимацию в случае с иллюстрацией достаточно сложна. В ней будет множество костей, путей и ограничителей, от которых зарядит в глазах. Нужна очень четкая структура всего этого многообразия настроек, иначе вы запутаетесь при первой же попытке анимировать изображение.

Но есть и небольшая облегчающая деталь в анимации иллюстраций — нам в большинстве случаев не требуются интенсивные движения. Персонаж может оглядываться, одежда развеиваться на ветру, растительность покачиваться от движения персонажа или ветра. К этому мы добавляем различные эффекты. То есть, по сути, мы делаем анимацию покоя. Иллюстрация должна дышать и жить. В таком типе движений нам не нужно крутить конечности в различные положения или создавать сложные анимации, такие как бег, прыжок и атака.

Задача сложная, но очень эффектная, особенно для обычного зрителя. А портфолио часто просматривают специалисты по найму, которые не являются аниматорами. Они цепляются за красивые картинки, а недочеты в самой анимации могут и пропустить. Поэтому важна презентабельность портфолио и красота исходников, которые вы анимируете. Иллюстрация в портфолио добавит притягательности вашему портфолию и покажет все ваши навыки.

Широкий кругозор или конкретное направление: что выбрать?

Я перечислила довольно много различных задач. Каждый тип требует определенных навыков и времени на практику. Если так распыляться, можно потратить очень много времени на обучение и создание портфолио. Так что же делать? Пытаться охватить как можно больше разных областей или уделить время только одному направлению? У каждого из этих подходов есть и плюсы, и минусы. Выбор только за вами.

Изучая все и сразу и наполняя портфель разными работами, вы охватите большее количество вакансий. У вас будет что показать при разных требованиях к кандидатам, а значит, и больше возможностей получить работу. Плюс к тому вы сможете претендовать на вакансию человека-оркестра в сфере анимации, например, в команде, где нужен специалист на широкий спектр задач. А в перспективе и на руководящие позиции, так как вы сможете контролировать всю область анимации в проекте.

Но с другой стороны, заказчика может и отпугнуть разношерстность портфолио, потому как у него не сложится понимания, в какой конкретно области работает человек, в каком стиле он лучше всего себя проявит. Но если у вас много работ на разные темы и стили, которые сгруппированы по папкам, и все работы одинаково высокого качества — портфолио будет только притягивать заказчиков, а не смущать своей разносторонностью. Но достичь такого трудно.

Если вы будете распыляться на множество областей, все равно есть большой шанс оказаться на лидирующей позиции. А остальные моменты могут проседать. И уровень работ в портфолио будет разным. Учтите, что судить вас будут по самой слабой работе. А значит, разный уровень анимаций в портфеле может пойти во вред.

Развиваться сразу в нескольких направлениях сложнее, чем упорно идти вперед в одном. Вам нужно уметь постоянно переключаться между разными задачами, не забывать никакие из своих навыков, постоянно оттачивая их на практике. Кроме того, необходимо изучать новые приемы и тенденции в анимации сразу по всем областям. Это может не дать такого же высокого уровня, как при развитии в одном направлении. Совершенствуясь только в узком направлении, вы наверняка станете в нем суперспециалистом раньше, чем во всех областях сразу.

Но если у вас в портфеле, например, только анимации персонажей, то на вакансию аниматора интерфейса вам будет затруднительно попасть, так как у вас нет примеров работ на эту тему. Иногда такие моменты сглаживаются тестовыми заданиями, которые компании выдают спорным кандидатам или вообще всем соискателям. Это позволяет оценить навыки аниматора, даже если у него нет аналогичной работы в портфолио, а также просто проверить человека на реальной рабочей задаче компании, не нанимая его на работу.

Подытожим. Иметь разностороннее портфолио — круто, но сложно. Очень трудно поддерживать одинаково высокий уровень всех работ. Но это позволяет свободнее искать работу в сфере анимации. Узконаправленное портфолио сужает возможности поиска вакансий, но позволяет быстрее стать специалистом в данной области. И конкретно в этой области для заказчиков вы можете быть интереснее, чем специалист широкого профиля. Я бы советовала для начала выбрать что-то одно наиболее для вас интересное. После изучения базы, разумеется. Достигнув определенного уровня навыков, вы сможете добавлять новые грани в свою профессию и развивать их, не забывая про основную специализацию.

Где учиться? Самообразование, университет, курсы

Сейчас в интернете огромное количество информации на любую тему. Но, несмотря на огромные потоки информации, найти в них нужное бывает не так легко. Безусловно, возможностей для самообучения стало намного больше. Действительно, обладая навыками поиска и усвоения информации, вы можете обучиться анимации с помощью интернета. Нужно лишь уметь грамотно распределять время, составлять план собственного обучения и не забывать о самоорганизации и дисциплине.

Полезно будет знать английский хотя бы на базовом уровне, тогда для вас станет доступно еще больше обучающего контента. Кроме того, обычно более свежие фишки анимации сначала появляются на английском языке, поэтому со знанием английского будет проще всегда оставаться в теме новинок индустрии.

Но просматривая только ролики в интернете и практикуясь по ним, вы не получите обратной связи. Никто не исправит ваши ошибки и не натолкнет на нужные мысли. Можно выкладывать свои работы на профильные сайты, чтобы получать отзывы там. Можно вести свое сообщество в социальной сети и собирать впечатления рядовых пользователей. Если проявлять активность в творческой среде, можно даже получить отзывы и наставления от более опытных коллег. Но это не будет полноценной обратной связью, потому как у вас все равно не будет связки «учитель — ученик». А это порой необходимо для продуктивного обучения. Но в целом вести социальную активность очень полезно. Обмен опытом позволит расширить свои возможности специалиста. Вы всегда сможете найти поддержку коллег или даже получить определенные рекомендации на вакансию. Но не забывайте, что вам тоже нужно что-то вкладывать в сообщество.

Во многих университетах и колледжах сейчас появились программы по обучению мультипликации. Когда училась я, были только кафедры дизайна, которые давали наиболее близкую специальность к анимации. Но именно диплом аниматора получить было нельзя. Сейчас таких программ пусть и не так много, но они есть. Этот путь обучения позволит вам освоить профессию в общем и целом. Отличный вариант, если вы точно не знаете, чем конкретно хотите заниматься как аниматор и какой тип анимации вам наиболее близок. Но, само собой, получая образование, вы тратите несколько лет. А помимо университетских заданий, необходимо еще и самообучение, поскольку программа может не закрыть все потребности обучения профессии. Только тогда вы получите максимум опыта. Этот вариант хорош также тем, что вы сможете познакомиться и пообщаться со специалистами в разных областях, которые поделятся своими знаниями и опытом.

Более облегченный формат — курсы на определенную тематику. Такой формат хорош тем, что вы получите уже укомплектованную информацию, которую собрали

и обработали для вас, чтобы максимально быстро и продуктивно обучить определенным задачам. Также у вас будет обратная связь с преподавателем курса. Но в сравнении с программой университета в таком формате обратная связь будет от меньшего числа людей. Вы можете сами выбрать, что вам интересно и какие навыки хотели бы подтянуть с помощью курса. А значит, вместо того чтобы полагаться на подготовленный план обучения университета, вы будете составлять свой. Это хороший вариант, когда вы уже обладаете определенными навыками и хотите подтянуть свои слабые стороны или добавить новую грань к своим возможностям. Я обучалась именно так, комбинируя курсы и самообучение, поскольку уже была концепт-художником с определенными навыками и пониманием, как нужно обучаться и что изучать.

ЗАКЛЮЧЕНИЕ

Вот мы и закончили путешествие в мир скелетной анимации. Но это только начало. С каждым выполненным уроком и еще одной работой в портфолио вы будете открывать все новые грани анимации и программы Spine 2D. Только постоянная практика может помочь вам стать крутым аниматором. Возможно, в будущем вы захотите даже комбинировать Spine с другими программами или сделать свою игру.

Творчество помогает нам проявить себя и поделиться со зрителем своим внутренним миром и идеями. Создавать что-то новое, то, что раньше было возможно только у нас в голове, — главная цель художника, аниматора, да и любого творческого человека. Надеюсь, этой книгой я показала вам еще один путь для реализации своих идей. Мир анимации огромен, разнообразен и красочен. Каждый сможет найти себя в нем!

Все права защищены. Книга или любая ее часть не может быть скопирована, воспроизведена в электронной или механической форме, в виде фотокопии, записи в память ЭВМ, репродукции или каким-либо иным способом, а также использована в любой информационной системе без получения разрешения от издателя. Копирование, воспроизведение и иное использование книги или ее части без согласия издателя является незаконным и влечет уголовную, административную и гражданскую ответственность.

Производственно-практическое издание

РОССИЙСКИЙ КОМПЬЮТЕРНЫЙ БЕСТСЕЛЛЕР. ГЕЙМ-ДИЗАЙН

Лепешкина Анна Викторовна

**ОСНОВЫ АНИМАЦИИ В SPINE 2D ДЛЯ НАЧИНАЮЩИХ
КАК СОЗДАВАТЬ КРУТЫЕ ИГРЫ БЕЗ ГИГАНТСКИХ БЮДЖЕТОВ**

Главный редактор *Р. Фасхутдинов*
Руководитель направления *В. Обручев*
Ответственные редакторы *Д. Калачева, А. Захарова*
Литературный редактор *О. Рухлова*
Младший редактор *П. Смирнов*
Художественный редактор *Е. Анисина*
Компьютерная верстка *Э. Брегис*
Корректоры *Ю. Никитенко, Л. Макарова*

Страна происхождения: Российская Федерация
Шығарылған елі: Ресей Федерациясы

ООО «Издательство «Эксмо»
123308, Россия, город Москва, улица Зорге, дом 1, строение 1, этаж 20, каб. 2013.
Тел.: 8 (495) 411-68-86.
Home page: www.eksmo.ru E-mail: info@eksmo.ru
Өндіруші: «Издательство «Эксмо» ЖШҚ
123308, Ресей, Мәскеу қаласы, Зорге көшесі, 1-үй, 1-құрылыс, 20 қабат, 2013-каб.
Тел.: 8 (495) 411-68-86.
Home page: www.eksmo.ru E-mail: info@eksmo.ru
Таяар белгісі: «Эксмо»

Интернет-магазин : www.book24.ru

Интернет-магазин : www.book24.kz

Интернет-дүкен : www.book24.kz

Импортер в Республику Казахстан ТОО «РДЦ-Алматы»,
Қазақстан Республикасына импорттаушы «РДЦ-Алматы» ЖШС.
Дистрибьютор и представитель по приему претензий на продукцию,
в Республике Казахстан; ТОО «РДЦ-Алматы»
Дистрибьютор және Қазақстан Республикасында өнімге шағымдар
қабылдау жөніндегі өкіл: «РДЦ-Алматы» ЖШС.
Алматы қ., Домбровский көш., 3-а, литер Б, офис 1.
Тел.: 8 (727) 251-59-90/91/92; E-mail: RDC-Almaty@eksmo.kz

Сведения о подтверждении соответствия издания согласно законодательству РФ
о техническом регулировании можно получить на сайте Издательства «Эксмо»:
www.eksmo.ru/certification

Техникалық реттеу туралы РФ заңнамасына сай басылымның сәйкестігін растау туралы
мәліметтерді мына адрес бойынша алуға болады: <http://eksmo.ru/certification/>.

Произведено в Российской Федерации
Ресей Федерациясында өндірілген

Дата изготовления / Подписано в печать 04.10.2023.
Формат 70x100¹/₁₆; Печать офсетная. Усл. печ. л. 23,33.
Тираж экз. Заказ

ООО «Издательство Вперед»

ISBN 978-5-04-190811-9



9 785041 908119 >

12+

Москва. ООО «Торговый Дом «Эксмо»

Адрес: 123308, г. Москва, ул. Зорге, д.1, строение 1.
Телефон: +7 (495) 411-50-74. **E-mail:** reception@eksмо-sale.ru

По вопросам приобретения книг «Эксмо» зарубежными оптовыми покупателями обращаться в отдел зарубежных продаж ТД «Эксмо»
E-mail: international@eksмо-sale.ru

International Sales: International wholesale customers should contact Foreign Sales Department of Trading House «Eksmo» for their orders.
international@eksмо-sale.ru

По вопросам заказа книг корпоративным клиентам, в том числе в специальном оформлении, обращаться по тел.: +7 (495) 411-68-59, доб. 2151.
E-mail: borodkin.da@eksмо.ru

Оптовая торговля бумажно-беловыми и канцелярскими товарами для школы и офиса «Канц-Эксмо»:
Компания «Канц-Эксмо»: 142702, Московская обл., Ленинский р-н, г. Видное-2, Белокаменное ш., д. 1, а/я 5. Тел./факс: +7 (495) 745-28-87 (многоканальный).
e-mail: kanc@eksмо-sale.ru, сайт: www.kanc-eksмо.ru

Филиал «Торгового Дома «Эксмо» в Нижнем Новгороде
Адрес: 603094, г. Нижний Новгород, улица Карпинского, д. 29, бизнес-парк «Грин Плаза»
Телефон: +7 (831) 216-15-91 (92, 93, 94). **E-mail:** reception@eksmonn.ru

Филиал ООО «Издательство «Эксмо» в г. Санкт-Петербурге
Адрес: 192029, г. Санкт-Петербург, пр. Обуховской обороны, д. 84, лит. «Е»
Телефон: +7 (812) 365-46-03 / 04. **E-mail:** server@szko.ru

Филиал ООО «Издательство «Эксмо» в г. Екатеринбурге
Адрес: 620024, г. Екатеринбург, ул. Новинская, д. 2щ
Телефон: +7 (343) 272-72-01 (02/03/04/05/06/08)

Филиал ООО «Издательство «Эксмо» в г. Самаре
Адрес: 443052, г. Самара, пр-т Кирова, д. 75/1, лит. «Е»
Телефон: +7 (846) 207-55-50. **E-mail:** RDC-samara@mail.ru

Филиал ООО «Издательство «Эксмо» в г. Ростове-на-Дону
Адрес: 344023, г. Ростов-на-Дону, ул. Страны Советов, 44А
Телефон: +7(863) 303-62-10. **E-mail:** info@rnd.eksмо.ru

Филиал ООО «Издательство «Эксмо» в г. Новосибирске
Адрес: 630015, г. Новосибирск, Комбинатский пер., д. 3
Телефон: +7(383) 289-91-42. **E-mail:** eksмо-nsk@yandex.ru

Обособленное подразделение в г. Хабаровске
Фактический адрес: 680000, г. Хабаровск, ул. Фрунзе, 22, оф. 703
Почтовый адрес: 680020, г. Хабаровск, А/Я 1006
Телефон: (4212) 910-120, 910-211. **E-mail:** eksмо-khv@mail.ru

Республика Беларусь: ООО «ЭКМО АСТ Си энд Си»
Центр оптово-розничных продаж Cash&Carry в г. Минске
Адрес: 220014, Республика Беларусь, г. Минск, проспект Жукова, 44, пом. 1-17, ТЦ «Outleto»
Телефон: +375 17 251-40-23; +375 44 581-81-92
Режим работы: с 10.00 до 22.00. **E-mail:** exmoast@yandex.by

Казахстан: «РДЦ Алматы»
Адрес: 050039, г. Алматы, ул. Домбровского, 3А
Телефон: +7 (727) 251-58-12, 251-59-90 (91,92,99). **E-mail:** RDC-Almaty@eksмо.kz

Полный ассортимент продукции ООО «Издательство «Эксмо» можно приобрести в книжных магазинах «Читай-город» и заказать в интернет-магазине: www.chitai-gorod.ru.
Телефон единой справочной службы: 8 (800) 444-8-444. Звонок по России бесплатный.

Интернет-магазин ООО «Издательство «Эксмо»
www.eksмо.ru

Розничная продажа книг с доставкой по всему миру.
Тел.: +7 (495) 745-89-14. **E-mail: imarket@eksмо-sale.ru**



В электронном виде книги издательства вы можете
купить на www.litres.ru

ЛитРес:
один клик до книг



БОМБОРА – лидер на рынке полезных и вдохновляющих книг.
Мы любим книги и создаем их, чтобы вы могли творить, открывать мир, пробовать новое, расти. Быть счастливыми. Быть на волне.

bombora.ru bomborabooks bombora



eksмо.ru

Официальный интернет-магазин издательства «Эксмо»



Хочешь стать автором «Эксмо»?

Spine 2D – это программа для создания анимации персонажей в 2D. Она используется в игровой индустрии, а также в различных проектах для мобильных приложений, веб-сайтов и рекламных материалов, где требуется создание анимации.

С ПОМОЩЬЮ ЭТОЙ КНИГИ ВЫ НАУЧИТЕСЬ:

- Работать с программой Spine 2D
- Подготавливать материалы для создания анимации
- Использовать инструменты для редактирования и настройки анимации
- Применять скелетную анимацию
- Управлять движениями и эмоциями персонажей
- Создавать анимационные ролики с помощью Spine 2D и Unity
- Экспортировать готовую анимацию в различные форматы и многое другое!

Эта книга написана для художников, дизайнеров и программистов, которые хотят создавать высококачественную анимацию персонажей в играх и других проектах. Она поможет вам сократить время на создание анимации и улучшить качество работы благодаря широкому набору инструментов и функций Spine 2D.

АННА ЛЕПЕШКИНА – художник, гейм-дизайнер, сценарист и аниматор на собственных проектах Potata, Almost my floor. Более 10 лет работала концепт-художником в игровых проектах, анимации и сфере настольных игр. Автор собственных курсов в школе smirnov-school. Одна из основателей студии Potata company.

**Хотите своими руками создавать крутую анимацию в играх?
Тогда эта книга – именно то, что вам нужно!**

ISBN 978-5-04-190811-9



9 785041 908119 >

 **БОМБОРА**
ИЗДАТЕЛЬСТВО

БОМБОРА – лидер на рынке полезных и вдохновляющих книг. Мы любим книги и создаем их, чтобы вы могли творить, открывать мир, пробовать новое, расти. Быть счастливыми. Быть на волне.

 bombora.ru  [bomborabooks](https://www.bomborabooks.com)   [bombora](https://www.bombora.com)