8 Последние штрихи в игре «Колодец с сокровищами»

Больше ловушек и уровней

Игра начинает обретать форму: теперь гномик, пользовательский интерфейс и фон игры выглядят намного привлекательнее. Сейчас у нас имеется только одна ловушка: коричневые шипы. Следующим нашим шагом станет создание еще двух таких же ловушек, но окрашенных для разнообразия в другие цвета.

Мы также добавим ловушку нового типа: вращающееся лезвие. Вращающееся лезвие наносит такие же травмы, как шипы, но имеет более сложное устройство — оно состоит из трех спрайтов, один из которых — анимированный.

Наконец, мы добавим препятствия, не наносящие повреждений, в форме стен и блоков, которые игрок должен будет обходить. Размещение этих объектов в сочетании с ловушками заставит игрока думать, как управлять перемещениями по уровню.

Шипы

Начнем с добавления шипов другого цвета. Сейчас у нас уже есть шаблон для существующих шипов, поэтому нам осталось только обновить спрайты и создать коллайдер. Для этого выполните следующие действия.

1. *Создайте новые шаблонные объекты для шипов.* Выберите шаблон SpikesBrown и создайте его копию, нажав комбинацию Ctrl-D (Command-D на Mac). Дайте новому объекту имя SpikesBlue.

   Создайте еще одну копию с именем SpikesRed.

2. *Обновите спрайт.* Выберите шаблон SpikesBlue и замените его спрайт изображением SpikesBlue.

3. *Обновите многоугольный коллайдер.* Поскольку многоугольный коллайдер находится в том же объекте, что и визуализатор Sprite Renderer, Unity будет использовать спрайт для определения формы коллайдера. Однако при смене спрайта форма коллайдера не обновляется автоматически; чтобы исправить эту проблему, нужно сбросить коллайдер в исходное состояние.

Щелкните по пиктограмме с изображением шестеренки в правом верхнем углу компонента Polygon Collider 2D и выберите пункт Reset (Сбросить) в появившемся меню.

4. *Обновите объект SpikesRed.* Вслед за объектом SpikesBlue выполните те же действия с объектом SpikesRed (и используйте изображение SpikesRed).

Когда все будет сделано, можете добавить в уровень несколько объектов SpikesBlue и SpikesRed.

Вращающееся лезвие

Далее добавим вращающееся лезвие. Вращающееся лезвие выступает чуть дальше, чем шипы, и имеет угрожающий вид циркулярной пилы. С точки зрения игровой логики вращающееся лезвие производит действие, идентичное шипам: когда гномик касается его, он погибает. Однако добавление в игру разных ловушек помогает усложнить процесс игры и удержать интерес игрока.

Поскольку вращающееся лезвие — это анимированный объект, мы сконструируем его из нескольких спрайтов. Кроме того, один из этих спрайтов — циркулярная пила — будет вращаться с высокой скоростью.

Чтобы добавить вращающееся лезвие в игру, перетащите спрайт SpinnerArm в сцену и выберите в его свойстве Sorting Layer слой Level Objects.

Перетащите спрайт SpinnerBladesClean и сделайте его дочерним по отношению к объекту SpinnerArm. Выберите в его свойстве Sorting Layer слой Level Objects и установите свойство Order in Layer в значение 1. Разместите его в верхней части спрайта опоры (SpinnerArm), затем установите его координату X равной 0, чтобы выровнять по центру.

Перетащите спрайт SpinnerHubcab и сделайте его дочерним по отношению к объекту SpinnerArm. Выберите в его свойстве Sorting Layer слой Level Objects и установите свойство Order in Layer в значение 2. Установите его координату X равной 0.

В результате вращающееся лезвие должно выглядеть так, как показано на рис. 8.1.

Теперь сделаем его опасным для гномика: подключим сценарий `SignalOnTouch`. Сценарий `SignalOnTouch` должен посылать сообщение, когда гномик коснется коллайдера, присоединенного к объекту; чтобы он работал, нам также понадобится добавить коллайдер. Выполните следующие действия, чтобы произвести все необходимые настройки.

1. *Добавьте коллайдер к вращающемуся лезвию.* Выберите объект SpinnerBladesClean и добавьте компонент Circle Collider 2D. Уменьшите его радиус до 2; это уменьшит область поражения и немного облегчит преодоление ловушки.

2. *Добавьте компонент SignalOnTouch.* Щелкните по кнопке Add Component (Добавить компонент) и добавьте сценарий `SignalOnTouch`.

   Щелкните по кнопке + внизу инспектора и перетащите во вновь появившееся поле объект Game Manager. Выберите функцию `GameManager.TrapTouched`.

Рис. 8.1. Сконструированное вращающееся лезвие

Далее нам нужно заставить лезвие вращаться. Для этого мы добавим объект аниматора Animator и настроим выполняемый им анимационный эффект. Анимационный эффект очень прост: достаточно просто повернуть присоединенный объект на полный оборот.

Для настройки аниматора нужно создать контроллер аниматора Animator Controller. Контроллеры аниматоров позволяют с помощью разных параметров определять, какой анимационный эффект должен воспроизводить аниматор. В этой игре мы не будем использовать продвинутые возможности контроллера, но вам будет полезно знать, что они существуют. Для настройки выполните следующие действия.

1. *Добавьте аниматор.* Выберите лезвие и добавьте новый компонент Animator.
2. *Создайте контроллер аниматора.* В папке *Level* создайте новый ресурс Animator Controller с именем Spinner.

   Здесь же, в папке *Level*, создайте новый ресурс Animation с именем Spinning.
3. *Подключите к аниматору новый контроллер аниматора.* Выберите лезвие и перетащите только что созданный ресурс Animator Controller в поле Controller.

Далее настроим сам контроллер аниматора.

1. *Откройте аниматор Animator*. Дважды щелкните по Animator Controller, чтобы открыть вкладку Animation (Анимация).
2. *Добавьте анимацию Spinning*. Перетащите анимацию Spinning в панель Animator. После этого в Animator Controller должно появиться единственное состояние анимации, помимо уже существующих элементов Entry, Exit и Any State (рис. 8.2).

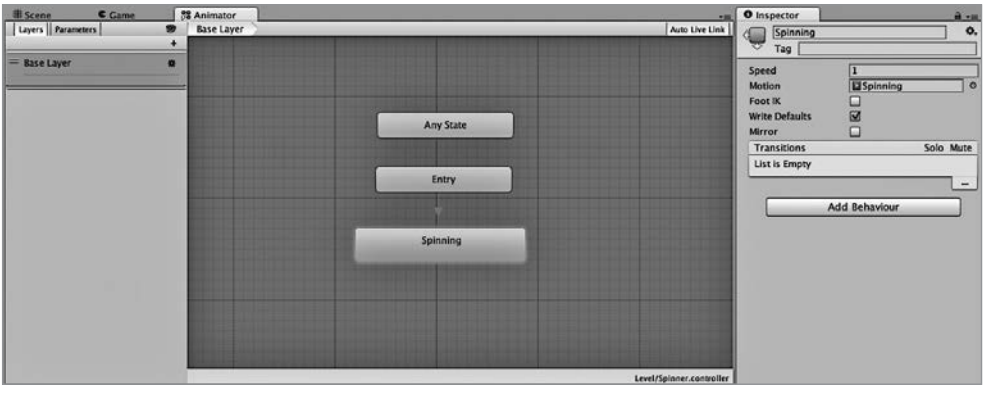

Рис. 8.2. Настройки Animator Controller для анимации Spinner

Теперь, когда аниматор Animator настроен на использование контроллера Animator Controller, который также настроен на запуск анимации Spinning, самое время настроить саму анимацию вращения.

1. *Выберите вращающееся лезвие*. Вернитесь обратно в представление сцены и выберите вращающееся лезвие.
2. *Откройте панель* Animation (Анимация). Откройте меню Window (Окно) и выберите пункт Animation (Анимация). Перетащите открывшуюся вкладку Animation (Анимация) в удобное для вас место. При желании можете даже присоединить ее к какому-нибудь разделу в интерфейсе, перетащив вкладку на панель в главном окне Unity.

   Прежде чем продолжить, убедитесь, что анимация Spinning выбрана вверху слева в панели Animation (Анимация).
3. *Выберите кривую в свойстве Rotation анимации Spinning*. Щелкните по кнопке Add Property (Добавить свойство) — откроется список анимируемых компонентов. Перейдите к элементу Transform ▸ Rotation и щелкните по кнопке + справа от списка.

По умолчанию свойства имеют два ключевых кадра — один соответствует началу анимации, а другой — концу (рис. 8.3).

Нам нужно, чтобы объект поворачивался на 360°. Для этого в начале объект должен иметь угол поворота 0°, а в конце — 360°. Чтобы добиться желаемого, нужно изменить последний ключевой кадр в анимации.

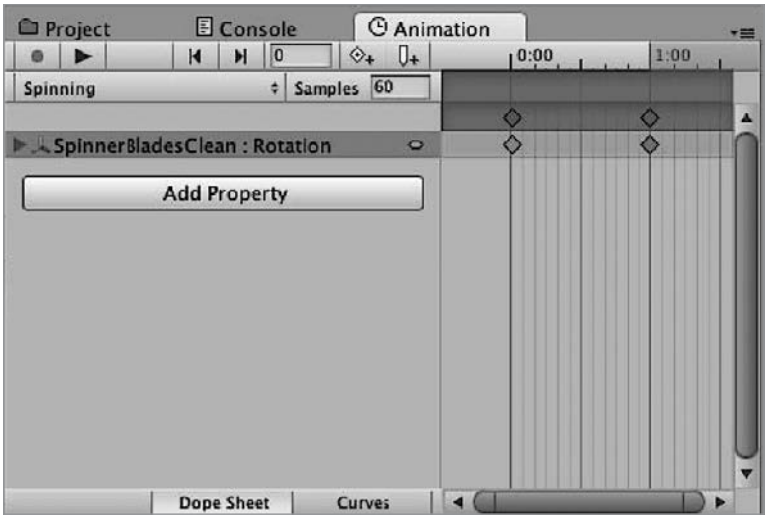

Рис. 8.3. Ключевые кадры для вновь созданной анимации

1. *Выберите самый правый ключевой кадр.*
2. *Щелкните по самому правому ромбику в панели Animation (Анимация), и анимация перескочит в эту точку на шкале времени.* После этого Unity перейдет в режим записи, то есть будет фиксировать любые изменения в анимации Spinning. Обратите также внимание, что квадратик в верхней части окна Unity станет красного цвета — это будет напоминать вам о происходящем.

   Взглянув на панель инспектора, можно заметить, что значение свойства Rotation компонента Transform также окрасилось в красный цвет.
3. *Измените угол поворота.* Установите угол поворота Z равным 360.
4. *Протестируйте анимацию.* Щелкните по кнопке Play (Играть) во вкладке Animation (Анимация) и понаблюдайте, как будет вращаться лезвие. Если оно вращается недостаточно быстро, щелкните и передвиньте последний ключевой кадр ближе к началу. Это уменьшит продолжительность анимации и заставит объект вращаться быстрее.
5. *Настройте бесконечное воспроизведение анимации в цикле.* Перейдите в панель обозревателя проекта Project и выберите созданный вами ресурс анимации Spinning. В инспекторе установите флажок Loop Time.
6. *Запустите игру.* Теперь циркулярная пила должна вращаться.

Прежде чем вращающееся лезвие будет готово к использованию, нужно уменьшить его размер, чтобы оно гармоничнее смотрелось в игровом поле.

1. *Измените масштаб вращающегося лезвия.* Выберите родительский объект SpinnerArm и установите масштаб по осям X и Y равным 0,4.

2. *Преобразуйте его в шаблон*. Перетащите объект SpinnerArm в панель обозревателя проекта Project. В результате будет создан новый шаблонный объект с именем SpinnerArm; переименуйте его в Spinner.

Теперь можно добавить вращающееся лезвие в нужное место в сцене, и гномик будет погибать, когда коснется его.

Препятствия

Кроме ловушек хорошо также добавить препятствия, не убивающие гномика. Эти препятствия будут замедлять продвижение игрока и вынуждать его думать о том, как обойти разные ловушки, добавленные вами.

Эти препятствия — самые простые из всех объектов, что вы добавляли в игру: для их создания вам потребуются только спрайт и коллайдер. Поскольку все они имеют очень простую конструкцию и похожи друг на друга, можно одновременно создать шаблоны сразу для всех препятствий. Вот что вы должны сделать для этого.

1. *Перетащите спрайты с изображениями блоков*. Добавьте в сцену спрайты BlockSquareBlue, BlockSquareRed и BlockSquareBrown. Затем добавьте спрайты BlockLongBlue, BlockLongRed и BlockLongBrown.
2. *Добавьте коллайдеры*. Выберите все шесть объектов и щелкните по кнопке Add Component (Добавить компонент) внизу инспектора. Добавьте компонент Box Collider 2D, и в каждом блоке появится зеленая граница, очерчивающая область определения столкновений.
3. *Преобразуйте их в шаблоны*. Перетащите каждый блок по отдельности в папку *Level*, чтобы создать шаблоны.

Блоки готовы, и теперь можно приступать к их размещению в уровне. Это было просто.

Эффекты частиц

Падение гномика вниз после гибели — не самый впечатляющий визуальный эффект. Чтобы создать эффект интереснее, нам придется задействовать системы частиц.

В частности, мы добавим эффекты частиц, появляющиеся в момент касания гномиком ловушки («брызги крови») и когда отделяется какая-то часть тела гномика («фонтан крови»).

Определение материала частиц

Поскольку в обоих случаях испускаемые частицы будут из одного и того же материала (кровь гномика), то начнем с создания единого материала, общего для двух систем. Выполните следующие действия, чтобы создать и настроить этот материал.

1. *Настройте текстуру Blood.* Найдите и выберите текстуру Blood. Измените ее тип со Sprite на Default и установите флажок Alpha Is Transparency (Альфа-прозрачность), как показано на рис. 8.4.
2. *Создайте материал Blood.* Создайте новый ресурс Material, открыв меню Assets (Ресурсы) и выбрав пункт Create ▸ Material (Создать ▸ Материал). Дайте материалу имя Blood и измените его шейдер на Unlit ▸ Transparent.

Затем перетащите текстуру Blood в поле Texture (Текстура). В результате настройки в инспекторе должны выглядеть так, как показано на рис. 8.5.

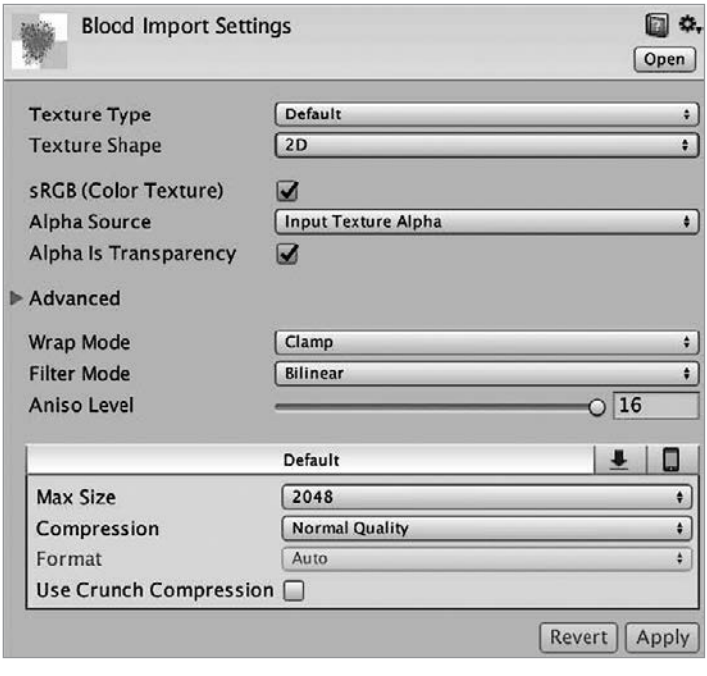

Рис. 8.4. Настройки для текстуры Blood

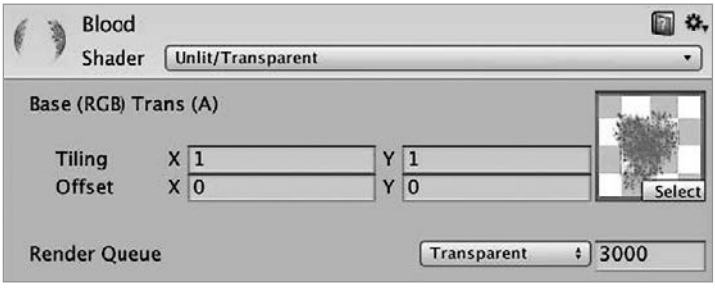

Рис. 8.5. Материал для эффекта частиц

Фонтан крови

Материал для крови готов к использованию, и теперь можно приступать к созданию эффектов частиц. Начнем с эффекта фонтана крови, который воспроизводится как эффект потока частиц, бьющего в определенном направлении и в итоге исчезающего. Вот как можно его создать.

1. *Создайте игровой объект для системы частиц.* Откройте меню GameObject (Игровой объект), откройте подменю Effects (Эффекты) и создайте новую систему частиц Particle System. Дайте новому объекту имя Blood Fountain.
2. *Настройте систему частиц.* Выберите объект и измените значения в разделе Particle System (Система частиц), как показано на рис. 8.6 и 8.7.

   Здесь есть пара параметров, о которых хотелось бы рассказать подробнее, поскольку они имеют нечисловые значения, которые нельзя просто скопировать из скриншота. В частности:

   - значение Color Over Lifetime (Цвет с течением времени) альфа-канала цвета изменяется от полностью непрозрачного (100 %) в начале до полностью прозрачного (0 %) в конце. Значение самого цвета изменяется от белого в начале до черного в конце;
   - в разделе Renderer (Визуализатор) системы частиц Particle System используется только что созданный материал Blood.
3. *Преобразуйте объект Blood Fountain в шаблон.* Перетащите объект Blood Fountain в папку *Gnome*.

Брызги крови

Теперь создадим шаблон Blood Explosion, испускающий одиночные брызги крови вместо сплошного потока.

1. *Создайте объект для системы частиц.* Создайте еще один игровой объект Particle System и дайте ему имя Blood Explosion.
2. *Настройте систему частиц.* Измените значения свойств в инспекторе, как показано на рис. 8.8.

   В этой системе частиц используется тот же материал и настройки изменения цвета с течением времени, как в эффекте Blood Fountain; единственное существенное отличие — новый эффект использует круговой эмиттер, и испускание всех частиц настроено так, что происходит одномоментно.
3. *Добавьте сценарий RemoveAfterDelay.* Чтобы не засорять сцену, эффект Blood Explosion должен удалять себя через некоторое время.

   Добавьте компонент RemoveAfterDelay в объект и установите его свойство Delay в значение 2.
4. Преобразуйте Blood Explosion в шаблон.