

5. Физика.

5.1. Твердые тела.

Rigidbody - это основной компонент, подключающий физическое поведение для объекта. С прикрепленным Rigidbody, объект немедленно начнёт реагировать на гравитацию. Если добавлен один или несколько компонентов Collider, то при коллизиях (столкновениях) объект будет передвигаться.

Так как компонент Rigidbody управляет перемещением объекта, к которому он прикреплен, вам не следует пытаться воздействовать на объект из кода с помощью изменения таких свойств Transform, как position и rotation. Вместо этого вам следует применять силы для того, чтобы толкать объект и позволить физическому движку рассчитать результаты.

Существуют случаи, при которых вы можете захотеть, чтобы у объекта был Rigidbody, но физический движок не управлял его движением. Например, вы можете пожелать управлять своим персонажем напрямую из кода, но при этом хотите, чтобы он взаимодействовал с триггерами. Этот тип не физического движения, производимый с помощью кода, называется кинематическим движением. У компонента Rigidbody есть свойство IsKinematic, которое может исключить объект из-под контроля физического движка, и позволить перемещать его кинематически из скрипта. Значение IsKinematic можно менять из кода, чтобы включать или отключать физику для объекта, но эта возможность требует дополнительных ресурсов и должна использоваться как можно реже.

Когда твёрдое тело перемещается со скоростью, меньшей определённого минимального порога, физический движок предполагает, что оно остановилось и находится в покое. При этом, объект не будет вновь двигаться до тех пор, пока с ним не произойдёт столкновение или пока к нему не применят силу, так что он уходит в “спящий” режим. Эта оптимизация означает, что на объект не будут расходоваться ресурсы CPU, пока его вновь не “разбудят” (т.е. не вновь не приведут в движение). По многим причинам засыпание и пробуждение твёрдых тел происходит прозрачно. Однако, иногда объект не удаётся разбудить, если в него или от него переместится статичный коллайдер (тот, что без твёрдого тела) изменяя положение трансформации. Это может привести, скажем, к висящему в воздухе твёрдому телу, когда пол под ним сдвинулся вниз. В таких случаях объект можно разбудить принудительно, с помощью функции WakeUp.

5.2. Коллайдеры.

Компоненты **коллайдера** определяют форму объекта для физических столкновений. Невидимый коллайдер не обязательно должен иметь ту же форму, что и сетка объекта, и на самом деле грубое приближение часто бывает более эффективным и неразличимым в игровом процессе.

Самые простые (и наименее ресурсоемкие) коллайдеры — это так называемые типы примитивных коллайдеров. В 3D это Box Collider, Sphere Collider и Capsule Collider. В 2D вы можете использовать Box Collider 2D и Circle Collider 2D. Любое их количество может быть добавлено к одному объекту для создания составных коллайдеров.

При тщательном позиционировании и изменении размера составные коллайдеры часто могут довольно хорошо аппроксимировать форму объекта, сохраняя при этом низкую нагрузку на процессор. Дополнительную гибкость можно получить за счет дополнительных коллайдеров на дочерних объектах (например, прямоугольники можно поворачивать относительно локальных осей родительского объекта). При создании такого составного коллайдера должен быть только один компонент Rigidbody, размещенный на корневом объекте в иерархии.

Обратите внимание, что примитивные коллайдеры не будут корректно работать с сдвиговыми преобразованиями — это означает, что если вы используете комбинацию поворотов и неравномерных масштабов в иерархии преобразований, чтобы результирующая форма больше не соответствовала примитивной форме, примитивный коллайдер не будет уметь правильно представлять полученный результат преобразования.

Однако бывают случаи, когда даже составные коллайдеры недостаточно точны. В 3D вы можете использовать Mesh Colliders, чтобы точно соответствовать форме сетки объекта. В 2D Polygon Collider 2D, как правило, не будет идеально соответствовать форме спрайта, но вы можете уточнить форму до любого уровня детализации, который вам нравится. Однако эти коллайдеры гораздо более требовательны к процессору, чем примитивные типы, поэтому используйте их экономно, чтобы поддерживать хорошую производительность. Кроме того, меш-коллайдер обычно не может столкнуться с другим меш-коллайдером (т. е. ничего не произойдет, когда они вступят в контакт). В некоторых случаях вы можете обойти это, пометив коллайдер сетки как Convex в инспекторе. Это создаст форму коллайдера в виде «выпуклой оболочки», которая похожа на исходную сетку, но с заполненными «внутренностями». Преимущество этого заключается в том, что коллайдер выпуклой сетки может сталкиваться с другими коллайдерами сетки, поэтому вы можете использовать эту функцию, когда у вас есть подвижный персонаж с

подходящей формой. Однако хорошим общим правилом является использование коллайдеров сетки для геометрии сцены и аппроксимация формы движущихся объектов с помощью составных примитивных коллайдеров.

Коллайдеры могут быть добавлены к объекту без компонента Rigidbody для создания полов, стен и других неподвижных элементов сцены. Их называют статическими коллайдерами. Как правило, вы не должны перемещать статические коллайдеры, изменяя положение Transform, так как это сильно повлияет на производительность физического движка. Коллайдеры на объекте, у которого есть Rigidbody, называются динамическими коллайдерами. Статические коллайдеры могут взаимодействовать с динамическими коллайдерами, но, поскольку у них нет Rigidbody, они не будут двигаться в ответ на столкновения.

5.3. Сочленения.

С помощью компонента Joint, вы можете добавить один rigidbody объект к другому или к фиксированной точке в пространстве. Обычно вы желаете, чтобы у сочленения была хоть какая-то свобода движения, так что Unity предоставляет различные Joint компоненты, реализующие разные ограничения. Например, Hinge Joint позволяет осуществлять вращение вокруг заданной точки и оси, в то время как Spring Joint держит объекты раздельно, но расстояние между ними может немного пружинить. Как обычно, названия 2D компонентов оканчиваются на 2D, например, Hinge Joint 2D.

У сочленений также есть другие опции, которые можно включить для различных эффектов. Например, вы можете настроить сочленение так, чтобы оно разрывалось при применении к нему силы больше заданного порога. Некоторые сочленения также позволяют возникать движущей силе (drive force) между соединёнными объектами для автоматического приведения их в движение.