

2 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

Целью данного дипломного проекта является разработка программного средства для моделирования реалистического поведения тел путем ограниченной симуляции некоторых основных физических законов. Таким образом, в ходе проектирования было выделено восемь основных модулей:

- 1) Модуль физического мира.
- 2) Модуль обнаружения столкновений.
- 3) Модуль разрешения контактов.
- 4) Модуль рэйкастинга.
- 5) Модуль движения тел.
- 6) Модуль линейной динамики.
- 7) Модуль угловой динамики.
- 8) Модуль отладочной отрисовки.

Рассмотрим подробнее цели и задачи каждого из вышеперечисленных модулей.

2.1 Модуль физического мира

Физический мир представляет собой центральный компонент нашего физического движка. Он хранит в себе информацию о всех физических телах, а также описания соответствующих им геометрических примитивов, являющихся, по сути, формой тел, если таковые присутствуют. Этот модуль осуществляет взаимодействие между остальными и именно через него происходит контроль глобальных параметров физической симуляции, таких как, частота обновления, сила гравитации и т.д.

2.2 Модуль обнаружения столкновений

Чтобы объекты имели возможность сталкиваться, мы должны предоставить им описание столкновения, формы объекта, его положения и ориентации в мировом пространстве. Это особая структура данных, отдельная от представления игрового процесса объекта (код и данные, определяющие его роль и поведение в игре) и от его визуального представления.

С точки зрения обнаружения столкновений в целом более предпочтительны формы, геометрически и математически простые. На примере представления реальных объектов можно привести следующие:

- скала может быть смоделирована как сфера;
- капот автомобиля может быть представлен прямоугольной коробкой;
- человеческое тело может быть аппроксимировано совокупностью взаимосвязанных капсул (фигуры в форме таблеток).

В идеале мы должны прибегать к более сложной форме только тогда, когда более простое представление оказывается недостаточным для

достижения желаемого поведения в игре. На рисунке 2.1 показаны несколько примеров использования простых форм для аппроксимации фигур объектов для обнаружения столкновений.



Рисунок 2.1 – Использование простых фигур для представления формы объектов

Для дальнейшего описания модуля введем такие понятия, как форма и трансформация. Форма описывает геометрические свойства объекта, а трансформация описывает его положение и ориентацию в мировом пространстве. Нам необходима трансформация по трем причинам:

1) С технической точки зрения, форма описывает только геометрию объекта (то есть, будь то сфера, коробка, капсула или какой-то другой примитив). Она также может описывать размер объекта (например, радиус сферы или размеры ящика). Но форма обычно определяется с центром в начале координат и в некоторой канонической ориентации относительно осей координат. Чтобы ее правильно использовать, форма должна быть трансформирована таким образом, чтобы позиционировать ее и соответствующим образом ориентировать ее в мировом пространстве.

2) Объекты являются динамическими. Перемещение произвольной сложной формы в пространстве может быть дорогостоящим, если нам нужно индивидуально перемещать составляющие части формы (вершины, плоскости и так далее). Но с помощью трансформации любая форма может перемещаться в пространстве легко, независимо от того, насколько она проста или сложна.

3) Информация, описывающая некоторые более сложные виды фигур, может занимать большой объем памяти. Таким образом, может быть полезно разрешить нескольким объектам совместно использовать одно описание

формы. Например, в гоночной игре информация о форме для многих автомобилей может быть идентичной. В этом случае все сопутствующие автомобили в игре могут иметь одну форму автомобиля.

Модуль обнаружения столкновений упаковывает информацию о столкновении в удобную структуру данных, которая может быть создана для каждого обнаруженного контакта. Эта информация часто включает в себя разделительный вектор – вектор, по которому мы можем провести объекты, чтобы правильно вывести их из столкновения. Структура также, как правило, содержит информацию о том, какие два объекта столкнулись, в том числе, какие отдельные фигуры пересекались. Модуль также может возвращать дополнительную информацию, такую как скорость тел, проецируемую на разделительную нормаль. На рисунке 2.2 представлен пример обнаружения столкновений.

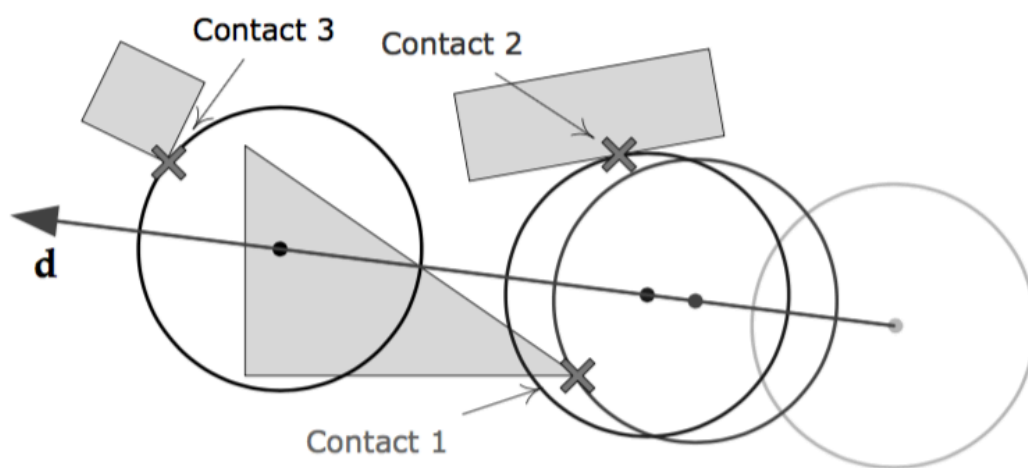


Рисунок 2.2 – Обнаружение столкновений

Система столкновений использует аналитическую геометрию – математические описания трехмерных объемов и поверхностей – для того, чтобы вычислить пересечения между формами.

Большинство систем обнаружения столкновений широко используют теорему, известную как теорема о разделительной оси. Она утверждает, что если найдется ось, вдоль которой не пересекается проекция двух выпуклых форм, то мы можем быть уверены, что две формы вообще не пересекаются. Если такая ось не существует и формы выпуклы, то мы точно знаем, что они пересекаются. Если формы вогнуты, то они не могут быть взаимопроницаемыми, несмотря на отсутствие разделительной оси. Это одна из причин, по которым предпочтительнее использовать выпуклые фигуры при обнаружении столкновений.

Эту теорему легче всего представить в двух измерениях. Интуитивно она говорит, что если линия может быть найдена, так что объект А целиком

находится на одной стороне линии, а объект В целиком на другой стороне, то объекты А и В не пересекаются. Такая линия называется разделительной линией и всегда перпендикулярна оси разделения. Поэтому, как только мы нашли разделяющую линию, легко убедиться, что теория на самом деле верна, глядя на проекции наших фигур на ось, перпендикулярную разделительной линии.

Проекция двумерной выпуклой формы на ось действует как тень, которую объект должен был оставить на тонкой проволоке. Это всегда отрезок линии, лежащий на оси, который представляет максимальные размеры объекта в направлении оси. Мы можем также думать о проекции как о минимальной и максимальной координатах вдоль оси, которую мы можем записать как полностью закрытый интервал. Как видно на рисунке 2.3, когда разделительная линия существует между двумя формами, их проекции не пересекаются вдоль оси разделения. Однако выступы могут перекрываться по другим, не разделяющим осям.

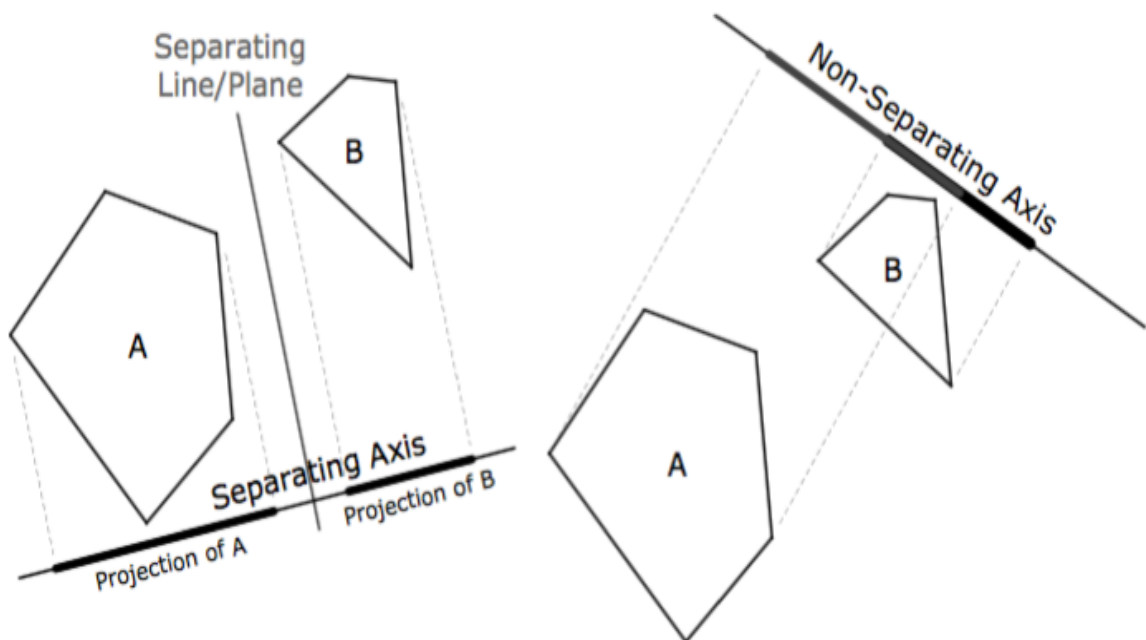


Рисунок 2.3 – Разделительная ось

В трех измерениях разделительная линия становится разделительной плоскостью, но ось разделения остается осью (то есть бесконечной линией). Опять же, проецирование трехмерной выпуклой формы на ось является отрезком линии, который мы можем представить полностью закрытым интервалом.

Одним из примеров этого принципа в действии является проверка пересечения сферы со сферой. Если две сферы не пересекаются, то ось, параллельная отрезку, соединяющему центральные точки сфер, всегда будет

действительной разделительной осью (хотя могут существовать и другие разделительные оси, в зависимости от того, как далеко друг от друга находятся две сферы). Чтобы визуализировать это, можно рассмотреть предел, когда две сферы как раз коснутся, но еще не вступили в контакт. В этом случае единственной разделяющей осью является одна параллель к сегменту от центра к центру. По мере того, как сферы раздвигаются, мы можем поворачивать ось разделения все больше и больше в любом направлении. Это показано на рисунке 2.4.

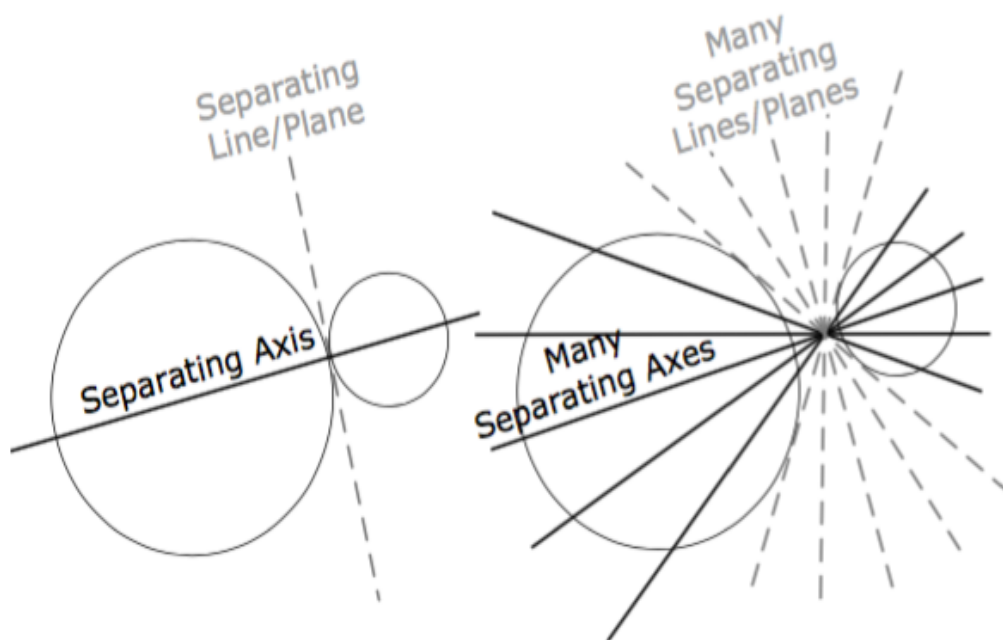


Рис 2.4 – Проверка столкновения сфер

2.3 Модуль разрешения контактов

Этот модуль, используя данные о столкновении, полученные от модуля обнаружения столкновений, корректирует поведение тел после контакта. Это значит, что на основе скоростей, масс и других характеристик столкнувшихся тел, он прикладывает к ним такие силы, чтобы произвести реалистичный эффект отскакивающих в результате коллизии объектов.

2.4 Модуль рэйкастинга

Еще одна важная задача физического движка – ответить на гипотетические вопросы о столкновениях в мире. Например:

- если пуля вылетит из оружия игрока в определенном направлении, какова будет первая цель, на которую она попадет, если таковая имеется;
- может ли транспортное средство переместиться из пункта А в пункт

Б без каких-либо действий по пути;

– найти все объекты противника в пределах заданного радиуса персонажа.

В общем случае такие операции называются коллизионными запросами. Наиболее распространенный вид запроса – это *collision cast*, иногда просто называемый *cast*. Этот термин определяет, что будет, если гипотетический объект будет помещен в физический мир и перемещен вдоль луча или отрезка. Он отличается от обычных операций обнаружения столкновений, потому что объект, находящийся в процессе трансляции, на самом деле не находится в физическом мире – он никак не может повлиять на другие объекты в мире. В качестве объектов для проведения данных запросов могут быть сферы, отрезки, кубы и так далее. На рисунке 2.5 представлен пример запросов для окружности.

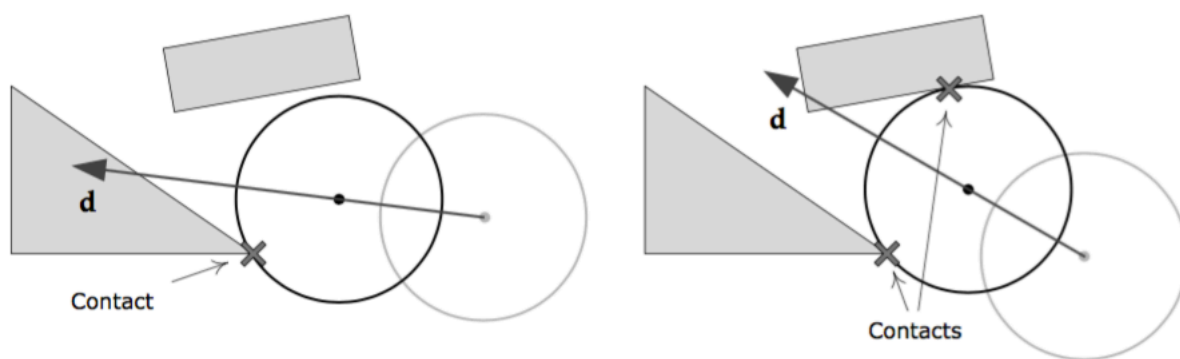


Рисунок 2.5 – Коллизионные запросы для окружности

Рэйкастинг часто используется в играх. Например, мы могли бы спросить систему столкновений, может ли персонаж А видеть персонажа В. Чтобы определить это, мы просто проводим направленный отрезок линии от глаз персонажа А до персонажа В. Если луч обращается к символу В, мы знаем, что А может «видеть» В. Но если луч ударяет какой-то другой объект до достижения символа В, мы знаем, что поле зрения блокируется этим объектом. Рэйкасты используются системами оружия (например, для определения попадания пули), механикой игрока (например, чтобы определить, есть ли твердая почва под ногами персонажа), системы искусственного интеллекта (например, проверки прямой видимости, запросы движения и т.д.), системы транспортных средств (например, для определения местоположения и привязки шин транспортного средства к местности) и т.д.

Другой распространенный запрос заключается в том, чтобы спросить систему столкновений, насколько воображаемая выпуклая форма сможет перемещаться вдоль направленного отрезка перед тем, как она ударит что-то твердое. Это называется столкновение со сферой, когда проверяемая форма

представляет собой сферу или другую фигуру. Как и в случае с трансформированием луча, заливка фигуры обычно описывается указанием начальной точки, расстояния до перемещения и, конечно, типа, размеров и ориентации фигуры, которую мы хотим наложить.

Есть два случая для проверки выпуклой фигуры:

- фигура уже взаимопроникает или контактирует, по крайней мере, с одним объектом, препятствуя своему удалению от своего исходного положения;
- фигура не пересекается ни с чем другим в начальном местоположении, поэтому она может свободно перемещаться на ненулевое расстояние вдоль своего пути.

В первом случае система столкновения обычно сообщает о контакте между фигурой и всеми элементами, с которыми она изначально пересекается. Эти контакты могут находиться внутри фигуры или на ее поверхности, как показано на рисунке 2.6.

Во втором случае фигура может переместиться на ненулевое расстояние вдоль отрезка перед тем, как с чем-то столкнуться. Мы предполагаем, что она столкнется с чем-то и, как правило, это только один объект. Тем не менее, фигура может удариться одновременно о несколько фигур, в зависимости от траектории. И, конечно, если столкнувшаяся фигура является невыпуклым многоугольником, она может в конечном итоге касаться более чем одной его части одновременно. Мы можем с уверенностью сказать, что вне зависимости от того, какая из выпуклых фигур проверяется, для кастинга можно создать несколько точек контакта. В этом случае контакты всегда будут на поверхности формы, а не внутри нее (потому что мы знаем, что фигура не соприкасалась с чем-либо, когда она начала свое движение). Этот случай показан на рисунке 2.5.

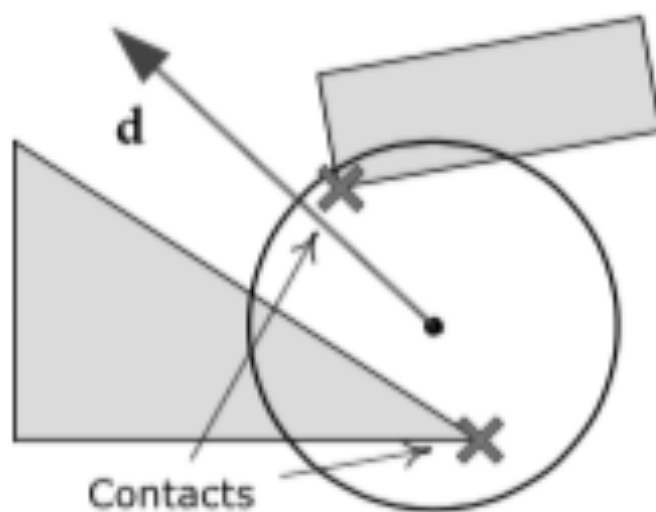


Рисунок 2.6 – Проверка сферы с несколькими контактами

2.5 Модуль движения тел

Данный модуль производит расчеты, связанные с движением, для всех тел, находящихся в физическом мире. Он определяет и сообщает в какой позиции и с каким вращением окажется физическое тело при данных силах, действующих на него в следующий момент времени. Модуль движения тел использует модули угловой и линейной динамики для этих вычислений, и впоследствии обновляет состояние физического мира новыми данными. Так как динамика движения тел разделена на угловую и линейную, этот модуль также должен применить изменения как позиции объекта, так и его вращения. На рисунке 2.7 изображено линейное и угловое движение тела. Еще одной задачей модуля является вычисление моментов вращения, если сила к телу была приложена не в центре масс, наряду с расчетом центров масс тел и их плотности.

2.6 Модуль линейной динамики

Модуль предоставляет возможности для управления всем что связано с линейным движением тел – приложение сил, импульсов и так далее. Любое количество сил может быть применено к телу в физическом мире. Сила всегда действует за конечный промежуток времени. (Если бы это действовало мгновенно, это было бы импульсом). Силы носят динамический характер – они часто изменяют свои направления и величины с каждым кадром. Таким образом, применять силу к телу можно только раз в течении кадра физического мира и всего времени воздействия силы.

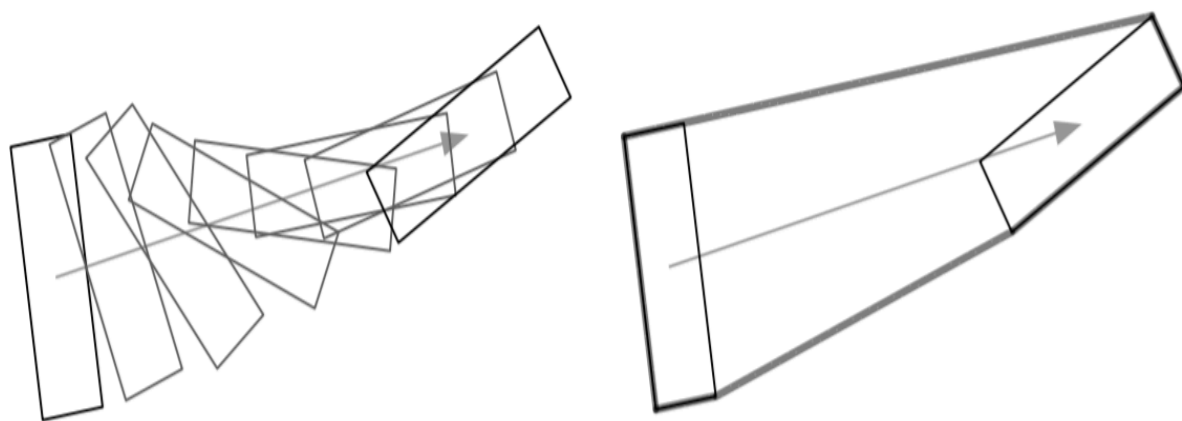


Рисунок 2.7 – Движение объекта с вращением

2.7 Модуль угловой динамики

Модуль предоставляет возможности для управления всем что связано с

угловым движением тел. Самая главная функция – это приложение момента вращения к телу. Чистый крутящий момент можно применить к телу, применяя две равные и противоположные силы к точкам, равноудаленным от центра масс. Линейные движения, вызванные такой парой сил, будут компенсировать друг друга, и это оставляет только их вращательные эффекты.

2.8 Модуль отладочной отрисовки

При разработке физического движка очень важно иметь возможность видеть результаты симуляции тел, чтобы можно было оценить правильность его работы. Модуль отладочной отрисовки ставит своей целью графическое отображение всех объектов физического мира в наглядной форме.