

РАЗРАБОТКА АЛГОРИТМИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

Цибенко А.С.

Научный руководитель - А.И. Шерстнёва

Томский политехнический университет

asc22@tpu.ru

Актуальность

На данный момент современный рынок мобильных приложений перенасыщен однотипными играми, но при этом качественных проектов не так много. При этом, если сравнить игровую индустрию для мобильных устройств с игровой индустрией для персональных компьютеров, можно сказать, что у первой гораздо большие возможности для роста и дальнейшего развития.

Введение

Не стоит относиться к современным играм просто как к развлечениям. Любая игра состоит из множества взаимосвязанных, но разносторонних аспектов: графика, игровая механика, интересный сюжет, отлично подобранное звуковое сопровождение, интуитивно понятный пользовательский интерфейс, оптимизация и так далее. При этом каждый из таких аспектов разбивается еще на десятки более узкоспециализированных. На разработку качественной игры порой уходят годы работы даже крупных компаний. Но с популяризацией игр в современном мире, их создание становится доступным не только гигантам игровой индустрии, но и любому желающему игроку, который уже очень давно вовлечен в мир игр в качестве потребителя.

Проектирование

Для успешной реализации игры требуется ее качественное проектирование. Одним из самых главных этапов проектирования является выбор игрового движка [1]. Одним из таких движков является Unity [2], на которую и пал выбор на ранних стадиях разработки. Такой выбор был связан с тем, что на момент начала работы над проектом уже был некоторый опыт работы с этим движком.

Так же очень важным этапом создания игры является разработка игровой механики. Было решено создать игру-головоломку. По изначальной задумке игроку предлагался большой стеклянный кубический закрытый лабиринт, в котором находился шарик. Лабиринт можно было поворачивать и наклонять для перемещения шарика. Шарик нужно докатить до финиша, выбирая нужный путь в лабиринте. Для удобства навигации и управления шариком было решено позволить игроку следить за шариком изнутри лабиринта. В ходе разработки идея

модернизировалась и менялась. Игроку уже предлагалось управлять шариком в лабиринте и гравитацией вокруг него, что принуждало шарик менять своё направление движения.

Развивая такой подход, было решено сделать локальный эффект изменения гравитации, который бы действовал на все игровые объекты в определенном радиусе вокруг шарика. При этом когда игровые объекты удалялись дальше этого радиуса, на них начинала действовать обычная гравитация, которая выглядела бы нормальной для игровой сцены.

Unity 5 предоставляет определенный минимальный набор объектов, эффектов и графических примитивов для работы с 3D пространством. Движок поддерживает несколько сценарных языков, импорт множества форматов файлов разных типов. Также используется программная библиотека PhysX [3], позволяющая рассчитывать выполнение физических законов. Unity 5 предоставляет определенный набор коллайдеров, из которых для тестов было решено использовать только Box Collider и Sphere Collider. Для создания гравитации был использован стандартный компонент Rigidbody и его свойства. После определения базовой игровой механики была спроектирована подсистема управления персонажем (шариком). Она представляет собой нескольких сенсорных зон управления приложением (touch-зон) для переката шарика путем придания ему ускорения. В качестве алгоритмического обеспечения были разработаны первичные формулы для создания эффекта общего изменения гравитации, заключавшегося в повороте всей сцены вокруг персонажа.

Идея реализации локального эффекта изменения гравитации заключалась в приложении к физическому объекту силы в направлении мнимой гравитации, просчете расстояния от главного героя до физического объекта и в случае расстояния, меньше указанного, отключения мнимой гравитации для этого объекта и включения стандартной. После проверки различных методов реализации этого эффекта был найден наиболее быстродействующий метод реализации. Метод заключается в проверке игровых объектов на попадание в некоторую геометрическую область пространства сферической формы, расположенной вокруг игрового персонажа.

В процессе работы над этими было решено, что игра должна позволять игроку выбирать произвольное направление гравитации в лабиринте. Переработав имеющиеся алгоритмы, была получена итоговая подсистема изменения игровой гравитации в любом направлении, которая работает полностью автономно и гибко настраивается. Для ее активации достаточно было передать в соответствующий скрипт подсистемы изменения гравитации только вектор, указывающий новое направление гравитации, и её параметры. На основе этого изменялось поведение всех близлежащих игровых объектов.

Реализация

На определенном этапе разработки было принято решение переделывания с нуля всех имевшихся наработок и их систематизации. Новая реализация управления в приложении кроссплатформенное и может успешно уживаться и с мобильными устройствами, и с ПК, и с консолями, что охватывает все способы управления (в т.ч. в будущем можно даже совместить с виртуальной реальностью и самым дешевым ее вариантом: google cardboard [4]). Принцип действия подсистемы управления следующий: игра реагирует на определенное событие (сейчас это, встряхивание телефона определенным образом), запускается режим выбора “нового пола для персонажа”, т.е. плоскости, по которой персонаж сможет свободно перемещаться, как по обычному полу. В этом режиме время в игре замедляется, игрок может вращать камеру вокруг персонажа, а также вокруг самой себя, тем самым осмотреться и выбрать подходящую плоскость. В случае с мобильным управлением, игроку достаточно будет коснуться пальцем этой плоскости. Тогда он выходит из этого режима, и игра продолжается с нормальным течением времени. Если через определенный промежуток времени игрок ничего не выбрал, то режим отключается сам.

Так как весь код был переписан, то в алгоритм подсистемы управления был введен вспомогательный вектор, хранящий текущее направление гравитации, через который производились физические расчеты. Прочие вектора, используемые при перемещении персонажа и вращении камеры, рассчитывались с помощью этого вектора и векторов, получаемых из координат камеры и персонажа.

Так же был реализован скрипт, позволяющий выполнить синхронный плавный поворот вектора гравитации и камеры в любом направлении (учитывая работу с градусами в одном случае и с радианами в другом). Скрипт

автоматически активирует поворот вектора (а с ним и камеры) при изменении переменной с желаемым конечным вектором гравитации. Для придания динамичности поворотам камеры, было составлено квадратное уравнение зависимости скорости поворота вектора от времени с начала поворота. Уравнение учитывало необходимые время обратного отклонения, скорость и амплитуду движения камеры. Использование этого алгоритма даст игроку эффект небольшого плавного отклонения камеры в противоположную сторону перед началом поворота.

Дополнительно, на основе метода трассировки луча, был разработан алгоритм, ограничивающий прохождение камеры сквозь стены лабиринта. В целом, алгоритм оценивал наличие препятствий между камерой и игровым персонажем. Если луч обнаруживал препятствие между ними, то камера сдвигалась ближе к персонажу. Если же препятствий не было, то камера отодвигалась на требуемую дистанцию. Таким образом, камера больше не попадает в области за препятствиями, которые загораживают вид. На этапе отладки игровое устройство (телефон) подключалось по USB к компьютеру. При этом с помощью специальной утилиты (при включении режима игровой отладки на компьютере) все параметры игрового процесса выводились на экран в реальном времени и позволяли отслеживать текущие показания акселерометра.

Заключение

На данный момент приложение полностью работоспособно и требуется только калибровка под различные игровые ситуации. В дальнейшем планируется развивать данное приложение, с последующим его продвижением на мобильный рынок Google Play. В случае успехе на данной платформе, можно будет собирать версию для устройств под управлением IOS.

Список использованных источников

1. Игровой движок. [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/Игровой_движок (дата обращения 11.10.2016).
2. Unity 5. [Электронный ресурс]. – URL: <https://unity3d.com/ru/5> (дата обращения 11.10.2016).
3. Технология NVIDIA PhysX. NVIDIA Corporation. [Электронный ресурс]. – URL: <http://www.nvidia.ru/object/nvidia-physx-ru.html> (дата обращения 11.10.2016).
4. Google Cardboard. Google Corporation. [Электронный ресурс]. – URL: <https://vr.google.com/cardboard/>