FIZICĂ ŞI INGINERIE

РАЗРАБОТКА МОБИЛЬНОЙ ИГРЫ ЖАНРА АРКАДА НА UNITY

Stanislav CARAGACIAN

CZU: 004.9:004.415.2 skaragachan@gmail.com

The purpose of this work is to consider the technologies that are used in the development of mobile games, to study popular genres and the mobile games market, and to develop a mobile game using one of the technologies. In the practical part, a mobile game prototype was developed with a detailed description of this process.

Разработка игр – это одна из самых прибыльных и развивающихся сфер в IT. Компьютерные и мобильные игры являются огромным сектором развлечения, который приносит игровым компаниям большую прибыль. В современном мире мобильные игры получают всё большее распространение, так как мобильные устройства становятся всё мощнее и соответственно могут поддерживать более качественные в плане графики и более нагруженные игры. В соответствии с этим разработка мобильных игр становится всё более востребованным и прибыльным делом. В первом десятилетии 21 века игровая индустрия продолжает развиваться.

Ежегодно выходят тысячи игр, которые продаются миллионными тиражами по всему миру. Самые успешные из них принесли разработчикам сотни миллионов долларов. Разработка мобильных игр также является одной из самых востребованных областей, поскольку мобильные игры становятся все более популярными с появлением смартфонов.

Целью исследования было изучить одни из самых популярных технологий для разработки мобильных игр (выявить их преимущества и недостатки), изучить процесс разработки полноценного игрового проекта и, соответственно, разработать своё игровое приложение на основе полученной информации.

В теоретической части работы была описана история создания мобильных игр, какие игры были началом своего жанра, также описаны какие игры повлияли на развитие того или иного жанра; была описана классификация мобильных игр по жанру, количеству пользователей в сессии, и платформам; были проанализированы средства разработки мобильных игр и обоснован выбор конкретной технологии и IDE.

Были рассмотрены следующие средства разработки: Unity3D, Unreal Engine, Cry Engine.

Для разработки игры был выбран движок Unity, так как он отлично подходит для кроссплатформенной разработки и мультиплатформенных игр. Игровой движок Unity также проще в использовании по сравнению со многими другими технологиями. Есть много других сложных технологий, которые становятся сложнее, пока мы их используем. Unity здесь является исключением. Unity является бесплатной для разработчиков или компаний, чей доход по играм использующих Unity меньше 200000$ в год. Unity поддерживает разработку только на языке программирования C#.

В практической части работы была разработана концептуальная модель игры, были подготовлены необходимые данные к проекту, такие как: звуки и музыка, модели, UI объекты и шрифты. Также была описана пошаговая разработка игры с её дальнейшем тестированием.

В игре были разработаны следующие основные алгоритмы и механики:

1. Был Создан универсальный загрузочный экран. Алгоритм этого загрузочного экрана заключается в том, что из любого скрипта можно обратиться к этому классу(скрипту) загрузочного экрана и запустить его или выключить когда нам это нужно. Этот объект не уничтожается при переходе между сценами и использует паттерн singleton, это означает, что не может быть дубликатов этого класса.
2. Был написан класс MySceneManager для опитмизации перехода между сценами. Этот класс является неким дополнением к встроенному классу SceneManager, который отвечает за переход между сценами. В нем был реализован алгоритм, который перед загрузкой новой сцены, сначала полностью выгружает старую сцену из памяти. При обычном переходе между сценами, выгрузка старой и загрузка новой сцены происходит параллельно, что сильно нагружает оперативную память. В данном случае оперативная память будет всегда загружена только одной сценой, что избавит от неожиданных вылетов из игры на слабых устройствах при переходе между сценами.
3. Была создана система улучшений. Она сделана таким образом, что её можно было бы легко дополнить не переписывая код, добавив еще одну характеристику.
4. Был написан скрипт управления космическим кораблём. Создание хорошего управления было одно из самых сложных задач. Основные классы и методы, которые были использованы при его создании – это классы Quaternion и Transform, и методы *Quaternion.LookRotation, Quaternion.Slerp, Transform.RotateAround.* Методы класса Quaternion были применены для плавного вращения камеры за космическим кораблём, а метод *Transform.RotateAround* для вращения космического корабля. Таким образом была написана основная механику управления. Дальше только дорабатывались мелкие детали в управлении и добавлялись некоторые особенности.
5. Был написан алгоритм для таргета ботов на UI. Суть алгоритма заключается в том, что он добавляет таргеты на вражеские цели, тем самым помогая

игроку отслеживать ботов на сцене и стрелять по ним. Это выглядит следующим образом:



Также были написаны множество других скриптов и механик, например, создание денежной системы, системы сохранения, системы контроля звуков и музыки, создание игрового интерфейса, джойстика и др. Всю нужную информацию о разработке этой игры вы можете найти в дипломной работе.

В конце хочу сказать, что разработка игры это сложный и комплексный процесс, который включает в себя создание концептуальной модели и сценария игры, создание UI элементов и 2D и 3D моделей, создании анимаций, написания кода, тестирования и исправление багов. Поэтому, обычно, разработкой игры занимается группа людей (команда). Редко можно встретить человека, который сам бы занимался всей разработкой игры, ведь для этого нужно уметь всё это делать и на разработку уйдет гораздо больше времени.

Библиография:

1. Хокинг, Джозеф. Unity — в действии. Мультиплатформенная разработка на C#
2. Джереми Гибсон Бонд. Introduction to Game Design, Prototyping, and Development. 3-е издание. – Издатель: Addison-Wesley.
3. Мэтт Смит и Чико Кейрос. Unity 5.x Cookbook. 3-е издание. – Издатель: Packt Publishing Limited

Рекомендовано
***Вадим СИРКЕЛИ**, др., конф унив.*