

Министерство образования и науки Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра дискретной математики

**Разработка видеоигры на платформе Unity3D**

**АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студента 4 курса 421 группы

направления 09.03.01 «Информатика и вычислительная техника»

факультета компьютерных наук и информационных технологий

Попова Дмитрия Романовича

Научный руководитель

д.ф.-м.н., профессор

должность, уч. степень, уч. звание

\_\_\_\_\_

дата, подпись

В.А. Молчанов

инициалы, фамилия

Заведующий кафедрой

к. ф.-м.н., доцент

должность, уч. степень, уч. звание

\_\_\_\_\_

дата, подпись

Л.Б. Тяпаев

инициалы, фамилия

Саратов 2018 год

**Введение.** Компьютерные игры появились в середине 70-х годов и за несколько десятилетий набрали огромную популярность с годовой прибылью в 10 миллиардов долларов США. Разработкой игр занимаются как крупные компании, так и небольшие фирмы, а также независимые частные сообщества.

В связи с быстрым ростом технологий у разработчиков появляется все больше возможностей для творчества. Первые игры представляли собой простейшие двухцветные 2D проекты, в настоящее время игровая индустрия достигла огромных высот. Современные 3D игры обладают настолько высоким уровнем реалистичности, что человек, играя в них, забывает, что находится в виртуальном мире. Так же в настоящее время активно ведется разработка игр для шлемов виртуальной реальности, которые позволяют пользователю с головой окунуться в игру.

Ни одна игра не обходится без игрового движка, который является основополагающим элементом для реализации проектов. Его основная задача - обеспечить базовыми технологиями, а так же упростить работу над проектом для разработчика; игровой движок дает возможность кроссплатформенности (запуск и разработка игр на нескольких платформах одновременно). Обычно игровой движок включает в себя:

- движок рендеринга;
- физический движок;
- звук;
- система скриптов;
- анимация;
- искусственный интеллект;
- сетевой код;
- управление памятью;
- многопоточность.

Unity – это мультиплатформенный игровой движок для разработки двух- и трехмерных игр, работающих под операционными системами

Windows OS X, Windows Phone, Android, Apple IOS, Linux, а так же на игровых приставках Wii, PS 3, PS 4, Xbox 360, Xbox One. Есть возможность создания браузерных приложений с помощью модулей Unity (Unity Web Player). Unity поддерживает 2 языка программирования JavaScript, C#. Для физических расчетов используется встроенный движок PhysX, разработанный компанией NVIDIA.

Целью работы является разработка игры на движке Unity 3D.

Тема является актуальной в связи с высоким ростом игровой индустрии в наше время. На рынке компьютерных технологий существуют как крупные компании (EA, Valve, Ubisoft), так и отдельные инди-разработчики. В большинстве случаев крупные компании используют собственные движки для разработки проектов, а Инди-разработчики же предпочитают готовые, среди которых есть и бесплатные (Unity, Unreal Engine и т.д.)

В связи с поставленной целью в работе решаются следующие задачи:

- Изучить интерфейс платформы Unity3D
- Описать сценарии видеоигры на языке программирования C#
- Провести окончательную сборку проекта

Объектом исследования являются видеоигры и принципы их разработки на платформе Unity 3D.

Предметом исследования является разработка игры на Unity.

В первой главе «Обзор игрового движка Unity3D» рассмотрены достоинства и недостатки платформы Unity3D и кратко описаны ее инструменты и интерфейс.

Вторая глава «Подготовка в разработке видеоигры на платформе Unity3D» включает в себя этапы подготовки платформы, план разработки видеоигры и подробное описание интерфейса программы Unity3D.

Третья глава содержит в себе весь цикл разработки, подробно описаны возможности платформы, создана видеоигра, сценарии для объектов и сами объекты.

В четвертой главе описан процесс итоговой сборки проекта, которая позволит импортировать игру на другие устройства и в интернет ресурсы.

### **Краткое содержание работы**

**Глава 1 Обзор игрового движка Unity 3D.** В настоящее время существует большое количество инструментов для разработчика игр, для исследования был выбран один из самых новых и мощных движков – Unity 3D. У данного инструмента существуют как преимущества, так и недостатки.

Что мы можем в Unity:

- моделировать физическую среду
- моделировать карты нормалей
- моделировать преграждение окружающего света в экранном пространстве (SSAO)
- моделировать динамические тени и т. д.

Данный функционал существует во многих игровых движках, преимущества Unity в том, что в нем есть чрезвычайно производительный визуальный процесс и мощная межплатформенная поддержка.

Визуальный редактор Unity значительно упрощает взаимодействие объектов на сцене и процесс сборки проекта. Благодаря Unity разработчики избавляют себя от такой проблемы, как ограниченная и недостаточно гибкая поддержка возможности написания сценариев. Особенно полезен этот редактор при разработке проектов с последовательным улучшением, например, в циклах создания прототипов или тестирования.

Дополнительным преимуществом платформы Unity является присутствие сильной межплатформенной поддержки, которая подразумевает собой возможность создания и развертывания игры на различных платформах

Третьим, менее значимым преимуществом является обеспечиваемое модульной системой компонентов, которая необходима для создания игровых объектов.

Имея большое количество преимуществ, в Unity присутствуют и недостатки. Например, сочетание визуального редактора и сложного кода может вызвать массу затруднений у разработчика. В обширных сценах можно потерять некоторые присоединенные компоненты.

Вторым и самым неожиданным недостатком для большинства опытных программистов является отсутствие возможности обращения к внешним библиотекам кода.

Также обзор игрового движка не обойдется без описания окна и интерфейса. Компонентами интерфейса Unity являются:

- вкладка Scene
- вкладка Game
- вкладка Hierarchy
- панель Inspector
- вкладки Project и Console

Верхнюю часть занимает панель инструментов, слева кнопки для осмотра и перемещения объектов, а в центре располагается кнопка Play. С правой стороны панель, отображающая информацию о выделенном в данный момент объекте.

На начальном этапе работы с движком необходимо выбрать язык, с помощью которого будет проходить разработка игры. В Unity поддерживаются всего 2 языка программирования JavaScript и C#. Язык C# имеет больше преимуществ чем недостатков перед языком JavaScript, особенно с точки зрения профессионального разработчика.

**2 Подготовка к разработке видеоигры на Unity3D.** Для того чтобы начать работать над проектом в Unity, не нужно быть профессиональным разработчиком видеоигр, но необходимо тщательно продумать и обговорить все нюансы игры. Даже имея под рукой такой гибкий инструмент как Unity, все равно нужно представлять, что должно получиться в итоге.

Для того, чтобы полностью разобраться с работой инструмента Unity, в данной работе будет разработана двухмерная видеоигра. Выбор пал на жанр

платформер для того, чтобы охватить как можно больше возможностей Unity за довольно короткий промежуток времени и невысокой сложностью разработки.

### **3 Процесс разработки игры на платформе Unity3D**

Перед началом работы необходимо подготовить рабочее поле программы, выставить необходимые настройки и изучить ее интерфейс.

**Подготовка рабочего поля.** Выставляем настройки рабочей области, для этого нужно создать новый проект, выбрать шаблон, место хранения, добавить дополнительные Asset в проект, если это необходимо. Далее, создав проект, открывается пустая сцена в которой будет лишь один объект – камера, сохраняем результат. После этого начинаем с ней работать: импортируем объекты сцены, добавляем им свойства, элементы сцены. Сделав эту работу, приступаем к созданию фона. В первую очередь нужно импортировать спрайты (однотонный фон, элементы заднего плана, платформы), и настроить их для использования. Получив составной спрайт из нескольких текстур, каждую из них можем использовать как отдельный спрайт, сделав его нарезку. В каждый из них добавляем элементы сцены и объекты так, чтобы они перекрывали фоновое изображение в нужном порядке. Далее настраиваем положение фона в сцене и слои. Создаем персонажа, импортировав необходимые текстуры в Unity, добавляем элементы анимации и настраиваем параметры.

**Логика игры.** Описанных выше свойств недостаточно для создания полноценной видеоигры. Необходимо описать логику поведения объектов сцены и их взаимодействие между собой. Для этого в Unity используется два языка программирования JavaScript и C#. В этой работе будет использован язык C#. В качестве редактора кода будет выступать Visual Studio Code.

**Поведение персонажа.** Создаем в папке Assets C# script. В окне Inspector существует предпросмотр кода, но для его редактирования необходимо зайти в редактор кода. Для того, чтобы управлять объектом

player с клавиатуры, необходимо написать скрипт и переместить его на объект в окне Hierarchy.

**Поведение камеры.** По стандарту камера на сцене является статичной, нам же необходимо, чтобы камеры следила за движением компонента `player`. Для этого напишем скрипт, в соответствии с рисунком 3.12, который в последствии перенесем на элемент `Main Camera`.

**Поведение противника.** Для полноценной картины необходимо создать врагов и написать сценарий их передвижения.

**Создание меню.** Обычно при запуске готовой видеоигры пользователь попадает на главное меню, из которого запускает непосредственно игровой процесс. Создадим меню, которое будет являться стартовой сценой при запуске проекта. Для этого создаем новую сцену, в которой присутствует лишь `MainCamera` и пишем для нее скрипт.

**Анимация.** Благодаря Unity, разработчикам предоставляется огромная возможность для работы с анимацией. У персонажа уже есть компонент `Animator`, как показано на рисунке 3.17. Теперь надо сделать анимации и настроить контроллер анимации.

**Система частиц.** Для создания живой картинке используются визуальные эффекты, такие как взрывы, огонь и т.д. В Unity этого можно добиться с помощью системы частиц. В магазине Unity Asset Store можно найти большое количество разнообразных наборов эффектов от самых простых до невероятно сложных.

**Графический интерфейс пользователя (GUI) в UnityEngine.** Практически ни одна игра не обходится без графического интерфейса. Стартовое меню, надписи, HUD (полоса здоровья и обозначения характеристик персонажа на экране) и многое другое называется – GUI.

**4 Сборка проекта.** На этапе завершения основной работы над проектом была произведена его сборка. Это происходит так: сперва процесс сборки размещает пустую копию игры. Затем он будет прорабатываться список сцен в настройках сборки: открывать в редакторе одну из них за раз, оптимизировать их и интегрировать их в пакет приложения. Он также подсчитает все ассеты, необходимые включенным в сборку сценам и сохранит эти данные в отдельном файле внутри пакета приложения.

**Заключение.** При выполнении данной работы разработана игра на платформе Unity 3D, а также описан сам процесс разработки. Данная игра является законченным продуктом, но при регулярной поддержке можно сделать его конкурентоспособным на рынке. Также можно использовать любой из элементов данной игры в любом другом проекте Unity, или же выложить их в магазин Unity, чтобы поделиться с сообществом или же для получения прибыли. Были рассмотрены основные аспекты разработки игр в целом и на Unity в частности. Платформа показала себя как очень гибкий, удобный и простой инструмент для создания игр, не требующий установки дополнительного программного обеспечения (для разработки игры использовался только Unity), при этом значительных недостатков при работе с платформой замечено не было. Unity постоянно развивается и довольно часто выходят обновления, в которых появляются новые возможности, а также улучшаются старые, поэтому некоторые аспекты данной работы могут не совпадать с версией новейшей платформы.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Game Engine // Unity. [Электронный ресурс] URL: <http://unity3d.com> (дата обращения 10.11.17) Загл. с экр. Яз. рус.
- 2 Хокинг, Джозеф. Unity в действии, мультиплатформенная разработка на C# / Джозеф Хокинг: Питер, 2016, 336 стр
- 3 Unity3D // Хабрахабр. [Электронный ресурс] URL: <https://habrahabr.ru/hub/unity3d/> (дата обращения 20.11.17) Загл. с экр. Яз. рус.
- 4 Игровой дизайн, гейм дизайн (game design) // GameDev.ru – Разработка игр. [Электронный ресурс] URL: <http://www.gamedev.ru/gamedesign/terms/gameplay> (дата обращения 10.01.18) Загл. с экр. Яз. рус.
- 5 Основные ошибки при проектировании главного меню игры // Хабрахабр. [Электронный ресурс] URL: <https://habrahabr.ru/post/195608/> (дата обращения 02.02.18) Загл. с экр. Яз. рус.
- 6 Создание первой игры на Unity, от идеи до релиза. [Электронный ресурс] URL: <https://habr.com/post/321038/> (дата обращения 14.12.17) Загл. с экр. Яз. рус.
- 7 Unity и C# скрипты – основы. [ Электронный ресурс ] URL: [http://xrayisgray.de/sites/tutorial\\_0\\_ru.html](http://xrayisgray.de/sites/tutorial_0_ru.html) (дата обращения 10.03.18) Загл. с экр. Яз. рус.
- 8 Алан Торн. Основы анимации в Unity, 2016. Русский перевод
- 9 Кенни Ламмерс. Шейдеры и эффекты в Unity, 2014. Русский перевод
- 10 Алан Торн. Искусство создания сценариев в Unity, 2016 Русский перевод.

11 Джозеф Хокинг. Unity в действии. Мультиплатформенная разработка на C#, 2016. Русский перевод

12 Делаем игру за 6 уроков [Электронный ресурс] URL:

<http://tceh.com/e/unity/> (дата обращения 25.01.18) Загл. с экр. Яз. рус.

13 Разработка 2D игры [Электронный ресурс] URL: <https://null-code.ru/> (дата обращения 08.03.18) Загл. с экр. Яз. рус.

14 Введение в программирование игр на Unity [Электронный ресурс] URL:

<https://mva.microsoft.com/ru/training-courses/-unity-8635> (дата обращения 01.04.18) Загл. с экр. Яз. англ.

15 Издание о разработке и обо всем [Электронный ресурс] URL:

<https://tproger.ru/tag/unity/> (дата обращения 18.10.17) Загл. с экр. Яз. рус.

16 Chris Dickinson.Unity 5 Game Optimization, 2015

17 Alan R. Stagner. Unity Multiplayer Games, 2013