# СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc53424231)

[1 СРЕДСТВА РАЗРАБОТКИ ИГРОВЫХ ПРИЛОЖЕНИЙ 5](#_Toc53424232)

[1.1 Обзор игрового движка *Unity* 5](#_Toc53424233)

[1.2 Обзор игрового движка *Unreal Engine* 4 7](#_Toc53424234)

[1.3 Сравнение игровых движков *Unity* и *Unreal Engine* 4 11](#_Toc53424235)

[Список используемых источников 14](#_Toc53424236)

# ВВЕДЕНИЕ

Игры наравне с наукой и работой относятся к одним из ключевых видов человеческой деятельности. Люди с незапамятных времён увлекаются играми, чтобы снять напряжение, расслабиться и забыть на время о проблемах окружающего мира. Индустрия компьютерных игр, в наше время, очень сильно развилась, и может легко, быстро и относительно недорого предоставить игру любого жанра и удовлетворить предпочтения каждого игрока, а вычислительная мощность современных цифровых устройств, с помощью фантастической красоты компьютерной графики, может надолго затянуть его в виртуальный мир, наполненный потрясающими пейзажами и завораживающими приключениями.

В настоящее время компьютерные видеоигры помимо развлечения также несут обучающий, расслабляющий и даже лечебный характер. Ведь ни для кого не секрет, что всего лишь пара часов, проведённая за любимой игрой, может поднять настроение и избавить игрока от депрессии.

На данный момент, видеоигры являются очень популярным и легкодоступным источником удовольствия. В них можно играть как одному, так и в компании друзей. Видеоигры доступны на большинстве современных гаджетов: смартфонах, компьютерах, консолях, телевизорах и даже на некоторых электронных наручных часах. В видеоигры играют люди всех возрастных категорий, потому что это весело и они, как правило, имеют низкий порог вхождения, удобное управление и красивую графику.

Разработка видеоигр является очень увлекательным и прибыльным занятием. Чаще всего популярные видеоигры относятся к категории бесплатных игр, но за счёт размещения рекламы и продажи костюмов для игровых персонажей разработчики получают хороший доход. Также разработка игр вносит свой вклад в развитие культуры, так как любимые игры вдохновляют людей писать музыку, монтировать видео и рисовать картины. Видеоигры сближают людей и помогают им лучше узнать друг друга и найти общий язык. Некоторые игры так быстро набирают популярность и захватывают молодую аудиторию, что по ним проводят соревнования, в которых любой желающий игрок может принять участие и побороться за главный приз и всеобщую славу. Из всего вышеперечисленного ясно, что разработка игр весьма актуальна на данный момент и в будущем будет только развиваться.

# СРЕДСТВА РАЗРАБОТКИ ИГРОВЫХ ПРИЛОЖЕНИЙ

## 1.1 Обзор игрового движка *Unity*

*Unity*3*D* – кроссплатформенный игровой движок от компании *Unity* *Technologies*. История создания движка довольно интересная и поучительная. Она интересная, потому что двое парней захотели сделать игру, но для этого им не подходили существующие инструменты. И они решили сделать свой движок, а потом уже делать на нем игру. И после того, как они сделали движок, они поняли, что им не так-то и интересно делать игры, а больше нравится заниматься непосредственно движком. Так и началась история одного из самых известных и мощных движков. А поучительная эта история потому, что никогда не знаешь, чем обернется то или иное начинание [1].

*Unity* – это бесплатный движок. Единственное ограничение – при запуске игры показывается логотип *Unity*, от которого можно избавиться, только купив расширенную версию. Доступность и простота движка – это то, что привлекло многих к разработке игр на нём.

*Unity* использует компонентно-ориентированный подход. Все в игре – это объект *GameObject*, куда добавлены различные компоненты. Например, если мы делаем платформер, мы добавляем *GameObject*, и к нему добавляем графический компонент для рендеринга игрока и компонент управления, чтобы можно было перемещать игрока по сцене, используя клавиатуру или мышь. Таких различных компонентов можно добавить любое количество к любому *GameObject*. То есть, создание игры в *Unity* – это добавление *GameObject*-ов, и добавление им нужных компонентов.

Но игровой движок не так уж и прост, как кажется на первый взгляд. Чтобы сделать что-то нетривиальное, придется создавать свои компоненты. В терминах *Unity* они называются скриптами. Пишутся они на языке *C*#. Написание своих компонентов – это довольно сложное занятие. Фактически, это обычное программирование. Так что, без знания языка программирования придется не просто.

*Unity* – хороший выбор для создания средних и лёгких по сложности проектов, как для ПК, так и для мобильных устройств. Вдобавок, разработку упрощает огромное сообщество пользователей, которое всегда помогает разобраться с возникшими вопросами, и большое количество готовых ассетов, включая скрипты.

На *Unity* не рекомендуется разрабатывать огромные проекты *AAA*-класса. Сама по себе идея со скриптами хороша, но довольно медлительна. К тому же, язык *C*# – интерпретируемый. Несмотря на все *JIT*-оптимизации, он медленней, чем тот же *C*++. Сотни объектов на сцене со сложными компонентами могут убить производительность.

Впрочем, все зависит от мастерства программиста. Существуют исключения, когда опытные люди создавали на *Unity* большие проекты с хорошей производительностью. Правда, им приходилось много чего переделывать в движке под себя. Так что, перед тем как браться за что-то грандиозное, требуется хорошо разобраться во внутреннем устройстве движка и его деталях. Пользовательский интерфейс игрового движка *Unity* представлен на рисунке 1.1:

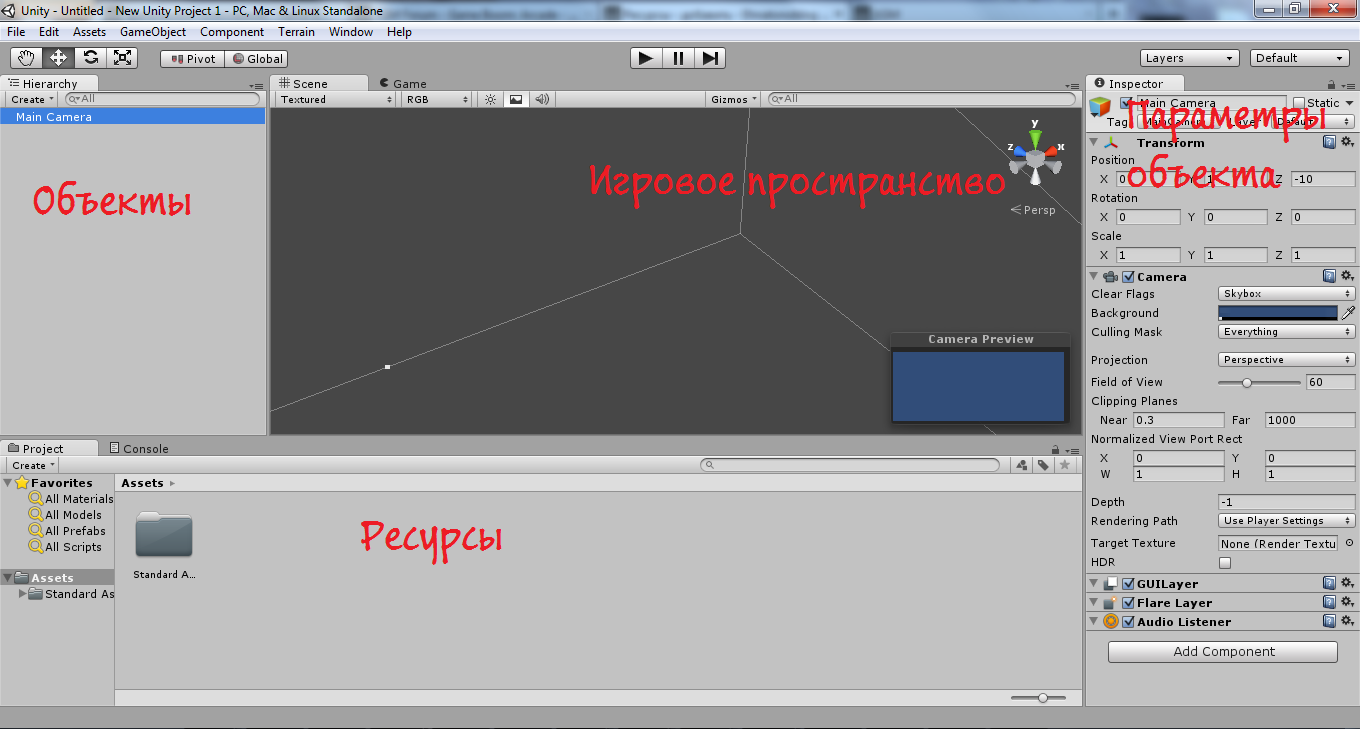


Рисунок 1.1 – Пользовательский интерфейс игрового движка *Unity*

Один из козырей *Unity* – это список поддерживаемых платформ, где может запускаться приложения. *Unity* работает почти везде – на ПК (все операционные системы), на *Android*, на *iOS*, на *SmartTV*, в браузере, на различных экзотических системах – например, *Tizen* *OS*. Написание скриптов для игровых объектов происходит в *MonoDevelop* (идет по умолчанию), либо при использовании стороннего редактора. Многие используют *Visual Studio*. Некоторые настраивают для этих целей *Sublime Text*.

Мощный плюс *Unity* – это ассеты. Все в игре, включая код, картинки, представляется ассетами. Ассеты можно экспортировать, импортировать. Таким образом, сторонние разработчики могут делать целые заготовки для игр. Все, что вам останется – это заменить картинки, подправить скрипты – и можно считать игру уникальной. Нюансом является то, что различные ассеты могут быть несовместимы между собой как в прямом смысле, так и не подходить по стилю. Также есть специальный онлайн-магазин – *Unity Asset Store*. Там продаются готовые ассеты от сторонних разработчиков. Любой желающий может сделать свой ассет, и выложить его в продажу в этом магазине. Некоторые люди сделали на этом целый бизнес, благодаря большому рынку *Unity*-пользователей. Также важный момент, что магазин доступен прямо из редактора *Unity*. То есть, добавление новых ассетов максимлально упрощается. Можно зайти в магазин, кликнуть на нужный ассет, и он сразу скачается и добавится в текущий проект. Быстро и удобно [2].

За счёт своей популярности *Unity* имеет огромное комьюнити. Если у пользователя возник какой-то вопрос в процессе разработки, скорей всего, он уже много раз задавался, и столько же раз уже был решен. Информации по игровому движку очень много, в том числе и официальной. Это огромный плюс движка по сравнению с другими. Разработчик не чувствует себя одиноким и при решении проблем ему всегда помогут. Если требуется написать свою игру, но не хочется сильно заморачиваться с программированием – движок *Unity* подходит лучше других. Набросать объектов на сцену, и связать их между собой получится и без знания языка программирования. В случае возникновения проблем большое сообщество пользователей и документация с официального сайта помогут справиться с любыми трудностями.

## Обзор игрового движка *Unreal Engine* 4

Фотореализм – это первое слово, которое приходит на ум, когда думаешь об *Unreal* *Engine* (*UE*4). При использовании технологий студии разработки движка Epic Games получаются настолько реалистичные кат-сцены, что их сложно отличить от реальной съёмки. Благодаря этому движок широко используется и за пределами игр, в таких секторах, как архитектура, дизайн продукта и кинопроизводство. При помощи его возможностей создавались такие картины как «Звёздные войны» и «Мандалорианец».

С 1998-го года *Epic* *Games* радует своих пользователей техническими демонстрациями, показывающими, на что способен *Unreal* *Engine*, от трассировки лучей до технологий реального времени, физики и систем разрушений. Такие возможности делают его предпочтительным движком в разработке *ААА*-игр. И по сегодняшний день *Unreal* *Engine* доказывает свою состоятельность в разработке различных жанров и стилей игр, но его успех основан на проектах с реалистичной графикой. Возможности Unreal Engine представлены на рисунке 1.2:



Рисунок 1.2 – Фотореализм игрового движка *Unreal Engine*

*UE*4 поддерживает широкий спектр платформ: *Windows* *PC*, *PlayStation* 4, *Xbox* *One*, *Mac* *OS* *X*, *iOS*, *Android*, *AR*, *VR*, *Linux*, *SteamOS* и *HTML*5. На *Unreal* *Engine* разрабатывают свои проекты более чем семи с половиной миллионов разработчиков, из-за чего его можно считать популярным. На *Unreal Engine* можно разрабатывать самые разнообразные проекты. Гибкость *UE*4 является одной из основных причин его популярности. Каким бы ни был проект нестандартным, данный движок сможет справиться с этой задачей.

Доступ ко всему исходному коду *C*++ для движка – одна из причин, по которым *UE*4 настолько гибок. Код можно скачать с *GitHub*, любой желающий может настроить его под себя, и это совершенно бесплатно. Для сравнения, Unity предоставляет только профессиональным и корпоративным клиентам доступ к своему исходному коду – и это стоит дорого. Разработка и выпуск игры, сделанной с помощью *Unreal Engine*, полностью бесплатный. Плата в размере 5% выплачивается *Epic Games* только тогда, когда игра приносит доход в размере 3000 долларов в квартал.

*Unreal Engine* упрощает разработку игр для нетехнических пользователей с помощью *Blueprint*. Визуальная скриптовая система *UE*4, *Blueprint*, представленная на рисунке 1.3, является основной причиной того, что движок является отличным инструментом для конструирования игровых прототипов. В *Blueprint* вы перетаскиваете узлы и добавляете связи, чтобы соединить их и добавить логику. Этот узловой интерфейс позволяет дизайнерам, не имеющим технической подготовки, исследовать свои идеи без необходимости знать, как кодировать.

Движок *Unreal Engine* отлично подходит для крупномасштабных игр. Как показывают его достижения в качестве развития *AAA*-игр, движок хорошо масштабируется для больших команд и игр с большим числом игроков. *Unreal Engine* опустила планку до кроссплатформенной разработки. *UE*4 не только может создавать большие игры, но и его функции рендеринга отлично показывают себя на разных платформах. Конвейер рендеринга является современным, поддерживая огромное разнообразие платформ от мобильных до высококлассных компьютеров. Портирование проектов *Unreal Engine* на различные платформы особенно ценно при работе в виртуальной реальности, так как при разработке требуется учитывать комфорт пользователя.

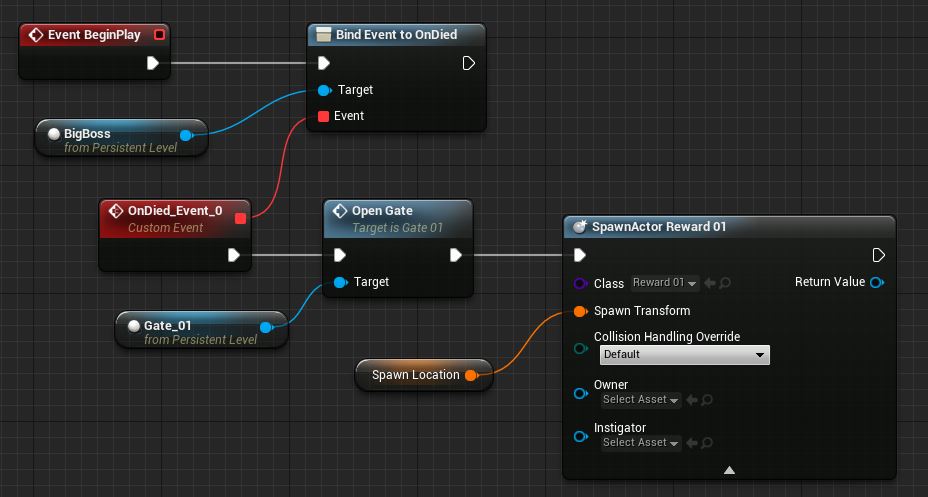


Рисунок 1.3 – Визуальная скриптовая система *Blueprint*

*Unreal Engine* имеет широкий набор инструментов и отличный редактор. Простота использования движка демонстрируется широким спектром функций и инструментов в интуитивно понятном редакторе с дружелюбным интерфейсом, доступном для многих настроек. Широкий спектр инструментов предоставляет решения практически для каждой макрозадачи, которую придется решать: макет сцены, включая ландшафт и листву, кинематографию, визуальные сценарии, систему частиц, анимационную сеть, создание материалов, световое отображение, управление движением, уровень детализации, физику, основные модели искусственного интеллекта, инструменты отслеживания производительности и памяти для оптимизации [3].

*Unreal Engine* имеет хорошую официальную и общественную поддержку. Будучи широко используемым движком, отдача комьюнити производится на высшем уровне. Стоит отметить, что уровень предоставляемой поддержки может зависеть от того, какой разработчик имеет доступ к движку. Платящий пользователь *UE*4 с пользовательской лицензией будет иметь непосредственное внимание команды разработчиков *Epic Games*, в то время как свободным пользователям, придется больше полагаться на богатство учебников и интернет-ресурсов для поиска ответов. Пользовательский интерфейс *Unreal Engine* представлен на рисунке 1.4:

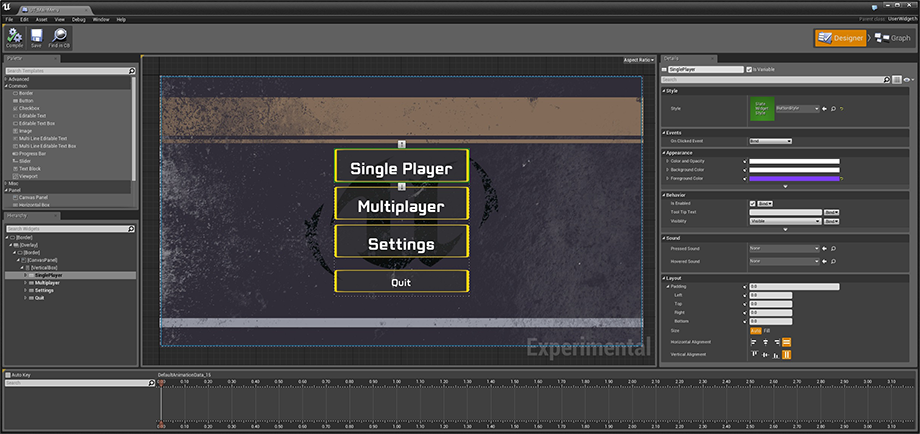


Рисунок 1.4 – Пользовательский интерфейс *Unreal Engine*

Движку *Unreal Engine* сложно бороться с большим количеством данных. В момент проектирования больших игровых уровней способность хорошо их редактировать в редакторе начинает страдать, а во время отладки возникают проблемы с производительностью, которые трудно решить. *UE*4 не предназначен для *RPG* игр, визуализации и управления огромным количеством оружия, брони, расходных материалов, разговоров и так далее. К примеру, игра *Fortnite*, разработанная на этом движке, представляет собой большую игру с открытым миром, но в ней намного меньше данных, чем в большинстве проектов своего жанра. Она не вызывает заметных зависаний, но стоит её ещё немного расширить новым игровым контентом, как проблемы тут же дадут о себе знать. *Unreal Engine* нуждается в точной настройке для удовлетворения нестандартных потребностей разработчика, а чтобы правильно настроить движок, потребуется много времени и нервов. К тому же, это сложно будет сделать без технического образования. В то время как *UE*4 обслуживает менее опытных разработчиков благодаря технологии *Blueprint*, некоторые аспекты движка остаются трудными для понимания пользователям, у которых отсутствует хорошая техническая подготовка. Наличие серьёзного опыта программирования на *С*++ поможет с правильной настройкой окружения, но даже с этим багажом знаний *UE*4 имеет довольно крутую кривую обучения.

# 1.3 Сравнение игровых движков *Unity* и *Unreal Engine* 4

Для того чтобы подытожить и выяснить какой игровой движок оптимальнее использовать при разработке игрового приложения жанра «сетевой шутер» необходимо разобрать плюсы и минусы каждого по отдельности.

К плюсам игрового движка *Unity* можно отнести следующее:

* кроссплатформенный;
* позволяет быстро разрабатывать игры за счёт готовых шаблонов (ассетов);
* содержит встроенный магазин ассетов;
* хорошо подходит для создания небольших проектов;
* имеет низкий порог вхождения;
* реализует конструктивное создание объектов, что значительно упрощает разработку;
* наличие большого комьюнити, видеоуроков и документации;
* имеет дружелюбный интерфейс;
* является условно бесплатным.

Минусы игрового движка *Unity*:

* закрытый исходный код;
* плохо оптимизирован для разработки больших игровых проектов с открытым миром.

К плюсам игрового движка *Unreal Engine* можно отнести:

* возможность создавать фотореалистичные проекты;
* кроссплатформенная разработка;
* открытый исходный код;
* бесплатный;
* наличие технологии конструктивной разработки игр *Blueprint*;
* дружелюбный интерфейс;
* наличие большого комьюнити.

Помимо многочисленных плюсов *Unreal Engine* также имеет ряд серьёзных недостатков:

* проседание в производительности при большой нагрузке данными;
* наличие проблем с отладкой больших игр;
* высокий порог вхождения;
* движок имеет избыточное число инструментов для разработки маленьких проектов;
* сложная индивидуальная настройка редактора и функций движка.

Из перечисленных положительных и отрицательных сторон рассматриваемых игровых движков можно сделать вывод, что для разработки простой некоммерческой игры в рамках курсового проекта оптимальнее будет использовать движок *Unity*, так как он позволяет больше времени уделить проектированию игры и непосредственно её дизайну, нежели настройке движка и редактора под собственные нужды. К тому же, большинство инструментов *Unreal Engine* будут излишними для разработки мелкой сетевой игры.

* 1. **Актуальность разработки компьютерных игр в настоящее время**

С каждым годом популярность компьютерных игр растет, все больше и больше людей играют в компьютерные игры. Результатом технологического развития, а также массового распространения персональных компьютеров, планшетов и мобильных устройств, стал все более расширяющийся рынок развлечений, что является веским аргументом в пользу проектов по созданию новых игр.

Актуальность разработки игр становится все более неоспоримой, рынок игр увеличивается в геометрической прогрессии. Самые удачные проекты продаются миллионными тиражами. Зарождение игровой индустрии началось в 1970-х годах одновременно с продажей первых компьютеров, рассчитанных на массового потребителя.

За 40 с небольшим лет индустрия компьютерных игр развилась до таких масштабов, что уже во многом опережает своих ближайших конкурентов: киноиндустрию, музыкальную индустрию, шоу-бизнес. Многие разработчики рассматривают индустрию компьютерных игр лишь как сектор экономики, в котором можно неплохо заработать. Но, на самом деле, создание игр – это ещё и инструмент культуры. Ретро игры были придуманы много лет назад, но и сейчас они пользуются большой востребованностью. Яркий пример – *Mario*, серия компьютерных аркад, в которые с удовольствием играют пользователи нескольких поколений. Но она бы не была такой популярной, если бы не имела развитие. «*Super Mario Maker*» – игра, в которой игрок имеет возможность не только проходить уровни, но и создавать свои собственные [4].

На сегодняшний день возрождение старых игр может являться актуальным и выгодным проектом. На рисунке 1.5 представлена текущая динамика доходов игровой индустрии, а также приблизительные планы развития на ближайшую пару лет:



Рисунок 1.5 – Динамика доходов игровой индустрии по категориям в мире

Исходя из данных представленных на диаграмме можно сделать вывод, что разработка новых игровых приложений будет актуальна и прибыльна ещё долгие годы, а производство новой и более мощной вычислительной техники позволит не только удовлетворить все потребности потребителей, но и удешевить разработку игр, что, в свою очередь, увеличит доход игровых разработчиков.

# Список используемых источников

1. Хокинг Д. Unity в действии. Мультиплатформенная разработка на C# / Хокинг, Д. – М.: СПб: Питер, 2019. – 58 c.
2. Гибсон Б. Д. Unity и C#. Геймдев от идеи до реализации / Гибсон, Б. Д. – М.: СПб: Питер, 2019. – 112 c.
3. Куксон А. Разработка игр на Unreal Engine 4 за 24 часа / Куксон, А. – М.: Бомбора, 2019. – 98 c.
4. Костер Р. Разработка игр и теория развлечений / Костер, Р. – М.: ДМК, 2018. – 113 c.