Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Российский государственный профессионально-педагогический университет»

Институт инженерно-педагогического образования

Кафедра информационных систем и технологий

**Отчет**

по практике

**«Преддипломная практика»**

место прохождения практики:

ФГАОУ ВО РГППУ, ИПО, кафедра информационных систем и технологий

|  |  |  |  |
| --- | --- | --- | --- |
| с | 29 апреля 2019 г. | по | 26 мая 2019 г. |

|  |  |
| --- | --- |
| Обучающийся: | Шульгин Петр Владимирович |
|  | (Фамилия, имя, отчество) |
| Группа: | ИТм-403 |
| Руководитель практики от профильной организации: | Чернякова Татьяна Викторовна |
|  | (Фамилия, имя, отчество) |
| Руководитель практики от РГППУ | Суслова Ирина Александровна |
|  | (Фамилия, имя, отчество) |

|  |  |  |  |
| --- | --- | --- | --- |
| Итоговая оценка |  |  |  |
|  | |  | (подпись руководителя практики от РГППУ) |

Екатеринбург

2019

**Содержание:**

# ****Введение****

Компьютерные игры имеют безграничные возможности для получения особого опыта времяпрепровождения.

С каждым годом появляются всё новые увлекательные и разнообразные игры, которые привлекают внимание множества людей по всему миру.

Особенностью компьютерных игр является геймплей. Он обеспечивает взаимодействие человека с игровой средой, определяет какие возможности будут доступны игроку.

# 1. Цель

Разработка увлекательной компьютерной 2D игры, которая позволит человеку провести свободное время, получая определённый опыт от прохождения.

# 2. Анализ разработок

За десятки лет было выпущено огромное количество разнообразных компьютерных игр.

Игры подразделяются на жанры, а также могут различаться по сеттингу, целям, формам представления и т.д.

Жанры игр формировались бесструктурно и интуитивно на протяжении длительного времени. Разработчики игр проводили смелые эксперименты, создавая новые игровые механики.

**Основные жанры компьютерных игр:**

* приключенческая игра (англ. Adventure) — главной частью игры является история;
* экшен (англ. Action) — игра, характеризующая частым и активным нажатием кнопок управления;
* ролевая игра (англ. RPG) – в ролевой игре игрок управляет одним или несколькими персонажами, каждый из которых описан набором численных характеристик, списком способностей и умений;
* стратегическая игра (англ. Strategy) — необходимость игроку делать нетривиальный выбор;
* компьютерный симулятор (англ. Simulator) — игрок делает множество упражнений и оттачивает свою технику;
* головоломка (англ. Puzzle) — требует аналитического мышления;
* обучающая игра (англ. Educational) — игрок обучается во время выполнения каких-либо действий в игре;
* игрушки (англ. Toys) — программы, взаимодействуя с которыми, игрок получает удовольствие.

Для разрабатываемой игры были выбраны следующие жанры: action, logic. Они полностью отражают суть игры.

**Определение сеттинга игры**

Место действие, время и реальность – это и есть то, что называется Сеттинг. Он определяет атмосферу игры, её особую и узнаваемую обстановку.

Наиболее популярные сеттинги:

* **фэнтези** (сказочный мир с наличием магии);
* **средневековье** (рыцарские походы, междоусобицы, инквизиция);
* **WWII (**игры про Вторую Мировую войну);
* **научная фантастика** (космические сражение, инопланетные расы).

Игра будет реализована в сеттинге «Средневековье».

# 3. Выбор средств разработки

Компьютерная игра является программой. Для её реализации требуется написать множество программного кода, который будет определять все взаимодействия между объектами (графика, аудио, сценарий) и возможностями игрока.

Есть два варианта решения задачи:

1. Писать всё самостоятельно, выбрав язык программирования.
2. Использовать готовый игровой движок.

Второй вариант позволяет существенно сэкономить время разработки. Поэтому следует выбрать его, но для этого потребуется ознакомится с современными решениями.

**Unreal Engine** – движок позволяет создавать игры для большинства операционных систем и платформ: Microsoft Windows, Linux, Mac OS и Mac OS X; консолей Xbox, Xbox 360, Xbox One, PlayStation 2, PlayStation 3, PlayStation 4, PSP, PS Vita, Wii, Dreamcast, GameCube и др., а также на различных портативных устройствах, например, устройствах Apple (iPad, iPhone), управляемых системой iOS и прочих.

Для упрощения портирования движок использует модульную систему зависимых компонентов; поддерживает различные системы рендеринга (Direct3D, OpenGL, Pixomatic; в ранних версиях: Glide, S3, PowerVR), воспроизведения звука (EAX, OpenAL, DirectSound3D; ранее: A3D), средства голосового воспроизведения текста, распознавание речи, модули для работы с сетью и поддержки различных устройств ввода.

**Unity3D** – это игровой движок, позволяющий создавать игры под большинство популярных платформ. С помощью данного движка разрабатываются игры на персональные компьютеры (работающих под Windows, MacOS, Linux), на смартфонах и планшетах (iOS, Android, Windows Phone), на игровых консолях (PS, Xbox, Wii).

Этот игровой движок крайне популярен среди инди-разработчиков, стартаперов и примыкающих к ним школьников. Имеет очень удобную систему для разработки различных игр.

В данном случае был выбран Unity3D, с его помощью проще и качественнее разрабатывать 2D игры.

# 4. Творческий процесс

# 4.1 Описание игры

Игру можно начинать с продумывания геймплея или же сначала написать историю. Т.к. уже определены жанр и сеттинг игры, можно создать небольшое сюжетное описание игры.

В королевстве Этельсторф каждые 50 лет проводится избрание нового героя королевства, который будет защищать всех жителей от внешних угроз

Игроку предстоит победить в турнире Избрания Героя. Бой проходит 1 на 1 на ограниченной арене. С помощью меча и щита можно наносить и блокировать удары.

# 4.2 Сюжет

История повествует о бывшем рыцаре одного графа, Алане Рутланде. Хороший воин, но уйдя в отставку, он особо не занимался военными тренировками, что сказалось на его боевой форме.

Годелотт Гесс относительно недавно стал графом и желает укрепить свои позиции среди остальных, а может даже стать кем-то большим, чем Граф. Он решает учувствовать в выборе героя, но сам не является сильным воином, поэтому обращается к бывшему рыцарю своего отца Алану Рутланду. По началу, Алан отказывается, но граф его переубеждает.

Алану предстоит выиграть турнир, одолев соперников, чтобы заслужить право стать героем.

# 4.3 Игровое взаимодействие

Игровая механика – это правила, которые затрагивают игроков, аватары, игровые частицы, игровое состояние, поле обозрения и описывают все способы изменения игрового состояния.

Механики — ингредиенты геймдизайна. Их понимание крайне необходимо для всех геймдизайнеров.

**Механики:**

* передвижение по горизонтали (впаво/влево);
* прыжок;
* удар и нанесение урона (уменьшение здоровья врага);
* получение урона (уменьшение здоровья игрока);
* блокирование урона (блокировать урон щитом);

# 5. Наполнение игры контентом

При разработке игры нужно написать множество скриптов, создать или использовать готовые спрайты, создать анимацию, создать звуковое сопровождение.

# 5.1 Написание скриптов

Игровой движок упрощает разработку игру, но полностью не решает проблему написания кода, а также контроля изменений.

**Технологии программирования**

**Microsoft Visual Studio** — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

**GitHub** – веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub. Сервис бесплатен для проектов с открытым исходным кодом и небольших частных проектов, предоставляя им все возможности, а для крупных корпоративных проектов предлагаются различные платные тарифные планы.

Системы контроля версий (СКВ, VCS, Version Control Systems) позволяют разработчикам сохранять все изменения, внесённые в код. Поэтому они могут просто откатить код до рабочего состояния вместо того, чтобы тратить часы на поиски маленькой ошибки или ошибок, ломающих весь код.

**Скрипты**

Для любого действия в игре необходимо написать скрипт. Для игры были скрипты управления персонажем, поведение врага, управление звука, контроллер камеры и др.:

**PlayerScript –** это скрипт управление персонажем, он включает в себя методы по передвижению персонажа по горизонтали, метод прыжка, получение и нанесение урона, а также отвечает за контроль анимации;

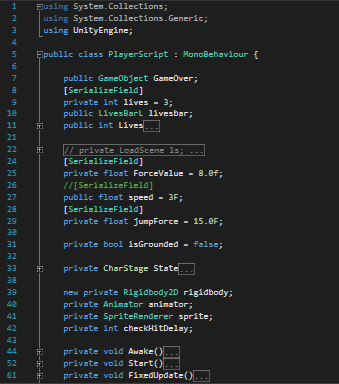


Рисунок 1 – PlayerScript

**EnemyL** – скрипт, отвечающий за действия и поведение врага;

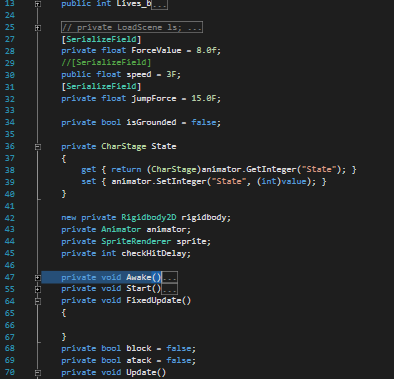


Рисунок 2 – EnemyL

**CameraController** – этот скрипт управляет камерой, которая отслеживает положение игрока;

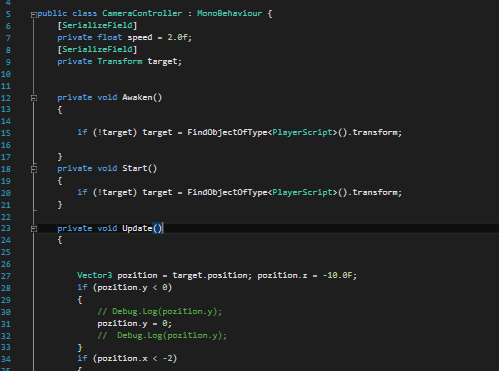


Рисунок 3 – CameraController

**Sound –** скрипт отвечает за воспроизведение и остановку фоновой музыки.

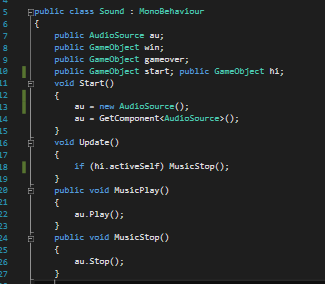


Рисунок 4 – Sound

# 5.2 Создание спрайтов и анимации

Спрайты – растровые или векторные изображения, используемые в разработке компьютерных игр (в основном 2D). Они определяют внешний вид объектов на сцене.

**Программы по работе с графикой**

Для создания и редактирования графики, следует использовать специальное программное обеспечение.

**Adobe Photoshop** — многофункциональный графический редактор, разработанный и распространяемый фирмой Adobe Systems. В основном работает с растровыми изображениями, однако имеет некоторые векторные инструменты;

**Adobe Illustrator** — векторный графический редактор, разработанный и распространяемый фирмой Adobe Systems;

**GNU Image Manipulation Program** или **GIMP** — свободно распространяемый растровый графический редактор, программа для создания и обработки растровой графики и частичной поддержкой работы с векторной графикой;

**Krita** — бесплатный растровый графический редактор с открытым кодом, программное обеспечение, входящее в состав KDE. Ранее распространялось как часть офисного пакета Calligra Suite, но впоследствии отделилось от проекта и стало развиваться самостоятельно.

**Autodesk SketchBook** — это сочетание растрового и векторного редакторов. Программа позволяет создавать отличные скетчи. Наличие множества инструментов рисования.

**Спрайты**

Для создания спрайта необходим объект на сцене, а также изображение, содержащее один или несколько элементов, которые будут использоваться в качестве спрайтов.

Для персонажей, земли и стен созданы свои спрайты, по изображениям из открытых источников.



Рисунок 5 – Спрайты мечника

**Фоны**

Для каждого уровня были подобраны фоны.

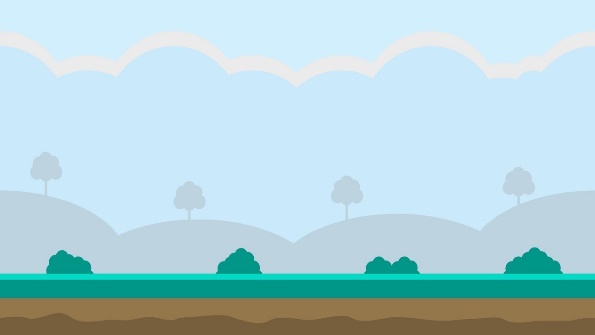


Рисунок 6 Фоны уровней

**Анимация**

Анимация – это набор нескольких спрайтов, размещённых на временной шкале. Любые действия персонажей нуждаются в анимации.

В игре созданы анимации персонажей в 5 стадий:

* idle – анимация бездействия;
* run – анимация передвижения;
* jump – анимация прыжка;
* attack – анимация атаки;
* block – блокирование;
* dead – смерть персонажа.



Рисунок 7 – Анимация атаки

# 5.3 Создание звукового сопровождения

Хорошее музыкальное и звуковое сопряжение создает определённую атмосферу и настрой всей игры.

**Программы по работе с аудио**

**Adobe Audition** (бывший **Cool Edit Pro**) — аудиоредактор производства фирмы Adobe. Поддерживает практически все звуковые форматы файлов. Входит в пакет Adobe Creative Suite, а также в Adobe Creative Cloud.

**FL Studio** — цифровая звуковая рабочая станция и секвенсер для написания музыки. Музыка создаётся путём записи и сведения аудио- или MIDI-материала. Готовая композиция может быть записана в формате WAV, FLAC, MP3 или OGG.

Для игры была подобрана фоновая музыка из бесплатных источников. Также музыка была обработана в аудиоредакторе и импортирована в проект.

# 6. Создание уровней

Уровень — в компьютерных играх — отдельная область виртуального мира игры, обычно представляет собой определённую локацию, например, здание или город

Содержание уровня:

* игровой персонаж – это персонаж, которым управляет игрок;
* неигровой персонаж (враг) – запрограммированный на определённые действия враг;
* земля – площадка, на которой могут находиться персонажи и объекты, не падая вниз;
* стены – служат ограничителями локации;
* фон.

# 7. Сборка проекта

Закончив создание игры, проект нужно собрать в готовое приложение. В Unity3D это можно сделать довольно просто. Сначала все уровни (сцены) добавляются в список сборки, затем в настройках сборки определяется платформа. В данном случае, игра разрабатывалась для ПК с операционной системой семейства Windows.

Закончив с настройками, запускается процесс сборки. Он может занять много времени (зависит от мощности ПК разработчика).

В итоге будет создана новая папка с готовой игрой, которую можно запускать.

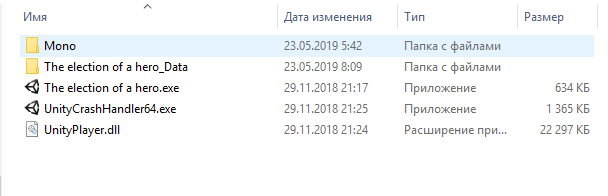


Рисунок 8 – Файлы готовой игры

# 8. Тестирование и оптимизация

Каждый готовый программный продукт должен пройти тестирование на той платформе, для которой разрабатывался. В производстве игр применяются несколько типов тестирования, каждый из которых предназначен для выявления соответствующих данному типу недостатков.

# 8.1 Тестирование функциональности

Цель – выявить отклонения от требуемой функциональности и ошибки или так называемые «баги». Как правило, этот тип тестирования не требует от тестировщика больших технических познаний, помимо понимания базовых концепций программирования и сводится к многократному прохождению игры, выявлению неполадок и условий, в которых их можно воспроизвести.

Этот вид тестов является общим для всего ПО, поэтому нецелесообразно останавливаться на нем подробно.

# 8.2 Тестирование соответствия аппаратному обеспечению

Программирование игр, как и любого другого ПО, производится на персональных компьютерах – настольных или ноутбуках.

Первоначально разработка осуществляется с помощью симуляторов указанных устройств, однако, как и всякая модель, они отличаются от оригинала, зачастую существенно. Поэтому игра, успешно работающая на симуляторе, может иметь проблемы при запуске на устройстве.

# 8.3 Нагрузочное тестирование

Часто бывает, что программа, нормально работающая в большинстве ситуаций, обнаруживает проблемы при эффективных вычислениях. Этот вид проблем особенно неприятен, поскольку интенсивность вычислений может не зависеть от логики игры. Кроме того, игровые ситуации с большим объемом вычислений может не заметить тестировщик.

Поэтому целесообразно создать тестовые ситуации, требующие большой вычислительной нагрузки. Так гораздо легче заметить и улучшить имеющиеся потенциально ненадежные участки кода.

# 8.4 Оптимизация

По завершению тестирования, собираются и анализируется результаты, выявленные баги устраняются.