СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Темой курсового проекта является «Разработка игрового приложения «Extrandium».

С развитием технологий и появлением новых игровых платформ, таких как виртуальная реальность и дополненная реальность, компьютерные игры стали ещё более захватывающими и реалистичными. Игроки могут погрузиться в виртуальные миры, где они могут взаимодействовать с другими игроками со всего мира, создавать исключительные персонажи. С развитием новых технологий и растущей популярностью компьютерных игр, люди могут наслаждаться эмоциями, которые ранее можно было испытать только в реальной жизни. Несмотря на это, важно помнить о необходимости поддерживать здоровый баланс между игровым временем и другими аспектами жизни. Компьютерные игры также имеют множество положительных аспектов, они могут быть использованы в образовательных целях, помогая развивать навыки решения задач, логическое мышление и творческое мышление.

Целью данного курсового проекта является создание 3D игры для ОС Windows.

Объектом исследования являются игры для ОС Windows.

Предметом данного курсового проекта являются особенности реализации игрового приложения для операционной системы Windows, а также средства реализации программного продукта.

Для достижения поставленной цели в проекте были рассмотрены и решены следующие задачи:

- выбран жанр для игрового приложения;
- спроектирована структура и функциональность игры;
- спроектированы история, персонажи и уровни;
- выбраны средства реализации компьютерных игр для ОС Windows;
- определены минимальные требования к техническим и программным средствам;

– описаны входные и выходные данные;

– разработан программный код приложения;

– создана справочная система;

– создана инсталляция.

Для решения поставленных задач использовались методологии проектирования функциональности приложения, особенности реализации программного кода с помощью конкретного языка программирования C#, принципы и алгоритмы применения инструментальных средств на различных этапах разработки программного продукта.

Данный программный продукт предназначен для широкого круга пользователей и может использоваться коммерчески только с помощью встроенной рекламы.

1 ТЕХНИЧЕСКИЙ ПРОЕКТ

1.1 Анализ предметной области

Игровые жанры формировались органично и эмпирически на протяжении значительного времени. Разработчики игр отважно экспериментировали, создавая новые игровые механики. Неудачные эксперименты быстро забывались, а успешные игры становились вдохновением для других разработчиков. Это приводило к тому, что разработчики начинали копировать популярные игровые механики и добавляли в них свои инновационные идеи. Именно таким образом возникали различные классы игр, которые получали название игровых жанров. В отличие от классических жанров фильмов и книг, жанры игр подразумевают под собой совсем другие особенности. Они определяются не по сюжету, а по игровым действиям, которые наиболее часто совершает игрок.

Наличие жанров в игровой индустрии действительно играет важную роль. Жанры позволяют игрокам быстро ориентироваться и понимать, что ожидать от конкретной игры. Как разработчик, важно учитывать предпочтения аудитории и создавать игры, которые бы соответствовали ожиданиям игроков этого жанра. Однако, при разработке новых и инновационных игровых продуктов, я считаю, что важно также удивлять и удовлетворять игроков новыми и нестандартными идеями. Иногда можно слегка отступить от четкого определения жанра и предложить что-то свежее и уникальное. Это может привлечь внимание новой аудитории и добавить больше интереса к игре.

Основные игровые жанры рассмотрим ниже.

Аркады– распространённый в постсоветской индустрии компьютерных игр термин, обозначающий компьютерные игры с нарочно примитивным игровым процессом. Задачей игр этого жанра будет набор максимального количества очков и/или минимальное время прохождения игры.

Приключенческие игры (квесты) – сюжетные игры, приключение, бродилка, квест англ. quest — поиски) — один из основных жанров игр, требующих от игрока решения умственных задач для продвижения по сюжету. Сюжет может быть предопределённым или же давать множество исходов, выбор которых зависит от действий игрока.

Ролевые игры (англ. RPG – Role Playing Games) основанный на элементах игрового процесса традиционных настольных ролевых игр. Отличительной чертой жанра является свобода в определении развития персонажа (или группы персонажей), сюжетного поведения и характера главного героя (героев) в соответствии с индивидуальными склонностями самого игрока, особенностями игрового сеттинга и действующей в игре системы правил. Игра строится на том, что игрок постоянно сталкивается с вопросом выбора (как в направлении развития, так и в направлении сюжетного поведения), последствия которого заметно меняют игровой процесс и финальный результат игры. Притягательность таких игр во многом определяется адекватностью искусственного интеллекта, управляющего поведением прочих персонажей.

Развитие Интернета привело к появлению разновидности ролевых игр – массовых многопользовательских ролевых онлайн-игр (ММОРПГ). В этом виде игровых программ виртуальные персонажи, управляемые реальными игроками, способны взаимодействовать друг с другом.

Симуляторы – жанр компьютерных игр, предназначенных для тщательного моделирования реальной деятельности. Симулятор пытается скопировать различные действия из реальной жизни для различных целей, таких как обучение, анализ, прогнозирование или развлечения. Обычно в игре нет строго определённых целей, игроку разрешено свободно управлять персонажем или окружающей средой.

Стратегические игры – задача этого жанра сводится к получению преимущества над противником, достигаемого путем выработки и реализации определенного плана. Объектами управления являются не отдельные

персонажи, а целые корпорации, армии, государства и даже цивилизации. Различают пошаговые стратегические игры, где игроки ходят поочередно и стратегии реального времени (RTS – Real-time strategy), в которых все игроки действуют асинхронно.

Боевики – жанр компьютерных игр, в котором делается упор на эксплуатацию физических возможностей игрока, таких как зрительно-моторная координация и скорость реакции. Жанр включает в себя большое количество поджанров — от файтингов, шутеров и платформеров, которые считаются наиболее важными для жанра, до MOBA и некоторых стратегий в реальном времени, которые также возможно отнести к жанру экшен.

Раннер (Runner и Endless Runner) – жанры игр, в центре внимания которых герой-бегун. Практически всегда персонаж бежит автоматически и от игрока требуется лишь корректировать его траекторию движения, вовремя совершать прыжки и уклоняться от препятствий. Это требует концентрации внимания и хорошей реакции.

Рогалик (Roguelike) – это игры с процедурной генерацией. С помощью алгоритмов содержимое таких видеоигр создается случайно при каждом новом прохождении. Изначально нишевый жанр, он вскоре захватил не только бюджетные инди, но и крупные блокбастеры: процедурная генерация используется в видеоиграх всех жанров и масштабов.

Хоррор (Horror) (с англ. — «ужас ») — жанр компьютерных игр, для которого характерными являются упор на выживание игрового персонажа и нагнетание атмосферы страха и тревоги, подобно литературе и фильмам ужасов.

В курсовом проекте будет реализовываться игра в жанре Приключение.

Операционная Система (ОС) – комплекс программ, обеспечивающий управление аппаратными средствами компьютера, организующий работу с файлами и выполнение прикладных программ, осуществляющий ввод и вывод данных.

Сегодня наиболее известными ОС являются системы семейства Microsoft Windows, Linux, iOS и Android.

Microsoft Windows представляет целую серию ОС и рабочих сред, разработанных корпорацией Microsoft. Самая первая версия ОС Windows – MS-DOS, была представлена в 1985 году, имела графический пользовательский интерфейс. В ней была обеспеченна поддержка нескольких документов, поддержка компьютерной мыши, выпадающее меню, и всё это было возможным увидеть в цветах.

Следующие версии Windows постепенно заменяли многое построенное в MS-DOS, аппаратный и программный функционал. Все внесённые Microsoft изменения в MS-DOS с дальнейшей интеграцией поспособствовали получению развитой ОС.

В настоящее время Microsoft Windows является самой популярной ОС, не только из-за привычного удобства для использования и внушительного функционала, но и из-за высокого уровня интеграции с возможностями её ядра и другого программного обеспечения, в том числе MS Office.

Linux – общее название UNIX-подобных ОС на основе одноимённого ядра и собранных для него библиотек и системных программ. К ОС GNU/Linux также часто относят программы, дополняющие эту ОС, и прикладные программы, делающие её полноценной многофункциональной ОС. В отличие от большинства других ОС, GNU/Linux поставляется в большом количестве так называемых дистрибутивов, в которых программы GNU соединяются с ядром Linux и другими программами.

Игра будет реализовываться для ОС Windows.

1.2 Постановка задачи

Необходимо разработать и создать 3D-игру в жанре Приключение для ОС Windows.

Приложение должно иметь следующую структуру:

– главное меню;

– игровые уровни.

Приложение должно иметь дружественный интерфейс и обеспечивать:

– интерфейс главного меню (начать, настройки, выход);

– отображение справочного окна;

– музыкальное сопровождение;

– игровой интерфейс.

При создании игрового приложения необходимо будет:

– создать сценарий игры и разработать её логику;

– спроектировать персонажа и уровни;

– разработать функционал игры;

– разработать графический интерфейс пользователя;

– прорисовать главного героя;

– разработать систему хранения инвентаря.

Способ решения поставленных задач зависит от выбора среды разработки и языка программирования.

Тестирование будет проводиться на персональном компьютере с 64-разрядной ОС Windows 10, процессором IntelCore i3-10105f, CPU 3.70 GHz, 16Гб RAM, дисплеем VGA с разрешением не более 1920x1080 точек на дюйм, видеокарта(amd radion hd 7700 series), оснащенном клавиатурой(hp ku-1156) , мышью(sven rx-g750) , звуко-выводящие устройства.

1.3 Выбор средств реализации

Прежде чем начать создавать новую игру, необходимо выбрать подходящий игровой движок, который предоставляет программные инструменты для разработки и запуска видеоигр. Игровой движок включает в себя компоненты, необходимые для создания различных аспектов игры, включая обработку изображений, физическую моделирование поведения объектов, звуковое оформление, скриптинг, создание искусственного интеллекта и работы с сетевыми аспектами.

На данный момент в мире существует огромное количество игровых движков. Рассмотрим наиболее популярные из них – UnrealEngine и Unity.

Unity – это кроссплатформенный игровой движок для разработки двухмерных и трехмерных приложений и игр. Unity3d имеет очень простой Drag and Drop интерфейс, который человек осваивает за месяц. Весь движок только на английском языке. Русификации Unity5 нет.

Интерфейс Unity разбит на несколько окон:

- Hierarchy – здесь находятся названия всех объектов на сцене, которые можно группировать и легко переходить по ним;
- Scene – здесь можно рассмотреть определенную сцену под нужным ракурсом;
- Inspector – поможет с настройкой выделенного объекта;
- Project – в нем отображаются все материалы проекта;
- Toolbar– содержит все доступные инструменты.

Unity 5 поддерживает использование двух языков программирования: C# и графические ноды(Shader graph). Разработчику необходимо знать один из языков в совершенстве, а другой на среднем уровне, так как некоторые моменты Unity 5 делает только на одном из двух языков, или это делается намного труднее, чем на другом языке программирования.

Для разработки 3D игры Unity 5 подходит по всем параметрам. В нем очень просто собирать проекты. Причем можно создать один проект под множество платформ, что очень сильно облегчает процесс дальнейшего распространения игр. Все скрипты, используемые в Unity 4, можно будет автоматически исправить в Unity 5.

Для данного курсового проекта был выбран движок Unity 5. Главные аргументы выбора – это удобство и скорость разработки под различные платформы, а также основной язык программирования C#.

1.4 Проектирование и моделирование игрового приложения

1.4.1 Сценарий и стратегия игры

Персонаж игры является наследником дома, который принадлежит его семье уже много поколений. Это прекрасное родовое имущество, полное историй и тайн, которое всегда мечтали исследовать.

Однако, пока наследник рос, дом был недоступным для него, так как находился под опекой и требовал большого ремонта.

Повзрослев, одним летним днем наследник решили отправиться в этот дом. Его все время привлекало его величие и загадочность. Персонаж взял свою машину и направились в сторону дома, жажду исследования уже невозможно было подавить.

Для вступления в наследство, чтобы его получить главному герою нужно пройти испытания, которое родственники подготовили.

После успешного завершения испытаний главный герой получит документ на дом и будет там проживать.

Если испытания не пройдены, то игра завершается.

1.4.2 Проектирование функциональности программы

Разработка функции игрового персонажа, управлять которым будет пользователь;

У героя будут следующие состояния:

– бездействие;

– передвижение.

У героя возможны следующие функции:

– умереть;

– восстановить здоровье;

– сместиться вправо, влево, назад, вперёд

Разработка и создание графических объектов игрового приложения разделена на следующие направления:

– прорисовка игровой локации Улица (рисунок 1);
– прорисовка игровой локации Дом сверху (рисунок 2);
– прорисовка игровой локации Дом сбоку (рисунок 3);
– прорисовка предметов инвентаря (рисунок 4).

1.4.3 Основные функции игрового приложения

Игровое приложение будет иметь следующие функции:
– возможность начать новую игру;
– установка на паузу;
– возможность изменить качество графики;
– возможность изменить уровень громкости звуков;
– отображения здоровья;

Новая игра, пауза, настройка графики будут реализованы в виде кнопок. Отображения здоровья будет в правом верхнем углу, при прохождении дальше, будут воспроизводиться разные пугающие ситуации и тем самым будут уменьшать его, но можно и восполнить с помощью лекарства в локации дом. Сохранения в игре не предусмотрено.

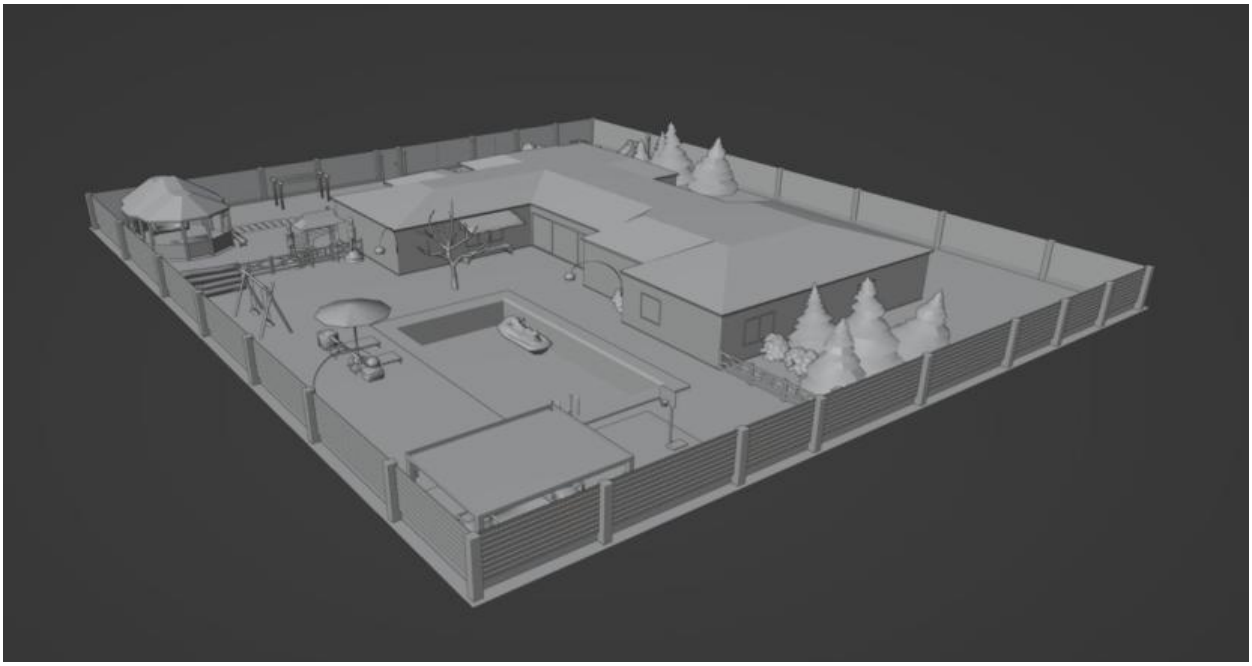

Рисунок 1 – Игровая локация Улица

Рисунок 2 – Игровая локация Дом сверху

Рисунок 3 – Игровая локация Дом сбоку

Рисунок 4 – Предметы в инвентаря

1.5 Требования к программному обеспечению и техническим средствам

Для корректной и комфортной разработки игрового приложения необходимо, чтобы компьютер удовлетворял следующим минимальным системным требованиям:

- процессор 3 GHz и выше;
- 8 Gb RAM;
- свободное пространство на диске: 128 Гб и больше;
- дисплей SVGA с разрешением не менее 1920x1080 точек на дюйм;
- клавиатура и мышь
- звуко-выводящие устройства.

Для разработки приложения на компьютере должно быть установлено следующее программное обеспечение:

- ОС Windows 10;
- DirectX 11.0c;

– Visual Studio 2022;

– среда разработки Unity5.

– графически редактор Blender.

Для обеспечения корректной работы игрового приложения, необходимо чтобы компьютер был оснащен:

– процессором Intel Core 2 Duo 2, 2 GHz;

– оперативной памятью не менее 4 ГБ;

– дисплеем не менее 1280x720 точек;

– видеокартой с 2 Гб;

– местом на жестком диске 900 МБ;

– манипулятором «мышь»;

– клавиатурой.

– звуко-выводящие устройства.

2 РАБОЧИЙ ПРОЕКТ

2.1 Стандартные объекты игрового приложения

На основе технического проекта было разработано игровое приложение «Extrandium» для ОС Windows.

Для создания игры была использована среда разработки VisualStudio 2022, язык программирования C# и движок Unity5, а также ресурсы, взятые из открытых источников.

В программе используются следующие системные классы C#:

- System – содержит фундаментальные и базовые классы, определяющие часто используемые типы значений и ссылочных данных, события и обработчики событий, интерфейсы, атрибуты и исключения обработки;
- System.Collections – содержит функции для работы с массивами, списками и прочими коллекциями;
- System.Collections.Generic – системный класс, предоставляющий функционал для работы с динамическим типом.

В движке Unity5 применяется объектно-ориентированный подход, при котором используется множество классов, каждый объект является сочетанием множества скриптов, которые добавляют им функциональности и могут подключаться и отключаться по требованию.

В программе используются следующие классы, предоставляемые движком Unity:

- UnityEngine.SceneManagement – класс, позволяющий управлять сценами, переключаться между ними, перемещать объекты между сценами и многое другое;
- UnityEngine – содержит основные функции для работы с движком;
- UnityEngine.UI – предоставляет типы и функции для работы с графическим интерфейсом игрового приложения.

Самые часто используемые методы:

– void Start() – вызывается во фрейме, когда сценарий включен непосредственно перед тем, как любой из методов Update вызывается в первый раз;
– void Update() – вызывает каждый кадр, если MonoBehaviour включен;
– public Component GetComponent(Type type) – возвращает компонент Type компоненту type, если он прикреплен к игровому объекту, и NULL, если его нет;
– void OnTriggerEnter (Collider other) – Это сообщение посылается на триггер, и на принадлежащий ему rigidbody(если таковые имеются), и на rigidbody(или коллайдер, если нет rigidbody), которые соприкасаются с триггером.

Среди встроенных модификаторов игровых объектов движка Unity можно выделить следующие основные компоненты:

– AudioSource –прикрепляется к для воспроизведения звуков в трехмерной среде. Для воспроизведения 3D-звуков вам также понадобится AudioListener . Аудиопрослушиватель обычно подключается к камере, которую вы хотите использовать. Воспроизводятся ли звуки в 3D или 2D, определяется настройками AudioImporter .[6]
– Animator – компонент, позволяющий создавать анимации и менять картинки в соответствии с состоянием объекта;
– Rigidbody – компонент, отвечающий за физические свойства объекта и влияние на него гравитации;
– Collider – создает границы столкновения объекта с другими объектами.

2.2 Пользовательские объекты игрового приложения

Сцены в Unity содержат объекты игры. Они могут использоваться для создания главного меню, отдельных уровней и для других целей. Можно считать каждый файл сцены отдельным игровым уровнем. В каждой сцене

можно разместить объекты локации, заграждения, декорации, по кусочкам создавая дизайн и саму игру.

В игре есть три основные сцены:

– Главное меню;

– Дом;

– Улица;

Стартовый экран состоит из трех пунктов:

– Играть;

– Настройки;

– Выход.

Сцены являются аналогами окон в Windows и содержат в себе так называемые игровые объекты (GameObject), к которым могут применяться различные модификаторы поведения и которые могут взаимодействовать друг с другом с помощью привязанных к ним модификаторов.

Для добавления особой функциональности игровым объектам были созданы следующие скрипты:

– FPSController – контроллер игрока;

– CameraRayCast –управления камерой игрока;

– Inventorymanager –управления инвентарём игрока;

– MainMenu–контроллер главного меню;

– Indicator – определяет рассудок игрока;

Контроллер героя (класс FPSController) организует перемещение персонажа в игровом пространстве.

Методы класса:

– Awake() – обновляет настройки паузы для получения с другой сцены;

– Start() – получает информацию о физике теле и отключает курсор;

– FixedUpdate() –прочитывает перемещения игрока;

– ActivePanel() – отвечает за включения и выключения паузы;

Инвентарь (класс Inventorymanager) показывает игроку ,что у него в руках находиться.

Методы класса:

- Start() – при запуске сцены обновляет информацию ячейки инвентаря проходясь;
- AddItem() – проверяет ,если ячейка не занята ,то он передаёт информацию с взятого объекта в пустую или такого же предмета ячейку

Управления камерой игрока (класс CameraRayCast).

Методы класса:

- Update() – в нем просчитывается движения мыши и тем самым поворота камеры. При нажатии кнопки выпускается луч в направления камеры и проверяет об какой предмет он коснулся(т.е. какой скрипт у предмета) и выполняет соответствующие действия с ним;

Определяет рассудок игрока (класс Indicator).

Методы класса:

- Update() – считывает количество рассудка и при его обнулении переходит на историю с плохой концовкой

2.3 Установка и настройка игрового приложения

Для установки программы необходимо запустить файл инсталляции setup.exe. В результате в выбранном каталоге будет создана папка Extrandium, в которой будут размещены исполняемы файлы (Extrandium.exe, UnityCrashHandler32.exe, UnityCrashHandler64.exe), библиотека (UnityPlayer.dll) и служебные каталоги (MonoBleedingEdge, Extrandium _Data).

Также на рабочем столе будет создан ярлык для запуска приложения. Ярлык для деинсталляции программы будет размещен в папке программы.

Для создания инсталляции была использована программа Inno Setup. В ходе инсталляции пользователю показываются следующие окна:

– выбор языка установки;
– выбор папки установки;
– создание дополнительных ярлыков – позволяет создать ярлык на рабочем столе или отказаться от создания ярлыка;
– ход выполнения процесса установки;
– завершение установки – позволяет выбрать файлы, которые нужно запустить после установки.

После запуска программы инсталляции нужно следовать указаниям в появляющихся окнах.

2.4 Работа с игровым приложением

Для запуска игрового приложения необходимо щёлкнуть на ярлык игры Extrandium.exe. После запуска приложения открывается главное меню. При нажатии на кнопку «Новая игра», начнётся новая игра (рисунок 5).

Рисунок 5 – Главное меню приложения

При нажатии в главном меню на кнопку «Настройка» в главном меню, можно изменить разрешение , настроить громкость звуков , настроить графику игры,

выбрать язык игры и посмотреть управление (рисунок 6).

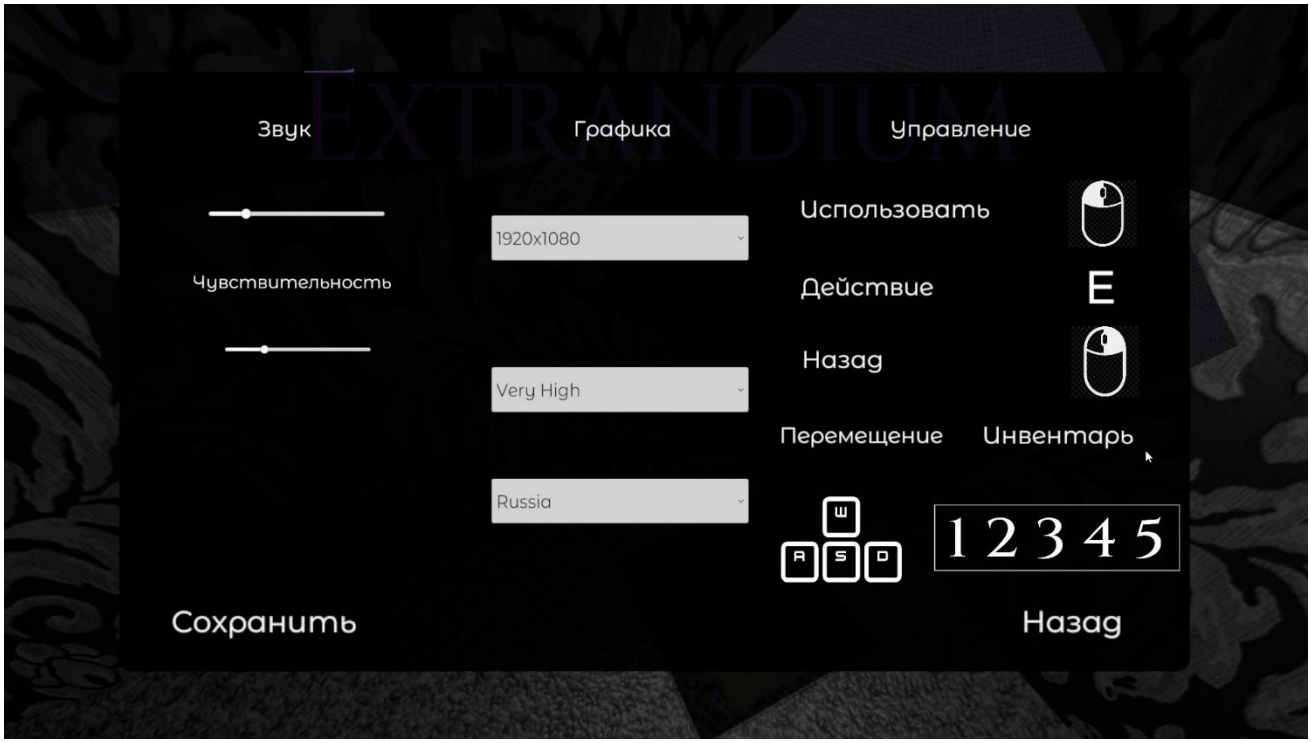

Рисунок 6 – Меню настройки

Для перемещения персонажа используются клавиши «W», «A», «S», «D» и для поворота камеры компьютерную мышь(Рисунок 7).

Рисунок 7 – Игровой уровень

При нажатии на клавишу «Escape» откроется меню паузы (рисунок 8).

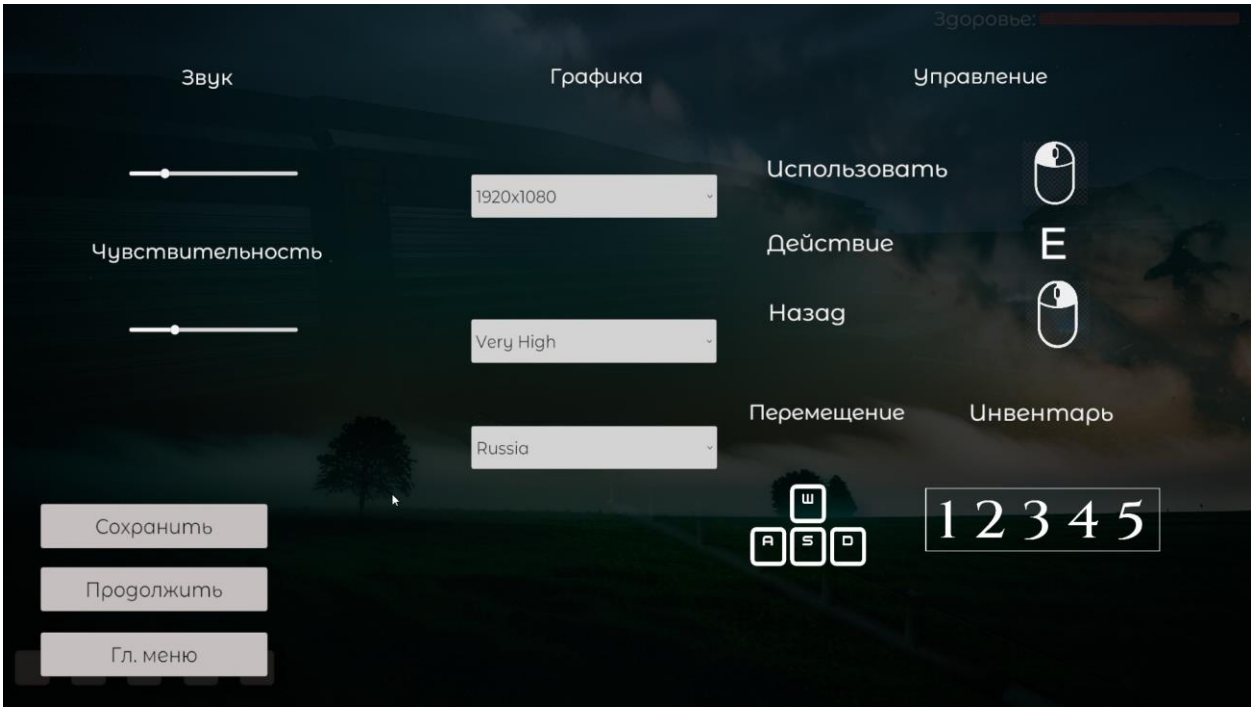

Рисунок 8 – Меню паузы

При этом игра останавливается и предлагается закрыть меню паузы (Продолжить), выйти в главное меню, начать уровень заново, либо в меню паузы настройть игру.

Чтобы закончить игру необходимо пройти все головоломки и открыть главную дверь.

ЗАКЛЮЧЕНИЕ

В результате выполнения данного курсового проекта была разработана игровая программа «Extrandium» в жанре Приключение для ОС Windows, и технический и рабочий проекты.

Данная программа включает в себя главное меню, игровую зону. Она позволяет хорошо провести время и потренировать реакцию. Удобный интерфейс программы позволяет легко ориентироваться в приложении, не требуя от пользователя каких-либо специальных навыков.

Приложение обеспечивает следующие возможности:

- просмотр управления;
- перезапуск игры;
- установку игры на паузу;
- настройка громкости звуков и музыки, разрешения экрана;

В ходе выполнения курсового проекта были достигнута поставленная цель и решены следующие задачи:

- изучены средства разработки компьютерных игр на Windows;
- выбран жанр для игрового приложения;
- спроектирована структура и функциональность игры;
- спроектированы история, персонажи и уровни;
- выбраны средства реализации;
- определены минимальные требования к техническим и программным средствам;
- описаны входные и выходные данные;
- разработан программный код приложения;
- создана инсталляция.

Основными достоинствами программы является:

- наличие простого и удобного интерфейса;
- данный программный продукт будет предназначен для широкого круга пользователей;

– имеет инсталляцию;

– наличие музыкального сопровождения.

В заключение, проделанная работа является результатом тщательного исследования и анализа темы, которая была поставлена перед нами. В ходе выполнения курсовой проекта столкнулся с различными трудностями, но смог успешно преодолеть их

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. «Unity в действии: разработка мультиплатформенных игр на C#», Джо Хокинг(дата посищение:19.12.2023)
2. «Изучение C# путем разработки игр на Unity», Харрисон Ферроне(дата посищение:19.12.2023)
3. «Освоение разработки 2D-игр на Unity», Саймон Джексон. (дата посищение:19.12.2023)
4. «Microsoft Visual C# шаг за шагом», Джон Шарп. (дата посищение:19.12.2023)
5. «Разработка игр на Unity за 24 часа», Майк Гейг. (дата посищение:19.12.2023)
6. «Разработка игр на C# для профессионалов Unity», Харрисон Ферроне(дата посищение:19.12.2023)
7. Unity Documentation: Официальная документация Unity, которая содержит подробную информацию о языке программирования C# и разработке игр на платформе Unity. Ссылка: https://docs.unity3d.com/Manual/ScriptingSection.htmlБиллиг В. Основы программирования на C#. СПб.: БХВ-Петербург, 2015. (дата посищение:19.12.2023)
8. Microsoft Developer Network (MSDN): Официальный сайт Microsoft для разработчиков, содержащий документацию по C# и .NET Framework. (дата посищение:19.12.2023). Ссылка: https://docs.microsoft.com/en-us/dotnet/csharp/МэйнС. Основы Windows Communication Foundation для .NET Framework 3.5. СПб.: БХВ-Петербург, 2015.
9. Stack Overflow: Онлайн-сообщество, где разработчики могут задавать вопросы и получать ответы от других опытных программистов. Здесь вы можете найти различные вопросы и ответы на темы, связанные с C# и Unity. Ссылка: https://stackoverflow.com/ .(дата посищение:19.12.2023)

.

ПРИЛОЖЕНИЕ А

Листинг скриптов управления главным героем

Название файл:FPSController.cs

```
public class FPSController : MonoBehaviour
{
    [SerializeField] public GameObject _panelMenu;
    [SerializeField] private GameObject inventory;
    public bool Activepanelmenu;
    [SerializeField] private Transform mainCamera;
    [SerializeField] private Rigidbody rig;
    [SerializeField] private float speed = 5;
    [SerializeField] private float _movementSpeed = 2;
    [SerializeField] private Animator animationPlayerCamera;
    [SerializeField] private AudioSource dihanieSpokoinoe;
    [SerializeField] private AudioSource shag;
    [SerializeField] public CameraRayCast cameraRayCast;
    private void Awake()
    {
        _panelMenu.SetActive(true);
        _panelMenu.SetActive(false);
    }
    void Start()
    {
        rig = GetComponent<Rigidbody>();
        Cursor.lockState = CursorLockMode.Locked;
    }
    public void Update()
    {
        if (Input.GetKeyDown(KeyCode.Escape))
```

```
        {
            ActivePanel();
        }

        if (Input.GetAxis("Horizontal") == 0 && Input.GetAxis("Vertical") ==
0)
        {
            shag.Play();
        }
        else
        {
            dihanieSpokoinoe.Play();
        }
    }
    private void FixedUpdate()
    {
        float horizontal = Input.GetAxis("Horizontal");
        float vertical = Input.GetAxis("Vertical");
        Vector3 cameraForward = mainCamera.forward;
        cameraForward.y = 0;
        Vector3 cameraRightDir = mainCamera.right;
        Vector3 movementDiraction = cameraForward.normalized * vertical +
cameraRightDir.normalized * horizontal;
        movementDiraction = Vector3.ClampMagnitude(movementDiraction,
1) * _movementSpeed;
        rig.velocity = new Vector3(movementDiraction.x, rig.velocity.y,
movementDiraction.z);
        rig.angularVelocity = new Vector3(0, 0, 0);
    }
    public void ActivePanel()
```

```
    {
        Activepanelmenu = !Activepanelmenu;
        if (Activepanelmenu)
        {
            _panelMenu.SetActive(true);
            Cursor.lockState = CursorLockMode.None;
            cameraRayCast.enabled = false;
            Time.timeScale = 0f;
        }
        else
        {
            _panelMenu.SetActive(false);
            Cursor.lockState = CursorLockMode.Locked;
            cameraRayCast.enabled = true;
            Time.timeScale = 1f;
        }
    }
}
```