

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Забайкальский государственный университет»
(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

КУРСОВОЙ ПРОЕКТ

по дисциплине: Технология разработки программного обеспечения

на тему «Компьютерная игра “Nightmare” в режиме 2D платформера в жанре приключения»

Выполнил ст. гр. ИВТ-18
Долгов А.А.

Проверил доцент кафедры ИВТ и ПМ,
к.т.н., доцент Валова О.В.

Чита

2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Забайкальский государственный университет»
(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА к курсовому проекту

по 09.03.01 Информатика и вычислительная техника

на тему: Компьютерная игра “Nightmare” в режиме 2D платформера в жанре приключения

Выполнил студент группы ИВТ-18 Долгов Александр Артемович

Руководитель работы: доцент кафедры ИВТ и ПМ, к.т.н., доцент Валова Ольга Валерьевна

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Забайкальский государственный университет»
(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

ЗАДАНИЕ

на курсовой проект

По дисциплине: Технология разработки программного обеспечения
Студенту Долгову Александру Артемовичу

направление подготовки 09.03.01 Информатика и вычислительная техника

1 Тема курсового проекта: Компьютерная игра “Nightmare” в режиме 2D платформера в жанре приключения

2 Срок подачи студентом законченной работы: 04.04.2022 г.

3 Исходные данные к проекту: описание предметной области

4 Перечень подлежащих разработке в курсовом проекте вопросов:

- а) Постановка и анализ задачи;
- б) Анализ задачи;
- в) Программная реализация;
- г) Техническое задание;
- д) Руководство пользователя;
- е) Тестирование;

5 Перечень графического материала: –

Дата выдачи задания 09.02.2022 г.

Руководитель курсового проекта _____ / Валова О.В./
(подпись, расшифровка подписи)

Задание принял к исполнению

«09» февраля 2022 г.

Подпись студента _____ / Долгов А.А./

РЕФЕРАТ

Пояснительная записка – 24 с, 6 рис., 2 таб., 3 источников

GAME DEVELOPMENT, РАЗВЛЕЧЕНИЕ, UNITY, C#,
ПРИКЛЮЧЕНИЕ, ФЭНТЕЗИ, 2D, ПЛАТФОРМЕР

Целью проекта является разработка игрового программного продукта в жанре приключения.

Поставленная цель достигается путем насыщенного игрового мира и сложности прохождения. Кроме этого были разработаны различные игровые механики.

Программный продукт предназначен для приятного время препровождения, получения эмоций и развлечения игрока в свободное время. Для достижения этой цели был придуман сюжет, который погружает пользователя в выдуманный мир и в этом мире он бросает вызов самому себе.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1. Постановка и анализ задачи	7
1.1 Описание предметной области.....	7
1.2 Постановка задачи	8
1.3 Обоснование выбора средств реализации.....	9
2. Анализ данных	11
2.1 Входные и выходные данные	11
3. Программная реализация	13
4. Тестирование.....	17
5. Техническое задание	18
6. Руководство пользователя	21
ЗАКЛЮЧЕНИЕ.....	23
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	24

ВВЕДЕНИЕ

Игровой рынок – один из самых популярных в компьютерной индустрии развлечения. С развитием индустрии появляются новые жанры, которые охватывают все большее количество игроков. О жанре игр будет говориться в предметной области. Каждый пользователь сможет найти именно то, что ему нужно, а также получить удовольствие от игры, испытав спектр эмоций. Поэтому и нужны игры для того, чтобы человек мог получить удовольствие во время игры, а также для того, чтобы он мог выбрать мир, который ему нравится и наслаждаться им в свободное время. Эта сфера подарила всем пользователям то, что они желали.

Целью работы было удовлетворение современных требований в игровой индустрии и получение опыта в разработке игрового продукта. Для реализации продукта был выбран игровой движок Unity Engine, он предоставляет своим пользователям большой набор инструментов и подходов в разработке игры [1]. Одним из главных плюсов движка является возможность создания игры в одиночку, так как разработчики постоянно добавляют новые ответы на вопросы, которые могут возникнуть у программистов во время разработки игры. Именно поэтому игровой движок с низким порогом вхождения позволяет разработчикам заниматься их работой, не боясь, что в чем-то они не смогут разобраться.

Созданный продукт развлекает игроков. Система, реализованная в проекте, легко интегрируется, модифицируется и поддерживается на этапе производства. Это позволяет в будущем дорабатывать работу для её успешной коммерциализации и публикации на игровых площадках.

1. Постановка и анализ задачи

1.1 Описание предметной области

Множество жанров и высокий спрос на них есть в игровой индустрии, поэтому она является очень большой. С помощью различных жанров можно подобрать для каждого желающего именно то, что он хочет видеть. Экшен, платформер, шутеры, файтинги, стэлсы, выживание, приключение, текстовый квест, графический квест, симулирующие и RPG – это одни из представителей жанров в игровой индустрии. По этим признакам можно понять, что игровая индустрия постоянно развивается и дополняются новыми жанрами и множеством новых механик, которые в соединении с друг другом дают игрокам незабываемые эмоции и приятное время – препровождение, а у некоторых игр являются основным доходом и источником популярности.

Задача игр состоит не в создании программы для решения конкретно поставленной проблемы, сколько для того, чтобы развлечь людей.

В проекте основным жанром является приключения.

Приключение в проекте представлено сюжетной линией главного героя, которому нужно преодолеть испытания «некой сущности» для получения возможности выбраться из своего сознания целым и невредимым. В пример я взял серию игр «Dark Souls», где основной акцент ставится на преодолении себя и исследовании мира. Эта игра не для всех потому как, она не ведет игрока «за ручку» как это делают многие другие игры. Так же в ней присутствует множество уникальных боссов, которых не одолеть с первой попытки, а на некоторых и 20 попыток будет мало. Но сложность в этой игре и является ее уникальной чертой, которая выделяет ее на фоне остальных игр.

Успешность игрового проекта зависит от сюжета, интересных механик, а также правильной порционной подачи информации игроку. Интерфейс в игре должен быть понятный, и чтобы люди тех возрастов, для которых ориентирована игра, могли в нем легко разобраться [2]. К сюжету игры

необходимо отнестись очень внимательно ведь от того, насколько хорошо он будет предоставлен игроку, от этого зависит, будет ли игрок втянут в процесс игры и погрузиться в неё.

В связи с факторами успешной игры проект реализован в жанре приключения. Это популярный жанр, объединявший в себе продуманный и глубокий сюжет вместе с интересными механиками для прохождения. В игре продуманна ее сложность, так как она является неотъемлемой частью игрового процесса. Так же в игре реализован магазин, где игрок сможет улучшить свою выживаемость и узнать больше о ситуации в которой он оказался.

Качественные и проработанные игры такого жанра требуют от исполнителей большого труда и времени. Из-за этого в выпускной квалификационной работе не может быть создан полноценный большой проект. А вот разработка версии, которая сможет дорабатываться, обновляться, и поддерживаться вполне реальная задача.

1.2 Постановка задачи

На рисунке 1.1 показана диаграмма вариантов использования для игрока.



Рисунок 1.1 Диаграмма вариантов использования

В игре предусматривается один пользователь. Возможности игрока зависят от того в каком процессе игрового приложения он находится. Во время посещения игроком меню ему предоставляется возможность продолжения игры, загрузка определенного уровня, изменения настроек, информация о разработчике, а также возможность выйти из приложения. Если игрок находится в процессе основной игры ему предоставляется возможность собирать предметы, сражаться с врагами, открывать запертые двери, использовать магазин для покупки предметов [1].

Для пользователей игра должна предоставлять следующие возможности:

- подбора игровых предметов в виде ключей, монет и пополнения здоровья игрока;
- удобный интуитивно понятный интерфейс главного меню, настроек, и игрового поля;
- взаимодействие с игровым миром;
- покупки дополнительного здоровья и возобновляющейся брони для упрощения игрового процесса;
- корректный уровень сложности, который должен увлекать пользователь;
- система сохранений покупок в магазине, количества монет и ключей, прогресса игрока, открытых частей уровня.

1.3 Обоснование выбора средств реализации

Для создания игры был использован игровой движок Unity Engine. Это очень удобный и функциональный движок, который легко осваивается. Его порог входа для новых пользователей очень низок. Помимо этого, у него есть большой набор документации и даже имеется все необходимое для создания современного игрового приложения. Большое количество разработчиков выбирают именно Unity из-за его относительной простоты, а также

возможностью поддержки всех современных платформ для игровых приложений.

Также в качестве языка написания скриптов для игрового приложения был выбран C#, благодаря его высокой совместимости с Unity [1].

2 Анализ данных

2.1 Входные и выходные данные

В игре входными параметрами являются настройки громкости музыки и звуков, качество графики и уровень на котором сейчас остановился игрок.

В игровом приложении как входные, так и выходные данные используются файлы сохранения. В файлах сохранения содержится информация о прогрессе игры, о купленных игроком дополнительном здоровье или броне, полученных предметах, открытых участках уровня. Файлы сохранения создаются автоматически при заходе игры в первый раз, либо при прохождении контрольной точки.

Сохранения игры записываются в два бинарных файла в формате «.nm», что должно немного помешать недобросовестным пользователям переписать файлы сохранения. Первый файл отвечает за сохранение таких данных как: уровень на котором остановился игрок, какую контрольную точку он прошел, что он купил в магазине, количество монет и ключей в инвентаре, открытые участки уровня. Второй файл отвечает за сохранение количества предметов на уровне.

Нажатия на клавиши так же являются входными данными:

- 1) «ESC» – открывается меню паузы;
- 2) «A» или «D» – движение игрока в лево или в право соответственно;
- 3) «Space» – прыжок;
- 4) «S» – опускание камеры вниз;
- 5) «S» + «Space» – спрыгивание с платформы;
- 6) «K» – нанесение урона врагу;
- 7) «L» – открытие участков уровня, взаимодействие с лифтом, открытие магазина.

При нажатия левой кнопкой мыши в UI где идет повествования продолжает повествование истории и делает переход.

Во время игры меняются промежуточные данные, которые были записаны в массивы разных типов. С помощью этого можно более удобно хранить всю информацию прогресса игрока. После чего все будет помещено в файл сохранения для дальнейшего использования. Такая структура позволяет сохранять большое количество изменяемых данных и не упускать мелкую информацию.

Вместе с массивами данных в классах используется Prefab это особый тип данных используемый в Unity. Prefab позволяет хранить в себе необходимую информацию об игровом объекте. Этот тип данных позволяет избегать проблемы появления множества копий одного значения в компонента так как компоненты могут получать объекты из одного файла со значениями. Это позволяет увеличить производительность.

3 Программная реализация

Программная реализация игры имеет модульную структуру. Модельная структура отображена на диаграмме компонентов в соответствии с рисунком 3.1.

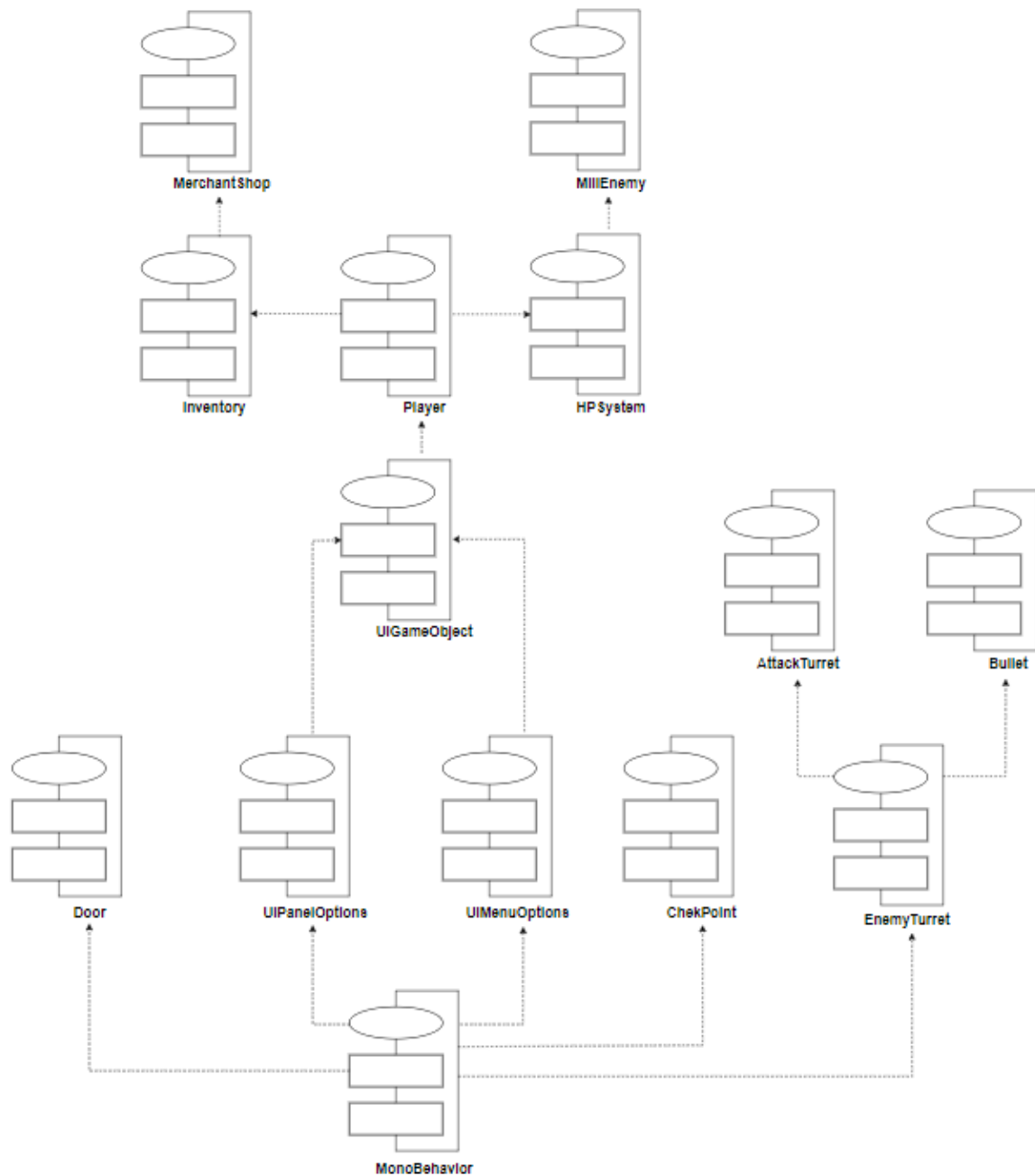


Рисунок 3.1 – Диаграмма компонентов

Таблица 3.1 – Описание компонентов

Название компонента	Описание компонента
MonoBehavior	Базовый класс Unity, от которого наследуются все остальные
Player	Компонент, отвечающий за передвижение и атаку
UiGameObject	Компонент, содержащий в себе массив объектов интерфейса
UiMenuOptions	Компонент, содержащий в себе логику интерфейса в меню
UiPanelOptions	Компонент, содержащий в себе логику интерфейса в игровом процессе
ChekPoint	Компонент, отвечающий за сохранение прогресса игрока
HPSystem	Компонент, отвечающий за здоровье игрока
Inventory	Компонент отвечает за инвентарь игрока
EnemyTurret	Компонент, отвечающий за «врага-турель»
AttackTurret	Компонент отвечает за атаку «турели»
Bullet	Компонент отвечает за то, чем стреляет «турель»
MiliEnemy	Компонент, отвечающий за патрулирующего врага
Door	Компонент, отвечающий за открытие закрытых участков уровня
MerchantShop	Компонент отвечает за магазин

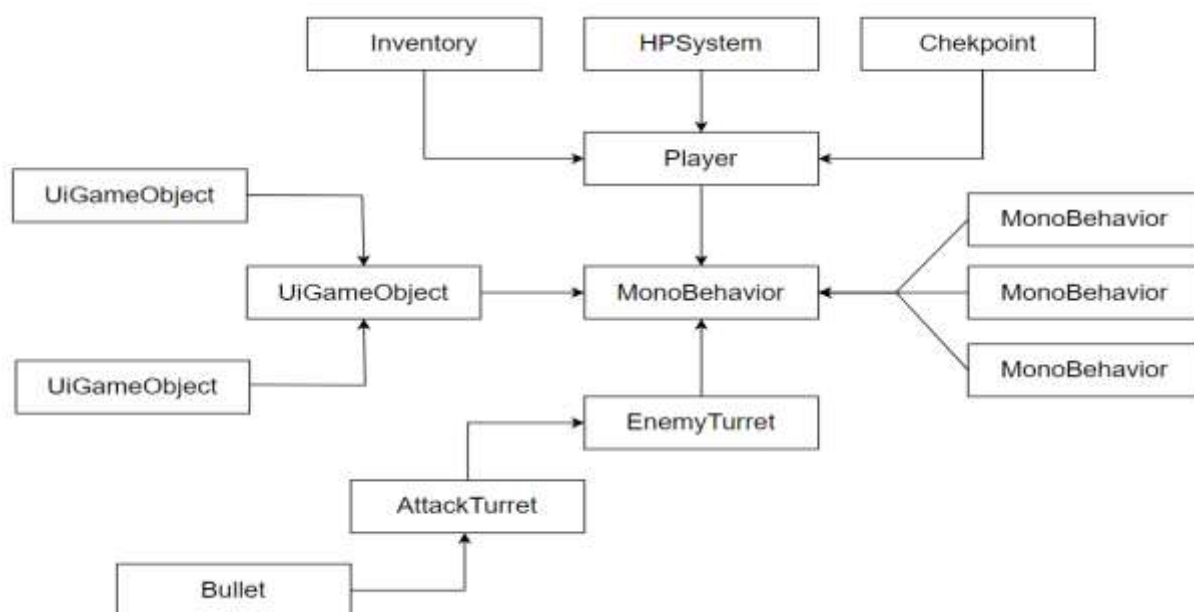


Рисунок 3.2 – Диаграмма Классов

Разработанная программа содержит несколько больших модулей (классов). У игрового движка Unity есть одна особенность, проявляющаяся в его архитектуре — большинство классов представляют собой скрипты движка, решающие четко поставленные задачи. Классы наследуются от базового класса `MonoBehavior`, обеспечивающего взаимодействие между другими классами. Так как скрипты являются компонентами подключенные к игровым объектам, диаграмма классов может получиться очень разветвленной ведь на одном игровом объекте может находиться несколько скриптов что обеспечивает различные или связанный друг с другом функционал [3]. На рисунке 3.2 представлена диаграмма классов в соответствии с таблицей 3.2.

Таблица 3.2 – Описание классов

Название компонента	Описание компонента
<code>MonoBehavior</code>	Базовый класс Unity
<code>Player</code>	Класс, содержащий в себе передвижение и атаку игрока
<code>UiGameObject</code>	Класс, содержащий в себе массив объектов интерфейса

Название компонента	Описание компонента
UiMenuOptions	Класс, содержащий в себе логику интерфейса в меню
UiPanelOptions	Класс, содержащий в себе логику интерфейса в игровом процессе
ChekPoint	Класс, содержащий в себе сохранение прогресса игрока
HPSystem	Класс, содержащий в себе здоровье игрока
Inventory	Класс, содержащий в себе инвентарь игрока
EnemyTurret	Класс, содержащий в себе «врага-турель»
AttackTurret	Класс, содержащий в себе атаку «турели»
Bullet	Класс, содержащий в себе снаряды для «турели»
MiliEnemy	Класс, содержащий в себе патрулирующего врага
Door	Класс, содержащий в себе открытие закрытых участков уровня
MerchantShop	Класс, содержащий в себе магазин

В ходе программной реализации разработана система передвижения и атаки игрока, поведение врагов, логика системы здоровья, а также удобный для пользователя интерфейс.

Система сохранения позволяет собирать данные с множества объектов, модулей и компонентов, что позволяет создавать контрольные точки для игрока. Это необходимо для того, чтобы игроку не приходилось перепроходить игру несколько раз, а начинать с точки сюжета на которой остановился [3].

Разработанный удобный и функциональный интерфейс поможет пользователю управлять игрой и выполнять необходимые действия. По мере развития сюжета интерфейс будет динамически обновляться [2].

4 Тестирование

В ходе разработки программного средства для того чтобы конечный пользователь получил завершённую отлаженную программу были проведены многочисленные тесты, в ходе которых были выявлены и исправлены возникающие ошибки.

Первый тест проверял наличие багов в передвижении и анимации главного героя. Все баги связанные с передвижением и анимации были исправлены.

Второй тест проверял логику работы главного меню и логику магазина. Они должны выполнять задуманные функции и работать без появления ошибок. Удаление и отображение объекта со сцены должны выполняться без ошибок, которые могли бы привести к потере производительности. После чего главное меню и магазин выполняли задуманные функции.

5 Техническое задание

5.1.1 Введение

Программный продукт является технологической демонстрационной версией компьютерной игры в режиме 2D платформера в жанре приключения. Проект реализован при помощи игрового движка Unity Engine. Игроку предстоит проходить уровни с проработанной сложностью и находить игровые предметы.

5.1.2 Назначение разработки

Программный продукт предназначен для развлечения игрока. Он помогает получить эмоции от прохождения и отвлечься от проблем. Назначение игры в предоставлении возможности насладиться игровым процессом.

5.1.3 Требования к функциональным характеристикам

Программное обеспечение представляет собой игру в режиме 2D платформера с пиксельной графикой.

Программное обеспечение(игра) предоставляет следующие функции:

1) в главном меню:

- возможность запустить новую игру;
- загрузить ранее сохраненную игру;
- возможность изменить настройки игры (графика, громкость музыки и вкл/выкл музыку);
- возможность прочитать об авторе;
- возможность выхода.

2) игровой процесс:

- возможность подбора предметов;
- возможность открытия участков уровня;

- возможность нанесения и получения урона;
- возможность покупки в магазине;
- возможность начать диалог с торговцем;
- возможность взаимодействия с окружением.

5.1.4 Требования к входным и выходным данным

Входные и выходные данные описаны в разделе «Анализ данных».

5.1.5 Требования к надежности

В программе должна присутствовать проверка входной информации на соответствие типов, принадлежность диапазону допустимых значений и соответствие структурной корректности. В случае возникновения ошибок предусмотреть возможность вывода информативных диагностических сообщений. В проекте должны легироваться данные о состоянии игрока в файле для меньшей вероятности её изменение пользователем.

5.1.6 Требования к составу и параметрам технических средств

Минимальные системные требования:

- ОС: Windows 7 (64-bit);
- процессор: Intel Core 2 Duo E5200;
- оперативная память: 4 GB ОЗУ;
- видеокарта: GeForce 9800GTX+ (1GB);
- DirectX: Версии 9;
- место на диске: 250 MB;
- дополнительно: 1080p, 16:9 recommended.

Оптимальные системные требования:

- ОС: Windows 10 (64-bit);
- процессор: Intel Core i5;
- оперативная память: 8 GB ОЗУ;

- видеокарта: GeForce GTX 560;
- DirectX: Версии 10;
- место на диске: 250 MB;
- дополнительно: 1080p, 16:9 recommended.

5.1.7 Требования к информационной и программной совместимости

Программа должна функционировать на ОС семейства Windows 7 и выше, с установленным DirectX 9 и выше, а также с установленным Net. Framework 4.5 и выше.

5.1.8 Требования к программной документации

Программная документация должна включать в себя руководство пользователя.

6 Руководство пользователя

При входе в игру пользователь попадает в главное меню игры. В главном меню у пользователя есть возможность запустить новую игру, продолжить игру, открыть настройки, прочитать о создателе игры, кто над ней работал и выйти из игры. Главное меню игры представлено на рисунке 6.1.

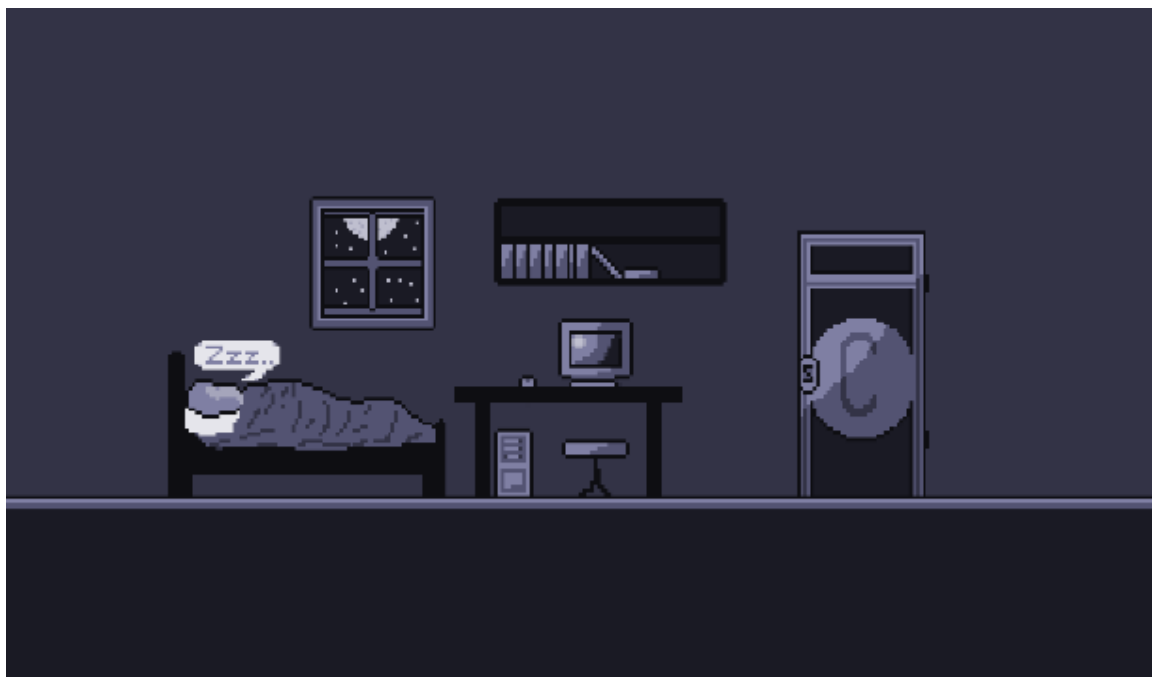


Рисунок 6.1 – Главное меню

В главном меню при наведении курсора на какой-либо объект, тот будет подсвечиваться и появится надпись за что этот объект отвечает.

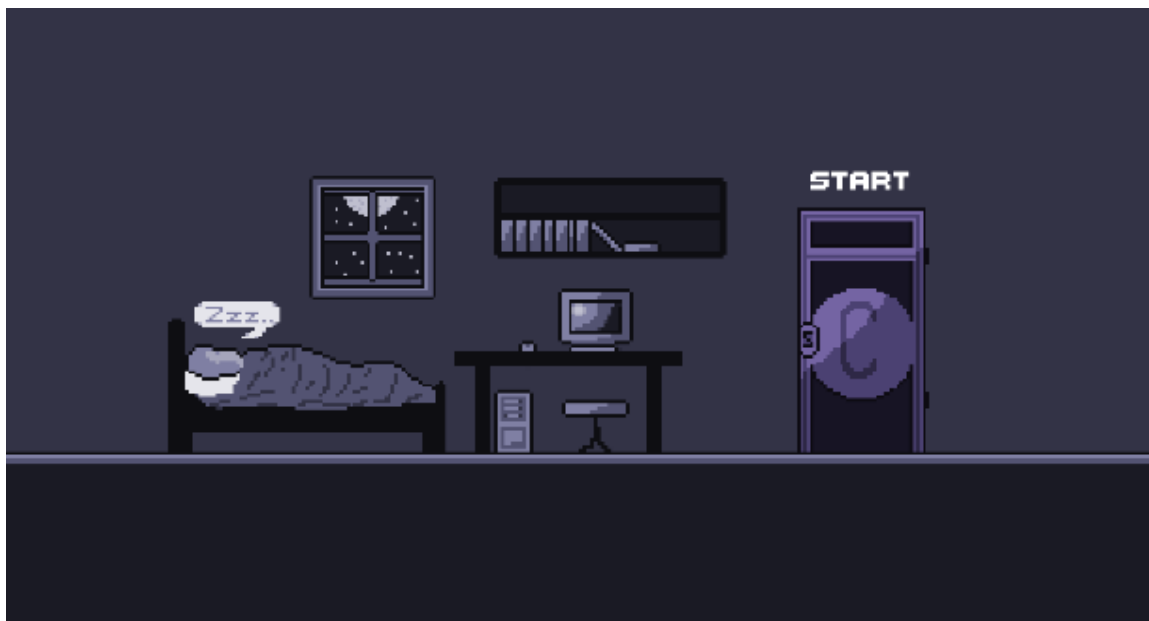


Рисунок 6.2 – Кнопка ‘START’

При нажатии на кнопку «START» игроку предоставляется выбор начать новую игру или же продолжить предыдущую (рис. 6.2).

При нажатии на кнопку «NEW GAME» игрок получит повествование сюжетной линии. Также появится возможность пропустить повествование. После повествования начнется загрузка уровня по завершении которой понадобится нажать любую кнопку для продолжения.

При переходе на первый уровень (рис. 6.3) игрок сможет передвигаться, атаковать, подбирать предметы и т.д.



Рисунок 6.3 – Первый уровень

ЗАКЛЮЧЕНИЕ

В ходе проделанной работы была создана игра в режиме 2D платформера в жанре приключения. Создан и внедрен сюжет игры, добавлены покадрово нарисованные анимации, откорректирована сложность игры. Добавлена система повествования сюжета, система сохранения прогресса. Адаптивный, динамический и удобный пользовательский интерфейс.

Разработанный программный инструмент выполняет предусмотренные и присущие ему функции. Продукт развлекает игроков, вызывая у них эмоции от прохождения игры. Магазин мотивирует игроков открывать закрытые части уровня, получая больше удовольствия от игры.

В дальнейшем систему можно модифицировать, интегрировать и обслуживать. В перспективе в игровой продукт можно добавить возможность играть на телефоне, достижения, запоминающихся боссов, большое количество настроек и т.д.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Официальная документация Unity [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/ru/530/Manual/> (дата обращения 21.01.2022).
2. Habr интернет ресурс программистов [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/otus/blog/532284/> (дата обращения 21.01.2022).
3. Тюкачев, Н. А. С#. Алгоритмы и структуры данных: учебное пособие / Н. А. Тюкачев, В. Г. Хлебостроев. – 3-е изд., стер. – Санкт-Петербург: Лань, 2018. – 232 с. – ISBN 978-5-8114-2566-2. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/104961> (Дата обращения: 15.02.2022) – Режим доступа: для авториз. пользователей.