

Муниципальное бюджетное общеобразовательное учреждение средняя
общеобразовательная школа № 39 станицы Юго-Северной муниципального
образования Тихорецкий район имени Героя Советского Союза Анатолия
Васильевича Ляпидевского

Международный конкурс «Research start» 23/24

Проектная работа

Создание игры на платформе Unity

"Мусорный подход: приключение по управлению отходами"

Автор проекта:
Сидоренко Никита, ученик 10 класса
Руководитель проекта:
Сидоренко Ирина Владимировна,
учитель изобразительного искусства

ст.Юго-Северная
2024

СОДЕРЖАНИЕ

Введение	3
Глава 1. История компьютерных игр	5
1.1 История развития компьютерных игр в мире	5
1.2 История развития компьютерных игр в России	6
Глава 2. Платформа для разработки компьютерных игр Unity	7
2.1 Разработка игры на Unity	7
Глава 3. Практическая часть	10
Заключение	12
Список источников	13
Приложение (файл с видео игрой)	

ВВЕДЕНИЕ

*«Компьютерные игры должны помогать человеку развиваться,
воспитывать в людях общечеловеческие ценности»*

В.В.Путин

Что такое игра? Ответить на данный вопрос стремились многие философы, культурологи, психологи. В XX–XXI в. появились новые формы игры, в том числе видеоигры, которые оказали значительное влияние на поле современной культуры, массовое сознание и социальные практики. Мировой рынок видеоигр растет стремительно, а сами видеоигры выходят далеко за пределы индустрии развлечений, они проникают в образование и науку, искусство, менеджмент, маркетинг, медицину, политику, спорт и т.д.

В современном мире компьютерные игры становятся все более популярным и важным элементом образовательного процесса. Игры не только увлекают детей и подростков, но и способствуют развитию различных навыков, таких как логическое мышление, решение проблем, коммуникация и сотрудничество.

В рамках данного проекта я предлагаю создать компьютерную игру для учащихся школы, которая поможет им развить волонтерские навыки, а также повысить мотивацию к сохранению экологической безопасности.

Цель: показать, что геймеры могут не только играть, но и создавать игры с 3D-графикой

Гипотеза: с помощью игровой среды Unity и языка программирования C#, я смогу сделать свою игру.

Проблема: современные дети увлечены различными компьютерными играми, и не задумываются над тем, где они могут применить свой накопленный опыт.

Объект исследования: Игровой движок Unity и язык программирования C#.

Предмет исследования: Возможности технологий программирования.

Методы исследования:

1. Изучение литературы
2. Обработка и интерпретация данных.
3. Программирование

Задачи:

1. Узнать историю возникновения компьютерных игр;
2. Изучить общие сведения о платформе “Unity” и языке программирования C#;
3. Изучить выбранную программу и создать в ней игру;
4. Протестировать игру среди своих друзей и одноклассников;
5. Сделать вывод о подтверждении или опровержении выдвинутой гипотезы.

Глава 1. История компьютерных игр

1.1 История развития компьютерных игр в мире

История развития компьютерных игр насчитывает несколько десятилетий и включает в себя ряд значительных этапов и достижений. Зарождение компьютерных игр началось в 1950-х годах, когда ученые и инженеры начали создавать простые игры для развлечения и демонстрации возможностей компьютеров. Одной из первых игр была "OXO" - компьютерная версия крестиков-ноликов. В 1960-е годы появились первые текстовые приключенческие игры, такие как "Adventure" (или "Colossal Cave Adventure"), которые стали популярными среди пользователей компьютеров. В 1970-е годы компьютерные игры начали развиваться быстрыми темпами. Были созданы первые аркадные игры, такие как "Pong" и "Space Invaders", которые стали массовыми хитами. В 1980-е годы компьютерные игры стали все более популярными благодаря развитию персональных компьютеров и игровых консолей. Были созданы классические игры, такие как "Super Mario Bros", "Tetris" и "The Legend of Zelda". В 1990-е годы произошел бум компьютерных игр с появлением трехмерной графики и новых жанров, таких как шутеры от первого лица, стратегии в реальном времени и MMORPG. В 2000-е годы компьютерные игры стали еще более популярными и разнообразными благодаря развитию онлайн-игр, мобильных игр и виртуальной реальности. В настоящее время компьютерные игры являются одной из самых популярных форм развлечений, объединяющих миллионы игроков со всего мира. Игры становятся все более сложными, красочными и интерактивными, предлагая игрокам уникальные возможности для взаимодействия и экспериментирования.

История развития компьютерных игр свидетельствует о том, что они стали неотъемлемой частью современной культуры и развлечений, привлекая внимание людей всех возрастов и интересов.

1.2 История развития компьютерных игр в России

История развития компьютерных игр в России имеет свои особенности и включает в себя несколько ключевых этапов. В конце 1980-х годов в Советском Союзе начали появляться первые компьютерные игры, созданные любителями программирования. Одной из самых известных игр того времени была "Тетрис", разработанная Алексеем Пажитновым. В 1990-е годы российская индустрия компьютерных игр начала активно развиваться. Были созданы такие известные игры, как "Космические Рейнджеры", "Корсары" и "Демиурги". В 2000-е годы российские разработчики продолжили создавать качественные компьютерные игры, которые получили признание как на внутреннем, так и на мировом рынке. Некоторые известные проекты этого периода включают в себя "Синдром Дракона", "Космические Рейнджеры 2" и "Pathologic". В настоящее время российские компьютерные игры становятся все более популярными и признанными за рубежом. Разработчики из России создают игры различных жанров, от инди-проектов до крупных AAA-проектов.

Российские компьютерные игры отличаются своим уникальным стилем, атмосферой и сюжетом, привлекая внимание игроков со всего мира. Развитие индустрии компьютерных игр в России продолжается, и она остается важным сегментом мировой игровой индустрии.

Глава 2. Платформа для разработки компьютерных игр Unity

Существует несколько видов платформ для разработки компьютерных игр, каждая из которых имеет свои особенности и преимущества. Unity - одна из самых популярных и широко используемых платформ для разработки компьютерных игр. Unity поддерживает разработку игр для различных платформ, включая ПК, мобильные устройства, консоли и виртуальную реальность. Он предлагает мощный движок, инструменты для создания графики, аудио и физики, а также обширное сообщество разработчиков.

Особенности:

Unity поддерживает разработку игр для различных платформ, включая iOS, Android, Windows, macOS, Linux, PlayStation, Xbox и другие.

Unity имеет интуитивно понятный визуальный редактор, который позволяет создавать игровые уровни, персонажей и интерфейс без необходимости писать код.

Unity использует язык программирования C# для создания игровой логики и взаимодействий между объектами.

Unity Asset Store предоставляет широкий выбор готовых ресурсов, скриптов, моделей и текстур для ускорения процесса разработки.

Unity имеет активное сообщество разработчиков, которые делятся знаниями, опытом и ресурсами.

2.1 Разработка игры на Unity

Разработка игры на Unity - это увлекательный и творческий процесс, который требует знаний программирования, дизайна и игровой механики. Вот некоторые шаги, которые обычно включаются в процесс разработки игры на Unity:

1. Определить жанр игры, целевую аудиторию, игровую механику и основной сюжет;

2. Создать дизайн игровых уровней, персонажей, предметов и интерфейса;
3. Использовать язык программирования C# и инструменты Unity для создания игровых механик, взаимодействий и анимации;
4. Проверить игру на наличие ошибок, багов и оптимизировать производительность.
5. Подготовить игру для публикации на платформах, таких как Steam, App Store или Google Play.

2.2. Язык программирования C#

C#(произносится как "си шарп") — современный объектно-ориентированный язык программирования. C# позволяет разработчикам создавать разные типы безопасных и надежных приложений, выполняющихся в NET. C# относится к широко известному семейству языков C, и покажется хорошо знакомым любому, кто работал с C, C++, Java или JavaScript. Здесь представлен обзор основных компонентов языка C# 8 и более ранних версий. Если вы хотите изучить язык с помощью интерактивных примеров, рекомендуем поработать с вводными руководствами по C#.

Вот лишь несколько функций языка C#, которые позволяют создавать надежные и устойчивые приложения. *Сборка мусора _ автоматически освобождает память, занятую недоступными неиспользуемыми объектами. Типы, допускающие значение null, обеспечивают защиту от переменных, которые не ссылаются на выделенные объекты. Обработка исключений предоставляет структурированный и расширяемый подход к обнаружению ошибок и восстановлению после них. Лямбда-выражения поддерживают приемы функционального программирования. Синтаксис LINQ создает общий шаблон для работы с данными из любого источника. Поддержка языков для асинхронных операций предоставляет синтаксис для создания распределенных систем. В C# действует _ *единая система типов**_. Все типы C#, включая типы-примитивы, такие как int и double, наследуют от одного корневого типа object.

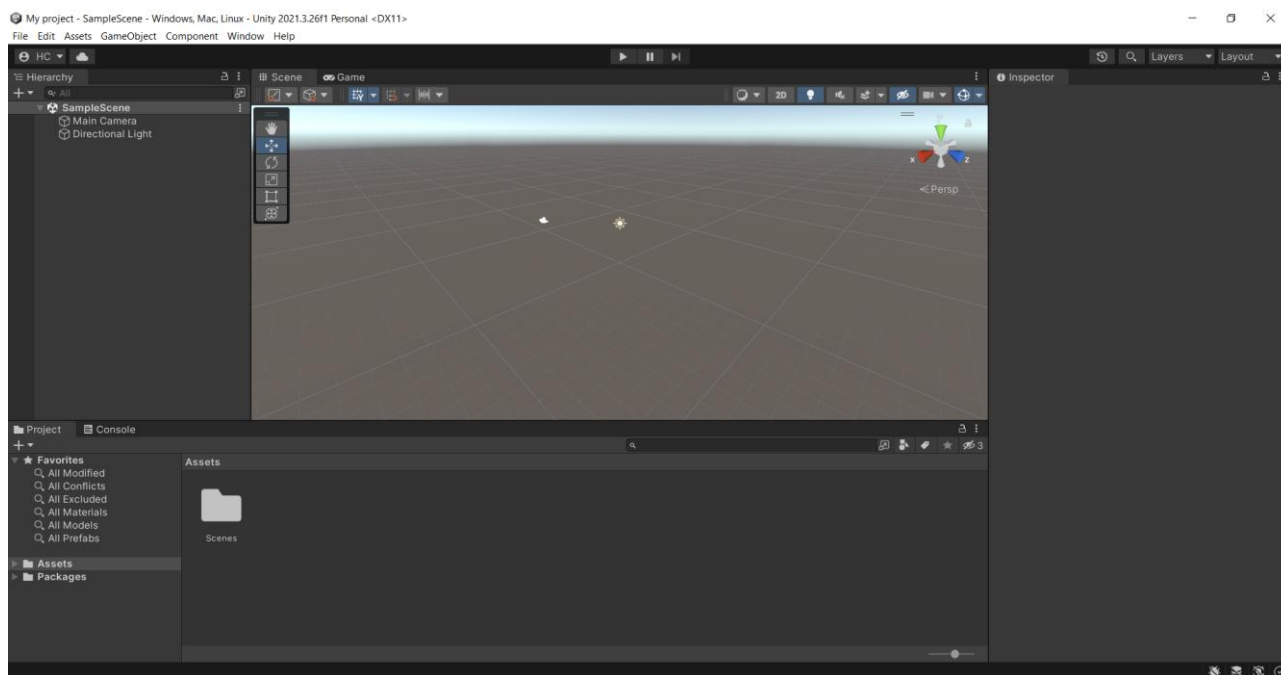
Все типы используют общий набор операций, а значения любого типа можно хранить, передавать и обрабатывать схожим образом. Более того, С# поддерживает как определяемые пользователями ссылочные типы, так и типы значений. С# позволяет динамически выделять объекты и хранить упрощенные структуры в стеке. С# поддерживает универсальные методы и типы, обеспечивающие повышенную безопасность типов и производительность. С# предоставляет итераторы, которые позволяют разработчикам классов коллекций определять пользовательские варианты поведения для клиентского кода.

В С# особое внимание уделяется управлению версиями для обеспечения совместимости программ и библиотек при их изменении. Вопросы управления версиями существенно повлияли на такие аспекты разработки С#, как отдельные модификаторы `virtual` и `override`, правила разрешения перегрузки методов и поддержка явного объявления членов интерфейса.

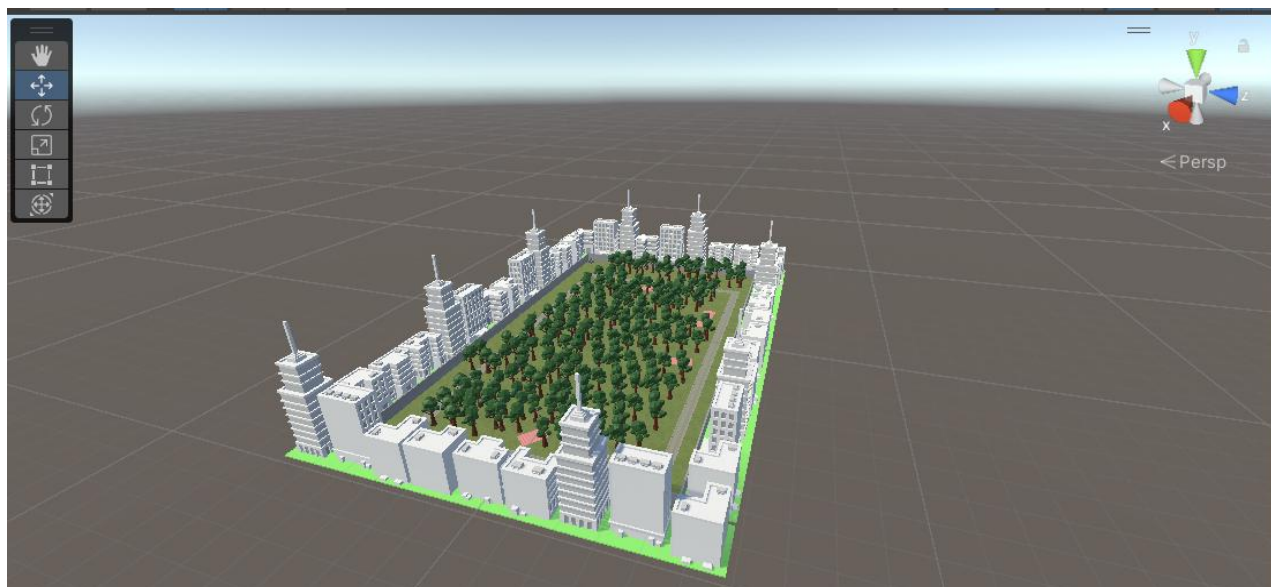
Глава 3. Практическая часть

Начал я с поиска объектов на Itch.io и анимации на Mixamo.

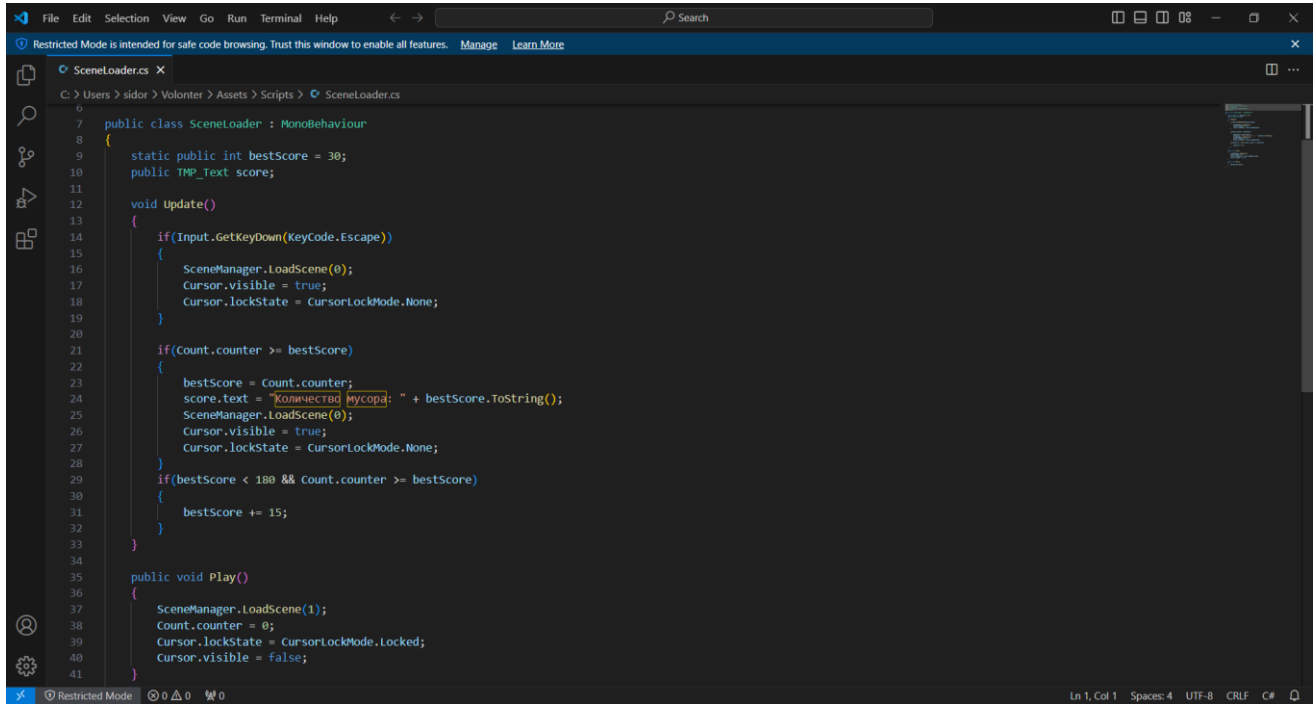
Потом создал пустой проект.



Затем заполнил сцену объектами, создал персонаж



Написал скрипты управления, сбора мусора и т.д



```
7 public class SceneLoader : MonoBehaviour
8 {
9     static public int bestScore = 30;
10    public TMP_Text score;
11
12    void Update()
13    {
14        if(Input.GetKeyDown(KeyCode.Escape))
15        {
16            SceneManager.LoadScene(0);
17            Cursor.visible = true;
18            Cursor.lockState = CursorLockMode.None;
19        }
20
21        if(Count.counter >= bestScore)
22        {
23            bestScore = Count.counter;
24            score.text = "Количество мусора: " + bestScore.ToString();
25            SceneManager.LoadScene(0);
26            Cursor.visible = true;
27            Cursor.lockState = CursorLockMode.None;
28        }
29        if(bestScore < 180 && Count.counter >= bestScore)
30        {
31            bestScore += 15;
32        }
33    }
34
35    public void Play()
36    {
37        SceneManager.LoadScene(1);
38        Count.counter = 0;
39        Cursor.lockState = CursorLockMode.Locked;
40        Cursor.visible = false;
41    }
}
```

В конце отдал игру на тестирование и исправил обнаруженные ошибки.

Заключение

Создание игры на Unity - увлекательный и трудоемкий процесс, который требует от разработчиков творческого подхода, профессионализма и стремления к совершенству. Успешное завершение проекта по созданию игры на Unity может стать отличным достижением и позволит насладиться результатом своего труда.

В результате работы над проектом мной были детально изучены:

теоретические основы языка программирования C#;

основы создания видеоигрового мира;

способы оптимизации видеоигры;

В ходе работы я подтвердил гипотезу и пришел к следующим выводам:

1. Чем сложнее и качественнее видеоигра, тем больше времени нужно потратить на ее разработку;

2. Оптимизация видеоигры – неотъемлемая часть её разработки;

3. Легче и быстрее разработать игру в игровой среде, чем с помощью одного языка программирования.

Полученные знания пригодятся мне при выборе будущего направления обучения и специализации профессии: программист, игровой дизайнер, художник по окружению или бета-тестер. Я планирую продолжить работу над своим проектом.

Список источников

Интернет источники:

<https://www.kp.ru/online/news/5368609/>

<https://itch.io/game-assets>

<https://www.mixamo.com/#/>