**Государственное бюджетное общеобразовательное учреждение**

**города Москвы "Школа № 1537 "Информационные технологии"**

**Обучающая видеоигра**

**для освоения основ программирования**

Выполнили:

Лаврецкий Константин Андреевич,

Пономарёв Савва Ильич,

ученики 11 класса

ГБОУ Школа № 1537

Научный руководитель:

Минченко Михаил Михайлович,

к.э.н., учитель информатики

ГБОУ Школа № 1537

Москва 2023

**ОГЛАВЛЕНИЕ**

1. **ВВЕДЕНИЕ**
2. **ЦЕЛЬ И ЗАДАЧИ РАБОТЫ**
3. **МЕТОДИКА ВЫПОЛНЕНИЯ РАБОТЫ**
4. **ВЫВОДЫ**
5. **СПИСОК ЛИТЕРАТУРЫ**

**Введение**

В век прорыва IT технологий будет очень актуальна для начала изучения самых востребованных языков программирования с нуля в простой и понятной подаче.

Игры являются одним из самых популярных видов развлечений, которые привлекают миллионы людей по всему миру. В то же время, изучение языка программирования может быть довольно сложным и утомительным процессом. Что если бы можно было объединить эти две вещи? Разработка игры с возможностью изучения языка - это отличный способ сделать процесс обучения более интересным и увлекательным. Игра может стать не только средством для развлечения, но и для обучения, позволяя игрокам изучать основы языка программирования через интерактивный геймплей. Такой проект может привлечь как начинающих программистов, так и опытных игроков, желающих улучшить свои навыки программирования.

**Цели и задачи**

**Цель**: создать игру, где каждое действие пользователь будет прописывать кодом, тем самым привыкая к определенному синтаксису и вырабатывая привычку

**Задачи:**

1. Нарисовать графику
2. Прописать основные скрипты движения
3. Прописать UI
4. Написать язык программирования для пользователя
5. Прописать дополнительные функции, которые придадут проекту интересность
6. Создать билд на интересующую нас систему

**МЕТОДИКА ВЫПОЛНЕНИЯ РАБОТЫ**

Игра разработана на игровом движке Unity с использованием языка C#. Движок позволяет создать полноценную игру вместе с продуманным интерфейсом. В нем используются стандартные элементы, встроенные в Unity, такие как: Image, Button, Canvas, Text.

В скриптах, написанных на C#, используются библиотеки. Например: UnityEvents.System, UnityEngine.UI, Input System, CinemaMachine

Также используются встроенные функции Unity.

Функция void Update делает игру бесконечной, то есть сама она не закроется (в данной реализации программа закрывается кнопкой "Alt+F4")

Расскажем о некоторых из них:

**CinemaMachine**

CinemaMachine Unity - это мощный инструмент для создания кинематографических сцен в играх. Он позволяет создавать сложные камерные движения, управлять их параметрами и создавать гладкие переходы между различными камерами. Вот некоторые из возможностей CinemaMachine Unity:

1. Создание сложных камерных движений. CinemaMachine Unity позволяет создавать сложные движения камеры, такие как панорамирование, повороты и зумирование. Вы можете настроить параметры движения и получить максимально реалистичный эффект.

2. Гладкие переходы между камерами. С помощью CinemaMachine Unity вы можете создавать плавные переходы между различными камерами, чтобы сохранить непрерывность сцены.

3. Управление параметрами камеры. Вы можете настроить параметры камеры, такие как фокусное расстояние, высота и угол обзора, чтобы получить наилучший результат.

4. Создание комплексных сценариев. CinemaMachine Unity позволяет создавать сложные сценарии, в которых камера будет перемещаться по заранее заданным точкам. Это может быть полезно для создания кинематографических сцен, таких как битвы или погони.

5. Интеграция с другими инструментами Unity. CinemaMachine Unity легко интегрируется с другими инструментами Unity, такими как Timeline и Animator, что позволяет создавать еще более сложные кинематографические сцены.

**Пользовательский интерфейс**

Пользовательский интерфейс игры реализуется в составе следующих разделов:

* Окно выбора уровня - это основной экран игры, куда пользователь будет возвращаться каждый раз, после прохождения уровня
  1. “Таблица уровней” - это таблица, при нажатие на отдельный элемент которой, открывается соответствующий уровень
  2. Окно помощи - при нажатии на клавишу “ESC” появляется окно, в котором можно посмотреть управление и команды.
* Окно самого уровня - это окно, непосредственно в котором проходит геймплей. Оно разделяется на две части:
  1. “Визуальное отображение” - там отображаются все действия пользователя, там же расположен персонаж, который находится под управлением игрока
  2. Окно редактора - там пользователь может писать код, тем самым передвигать своего персонажа, и проходить уровень

Стоит отметить, как организовано движение и редактор. Все происходит за счет двух скриптов. Скрипт Compiler отвечает за распознание текста и перевода его код, а HeroKnight за движение.

По нажатию на кнопку “RUN” данные отправляются в скрипт Compiler, переводятся в string и записываются в переменную code. Мы режем code функцией Split и записываем это в массив codeArray.

Далее в цикле for мы проверяем соответствие значений массива базовым командам и обрабатываем количество команд с помощью функции StrToInt, которая строит число посредством его умножения на 10

И затем запускается функция движения из скрипта HeroKnight соответствующая команде.

Она принимает Transform персонажа и изменяет его в соответствии с вектором направления.

Также важным в проекте является менеджмент уровней. Он состоит из нескольких скриптов: LevelWindow, GameSession, LevelItem. Все загрузки уровней, меню реализуются через него.

В самом начале, при заходе в меню с выбором уровня LevelWindow получает количество пройденных уровней из GameSession и делает возможность их выбрать и пройти.

Сам скрипт GameSession перемещается вместе с игроком из уровня в уровень, храня и перезаписывая различные изменения, которые происходят.

Так при нажатии на ячейку уровня в менеджере активируется метод отвечающий за изменение визуала ячейки, показывая, что она выбрана. Появляется окно позволяющее начать или перевыбрать уровень. При запуске уровня LevelWindow обращается к LevelItem, получая из него уровень, который надо запустить, и запускает его.

При отказе, окно подтверждения скроется позволяя перевыбрать уровень вновь.

**ВЫВОДЫ**

Как и планировалось, создана игра, с простым функционалом. Файлы игры содержатся на GitHub, для того чтобы начать играть - нужно перейти в папку Buildes и нажать на файл c расширением .exe

Игра уже может использоваться, необходимый функционал доступен, но еще не достаточно велик. Необходимо добавить новые механики и команды, тем самым расширить потенциал и увеличить интерес к игре.

Приложение достаточно простое в использовании, имеет понятный интерфейс. Рекомендуется в пользовании всем, может помочь начать изучать языки программирования.

В перспективе имеет смысл добавить в игру функцию определения ошибки, которая будет определять и показывать где неправильно написан код. Также стоит добавить собираемые предметы и врагов и соответствующие команды для них

**СПИСОК ЛИТЕРАТУРЫ**

1. [**Руководство Unity**](https://docs.unity3d.com/ru/530/Manual/)
2. [**Unity Forum**](https://forum.unity.com/threads/convert-string-to-int-in-c.58867/)
3. [**Документация по GitHub**](https://docs.github.com/ru)
4. [**Stack Overflow**](https://stackoverflow.com/)

**ССЫЛКИ**

Подробную информацию можете посмотреть по этой ссылке: [GitHub](https://github.com/Kapitan999/TeamProject.git)