

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ТЕМА: ТРЕНАЖЕР ДЛЯ КОРРЕКЦИИ ЗРЕНИЯ В ИГРОВОЙ ФОРМЕ

Научный Руководитель:
кан. тех. наук, доцент
Е.С. Ярош
Автор:
Студент группы КЭ-405
И.А. Плотников

Актуальность темы

Коррекция зрения с применением программных продуктов является весьма актуальной задачей, так как на сегодня это один из самых доступных, дешевых и удобных способов тренировки зрительной системы.

Проблемы аналогов:

- трудности фокусировки внимания детей на тренажере;
- отсутствие поддержки современных операционных систем
- устаревший интерфейс;
- распространение на платной основе.

Цель работы

1. Провести анализ существующих программных продуктов для поддержания и коррекции зрения.
2. На основе рекомендаций врача-офтальмолога и результатов анализа п. 1 сформулировать подробные требования к разрабатываемому программному обеспечению.
3. Разработать архитектуру программного продукта, осуществить обоснованный выбор средств его реализации.
4. Выполнить программную реализацию в соответствии с предъявляемыми требованиями.
5. Выполнить тестирование разработанного программного обеспечения.

Анализ и сравнение аналогов

	Фокусировка внимания	Отслеживание результатов	Поддержка современных ОС	Проприетарность
Чибис	-	-	-	-
ЦВЕТОК	-	+	-	+
eYe	-	-	-	-
Контур	-	-	+	-
Крестики	-	+	-	+
Паучок	-	+	-	+

Выбор среды разработки

Для достижения целей данной выпускной квалификационной работы необходимо выбрать среду разработки. При коррекции зрения используются графические изображения. Также требуется удерживать внимание пользователя программного продукта таким образом, чтобы соблюдалась методика тренировки. Наиболее подходящим для этого является игровой подход. Поэтому рассматривались игровые движки и сопутствующее им системное программное обеспечение.

Рассматривались следующие варианты:

- Unity 2019;
- CryEngine 3;
- Unreal Engine 4.

Функциональные требования:

- методика лечения должна быть основана на тренировках;
- в качестве методической основы взять прибор Амблиотренер;
- основной метод тренировки – прохождение лабиринта;
- конфигурация лабиринта должна меняться при неоднократном прохождении одного и того же уровня;
- общее количество уровней – не менее 5;
- при успешном прохождении уровня должны быть предусмотрены призы;
- возможность при загрузке приложения выбрать уровень, настроить сложность в соответствии со степенью амблиопии, узнать количество накопленных на данный момент игровых баллов;
- по запросу показ статистики за предыдущие сеансы тренировки и вывод её в отдельный файл;
- возможность в процессе работы настроить уровень сложности в соответствии со степенью амблиопии;
- возможность перемещаться между уровнями, проходить их, зарабатывать баллы, разблокировать новые уровни.

Нефункциональные требования:

- после загрузки программы возможность доступа в меню выбора уровня, настроек и статистики, возможность выйти из приложения и вернуться в главное меню;
- реализация в сцене главного меню кнопок «Выбор уровня», «Настройки», «Статистика», «Выход»;
- возможность просмотра статистики за предыдущие сеансы тренировки и вывод её в файл.;
- возможность изменения настройки разрешения, установки полноэкранного режима, изменения языка интерфейса. Языком по умолчанию является русский;
- адаптация к мониторам с различным соотношением сторон;
- патентная чистота в отношении территории Российской Федерации;
- регистрация в Едином реестре российских программ для электронных вычислительных машин и баз данных.

Концепция программы

Программа представляет собой аналог существующего аппаратного комплекса под названием Амблиотренер.

Амблиотренер – настольный прибор, представляющий собой корпус, на верхней наклонной плоскости которого расположена панель из органического стекла и пульт управления. В углубление панели помещают сменные решетки, платы и рисунки с опто типами, под панелью размещено семнадцать ламп. На боковой панели прибора расположены гнезда для подключения трансформатора, головного телефона и токопроводящего стержня.



Концепция программы

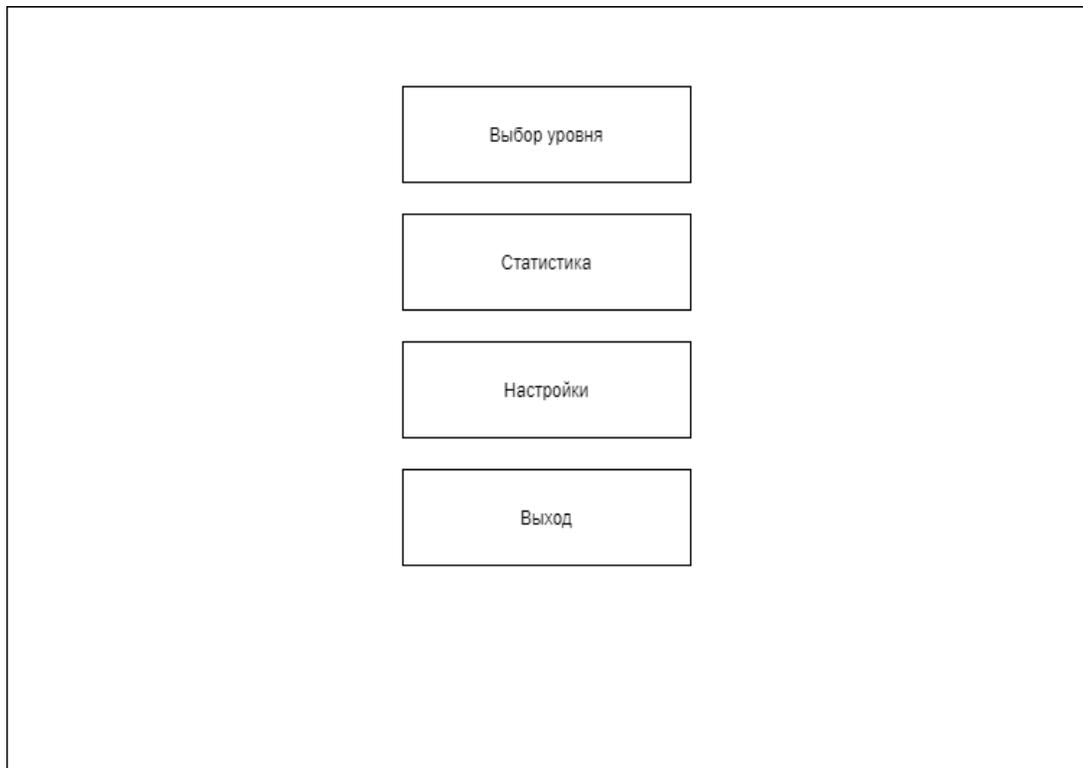
Программный продукт представляет собой набор определенных уровней, каждый из которых оформлен в той или иной тематике и имеет определенную траекторию прохождения.

Чтобы попасть на новый уровень, пациенту необходимо пройти предыдущий на максимально возможное количество баллов.

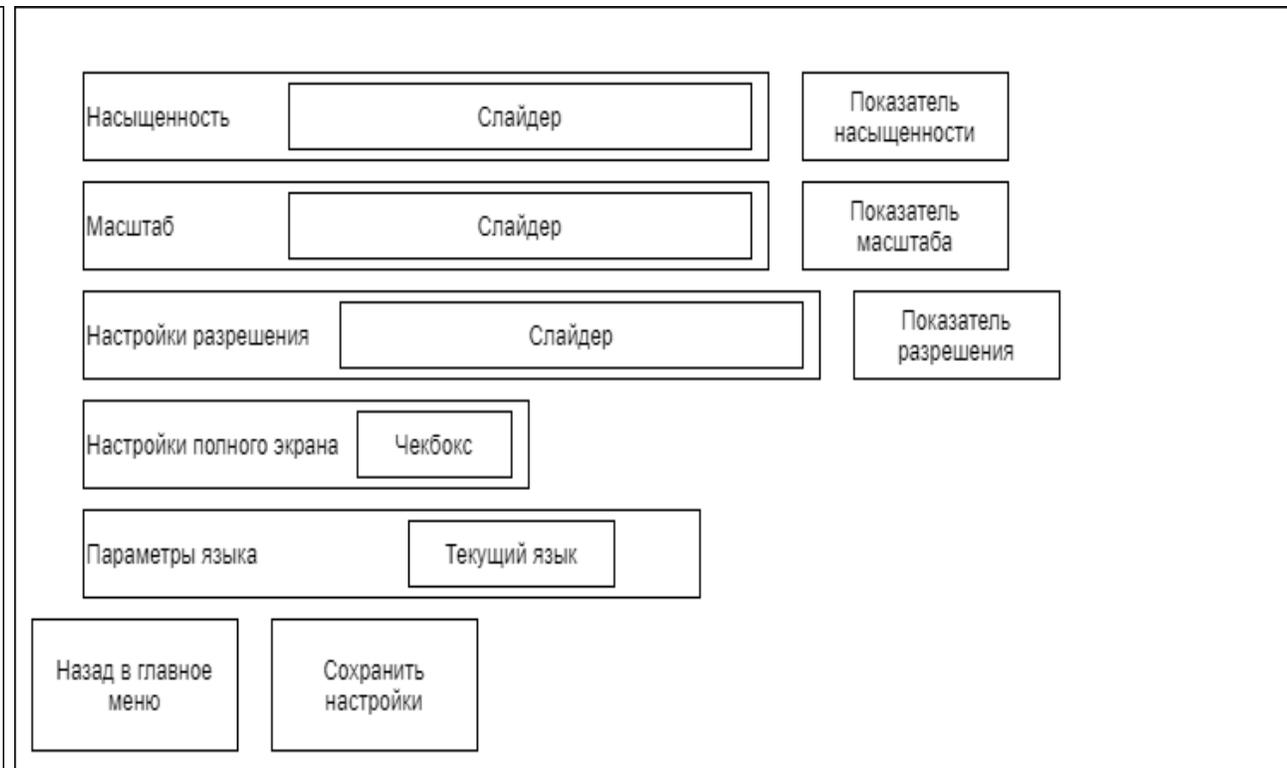
Чтобы обеспечить лечебный эффект для пациентов с разными степенями амблиопии, в разделе «Настройки» находится несколько параметров, изменение которых влечет за собой повышение или понижение сложности тренировки.

Концепция интерфейса

Макет главного меню

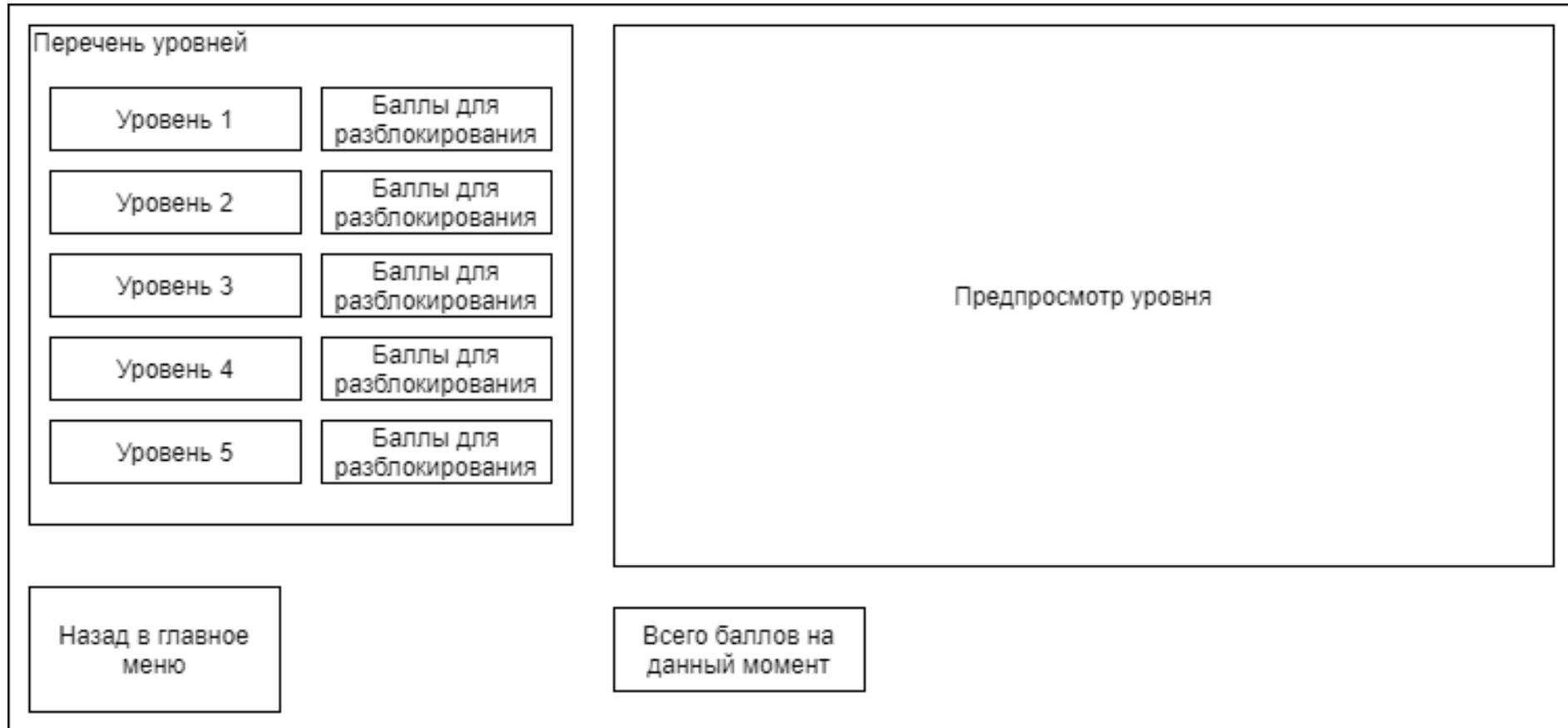


Макет меню настроек



Концепция интерфейса

Макет меню выбора уровня



Концепция интерфейса

Макет меню статистики



Проектирование

Диаграмма прецедентов тренировочного уровня

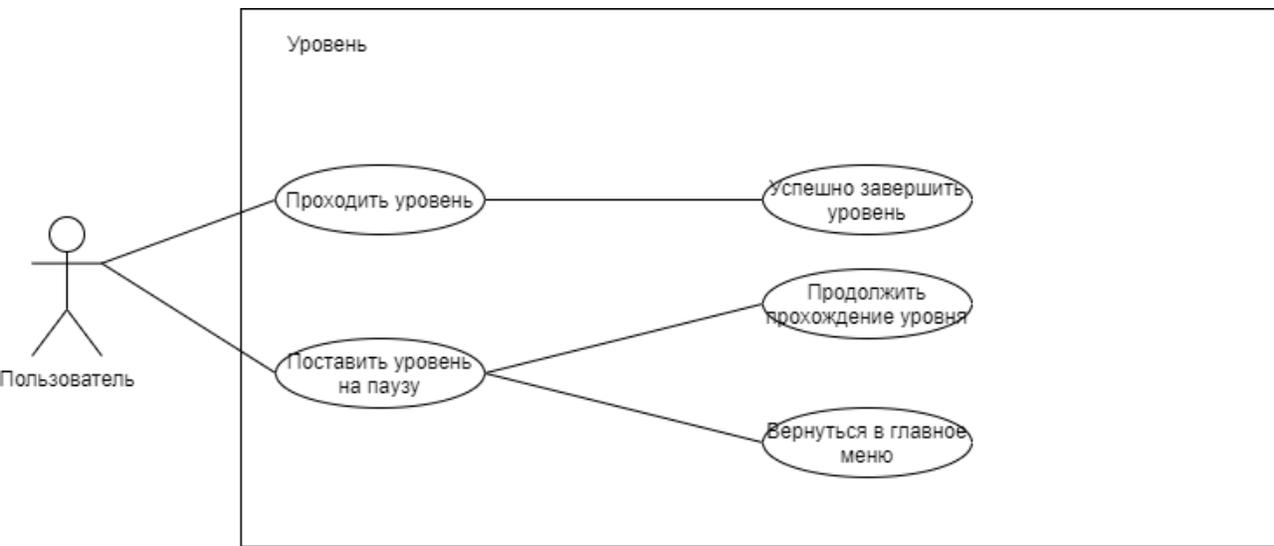
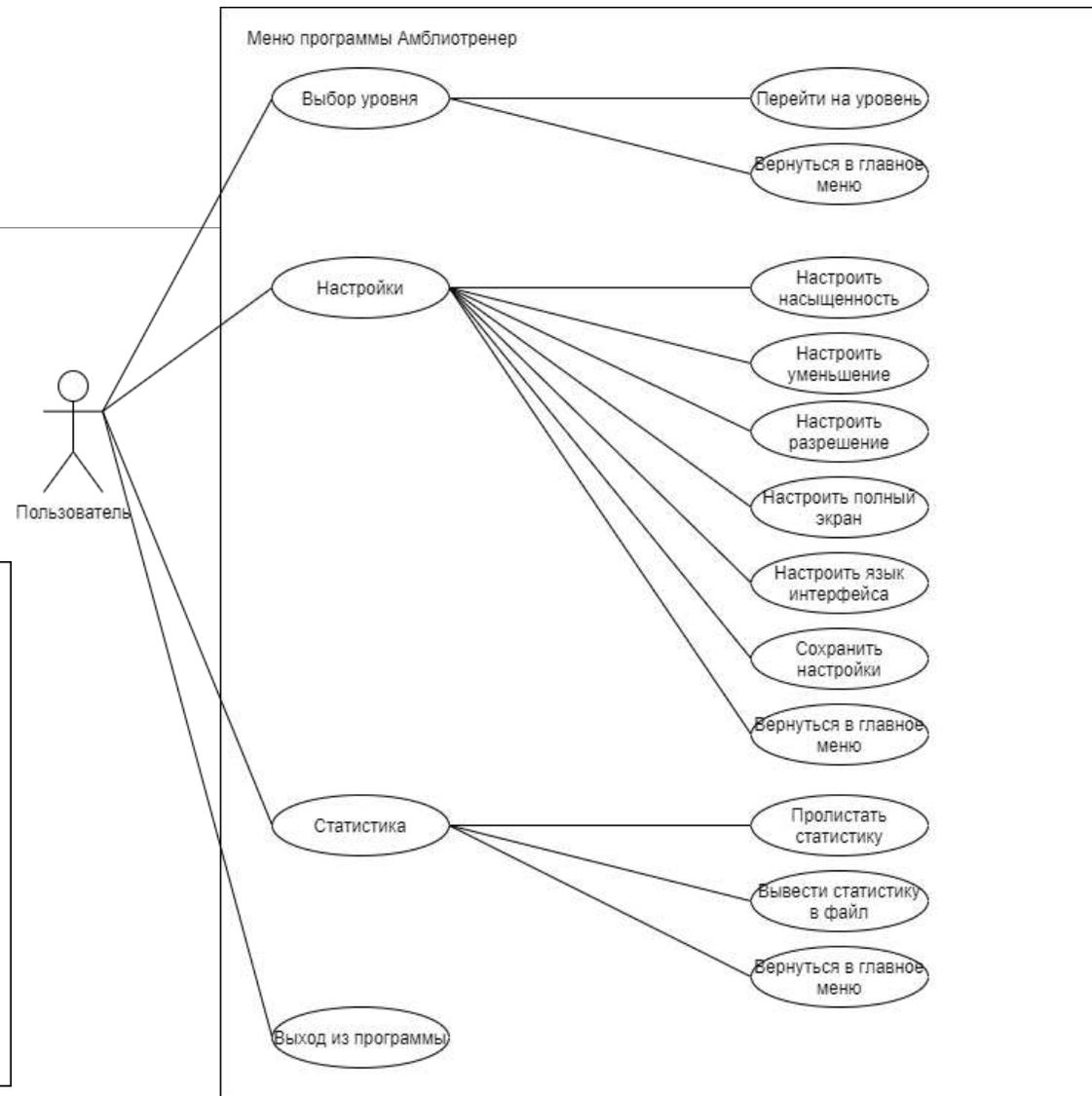
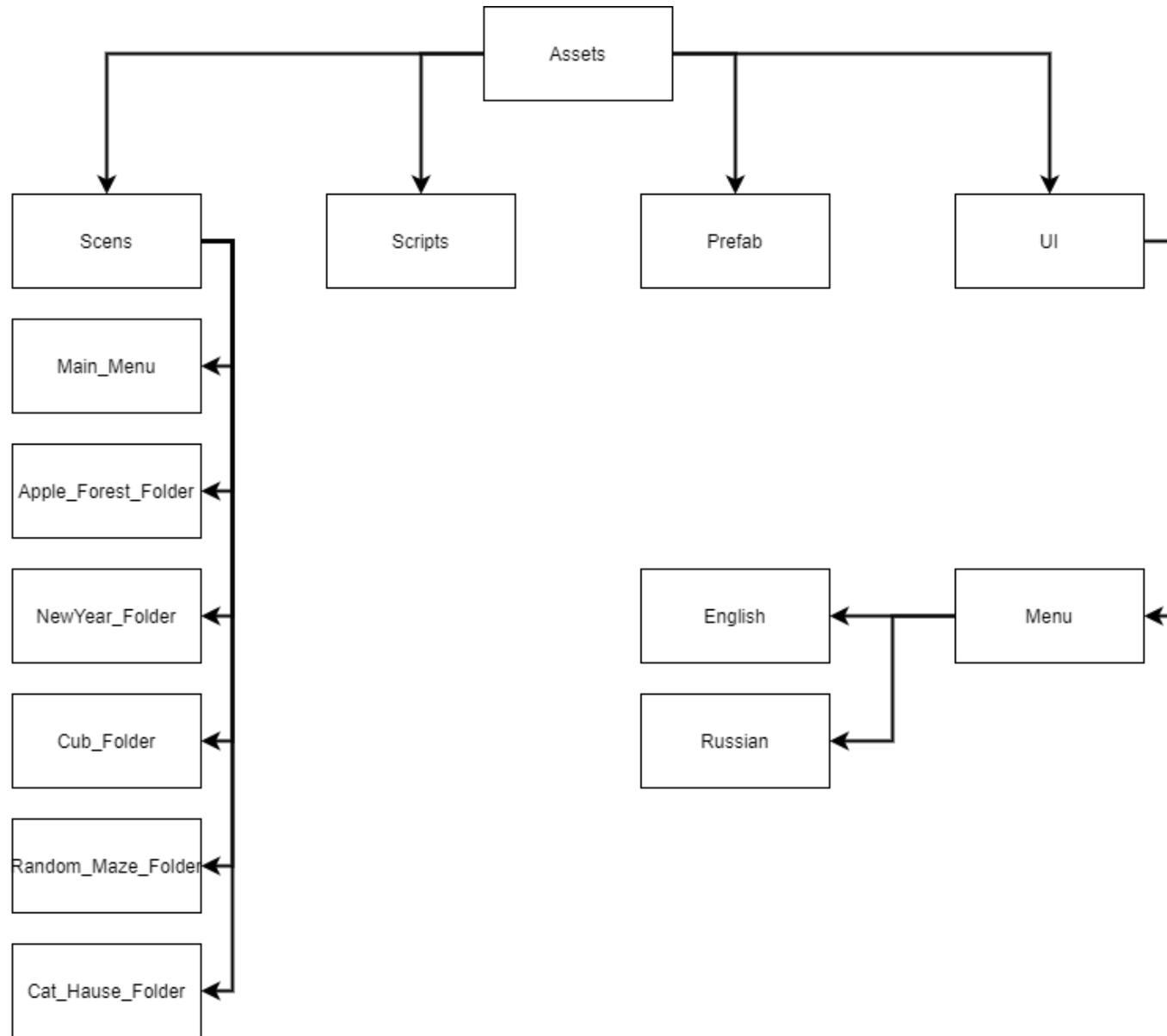


Диаграмма прецедентов меню программы



Файловая структура



Реализация

Для реализации приложения были использованы скрипты, написанные на языке программирования C#.

Всего в данной программе используется **24 скрипта**: AllCoins, CameraControls, Cell, ChangeLanguage, Coin, CoinCounter, CoinView, DataSaver, Finish, Gammcfg, HintRenderer, Loader, Maze, MazeGenerator, MazeSpawner, MenuScript, Options, Pause_Menu, PlayerComtrolls, Preview, PreviewImage, Scalecfg, LevelSelector, Statistics.

Всего в данной программе используется **28 классов**, каждый из которых размещен в отдельном, одноименном скрипте, за исключением DataSaver и GameData, они оба находятся в скрипте DataSaver, также Maze и MazeGeneratorCell, они оба находятся в скрипте Maze.

Всего данная программа заняла **874 строк** кода.

Название скрипта	Назначение скрипта
AllCoins	Подсчет суммарного количества баллов.
CameraControls	Перемещение камеры за игроком.
Cell	Шаблон стены в процедурно генерируемом лабиринте.
ChangeLanguage	Перевод программы на английский язык.
Coin	Начисление баллов при соприкосновении игрока с объектом.
CoinCounter	Создание собираемых объектов в процедурногенерируемом лабиринте.
CoinView	Отображение набранных на уровне баллов.
DataSaver	Сохранение и загрузка данных в XML-файл.
Finish	Завершение уровня по его прохождению и начисление баллов.
GammaCfg	Конфигурация насыщенности изображения на тренировочном уровне.
HintRenderer	Поиск пути в процедурногенерируемом лабиринте.

Название скрипта	Назначение скрипта
Loader	Хранение данных.
Maze	Один из составных скриптов, отвечающий за генерацию процедурногенерируемого лабиринта.
MazeGenerator	Один из составных скриптов, отвечающий за генерацию процедурногенерируемого лабиринта.
MazeSpawner	Один из составных скриптов, отвечающий за генерацию процедурногенерируемого лабиринта.
MenuScript	Отрисовка главного меню программы.
Options	Функционал меню настроек.
PauseMenu	Меню паузы.
PlayerControls	Управление персонажем.
Preview	Предпросмотр уровней.
PreviewImage	Скрипт, являющийся составным скрипту Preview
Scalecfg	Конфигурация масштаба на тренировочных уровнях.
LevelSelector	Перемещение между уровнями.
Statistics	Сохранение, загрузка и отображение статистики.

Диаграмма классов, ответственных за взаимодействие с XML-файлом

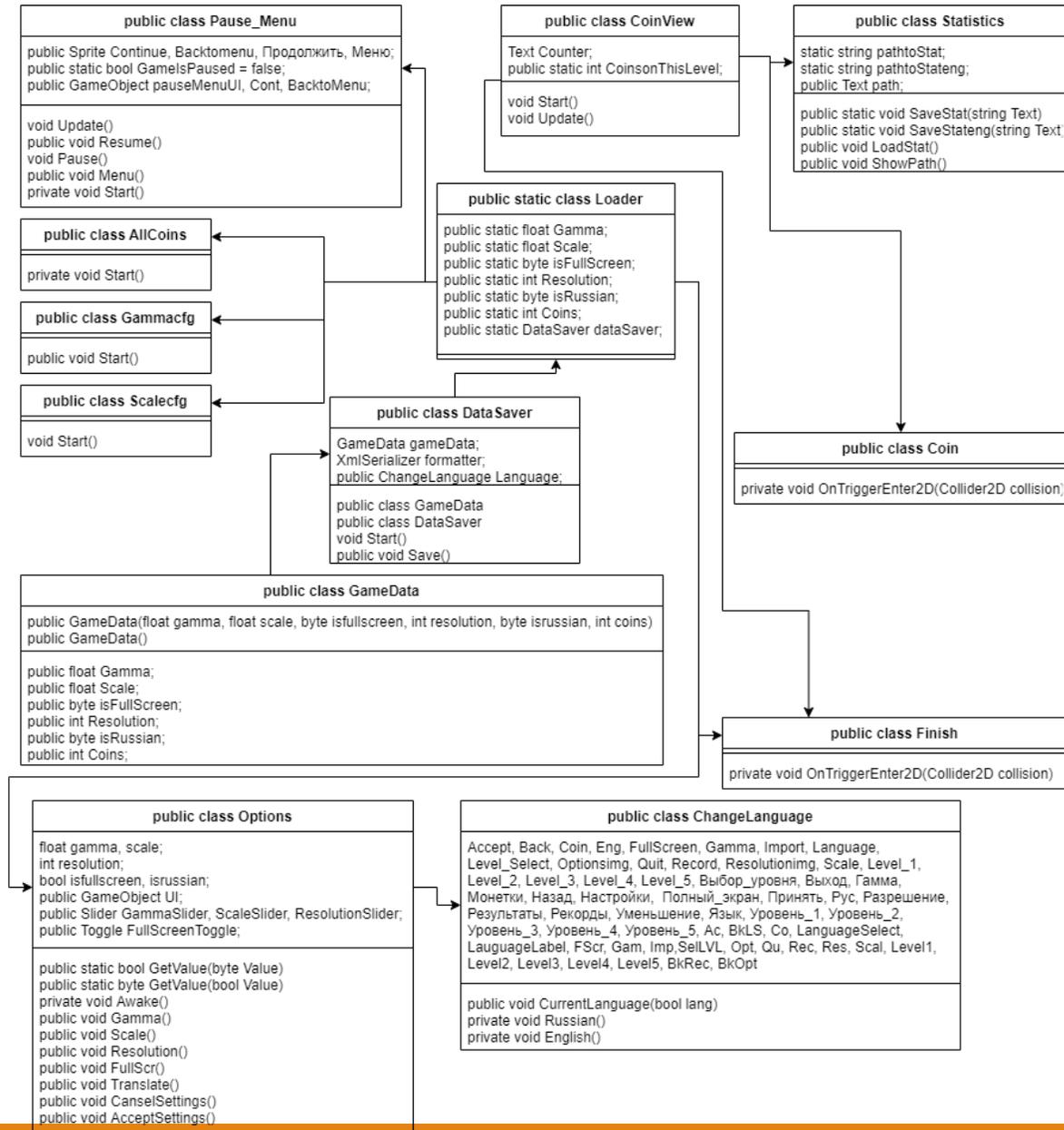


Диаграмма классов, отвечающих за генерацию процедурно генерируемого лабиринта

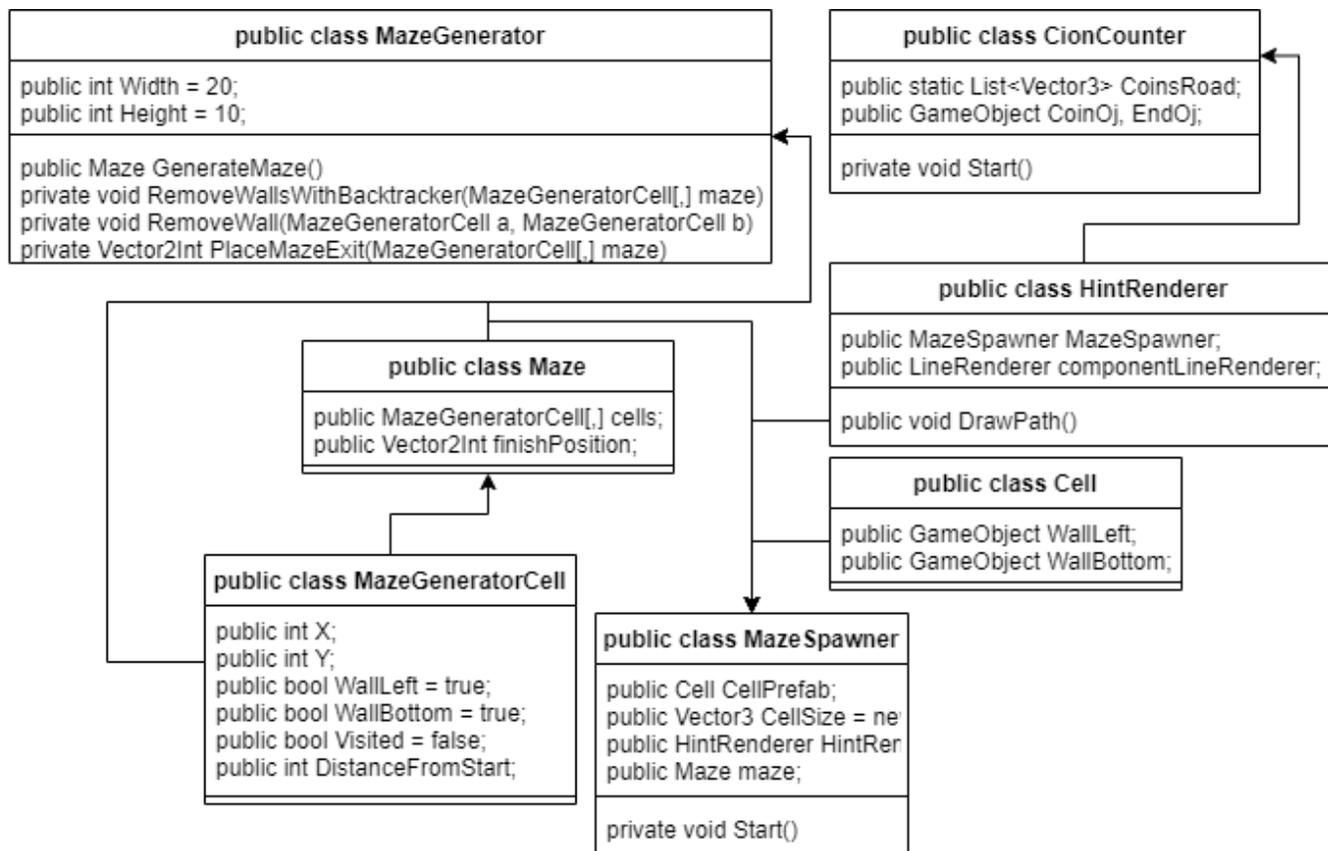


Диаграмма классов, отвечающих за предпоказ уровня

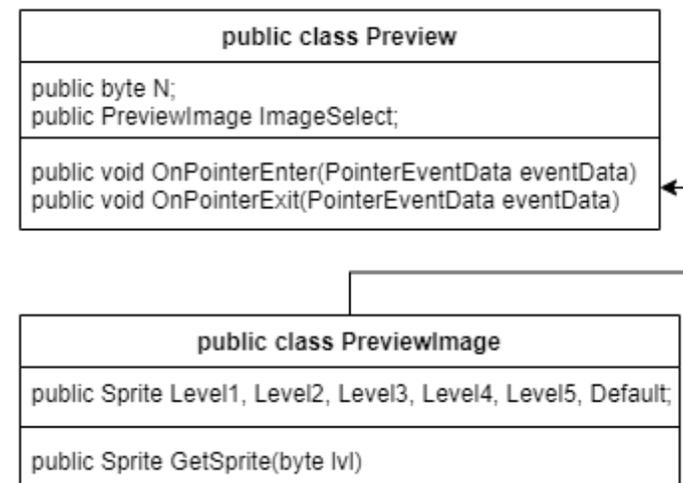


Диаграмма служебных классов

```
public class MenuScript
[SerializeField] GameObject MainMenu,
SelectLevelMenu, OptionsMenu,
RecordMenu;
[SerializeField] GameObject MainMenu,
SelectLevelMenu, OptionsMenu,
RecordMenu;
public void SelectLevel()
public void BacktoMainMenu()
public void Optins()
public void Record()
public void Exit()
```

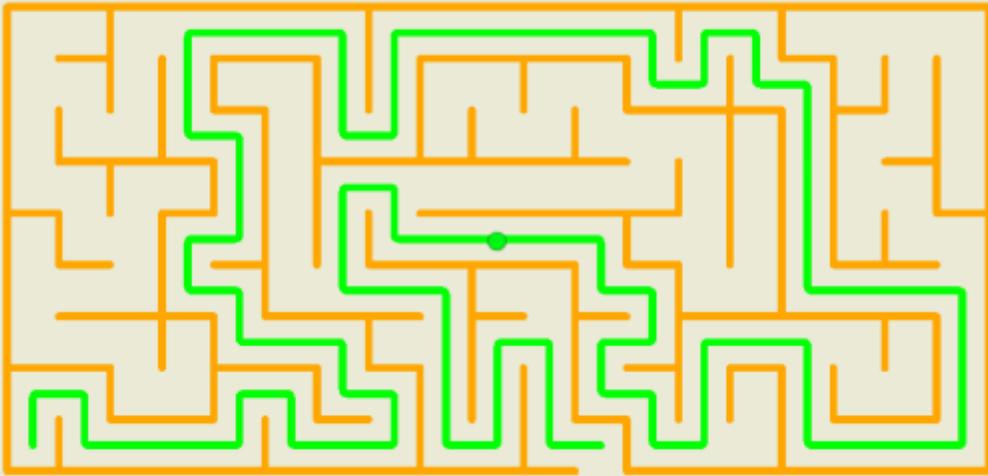
```
public class PlayerControls
public float Speed = 2;
private Rigidbody2D componentRigidbody;
private void Start()
private void Update()
```

```
public class CameraControls
public Transform Player
void Update()
```

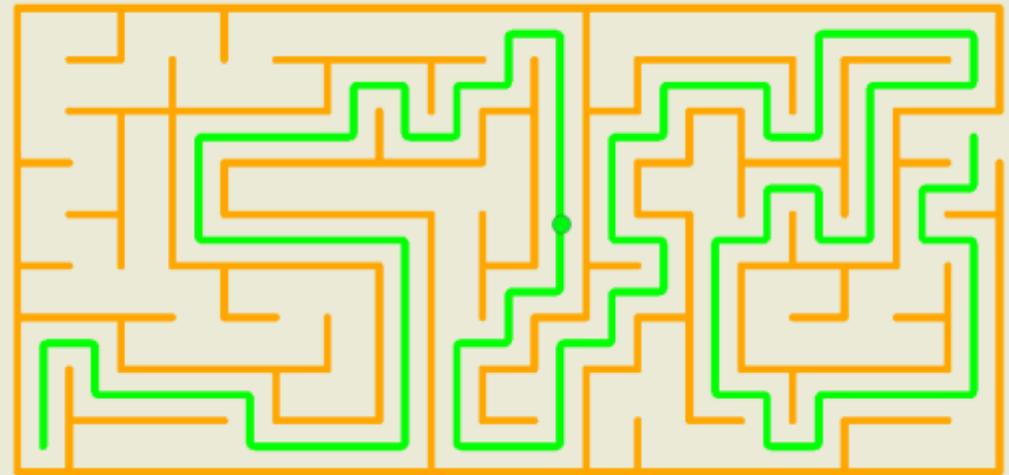
```
public class ScenesSelector
public void Level1()
public void Level2()
public void Level3()
public void Level4()
public void Level5()
```

Процедурный лабиринт

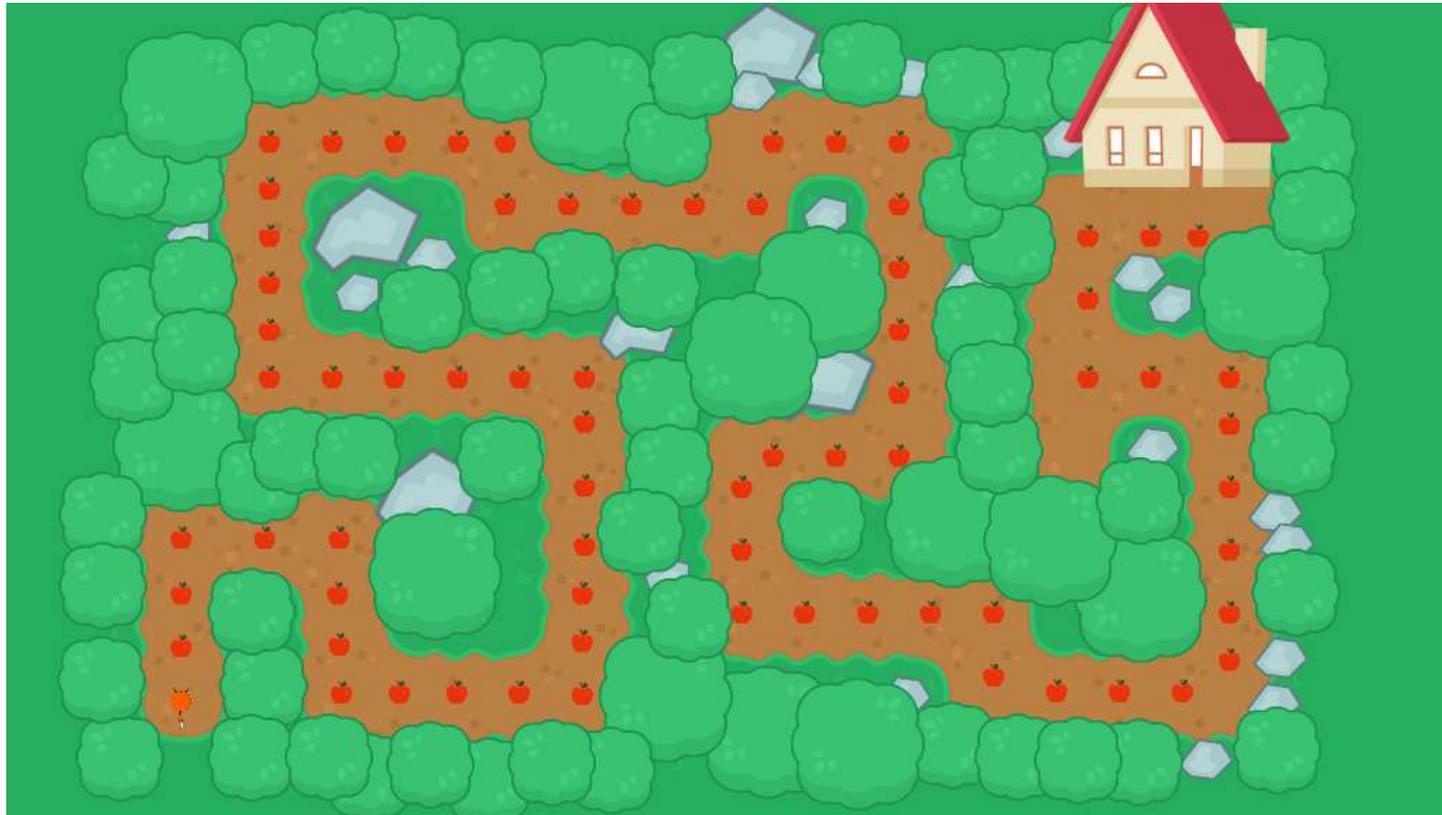
COINS 88



COINS 91



Уровень «Яблочный лес»

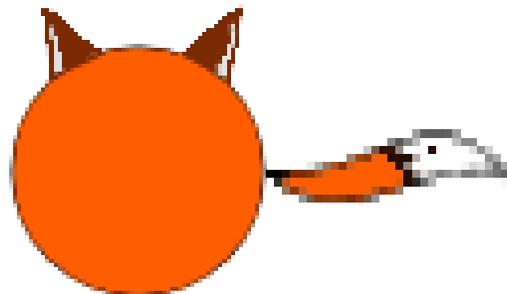


Уровень «Кошкин дом»



Персонажи

Персонаж на уровне «Яблочный лес» и
«Охота за подарками»



Персонаж на уровне «Кошкин дом»

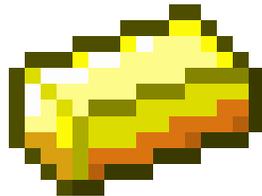


Указатели (Hints)

Указатель на уровне «Яблочный лес»



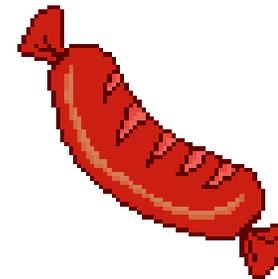
Указатель на уровне «Кубач»



Указатель на уровне «Охота за подарками»



Указатель на уровне «Кошкин дом»

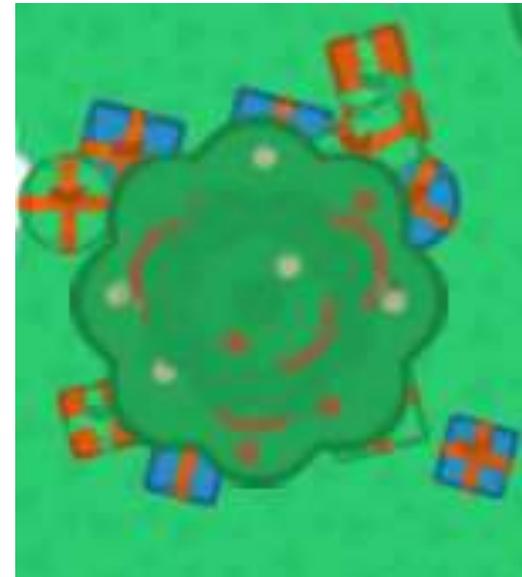


Финиш

Финиш на уровне «Яблочный лес»



Финиш на уровне «Охота за подарками»



Финиш на уровне «Кубач»

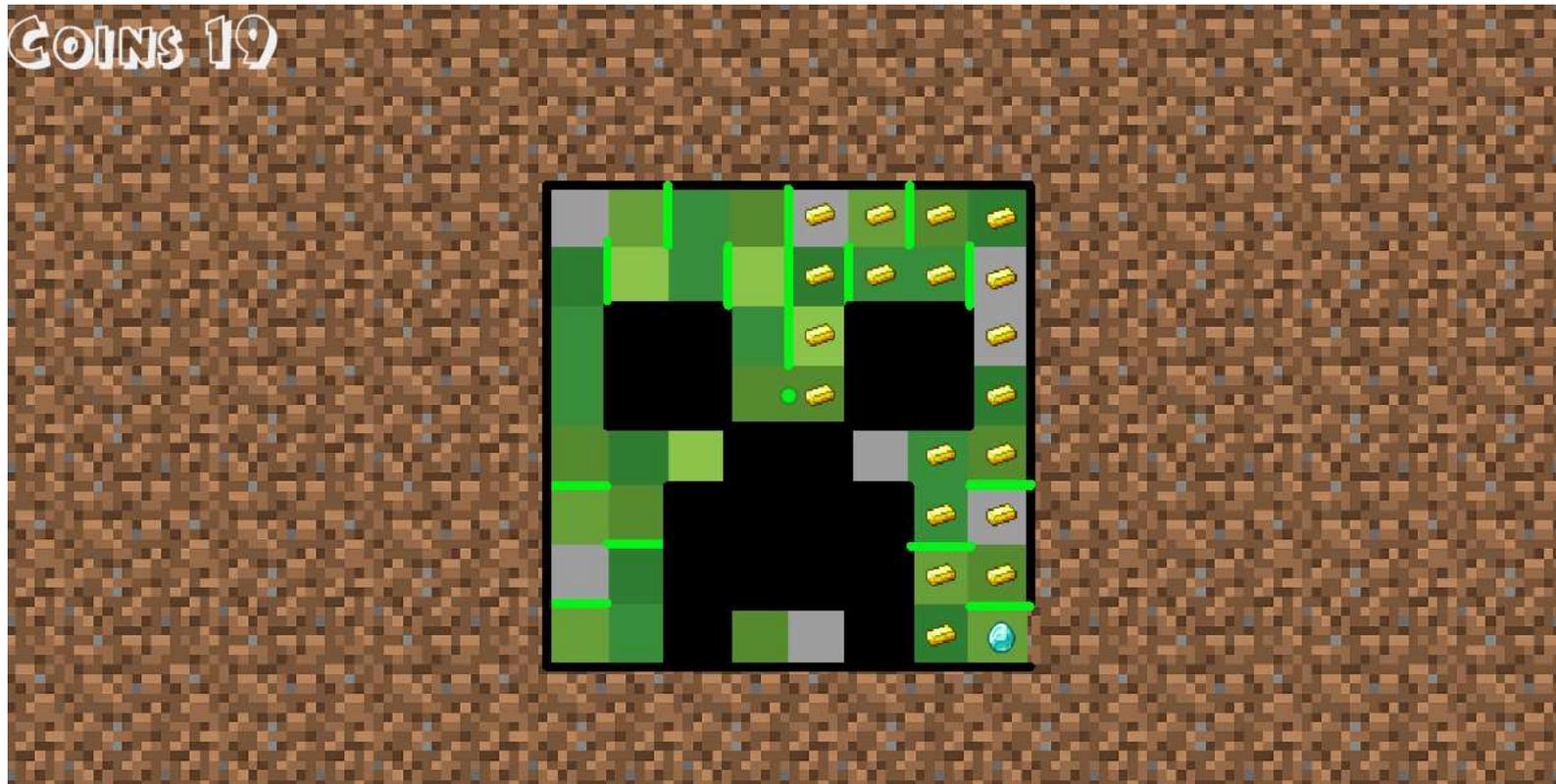


Заключение

В рамках данной работы была спроектирована и реализована программа на платформе Unity, для лечения амблиопии у детей.

В процессе реализации программного продукта потребовалось выполнить ряд задач, связанных с созданием концепции, функциональной и графической составляющей программы. К таким задачам относятся дизайн главного меню, меню выбора уровня, меню настроек, меню статистики программы, персонажа, предметов, тренировочных уровней. Написаны скрипты, отвечающие за весь функционал. Кроме этого, программа была переведена на английский язык для привлечения англоязычной аудитории.

Уровень «Кубач»



Уровень «Охота за подарками»

COINS 24

