1

Санкт-Петербургский политехнический университет Институт компьютерных наук и технологий

**Кафедра «Информатика и вычислительная техника»**

**КУ Р С О В ОЙ П Р О Е К Т**

**Разработка игры "The Darkest Lands"**

по дисциплине «Технологии программирования»

Выполнил студент

гр. 3530901\20002 Сорокин А.О.

Преподаватель Степанов Д.С.

2023 г.

Санкт-Петербург

2023

Санкт-Петербургский политехнический университет

**ЗАДАНИЕ**

**НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА**

студенту группы 3530901\20002 Сорокину Артёму Олеговичу

1. Тема проекта: создание игры The Darkest Lands с случайной генерацией уровня
2. Срок сдачи законченного проекта: 31 мая 2023
3. Исходные данные к проекту: требования к реализовываемому проекту
4. Содержание пояснительной записки: введение с описанием правил игры, основная часть, описание автоматических тестов для бизнес-логики, заключение, список используемых источников.

Дата получения задания: апрель 2023 г.

Руководитель Степанов Д.С.

Задание принял к исполнению Сорокин А.О.

22 апреля 2019

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_TOC_250004)

[ОПИСАНИЕ ПРЕДЛОЖЕННОГО РЕШЕНИЯ 5](#_TOC_250003)

[ТЕСТИРОВАНИЕ ПРОГРАММЫ 6](#_TOC_250002)

[ЗАКЛЮЧЕНИЕ 7](#_TOC_250001)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 8](#_TOC_250000)

# ВВЕДЕНИЕ

**Цель работы:** создать и протестировать игру The Darkest Lands с случайной генерацией уровня.

**Правила игры:**

Игра сделана в жанре Rough like. Игры в этом жанре основаны на перепрохождении одного и того же контента с одним условием – уровень каждый раз генерируется случайно. Это означает, что 2 разных прохождения будут отличаться построением уровня, но не игромеханиками.

Игроку предстоит, проходить сгенерированный уровень, прокачивая характеристики, зачищая комнаты от монстров, с целью убийства финального босса. Которого можно призвать с помощью вещи, выпадающей в качестве награды за убийство мини босса, которого можно призвать с помощью предмета, покупаемого в магазине уровня за 50 душ – игровой валюты, которую можно получить, убивая мобов в комнатах. При этом после убийства босса игрок всё ещё может находиться на уровне и зачищать комнаты до тех пор, пока не захочет перезапустить уровень.

На данный момент в игре присутствуют 4 игровых персонажа, получаемых с 50\30\15\5 % шансом с помощью “молитв” – внутриигровой валюты, выпадающей с 5% шансом после мобов. 4 вида обычных противников, 1 мини босс и 1 финальный босс, имеющий 2 стадии.

Финальной целью игры можно считать получение всех 4х персонажей.

# ОПИСАНИЕ ПРЕДЛОЖЕННОГО РЕШЕНИЯ

Для создания игры был выбран движок Unity, так как он наиболее подходит для освоения с самого нуля. Язык программирования же использовался C#, который писался на Visual Studio 2022.

Игра была написана с использованием концепция MVC (model- view-controller) для отделения бизнес-логики от визуализации, поэтому структура игры разбита на: префабы, скрипты и игровые сцены.

Весь код разбит на 64 скрипта, 3 игровых сцены и папку с префабами.

Такое большое количество скриптовых файлов обусловлено возможностью упрощения создания префабов возможностью повесить на них конкретные нужные скриптовые свойства, чего нельзя было бы добиться, если бы на каждый игровой префаб соответствовал один скриптовый файл. Тогда приходилось бы писать новый файл, копируя множество необходимых частей кода.

В игре присутствуют “пустые” игровые объекты, являющиеся контролерами и генераторами. Так, в сцене уровня есть один генератор, который при запуске уровня создаёт персонажа, комнаты и тд, после чего самоуничтожается, и один менеджер, в котором находятся скрипты, которые не должны быть закреплены за какими-либо игровыми объектами (так как при выключении объекта скрипты также перестают работать), такие как данные инвентаря, менеджер инвентаря, менеджер кнопок и интерфейса, менеджер игровых ресурсов и тд.

В соответствии с выбранным шаблоном разработки, пользователь взаимодействует с частью интерфейса, все нажатия кнопок пользователя вызывают срабатывание метода скрипта менеджера, который в свою очередь обращается к игровым объектам сообщая им, что конкретно должно произойти. Смена интерфейса реализована через переключение UI панелей.

# ТЕСТИРОВАНИЕ ПРОГРАММЫ

Хотя в случае Unity все тесты можно провести визуальным путём в самой игре (если уровень сгенерировался, то значит генератор работает, другого не дано), было разработано несколько тестов игровых скриптов. Также было проведено тестирование в рамках группы, во время которого выявлены и исправлены несколько багов.

Тесты проводились на отдельной игровой сцене, которая не включается в финальный билд проекта. Они проверяют: создание модели графа расположения комнат, корректировку комнаты в зависимости от входных данных, создание выбранного персонажа и работу инвентаря.

Для упрощения тестирования приложения была сделана небольшая консоль, поддерживающая несколько функций: неуязвимость, накрутка ресурсов, открытие всех персонажей сразу.

Все тесты прошли успешно.

# ЗАКЛЮЧЕНИЕ

Была разработана игра в жанре Rough Like, в которой присутствует случайно генерируемый уровень, 4 вида персонажей, 4 вида врагов, 2 вида обычных комнат, инвентарь, магазин, мини босс и финальный босс.

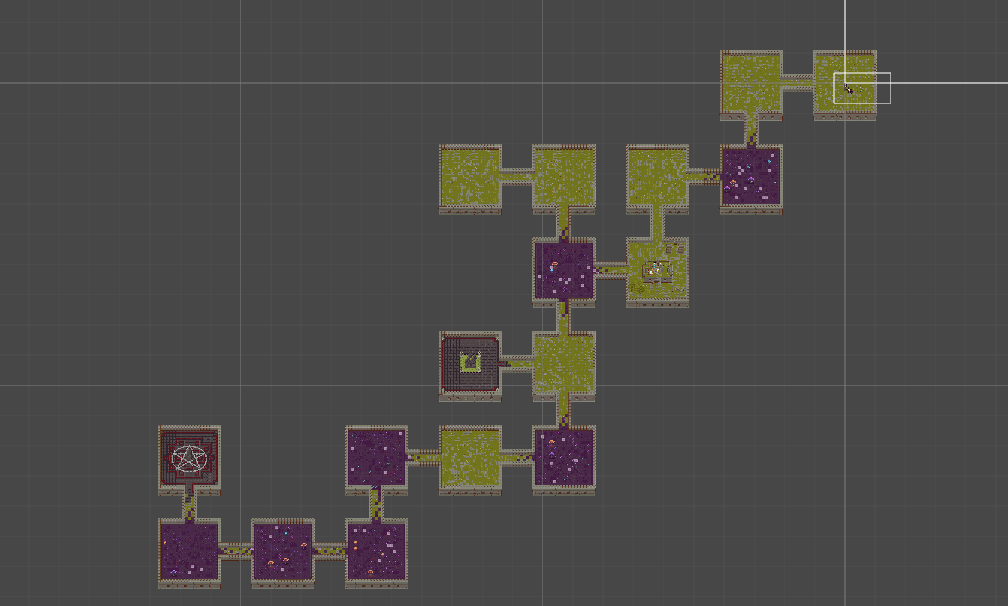
Также были разработаны тесты для игры, проверяющие правильность работы её элементов.

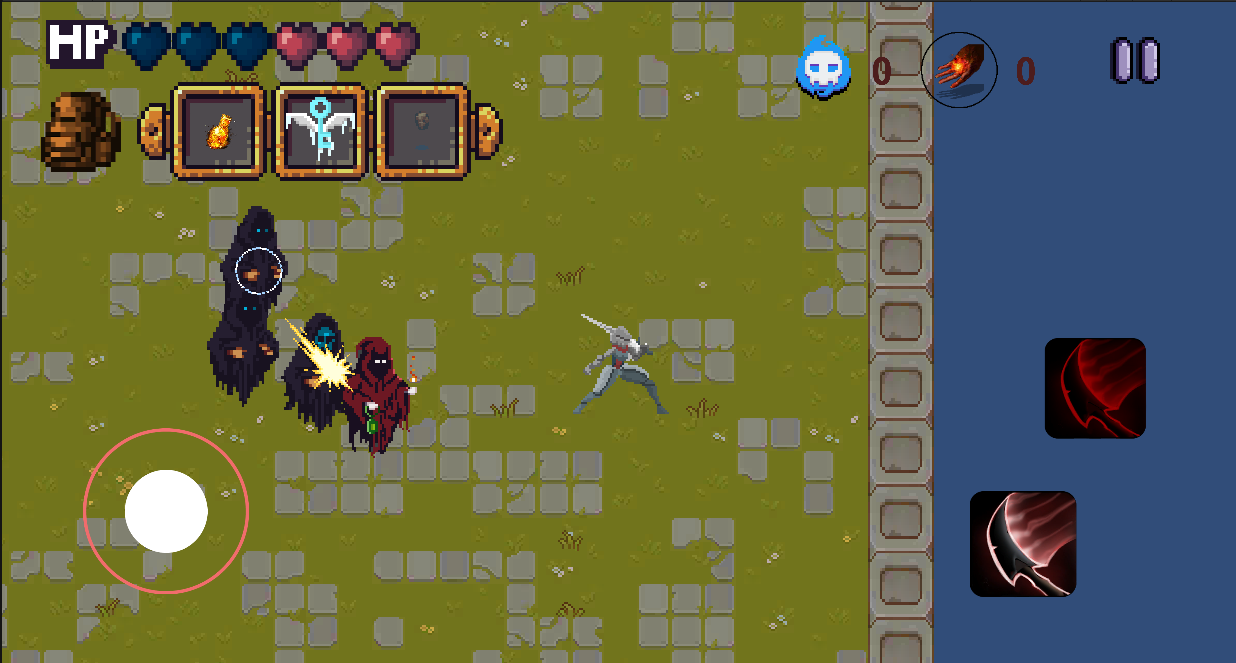
Освоен язык программирования C#.

Исходные файлы приложения лежат в репозитории на GitHub: <https://github.com/ArtemSorokinO/Not-your-pr>









# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

* 1. <https://youtu.be/72sHrcL4kqI> - задание движения игрока и мобов в Unity
  2. <https://docs.unity3d.com/Manual/index.html> - описание языка C# в Unity
  3. <https://habr.com/ru/articles/709960/> - столкновения в Unity
  4. <https://habr.com/ru/articles/248391/> - 2D физика в Unity