

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 21.Б10-мм

Создание 3D-игры в жанре научной фантастики с элементами хоррора

Береснёв Руслан Анатольевич

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
доцент кафедры СП, к.т.н., Ю. В. Литвинов

Санкт-Петербург
2023

Оглавление

Введение	3
1. Постановка задачи	4
1.1. Цель	4
1.2. Задачи на осенний семестр	4
1.3. Задачи на весенний семестр	4
2. Обзор	5
2.1. Существующие решения	5
2.2. Существующие технологии	7
2.3. Используемые технологии	8
3. Реализация	10
3.1. Основные игровые механики	10
3.2. Особенности реализации	13
3.3. Архитектура	14
3.4. Что планируется в весеннем семестре	16
4. Апробация	17
Заключение	18
Список литературы	19

Введение

В современном мире каждый год создаётся очень много видеоигр, с разным бюджетом и разными целями. Практической значимостью многих игр является именно создание уникального игрового мира и захватывающей атмосферы, благодаря которым игрок сможет погрузиться в альтернативную реальность и интересно провести время.

Процесс создания игр упрощается с помощью игровых движков, которые симулируют физику мира и позволяют выстроить удобное взаимодействие между игровыми объектами. На данный момент существует несколько таких движков, однако самыми популярными из них являются Unity Engine [9] и Unreal Engine [10].

Для игровой индустрии были придуманы торговые площадки, на которых после проектирования и разработки игры её можно опубликовать как готовый проект. Примером такой площадки, осуществляющей продажу большинства игр, является Steam [8], где есть как бесплатные, так и платные игры. Видеоигры в Steam интересны как подростковой аудитории (в большей степени), так и более взрослым пользователям.

1. Постановка задачи

1.1. Цель

Целью данной учебной практики стало изучение движка Unity путем создания 3D-игры в жанре научной фантастики с элементами хоррора.

1.2. Задачи на осенний семестр

- Найти и исследовать несколько игр с похожей тематикой;
- сделать обзор используемых технологий и выбрать подходящие;
- спроектировать общую игровую логику;
- реализовать базовые механики игры, такие как перемещение игрока, генерация бесконечной лестницы, оружейная система, создание врагов и генерация комнат;
- описать план апробации.

1.3. Задачи на весенний семестр

- Реализовать остальные игровые механики, такие как генерация случайных предметов в комнатах, поведение различных устройств, двери с системой пропусков, основная игровая локация;
- опубликовать игру на торговой площадке Steam;
- провести апробацию среди игроков, которым интересна данная тематика.

2. Обзор

2.1. Существующие решения

Для сравнения отбирались видеоигры, относящиеся к жанру научной фантастики или к жанру хоррора. Целью обзора существующих решений является выявление интересных механик, использованных в играх с аналогичной тематикой.

Portal

Portal [5] — компьютерная игра от первого лица, состоящая из уровней с различными головоломками. На последнем уровне игроку предстоит выбраться из лаборатории исследования природы порталов. Игра содержит элементы научной фантастики в виде лазеров, движущихся платформ, порталов и устройства для их создания (см. рис. 1). Также здесь реализован интересный концепт плавного обучения игрока в виде разных препятствий, чтобы к моменту побега из лаборатории он знал, как использовать все игровые механики.



Рис. 1: Кадр с созданным порталом

SCP-087

SCP-087 [6] — компьютерная игра в жанре выживальческого хоррора, основанная на материалах статьи из базы данных несуществующей организации «SCP Foundation», которая занимается исследованием и содержанием аномальных объектов [1]. Цель игрока заключается в том, чтобы прожить как можно дольше, спускаясь вниз по лестнице, пока не будет встречен враг, который убьёт игрока. Это может произойти очень неожиданно, поэтому таким образом создается пугающая атмосфера (см. рис. 2). Из данной игры был взят концепт страшной лестницы с большим количеством этажей, по которой игрок может спускаться только вниз.



Рис. 2: Кадр из игры SCP-087

SCP — Containment Breach

SCP — Containment Breach [7] — бесплатная игра с открытым кодом, которая так же имеет жанр выживальческого хоррора. Целью игры является побег из организации SCP в условиях нарушения содержания аномальных объектов (см. рис. 3). Чтобы выбраться, игроку предстоит взаимодействовать с различными объектами SCP и исследовать офис организации. Отсюда была взята интересная идея, заключающаяся в

возможности смотреть информацию об объектах и тем самым исследовать их, а также концепт случайной генерации комнат и различных предметов в них, что позволит немного разнообразить игру.



Рис. 3: Аномальный объект из SCP: Containment Breach

В результате обзора существующих решений были рассмотрены несколько игр, содержащие элементы научной фантастики и хоррора, а также выявлено несколько интересных механик в данных играх.

2.2. Существующие технологии

Для упрощения работы с объектами, визуализации игровой сцены и моделирования физических процессов часто используются игровые движки. Среди существующих игровых движков были отобраны Unity Engine и Unreal Engine, так как они кажутся наиболее простыми для первого опыта разработки игр.

Unity Engine

Это кроссплатформенный игровой движок, выпущенный в 2005 году, изначально предназначенный для операционной системы Mac OS

Х. Используется для разработки мобильных игр под iOS и Android, а также для разработки инди-игр. С помощью Unity создаются и 2D, и 3D игры, а языком программирования игровой логики является C#. В Unity есть несколько основных окон, например, окна сцены и иерархии объектов, где можно оперировать всеми игровыми объектами. Также можно удобным способом задавать поведение каждому объекту, просто прикрепляя к нему файл с кодом в виде отдельного компонента.

Unreal Engine

Движок, который первоначально был предназначен для разработки шутеров от первого лица, однако последующие версии применялись в играх самых различных жанров. Как и Unity Engine, является кроссплатформенным. На Unreal Engine создаются как двухмерные, так и трёхмерные игры. Язык программирования игрового движка - C++. Большим плюсом Unreal Engine является технология Blueprints — это система визуального скриптинга с помощью которой возможно быстрое создание простых игр. Вместо строчного написания кода можно перетаскивать узлы с компонентами и задавать их свойства в интерфейсе.

2.3. Используемые технологии

Выбор был сделан в пользу Unity Engine (далее Unity), который устанавливается в виде отдельного приложения, потому что он является более простым в изучении для начинающих разработчиков, и часто применяется при разработке инди-игр. В приложении есть панель с иерархией объектов, которая описывает зависимости между ними, меню свойств для каждого объекта, а также основное окно с игровой сценой, куда размещаются все нужные объекты.

Для описания поведения каждого объекта в Unity используются скрипты — файлы с кодом, написанные на языке программирования C#. Для работы с движком необходимо использовать пространство имен UnityEngine, а далее для каждого объекта создать свой класс

в отдельном файле и добавить методы, описывающие его поведение. Каждый класс наследуется от базового класса `MonoBehaviour` [3].

Для построения карты использовался набор инструментов ProBuilder [2], так как он прост в использовании и не требует значительных умений в 3D-моделировании объектов. ProBuilder можно загрузить во вкладке Package Manager в Unity.

Также для улучшения графики, а именно добавления свечения и более реалистичных теней, в силу своей простоты в изучении использовался пакет Post-Process Volume [4], который так же можно загрузить во вкладке Package Manager в Unity.

3. Реализация

В этом разделе описаны основные игровые механики, некоторые особенности реализации, приведены архитектура важных классов и связи между ними, а также продемонстрированы планы по разработке на весенний семестр.

3.1. Основные игровые механики

- Игрок появляется на самом верхнем этаже бесконечной лестницы, по которой можно идти только вниз. Управление персонажем происходит от первого лица, передвигаться можно с помощью нажатия клавиш W, A, S, D. У персонажа есть полоса здоровья, которая расположена в левом нижнем углу экрана.
- Реализован оружейный арсенал из трех слотов в правом нижнем углу, между которыми можно переключаться нажатием на клавиши 1, 2, 3, причем активный слот подсвечивается. В самом начале игры у игрока есть два плазменных пистолета и штурмовая винтовка.
- В игре присутствует механика стрельбы и перезарядки. Стрельба осуществляется нажатием на левую кнопку мыши, перезарядка — нажатием на клавишу R. У каждого оружия есть определенное количество патронов в магазине, а также патроны в запасе. Информация о количестве оставшихся патронов отображается на каждом оружии. Игрок не может производить выстрелы во время перезарядки.
- Также персонаж может выбросить оружие нажатием на клавишу Q, а затем поднять его нажатием на клавишу E, если центр экрана, где находится прицел, направлен на оружие, и игрок находится не слишком далеко от него. В таком случае оружие разместится в первый свободный слот. Если перед выбрасыванием оружие находится в стене, оно выталкивается в обратную сторону, чтобы не

вылететь за пределы карты.

- При прохождении каждого этажа с некоторым шансом появляется враг (см. рис. 4). Он будет следовать за игроком, пока не коснется его. В таком случае персонаж мгновенно умирает.

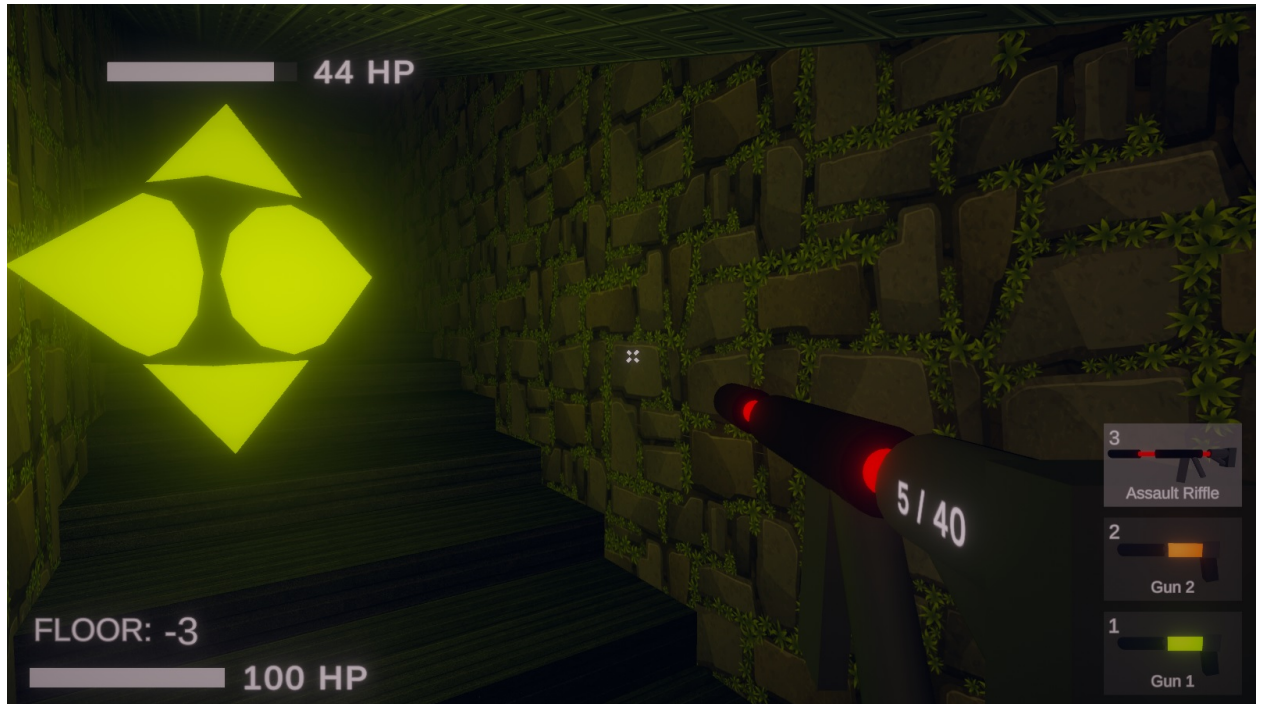


Рис. 4: Враг, следующий за игроком

- У игрока есть графический анализатор — устройство, при активации которого нажатием клавиши G можно увидеть дополнительную информацию о предметах и сущностях. При наведении прицела на объект вблизи высвечивается панель с типом объекта и его характеристиками. В данном режиме над каждой сущностью отображается количество ее единиц здоровья, а также на экране пользователя показывается номер текущего этажа. Данное устройство было добавлено в игру, чтобы реализовать концепт изучения объектов, а также чтобы не слишком сильно заполнять информацией экран игрока (см. рис. 5).
- На каждом этаже помимо врага с определенным шансом создается комната случайного размера и случайным положением входа

в неё (см. рис. 6). В комнатах будут различные предметы, обмундирование и враги разных типов.



Рис. 5: Информация об оружии при активированном графическом анализаторе

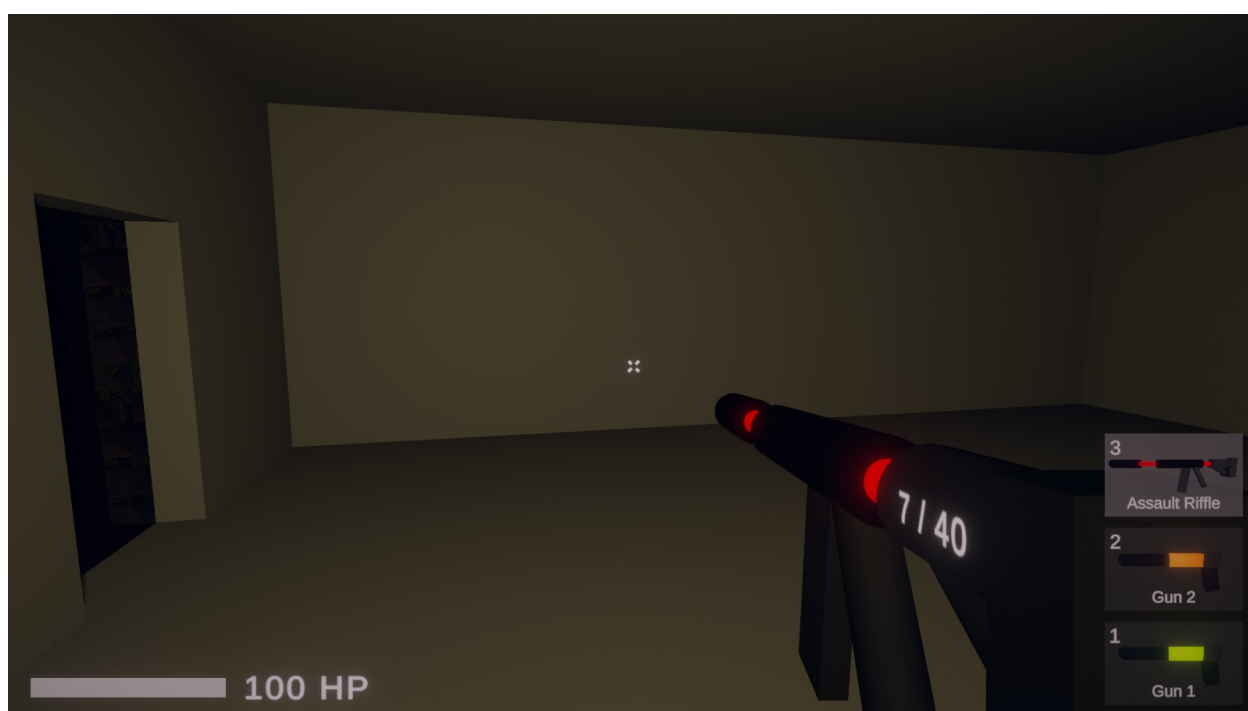


Рис. 6: Сгенерированная комната

- Перед началом игры пользователь находится в главном меню, где можно настроить вероятность появления врага на каждом этаже, а также открыть меню настроек и отрегулировать громкость музыки и звуков, чувствительность мыши, или отключить параметр “Improved Graphics”, который добавляет в игру такие эффекты пост-обработки, как свечение и более реалистичные тени у объектов.
- Во время игры пользователь может нажать на клавишу Escape, чтобы открыть меню паузы, из которого можно выйти в главное меню или закончить игру (см. рис. 7).

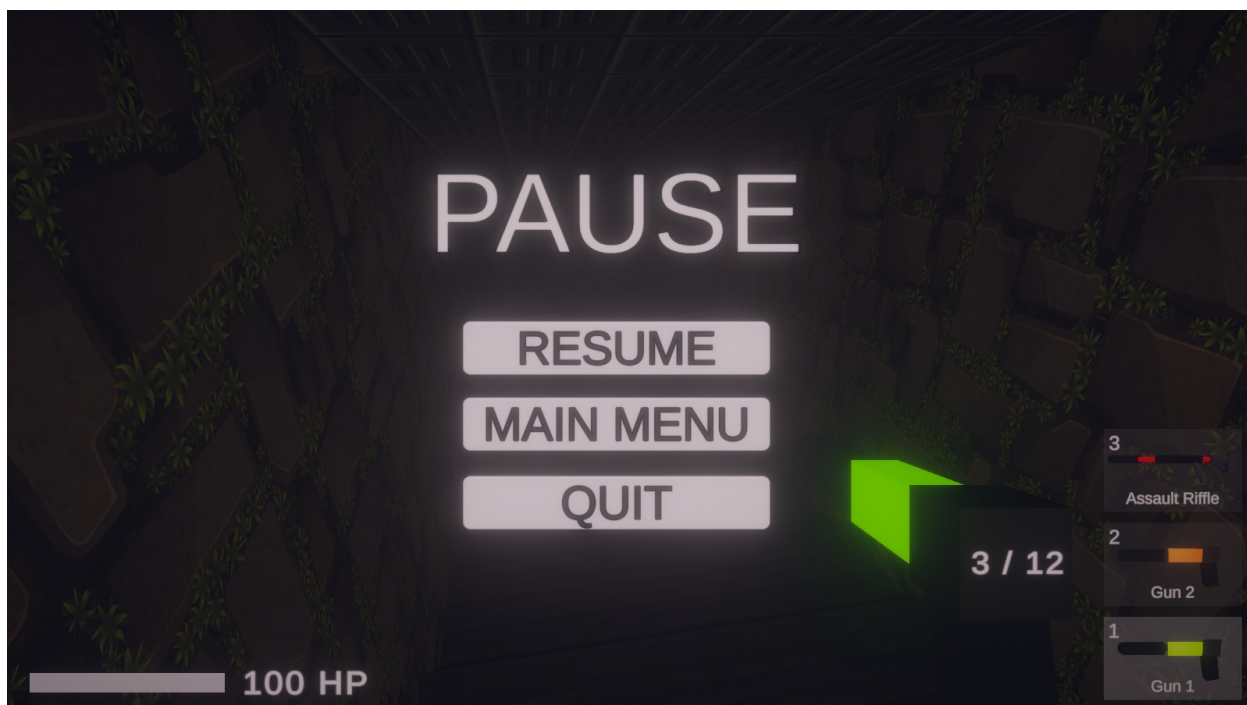


Рис. 7: Меню паузы

3.2. Особенности реализации

Реализация игры содержит следующие особенности, связанные с игровыми механиками:

- Создание бесконечной лестницы осуществляется с помощью двух лестничных структур (нижняя и верхняя), которые поочередно

перемещаются и создают иллюзию бесконечности. Перемещение происходит, когда игрок касается триггера на каждой из лестничных структур.

- Комнаты генерируются из обычных кубов. Сначала создаются все стены и вход, а затем они располагаются таким образом, чтобы образовать пространство в виде комнаты. Диапазон возможных габаритов задаётся программно.
- Для взаимодействия игрока с предметами и сущностями создается луч, исходящий из центра экрана, который возвращает первый объект, с которым он пересечется.

3.3. Архитектура

Игра реализована в виде четырех сцен и переключения между ними с помощью скриптов: сцена с главным меню, сцена с меню настроек, основная игровая сцена и сцена проигрыша, на которую игрок попадает, когда умирает. При загрузке новой сцены все объекты текущей удаляются автоматически, что очень удобно при создании игры.

Все нестатические классы в Unity наследуются от класса `MonoBehaviour`. Приведена диаграмма классов (см. рис. 8), содержащая классы с основными игровыми механиками.

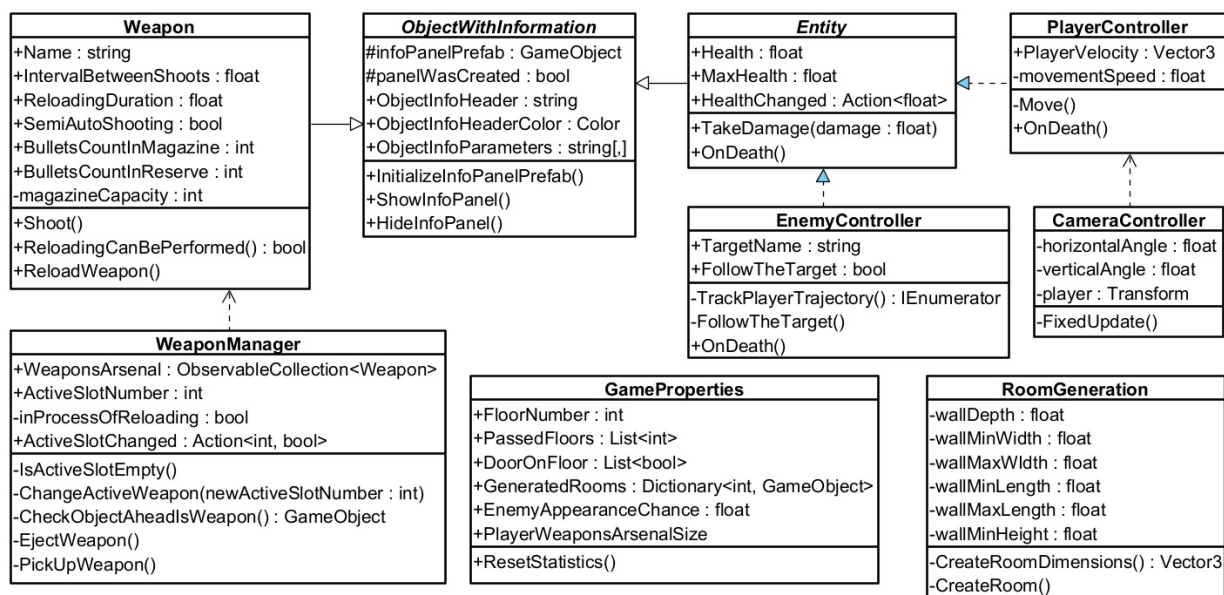


Рис. 8: Диаграмма основных классов

- Абстрактный класс `ObjectWithInformation` реализует концепцию объекта, при наведении прицела на который можно посмотреть информацию о нем: класс формирует панель с типом объекта, его характеристиками и их значениями.
- Общим классом для всех сущностей является абстрактный класс `Entity`, от которого наследуются все объекты, которые должны иметь очки здоровья и уметь получать урон. `Entity` наследуется от `ObjectWithInformation`.
- Основная информация о прогрессе во время игры хранится в статическом классе `GameProperties`. Здесь есть статические свойства с данными о шансе появления врага на каждом этаже или, например, список пройденных игроком этажей. При каждом запуске игры из главного меню статистика обновляется.
- Перемещение игрока регулируются классом `PlayerController`, который наследует класс `Entity`.
- Управление взглядом игрока происходит с помощью движения мыши и реализовано в классе `CameraController`.

- Поведение врага регулируется классом `EnemyController`, унаследованном от класса `Entity`. Здесь реализовано движение врага, его действия при соприкосновении с игроком и отслеживание траектории игрока, чтобы преследовать его.
- Возможности оружейного арсенала игрока, а также возможные взаимодействия игрока с оружием, задаются в классе `WeaponManager`.
- Поведение отдельного оружия реализовано в классе `Weapon`, унаследованном от `ObjectWithInformation`.
- Процесс генерации комнат и параметры самих комнат описываются в классе `RoomGeneration`.

3.4. Что планируется в весеннем семестре

На данный момент игра реализована не полностью, поэтому работа над проектом будет продолжена в весеннем семестре. Планируется добавить генерацию наполнения комнат: патроны, различные объекты жанра научной фантастики, новое оружие и различных врагов, с которым надо будет сразиться, когда игрок попадает в комнату. Также планируется первичное обучение игрока в самом начале, в ходе которого ему будут продемонстрированы основные механики, такие как стрельба, использование графического анализатора и т.д. Можно сделать конечное количество этажей на лестнице и добавить основную игровую локацию после спуска в самый низ, где для персонажа будут подготовлены более серьезные испытания.

4. Апробация

На данный момент апробация не проведена, так как работа над учебной практикой будет продолжена в весеннем семестре. Предположительный план апробации: как минимум, протестировать игру с помощью друзей и знакомых и предложить им оценить проект по методике System Usability Scale; как максимум, опубликовать игру на торговой площадке Steam и собрать оценки и комментарии пользователей, которые захотят в нее поиграть.

Заключение

В итоге были достигнуты следующие результаты:

- Найдены и исследованы несколько игр с похожей тематикой;
- проведен обзор существующих и использованных технологий;
- спроектирована общая игровая логика;
- реализованы базовые механики игры, такие как перемещение игрока, генерация бесконечной лестницы, оружейная система, создание врагов и генерация комнат;
- описан план апробации среди игроков, которым интересна данная тематика.

Ссылка на исходный код — <https://github.com/RuslanBeresnev/SCP-087> (дата обращения: 02.12.2022)

Список литературы

- [1] SCP Foundation. SCP-087 — Лестница. — URL: <https://scpfoundation.net/scp-087> (дата обращения: 1 декабря 2022 г.).
- [2] Unity. PROBUILDER. — URL: <https://unity.com/features/probuilder> (дата обращения: 2 декабря 2022 г.).
- [3] Unity Documentation. MonoBehaviour. — URL: <https://docs.unity3d.com/ru/530/ScriptReference/MonoBehaviour.html> (дата обращения: 1 декабря 2022 г.).
- [4] Unity Learn. Creating a Global Post Processing Volume. — URL: <https://learn.unity.com/tutorial/creating-a-global-post-processing-volume-2019-3#> (дата обращения: 2 декабря 2022 г.).
- [5] Wikipedia. Portal. — URL: <https://ru.wikipedia.org/wiki/Portal> (дата обращения: 1 декабря 2022 г.).
- [6] Wikipedia. SCP-087. — URL: <https://ru.wikipedia.org/wiki/SCP-0871> (дата обращения: 1 декабря 2022 г.).
- [7] Wikipedia. SCP — Containment Breach. — URL: https://ru.wikipedia.org/wiki/SCP_%E2%80%94_Containment_Breach (дата обращения: 1 декабря 2022 г.).
- [8] Wikipedia. Steam. — URL: <https://ru.wikipedia.org/wiki/Steam> (дата обращения: 1 декабря 2022 г.).
- [9] Wikipedia. Unity (game engine). — URL: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)) (дата обращения: 1 декабря 2022 г.).
- [10] Wikipedia. Unreal Engine. — URL: https://ru.wikipedia.org/wiki/Unreal_Engine (дата обращения: 1 декабря 2022 г.).