

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
институт
Информационные системы
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС

_____ П. П. Дьячук
подпись инициалы, фамилия

« ____ » _____ 2021 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 – «Информационные системы и технологии»

Разработка компьютерной игры с Неевклидовой геометрией на платформе
Unity

Руководитель	_____	<u>ст. преподаватель</u>	<u>Ю. В. Шмагрис</u>
	подпись, дата		инициалы, фамилия
Консультант	_____	<u>доцент, к. п. н.</u>	<u>С. А. Виденин</u>
	подпись, дата		инициалы, фамилия
Выпускник	_____		<u>А. И. Шеенко</u>
	подпись, дата		инициалы, фамилия
Нормоконтролер	_____	<u>ст. преподаватель</u>	<u>Ю. В. Шмагрис</u>
	подпись, дата		инициалы, фамилия

Красноярск 2021

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
институт

Информационные системы
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС

_____ П. П. Дьячук
подпись инициалы, фамилия

« ____ » _____ 2021 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Студенту Шеенко Александру Ивановичу

Группа: Ки17-14Б Направление: 09.03.02 – «Информационные системы и технологии»

Тема выпускной квалификационной работы: «Разработка компьютерной игры с Неевклидовой геометрией на платформе Unity»

Утверждена приказом по университету №4894 от 09.04.2021г.

Руководитель ВКР: Ю. В. Шмагрис, старший преподаватель кафедры «Информационные системы» ИКИТ СФУ.

Консультант ВКР: С. А. Виденин, доцент кафедры «Информационные системы» ИКИТ СФУ, кандидат педагогических наук.

Исходные данные для ВКР: Требования к разрабатываемой системе, методические указания научного руководителя, учебные пособия.

Перечень разделов для ВКР: Введение, теоретическая часть, реализация игры, заключение, список использованных источников.

Перечень графического материала: Презентация, выполненная в Microsoft Office PowerPoint 2016.

Руководитель ВКР _____
подпись, дата

ст. преподаватель Ю. В. Шмагрис
инициалы, фамилия

Выпускник _____
подпись, дата

А. И. Шеенко
инициалы, фамилия

« ____ » _____ 2021

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка компьютерной игры с Неевклидовой геометрией на платформе Unity» содержит – 45 страниц текстового документа, 38 иллюстраций, 12 использованных источников.

КОМПЬЮТЕРНЫЕ ИГРЫ, UNITY, 3D-МОДЕЛИ, НЕЕКЛИДОВЫЕ МИРЫ, GAMEDEV.

Актуальность

Игровая индустрия считается на данный момент одной из самых быстроразвивающихся индустрий развлечения. Количество игроков увеличивается, как и количество запросов от них. Людям интересны все новые и новые игровые проекты. В связи с этим разработчики придумывают все новые и новые идеи для игр, а иногда и создают целые жанры.

В качестве объекта исследования выступают компьютерные игры с неевклидовыми мирами.

Предметом исследования является 5 уровней познания игр: восприятие, интерфейс, функции, код и платформа.

Целью выпускной квалификационной работы является разработка компьютерной игры в жанре головоломка с неевклидовым миром.

Исходя из поставленной цели можно выявить следующие задачи:

- Исследовать возможности платформы для разработки игр Unity
- Проанализировать другие проекты данной прикладной области.
- Разработать объекты, скрипты, музыку и другие составляющие проекта и внедрить их в игру.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Анализ предметной области.....	6
1.1 Компьютерные игры.....	6
1.2 Игровая индустрия.....	7
1.3 Игры с Неевклидовыми мирами.....	8
1.4 Примеры игр с неевклидовым миром.....	10
1.4.1 Antichamber.....	10
1.4.2 Superluminal.....	11
2 Средства разработки.....	13
2.1 Unity.....	13
2.2 Blender.....	15
2.3 Visual Studio 2019.....	17
2.4 Fl Studio 20.....	18
2.5 Gimp.....	18
3 Разработка игры.....	20
3.1 Разработка концепта.....	20
3.2 Разработка моделей.....	21
3.3 Совмещение уровней в Unity.....	22
3.4 First Person Controller.....	24
3.5 Музыка и звуки.....	25
3.6 Свет и небо.....	27
3.7 Описание уровней.....	28
3.7.1 Стартовая комната.....	28

3.7.2	Первый уровень	29
3.7.3	Второй уровень	32
3.7.4	Третий уровень.....	33
3.7.5	Закручивающийся коридор между уровнями.....	35
3.7.6	Четвертый уровень	36
3.7.7	Пятый уровень.....	37
3.7.8	Шестой уровень	38
3.7.9	Меню игры.....	39
ЗАКЛЮЧЕНИЕ		43
СПИСОК СОКРАЩЕНИЙ		44
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИК.....		45

ВВЕДЕНИЕ

По данным компании DFC Intelligence на 2020 год в видеоигры играют около 40% населения Земли, это примерно 3,1 млрд человек. Из них 1,5 млрд это пользователи ПК (примерно 48%). И это число не перестает расти.

Ранее компьютерные игры могли предоставить лишь простой игровой процесс, примером могут служить одни из первых игр: Крестики-нолики, Пинг-понг, Ним. Забавным было то, что эти игры имели выигрышную стратегию, которая вбивалась в компьютер и человек физически не мог у него выиграть. И даже в то время у людей появлялось пространство для творчества, сначала были предприняты попытки помешать компьютеру выигрывать в 100% процентов случаев добавляя в него замедляющие вычисления механизмы, а позже сделав компьютер лишь связующим звеном, дающим возможность играть людям друг против друга.

Но если в начале и середине прошлого века игры были просты и включали в себя лишь однообразный геймплей, то на данный момент игры — это большое пространство для творчества и реализации таких идей, которые невозможно реализовать в реальной жизни.

Одной из таких концепций стали Неевклидовы миры. Здесь стоит уточнить, что термин Неевклидова геометрия, не используется в играх с научной точки зрения. С научной точки зрения неевклидова геометрия — это любая геометрическая система, отличающаяся от геометрии Евклида или если говорить по-другому геометрическая система с ненулевым показателем кривизны. В играх же под Неевклидовостью понимается не показатель кривизны пространства, а сама невозможность фигур и пространств в нашей реальности, таких, как Лента Мебиуса, Треугольник Пенроуза и т. д. В общем суть Неевклидовых миров в играх создать оптическую иллюзию невозможности происходящего на экране игрока.

Именно такие игры называют играми с Неевклидовыми мирами (можно также использовать “Невозможными” мирами, но Неевклидовые миры, как термин более распространен).

Целью выпускной квалификационной работы является разработка компьютерной игры в жанре головоломка с неевклидовым миром.

Исходя из поставленной цели можно выявить следующие задачи:

- Исследовать возможности платформы для разработки игр Unity
- Проанализировать другие проекты данной прикладной области.
- Разработать объекты, скрипты, музыку и другие составляющие проекта и внедрить их в игру.

1 Анализ предметной области

1.1 Компьютерные игры

Крестики-нолики считаются первой компьютерной игрой, созданной в одиночку А. С. Дугласом в 1952 году. Как она выглядела показано на рисунке 1.

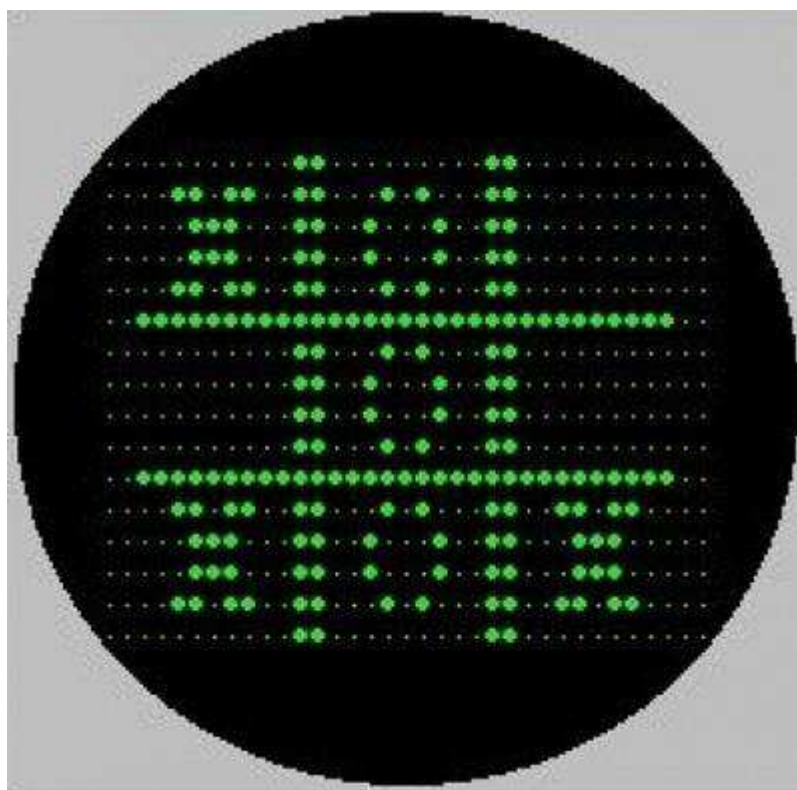


Рисунок 1 - Крестики-Нолики образца 1952 года

С того момента прошло почти 70 лет и за это время компьютерные игры эволюционировали до неузнаваемости. Благодаря развитию компьютерных технологий у разработчиков появлялось все больше пространство для творчества. Появляются новые технологии, которые позволяют делать графику все более реалистичной. Объемы памяти стали позволять вмещать

целые миры с большими количествами объектов. Например, на рисунке 2 показана Forza Horizon 4 выпущенная в 2018 году.



Рисунок 2 - Forza Horizon 4

Сразу же можно увидеть, как сильно все изменилось, сколько объектов вмещается в кадр, качество графики и это, ещё не просматривая технические характеристики, как частота кадров и т. д.

На данный момент компьютерные игры уже называют отдельным видом искусства, по ним пишут книги, создают множество рисунков и другого контента и даже проводят большие турниры на которых разыгрывают миллионы долларов.

1.2 Игровая индустрия

В современном мире создание видеоигр является одним из наиболее крупных сегментов индустрии развлечений. Масштабы игровой индустрии сопоставимы, например, с киноиндустрией. А по скорости роста за последние пять лет индустрия видеоигр существенно ее опережала.

Если раньше игры писали на чистом коде, то сейчас для создания игр используются игровые движки, они выступают программной прослойкой между платформой и собственно кодом игры. Использование готового игрового движка позволяет существенно упростить разработку новых игр, удешевить их производство и существенно сократить время до запуска. Также современные игровые движки обеспечивают кроссплатформенность создаваемых продуктов.

1.3 Игры с Неевклидовыми мирами

Неевклидовы миры отлично прижились в играх жанра – головоломка. Там они раскрыли себя во всей красе благодаря сочетанию в себе двух факторов: эффект озарения, который к нам приходит в тот момент, когда мы понимаем, как нам решить пазл и эффект удивления, когда мы видим, что все происходит не так как должно быть на самом деле. Простой пример, на рисунке 3 можно увидеть туннель, который со стороны выглядит достаточно длинным.

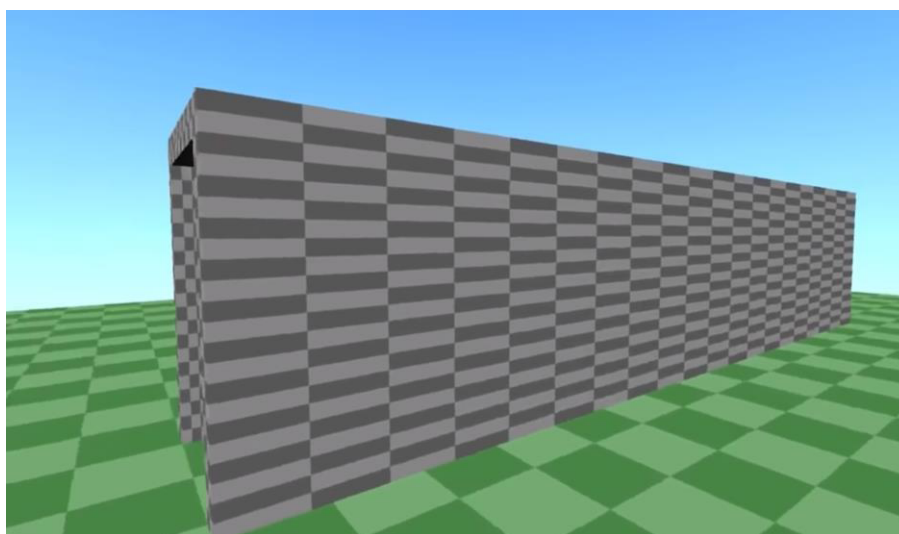


Рисунок 3 - Снаружи туннель длинный

Но стоит нам посмотреть внутрь на рисунке 4, и он стал очень коротким.

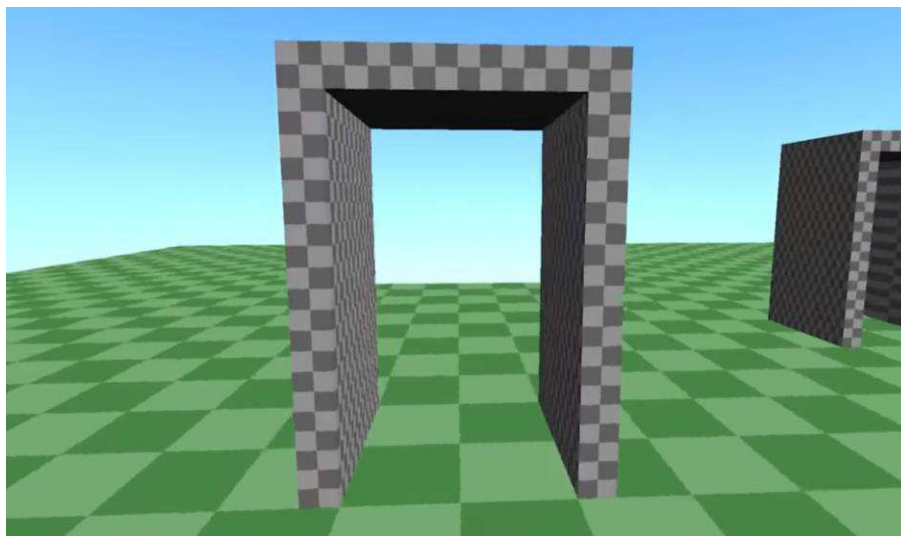


Рисунок 4 - Вид туннеля внутри

Как это реализуется? Для того, чтобы нам сделать такой туннель, нам нужно два туннеля, текстура на входах этих туннелей и телепорты в нужных местах, сначала мы видим текстуру маленького туннеля вначале большого, а когда мы входим в него, нас сразу незаметно телепортирует в тот самый маленький туннель, а на выходе телепортирует обратно. Именно такие «невозможные» элементы реализуются в неевклидовых мирах. Вся суть в том, чтобы придумать способ смоделировать трехмерные элементы так, чтобы с ракурса камеры они выглядели невозможными и когда персонаж с ними взаимодействует он телепортируется между этими элементами так, чтобы целостность картинка не разрушалась.

Но это лишь один пример, неевклидовые миры не ограничиваются одними короткими и длинными коридорами, разработчики придумывают самые разные способы удивить игрока.

1.4 Примеры игр с неевклидовым миром

1.4.1 Antichamber

Если спрашивать про игру в неевклидовом пространстве, большинство игроков вспомнят эту игру и её знаменитую загадку с лестницей рисунок 5.

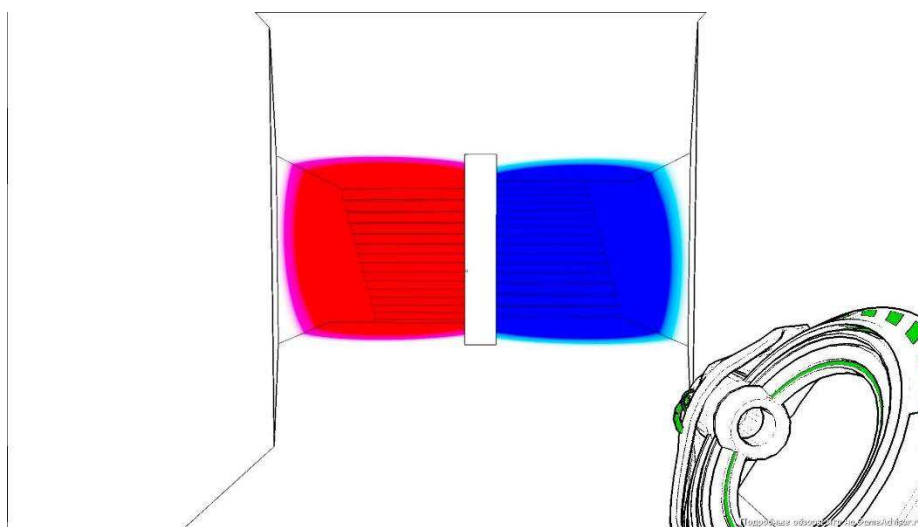


Рисунок 5 - Красная и синяя лестница (также является отсылкой на матрицу)

Вся суть в том, что какую лестницу не выбери, тебя телепортирует в начало и ты опять окажешься перед этими лестницами и так можно ходить по этим лестницам бесконечно. Также можно посмотреть на следующую головоломку рисунок 6, которая представляет из себя рекурсивный лабиринт.

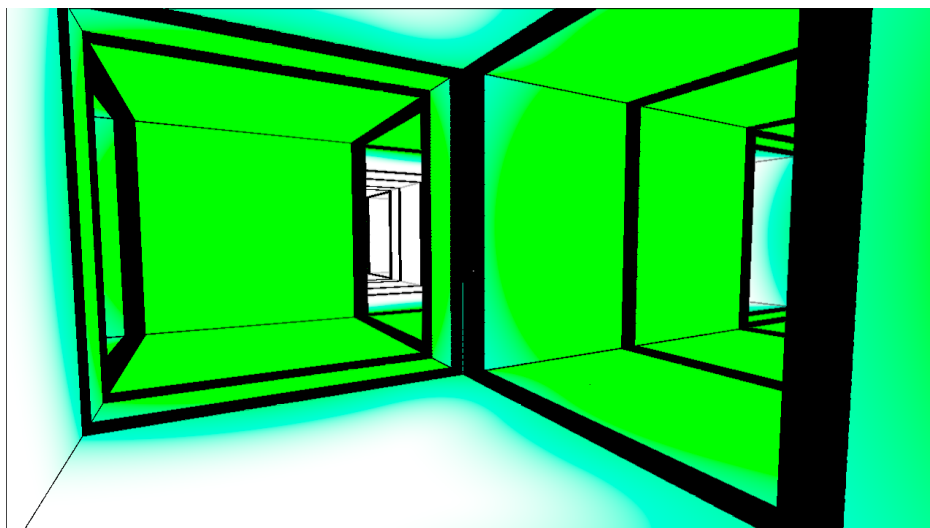


Рисунок 6 – Лабиринт

Если в таком лабиринте сойти с верной тропы, то за следующим поворотом произойдет телепортация в начало лабиринта.

1.4.2 Superluminal

Эта игра целиком и полностью задействует перспективу, в ней есть куча интерактивных объектов и, когда игрок берет их в руки они изменяются в размере в зависимости от того, как далеко от себя их отодвигать. Так же эта игра требовательна к позиции игрока и повороту камеры, это запускает большинство триггеров. Например, пазл в котором нужно встать в правильную позицию, показан на рисунке 7.



Рисунок 7 - Рисунки частей куба на разных объектах

Сначала нужно встать в правильном месте и под правильным углом на этот куб посмотреть, благодаря чему он появится перед игроком, хотя точнее будет сказать игрока телепортирует в такую же комнату, где уже не будет рисунка на стене, а будет лежать куб, хотя можно как раз телепортировать куб в комнату и пока будет перекрыт обзор, рисунок за ним стереть.

2 Средства разработки

2.1 Unity

Unity — межплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие.

Основными преимуществами Unity являются наличие визуальной среды разработки, межплатформенной поддержки и модульной системы компонентов. К недостаткам относят появление сложностей при работе с многокомпонентными схемами и затруднения при подключении внешних библиотек.

На Unity написаны тысячи игр, приложений, визуализации математических моделей, которые охватывают множество платформ и жанров. При этом Unity используется как крупными разработчиками, так и независимыми студиями.

Рабочее пространство Unity можно разделить на несколько блоков, которые каждый может располагать, как ему удобно. Пример рисунок 8.

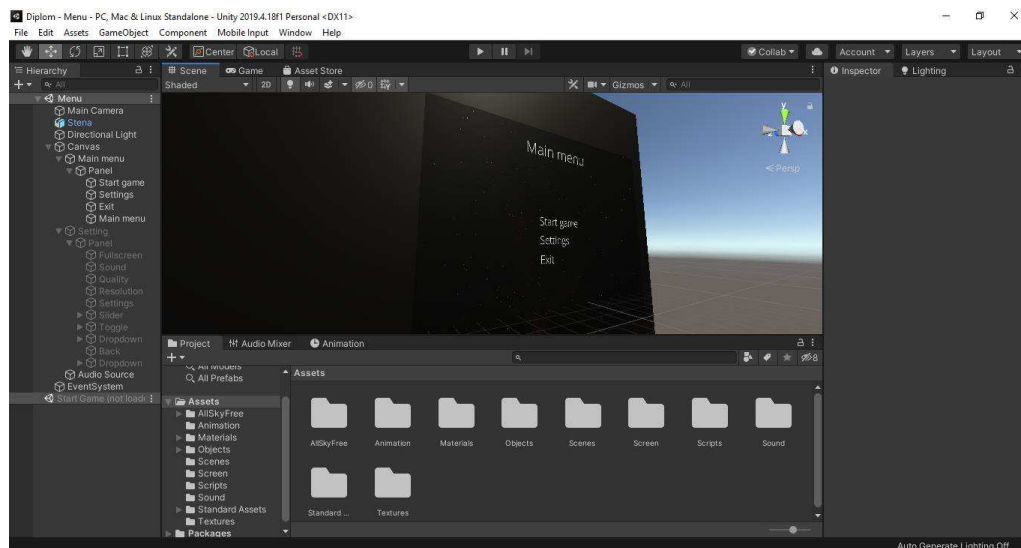


Рисунок 8 - Рабочее пространство Unity

Пространство в данном случае разбито на 4 блока, и так же важный блок Game скрыт, так как удобнее всего переключатся между ним и блоком Scene посередине

Блок Project – расположен посередине снизу. В нем лежат все файлы, которые задействуется в проекте от сцен до музыки, от объектов до текстур. В любой момент мы можем использовать объекты, расположенные в нем перетащив на сцену или в иерархию.

Блок Scene – здесь располагается сверху посередине и является рабочим полем игры. Здесь можно перетаскивать объекты, располагать, как нам нужно и в целом видеть картинку игры со стороны. Говоря кратко, это визуализация

Блок Hierarchy – блок, слева содержащий список всех объектов на сцене, нужен для того, чтобы ориентироваться во всем, что расположено на сцене, так как даже в маленьких играх объекты переваливают за сотню и на самой сцене их бывает искать не удобно, а также есть множество скрытых объектов, которые на сцене невидимы, такие как свет, камера и скрытые объекты, отвечающие за механики, которые игрок не должен видеть.

Блок Inspector – располагается справа и отвечает за показ свойств объекта, который выделен в иерархии. Свойство Transform есть у всех объектов, отвечает за расположение объекта на сцене с помощью трехмерной системы координат, его поворот и размер. Остальные блоки добавляются по мере необходимости. Таких блоков очень много. Перечислим основные из них.

Mesh Rendered – отвечает за визуальную часть объекта, то как он будет выглядеть.

Collider – отвечает за физические свойства объекта, как он взаимодействует с другими объектами, за его гравитацию, упругость и т. д.

Scripts – Важно отметить, что сам код, который выполняется в играх не может быть самостоятельным объектом, скрипт должен быть прикреплен к чему-то. Обычно это то с чем он и работает, например, если у нас есть дверь, которая должна открываться, то скрипт открытия двери к ней и прикрепляется. Если скрипт явно с чем-то конкретным не взаимодействует, то для него нужно создать объект пустышку, который никак не будет отображаться на сцене, но будет существовать на ней.

2.2 Blender

Сам Unity содержит в себе несколько стандартных объектов в виде геометрических фигур, но на них игру не создашь. И поэтому для этого нужно прибегать к другим программам, заточенным на создание 3d-моделей. Blender одна из самых популярных таких программ, за счет удобства использования и бесплатного распространения.

Интерфейс у данной программы достаточно прост, но в первое время можно запутаться во всех кнопках и горячих клавишах, так как блендер

предоставляет кучу возможностей видоизменять объект. В основном рабочее пространство блендера разбито на 3 элемента, рисунок 9.

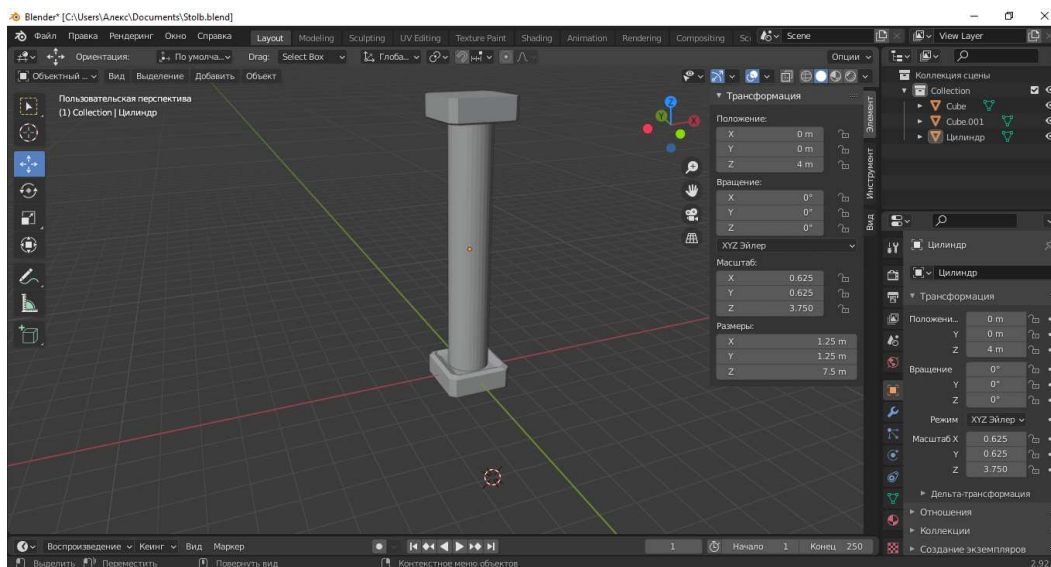


Рисунок 9 - Рабочее пространство Blender

Сцена – большое окно занимающее почти все пространство, является местом, где визуализированы наши объекты и где мы совершаем все видоизменения этих объектов. Изменяем в размерах, режем на части, соединяем, создаем выпуклости и т. д.

Иерархия – располагается справа вверху и содержит в себе список всех объектов, находящихся на сцене, так как при создании сложных объектов все равно используются стандартные геометрические фигуры с видоизменениями. Например, столб на рисунке 9, имеет по кубу сверху и снизу и цилиндр по середине.

Настройки – справа снизу располагаются тонкие настройки всего, как самого блендера, так и объекта, который сейчас выделен, например, логическая функция, когда мы хотим соединить несколько объектов или вычесть один из другого.

2.3 Visual Studio 2019

Когда Unity создавался он изначально был заточен под C# (на данный момент можно также использовать JavaScript). Поэтому при настройке Unity нужно выбрать редактор кода, по умолчанию будет Visual Studio. VS является хорошим вариантом в данном случае потому что при создании скриптов из Unity там сразу же будет создан удобный шаблон для скрипта и так же VS поддерживает интерактивные подсказки для библиотеки UnityEngine (библиотека для работы с логикой объектов в Unity). На рисунке 10 показан стандартный шаблон скрипта без внесения доп. кода.

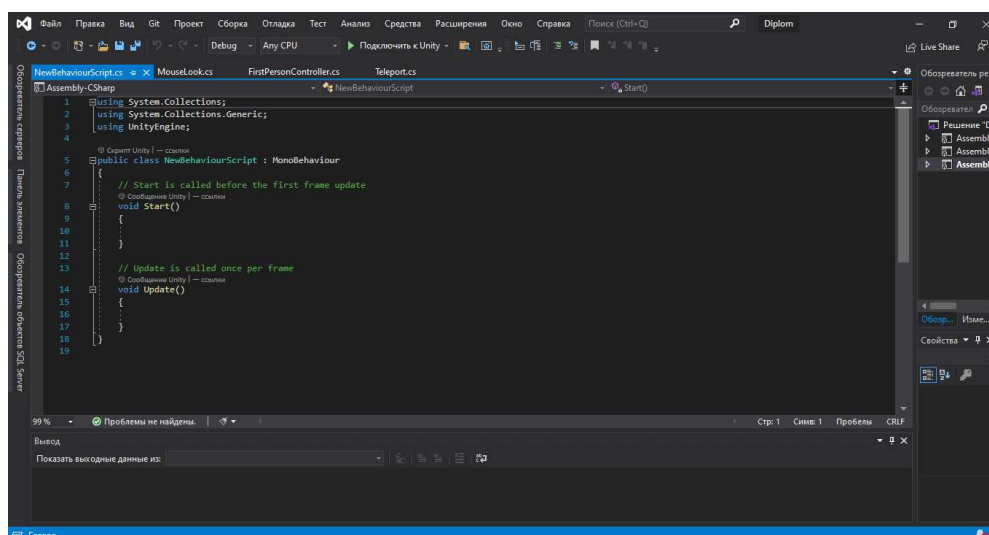


Рисунок 10 - Стандартный шаблон

Здесь расположено два метода Start() и Update(). Start() вызывается до первого фрейма работы игры, а Update(), каждый фрейм, но его рекомендуют вызывать как можно реже так как это ресурсозатратный метод.

2.4 Fl Studio 20

В каждой игре есть звуки, а в большинстве и мелодии на заднем фоне. Fl Studio давно зарекомендовало себя, как хорошая средство для создания музыки. Для игры было написано 2 музыкальных сопровождения в меню и в саму игру. Интерфейс программы можно увидеть на рисунке 11.

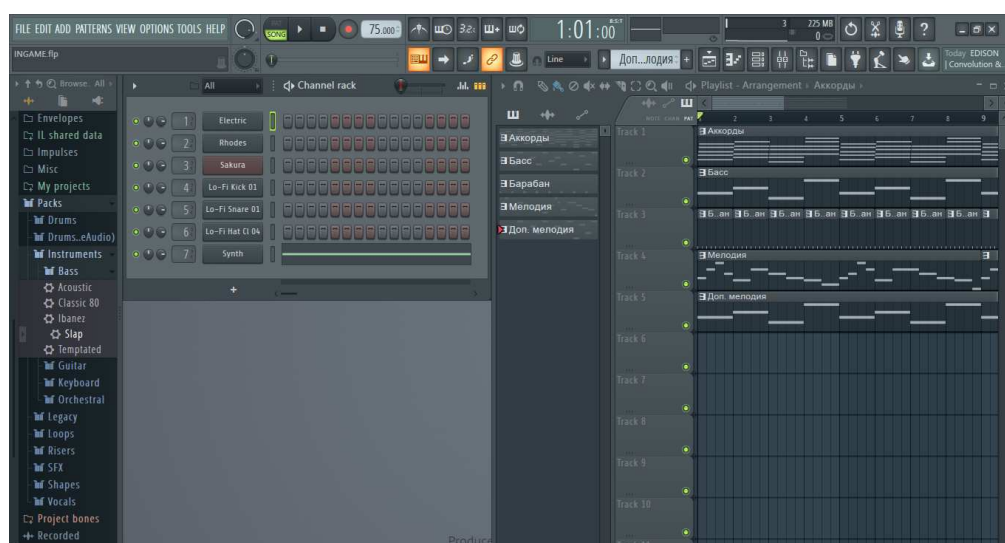


Рисунок 11 - Fl Studio с открытой мелодией, которая используется в игре

2.5 Gimp

Для реализации текстур был использован бесплатный аналог Photoshop – Gimp. В Unity используются Material для текстурирования объектов. К ним можно прикрепить несколько изображений, каждое из которых будет выполнять свой функционал. Это нужно для создания псевдообъемной картинке. Например, асфальт не является полностью плоским, а имеет множество микротрещин и т. д. В Gimp есть возможность создания изображений для всех слоев Material, но в этом проекте, большинство слоев не использовалось. Интерфейс Gimp показан на рисунке 12.

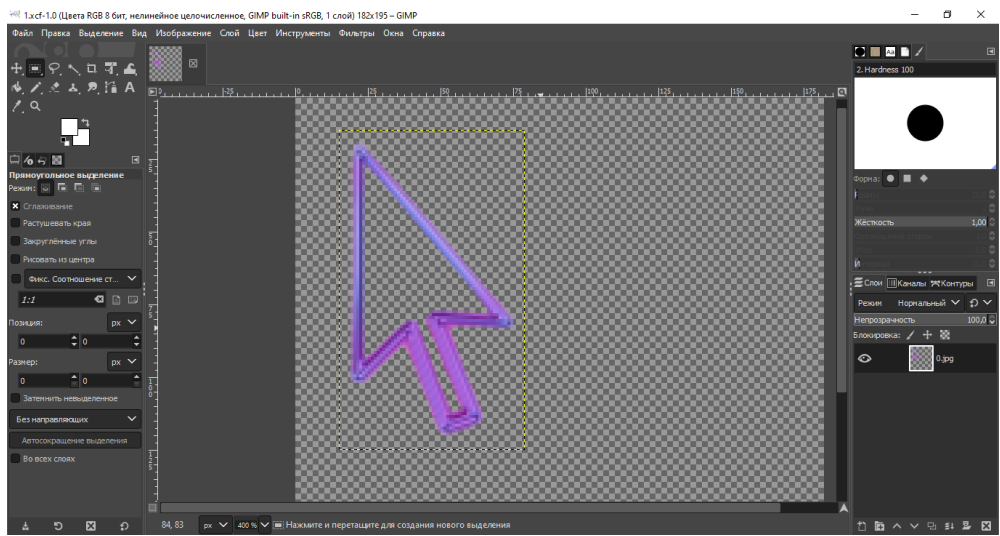


Рисунок 12 - Интерфейс Gimp с открытым файлом курсора для игры

3 Разработка игры

3.1 Разработка концепта

При разработке игр в жанре головоломка, обычно берется несколько подряд идущих уровней, которые соединены коридорами, дверьми или телепортами. На каждом уровне демонстрируется, какая-то новая механика и паззлы, связанные с ней. Ближе к концу механики могут совмещаться, но это происходит не всегда.

В данной игре переходы между уровнями соединятся коридорами. И на каждом уровне демонстрируется новый тип загадок. Сюжета в игре нет, игроку нужно добраться из пункта А в пункт Б, проходя попутно головоломки.

Уровней с уникальными механиками в игре 6, всего комнат 8, кроме основных, еще стартовая комната с финишной. На рисунке 13 можно увидеть все комнаты со стороны.

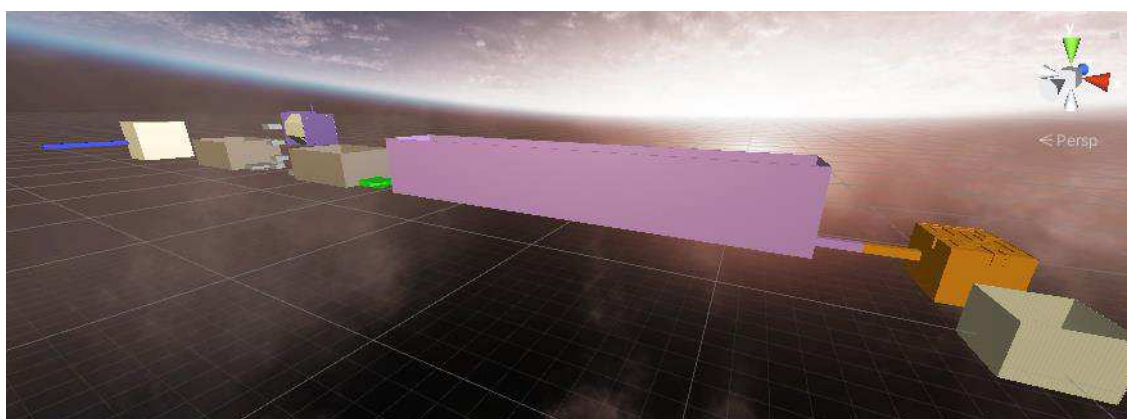


Рисунок 13 - Вид снаружи на все комнаты

3.2 Разработка моделей

Для того, чтобы разработать игру в Unity, сначала нужно было спроектировать 3D-модели объектов, из которых и состоит вся игра. Модели экспортируются из Blender в Unity в нужном формате. Модели условно можно поделить на 2 типа: модели уровней (комнат) и модели предметов.

Модели предметов – это модели, состоящие из небольшого количества объектов. Представляют из себя то, чем уровень наполняется: кнопки, двери и т. д. Модель двери показана на рисунке 14.

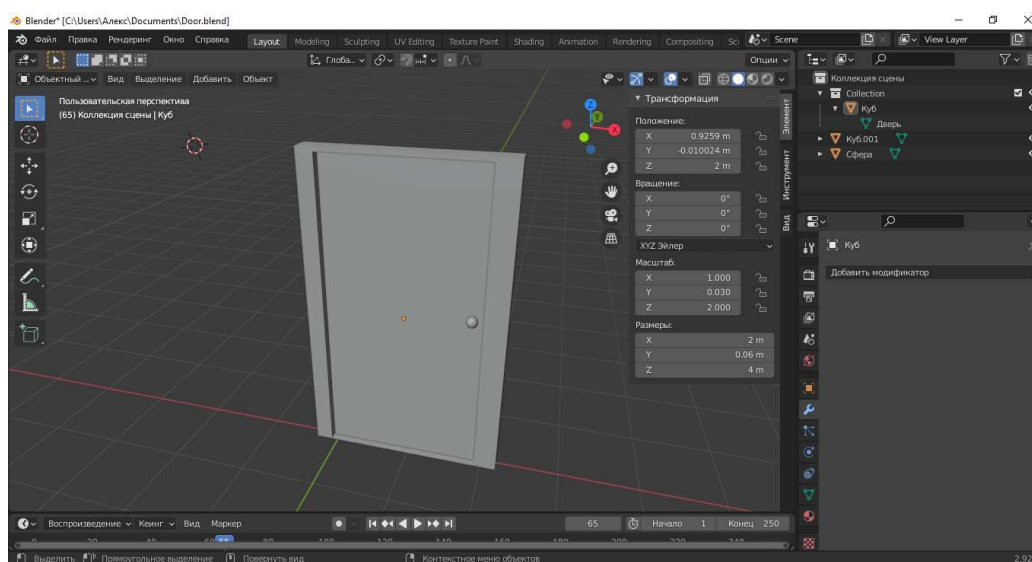


Рисунок 14 - Модель двери, состоящая из 2 кубов и 1 сферы

Модель уровней – это модели целых комнат, состоящих из большого количества объектов, так как каждая стена и каждый выступ — это отдельный объект. На рисунке 15 показана модель комнаты, состоящая из более чем 50 объектов.

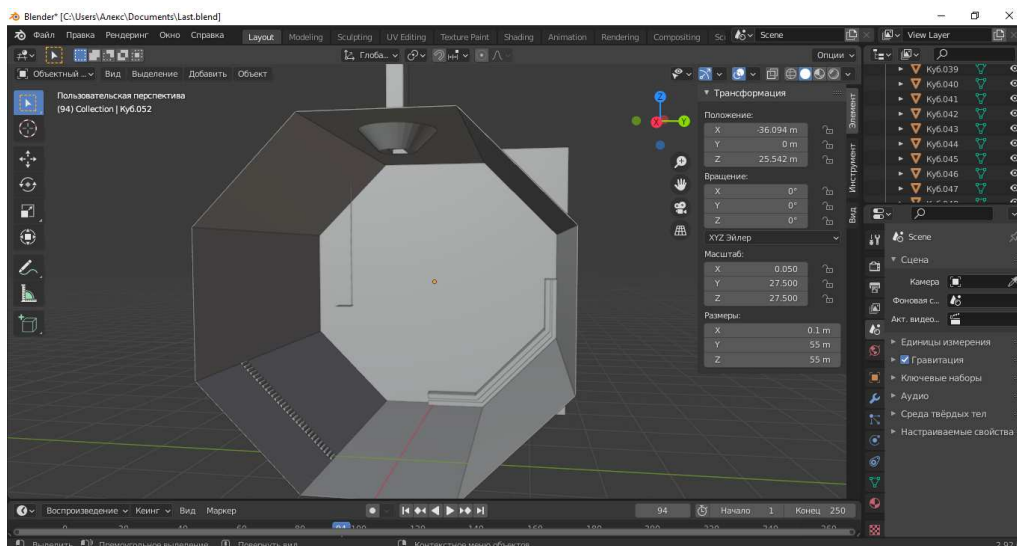


Рисунок 15 - Уровень в форме восьмиугольника в разрезе

В сумме было создано и использовано в игре 38 моделей из них 8 моделей комнат и 30 моделей предметов.

3.3 Совмещение уровней в Unity

Модели в формате .FBX загружаются в Unity и там идет их размещение на сцена и подгонка по размерам.

Как было сказано ранее, все уровни соединяются коридорами между собой и в некоторых местах телепортами в виду невозможности размещения моделей уровней рядом. На рисунке 16 показаны 2 комнаты, которые соединены коридором.

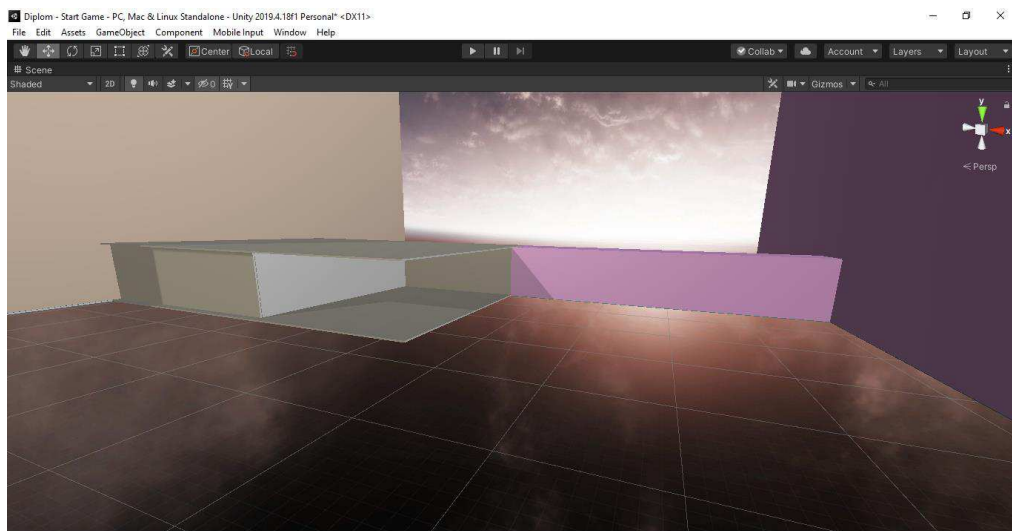


Рисунок 16 - Коридор перехода между комнатами

На рисунке 17 расположены 2 комнаты, которые соединены с помощью телепорта.

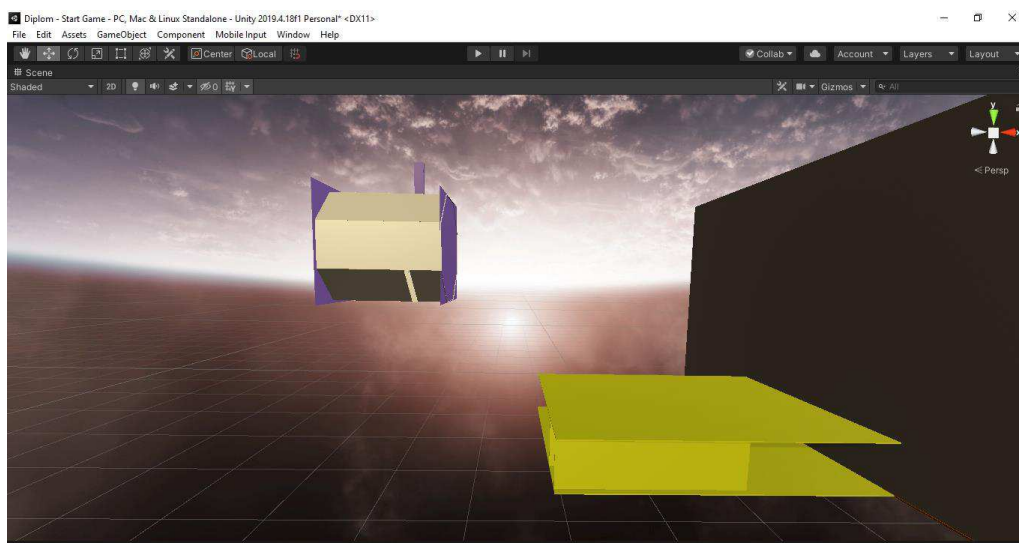


Рисунок 17 - В середине коридора игрока телепортирует в комнату на заднем плане

Телепортация в данном случае выступает вынужденным решением геймдизайна, так как по механике паззла комната на заднем плане изменяет свое положение и если её расположить близко к другим комнатам, то это

может вызвать баги, когда коллайдеры объектов будут входить друг в друга что может не позволить проходить игру дальше и в целом для игрока будет выглядеть некрасиво.

3.4 First Person Controller

Чтобы играть в игру, нужен персонаж, которым игрок будет управлять. В Unity есть специальный механизм для создания персонажей Character Controller, но это лишь средство для упрощения его создания. Персонаж, которым будет управлять игрок, есть почти в каждой игре, поэтому разработчики Unity создали Unity Standards Assets Pack, в котором содержатся многие стандартные вещи, которые разработчики могут использовать в своих играх: вода, деревья, трава, земля и т. д. Так же в ней содержится First Person Controller, который предназначен для управления персонажем, которым мы будем играть. В нём сразу прописаны методы для передвижения на WASD, прыжок на пробел, камера через которую игрок будет видеть саму игру и её развороты по повороту мыши, а также тряски камеры при ходьбе, звуки ходьбы и т. д.

Сам FPS не имеет изображения в игре, так как это контроллер персонажа от первого лица, это значит, что посмотреть на себя со стороны в игре нельзя. Но у него есть коллайдер для реализации логики столкновений с другими объектами.

Но пусть FPS очень мощное средство у него есть один существенный недостаток. Это взаимодействие контроллера с другими скриптами. Поэтому по мере разработки скрипт, отвечающий за работу контроллера претерпел сильные изменения, которые будут описаны в следующих разделах.

3.5 Музыка и звуки

В игре есть звуки, которые работают либо при взаимодействии с интерактивными объектами в игре, например, открытие двери запустит звук скрипа двери или звуки, которые звучат в определённой области. Их можно привязывать. Первые привязаны к тому объекту взаимодействие, с которым их включит, а вторые могут, как находится в какой-то области самостоятельно, так и быть привязаны к объекту, от которого они должны исходить. На рисунке 18, выделена дверь с привязанными к ней звуками.

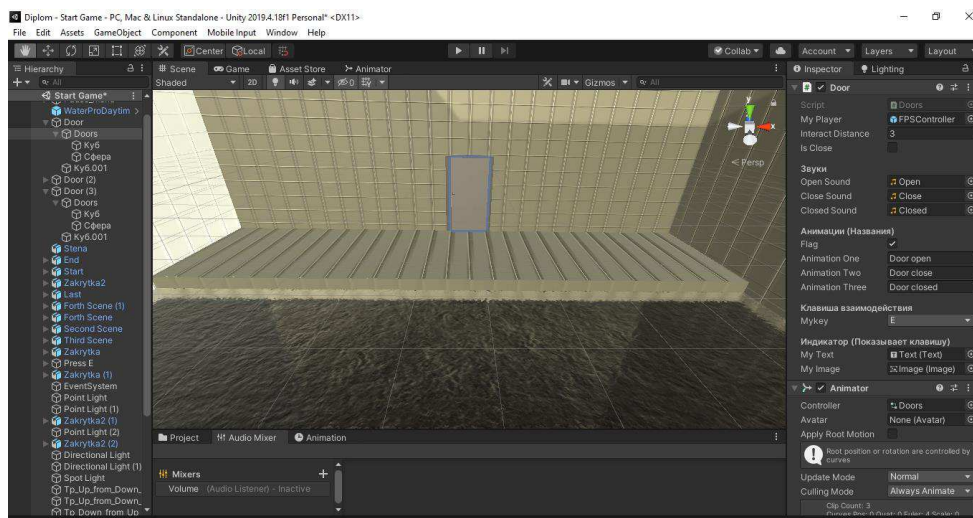


Рисунок 18 - Справа выделена дверь и её скрипт в котором есть 3 вида звука

В зависимости от взаимодействий с дверью (о них в следующих разделах), она будет издавать 3 разных звука, которые отображены справа в скрипте.

Также на рисунке 18 есть вода, она же издает звук не от взаимодействия, а если игрок находится рядом с ней.

Кроме звуков для игры было написано 2 мелодии. Одна в главное меню и одна в саму игру. Длительность мелодий 1:28 и 0:55 соответственно. На рисунках 19 изображена дорожка для мелодии в главное меню.

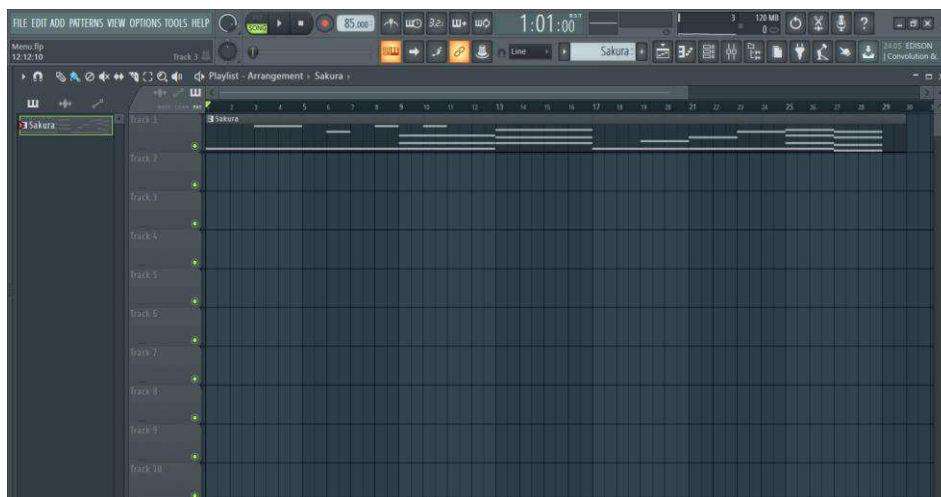


Рисунок 19 - Мелодия главного меню

Использован встроенный инструмент Sakura и добавлено несколько плагинов для создания объемного звучания. На рисунке 20 изображены дорожки главной мелодии.

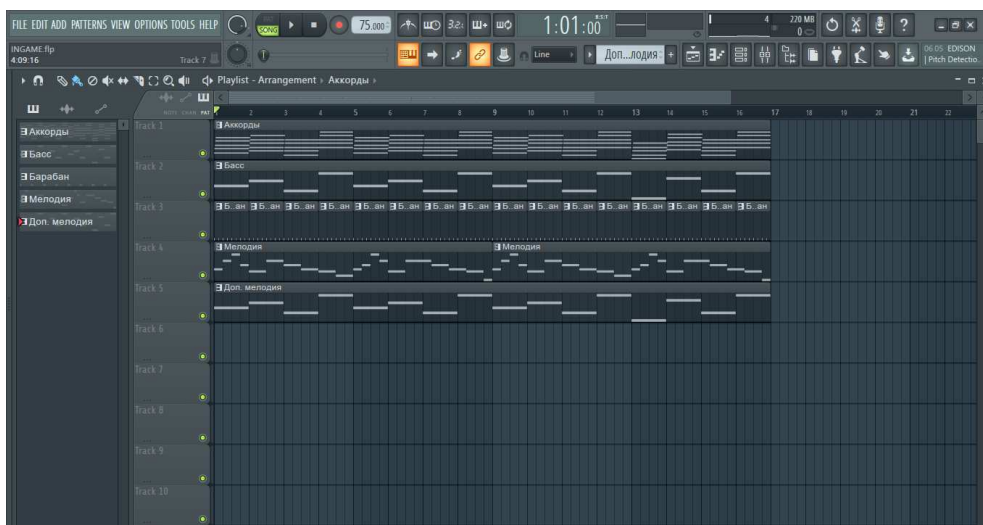


Рисунок 20 - Мелодия, играющая на протяжении игры

Были задействованы 3 разных синтезатора для аккордов, основной мелодии и дополнительной, а также бас и барабаны.

3.6 Свет и небо

В Unity для неба есть специальные два раздела рассчитанные под солнце и небо, Солнце отвечает, в какую сторону будет отбрасываться тени, а небо, какую текстуру Unity будет отрисовывать на заднем плане, где не расположено объектов. Текстура неба была взята из бесплатного набора с материалами для Unity, на рисунке 21 показано, как выглядит текстура неба.

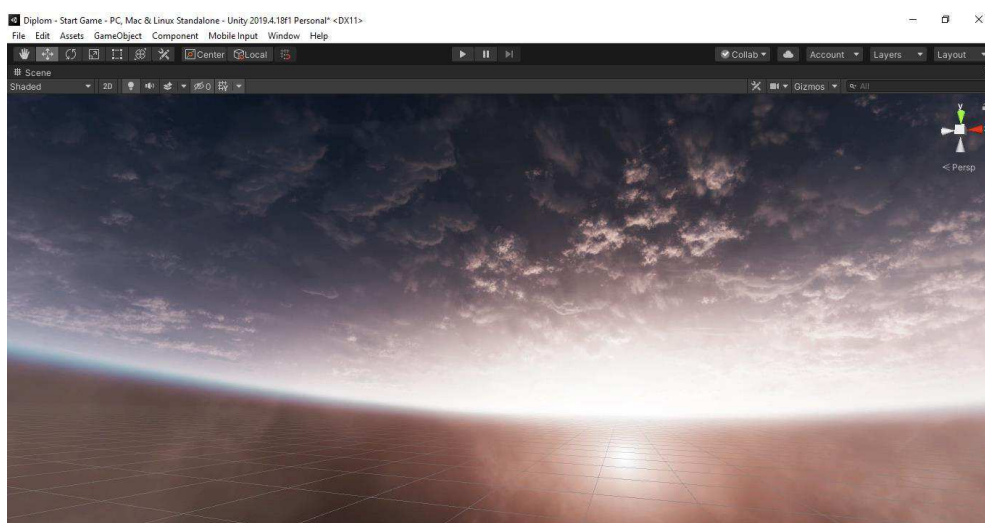


Рисунок 21 - Как выглядит небо в сцене

В Unity есть несколько видов света, глобальный называется Direction Light, он и располагается в солнце. Обычно он расположен один на сцене, но так как некоторые комнаты в игре замкнутые, то было размещено 2 DL, чтобы стены со всех сторон не были темными. Также в Unity есть ещё маленькие источники света, которые тоже использовались в игре о них далее в описаниях уровней.

3.7 Описание уровней

3.7.1 Стартовая комната

Стартовая комната представляет из себя комнату с открытым небом с клетчатыми белыми стенами, бассейном и дверью с переходом к первому пазлу. Сама комната уже была показана на рисунке 18.

Вода взята из набора из Unity Standards Assets Pack. Модель двери сделана в Blender и содержит 3 анимации: открытия, закрытия и поворота ручки, когда она заперта. Открытие и закрытие проигрываются в зависимости от флага, который отвечает за расположения самой двери и ручки, которые поворачиваются, если игрок нажмет клавишу взаимодействия на дверь на нужном расстоянии, также есть переменная, которая говорит заперта ли дверь и если её значение равно true, то проиграется анимация поворота ручки и дверь не откроется. В игре клавишей взаимодействия выступает английская E. При наведении на дверь появляется всплывающая подсказка, показанная на рисунке 22.

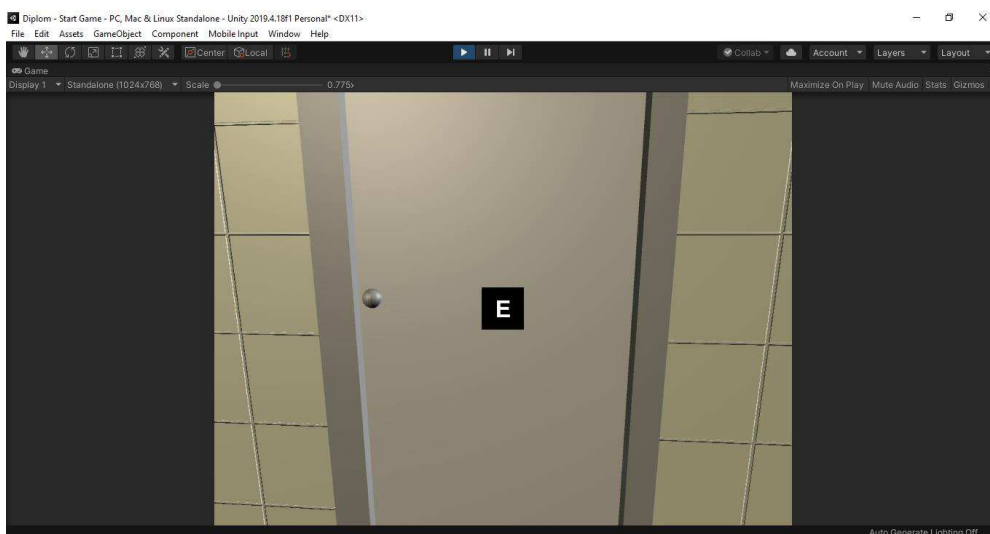


Рисунок 22 - Всплывающая подсказка

Чтоб вывести картинку всплывающей подсказки был использован встроенный метод `OnMouseOver()`, который отслеживает, наведена ли камера игрока на дверь, а когда дверь уходит из поля зрения метод `OnMouseExit()` убирает подсказку. Внутри метода прописана логика условий, какую из анимации нужно запустить и также вызвать событие, которое вызовет метод для запуска звука.

3.7.2 Первый уровень

Первый уровень встречает игрока закрытым помещением с проходом в коридор слева и справа, но правый коридор уже находится на втором ярусе комнаты и первоначально его не видно. Где также находится дверь выхода на следующий уровень. На рисунке 23 показана комната так, как её будет видеть игрок только в неё войдя.

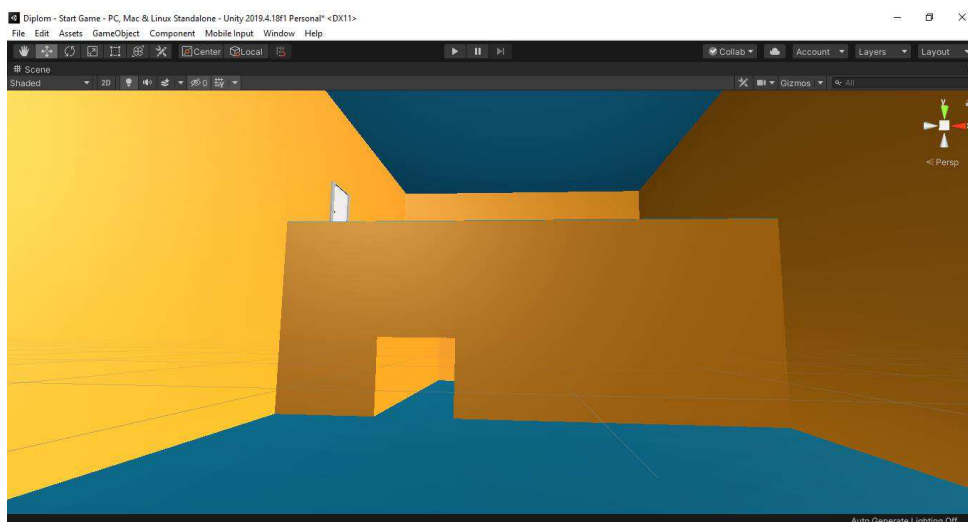


Рисунок 23 - Начало первого уровня

Войдя в проход слева, игрок будет телепортирован так, чтобы почти не было заметно, что он вообще был перемещен в лабиринт, из которого он должен найти выход (рисунок 24).

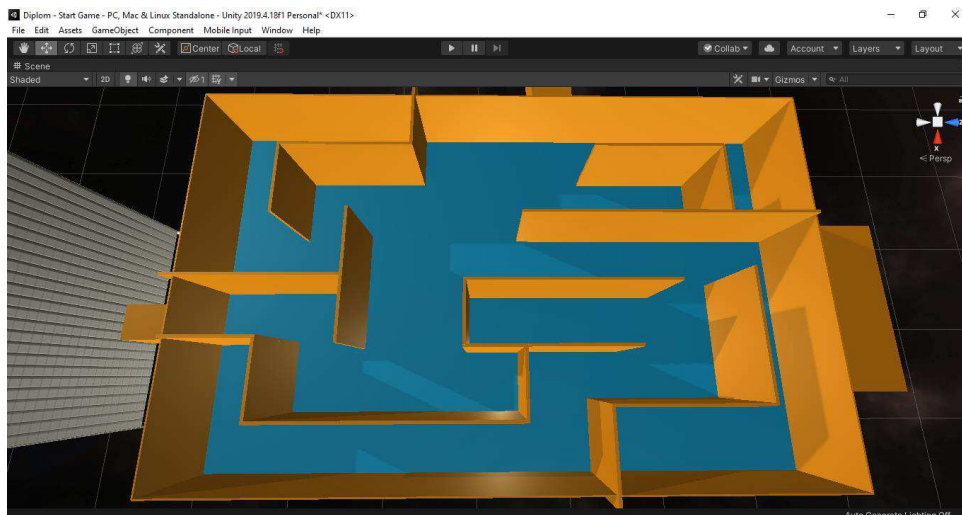


Рисунок 24 - Лабиринт вид сверху (без крыши)

На выходе из лабиринта он будет также незаметно телепортирован в выход сверху справа (рисунок 25).

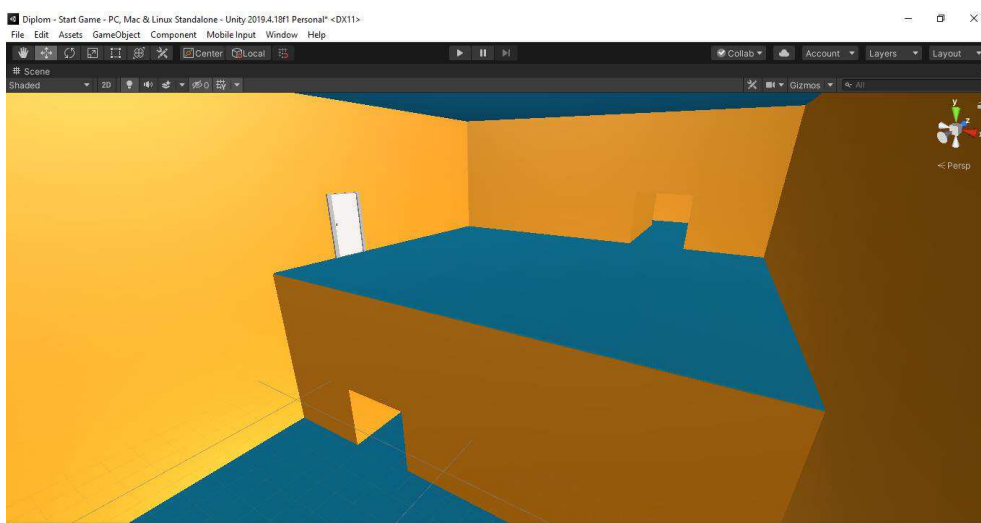


Рисунок 25 – Комната (вид сверху)

Чтобы реализовать телепортацию были созданы невидимые коллайдеры и скрипт, который отслеживал вхождения игрока в них перемещение между ними. Здесь стоит уточнить, что в Unity, когда несколько разных элементов отвечают за одно и то же и если эти элементы одновременно пытаются совершить действие, то между ними чтобы не возникло конфликтов произойдёт блокировка одного из действий, чтобы избежать этого. Например, Character Controller отвечает за позицию и перемещение игрока и когда происходит попытка переместить игрока из вне, то это происходит на один фрейм и Character Controller сразу возвращает игрока в первоначальную позицию. Чтобы переместить игрока без всяких проблем был создан скрипт, который отключал контроллер, далее перемещал игрока и после заново включал контроллер, чтобы игрок мог двигаться.

Но кроме перемещения объекта скрипту необходимо расположить камеру игрока так, чтобы перемещение было почти незаметно. Тут также возникает проблема приоритетов. В FPS за движение камеры отвечает движение мыши и для камеры оно приоритетнее и не даёт дергать её из вне. И так, как здесь уже работает скрипт, а не элемент Unity, то просто на время его отключать нельзя, потому что при включении скрипт отменит предыдущие действия и не даст повернуть камеру. Поэтому в FPS этот метод был переписан так, что теперь к нему можно было бы обращаться из других скриптов и были добавлены новые переменные, которые отвечали за дополнительный поворот, который прибавлялся к повороту камеры, в обычной ситуации эти переменные равны 0, но в ситуации, когда этот метод насильно вызывается извне им можно было бы присудить новые значения.

После нескольких телепортаций и прохождения лабиринта, игрок подходит к двери на втором ярусе и переходит на следующую локацию.

3.7.3 Второй уровень

При переходе на второй уровень игрока встретит два разных размера коридора, которые также отличаются по размерам на своих концах и выходах. А чуть далее стена и посередине ещё один коридор (рисунок 26).

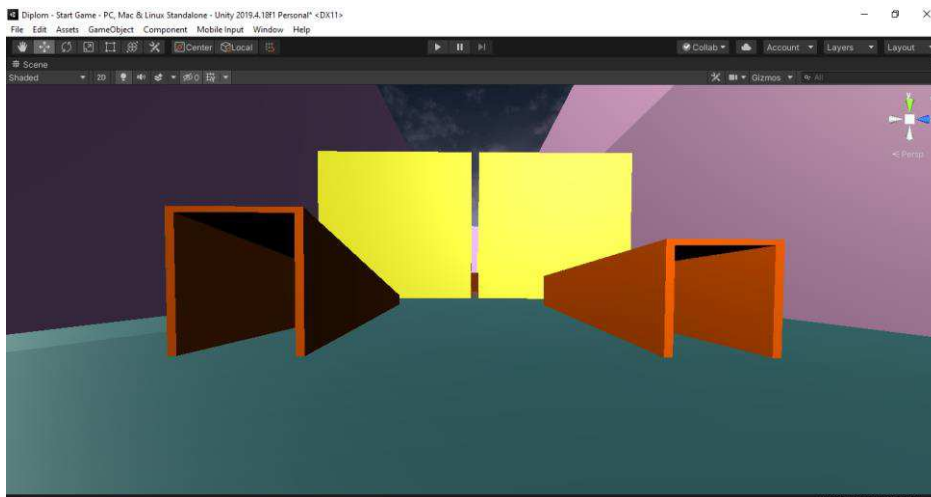


Рисунок 26 - Вторая локация с позиции игрока

При движении по такому коридору, размер игрока будет изменяться относительно разъема коридора, и получается так, что, войдя в большой разъем, игрок выходит в маленький разъем подстраиваясь под его размер.

Чтобы добраться до конца комнаты игроку нужно правильно пробраться сквозь эти коридоры к выходу с нужным размером, потому что коридоры так сделаны, чтоб войти в них можно было лишь имея подходящий размер персонажа, если он больше, то он не пролезет, а если он будет меньше, то из-за множителя в скрипте станет слишком маленьким, а дверь в конце комнаты откроется, только, если размер персонажа будет таким, какой он был в начале комнаты.

3.7.4 Третий уровень

На уровне расположено 5 рядов столбов по 4 штуки в каждом. Целью игрока является нахождение правильного маршрута среди столбов. На рисунке 27 можно увидеть, как все это выглядит.



Рисунок 27 - Начало уровня со столбами

В каждом ряду столбов расположены триггеры, которые отслеживают, между какими именно столбами игрок прошел. После прохождения ряда, коллайдеры во всем ряду блокируются и обратно уже вернуться нельзя, коллайдер показан на рисунке 28 посередине, слева располагается вся иерархия коллайдеров, а справа привязанный скрипт, который принимает ряд коллайдеров, переменную типа bool, которая говорит верный ли этот проход и свет, о котором далее.

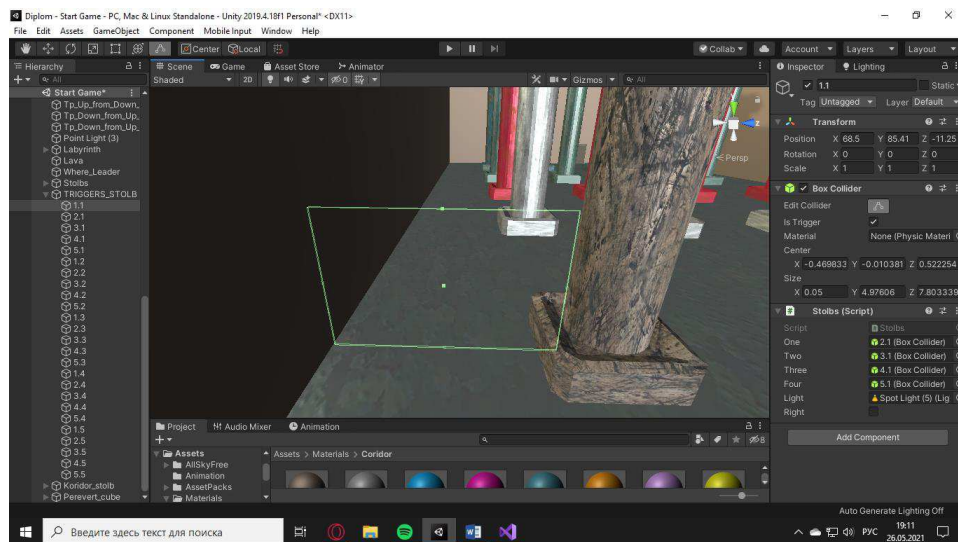


Рисунок 28 - Коллайдер между столбами

Так как количество вариантов прохода среди столбов равно 3125 , то в коридоре после столбов расположены 5 рамок с текстурами столбов и красным светом над ними, если пройдены коллайдер в ряду является верным, то свет меняет цвет на зеленый, когда все 5 рядов будут пройдены правильно, проход далее будет разблокирован, до этого же, при попытке покинуть уровень игрока будет телепортировать в начало. Коридор с подсказкой показан на рисунке 29.

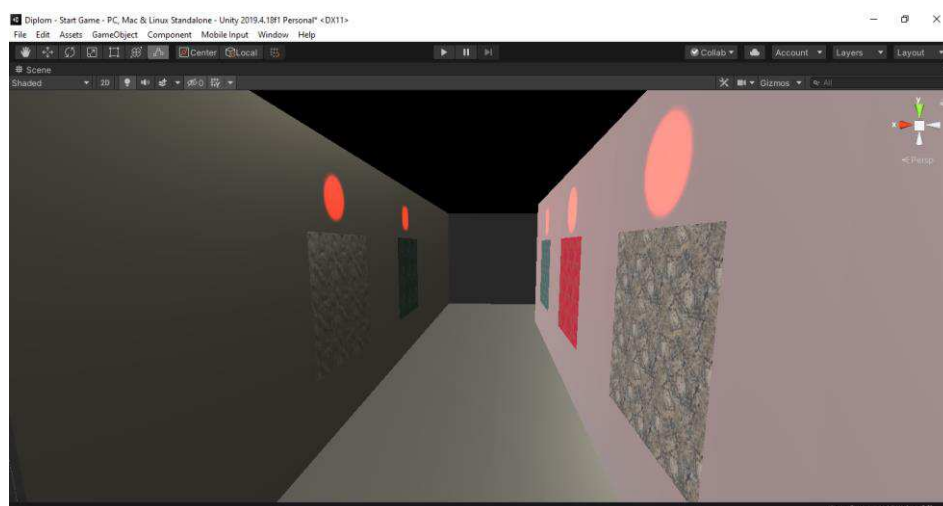


Рисунок 29 - 5 рамок с текстурами и светом

3.7.5 Закручивающийся коридор между уровнями

После прохождения уровня со столбами игрока ждет закручивающийся коридор, который не несет в себе загадок или паззлов. Это долгий коридор, который постоянно закручивается вправо, что может возникнуть ощущение, что игрок ходит по кругу, но в каждом углу коридора расположен разноцветный свет, говорящий, что это не одно кольцо. На рисунке 30 показано, как эти коридоры выглядят внутри.

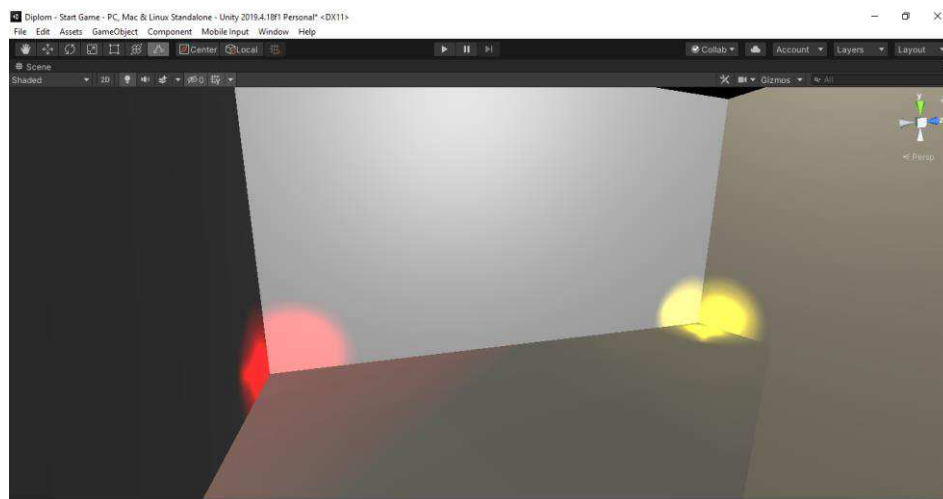


Рисунок 30 - Коридор внутри, каждый следующий угол имеет новый свет

В этих закручивающихся коридорах используется телепортация между кольцами с разным светом. Вид коридоров снаружи показан на рисунке 31.

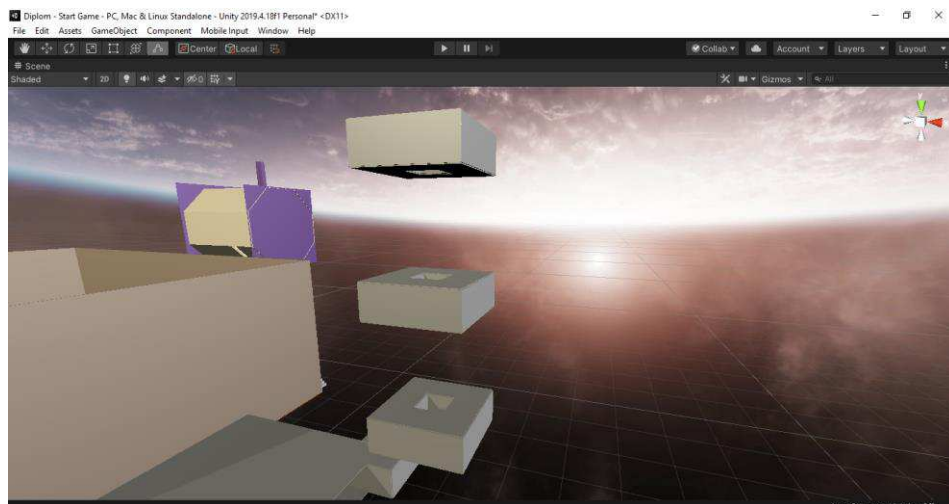


Рисунок 31 - Кольца коридоров

3.7.6 Четвертый уровень

В начале комнаты расположена лестница из 10 ступенек, когда игрок по ним поднимается, то перед ним располагается пустая комната. Суть пазла в том, что игрок должен пройти по невидимым плитам до другого конца комнаты. На рисунке 32 показан старт этого уровня.

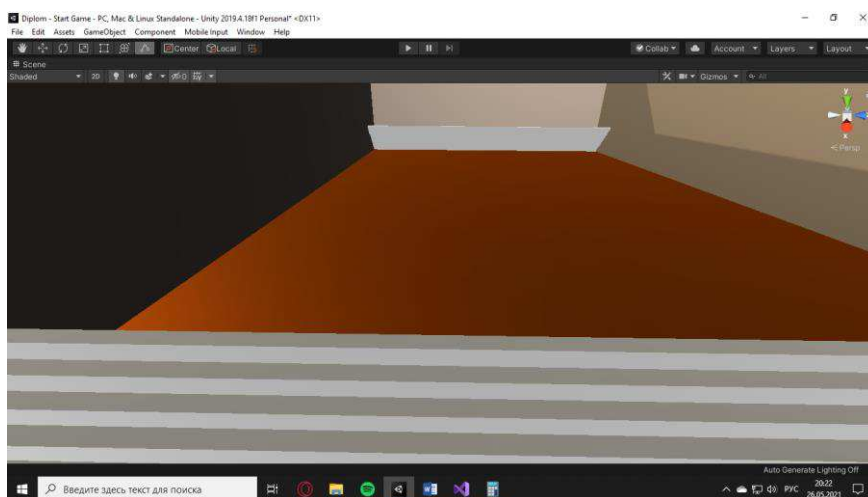


Рисунок 32 - Начало лестницы (в конце такая же лестница для спуска)

Среди плит расположено много дыр и если игрок провалится в них и коснется пола, то он будет телепортирован обратно в начало комнаты. Лабиринт выделен на рисунке 33 (в игре его игрок не увидит).



Рисунок 33 - Лабиринт, вид сверху

3.7.7 Пятый уровень

В следующий уровень персонажа уже телепортирует, потому что эта комната меняет свое положение. Это закрытый куб, у которого есть только один выход, но, чтобы в него выйти, сначала нужно открыть дверь и добраться туда. Здесь каждый угол имеет триггер переворота комнаты. Суть в том, что когда игрок подходит к стене, то происходит переворот комнаты на 90 градусов, а для игрока это выглядит так, как будто он пошел по стене. И такими переворотами, игрок должен загнать куб на кнопку, которая открывает дверь. Сам игрок куб в руки брать не может. Сама комната показана на рисунке 34.



Рисунок 34 - Комната 5 уровня

Сверху в углу комнаты спрятана кнопка, на которой надо расположить красный куб. И добраться до выхода так, чтоб куб оттуда не выпал.

3.7.8 Шестой уровень

Последняя комната с пазлом имеет форму восьмигранника и в ней с помощью кнопок со стрелками можно переворачивать всю комнату. Наверху комнаты есть заблокированная дверь. Её можно открыть, если правильно расположить кубы на кнопках в соответствии с цветами. В этой комнате в отличии от предыдущей, у нас есть возможность брать кубы в руки. Комната показана на рисунке 35.

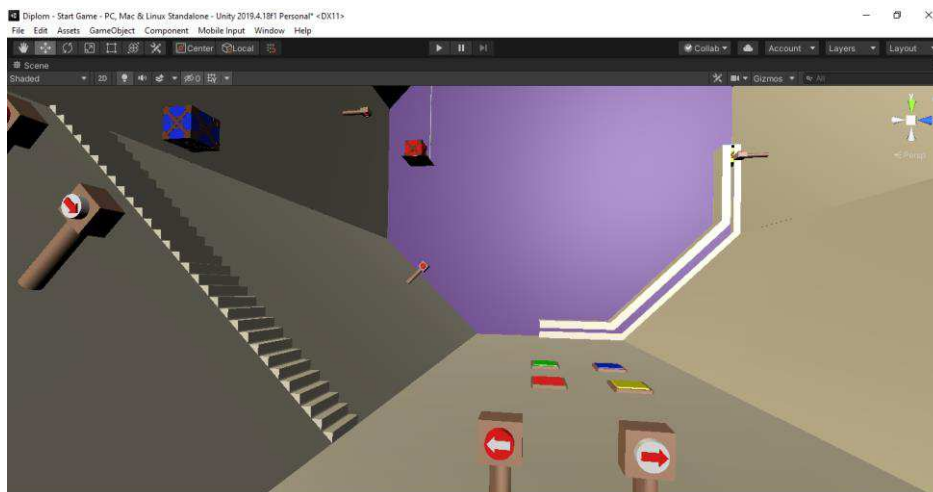


Рисунок 35 - Комната в форме восьмигранника

Кроме того, что комната переворачивается по нажатию кнопок со стрелками, её особенность заключается в том, что кубы имеют переключатель гравитации и первоначально гравитация у них отсутствует, но когда игрок берет куб в руки и затем отпускает, то у куба включается гравитация и он падает на землю. Но если нажать кнопки со стрелками и перевернуть комнату, то у кубов отключится гравитация и они останутся на стене. Эту механику игрок должен использовать, чтобы достать все кубы.

После того, как игрок расположит все кубы на кнопках, дверь на потолке откроется и выйдя в неё игра закончится.

3.7.9 Меню игры

В игре также помимо уровней были реализовано главное меню и пауза во время игры. Их отличие заключается в том, что меню игры расположено на другой сцене, то есть все объекты из основной игры начнут подгружаться только тогда, когда будет нажата кнопка Start game. В этот момент игрока перенесет на новую сцену, где уже начнется сам игровой процесс. Главное меню показано на рисунке 36.

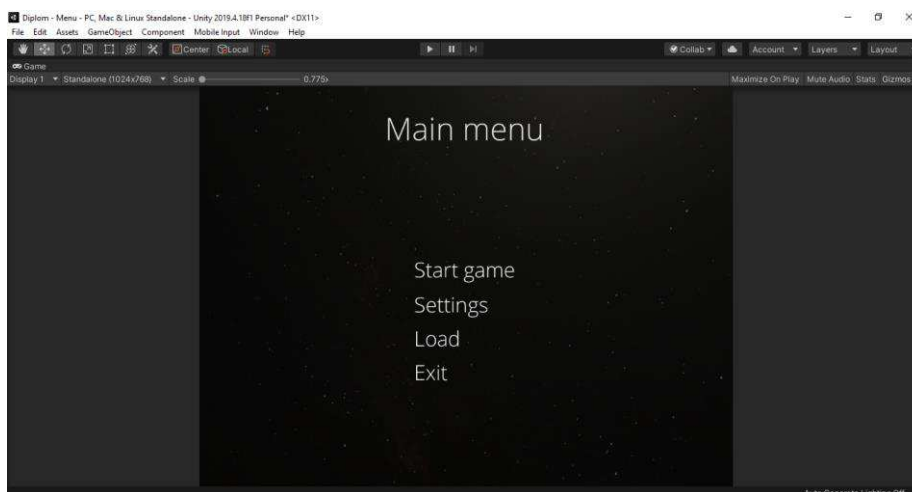


Рисунок 36 - Главное меню игры

Кнопка Setting переключает нас на другую панель меню, данная панель показана на рисунке 37.

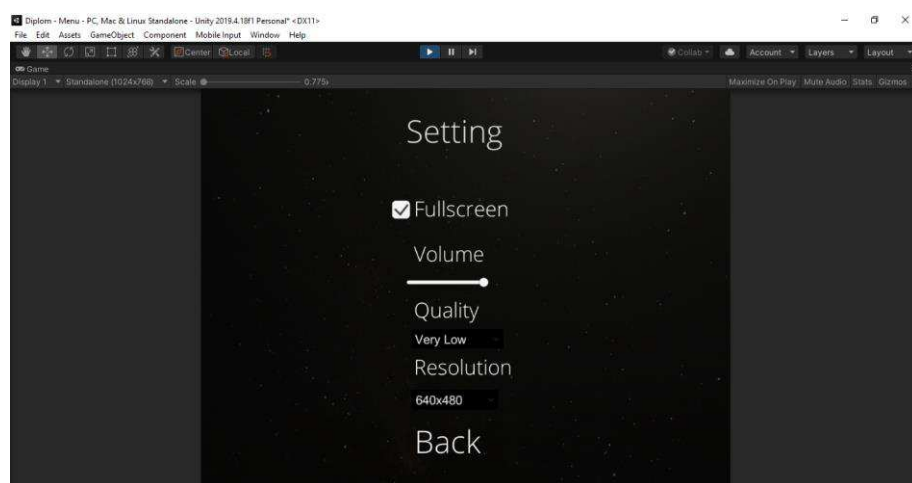


Рисунок 37 - Игровые настройки

В настройках мы можем развернуть игру на весь экран для этого у класса Screen отвечающего за экран есть переменная fullscreen, у которой меняется значения в зависимости от того стоит ли галочка в меню. Слайдер отвечает за громкость музыки, в Unity есть свой AudioManager, который отвечает за внутри игровой звук, слайдер обращается к миксеру и меняет звук. Качество

картинки отвечает за прорисовку текстур и теней, так как в игре нет обильного количества сложных текстур и теней, то даже на слабых компьютерах с видеокартами прошлых поколений все будет работать без зависаний на высоких настройках. Разрешение отвечает за растягивание картинка на монитор, в Unity учтены все возможные разрешения под все мониторы, которые существуют.

Кнопка Load отвечает за загрузку в игре, в игре нельзя сохраняться в любой момент. Автосохранения происходят только когда игрок прошел уровень.

Сохранения реализованы с помощью JSON файлов в которые фиксируется новая позиция игрока в начале уровня до которого он дошел и поворот его камеры, чтобы он не был сбит. При загрузке данные из файла присваиваются FPS и игрок начинает с уровня на котором он закончил прошлое прохождение.

Кнопка выхода сохраняет данные о том до какого уровня он дошел и закрывает игру.

Если во время игры нажать клавишу Escape, то все в игровом мире будет заморожено, экран затемнится, управление игроком будет отключено и перед камерой игрока появится меню (рисунок 38).

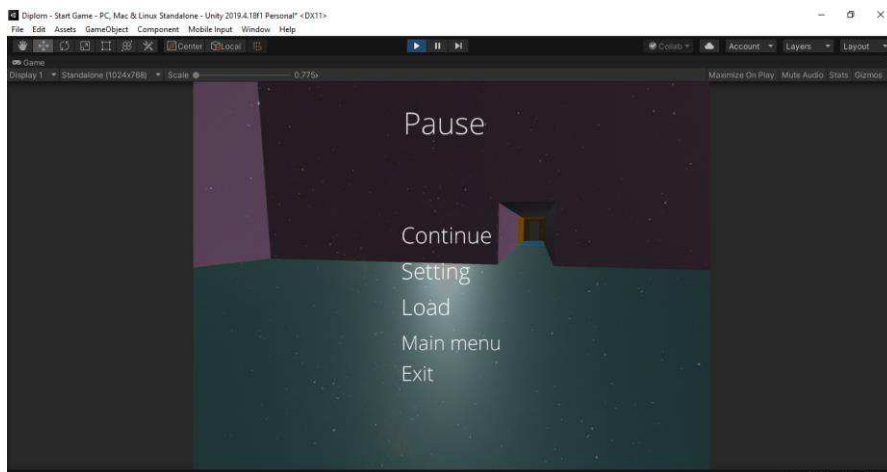


Рисунок 38 - Меню во время паузы

Функционал кнопок не изменен, за исключением появления новой кнопки Main menu, которая отвечает за выход в главное меню.

Чтобы реализовать меню паузы время в игре останавливается, а точнее скорость игры становится равна 0. Контроллер игрока отключается, и игрок не может двигаться во время паузы. У камеры значение скоростей мыши меняется на 0, так как мы не можем отключить саму камеру, ведь это приведет к тому, что экран станет черным.

ЗАКЛЮЧЕНИЕ

Видеоигры за последние десятки лет заработали большую аудиторию поклонников. Из небольшого развлечения они разрослись в целую игровую индустрию.

Разработчики придумывают все новые и новые идеи, чтобы удивить требовательных игроков. Одной из таких идей являются игры с неевклидовыми мирами.

В рамках выпускной квалификационной работы был исследован жанр игр головоломок с Неевклидовыми мирами, а также разработана игра в данном жанре.

Во время производства игры было сделано 38 моделей из них 8 моделей комнат и 30 моделей предметов, написано 2 музыкальных мелодии, нарисовано 4 текстуры, написано 15 скриптов. Во время разработки были задействованы программы: Unity, Blender, Visual Studio, Gimp, Fl Studio.

Результатом работы является игра в жанре головоломка с Неевклидовым пространством произведенная на движке Unity.

СПИСОК СОКРАЩЕНИЙ

ПК – персональный компьютер

DL – Direction Light

FPC – First Person Controller

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИК

- 1 Гейг М. Разработка игр на Unity 2018 за 24 часа: пер. с англ. Райтмана М. А. – Москва: Эксмо, 2020. – 464 с.: ил.
- 2 Дикинсон К. Оптимизация игр в Unity 5: пер. с англ. Рагимова Р. Н. – Москва: ДМК Пресс, 2017. – 306 с.: ил.
- 3 Паласиос Х. Unity 5.x. Программирование искусственного интеллекта в играх: пер. с англ. Р. Н. Рагимова. – Москва: ДМК Пресс, 2017. – 272 с.: ил.
- 4 Петелин Р. Ю., Петелин Ю. В. Fruity Loops Studio / Р. Ю. Петелин, Ю. В. Петелин – Санкт-Петербург: БХВ-Петербург, 2009. – 480 с.
- 5 Прахов А. А. Самоучитель Blender 2.6. / А. А. Прахов – Санкт-Петербург: БХВ-Петербург, 2013. – 384 с.: ил.
- 6 Торн А. Искусство создания сценариев в Unity / пер. с англ. Р. Н. Рагимова. – Москва: ДМК Пресс, 2016. – 360 с.: ил.
- 7 Торн А. Основы анимации Unity / пер. с англ. Р. Н. Рагимова. – Москва: ДМК Пресс, 2016. – 176 с.: ил.
- 8 Хахаев И. А. Графический редактор GIMP: первые шаги / И. А. Хахаев – Москва: ALT Linux; Издательский дом ДМК-пресс, 2009. – 232 с.: ил.
- 9 Шрейер Д. Кровь, пот и пиксели. Обратная сторона индустрии видеоигр / пер. с англ. Л. И. Степанова. – Москва: Издательство «Эксмо», 2018. – 368 с.: ил.
- 10 СТО 4.2-07-2014 Общие требования к построению, изложению и оформлению документов учебной деятельности: [Электронный ресурс]. URL: <http://about.sfu-kras.ru/docs/8127/pdf/248160>
- 11 Unity, платформа разработки в реальном времени: [Электронный ресурс]. URL: unity.com/ru
- 12 Unity – Руководство: Руководство Unity: [Электронный ресурс]. URL: <https://docs.unity3d.com/ru/530/Manual/UnityManual.html>

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
институт

Информационные системы
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС


подпись




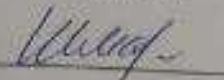
П. П. Дьячук
инициалы, фамилия

«08» 06 2021 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 – «Информационные системы и технологии»

Разработка компьютерной игры с Неевклидовой геометрией на платформе
Unity

Руководитель	 подпись, дата	08.06.21 ст. преподаватель	<u>Ю. В. Шмагрис</u> инициалы, фамилия
Консультант	 подпись, дата	08.06.21 доцент, к. п. н.	<u>С. А. Виденин</u> инициалы, фамилия
Выпускник	 подпись, дата	08.06.21	<u>А. И. Шеенко</u> инициалы, фамилия
Нормоконтролер	 подпись, дата	08.06.21 ст. преподаватель	<u>Ю. В. Шмагрис</u> инициалы, фамилия

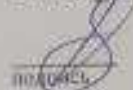
Красноярск 2021

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
ииспу

Информационные системы
ииспу

УТВЕРЖДАЮ

Заведующий кафедрой ИС


подпись

П. П. Дьячук
кандидат, филолог

«08» 06 2021 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

Студенту Шеенко Александру Ивановичу

Группа: Ки17-14Б Направление: 09.03.02 – «Информационные системы и технологии»

Тема выпускной квалификационной работы: «Разработка компьютерной игры с Неевклидовой геометрией на платформе Unity»

Утверждена приказом по университету № 1834/с от 03.04.2021

Руководитель ВКР: Ю. В. Шмагрис, старший преподаватель кафедры «Информационные системы» ИКИТ СФУ.

Консультант ВКР: С. А. Виденин, доцент кафедры «Информационные системы» ИКИТ СФУ, кандидат педагогических наук.

Исходные данные для ВКР: Требования к разрабатываемой системе, методические указания научного руководителя, учебные пособия.

Перечень разделов для ВКР: Введение, теоретическая часть, реализация игры, заключение, список использованных источников.

Перечень графического материала: Презентация, выполненная в Microsoft Office PowerPoint 2016.

Руководитель ВКР

Шмагрис

подпись, дата

ст. преподаватель

Ю. В. Шмагрис

инициалы, фамилия

Выпускник

Шеенко

подпись, дата

А. И. Шеенко

инициалы, фамилия

«08» 06 2021