

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

*Факультет Информационных технологий
Кафедра Информатики и информационных технологий*

направление подготовки

09.03.02 «Информационные системы и технологии»

КУРСОВОЙ ПРОЕКТ

Дисциплина: Разработка онлайн игр

Тема: Разработка онлайн игры

Выполнил(а): студент(ка) группы 201-726

Курносова Арсения Валерьевна

(Фамилия И.О.)

Дата, подпись 20.03.2024

(Дата)

(Подпись)

Проверил: _____

(Фамилия И.О., степень, звание)

Дата, подпись _____

(Дата)

(Подпись)

Замечания: _____

Москва

2022

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

*Факультет Информационных технологий
Кафедра Информатики и информационных технологий*

направление подготовки 09.03.02 «Информационные системы и технологии»,
профиль «Программное обеспечение игровой компьютерной индустрии»

УТВЕРЖДАЮ

Зав. каф., к.т.н. Е.В. Булатников

«__» _____ 2023 г. _____
(Подпись)

ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ

Студент Курносова Арсения Валерьевна группа **201-726**

(Фамилия, Имя, Отчество)

Тема Разработка онлайн игры

1. Срок представления работы к защите «__» июня 2023 г.

2. Исходные данные для выполнения работы:

исходные материалы

3. Содержание курсового проекта: введение, аналитическая часть, практическая часть, заключение, библиографический список

4. Перечень графического материала рисунки

5. Литература и прочие материалы, рекомендуемые студенту для изучения:
указаны в библиографическом списке


6. Дата выдачи задания: «28» января 2024 г.

7. Руководитель:

_____ / Булатников Е. В. /

(Подпись)

8. Задание к исполнению принял:

Курносова А.В. / ФИО / 

Оглавление

Введение	4
ГЛАВА 1. Аналитика	4
Цель.....	4
Техническое задание	4
Описание предметной области.....	4
Выбор инструментов.....	5
ГЛАВА 2. Разработка онлайн игры с использованием примитивов	6
Разработка игрового пространства	6
Разработка конструктора игрока.....	7
Разработка игрового лобби.....	10
Разработка окна подключения	11
Оценка работы проекта на примитивах	12
Вывод этапа разработки системы подключения	14
ГЛАВА 3. Добавление игрового окружения.....	14
Выбор элементов окружения.....	14
Создание моделей окружения	14
Создание модели игрока	17
Добавление анимированной модели персонажа в игровой проект	18
Добавление геймплейных элементов	21
Заключение.....	23
Список литературы.....	24

Введение

Цель проекта – разработка онлайн игры, с поддержкой единовременного подключения от пяти пользователей. Игровой проект должен быть функциональным с программной точки зрения и иметь мультимедийное наполнение. Конечным продуктом является многопользовательская игра, поддерживающее единовременное подключение пяти игроков и более, содержащая трёхмерные модели окружения и игровых персонажей.

ГЛАВА 1. Аналитика

Цель

Разработка онлайн игры с мультимедийным наполнением.

Техническое задание

Разработать онлайн игру, соответствующую следующим требованиям:

- Поддержка единовременного подключения пяти игроков и более;
- Возможность подключения игроков с разных устройств;
- Возможность подключения игроков, не находящихся в единой сети Интернет;
- Содержание в игре осмысленного игрового окружения – трёхмерных моделей;
- Содержание в игре моделей игровых персонажей;
- Наличие соответствующих анимаций у игровых персонажей.

Описание предметной области

Многопользовательские онлайн игры подразумевают возможность единовременного подключения нескольких игроков к единому игровому пространству.

В игровом пространстве может происходить различное взаимодействие между игроками, в зависимости от геймплея. В многопользовательских онлайн

играх могут присутствовать как соревновательные элементы между игроками, так и элементы кооперации с целью выполнения единой задачи.

Выбор инструментов

Для разработки игрового проекта был выбран игровой движок Unity. Для игрового движка Unity существует дополнительный пакет Photon, который позволяет разрабатывать онлайн игры с использованием уже готовых материалов для подключения и синхронизации.

Для установки Photon и добавления пакетов в Unity, был использован официальный сайт Photon.

Бесплатная версия Photon поддерживает до 12 подключений онлайн. По умолчанию, подключение происходит по сети Интернет.

Использование Photon в проекте обеспечивает:

- Создание игрового лобби с вводом необходимых данных;
- Создание игрового пространства (игрового мира) в сети;
- Подключение игроков к игровому миру по сети;
- Отображение игроков в игровом мире (с учётом указанных имён);

Синхронизацию между игроками в игровом пространстве.

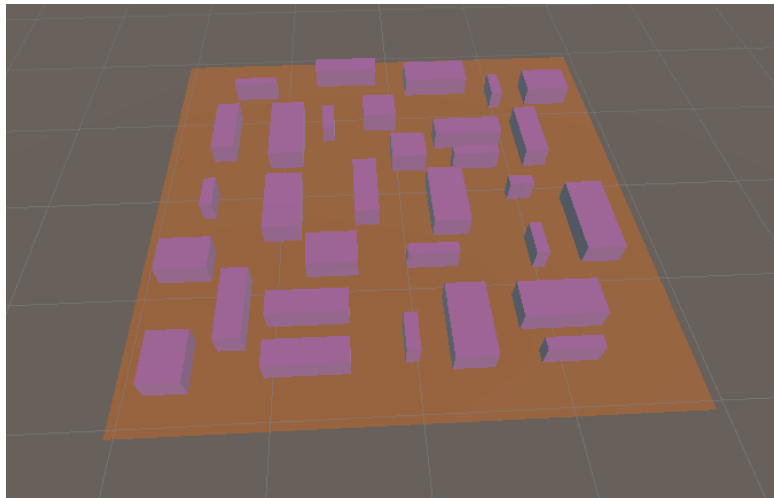
ГЛАВА 2. Разработка онлайн игры с использованием примитивов

Разработка игрового пространства

Для подключения игроков к игровому пространству, необходима предварительная организация игрового мира. Для организации игрового мира были использованы примитивы:

- Plane (в качестве основы игрового мира);
- Cube (в качестве препятствий и элементов окружения);
- Sphere (в качестве искомого объекта).

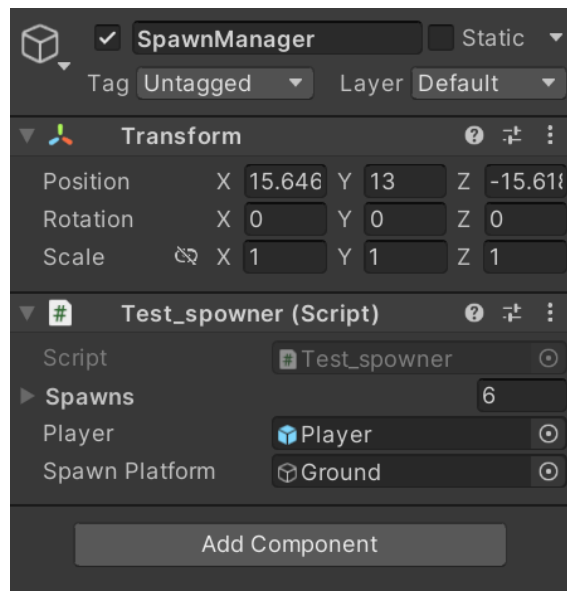
При помощи примитивов была собрана игровая карта.



Игровая карта из примитивов

Для организации дальнейшего появления игроков в игровом пространстве, на игровую карту была добавлена точка появления (Spawn Point).

Для организации точки появления игроков, был использован менеджер появления игроков на карте (Spawn Manager). Для его организации был добавлен скрипт появления игроков на карте. Скрипт определяет создание нового игрока в пределах существования игрового пространства.



Организация появления игроков в игровом мире

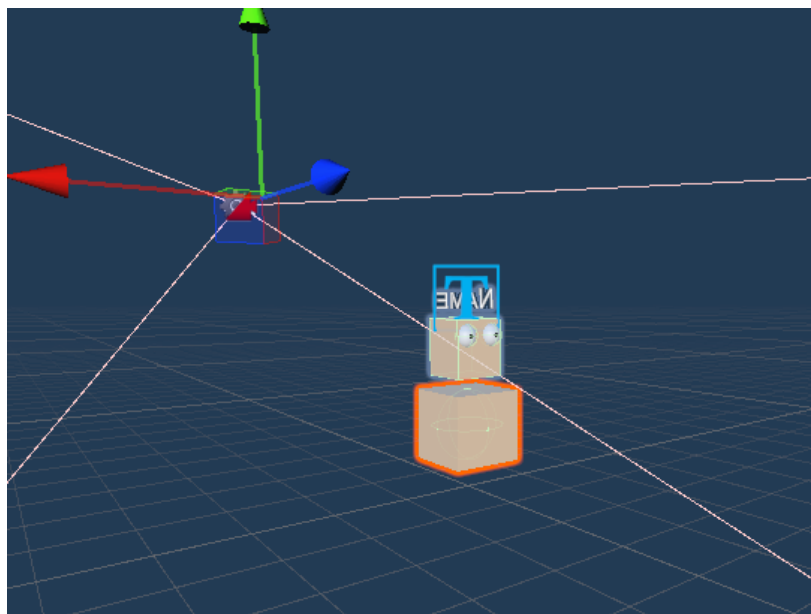
Разработка конструктора игрока

Для создания игроков в игровом пространстве используется модель игрока. Для каждого подключающегося клиента создаётся одинаковая модель (копия) игрока. Каждая «копия» имеет уникальное имя в игровом мире.

Модель игрока перенесена в prefab. Это позволяет создавать множество одинаковых по содержанию объектов, без создания каждого из этих объектов индивидуально. Изменение в модели игрока вносит изменения в каждую копию игрока в игровом пространстве.

Модель игрока содержит:

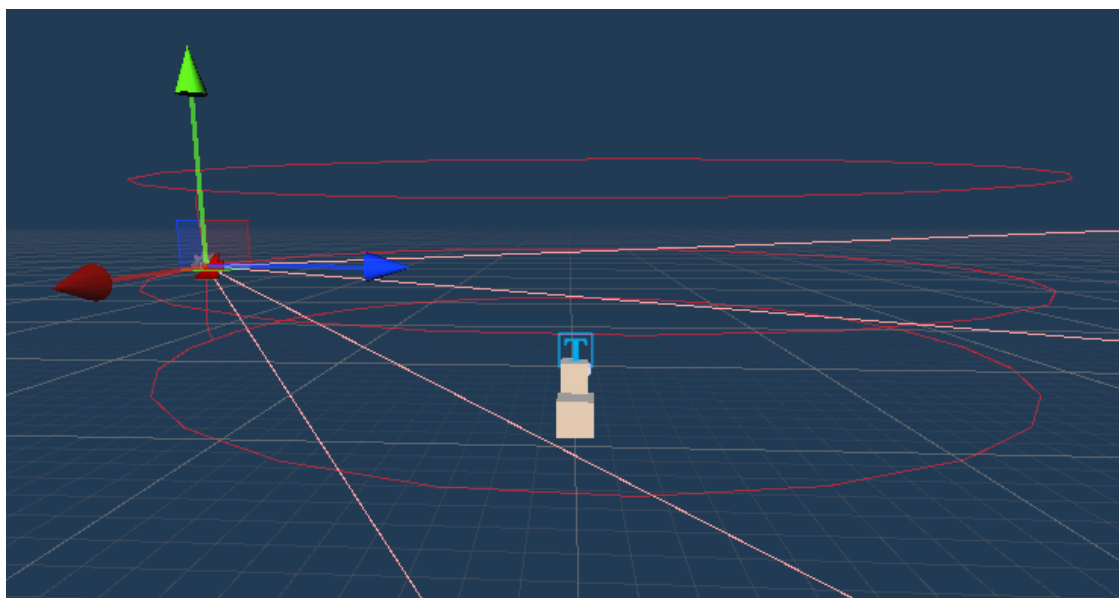
- Тело игрока;
- Камеру игрока;
- Скрипты для управления игроком (прикреплены к телу игрока);
- Поле отображения имени игрока.



Модель игрока

Тело игрока собрано из примитивов.

Для управления камерой задействованы гравная камера (MainCamera) и CinemachineBrain (из пакета Cinemachine, встроенного в Unity). Это позволяет перемещать камеру по окружности, вокруг игрока.



Камера для управления игроком

Для управления моделью игрока используется скрипт передвижения игрока в игровом пространстве и скрипт управления камерой игрока.

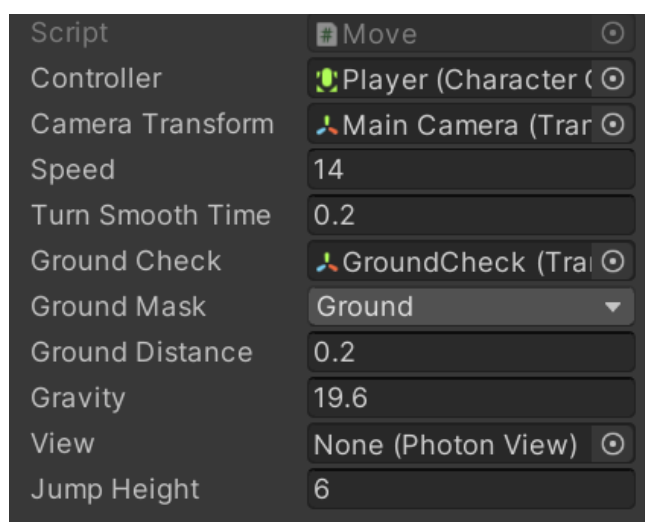
Скрипт управления передвижением игрока содержит:

- Передвижение по плоскости (вперёд, назад, вправо, влево);

- Прыжок (передвижение по вертикали).

Управление игроком осуществляется при помощи клавиш WASD. При нажатии на соответствующую клавишу происходит изменение текущего положения игрока в указанном направлении с учётом заданного параметра скорости передвижения игрока.

Прыжок осуществляется при нажатии на клавишу «пробел». При нажатии на клавишу происходит перемещение игрока по вертикали с учётом заданного параметра высоты прыжка. Скорость падения игрока после прыжка определяется параметром гравитации, действующей на игрока. Осуществление прыжка возможно только если игрок находится на поверхности игрового мира (Plane). Перед каждым прыжком осуществляется проверка нахождения игрока на поверхности.



Параметры управления игроком

Поле отображения имени игрока получает значение из поля имени игрока в лобби игры. Имя игрока отображается над игроком. Для отображения актуального имени игрока, используется скрипт, позволяющий получить данные из строки имени игрока в лобби игры.

Для корректного перемещения копий игрока в игровом пространстве используется Photon Transform View. Это позволяет корректно отображать перемещение персонажа в игровом пространстве. Поскольку каждый клиент управляет своей копией глобальной модели персонажа, отсутствие этой функции

на персонаже может приводить к некорректной передаче прав на управление персонажем между клиентами.

Для синхронизации действий игроков используется Photon View. Это позволяет синхронизировать действия различных клиентов в одном игровом мире. Без этой функции, один клиент может не видеть действий другого клиента в игровом мире.

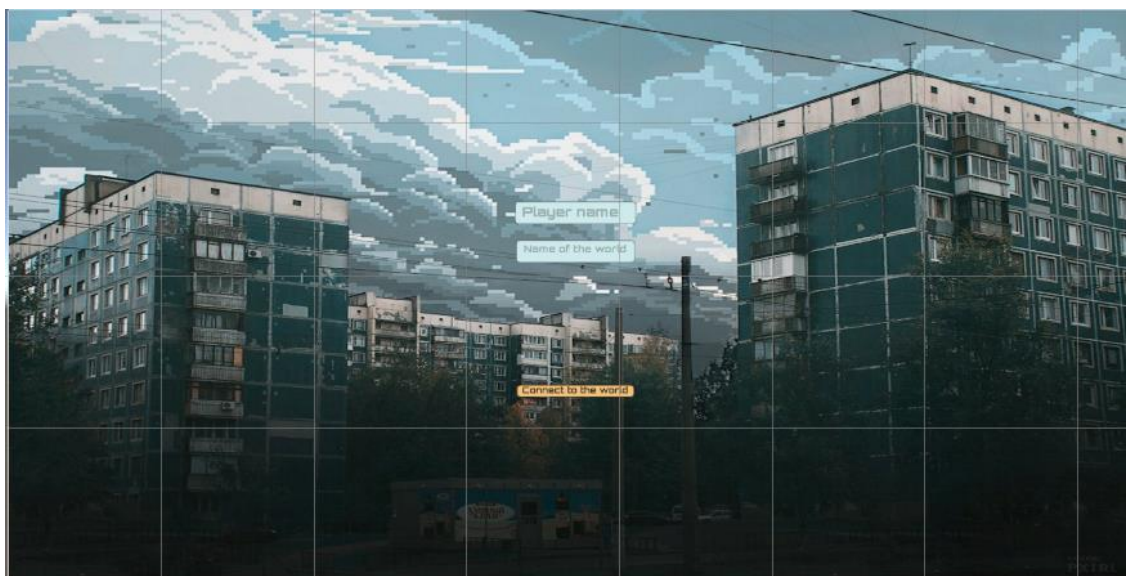
Разработка игрового лобби

Для корректного подключения игроков к игровому пространству требуется:

- Имя игрока (задействуется для отображения имени на сервере и для программного различия игроков для сервера);
- Название игрового мира (название мира, к которому может подключиться игрок, для создания игрового пространства в сети).

Лобби содержит следующие функциональные элементы:

- Поле для ввода имени игрока;
- Поле для ввода названия мира;
- Кнопка создания и подключения к игровому миру;
- Кнопка подключения к игровому миру.



Игровое лобби

Поля для ввода предназначены для получения информации от клиента. Данные, полученные из поля ввода имени игрока, используются для отображения игрового имени клиента в игровом пространстве (над персонажем клиента).

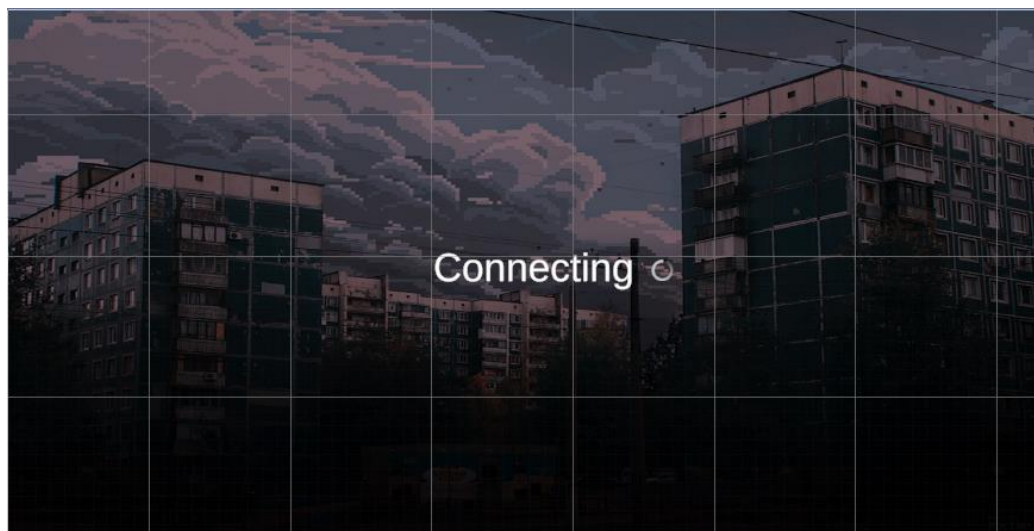
Данные, полученные из поля ввода названия игрового мира, используются для создания игрового пространства. Для игроков, не являющихся хостом, подключение к игровому миру возможно по указанному названию.

Кнопка создания и подключения к миру определяет клиента, который является хостом игрового пространства. Первый игрок создаёт новый игровой мир, остальные игроки могут подключаться к уже созданному игровому миру по его названию. Другие игроки так же могут создавать свой мир, с другим названием.

Кнопка подключения к игровому миру определяет наличие игрового пространства с указанным названием. Если игровой мир с указанным названием существует – происходит подключение. Если мир с указанным названием не существует – требуется сначала создать мир с данным названием.

Разработка окна подключения

Перед входом в игру необходимо наличие подключения к сети Интернет. Перед входом в игру, происходит проверка наличия подключения у клиента. Если подключение есть – происходит переход к игровому лобби. Если подключение отсутствует, клиент информируется о необходимости подключения к сети.



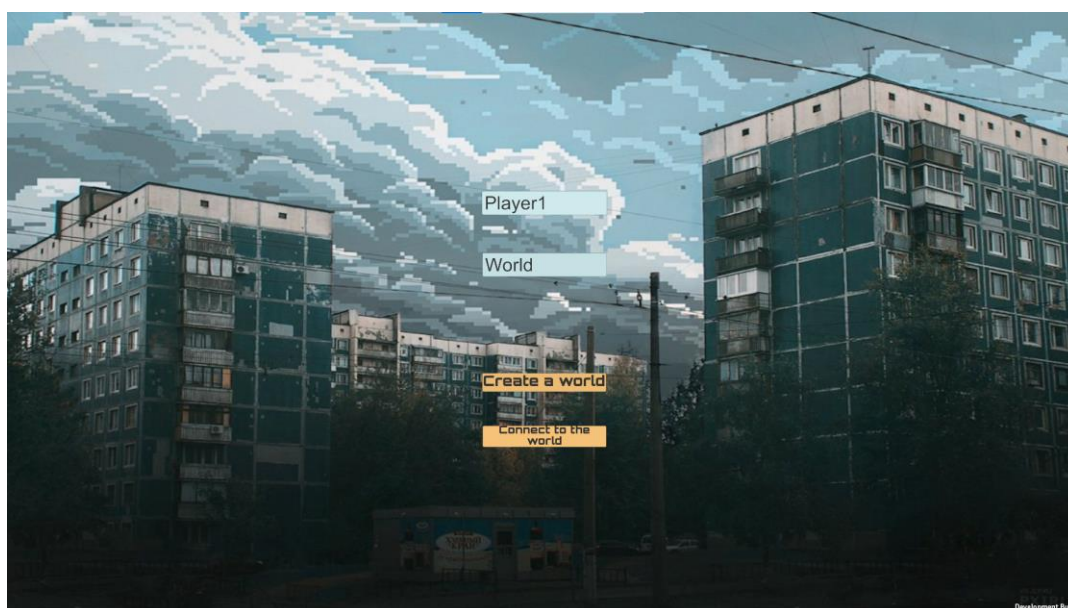
Окно проверки подключения к сети

Оценка работы проекта на примитивах

Для проверки работы игрового проекта, было произведено подключение игроков, находящихся в сети, к одному игровому миру.

Каждый игрок является самостоятельным клиентом. Запуск игры каждым клиентом осуществляется через игровой файл проекта.

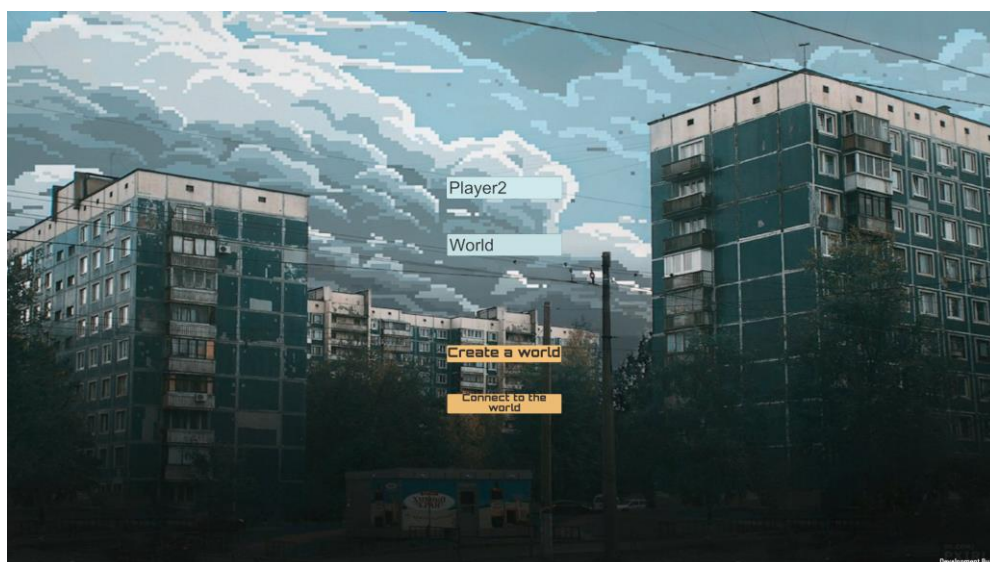
При входе каждый игрок указывает имя своего игрока. Первый клиент осуществляет вход в игру через создание игрового мира с указанным названием. Остальные клиенты осуществляют вход через указание мира с таким же названием и подключение к уже созданному миру.



Создание мира первым клиентом



Подключение первого клиента к игровому миру



Подключение второго и последующих клиентов к ранее созданному игровому миру



Отображение второго и последующих клиентов в игровом мире

Вывод этапа разработки системы подключения

В ходе работы над проектом был разработан игровой мир, состоящий из примитивов. Было создано онлайн подключение игроков к игровому миру, их корректное отображение в игровом пространстве, передача прав управления персонажами клиентам и синхронизация игроков.

ГЛАВА 3. Добавление игрового окружения

Выбор элементов окружения

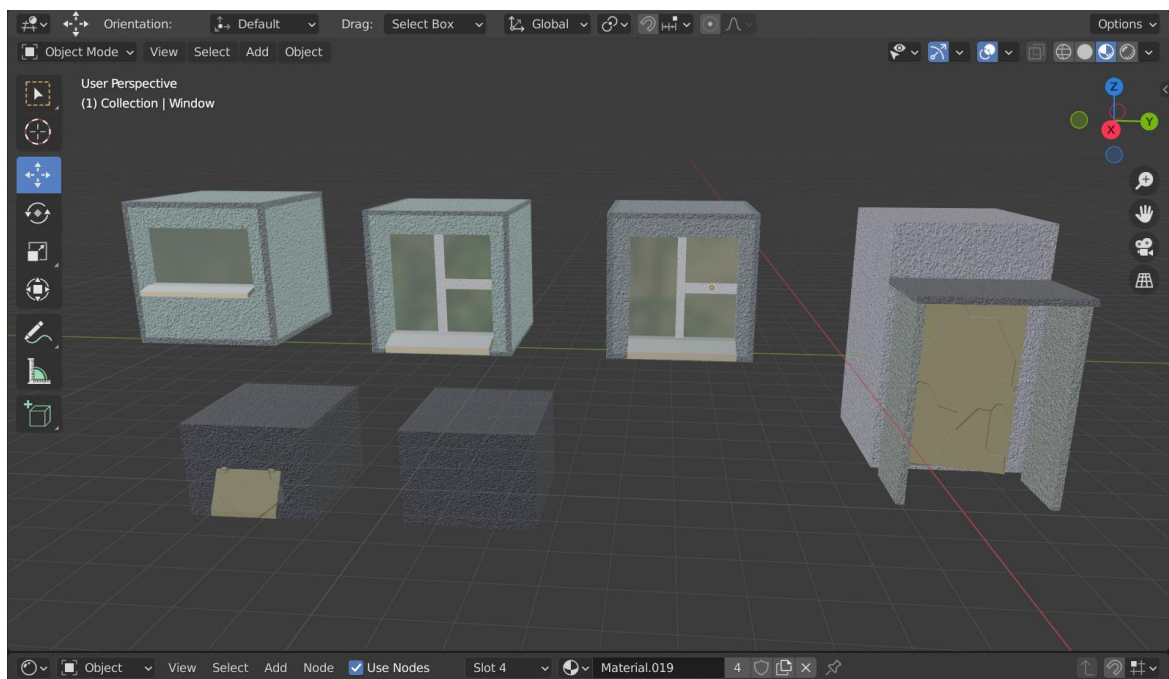
При создании игрового окружения, за основу была взята атмосфера постсоветского пространства. В качестве элементов окружения выступают панельные пятиэтажные и трёхэтажные дома. Также, для создания пространства дворов, задействуются модели деревьев.

При создании игрового окружения используются приглушённые цвета. Свет на карте также приглушён, для создания атмосферы вечера. Для текстуры неба выбраны более тёмные цвета.

Создание моделей окружения

Основными моделями в игровом окружении выступают модели пятиэтажных и трёхэтажных панельных домов.

При создании моделей, для оптимизации процесса разработки моделей и возможности вариативной сборки различных моделей домов, был разработан «конструктор», состоящий из нескольких элементов, позволяющих собрать целостную модель панельного дома.

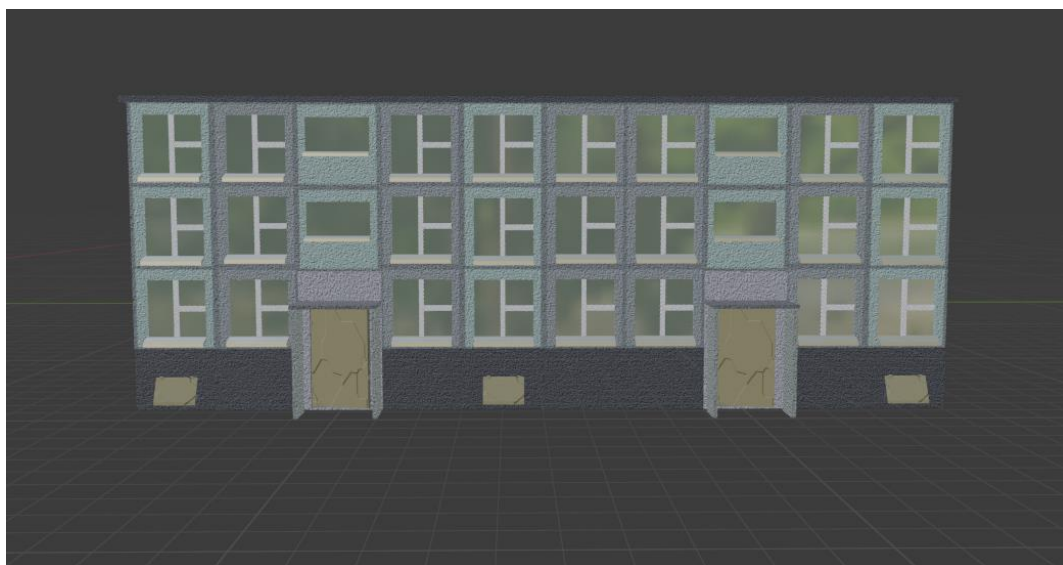


Конструктор для создания моделей домов

Такой конструктор позволяет создавать любые вариации домов, соответствующих указанным требованиям. При помощи конструктора может быть собрана целостная модель панельного дома на любое количество этажей, любой длины, с любым количеством подъездов и окон.

При создании моделей были задействованы методы скульптинга, добавления и настройки текстур.

При помощи конструктора были собраны целостные модели панельных домов на пять и три этажа.



Модель трёхэтажного дома



Модель пятиэтажного дома

При создании целостных моделей домов, детали конструктора были размножены и состыкованы друг с другом в удобном порядке. Некоторые из деталей конструктора были отзеркалены, для создания аналогичных деталей конструктора, применимых для другой стороны целостной модели дома.

Для создания моделей деревьев был использован инструмент построения полигонов вокруг направляющего ребра. Также был задействован инструмент, позволяющий создавать случайные формы для фигур в Blender.



Модель дерева

После создания в Blender, модели окружения были импортированы в проект Unity и добавлены на сцену. При добавлении на сцену, каждая модель была преобразована в префаб – шаблонный элемент проекта. В префабе каждой модели, был добавлен коллайдер, для создания физической оболочки объекта в игровом мире.

Создание модели игрока

При создании модели игрока были задействованы методы скульптинга. Модель была создана путём добавления примитивов на сцену разработки Blender, увеличения числа полигонов примитивов и скульптурирования полученных высокополигональных объектов.



Модель персонажа

После создания высокополигональной модели, была выполнена ретопология модели, что позволило создать аналогичную модель с более низким числом полигонов.



Ретопология модели персонажа

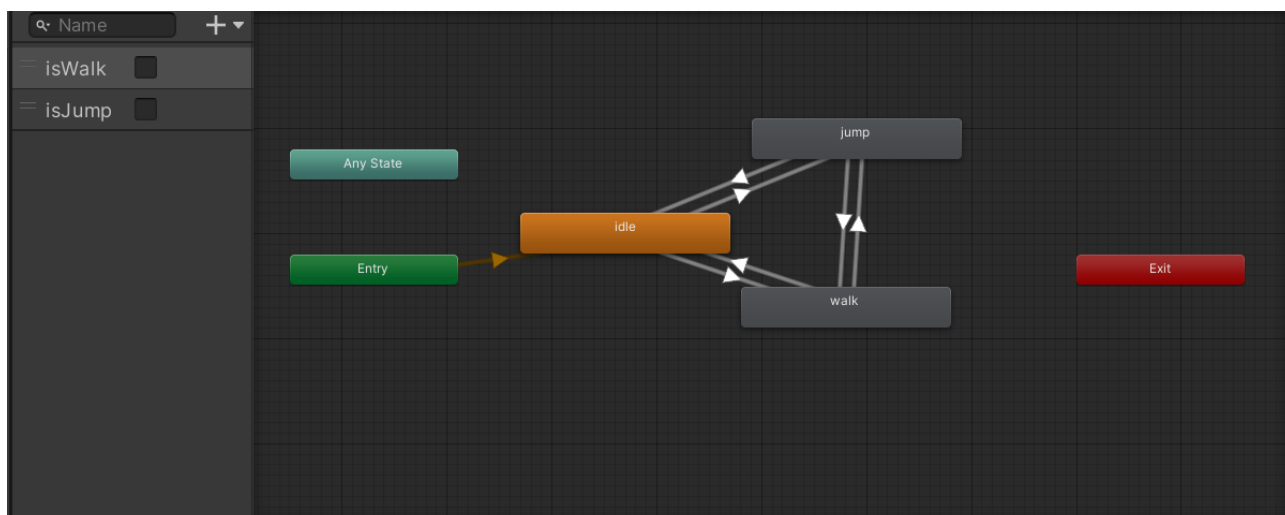
На модели также были запечены текстуры с высокополигональной модели.

После этого к модели были применены анимации.

Добавление анимированной модели персонажа в игровой проект

Для добавления модели персонажа в игровой проект, необходим не только импорт самой модели, но и настройка её анимаций и привязка исполнения анимаций к коду управления персонажем.

Для добавления самих анимаций к персонажу, был создан узел аниматора. В аниматор были добавлены все виды анимации персонажа: ходьба, прыжок и стояние на месте. Также, были настроены переходы между анимациями.

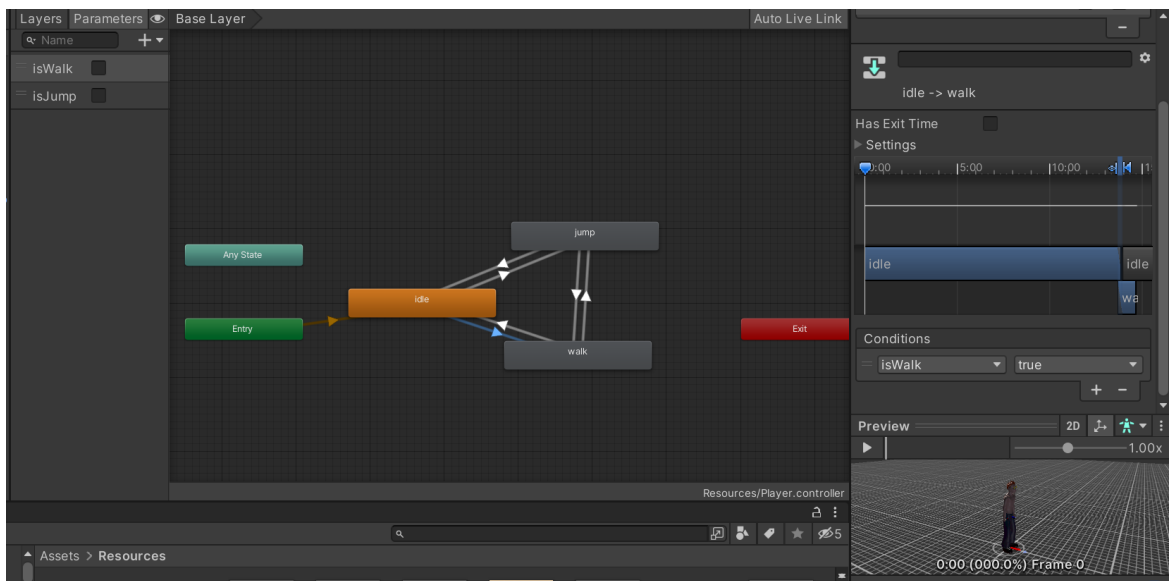


Настройки аниматора

Анимация стояния на месте является начальной, поскольку в этом положении персонаж находится при отсутствии команд от игрока.

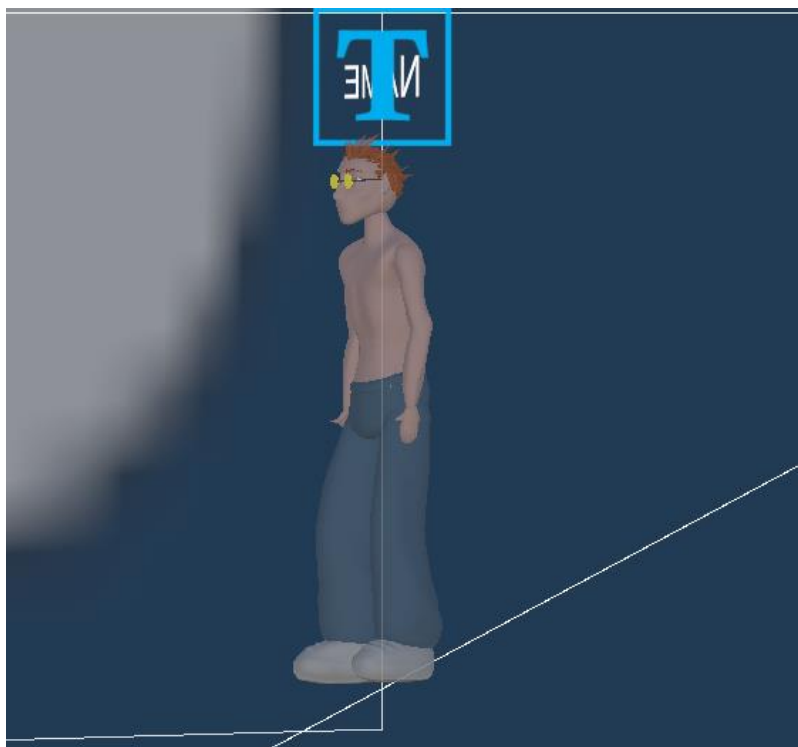
Переходы в обе стороны существуют между всеми анимациями, так как команды от игрока могут поступать в любом порядке.

Для переходов между анимациями были добавлены элементы типа bool, позволяющие управлять переходами анимации внутри кода. Для каждого перехода анимации был настроен свой параметр перехода.



Настройка параметров перехода анимаций

После настройки анимаций, модель игрока была добавлена в префаб игрока, заменив ранее стоящий набор примитивов. На префаб игрока был добавлен аниматор.



Префаб персонажа

После настройки анимаций в узле аниматора и добавления узла на префаб персонажа, переключение анимаций было добавлено в код. Так, при выполнении функции ходьбы, анимация ходьбы принимает положение «true», что приводит к её включению. При остановке игрока, функция ходьбы перестаёт выполняться и переключатель принимает положение «false», что приводит к обратной смене анимации на анимацию стояния на месте.

```
if (direction.magnitude >= 0.1f)
{
    anim.SetBool("isWalk", true);

    float targetAngle = Mathf.Atan2(
    float angle = Mathf.SmoothDampAr

    transform.rotation = Quaternion.

    Vector3 moveDirection = Quaternion
    _controller.Move(moveDirection.r
}
else
{
    anim.SetBool("isWalk", false);
}
```

Элемент кода с переключением анимации

Таким образом, модель персонажа была добавлена в игровое пространство и было настроено отображение анимаций персонажа.



Персонаж с анимацией на сцене игры

Таким образом, модель персонажа была добавлена в игровое пространство и было настроено отображение анимаций персонажа.

Добавление геймплейных элементов

Основной геймплей игрового проекта – поиск объекта на карте. Объект на карте генерируется в случайной точке. Задача игроков найти объект раньше других.

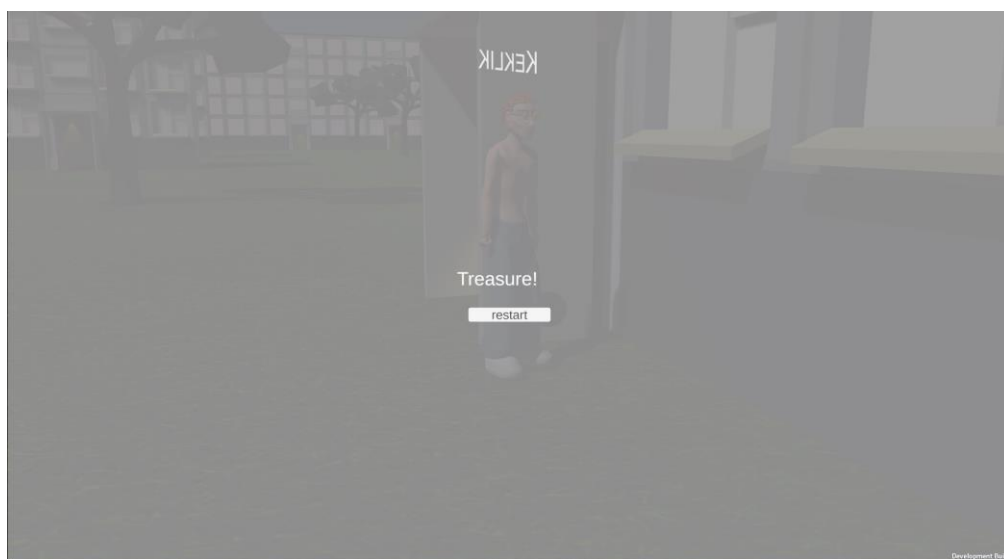
Для реализации этой задачи, в игровое окружение был добавлен объект для поиска. Для объекта был написан скрипт случайного расположения при создании мира.

Поскольку объект генерируется на определённой площади игрового мира, необходимо ограничить игрока на карте. Для этого была разработана система оповещения игрока в случае, если игрок слишком сильно отделился от области поиска. Система была реализована при помощи области триггера и окна оповещения.



Оповещение игрока о покидании зоны поиска

Для завершения игрового процесса при нахождении объекта на карте, была создана система, при которой в случае нахождения одним игроком искомого объекта, все игроки получают оповещение о выполнении задания и предложении перезапуска игры.



Оповещение игроков о выполнении задания

При перезапуске игры, игровой процесс повторяется.

Заключение

В ходе работ была разработана многопользовательская онлайн игра, поддерживающее единовременное подключение более пяти игроков и содержащее мультимедийное наполнение, представленное в виде трёхмерных игровых моделей.

Были добавлены анимации игрового персонажа, управляемого игроком. Анимации игрового персонажа проигрываются в соответствии с командами игрока.

Был разработан геймплей, позволяющий многократно повторять игровые сессии.

Список литературы

1. Ретопология в Blender [Электронный ресурс]. -
<https://www.youtube.com/watch?v=OHlxz3aPFiU&t=523s>
2. Запекание текстур в Blender [Электронный ресурс]. -
<https://www.youtube.com/watch?v=XIBOasHgttY&t=159s>
3. Основы скульптинга в Blender [Электронный ресурс]. -
<https://www.youtube.com/watch?v=I5RcF3CMnnE&t=478s>
4. Текстурирование в Blender [Электронный ресурс]. -
<https://www.youtube.com/watch?v=iSve1LF1E1Y>
5. Автоматическая ретопология в Blender [Электронный ресурс]. -
<https://www.youtube.com/watch?v=DolnQcFmpGw&t=66s>
6. Работа с Photon в игровом движке Unity [Электронный ресурс]. -
https://www.youtube.com/watch?v=3_E7SZ7g-H4
7. Импорт моделей из Blender в Unity [Электронный ресурс]. -
<https://www.youtube.com/watch?v=8myponhoJVQ>
8. Импорт анимаций из Blender в Unity [Электронный ресурс]. -
https://www.youtube.com/watch?v=2_Hn5ZsUIXM