


**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**  
**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

**СОГЛАСОВАНО**

Научный руководитель,  
приглашенный преподаватель  
департамента программной

 инженерии  
\_\_\_\_\_ Н.И. Веселко  
«11» апреля 2022 г.

**УТВЕРЖДАЮ**

Академический руководитель  
образовательной программы  
«Программная инженерия»,  
кандидат технических наук

\_\_\_\_\_ В.В. Шилов  
«11» апреля 2022 г.

Инв. № подл	Подп. и дата	Взам. инв.	№ Инв. № дубл.	Подп. и дата

**VR-ИГРА «ДИЗАЙНЕР ПЛЯЖЕЙ» НА ДВИЖКЕ UNREAL ENGINE 4**


**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.05.03-01 81 01-1-ЛУ**

**Исполнитель**

студентка группы БПИ198

 / М. А. Шалаева /

«11» апреля 2022 г.

**УТВЕРЖДЕН**

**RU.17701729.05.03-01 81 01-1-ЛУ**

**VR-ИГРА «ДИЗАЙНЕР ПЛЯЖЕЙ» НА ДВИЖКЕ UNREAL ENGINE 4**

**Пояснительная записка**

**RU.17701729.05.03-01 81 01-1-ЛУ**

**Листов 66**

Инв. № подл	Подп. и дата	Взам. инв.	№ Инв. № дубл.	Подп. и дата

## Содержание

<b>АННОТАЦИЯ</b>	<b>5</b>
<b>1 ВВЕДЕНИЕ</b>	<b>7</b>
1.1 Наименование программы	7
1.2 Документ, на основании которого ведется разработка	7
<b>2 НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ</b>	<b>8</b>
2.1 Назначение программы	8
2.1.1 Функциональное назначение	8
2.1.2 Эксплуатационное назначение	8
2.2 Краткая характеристика области применения программы	8
<b>3 ТРЕБОВАНИЯ К ПРОГРАММЕ</b>	<b>9</b>
3.1 Постановка задачи на разработку программы	9
3.2 Описание функционирования программы	12
3.2.1 Общая архитектура	12
3.2.2 Входные параметры	15
3.2.3 Используемые методы оптимизации	16
3.2.3.1 Использование Low-Poly и Mid-Poly объектов	16
3.2.3.2 Level of Details	16
3.2.3.3 Merge Actors	17
3.2.3.4 Foliage	18
3.2.3.5 Статический свет	20
3.2.4 Поднятие / бросание предметов	20
3.2.5 Передвижение	21

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3.2.6	Поворот камеры	24
3.2.7	Загрузка уровней	25
3.2.8	Логика работы уровня	26
3.2.9	Виджеты	27
3.2.9.1	Интерактивная сфера	28
3.2.9.2	Виртуальная клавиатура	28
3.2.9.3	Панель «Инструменты»	29
3.2.9.4	Меню паузы	30
3.2.9.5	Спавн объекта с виджетом	33
3.2.10	Режим продажи	34
3.2.11	Режим отделки	35
3.2.12	Метод бегущих лучей	36
3.2.13	Процедурная генерация мусора	36
3.2.14	Искусственный интеллект	37
3.2.14.1	NPC Кот	37
3.2.14.2	NPC Рыба	40
3.3	Описание и обоснование выбора метода организации входных и выходных данных	40
3.4	Описание и обоснование выбора состава технических и программных средств	45
3.4.1	Состав технических и программных средств	45
3.4.2	Обоснование выбора технических и программных средств	46
<b>4</b>	<b>ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ</b>	<b>47</b>
4.1	Ориентировочная экономическая эффективность	47
4.2	Предполагаемая потребность	47
4.3	Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами	47

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

<b>5 ПЕРСПЕКТИВЫ РАЗВИТИЯ</b>	<b>48</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>49</b>
<b>ПРИЛОЖЕНИЕ 1</b>	<b>1</b>
<b>ПРИЛОЖЕНИЕ 2</b>	<b>2</b>
<b>ПРИЛОЖЕНИЕ 3</b>	<b>6</b>
<b>ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ</b>	<b>15</b>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## АННОТАЦИЯ

В данном программном документе приведена пояснительная записка к программному проекту «VR-игра “Дизайнер пляжей” на движке Unreal Engine 4».

Настоящая пояснительная записка содержит следующий ряд разделов:

- 1) «Введение»;
- 2) «Назначение и область применения»;
- 3) «Требования к программе»;
- 4) «Ожидаемые технико-экономические показатели»;
- 5) «Перспективы развития»;
- 6) «Список использованных источников»;
- 7) приложения.

В разделе «Введение» указаны наименование программы и краткая характеристика области ее применения.

В разделе «Назначение и область применения» указаны: функциональное назначение программы, эксплуатационное назначение программы, краткая характеристика области применения программы.

В разделе «Требования к программе» содержатся следующие подразделы: постановка задачи на разработку программы; описание функционирования программы; описание и обоснование выбора метода организации входных и выходных данных; описание и обоснование выбора состава технических и программных средств.

Раздел «Ожидаемые технико-экономические показатели» содержит ориентировочную экономическую эффективность, предполагаемую потребность, экономические преимущества разработки программы по сравнению с аналогами.

В разделе «Перспективы развития» находятся предположительные пути совершенствования и развития разработанной системы.

В разделе «Список использованных источников» указаны ссылки на источники, которые были использованы при разработке.

Настоящий документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.101–77 Виды программ и программных документов [17];

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- 2) ГОСТ 19.102–77 Стадии разработки [18];
- 3) ГОСТ 19.103–77 Обозначения программ и программных документов [19];
- 4) ГОСТ 19.104–78 Основные надписи [20];
- 5) ГОСТ 19.105–78 Общие требования к программным документам [21];
- 6) ГОСТ 19.106–78 Требования к программным документам, выполненным печатным способом [22];
- 7) ГОСТ 19.404–79 Пояснительная записка. Требования к содержанию и оформлению [23].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603–78 [24], ГОСТ 19.604–78 [25].

Перед изучением данного документа предполагается предварительное ознакомление с приведенной в Приложении 1 терминологией.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 1 ВВЕДЕНИЕ

### 1.1 Наименование программы

**Наименование темы разработки:** «VR-игра “Дизайнер пляжей” на движке Unreal Engine 4»

**Наименование темы разработки на английском языке:** «“Beach Designer” Unreal Engine 4 VR-game»

### 1.2 Документ, на основании которого ведется разработка

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем тема курсового проекта.

**Организация, утвердившая этот документ:** Национальный исследовательский университет «Высшая школа экономики», Факультет компьютерных наук, образовательная программа «Программная инженерия».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 2 НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

### 2.1 Назначение программы

#### 2.1.1 Функциональное назначение

Разрабатываемое приложение дает пользователю возможность примерить на себя роль специалиста по благоустройству пляжей, который может взаимодействовать с объектами на картах, убирать территорию, покупать и расставлять пляжный инвентарь по своему усмотрению на нескольких пляжах Америки. За выполнение задания пользователь будет получать определенную сумму денежной валюты, которую можно будет использовать для приобретения более дорогих товаров на последующих уровнях.

#### 2.1.2 Эксплуатационное назначение

Основными конечными потребителями разрабатываемого приложения являются люди в возрасте от 12 до 25 лет, которым хочется удовлетворить свою эстетическую потребность и которым интересно примерить на себя роль пляжного дизайнера.

Для корректного использования данной программы пользователю будут нужны очки виртуальной реальности Oculus Quest 2, ПК, отвечающий требованиям из пункта 3.4.1, и кабель Oculus Link для подключения очков к ПК (если не используется AirLink).

### 2.2 Краткая характеристика области применения программы

«VR-игра “Дизайнер пляжей” на движке Unreal Engine 4» – прикладная программа, разрабатываемая в развлекательных целях и представляющая собой симулятор дизайнера пляжей, занимающегося уборкой мусора с территории, приобретением и расстановкой пляжного инвентаря.

Приложение предназначено для использования игроками, имеющими в распоряжении VR-очки Oculus Quest 2, в возрасте от 12 до 25 лет и направлено на удовлетворение эстетической потребности посредством декорирования территории, экологической потребности [28] посредством уборки и облагораживания территории, а также на формирование следующих мировоззренческих ценностей: важность сохранения окружающей среды, забота о ней, ответственное отношение к уборке мусорных отходов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

### 3 ТРЕБОВАНИЯ К ПРОГРАММЕ

#### 3.1 Постановка задачи на разработку программы

Разрабатываемая программа должна предоставлять следующий перечень обязательных функций:

- 1) Визуализация трех игровых локаций, а именно:
  - офиса (стартовая локация);
  - локации, реализованной на основе пляжа парка Wai'anapanapa, Гавайи, США;
  - локации, реализованной на основе Papakōlea Green Sand Beach, Гавайи, США;
- 2) Возможность просмотра окружающей местности посредством изменения поворота VR-очков в реальном мире;
- 3) Возможность изменения координат игрока в виртуальном пространстве посредством перемещения VR-очков в реальном мире;
- 4) Возможность изменения координат виртуальных рук в пространстве в соответствии с перемещением контроллеров Oculus Quest 2 в реальном мире;
- 5) Возможность взаимодействия с предметами с помощью «виртуальных рук» (контроллеров Oculus Quest 2);
- 6) Реализация перемещения игрока в игровом пространстве с помощью контроллеров на пляжных уровнях;
- 7) Реализация поворота камеры на 45° с помощью контроллеров;
- 8) Реализация телепортации в пространстве с помощью контроллеров;
- 9) Возможность взаимодействия с виджетами;
- 10) Реализация смены активного контроллера;
- 11) Реализация виртуальной клавиатуры;
- 12) Реализация обучения, запускающегося в начале игры;
- 13) Реализация ввода имени и роста пользователя с помощью интерактивной клавиатуры во время обучения;
- 14) Возможность повторного запуска обучения;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- 15) Реализация взаимодействия с игровым планшетом в локации «офис», а именно взятие гаджета в руки, взаимодействие с экраном (нажатие на элементы, расположенные на экране устройства);
- 16) Возможность просмотра информации об уровне, на который может отправиться пользователь, на экране планшета;
- 17) Перемещение между локациями из пункта 1 данного списка (далее — пляжные уровни) с помощью планшета в локации «офис»;
- 18) Реализация загрузочного экрана при переходе с одного уровня на другой;
- 19) Реализация процедурной генерации мусорных объектов на пляжном уровне;
- 20) Возможность брать, перемещать и бросать процедурно сгенерированный мусор с помощью контроллеров на пляжных уровнях;
- 21) Реализация мусорного бака на пляжных уровнях, который можно брать, перемещать и бросать с помощью контроллеров;
- 22) Возможность уборки (уничтожения) процедурно сгенерированного мусора с пляжных уровней с помощью помещения их в мусорный бак;
- 23) Возможность декорировать пляжные уровни с помощью мебели и элементов декора, которые можно купить в панели «Инструменты», поставить на уровне, поменять их координаты по осям X, Y, а также поворот по оси Z после завершения этапа уборки пляжа;
- 24) Возможность изменения цвета мебели;
- 25) Возможность завершения уровня и получения игровой валюты;
- 26) Возможность возврата на завершённый уровень в режиме просмотра;
- 27) Реализация NPC кота с искусственным интеллектом, который будет на пляжных уровнях следовать за игроком и искать мусорные объекты;
- 28) Реализация взаимодействия с котом, а именно возможность гладить его на любом уровне и кормить его в офисе;
- 29) Возможность сменить окрас кота на один из трех возможных во время обучения при прохождении обучения в первый раз;
- 30) Реализация NPC рыб в море на пляжных уровнях, обладающих искусственным интеллектом и умением следовать к случайной точке пространства;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

31) Реализация меню паузы, которое должно предоставлять следующий перечень функций на пляжном уровне:

1. Возможность возврата всех процедурно сгенерированных и существующих на данный момент на пляжном уровне объектов на их стартовые позиции;
2. Возможность регенерировать уровень;
3. Открытие меню настроек, в котором можно будет изменить громкость звуков, изменить громкость музыки, переключить текущий трек, начать игру заново (стереть текущее сохранение и начать с нуля);
4. Возможность воспроизвести обучение;
5. Просмотр управления;
6. Возможность сохранить игру;
7. Возможность вернуться с уровня в офис с сохранением текущего состояния уровня;
8. Возможность покинуть игру.

Меню паузы должно предоставлять следующий перечень функций в офисе:

1. Перезагрузка текущего уровня;
2. Открытие меню настроек, в котором можно будет изменить громкость звуков, изменить громкость музыки, начать игру заново (стереть текущее сохранение и начать с нуля);
3. Возможность воспроизвести обучение;
4. Просмотр управления;
5. Возможность сохранить игру;
6. Возможность покинуть игру.

32) Возможность открытия меню паузы;

33) Реализация панели «Инструменты» с двумя вкладками: «Действия» и «Магазин». Вкладка «Инструменты» должна предоставлять следующий перечень функций:

1. Создание (спавн) объектов еды для кота;
2. Вызов мусорного ведра на пляжном уровне во время уборки территории;
3. Телепортация мусорного ведра к игроку, если оно создано и находится в игровом мире;
4. Уничтожение мусорного ведра, если оно создано и находится в игровом мире;
5. Включение режима «Отделки» на пляжном уровне во время декорирования территории;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

6. Включение режима «Продажи» на пляжном уровне во время декорирования территории.

Вкладка «Магазин» должна предоставлять следующий перечень функций:

1. Возможность просмотра списка доступных товаров;
  2. Возможность просмотра информации о конкретном товаре;
  3. Возможность выбора цвета товара, если у него есть вариации расцветки;
  4. Создание объекта класса просматриваемого товара на уровне.
- 34) Возможность открытия панели «Инструменты»;
- 35) Реализация режима «Продажи», с помощью которого во время декорирования пляжного уровня можно будет продать выбранный пользователем товар;
- 36) Возможность входа в режим «Продажи»;
- 37) Реализация режима «Отделки», с помощью которого во время декорирования пляжного уровня можно будет изменить положение и поворот выбранного пользователем товара;
- 38) Возможность входа в режим «Отделки»;
- 39) Реализация выделения контура товара на пляжном уровне в режимах «Продажи» и «Отделки»:
- голубым цветом в случае наведения активного контроллера на объект;
  - зеленым цветом в случае выбора товара.
- 40) Воспроизведение музыки на любой локации игры;
- 41) Возможность смены трека на пляжном уровне с помощью меню паузы;
- 42) Воспроизведение звука падения объекта в воду в случае его соприкосновения с поверхностью воды;
- 43) Реализация сохранения имени, количества денег пользователя, громкости звука и музыки, трека, выбранного на пляжном уровне, выбранного окраса кота, статуса каждого пляжного уровня, класса и положения всех находящихся на нем объектов.

## 3.2 Описание функционирования программы

### 3.2.1 Общая архитектура

Разработанная система имеет Event Driven архитектуру, или событийно-ориентированную архитектуру. Классы сообщаются друг между другом посредством событий: одни классы событие за-

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

пускают, другие — получают уведомление о запуске события.

Схематично взаимодействие классов представлено на рисунке 1.

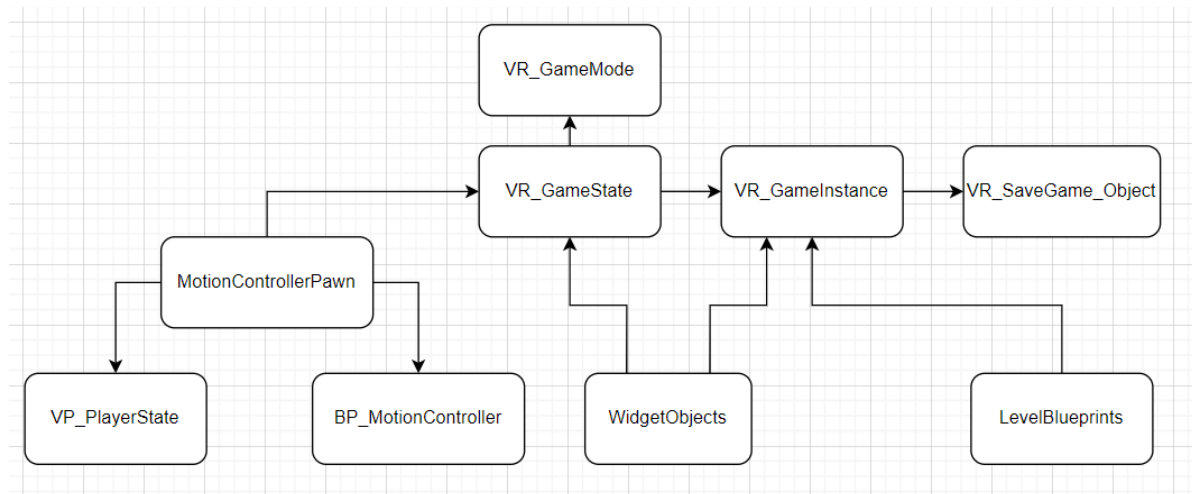


Рис. 1 – Архитектура

Unreal Engine 4 позволяет реализовать событийно-ориентированный подход с помощью использования Event Dispatchers, или диспетчеров событий.

Диспетчеры событий в разрабатываемом проекте находятся в классах VR\_GameState и VR\_GameInstance, а остальные классы либо вызывают нужные диспетчеры событий, либо их принимают.

Пример работы диспетчеров событий можно проиллюстрировать на работе загрузки потоковых уровней, или streaming levels (подробнее в пункте 3.2.7). Поточковые уровни загружаются через один общий «постоянный» уровень, который реализует выгрузку текущего уровня и загрузку нового. В классе VR\_GameInstance хранится диспетчер событий под названием «Smooth Load Level» с входным параметром «LevelName», являющимся названием уровня, который нужно загрузить. Смена уровня может вызываться из разных классов (например, из класса планшета пользователя, из меню паузы). Для того, чтобы уведомить класс «постоянного» уровня о том, что необходимо сменить текущий уровень на новый, нужно вызвать диспетчер событий «Smooth Load Level» в классе, который хочет данное событие инициировать, а «постоянный» уровень, как принимающий класс, должен на это отреагировать и запустить метод загрузки нового уровня.

Например, пользователь нажал на кнопку планшета, которая отвечает за загрузку пляжного уровня WaianapapaBeach из локации «офис». Тогда происходит получение ссылки на Game Instance, приведение найденного объекта к типу нашего Game Instance'a, а именно VR\_GameInstance, после чего вызывается диспетчер события «Smooth Load Level» (см. рисунок 2).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

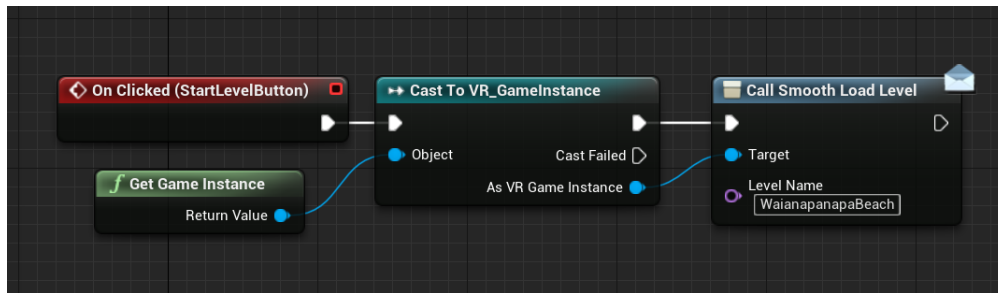


Рис. 2 – Класс планшета, нажатие на StartLevelButton

Принимающий класс «постоянного» уровня имеет заранее прописанный ответ на появление этого события. С помощью нода «Bind Smooth Load Level» можно задать событие, которые запустится, как только кто-то вызовет диспетчер событий Smooth Load Level. В нашем примере таким событием выступает SmoothLoadLevel (Binding), который и будет запущен после нажатия на рассмотренную выше кнопку планшета (см. рисунок 3).

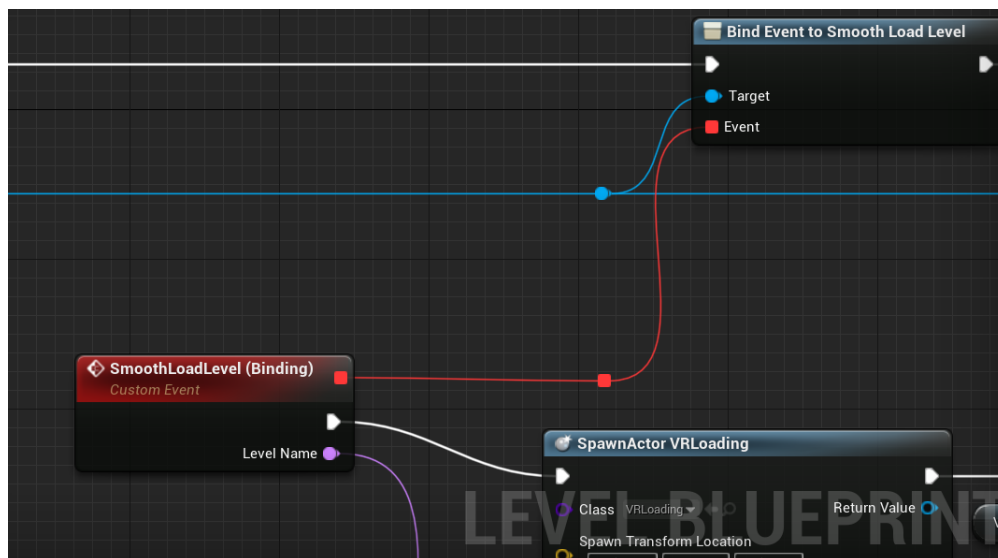


Рис. 3 – Ответ на событие

VR\_GameState реализует правила игры, прописанные в классе VR\_GameMode.

VR\_GameInstance реализует сохранение игры, поэтому сообщается с классом VR\_SaveGame\_Object. Для вызова сохранения из разных мест программы необходимо обратиться именно к VR\_GameInstance, а не к VR\_SaveGame\_Object напрямую.

Класс MotionControllerPawn, реализующий класс игрока, имеет ссылку на объекты класса BP\_MotionController, которые, в свою очередь, реализуют классы контроллеров пользователя, а также имеет ссылку на класс VR\_PlayerState, хранящий в себе информацию об игроке.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

### 3.2.2 Входные параметры

Для того, чтобы получать уведомления о событиях нажатия каких-либо кнопок на контроллерах, нужно указать их в качестве входных параметров для игры. Для этого нужно перейти в Project Settings движка Unreal Engine 4, перейти во вкладку Input → Bindings и добавить следующий список Action Mappings (см. рисунки 4, 5):



Рис. 4 – Action Mappings



Рис. 5 – Action Mappings 2

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

А также следующий список Axis Mappings:

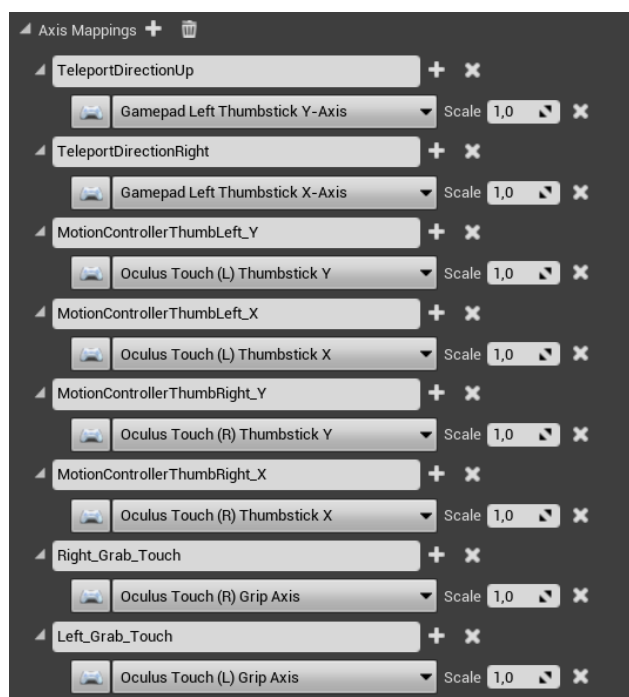


Рис. 6 – Axis Mappings

### 3.2.3 Используемые методы оптимизации

#### 3.2.3.1 Использование Low-Poly и Mid-Poly объектов

Современные игровые движки под капотом триангулируют полигоны объектов, то есть автоматически разбивают все плоскости на треугольники. Поэтому использование объектов с детализированной статической сеткой приводит к ухудшению производительности, что может быть критично для VR-игр.

В рамках данного проекта было принято решение использовать объекты с менее детализированной сеткой, то есть Low-Poly (см. рисунок 7) и Mid-Poly 3D модели, чтобы увеличить производительность.

#### 3.2.3.2 Level of Details

Unreal Engine 4 предоставляет возможность настраивать детализированность статической сетки разных объектов в зависимости от положения игрока в игровом пространстве с помощью панели Level of Details, или LODs [8] (см. рисунок 8).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

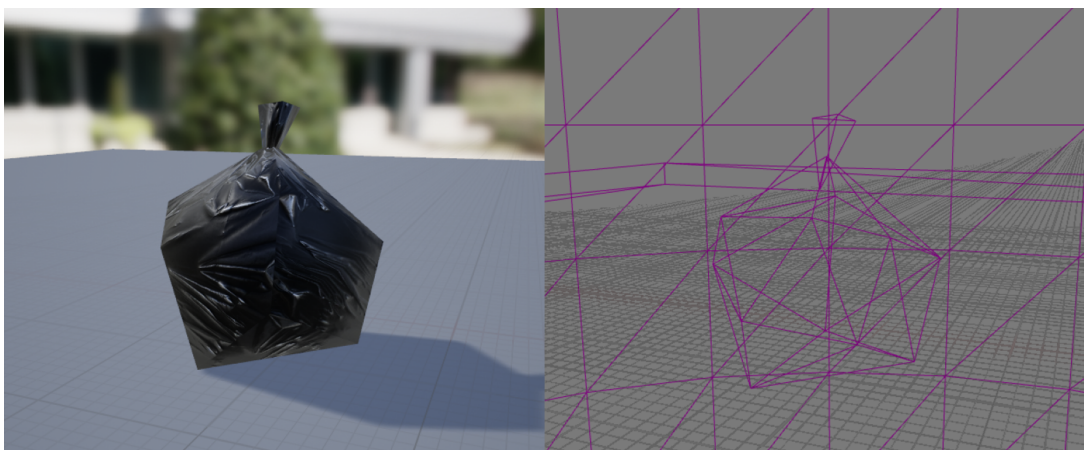


Рис. 7 – Low-poly 3D модель

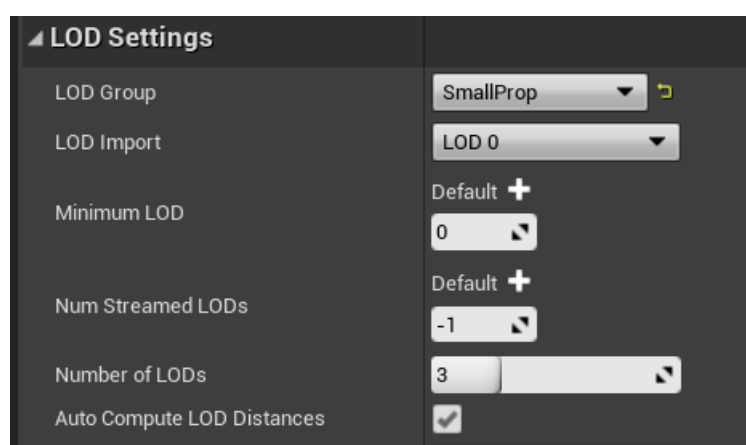


Рис. 8 – Level of Details

Когда пользователь находится близко к объекту, необходима детализированная статическая сетка. Однако когда пользователь находится далеко и предмет занимает всего несколько пикселей на экране, максимальная детализированность статической сетки избыточна. Правильно настроенные LOD'ы позволяют сменять сложность статической сетки предметов по мере приближения или отдаления игрока от них. Такой подход позволяет улучшить производительность игрового уровня и применяется на большей части 3D-моделей, задействованных в данном проекте.

### 3.2.3.3 Merge Actors

Действие «Merge Actors» означает «склеивание» статических сеток выбранных объектов. Когда разработчик расставляет объекты на карте, они воспринимаются движком как отдельные статические сетки и при загрузке уровня прорисовываются последовательно. Если воспользоваться функцией «Merge Actors» и объединить несколько статических сеток в один объект, движок начнет воспри-

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

нимать их как единое целое и будет прорисовывать полигоны всех статических сеток одновременно, а не последовательно. Данный подход позволяет не только увеличить производительность и уменьшить время загрузки уровня во время самой игры, но еще и позволяет уменьшить время запекания света.

Примером такого «склеивания» является стартовая локация «офис», в которой большая часть мебели, стены, пол и потолок комнаты объединены в один объект (см. рисунок 9).

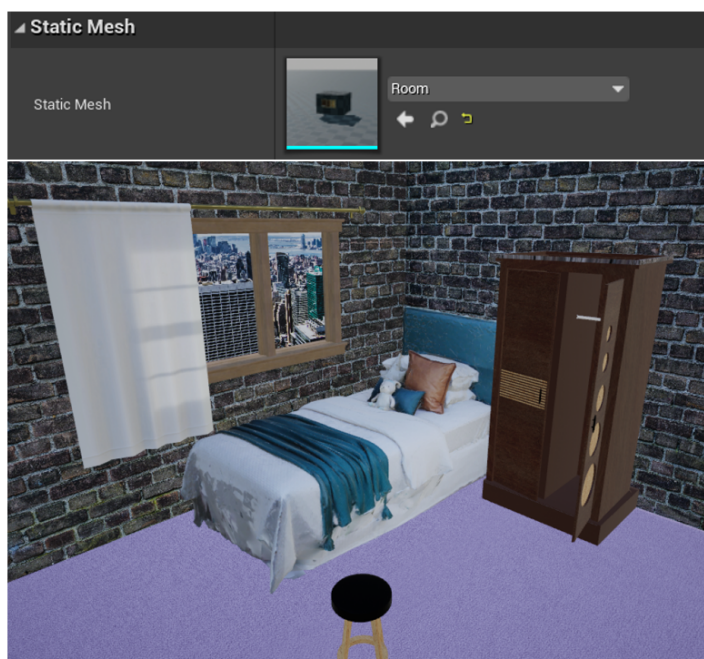


Рис. 9 – Merge Actors

### 3.2.3.4 Foliage

Unreal Engine 4 позволяет использовать инструмент под названием Foliage, который представляет из себя режим быстрого расставления объектов на ландшафте и позволяет за короткий промежуток времени создавать реалистичные пейзажи. С его помощью можно быстро создать лес, «посадить» растительность на землю и так далее [2].

Помимо удобства создания флоры игрового мира Foliage также предоставляет возможность изменять настройки расставляемых объектов. Для улучшения производительности у каждого расставляемого таким способом объекта была выставлена дистанция создания: то есть при отдалении камеры от объекта на определенное расстояние, объект уничтожается. При приближении к объекту он, наоборот, создается.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Для ручного настраивания данного параметра нужно перейти в режим Foliage, выбрать объект, выбрать вкладку «Instance Settings» и настроить поле «Max» у параметра «Cull Distance» (см. рисунок 10).



Рис. 10 – Instance Settings

Пример действия данного параметра представлен на рисунках 11, 12. Когда камера находится на расстоянии большем, чем 13000 юнитов, от объектов на дальнем холме, они не существуют. Однако как только камера приближается на расстояние, меньшее 13000 юнитов, объекты создаются на карте.

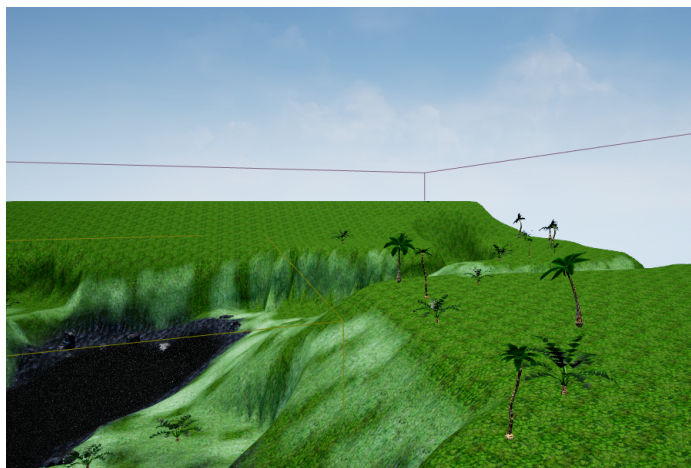


Рис. 11 – Работа Cull Distance

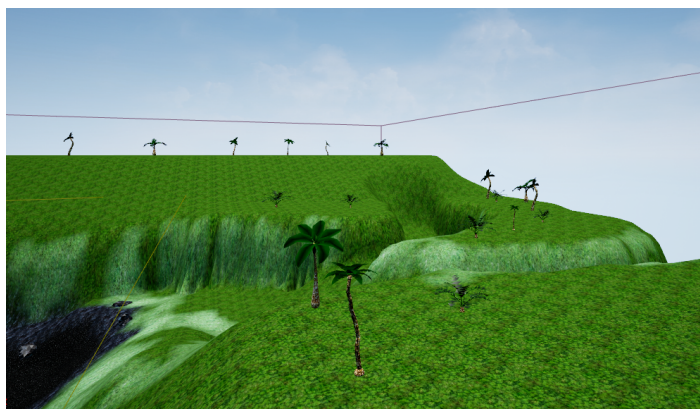


Рис. 12 – Работа Cull Distance 2

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

### 3.2.3.5 Статический свет

Также для улучшения производительности у всех источников света было выбрано значение «Stationary» или «Static» у параметра «Mobility» (см. рисунок 13). Данный подход подразумевает, что источник света не будет перемещаться по карте во время самой игры. Это позволяет увеличить производительность за счет заранее запеченных теней от статических объектов. Поскольку перерасчет тени статических объектов во время игры не происходит, на это не нужно тратить ресурсы.

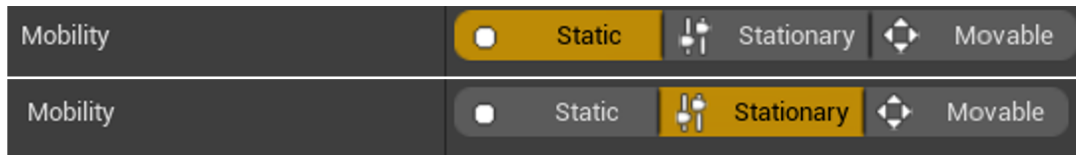


Рис. 13 – Stationary и Static Light

### 3.2.4 Поднятие / бросание предметов

Для того, чтобы предоставить пользователю возможность поднимать и бросать предметы на уровне, необходимо создать для каждого подобного объекта Blueprint-класс на базе его статической сетки. Данный класс должен реализовывать интерфейс PickupActorInterface и иметь реализацию событий Drop и PickUp (см. рисунки 14, 15).

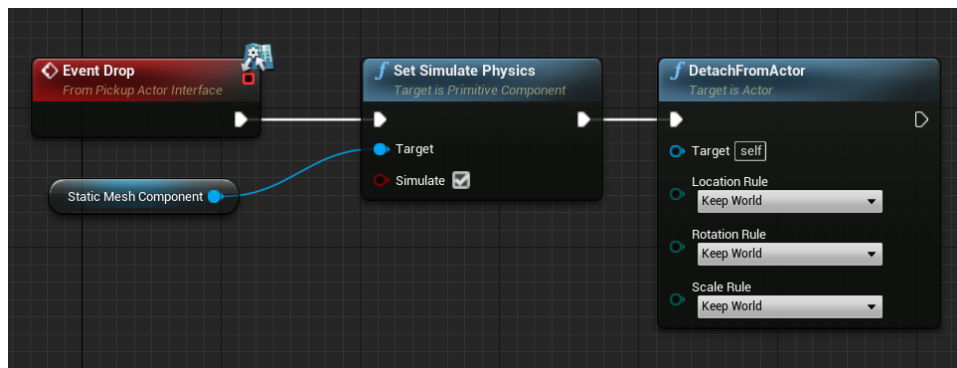


Рис. 14 – Событие Drop

В настройках статической сетки предмета параметр «Collision Complexity» должен быть выставлен на «Use Simple Collision As Complex» (см. рисунок 16), должен быть отмечен галочкой пункт «Simulate Physics» (см. рисунок 17), а также параметр «Collision Presents» должен быть выставлен на «PhysicsActor» (см. рисунок 18).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

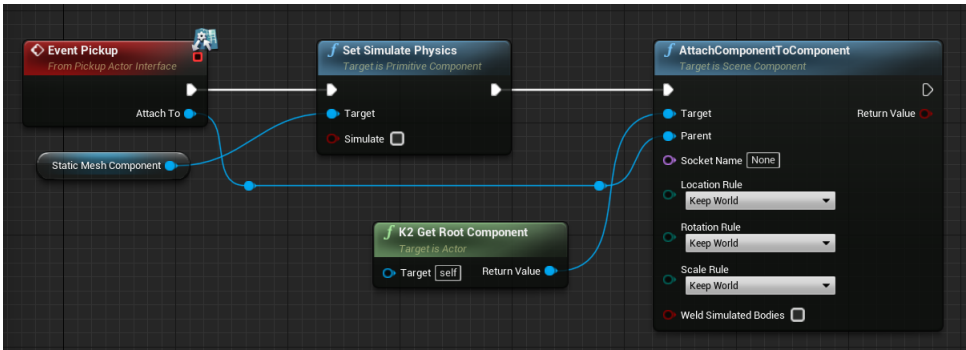


Рис. 15 – Событие PickUp



Рис. 16 – Collision Complexity

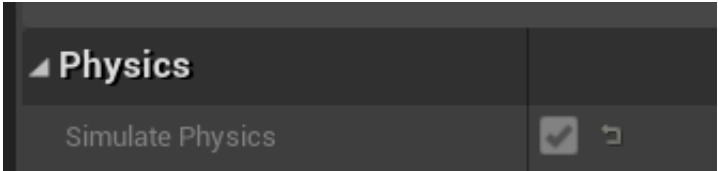


Рис. 17 – Simulate Physics

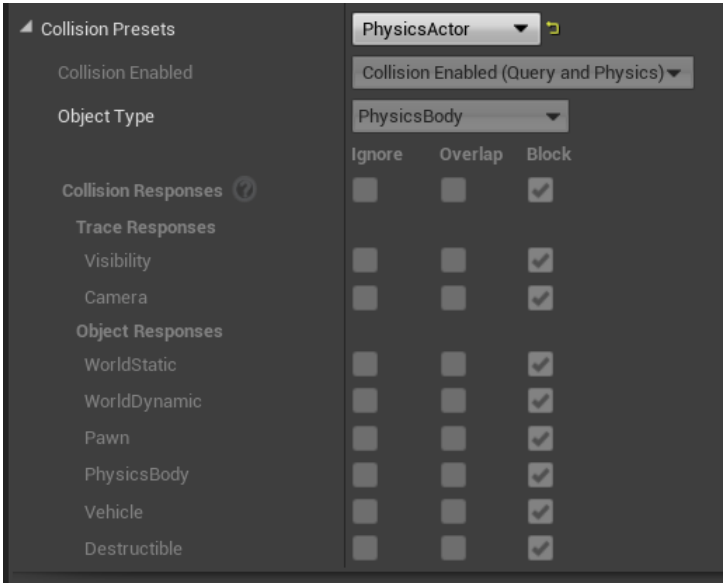


Рис. 18 – PhysicsActor

3.2.5 Передвижение

На пляжных уровнях должно быть реализовано передвижение с помощью джойстика левого контроллера.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

The image displays a Unreal Engine Blueprint graph titled "Movement by Controller Left". The graph is set against a dark background with a grid pattern. The logic flow is as follows:

- Start:** A "VR Player State" variable is connected to a "Target" input of a "Motion Speed" node.
- Condition:** A "Branch" node with a "Condition" input set to  $0.0 == 0.0$ . The "True" path leads to the "Get Right Vector" node, and the "False" path leads to the "Add Movement Input" node.
- Vector Calculation:** The "Get Right Vector" node (Target is Scene Component) is connected to a "Make Vector" node. The "Make Vector" node has three inputs: "Returns Value X", "Returns Value Y", and "Returns Value Z" (set to  $0.0$ ).
- Nav Mesh Check:** The "Is Movement Point in Nav Mesh" node (Target is Self, Movement Vector, Speed) receives input from the "Make Vector" node. Its "Condition" input is set to "Can Move".
- Final Action:** The "Add Movement Input" node (Target is Self, World Direction, Scale Value, Force) receives input from the "Is Movement Point in Nav Mesh" node. Its "Condition" input is set to "True".

The word "BLUEPRINT" is visible in the bottom right corner of the image.

```

graph TD
    Start([Начало]) --> GetAxis[/Получение поворота  
джойстика Axis/]
    GetAxis --> IsZero{Axis = 0?}
    IsZero -- Да --> End([Конец])
    IsZero -- Нет --> CalcAxis[Axis = Axis *  
скорость  
передвижения]
    CalcAxis --> GetVec[Вызов функции Get  
Right Vector для  
камеры игрока]
    GetVec --> CalcPoint[Расчет точки, в  
которую должен  
переместиться игрок]
    CalcPoint --> InZone{Точка находится в допустимой  
для перемещения зоне?}
    InZone -- Да --> Move[Переместить игрока  
в нужном  
направлении]
    InZone -- Нет --> End
    Move --> End
  
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

В случае, если пользователь поворачивает джойстик по оси Y, то есть вверх или вниз, следует использовать алгоритм, представленный в виде Blueprint-функции на рисунке 21, а также представленный в виде блок-схемы на рисунке 22.

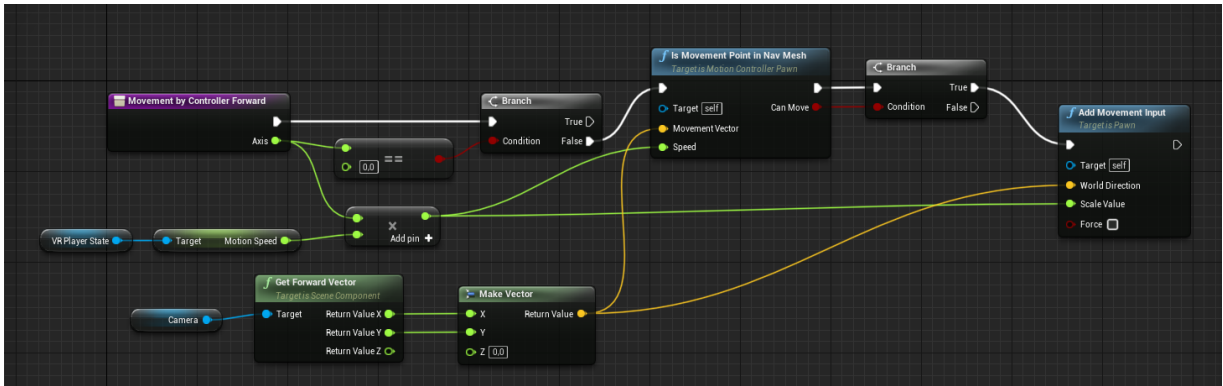


Рис. 21 – Movement by Controller Forward

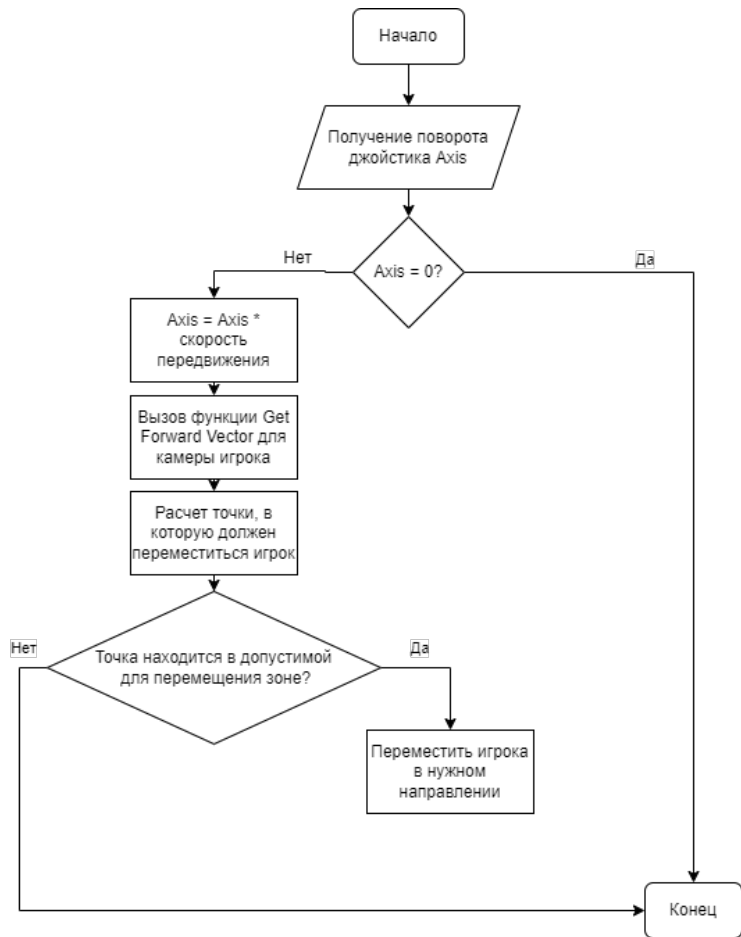


Рис. 22 – Блок-схема Movement by Controller Forward

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Допустимая для передвижения зона определяется заранее выставленным на карте объектом Nav Mesh Bounds Volume [10]. Он представляет из себя невидимый прямоугольный параллелепипед. Также используется для телепортации игрока и для ограничения передвижения ИИ.

### 3.2.6 Поворот камеры

Поворот камеры должен осуществляться в соответствии с поворотом джойстика правого контроллера по оси X, то есть влево-вправо. Алгоритм, рассчитывающий поворот камеры игрока по значению поворота джойстика, представлен в виде Blueprint-функции на рисунках 23, 24.

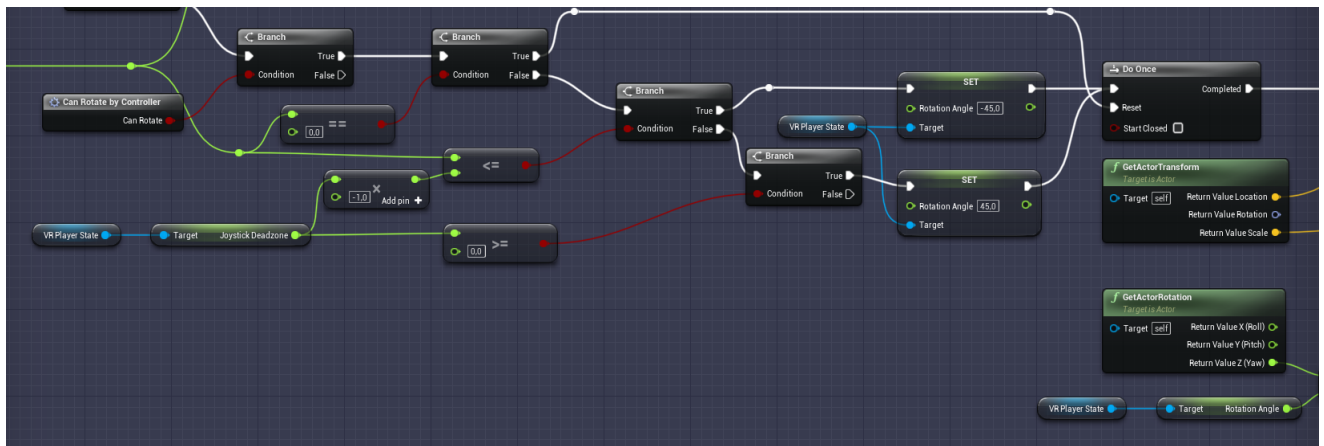


Рис. 23 – Поворот камеры



Рис. 24 – Поворот камеры. Продолжение

В случае, если полученное от джойстика значение Axis не равно нулю, мы проверяем направление поворота: если значение Axis отрицательно, следует повернуть камеру на  $-45^\circ$ , если положительно, то на  $45^\circ$ . Используется нода «Do once». Без нее изменение угла поворота камеры будет

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

происходить каждый тик, потому что значение Axis считывается непрерывно, а не один раз после начала движения джойстика. Данная функция сбрасывается, как только Axis становится равным нулю. Тем самым гарантируется, что за одно движение джойстика произойдет только один поворот. Как только мы получили значение угла поворота, нужно присвоить новый поворот объекту класса MotionControllerPawn, представляющему из себя игрока.

### 3.2.7 Загрузка уровней

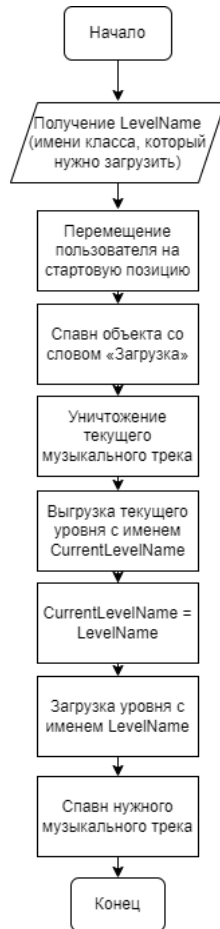


Рис. 25 – Блок-схема загрузки уровня

Unreal Engine 4 позволяет загружать уровни несколькими способами. Один — вызов функции Load Level Instance (by Name) или Load Level Instance (by Object Reference). Тогда движок самостоятельно выполнит выгрузку текущего уровня и загрузку нового. Такой подход позволяет задавать нескольким уровням разные GameMode классы, однако в данном подходе есть существенный минус для VR-игр — на время прогрузки нового уровня игра будет недоступна, и у пользователя буквально «зависнет» последний увиденный им кадр. Из-за этого в данном проекте было принято решение

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

использовать систему Level Streaming, которая позволяет из одного пустого «постоянного» уровня обращаться ко всем остальным. «Постоянный» уровень будет существовать на протяжении всей игровой сессии и позволит избежать зависания экрана у игрока.

Логика работы алгоритма загрузки уровня представлена на блок-схеме (см. рисунок 25).

Для того, чтобы добавить игровой уровень в «постоянный», нужно открыть «постоянный» уровень и нажать на кнопку «Add Existing», после чего выбрать и добавить готовый уровень (см. рисунок 26).

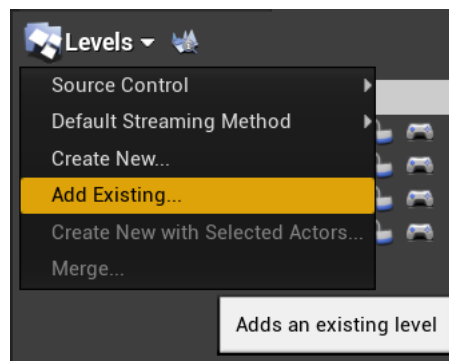


Рис. 26 – Добавление Streaming Level

### 3.2.8 Логика работы уровня

Во время прохождения пляжного уровня пользователь сможет перемещаться по территории с помощью джойстика на левом контроллере, поворачивать камеру на  $45^\circ$  с помощью джойстика на правом контроллере, телепортироваться с помощью нажатия и отпускания джойстика любого из контроллеров, вызывать меню паузы, вызывать панель «Инструменты», поднимать и бросать мусорные объекты и мусорный бак, расставлять элементы декора, менять их угол поворота по оси Z и менять координаты по осям X, Y.

Прохождение игрового уровня можно разделить на два задания:

- 1) Уборка мусора на территории пляжа;
- 2) Декорирование территории.

На первом этапе пользователю не должны быть доступны любые манипуляции с товарами из магазина. Ему должны быть доступны только действия, связанные с уборкой мусора. В меню «Инструменты» на этом этапе должны быть недоступны вкладка «Магазин», кнопка «Завершить уровень», а также кнопки «Кот» и «Дизайн» во вкладке «Действия».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Для того чтобы пройти данный этап, пользователю необходимо «выкинуть» все мусорные объекты на уровне. Для этого нужно вызвать панель «Инструменты», выбрать вкладку «Действия» → «Уборка» и нажать на кнопку «Создать мусорный бак». Мусорный бак должен заспавниться рядом с пользователем. Также с помощью панели «Инструменты» можно будет перемещать бак из любой точки карты к пользователю, а также уничтожать его. Чтобы мусорный объект уничтожился, необходимо поместить его в мусорный бак. Перед уничтожением объекта необходимо получить его координаты и заспавнить «летающее» сообщение с количеством денег, полученным пользователем за данный мусорный объект. Количество денег выбирается случайным образом из диапазона [4, 10]. После уничтожения объекта счетчик количества найденных мусорных объектов на уровне должен увеличиваться на единицу, а пользователю должно добавиться выбранное ранее количество денег. Как только все мусорные объекты будут уничтожены, бак должен автоматически уничтожиться.

На этапе декорирования пляжа пользователь может покупать предметы в панели «Магазин» из панели инструментов, перемещать и вращать их с помощью левого и правого джойстиков соответственно. Также в случае, если пользователь хочет изменить положение уже купленного объекта, должен быть предусмотрен режим «отделки». Если пользователь захочет продать объект, то должен быть предусмотрен режим «продажи».

Как только пользователь расставит нужные элементы декора и решит завершить уровень, в панели «Инструменты» нужно будет нажать на кнопку «Завершить уровень», после чего произойдет начисление гонорара и выход на стартовую локацию.

### 3.2.9 Виджеты

Unreal Engine 4 предоставляет инструмент под названием Unreal Motion Graphics (UMG). UMG используется для формирования элементов пользовательского интерфейса в игре. Пользовательский интерфейс можно создавать с помощью Виджет Blueprint класса, содержащего в себе список определенных элементов — виджетов [13].

Виджеты представляют из себя элементы пользовательского интерфейса, которые дают возможность визуализировать различные элементы взаимодействия на экране, такие как кнопки, блоки с текстом, прокрутка, пульсатор и так далее. При стандартном использовании виджеты прорисовываются на экране устройства «поверх» остальных объектов в игровом мире.

Классическое использование виджетов затруднительно в виртуальной реальности по двум причинам:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- 1) Отсутствие курсора мыши не позволяет сделать классический «клик» по виджету для активации кнопок, слайдеров и так далее;
- 2) В силу устройства шлема пользователю будет казаться, что элементы управления расположены слишком близко к нему.

Для разрешения данных проблем было принято решение использовать альтернативу курсора мыши в виде интерактивной сферы, а также не выводить виджеты на экран, а спавнить 3D объекты в игровом мире с привязанными к ним виджетами.

### 3.2.9.1 Интерактивная сфера

Альтернативой курсора мыши выступает интерактивная сфера. Для ее корректной работы нужно добавить в класс `MotionControllerPawn` статическую сетку `InteractionSphere`, а также компонент `WidgetInteraction`.

Компонент `WidgetInteraction` должен быть привязан к активному `MotionController`. Смена активного контроллера должна происходить по нажатию на триггер неактивной руки.

Аналогом «клика» является нажатие на триггер активного контроллера. «Клик» осуществляется посредством нода, представленного на рисунке 27.

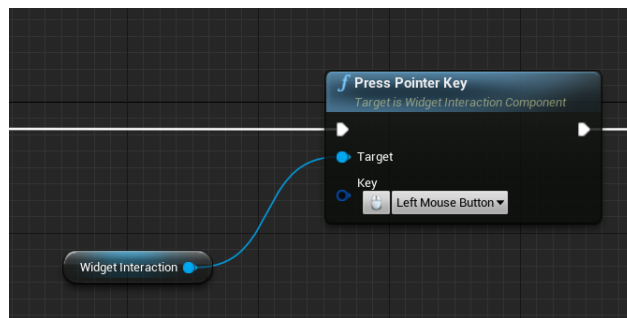


Рис. 27 – «Клик» с помощью интерактивной сферы

### 3.2.9.2 Виртуальная клавиатура

Разрабатываемая программа должна предоставлять возможность взаимодействовать с виртуальной клавиатурой. Ее можно использовать при активации виджета «Editable Text Box», который представляет собой строку для ввода текста. Для того, чтобы клавиатура, являющаяся отдельным виджетом и не имеющая ссылок на другие компоненты, могла сообщаться с виджетом «Editable Text Box», нужно убрать галочку с параметра «Is Focusable» на всех элементах виджета клавиатуры. Тогда

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

будет возможность не переключать фокус с виджета с текстовым блоком на клавиатуру, из-за чего сообщение между ними будет происходить автоматически [1].

### 3.2.9.3 Панель «Инструменты»

Панель «Инструменты» должна представлять из себя виджет с двумя вкладками: «Действия» и «Магазин». Вызываться данная панель должна по нажатию на кнопку Y левого контроллера Oculus Quest 2. Вкладка «Инструменты» должна предоставлять следующий перечень функций:

- 1) Создание (спавн) объектов еды для кота по нажатию на кнопку еды во вкладке «Действия» → «Кот» (см. рисунок 28). Пока кот ест, кнопки должны быть недоступны;

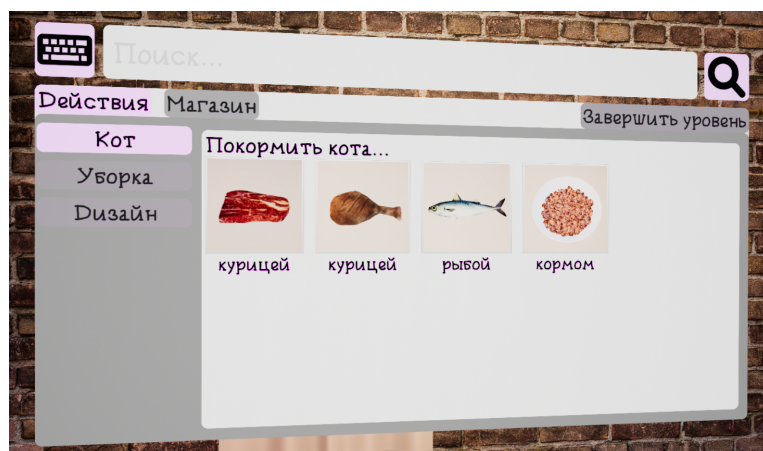


Рис. 28 – Спавн еды для кота

- 2) Вызов мусорного ведра на пляжном уровне во время уборки территории (см. рисунок 29);

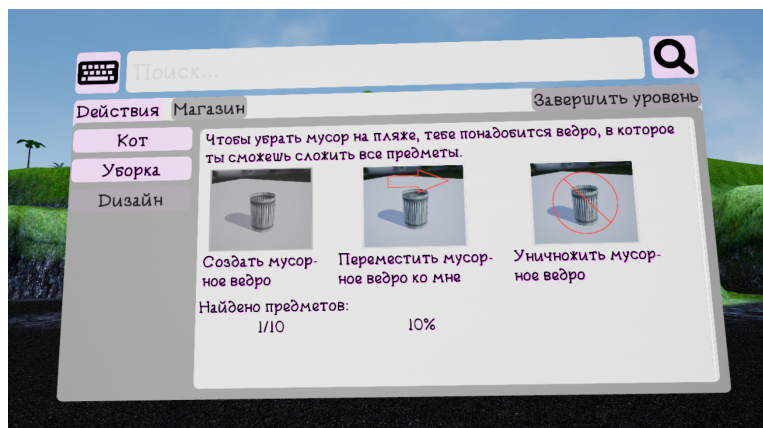


Рис. 29 – Мусорное ведро

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- 3) Телепортация мусорного ведра к игроку, если оно создано и находится в игровом мире (см. рисунок 29);
- 4) Уничтожение мусорного ведра, если оно создано и находится в игровом мире (см. рисунок 29);
- 5) Включение режима «Отделки» на пляжном уровне на этапе декорирования местности;
- 6) Включение режима «Продажи» на пляжном уровне на этапе декорирования местности.

Вкладка «Магазин» должна предоставлять следующий перечень функций:

- 1) Возможность просмотра списка доступных товаров;
- 2) Возможность просмотра информации о конкретном товаре;
- 3) Возможность выбора цвета товара, если у него есть вариации расцветки;
- 4) Создание объекта класса просматриваемого товара на уровне.

#### 3.2.9.4 Меню паузы

Пауза представляет из себя объект с виджетом, в котором находится несколько кнопок и надписей.

При открытой паузе игрок не должен иметь возможность использовать какие-либо функции игры, кроме выхода из меню паузы или взаимодействия с самим меню.

На пляжных уровнях пауза должна спавниться в мире по нажатию на кнопку В правого контроллера.

В офисе альтернативой паузы должна быть панель, всегда располагающаяся на стене напротив изначального положения пользователя (см. рисунок 30).

Панель паузы в офисе должна визуализировать имя, введенное пользователем во время прохождения обучения, количество денег пользователя, а также кнопки, представляющие следующий список действий:

1. Перезагрузка текущего уровня;
2. Открытие меню настроек. Меню настроек, в свою очередь, также является виджетом, предоставляющим следующие функции:
  1. Изменение громкости звуков;
  2. Изменение громкости музыки;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

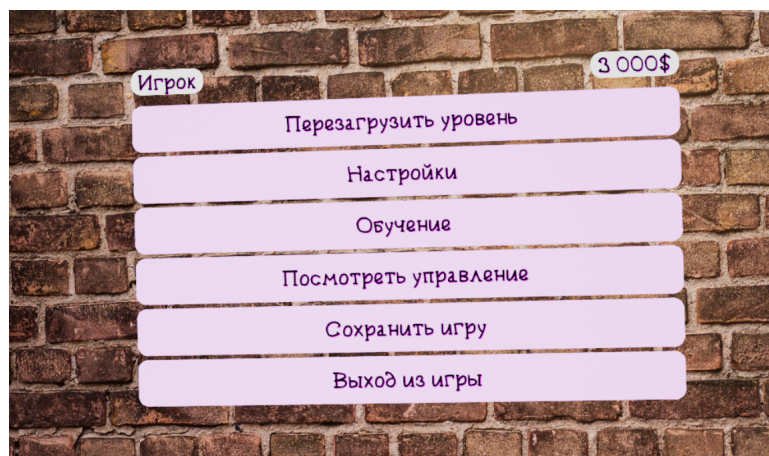


Рис. 30 – Пауза в офисе

3. Удаление текущего сохранения и запуск игры сначала.
3. Возможность повторного воспроизведения обучения;
4. Просмотр управления. По нажатию на данную кнопку должен происходить вывод виджета, в котором будет содержаться скролл-бар с функциями, предоставляемыми различными кнопками контроллера, а также картинка самих контроллеров. В случае прикосновения пользователя к различным кнопкам должно происходить цветовое выделение соответствующего элемента на картинке, а также выделение соответствующего текста в скролл-баре (см. рисунок 31);

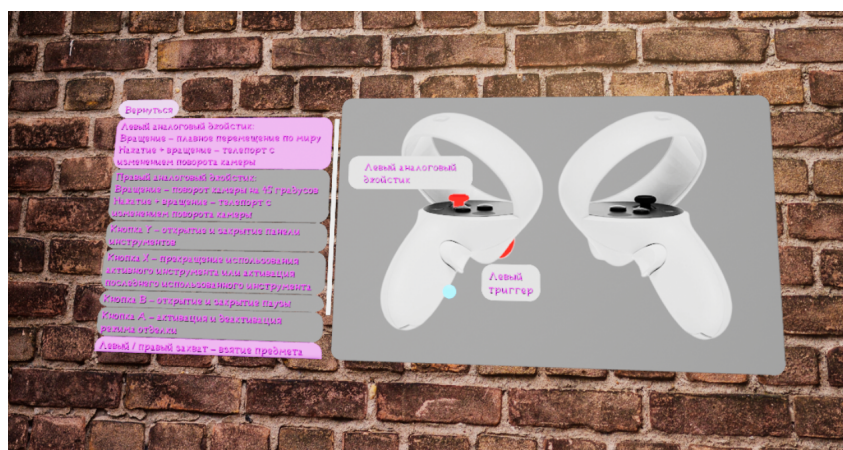


Рис. 31 – Панель для просмотра управления

5. Возможность сохранить игру. Нажатие на данную кнопку сохраняет текущие данные пользователя;
6. Возможность покинуть игру. Перед выходом также происходит сохранение данных пользова-

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

теля.

Меню паузы на пляжном уровне должно предоставлять следующий список функций:

1. Возврат всех процедурно сгенерированных и существующих на данный момент на пляжном уровне объектов на их стартовые позиции;
2. Возможность регенерировать уровень (при этом мусорные объекты будут сгенерированы заново и будут иметь уже другие классы и позиции в мире);
3. Открытие меню настроек. Меню настроек, в свою очередь, также является виджетом, предоставляющим следующие функции (см. рисунок 32):

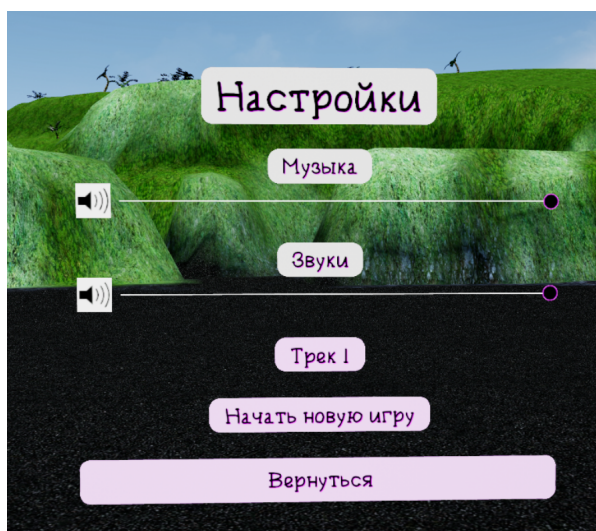


Рис. 32 – Настройки на пляжном уровне

1. Изменение громкости звуков;
2. Изменение громкости музыки;
3. Переключение текущего трека;
4. Удаление текущего сохранения и запуск игры сначала.
4. Возможность повторного воспроизведения обучения;
5. Просмотр управления. По нажатию на данную кнопку должен происходить вывод виджета, в котором будет содержаться скролл-бар с функциями, предоставляемыми различными кнопками контроллера, а также картинка самих контроллеров. В случае прикосновения пользователя к различным кнопкам должно происходить цветовое выделение соответствующего элемента на картинке, а также выделение соответствующего текста в скролл-баре;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

6. Возможность сохранить игру. Нажатие на данную кнопку сохраняет текущие данные пользователя и текущее состояние уровня;
7. Возможность вернуться с уровня в офис с сохранением данных пользователя и текущего состояния уровня;
8. Возможность покинуть игру. Перед выходом также происходит сохранение данных пользователя и данных об уровне.

### 3.2.9.5 Спавн объекта с виджетом

При вызове пользователем панели «Инструменты», меню паузы, при смене режима «отделки» и «продажи» происходит спавн объекта с виджетом в игровом мире по следующему алгоритму (см. алгоритм 1):

---

#### Алгоритм 1 Спавн объектов с виджетами

---

```

1: function SpawnObjectWithВиджет(ObjectToSpawn)
2:   Camera = GetPlayerCameraManager
3:   CameraVector = Camera.GetActorForwardVector
4:   CameraLocation = Camera.GetActorLocation
5:   Transform.Location.X = CameraVector.X · 120 + CameraLocation.X,
6:   Transform.Location.Y = CameraVector.Y · 120 + CameraLocation.Y,
7:   Transform.Location.Z = CameraLocation.Z
8:   Transform.Rotation.X = 0,
9:   Transform.Rotation.Y = 0,
10:  Transform.Rotation.Z = FindLookAtRotation(CameraLocation, ( CameraVector.X · 120 +
    CameraLocation.X, CameraVector.Y · 120 + CameraLocation.Y, 0 )).Z
11:  Transform.Scale = ( 1, 1, 1 )
12:  Spawn(ObjectToSpawn, Transform)
13: end function

```

---

Для того, чтобы заспавнить объект, нужно сделать следующее:

- 1) Получить ссылку на объект камеры игрока
- 2) Получить вектор направления камеры с помощью функции Get Actor Forward Vector (CameraVector)
- 3) Получить текущие координаты камеры с помощью функции Get Actor Location (CameraLocation)
- 4) Чтобы задать координаты, поворот и размер объекта, который нужно заспавнить, необходимо

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

проинициализировать переменную типа Transform. Она как раз будет содержать в себе координаты, поворот и размер объекта по осям X, Y и Z

- 5) Для того, чтобы проинициализировать координаты в переменной Transform, нужно умножить координаты X и Y CameraVector на 120, а затем сложить с текущими координатами X и Y CameraLocation. Координата Z передается из CameraLocation в Transform без изменений
- 6) Значения X и Y в Transform равны нулю. Для получения Z координаты необходимо вызвать метод Find Look At Rotation (Vector Start, Vector Target). В качестве первого аргумента подаются координаты CameraLocation, в качестве второго аргумента — координаты X, Y из Transform и Z = 0
- 7) Значения размера остаются по умолчанию равными 1 по всем осям
- 8) Вызывается метод Spawn Actor from Class, в который передается класс объекта для спавна и полученный в ходе выполнения предыдущих шагов Transform

### 3.2.10 Режим продажи

Режим продажи должен активироваться и деактивироваться на пляжном уроне по нажатию на кнопку А правого контроллера или по нажатию на соответствующую кнопку в панели «Инструменты». В случае, если пользователь нажал на кнопку контроллера и режим активировался или деактивировался, должна спавниться соответствующая надпись на 2 секунды. Данный режим должен быть доступен только в случае, если пользователь закончил уборку и приступил к декорированию, а также еще не завершил уровень.

В данном режиме должна быть возможность при наведении активного контроллера на объект подсветить его голубым цветом (см. рисунок 33), чтобы пользователь знал, на какой предмет наведен его контроллер, а также возможность подсветить зеленым цветом предмет (см. рисунок 34), который пользователь выделил нажатием на триггер активного контроллера. Поиск объекта должен происходить по методу бегущий лучей.

Когда предмет выделяется пользователем, над объектом должна спавниться надпись, спрашивающая пользователя о том, хочет ли он продать объект. Если пользователь выбирает ответ «да», то предмет должен уничтожиться, режим продажи деактивироваться, а сумма от продажи должна добавиться пользователю.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

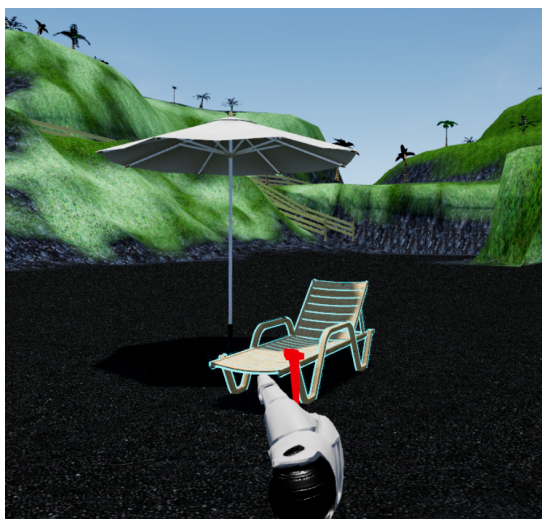


Рис. 33 – Наведение луча

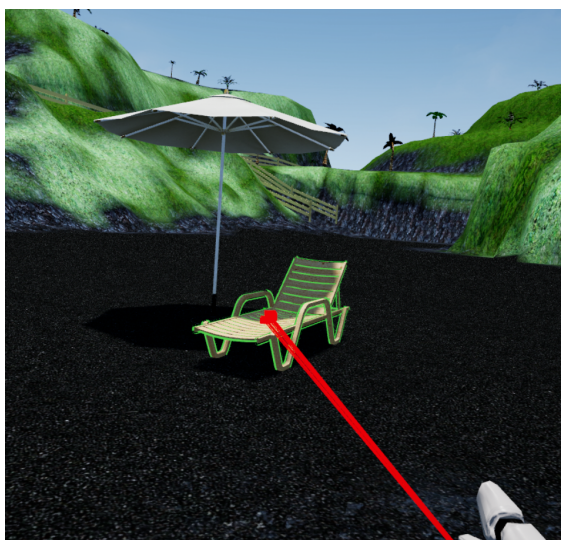


Рис. 34 – Выбор предмета

### 3.2.11 Режим отделки

Режим продажи должен активироваться и деактивироваться на пляжном уровне по нажатию на кнопку X левого контроллера или по нажатию на соответствующую кнопку в панели «Инструменты». В случае, если пользователь нажал на кнопку контроллера и режим активировался или деактивировался, должна спавниться соответствующая надпись на 2 секунды. Данный режим должен быть доступен только в случае, если пользователь закончил уборку и приступил к декорированию, а также еще не завершил уровень.

В данном режиме должна быть возможность при наведении активного контроллера на объ-

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ект подсветить его голубым цветом (см. рисунок 33), чтобы пользователь знал, на какой предмет наведен его контроллер, а также возможность подсветить зеленым цветом предмет (см. рисунок 34), который пользователь выделил нажатием на триггер активного контроллера. Поиск объекта должен происходить по методу бегущий лучей.

Когда предмет выделяется пользователем, должна быть возможность сменить его координаты X, Y в пространстве с помощью джойстика левого контроллера, а также сменить угол поворота по оси Z при помощи поворота джойстика правого контроллера.

### 3.2.12 Метод бегущих лучей

Метод бегущих лучей, или Ray casting, используется для определения наличия пересечения между объектом, от которого исходит луч, и другим объектом. Задается начальная точка луча и направление его движения, при наличии какого-либо объекта, удовлетворяющего проверяемым условиям, на пути луча, луч отправит информацию о том, с каким объектом он пересекается.

Начальной точкой луча в режимах «отделки» и «продажи» выступает активный контроллер. Пример того, что происходит при пересечении луча и предмета, купленного в магазине, представлен на рисунке 33.

### 3.2.13 Процедурная генерация мусора

После первого запуска любого пляжного уровня происходит процедурная генерация мусорных объектов.

Далее приведен псевдокод, описывающий данный метод:

---

#### Алгоритм 2 Процедурная генерация мусора

---

```

1: function CheckIfRubbishInCurrentRadius(Vector[] RubbishVectors, Vector PointToCheck)
2:   RubbishAlreadyExist = false
3:   for Rubbish in RubbishVectors do
4:     if IsPointInBox(PointToCheck, Rubbish, Vector(300, 300, 300)) then
5:       bool RubbishAlreadyExist = true
6:       break
7:     end if
8:   end for
9:   return RubbishAlreadyExist
10: end function

```

---

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

**Алгоритм 2** Процедурная генерация мусора. Продолжение

---

```

1: function GenerateNewRubbish(Integer NumberOfRubbishItems, Vector BoxSpawnCenter, Vector
  BoxSpawnExtent)
2:   for i = 0; i < NumberOfRubbishItems; i++ do
3:     Vector Point = RandomPointInBoundingBox(BoxSpawnCenter, Vector⟨BoxSpawnExtent.X -
      20, BoxSpawnExtent.Y - 20, BoxSpawnExtent.Z⟩)
4:     if CheckIfRubbishInCurrentRadius(RubbishVectors, Point) then
5:       i--
6:     else
7:       RubbishVectors.Add(Point)
8:       SpawnActorOfRandomRubbishClass()
9:       RubbishItemsArray.Add(⟨RubbishClass, RubbishActor.Transform⟩)
10:    end if
11:  return RubbishItemsArray
12: end function

```

---

На вход метода GenerateNewRubbish подается количество предметов, которое необходимо заспавнить на уровне (Integer NumberOfRubbishItem), а также границы спавна, представленные координатами центра (Vector BoxSpawnCenter) и вектором куба границ (Vector BoxSpawnExtent). В начале инициализируется массив RubbishVectors типа Vector, в которой позже будут присваиваться координаты уже заспавненных в мире мусорных объектов. В цикле for на каждой итерации генерируются координаты случайной точки в заданных границах, после чего вызывается вспомогательный метод CheckIfRubbishInCurrentRadius. Данный метод проверяет пересечение новой сгенерированной точки с уже заспавненными объектами. Если точка находится в радиусе 300 сантиметров любой уже существующей точки, индекс цикла for уменьшается на единицу и итерация происходит заново. Если точка успешно прошла проверку, то происходит выбор случайного класса мусорных объектов и выбор случайного значения поворота, после чего точка добавляется в массив RubbishVectors, а также класс сгенерированного мусорного объекта и его Transform добавляются в массив типа RubbishToSaveStruct. Данный массив возвращается методом и нужен для корректного сохранения положения мусорных объектов на уровне.

**3.2.14 Искусственный интеллект****3.2.14.1 NPC Кот**

Behaviour tree является графом, который занимается обработкой информации и служит для принятия решений и действия в соответствии с ними. В Unreal Engine 4 Behaviour tree также пред-

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ставляет собой граф, в котором разработчик создает ноды для дерева поведения. Основным отличием behavior tree в UE4 является то, что его выполнение продолжается с последнего выполненного нода, а не с корня дерева поведения. Ноды фактически являются действиями, которые прописываются при разработке дерева поведения. Ноды в дереве поведения делятся на четыре основных типа: composite, task, selection и sequence. Task представляет собой узел, который осуществляет определенное действие. В начале каждой ветви находится композит (composite), разные композиты выполняют ноды отличающимися способами. Совокупность нодов, которая находится под композитом, именуется поддеревом. Для последовательного выполнения задач в Behaviour tree в UE4 используется композит Sequence, которой выполняет дочерние ноды в строгом порядке, начиная с крайнего левого и заканчивая крайним правым. Если предыдущее действие не выполнено или не может быть выполнено, последующие действия выполняться не будут, то есть sequence просто перестанет выполняться. Возникают ситуации, когда необходимо объединить несколько поддеревьев, для выполнения данной задачи используется композит selector. Выделяя сходство selector'a и sequence'a отмечу, что действия и там, и там выполняются слева направо. Отличием Selector'a является то, что он останавливается при успешном выполнении дочерних задач, данный композит используется для выполнения только одного поддерева из нескольких.

Blackboard представляет собой панель, которая является хранилищем ключей. Под ключами понимаются переменные, в зависимости от которых будет выбрано то или иное действие [29].

На рисунке 35 представлено дерево поведения кота в локации «офис».

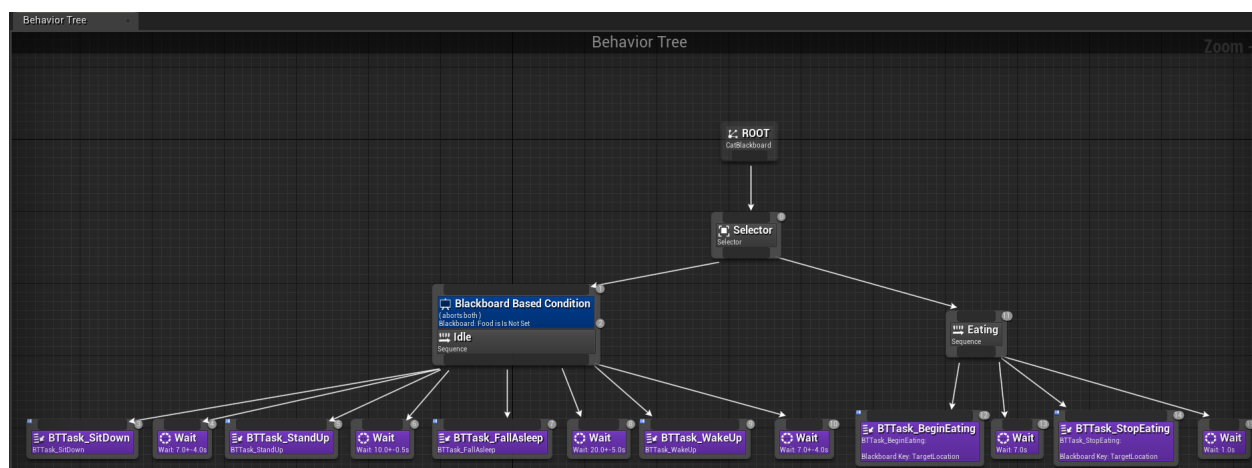


Рис. 35 – Office Cat Behaviour Tree

Корнем дерева является Root. Далее объединяются два поддерева, для чего используется композит selector. Первое поддерево представлено композитом типа sequence под названием «Idle». Вет-

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ви поддерева Idle можно перечислить в строгом порядке слева направо: BTTask\_SitDown (кот находится в состоянии «сидеть»), Wait (состояние бездействия в состоянии «сидеть»), BTTask\_StandUp (кот переходит в состояние «стоять»), Wait (состояние бездействия в состоянии «стоять»), BTTask\_Fall-Asleep (кот переходит в состояние «спать»), Wait (состояние бездействия в состоянии «спать»), BTTask\_WakeUp (кот переходит в состояние «стоять»), Wait (состояние бездействия в состоянии «стоять»). При появлении объекта еды происходит переход ко второму поддереву. В Blackboard расположено поле Food типа родительского класса еды. Когда объект спавнится в мире, в поле Food присваивается ссылка на него. В мире может только один объект класса еды. Второе поддерево представлено композитом типа sequence под названием «Eating». Ноды поддерева «Eating» могут быть описаны следующим образом в порядке слева направо: BTTask\_BeginEating (кот переходит в состояние «есть»), Wait (состояние бездействия в состоянии «есть»), BTTask\_StopEating (кот возвращается в состояние «стоять»), Wait (состояние бездействия в состоянии «стоять»). При отсутствии объекта еды после завершения второго поддерева, происходит переход к первому поддереву.

На рисунке 36 представлено дерево поведения кота на пляжном уровне.

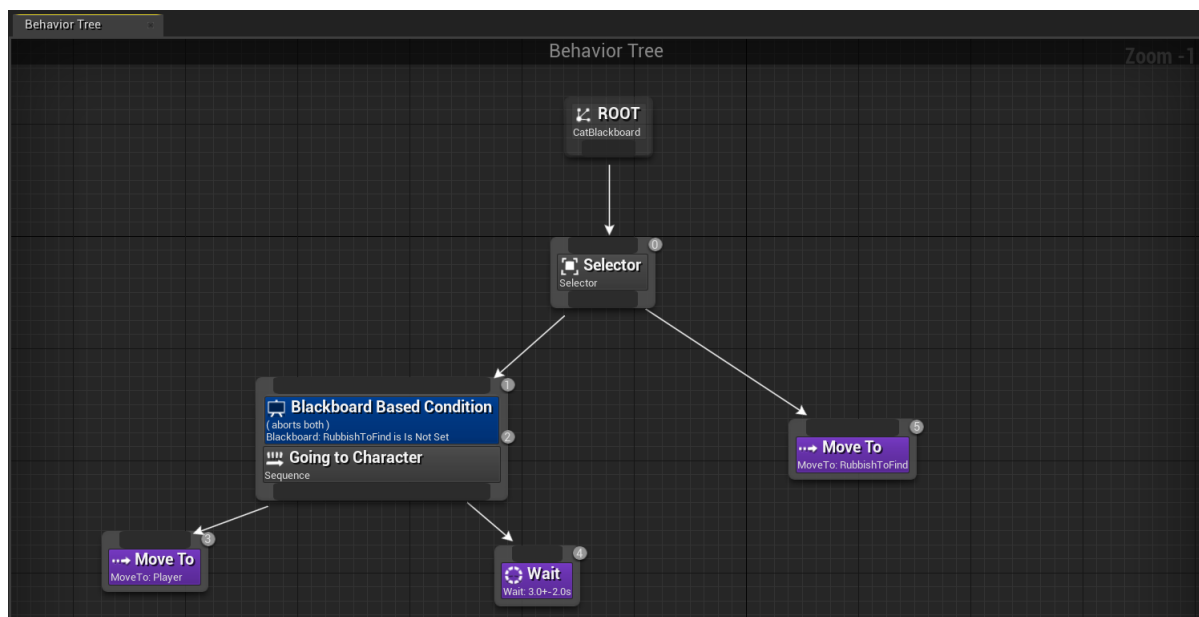


Рис. 36 – Beach Level Cat Behaviour Tree

Корнем дерева является Root. Далее объединяются два поддерева, используя композит selector. Первое поддерево представлено композитом типа sequence под названием «Going to Character». Поддеревом это композита являются следующие ноды (в порядке слева направо): Move to (кот движется к игроку), Wait (кот останавливается около игрока в состоянии «стоять»). Второе поддерево пред-

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ставлено единственным task'ом Move to (кот движется к мусорному объекту). Переход ко второму дереву происходит тогда, когда на определенном расстоянии от игрока появляется мусорный объект. В Blackboard расположено поле RubbishToFind типа родительского класса мусорного объекта. Когда пользователь заходит в Collision Box мусорного объекта, в поле RubbishToFind присваивается ссылка на него. В случае, если поле не null, ссылка не присваивается, что позволяет не менять фокус кота на другие объекты в случае перемещения пользователя по карте.

### 3.2.14.2 NPC Рыба

Behavior Tree поведения рыб представлено деревом с корнем Root. Далее находится поддерево, корнем которого является композит типа sequence под названием Idle, ветвями данного поддерева являются следующие ноды: wait (рыба находится в состоянии покоя), Move to (Рыба движется к случайной точке в радиусе 3000 от рыбы) (см. рисунок 37).

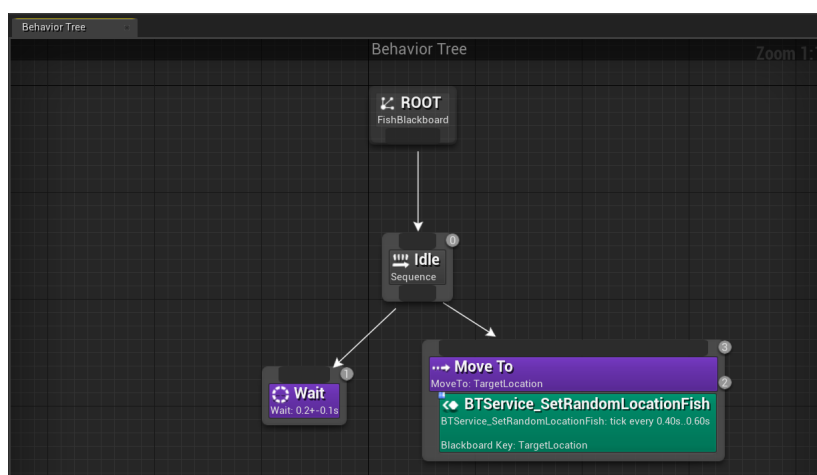


Рис. 37 – Fish Behaviour Tree

## 3.3 Описание и обоснование выбора метода организации входных и выходных данных

Входными данными разрабатываемого приложения являются:

- 1) Угол поворота VR-очков Oculus Quest 2;
- 2) Координаты VR-очков Oculus Quest 2 в пространстве;
- 3) Координаты контроллеров Oculus Quest 2 в пространстве.

Также входными данными являются взаимодействия со следующими элементами управления контроллеров Oculus Quest 2 (см. рисунок 38) [11]:

- 1) Вращение аналогового джойстика на левом контроллере Oculus Quest 2 (Left Thumstick);

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

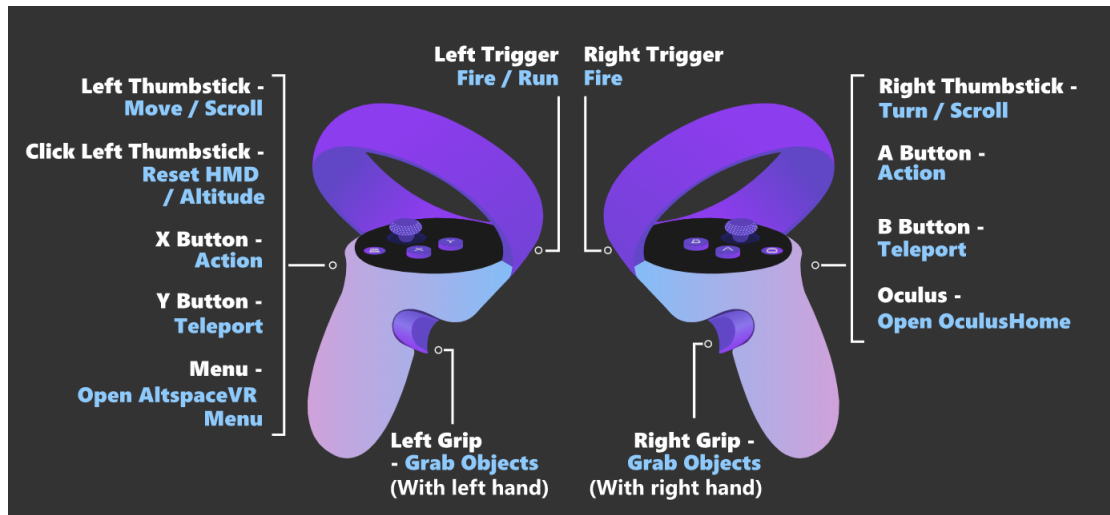


Рис. 38 – Управление Oculus

- 2) Нажатие на аналоговый джойстик на левом контроллере Oculus Quest 2 (Click Left Thumstick);
- 3) Прикосновение к аналоговому джойстику на левом контроллере Oculus Quest 2 (Touch Left Thumstick);
- 4) Нажатие на кнопку X левого контроллера Oculus Quest 2;
- 5) Прикосновение к кнопке X левого контроллера Oculus Quest 2;
- 6) Нажатие на кнопку Y левого контроллера Oculus Quest 2;
- 7) Прикосновение к кнопке Y левого контроллера Oculus Quest 2;
- 8) Нажатие на кнопку захвата предмета (под средним пальцем) на левом контроллере Oculus Quest 2 (Left Grip);
- 9) Нажатие на кнопку триггера (под указательным пальцем) на левом контроллере Oculus Quest 2 (Left Trigger);
- 10) Прикосновение к кнопке триггера (под указательным пальцем) на левом контроллере Oculus Quest 2 (Touch Left Trigger);
- 11) Вращение аналогового джойстика на правом контроллере Oculus Quest 2 (Right Thumstick);
- 12) Нажатие на аналоговый джойстик на правом контроллере Oculus Quest 2 (Click Right Thumstick);
- 13) Прикосновение к аналоговому джойстику на правом контроллере Oculus Quest 2 (Touch Right Thumstick);
- 14) Нажатие на кнопку A правого контроллера Oculus Quest 2;
- 15) Прикосновение к кнопке A правого контроллера Oculus Quest 2;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- 16) Нажатие на кнопку В правого контроллера Oculus Quest 2;
- 17) Прикосновение к кнопке В правого контроллера Oculus Quest 2;
- 18) Нажатие на кнопку «Окулус» правого контроллера Oculus Quest 2;
- 19) Нажатие на кнопку захвата предмета (под средним пальцем) на правом контроллере Oculus Quest 2 (Right Grip);
- 20) Нажатие на кнопку триггера (под указательным пальцем) на правом контроллере Oculus Quest 2 (Right Trigger).
- 21) Прикосновение к кнопке (под указательным пальцем) на правом контроллере Oculus Quest 2 (Touch Right Trigger).

Далее приведен список выходных данных разрабатываемого приложения:

- 1) Выходными данными при вращении VR-очков Oculus Quest 2 в реальном мире является соответствующее вращение камеры в игре;
- 2) Выходными данными при перемещении VR-очков Oculus Quest 2 в пространстве является соответствующее изменение положения игрока в игровом мире;
- 3) Выходными данными при перемещении любого из двух контроллеров Oculus Quest 2 в пространстве является соответствующее изменение положения контроллера в игровом мире;
- 4) Выходными данными при вращении аналогового джойстика на левом контроллере Oculus Quest 2 (Left Thumstick) является плавное передвижение игрока в игровом мире в соответствующую направлению джойстика сторону;
- 5) Выходными данными при нажатии на аналоговый джойстик на левом контроллере Oculus Quest 2 (Click Left Thumstick) является появление параболаобразной линии со стрелкой на конце в случае, если возможен телепорт в место, к которому повернут контроллер;
- 6) Выходными данными при вращении аналогового джойстика на левом контроллере Oculus Quest 2 (Left Thumstick) после нажатия на него (Click Left Thumstick) является изменение направления стрелки телепорта, если телепорт возможен;
- 7) Выходными данными при отпускании аналогового джойстика на левом контроллере Oculus Quest 2 (Left Thumstick) является телепортация к выбранному месту на карте с изменением поворота камеры, соответствующем повороту стрелки;
- 8) Выходными данными при прикосновении к аналоговому джойстику на левом контроллере Oculus Quest 2 (Touch Left Thumstick) является выделение красным цветом соответствующего элемента на изображении контроллеров в случае, если панель просмотра управления открыта;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- 9) Выходными данными при нажатии на кнопку X левого контроллера Oculus Quest 2 является активация режима «Продажи» в случае, если он был неактивен, и деактивация в случае, если был активен;
- 10) Выходными данными при прикосновении к кнопке X левого контроллера Oculus Quest 2 является выделение красным цветом соответствующего элемента на изображении контроллеров в случае, если панель просмотра управления открыта;
- 11) Выходными данными при нажатии на кнопку Y левого контроллера Oculus Quest 2 является открытие является активация панели «Инструменты» в случае, если она был неактивна, и деактивация в случае, если была активна;
- 12) Выходными данными при прикосновении к кнопке Y левого контроллера Oculus Quest 2 является выделение красным цветом соответствующего элемента на изображении контроллеров в случае, если панель просмотра управления открыта;
- 13) Выходными данными при нажатии на кнопку захвата предмета (под средним пальцем) на левом контроллере Oculus Quest 2 (Left Grip) является изменение анимации руки и захват предмета, находящегося в зоне контроллера в случае неактивности панели просмотра управления; иначе — выделение цветом соответствующего элемента на изображении контроллеров в панели просмотра управления;
- 14) Выходными данными при нажатии на кнопку триггера (под указательным пальцем) на левом контроллере Oculus Quest 2 (Left Trigger) является активация левого контроллера в случае, если контроллер был неактивен; «клик» на объект в случае, если контроллер активен; выделение объекта, на который наведен контроллер в данный момент, в режимах «Отделки» и «Продажи», если контроллер активен;
- 15) Выходными данными при прикосновении к кнопке триггера (под указательным пальцем) на левом контроллере Oculus Quest 2 (Touch Left Trigger) является выделение красным цветом соответствующего элемента на изображении контроллеров в случае, если панель просмотра управления открыта;
- 16) Выходными данными при вращении аналогового джойстика на правом контроллере Oculus Quest 2 (Right Thumstick) является перемещение камеры игрока в игровом мире в соответствующую направлению джойстика сторону на 45°;
- 17) Выходными данными при нажатии на аналоговый джойстик на правом контроллере Oculus Quest 2 (Click Right Thumstick) является появление параболообразной линии со стрелкой на

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- конец в случае, если возможен телепорт в место, к которому повернут контроллер;
- 18) Выходными данными при вращении аналогового джойстика на правом контроллере Oculus Quest 2 (Right Thumstick) после нажатия на него (Click Right Thumstick) является изменение направления стрелки телепорта, если телепорт возможен;
  - 19) Выходными данными при отпускании аналогового джойстика на правом контроллере Oculus Quest 2 (Right Thumstick) является телепортация к выбранному месту на карте с изменением поворота камеры, соответствующем повороту стрелки;
  - 20) Выходными данными при прикосновении к аналоговому джойстику на правом контроллере Oculus Quest 2 (Touch Right Thumstick) является выделение красным цветом соответствующего элемента на изображении контроллеров в случае, если панель просмотра управления открыта;
  - 21) Выходными данными при нажатии на кнопку А правого контроллера Oculus Quest 2 является активация режима «Отделки» в случае, если он был неактивен, и деактивация в случае, если был активен;
  - 22) Выходными данными при прикосновении к кнопке А правого контроллера Oculus Quest 2 является выделение красным цветом соответствующего элемента на изображении контроллеров в случае, если панель просмотра управления открыта;
  - 23) Выходными данными при нажатии на кнопку В правого контроллера Oculus Quest 2 является активация игрового меню паузы в случае, если она была неактивна, и деактивация в случае, если была активна;
  - 24) Выходными данными при прикосновении к кнопке В правого контроллера Oculus Quest 2 является выделение красным цветом соответствующего элемента на изображении контроллеров в случае, если панель просмотра управления открыта;
  - 25) Выходными данными при нажатии на кнопку «Oculus» (правый контроллер Oculus Quest 2) является открытие меню игровых очков Oculus Quest 2;
  - 26) Выходными данными при нажатии на кнопку захвата предмета (под средним пальцем) на правом контроллере Oculus Quest 2 (Right Grip) является изменение анимации руки и захват предмета, находящегося в зоне контроллера в случае неактивности панели просмотра управления; иначе — выделение цветом соответствующего элемента на изображении контроллеров в панели просмотра управления;
  - 27) Выходными данными при нажатии на кнопку триггера (под указательным пальцем) на правом

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

контроллере Oculus Quest 2 (Right Trigger) является активация левого контроллера в случае, если контроллер был неактивен; «клик» на объект в случае, если контроллер активен; выделение объекта, на который наведен контроллер в данный момент, в режимах «Отделки» и «Продажи», если контроллер активен;

- 28) Выходными данными при прикосновении к кнопке триггера (под указательным пальцем) на правом контроллере Oculus Quest 2 (Touch Right Trigger) является выделение красным цветом соответствующего элемента на изображении контроллеров в случае, если панель просмотра управления открыта.

Перечисленные виды входных и выходных данных были выбраны для разрабатываемой системы из-за специфики приложений в виртуальной реальности: пользователю нужно не только видеть игровой мир вокруг себя в очках, но еще и имитировать взаимодействие с ним с помощью рук, аналогом которых, в данном случае, являются контроллеры с несколькими кнопками разного вида на них. Поскольку кнопок на контроллерах значительно меньше, чем кнопок на клавиатуре компьютера, было принято решение не сокращать количество входных данных, а использовать одну и ту же кнопку для вызова разных функций при различных условиях.

### 3.4 Описание и обоснование выбора состава технических и программных средств

#### 3.4.1 Состав технических и программных средств

Для бесперебойной и корректной работы системы необходимо, чтобы в распоряжении у пользователя были:

- 1) VR-шлем с двумя контроллерами Oculus Quest 2, совместимый с платформой SteamVR;
- 2) ПК с установленным SteamVR;
- 3) Провод Oculus Link (не нужен, если осуществлено подключение VR-очков к ПК с помощью Air Link)

Требование к software VR-очков — активный режим Air Link/Oculus Link.

Минимальные требования, предъявляемые к software ПК для поддержки VR-очков Oculus Quest 2 [27]:

- Операционная система: Windows 10;
- Pixel шейдеры: 5.1;
- Vertex шейдеры: 5.1;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- Выделенная видео-память: 3 GB;
- Установленный SteamVR;
- Установленное приложение Oculus с активным сопряжением с очками пользователя.

Минимальные требования, предъявляемые к hardware ПК пользователя для поддержки VR-очков Oculus Quest 2 [27]:

- Процессор — Intel i5-4590 / AMD Ryzen 5 1500X или лучше;
- Оперативная память минимум 8 GB;
- Видеокарта — NVIDIA GeForce GTX 970 / AMD Radeon 400 Series или лучше;
- USB 3.0 порт (если не используется Air Link).

### 3.4.2 Обоснование выбора технических и программных средств

Выбор VR-очков в рамках разработки данного курсового проекта пал на Oculus Quest 2, поскольку это один из самых мощных недорогих шлемов на рынке. Он позволяет бесперебойно находиться в приложениях из библиотеки Steam'a даже при беспроводном подключении к ПК, камеры для отслеживания допустимых границ передвижения встроены в сам шлем, в то время как некоторые аналоги требуют расстановку отдельных камер в помещении перед началом работы.

Движок для разработки игр Unreal Engine 4 был выбран в рамках данного курсового проекта, поскольку он имеет встроенную поддержку большого количества VR-шлемов и для него есть достаточно широкая и подробная официальная документация от разработчиков движка.

Выбор движка также повлек за собой выбор языка программирования. Unreal Engine 4 предоставляет возможность писать на его собственном языке визуального программирования Blueprint, который позволяет упрощать создание игровой логики.

SteamVR как инструмент для использования виртуальной реальности был выбран для разработки и билда проекта, поскольку он позволяет не только собрать проект так, чтобы его можно было запускать с компьютера, но и позволяет тестировать приложение в VR-очках непосредственно из движка [12].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 4 ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

### 4.1 Ориентировочная экономическая эффективность

Данный курсовой проект не предусматривает расчет экономической эффективности.

### 4.2 Предполагаемая потребность

Предполагаемая потребность обуславливается тем, что у пользователей существует необходимость в удовлетворении эстетической и экологической потребностей [28]. Данное приложение также будет востребовано среди людей, желающих развивать свои навыки дизайна открытых природных пространств.

### 4.3 Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

На данный момент не существует прямых аналогов разрабатываемого продукта. В таблице 1 приведена сравнительная характеристика с приложениями, наиболее близкими к теме разрабатываемого проекта.

Таблица 1 – Аналоги

	«House Flipper» [7]	«Garden Flipper» [6]	«Home Improvisation: Furniture Sandbox» [5]	«Home Designer — Living Room» [4]	«Home Designer — Home Sweet Home» [3]	«Beach Designer»
Поддержка VR	—	—	+	—	—	+
Разнообразие используемых инструментов	+	—	+	—	—	—
Основная цель — декорирование	+	+	+	—	—	+
Основная цель — мини-игры	—	—	—	+	+	—
Работа с пляжами	—	—	—	—	—	+

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 5 ПЕРСПЕКТИВЫ РАЗВИТИЯ

Предполагается, что в будущем в проект будут добавлены следующие возможности:

- 1) Поддержка большего количества моделей VR-очков;
- 2) Добавление режима игры multiplayer, при котором на одной и той же локации смогут находиться несколько человек, видеть друг друга в игровом пространстве и совершать прохождение уровня на время;
- 3) Добавление большего количества средств при уборке территории (удаление грязи с объектов, уже находящихся на карте, продажа поврежденных объектов);
- 4) Увеличение количества товаров, которые можно поставить на пляжных уровнях;
- 5) Увеличение количества доступных локаций.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Creating Virtual Keyboard Interactions [Электронный ресурс] / Virtual Keyboard. Режим доступа: <https://docs.unrealengine.com/4.27/en-US/InteractiveExperiences/UMG/HowTo/VirtualKeyboards/>, свободный. (дата обращения: 15.02.2022).
- [2] Foliage [Электронный ресурс] / foliage. Режим доступа: <https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/Foliage/>, свободный. (дата обращения: 01.03.2022).
- [3] Home Designer - Home Sweet Home [Электронный ресурс] / Home Sweet Home. Режим доступа: [https://store.steampowered.com/app/1531610/Home\\_Designer\\_Home\\_Sweet\\_Home/](https://store.steampowered.com/app/1531610/Home_Designer_Home_Sweet_Home/), свободный. (дата обращения: 15.11.2021).
- [4] Home Designer - Living Room [Электронный ресурс] / Living Room. Режим доступа: [https://store.steampowered.com/app/1393640/Home\\_Designer\\_Living\\_Room/](https://store.steampowered.com/app/1393640/Home_Designer_Living_Room/), свободный. (дата обращения: 15.11.2021).
- [5] Home Improvisation: Furniture Sandbox [Электронный ресурс] / Home Improvisation. Режим доступа: [https://store.steampowered.com/app/357670/Home\\_Improvisation\\_Furniture\\_Sandbox/](https://store.steampowered.com/app/357670/Home_Improvisation_Furniture_Sandbox/), свободный. (дата обращения: 14.11.2021).
- [6] House Flipper - Garden DLC [Электронный ресурс] / Garden Flipper. Режим доступа: [https://store.steampowered.com/app/942290/House\\_Flipper\\_Garden\\_DLC/](https://store.steampowered.com/app/942290/House_Flipper_Garden_DLC/), свободный. (дата обращения: 14.11.2021).
- [7] House Flipper [Электронный ресурс] / House Flipper. Режим доступа: <https://steamcommunity.com/app/613100>, свободный. (дата обращения: 14.11.2021).
- [8] Level of Details [Электронный ресурс] / LODs. Режим доступа: <https://docs.unrealengine.com/4.26/en-US/WorkingWithContent/Types/StaticMeshes/HowTo/LODs/>, свободный. (дата обращения: 27.01.2022).
- [9] Lowpoly, Highpoly и вертекс нормали [Электронный ресурс] / Lowpoly. Режим доступа: <https://d.tf.ru/gamedev/73251-statya-3-7-pro-setku-lowpoly-highpoly-i-verteks-normali>, свободный. (дата обращения: 28.01.2022).
- [10] Navmesh Content Examples [Электронный ресурс] / navmesh. Режим доступа: <https://docs.unrealengine.com/4.27/en-US/Resources/ContentExamples/NavMesh/>, свободный. (дата обращения: 01.02.2022).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- [11] Oculus Quest/Quest 2 controls and permissions [Электронный ресурс] / oculus controls. Режим доступа: <https://docs.microsoft.com/en-us/windows/mixed-reality/alt-space-vr/getting-started/oculus-controls>, свободный. (дата обращения: 01.02.2022).
- [12] SteamVR [Электронный ресурс] / steamVR. Режим доступа: <https://store.steampowered.com/app/250820/SteamVR/>, свободный. (дата обращения: 20.01.2022).
- [13] UMG [Электронный ресурс] / umg. Режим доступа: <https://www.oreilly.com/library/view/unreal-engine-4/9781784391966/ch08.html>, свободный. (дата обращения: 10.01.2022).
- [14] Virtual Reality, VR [Электронный ресурс] / vr. Режим доступа: <https://www.it.ua/ru/knowledge-base/technology-innovation/virtualnaja-realnost-vr>, свободный. (дата обращения: 10.01.2022).
- [15] Volume зоны [Электронный ресурс] / volume зоны. Режим доступа: <https://uengine.ru/site-content/docs/actors-geometry/volumes>, свободный. (дата обращения: 25.01.2022).
- [16] Виртуальные очки [Электронный ресурс] / vr-очки. Режим доступа: <https://www.testprom.ru/artprod/elektronika/sertifikacziya-virtualnyx-ochkov>, свободный. (дата обращения: 24.10.2021).
- [17] ГОСТ 19.101–77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [18] ГОСТ 19.102–77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [19] ГОСТ 19.103–77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [20] ГОСТ 19.104–78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [21] ГОСТ 19.105–78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [22] ГОСТ 19.106–78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [23] ГОСТ 19.404–79 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [24] ГОСТ 19.603–78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- [25] ГОСТ 19.604–78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [26] Неигровой персонаж [Электронный ресурс] / прс. Режим доступа: [https://ru.wikipedia.org/wiki/Неигровой\\_персонаж](https://ru.wikipedia.org/wiki/Неигровой_персонаж), свободный. (дата обращения: 09.02.2022).
- [27] Системные требования Oculus Quest 2 [Электронный ресурс] / oculus quest 2 requirements. Режим доступа: <https://ru.pickgamer.com/games/oculus-quest-2/requirements>, свободный. (дата обращения: 27.01.2022).
- [28] Тимошенко, Т.В. Экология и потребности человека [Электронный ресурс]: Известия ЮФУ. Технические науки — Режим доступа: <https://cyberleninka.ru/article/n/ekologiya-i-potrebnosti-cheloveka/viewer>, свободный. (дата обращения: 31.01.2022).
- [29] Тьюториал по Unreal Engine. Часть 9: Искусственный интеллект [Электронный ресурс] / ИИ. Режим доступа: <https://habr.com/ru/post/346498/>, свободный. (дата обращения: 27.02.2022).
- [30] Что такое Oculus Quest 2 [Электронный ресурс] / Oculus Quest 2. Режим доступа: <https://tech.onliner.by/2021/11/11/vpechatleniya-ot-oculus-quest-2>, свободный. (дата обращения: 21.01.2022).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 1

### ТЕРМИНОЛОГИЯ

Таблица 1 – Терминология

Термин	Значение
Виртуальная реальность (Virtual Reality, VR)	«Созданный техническими средствами мир, передаваемый человеку через его ощущения: зрение, слух, осязание и другие. Виртуальная реальность имитирует как воздействие, так и реакции на воздействие. Для создания убедительного комплекса ощущений реальности компьютерный синтез свойств и реакций виртуальной реальности производится в реальном времени» [14]
Очки виртуальной реальности (VR-очки, VR-шлем)	«Специализированный прибор, способный симулировать разнообразные аудиовизуальные трехмерные пространства. Состоит из пластикового (реже картонного) корпуса, экрана с перегородкой и асферических линз, которые фокусируют изображение» [16]
Контроллер	Устройство ввода информации, используемое в видеоиграх или развлекательных системах для получения входных данных. Обычно используется для управления объектом или персонажем в игре
Oculus Quest 2	«Полностью автономный VR-шлем, выпущенный компанией Oculus, которая входит в состав Meta» (Meta является признанной на территории России экстремистской организацией) [30]
NPC (Non-Player Character, Неигровой персонаж)	«Персонаж в играх, который не находится под контролем игрока» [26]
Low-Poly	Трёхмерная модель с малым количеством полигонов (примерно 5 — 10 тысяч) [9]
Mid-Poly	Трёхмерная модель со средним количеством полигонов
Nav Mesh	«Игровой объект, использующийся для определения зоны передвижения игрока и для вычисления навигационных путей для Искусственного Интеллекта на уровне» [15]
Nav Mesh Bounds Volume	«Игровой объект, использующийся для определения зоны расчета Nav Mesh на уровне» [15]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 2

### ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ

Таблица 2 – Описание и функциональное назначение классов

Класс	Назначение
ItemToBuyStruct	Структура, описывающая товар, который пользователь может приобрести в магазине
ShopItemToSaveStruct	Структура, содержащая в себе ту информацию о купленном товаре, которую необходимо поместить в сохранение
BP_ItemToBuy_Parent	Родительский класс, описывающий товар, который пользователь может приобрести в магазине
BP_BeachChairWood_Child	Класс, наследующий BP_ItemToBuy_Parent и представляющий собой пляжный лежак
BP_Umbrella_Child	Класс, наследующий BP_ItemToBuy_Parent и представляющий собой пляжный зонт
BP_ShopItemsSpawner	Класс для работы с товарами, которые пользователь может приобрести в магазине
RubbishToSaveStruct	Структура, содержащая в себе ту информацию о мусорном объекте, которую необходимо поместить в сохранение
BP_RubbishGenerator	Класс для работы с мусорными объектами, генерируемыми на пляжных уровнях
LevelStatusEnum	Перечисление, содержащее в себе статусы, которые могут быть у пляжного уровня
ModeEnum	Перечисление, содержащее в себе все режимы, в которых может находиться игрок
PlayerSaveData	Структура, описывающая сохраняемые данные пользователя
LevelSaveData	Структура, описывающая сохраняемые данные пляжного уровня
VR_SaveGame_Object	Класс, наследуемый от Blueprint класса SaveGame и содержащий в себе информацию, нужную для сохранения игры
VR_GameInstance	Класс, существующий на протяжении всей игровой сессии и не уничтожающийся при смене игровых уровней
VR_GameMode	Класс, наследуемый от Blueprint Game Mode Base и содержащий в себе общие правила игры
VR_GameState	Класс, наследуемый от Blueprint Game State Base и содержащий в себе логику оповещения других классов в случае совершения каких-либо действий, связанных с правилами игры
VR_PlayerState	Класс, содержащий в себе информацию об игроке
BP_MotionControllerPawn	Класс, описывающий поведение игрока

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 2 – Описание и функциональное назначение классов

Класс	Назначение
BP_MotionController	Класс, описывающий поведение виртуальной руки, или игрового контроллера
GripEnum	Перечисление, описывающее состояние контроллера, связанное со взятием и отпусканьем предметов
PickupActorInterface	Интерфейс, описывающий взятие и отпускание предмета контроллером
BP_TrashParent	Класс, описывающий мусорный предмет, с которым может взаимодействовать пользователь
BP_Can_Child	Класс, наследуемый от BP_TrashParent и представляющий из себя жестяную банку из-под напитка
BP_CoffeeCup_Child	Класс, наследуемый от BP_TrashParent и представляющий из себя стаканчик из-под кофе
BP_EmptyPlasticBottle_Child	Класс, наследуемый от BP_TrashParent и представляющий из себя пустую пластиковую бутылку
BP_GlassBottle_Child	Класс, наследуемый от BP_TrashParent и представляющий из себя стеклянную бутылку
BP_Matches_Child	Класс, наследуемый от BP_TrashParent и представляющий из себя коробок со спичками
BP_MilkBottle_Child	Класс, наследуемый от BP_TrashParent и представляющий из себя бутылку из-под молока
BP_TrashBar_Child	Класс, наследуемый от BP_TrashParent и представляющий из себя деревянную доску
BP_TrashBox1_Child	Класс, наследуемый от BP_TrashParent и представляющий из себя один из видов картонных коробок
BP_TrashBox2_Child	Класс, наследуемый от BP_TrashParent и представляющий из себя один из видов картонных коробок
BP_TrashBox3_Child	Класс, наследуемый от BP_TrashParent и представляющий из себя один из видов картонных коробок
BP_RubbishBin	Класс, представляющий из себя мусорный бак
BP_Tablet	Класс, представляющий из себя планшет пользователя
CatCharacter	Класс, представляющий из себя кота
AI_CatBP	Класс, представляющий из себя искусственный интеллект кота
BeachCatBT	Класс, представляющий из себя Behavior Tree и описывающий поведение кота на пляжных уровнях
RoomCatBT	Класс, представляющий из себя Behavior Tree и описывающий поведение кота в офисе
CatBlackboard	Класс, представляющий из себя Blackboard для BeachCatBT и RoomCatBT
BTTask_FallAsleep	Класс задачи, выполняемого в Behavior Tree кота и представляющего из себя переход в состояние сна

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 2 – Описание и функциональное назначение классов

Класс	Назначение
BTTask_SitDown	Класс задачи, выполняемого в Behavior Tree кота и представляющего из себя переход в состояние «сидеть»
BTTask_StandUp	Класс задачи, выполняемого в Behavior Tree кота и представляющего из себя переход в состояние «стоять»
BTTask_StopEating	Класс задачи, выполняемого в Behavior Tree кота и представляющего из себя переход из состояния «есть» в состояние бездействия
BTTask_WakeUp	Класс задачи, выполняемого в Behavior Tree кота и представляющего из себя переход в состояние бездействия из состояния сна
BP_CatsFoodParent	Класс, представляющий из себя еду для кота
BP_CatsFoodChicken-Leg	Класс, наследуемый от класса BP_CatsFoodParent и представляющий из себя куриную ножку
BP_CatsFoodFish	Класс, наследуемый от класса BP_CatsFoodParent и представляющий из себя рыбу
BP_CatsFoodMeat	Класс, наследуемый от класса BP_CatsFoodParent и представляющий из себя кусок мяса
BP_CatsWetFood	Класс, наследуемый от класса BP_CatsFoodParent и представляющий из себя влажный кошачий корм
BP_FishCharacter	Класс, представляющий из себя рыбу
AI_Fish	Класс, представляющий из себя искусственный интеллект рыбы
FishBT	Класс, представляющий из себя Behavior Tree и описывающий поведение рыбы
FishBlackboard	Класс, представляющий из себя Blackboard для FishBT
BTService_SetRandomLocationFish	Класс сервиса, выполняемого в Behavior Tree рыбы и представляющего из себя следование к случайной точке пространства
CatFindWidget	Виджет, представляющий из себя слово «Предмет», появляющееся над котом, когда он находит какой-либо мусорный объект
ControlsDescription-Widget	Виджет, представляющий из себя панель управления
VRControls	Объект для спавна ControlsDescriptionWidget в игровом мире
FlyingTextWidget	Виджет, представляющий из себя текст, который нужно заставить в качестве «летающего» сообщения
BP_FlyingText	Объект для спавна FlyingTextWidget в игровом мире
KeyboardWidget	Виджет, представляющий из себя виртуальную клавиатуру
LetterSizeMode	Перечисление для регистра букв на клавиатуре
VRKeyboard	Объект для спавна KeyboardWidget в игровом мире
LoadingWidget	Виджет, представляющий из себя загрузочный экран
VRLoading	Объект для спавна LoadingWidget в игровом мире
MessageWidget	Виджет для вывода текста на экран
VR_Message	Объект для спавна MessageWidget в игровом мире
SellItemWidget	Виджет для диалогового окна о продаже объектов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 2 – Описание и функциональное назначение классов

Класс	Назначение
VR_SellItemMessage	Объект для спавна SellItemWidget в игровом мире
BeachPause	Виджет для паузы на пляже
OfficePause	Виджет для паузы в офисе
SettingsWidget	Виджет для меню настроек
VRBeachPause	Объект для спавна BeachPause в игровом мире
VROfficePause	Объект для спавна OfficePause в игровом мире
VRSettings	Объект для спавна SettingsWidget в игровом мире
TabletWidget	Виджет для экрана планшета
WaianapanapaBeachWidget	Виджет для пляжа Waianapanapa
PapakoleaBeachWidget	Виджет для пляжа Papakolea
ToolbarWidget	Виджет панели «Инструменты»
VRToolbar	Объект для спавна ToolbarWidget в игровом мире
StartWidget1	Виджет, представляющий из себя первую вкладку в панели обучения
StartWidget2	Виджет, представляющий из себя вторую вкладку в панели обучения
StartWidget3	Виджет, представляющий из себя третью вкладку в панели обучения
StartWidget4	Виджет, представляющий из себя четвертую вкладку в панели обучения
StartWidget5	Виджет, представляющий из себя пятую вкладку в панели обучения
StartWidget6	Виджет, представляющий из себя шестую вкладку в панели обучения
StartWidget7	Виджет, представляющий из себя седьмую вкладку в панели обучения
StartWidget8	Виджет, представляющий из себя восьмую вкладку в панели обучения
BP_StartDialogue	Объект, представляющий из себя обучение и хранящий в себе виджеты StartWidget1-StartWidget8

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 3

# ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ КЛАССОВ

Таблица 3 – Структура ItemToBuyStruct

Поля			
Имя	Модификатор доступа	Тип	Назначение
ItemName	public	Name	Название товара
Cost	public	Integer	Изначальная стоимость товара
Description	public	Text	Описание товара
TagsToPrint	public	Map<Name, String>	Словарь названий тегов по-английски (ключ) и по-русски (значение)
ItemStatic-Mesh	public	StaticMesh	Статическая сетка товара
Image	public	Texture 2D	Картинка для отображения товара в каталоге магазина

Таблица 4 – Структура ShopItemToSaveStruct

Поля			
Имя	Модификатор доступа	Тип	Назначение
ShopItem-Class	public	BP_Item-ToBuy-Parent Class	Класс товара
Material	public	Material Interace	Материал товара
FinalCost	public	Integer	Окончательная стоимость товара
Transform	public	Transform	Координаты, поворот и размер товара, на которых его нужно будет заспавнить

Таблица 5 – Класс BP\_ItemToBuy\_Parent

Поля			
Имя	Модификатор доступа	Тип	Назначение
ItemStruct	public	ItemToBuy-Struct	Структура, описывающая конкретный товар из магазина
FinalCost	public	Integer	Окончательная цена за товар с учетом его материала

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 5 – Класс BP\_ItemToBuy\_Parent

ZCoordinate	public	Float	Z-координата, которую нужно присвоить объекту для его корректного спавна на карте	
Motion-Controller-Pawn	public	Motion-Controller-Pawn	Ссылка на объект класса MotionControllerPawn	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Change-FinalCost	public	void	Integer CostDif	Метод, высчитывающий финальную стоимость товара

Таблица 6 – Класс BP\_ShopItemsSpawner

Поля				
Имя	Модификатор доступа	Тип	Назначение	
BP_Spawn-Helper	public	BP_Spawn-Helper	Объект вспомогательного класса для создания объектов на конкретном Streaming Level	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Spawn-Existing-Shop-Items	public	ShopItem-ToSave-Struct[]	—	Справн товаров, полученных из сохранения, для повторного начала уровня
Create-Shop-ItemsAr-ray	public	void	RubbishTo-SaveStruct[] ShopItems	Метод для получения списка классов и расположений всех существующих на карте мусорных объектов

Таблица 7 – Структура RubbishToSaveStruct

<b>Поля</b>			
<b>Имя</b>	<b>Модификатор доступа</b>	<b>Тип</b>	<b>Назначение</b>
Rubbish	public	BP_Trash-Parent Class	Класс мусорного объекта
Transform	public	Transform	Координаты, поворот и размер мусорного, которые нужно будет присвоить объекту при спавне

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 8 – Класс BP\_RubbishGenerator

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Rubbish-Classes	public	BP_Trash-Parent[]	Массив ссылок на дочерние классы класса BP_TrashParent для выбора случайного класса генерируемого мусорного объекта	
Rubbish-Vectors	public	Vector[]	Массив векторов, хранящих расположение уже созданных на карте мусорных объектов	
NumberOf-AllRubbish	public	Integer	Количество всех мусорных объектов на уровне	
NumberOf-FoundRubbish	public	Integer	Количество найденных пользователем мусорных объектов	
BP_Spawn-Helper	public	BP_Spawn-Helper	Объект вспомогательного класса для создания объектов на конкретном Streaming Level	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Generate-NewRubbish	public	Rubbish-ToSave-Struct	Integer NumberOf-Rubbish, Vector Box-SpawnCenter, Vector Box-SpawnExtent	Метод для случайной генерации мусорных объектов на уровне.
Check-Rubbish-Current-Radius	public	Boolean	Vector Point-ToCheck	Метод для проверки того, что выбранная случайная точка пространства не находится в радиусе 300 сантиметров от уже существующих мусорных объектов
Change-Rubbish-Positions-ToStart	public	void	—	Метод для возвращения всех существующих на уровне мусорных объектов на их стартовые позиции
Spawn-Existing-Rubbish	public	Rubbish-ToSave-Struct[]	—	Справн мусорных объектов, полученных из сохранения, для повторного начала уровня
Create-Existing-Rubbish-Array	public	void	RubbishTo-SaveStruct[] RubbishItems	Метод для получения списка классов и расположений всех существующих на карте мусорных объектов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 9 – Класс VR\_GameInstance

Поля				
Имя	Модификатор доступа	Тип	Назначение	
LevelName	public	Name	Название текущего загруженного уровня	
SaveSlot-Name	public	String	Название файла с сохранением игры	
SaveGame-Data	public	VR_Save-GameOb-ject	Объект класса VR_SaveGameObject, нужный для со-хранения информации об игре	
Sound-Volume	public	Float	Значение громкости звуков	
Music-Volume	public	Float	Значение громкости музыки	
FirstStart	public	Boolean	Новая ли игра	
Beach-TrackNum-ber	public	Integer	Текущий музыкальный трек на пляжном уровне	
LevelStatus	public	LevelSta-tusEnum	Статус текущего игрового уровня	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
SavePlay-erData	public	void	—	Сохранение данных пользователя
LoadPlay-erData	public	void	—	Загрузка сохраненных данных
Clear-SaveData	public	void	—	Удаление текущего сохранения
LoadCat-Color	public	Material-Interface CatColor	—	Загрузка выбранной пользователем окраски кота из сохранения

Таблица 10 – Класс VR\_PlayerState

Поля			
Имя	Модификатор доступа	Тип	Назначение
Player-Name	public	Name	Имя игрока
Money	public	Integer	Количество денег игрока
Mode	public	ModeEnum	Текущий режим игрока

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 10 – Класс VR\_PlayerState

Поля			
Имя	Модификатор доступа	Тип	Назначение
IsTelepor-ting	public	Boolean	Телепортируется ли игрок в данный момент
IsPaused	public	Boolean	Включена ли пауза
IsBeachLe-vel	public	Boolean	Является ли текущий уровень пляжным
MotionSpe- ed	public	Float	Скорость движения игрока
Rotation- Angle	public	Float	Угол поворота камеры
DefaultPla- yerHeight	public	Float	Рост игрока
JoystickDe- adzone	public	Float	Слепая зона джойстика
DistanceTo- Widget	public	Float	Расстояние до виджета
PawnTrans- form	public	Transform	Координаты, поворот и размер Pawn'a
ControlsPa- nelsOpened	public	Boolean	Открыта ли панель просмотра управления
ShopItem- Rotation- Angle	public	Float	Угол поворота товара из магазина
ShopItem- MotionSpe- ed	public	Float	Скорость перемещения товара из магазина

Таблица 11 – Класс MotionControllerPawn

Поля			
Имя	Модификатор доступа	Тип	Назначение
LeftCont- roller	public	BP_Moti- onControl- ler	Ссылка на объект класса, отвечающий за левый кон- троллер
RightCont- roller	public	BP_Moti- onControl- ler	Ссылка на объект класса, отвечающий за правый контроллер
RightStick- Down	public	Boolean	Опущен ли правый джойстик вниз

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 11 – Класс MotionControllerPawn

Поля				
Имя	Модификатор доступа	Тип	Назначение	
LeftStick-Down	public	Boolean	Опущен ли левый джойстик вниз	
UseControllerRoll-ToRotate	public	Boolean	Используется ли ролик контроллера для поворота	
UseLeft-RightHand	public	Integer	Индекс активного котроллера	
IsWorking-WithTablet	public	Boolean	Используется ли планшет пользователем в данный момент	
Hovered-ShopItem	public	BP_Item-ToBuy_Parent	Товар на пляжном уровне, на который в данный момент указывает контроллер пользователя	
Current-ShopItem	public	BP_Item-ToBuy_Parent	Товар на пляжном уровне, который выделен пользователем на данный момент	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
GetRotationFromInput	public	Rotator	Float UpAxis, Float RightAxis, BP_Motion-Controller Controller	Получение поворота джойстика во время телепортации
InteractionLocation	public	void	BP_Motion-Controller Hand	Получение текущего положения компонента Widget Interaction
Update-WidgetInteraction-HitLocation	public	void	—	Обновление видимости и положения интерактивной сферы взаимодействия с виджетами
Activate-Widget-Button-Pressed	public	void	Integer HandNumber	Нажатие на триггер контроллера активной руки
Activate-Widget-ButtonReleased	public	void	Integer HandNumber	Отпускание триггера контроллера активной руки

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 11 – Класс MotionControllerPawn

<b>Методы</b>				
<b>Имя</b>	<b>Модификатор доступа</b>	<b>Тип</b>	<b>Аргументы</b>	<b>Назначение</b>
Move-mentBy-Control-lerLeft	public	void	Float Axis	Движение игрока за счет поворота джойстика по оси X
Move-mentBy-Control-lerForward	public	void	Float Axis	Движение игрока за счет поворота джойстика по оси Y
CanRe-moveMoney	public	Boolean	Integer AmountOfMoney	Метод, проверяющий, можно ли совершить покупку на определенную сумму денег
Remove-Money-Function	public	Boolean	Integer AmountOfMoney	Метод для вычитания денег со счета пользователя
AddMoney	public	void	—	Метод для добавления денег на счет пользователя
BuyItem	public	void	—	Метод для покупки товара из магазина
SellItem	public	void	—	Метод для продажи товара из магазина
IsMove-mentPointInNav-Mesh	public	Boolean	Vector MovementVector, Float Speed	Метод, проверяющий, можно ли передвинуться в точку пространства
SetUp-Player-Height	public	void	—	Метод, выставяющий высоту «глаз» игрока в игровом мире
CanOpen-Toolbar	public	Boolean	—	Метод, проверяющий, можно ли в данный момент открыть панель «Инструменты»
Check-ShopItem-ByLine-Trace	public	BP_MotionController Controller		

Таблица 12 – Класс CatCharacter

<b>Поля</b>			
<b>Имя</b>	<b>Модификатор доступа</b>	<b>Тип</b>	<b>Назначение</b>
IsEating	public	Boolean	Ест ли кот в данный момент

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 12 – Класс CatCharacter

Поля				
Имя	Модификатор доступа	Тип	Назначение	
IsSitting	public	Boolean	Сидит ли кот в данный момент	
IsSleeping	public	Boolean	Спит ли кот в данный момент	
IsPetting	public	Boolean	Гладит ли пользователь кота в данный момент	
IsRight-HandPet-ting	public	Boolean	Гладит ли пользователь кота в данный момент пра-вой рукой	
IsLeftHand-Petting	public	Boolean	Гладит ли пользователь кота в данный момент левой рукой	
IsFinding	public	Boolean	Ищет ли кот предмет на данный момент	
Finding-Rubbish	public	BP_Trash-Parent	Объект класса мусора, который кот в данный мо-мент ищет	
IsBeachLe-vel	public	Boolean	Пляжный ли уровень открыт на данный момент	
CatTrans-form	public	Transform	Изначальные координаты, поворот и размер кота	
AICat	public	AI_CatBP	Объект класса искусственного интеллекта кота	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Charac-terBegin-Overlap-WithRub-bish	public	void	—	Метод, вызывающийся в момент, когда пользователь заходит в область мусор-ного объекта
Spawn-FindRub-bishSound	public	void	—	Метод, спавнящий звук мяукания кота
RotateFin-dingRub-bishText	public	void	—	Метод для поворота надписи «пред-мет» над головой кота в момент, когда он идет к конкретному мусорному объ-екту

Таблица 13 – Класс AI\_CatBP

Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Character-Begin-	public	void	BP_TrashParent Rubbish	Метод, инициализирующий объект класса BP_TrashParent Rubbish в

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 13 – Класс AI\_CatBP

Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Rubbish-Overlap				Blackboard кота

Таблица 14 – Класс BP\_Start\_Dialogue

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Current-Widget	public	Integer	Номер открытого на данный момент виджета	
VR_Game-Instance	public	VR_Game-Instance	Ссылка на VR_GameInstance	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Change-Widgets-Visibility	public	void	SceneComponent Widget-Component, Boolean MakeVisible	Смена режима видимости элемента WidgetComponent на значение, равное MakeVisible

Таблица 15 – Класс LevelStatusEnum

Поля			
Имя	Модификатор доступа	Тип	Назначение
Blocked	—	—	Уровень недоступен для прохождения
CanStart	—	—	Уровень доступен для прохождения
RubbishGenerated	—	—	Этап уборки мусора с пляжа
Decorating	—	—	Этап декорирования пляжа
Finished	—	—	Уровень завершен

Таблица 16 – Класс GripEnum

Поля			
Имя	Модификатор доступа	Тип	Назначение
Open	—	—	Контроллер бездействует
CanGrab	—	—	Контроллер может взять предмет
Grab	—	—	Контроллер находится в состоянии взятия предмета

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

[illegible]