



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ

«Робототехника и комплексная автоматизация»

КАФЕДРА

«Системы автоматизированного проектирования (РК-6)»

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
***К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ***  
***НА ТЕМУ:***  
**«Разработка системы “цифровой аватар” на Unreal**  
**Engine»**

Студент РК6-82Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

**Эвоян Э. Б.**  
(Фамилия И.О.)

Руководитель ВКР

\_\_\_\_\_  
(Подпись, дата)

**Витюков Ф.А.**  
(Фамилия И.О.)

Нормоконтролёр

\_\_\_\_\_  
(Подпись, дата)

**Грошев С.В.**  
(Фамилия И.О.)

2023 г.

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

УТВЕРЖДАЮ

Заведующий кафедрой РК6  
(Индекс)

\_\_\_\_\_ Карпенко А.П.  
(Фамилия И.О.)

« \_\_\_\_ » \_\_\_\_\_ 2023 г.

**ЗАДАНИЕ**  
**на выполнение выпускной квалификационной работы бакалавра**

Студент группы РК6-82Б

Эвоян Эрик Борисович  
(фамилия, имя, отчество)

---

Тема квалификационной работы: «Разработка системы “цифровой аватар” на Unreal Engine»

Источник тематики (НИР кафедры, заказ организаций и т.п.): кафедра.

Тема квалификационной работы утверждена распоряжением по факультету РК №\_\_ от « \_\_\_\_ »  
\_\_\_\_\_ 2023 г.

**Техническое задание**

**Часть 1. Аналитическая часть**

Изучить полный метод создания 3d-моделей с помощью Meta Human Creator. Изучить средства разработки, предоставляемые трехмерным движком Unreal Engine. Провести продуктивное исследование методов считывания мимики с лица человека. Провести анализ методов трансляции объектов в Unreal Engine.

**Часть 2. Практическая часть 1. Создание небольшого ролика**

Создать небольшой ролик с анимированием персонажей, созданных при помощи Meta Human Creator. Наложить мимику человека на лицо персонажей.

**Часть 3. Практическая часть 2. Создание цифрового аватара**

Разработать систему трансляции цифрового аватара. Разработать метод связи Unreal Engine и OBS Studio.

## Оформление выпускной квалификационной работы

Расчетно-пояснительная записка на 61 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.):

<i>Работа содержит 7 графических листов формата А4.</i>

Дата выдачи задания: «10» февраля 2023 г.

В соответствии с учебным планом выпускную квалификационную работу выполнить в полном объеме в срок до «11» июня 2023 г.

Руководитель квалификационной работы

\_\_\_\_\_  
(Подпись, дата)

**Витюков Ф.А.**

(Фамилия И.О.)

Студент

\_\_\_\_\_  
(Подпись, дата)

**Эвоян Э. Б..**

(Фамилия И.О.)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

**Министерство науки и высшего образования Российской Федерации**  
**Федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**«Московский государственный технический университет имени Н.Э. Баумана**  
**(национальный исследовательский университет)»**  
**(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ РК

КАФЕДРА РК6

ГРУППА РК6-82Б

УТВЕРЖДАЮ

Заведующий кафедрой РК6  
(Индекс)

\_\_\_\_\_ Карпенко А.П.  
(Фамилия И.О.)  
«\_\_» \_\_\_\_\_ 2023 г.

**КАЛЕНДАРНЫЙ ПЛАН**  
**выполнения выпускной квалификационной работы**

Студент группы РК6-82Б

Эвоян Эрик Борисович  
(фамилия, имя, отчество)

Тема квалификационной работы: «Разработка системы “цифровой аватар” на Unreal Engine»

№ п/п	Наименование этапов выпускной квалификационной работы	Сроки выполнения этапов		Отметка о выполнении	
		план	факт	Должность	ФИО, подпись
1.	Задание на выполнение работы. Формулирование проблемы, цели и задач работы	10.02.2023	_____	Руководитель ВКР	Витюков Ф.А.
2.	1 часть. Аналитическая часть	18.02.2023	_____	Руководитель ВКР	Витюков Ф.А.
3.	Утверждение окончательных формулировок решаемой проблемы, цели работы и перечня задач	28.02.2023	_____	Заведующий кафедрой	Карпенко А.П.
4.	2 часть. Практическая часть 1	21.04.2023	_____	Руководитель ВКР	Витюков Ф.А.
5.	3 часть. Практическая часть 2	23.05.2023	_____	Руководитель ВКР	Витюков Ф.А.
6.	1-я редакция работы	28.05.2023	_____	Руководитель ВКР	Витюков Ф.А.
7.	Подготовка доклада и презентации	04.06.2023	_____	Студент	Котельникова Е.Ю.
8.	Заключение руководителя	10.06.2023	_____	Руководитель ВКР	Витюков Ф.А.
9.	Допуск работы к защите на ГЭК (нормоконтроль)	16.06.2023	_____	Нормоконтролер	Грошев С.В.
10.	Внешняя рецензия	19.06.2023	_____		
11.	Защита работы на ГЭК	21.06.2023	_____		

Студент \_\_\_\_\_  
(подпись, дата)

Эвоян Э.Б.  
(Фамилия И.О.)

Руководитель ВКР \_\_\_\_\_  
(подпись, дата)

Витюков Ф.А.  
(Фамилия И.О.)

## РЕФЕРАТ

Работа посвящена разработке системы “цифровой аватар”, предназначенной для использования в прямых трансляциях и фильмографии. Также рассматривается уровень производительности FPS при разных уровнях детализации 3D-модели. В данной работе рассмотрены и проиллюстрированы следующие этапы: прохождение полного цикла наложения мимики человека на 3D-модель, подготовка и импорт моделей в движок Unreal Engine, расположение моделей в сидящей позе, наложение озвучки на персонажа. Также была рассмотрена прямая трансляция модели с анимированной мимикой в Twitch.

Тип работы: выпускная квалификационная работа.

Тема работы: «Разработка системы “цифровой аватар” на Unreal Engine».

Объекты исследований: 3d-моделирование, разработка методов наложения мимики, разработка методов трансляции 3D-модели.

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

**Unreal Engine (UE)** – игровой движок, разрабатываемый и поддерживаемый компанией Epic Games.

**3d-model (3d-модель)** – математическое представление объекта в трехмерном пространстве.

**3d-modeling (3d-моделирование)** – процесс создания 3d-модели объекта.

**Actor (актёр)** – в рамках движка UE любой объект, который может быть размещен на уровне. Базовый класс всех actor'ов – AActor

**Graphics Processing Unit (GPU)** – графический процессор (видеокарта).

**Frames per second (FPS)** – количество кадров в секунду.

**Asset (ассет)** – в контексте компьютерных игр, объект, представляющий собой единицу контента. Игровыми ассетами являются 3d-модели, текстуры, материалы, аудиофайлы, и т.д. Как правило, созданием ассетов занимаются художники, дизайнеры, музыканты.

**LOD (Level of Detail, уровни детализации)** – техника, позволяющая подменять разные по детализации версии модели в зависимости от дистанции между камерой и объектом, либо в зависимости от процента площади экрана, занимаемой моделью.

**Vtube (Virtual Youtuber)** – набор приложений позволяющий транслировать 3D-модель с наложенной на неё в прямом эфире мимикой человека.

**Lip synchronization (синхронизация губ)** – элемент анимации, при котором движения губ совпадают с речью персонажа.

## СОДЕРЖАНИЕ

РЕФЕРАТ .....	5
ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ .....	6
ВВЕДЕНИЕ .....	9
1. Обзор Meta Human Creator .....	11
1.1 Инструменты для считывания мимики лица .....	11
2. Необходимые плагины для Unreal Engine с использованием Live Link Face	12
2.1 Необходимые плагины для Unreal Engine с использованием Faceware Studio .....	13
2.2 Плюсы и минусы Live Link Face и Faceware Studio .....	14
3. Обзор Quixel Bridge .....	15
3.1 Перенос 3D-модели из Quixel Bridge в проект Unreal Engine .....	16
4. Наложение мимики на Mesh персонажа .....	19
4.1 Подключение iPhone к Unreal Engine .....	19
4.2 Анимация мимики лица .....	20
4.3 Перенос персонажа на сцену .....	23
5. Исправление ошибки с отсутствием волос на лице персонажа .....	25
5.1 Отсутствие волос из-за не прорисовки .....	25
5.2 Отсутствие волос из-за отсутствия нужных плагинов .....	25
5.3 Отсутствие волос из-за ошибки прорисовки .....	27
5.4 Отсутствие волос из-за ошибки импорта файлов .....	29
6. Озвучка персонажа .....	33
6.1. Преобразование текста в речь с помощью Replica .....	33
6.2 Преобразование текста в речь с помощью xVASynth .....	37
6.3 Lip sync персонажа с помощью MetahumanSDK .....	38

6.4 Создание анимации по аудиозаписи .....	39
6.5 Наложение анимации на персонажа .....	42
7.Rigging персонажа.....	44
7.1 Расположение персонажа с помощью Meta Human Control Rig .....	46
8. Оборудование для работы с анимацией.....	48
9. Сравнение производительности разных моделей.....	50
10. MetaHuman для Vtube .....	55
10.1 Установка OBS и необходимых плагинов к Unreal Engine .....	55
10.2 Подключение трансляции из Unreal Engine к OBS .....	57
10.3 Трансляция Цифрового аватара в Twitch .....	59
ЗАКЛЮЧЕНИЕ .....	60
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	61



## ВВЕДЕНИЕ

При создании систем для Vtube одной из главных задач является достижение удобства потенциального пользователя при взаимодействии с конечным продуктом. Это удобство состоит из множества аспектов. В контексте прямых трансляций, разработки игр и фильмографий одним из таких аспектов может являться простота и скорость наложения мимики на 3D-модель.

При анимации персонажа одним из самых сложных элементов является его лицо. Делать анимацию лица вручную очень сложно и долго поэтому этот процесс необходимо оптимизировать и ускорить

В данной работе рассматривается разработка системы позволяющей как создавать фильмографические продукты, так и проводить прямые трансляции с цифровыми аватарами.

### *Часть первая. Создание небольшого ролика*

На сегодняшний день 3D-графиков много используется в современных фильмах, а в некоторых случаях и полностью состоит из них. В таком случае, удобно иметь систему, которая автоматизировала бы большую часть процесса или нуждалась бы в минимальном количестве актёров.

В рамках первой части работы рассмотрены способы наложения мимики на персонажа с помощью сторонних приложений, генерация речи для персонажа с применением нейросетей и автоматизирование наложение Lip sync на персонажа.

### *Часть вторая. Создание цифрового аватара*

На просторах интернета множество различных медийных личностей, которые используют 3D-эффекты для привлечения аудитории или иных целей. В большинстве случаев это приложения, которые имеют ограниченный функционал или не могут транслироваться в прямом эфире.

В рамках второй части работы рассмотрена разработка системы, которая позволяет накладывать мимику на персонажа Meta Human из Unreal Engine и транслировать её в Twitch и иные платформы.

# **1. Обзор Meta Human Creator**

Meta Human Creator — это облачное приложение, предназначенное для создания цифрового человека в реальном времени менее чем за час без ущерба для качества [1].

Разработчики могут создавать своих собственных уникальных MetaHumans и загружать их для непосредственного использования в Unreal Engine. В качестве альтернативы они могут использовать приложение DCC, такое как Autodesk Maya, для дальнейшего редактирования.

Более того можно использовать уже готовые модели размещённые на Quixel Bridge, что и будет сделано в процессе научно-исследовательской работы. Это существенно облегчит процесс и позволит не тратить время на создание новой 3D-модели так как нам нужно непосредственно наложить анимацию на готовый mesh.

## **1.1 Инструменты для считывания мимики лица**

В результате анализа возможных подходов к считыванию мимики были изучены два метода: с помощью Live Link Face или Faceware Studio. Эти два приложения есть в свободном доступе, но у каждого из них есть свои преимущества и недостатки как в требовании к аппаратной части, так и в финансовой доступности этих приложений. Оба ПО будут рассмотрены в данной научно-исследовательской работе.

## **2. Необходимые плагины для Unreal Engine с использованием Live Link Face**

Live Link Face позволяет использовать камеру iPhone 13 для считывания мимики с лица и переносить её в режиме прямой трансляции на mesh персонажа в Unreal Engine 4. К тому же это приложение доступно бесплатно в AppStore. После того как мы убедились, что приложение считывает мимику нам необходимо соединить наш iPhone вместе с Unreal Engine 4 через Live Link Face. Для этого нам нужно в самом приложении указать IPv4 нашего компьютера. Также очень важно, чтобы iPhone и компьютер находились в одной сети.

Для того, чтобы можно было использовать Live Link Face нам необходимо подключить ряд необходимых плагинов в проекте Unreal Engine 4.

Плагины от компании Apple:

- Apple ARKit
- Apple ARKit Face Support

Плагины от компании Epic Games:

- Live Link
- Control Rig
- Live Link Control Rig
- RigLogic Plugin

Стоит отметить, что все вышеупомянутые плагины актуальны для метода, использующего Live Link Face так как для Faceware Studio требуются другие настройки [7].

## 2.1 Необходимые плагины для Unreal Engine с использованием Faceware Studio

Faceware Studio также как и Live Link Face позволяет считывать мимику с лица человека. Это приложение является самостоятельным, но Unreal Engine 4 имеет поддержку этого приложения и поэтому его можно использовать для анимации вместе с MetaHumanCreator.

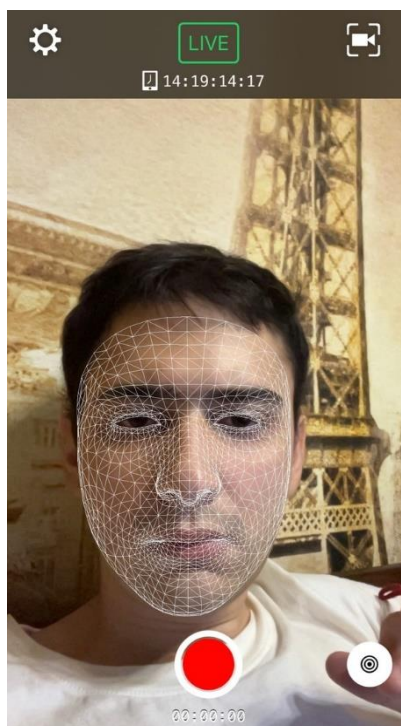


Рисунок 1. Лицо, с которого считывается мимика

Для того, чтобы можно было использовать данное ПО нам необходимо подключить лишь один плагин Faceware Live Link в проекте Unreal Engine.

Стоит отметить, что Faceware Studio является менее предпочтительной для маленьких студий и разработчиков-одиночек. Хотя программа и не требует устройств помимо ПК, она является требовательной к аппаратной части компьютера. Для нормальной поддержки FPS записи требуется мощная видеокарта также нужна хорошая камера, так как без неё считывание мимики может искажаться, приводя к “кривлянию” лица персонажа. На ноутбуке, в котором проводилась запись с Faceware Studio стоит встроенная камера из-за чего качество считывания оставляла желать лучшего.

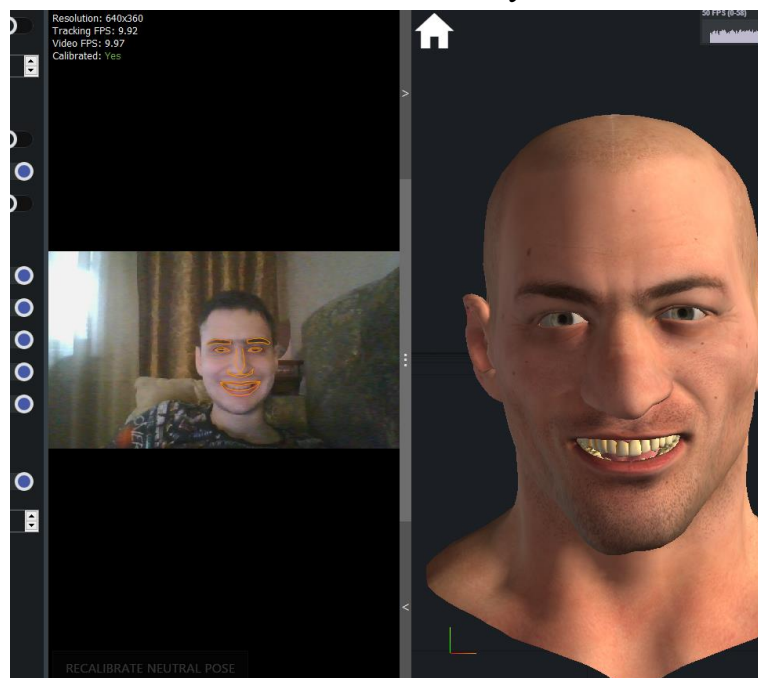


Рисунок 2. Считывание мимики с лица в Faceware Studio

## 2.2 Плюсы и минусы Live Link Face и Faceware Studio

Итак, сравнив две программы для анимации лица можно выделить преимущества и недостатки этих ПО:

Faceware Studio

Плюсы:

- Не требует сторонних устройств помимо ПК
- Довольно простое для использования
- Не конфликтует с антивирусом

Минусы:

- Платное ПО
- Требовательна к аппаратной части
- Будет работать некорректно без хорошей камеры

Live Link Face:

Плюсы:

- Бесплатная программа
- Доступна для установки через AppStore
- Требуется лишь камера от iPhone

Минусы:

- Требуется iPhone

Итак, взвесив плюсы и минусы обеих программ, для нашей работы в рамках научно-исследовательской работы была выбрана программа Live Link Face так как она является менее требовательной к аппаратной части.

### **3. Обзор Quixel Bridge**

Quixel Bridge — это приложение, которое служит вспомогательным инструментом для просмотра, поиска, загрузки, импорта и экспорта различных материалов Quixel Megascans. Оно абсолютно бесплатно и за ассеты не нужно платить даже, если вы используете их в коммерческой деятельности, но с условием того, что используете их в своём проекте Unreal Engine

### 3.1 Перенос 3D-модели из Quixel Bridge в проект Unreal Engine

Для того, чтобы перенести уже готового персонажа из Quixel Bridge для начала надо подключить расширение Meta Humans через Epic Games Store. Далее необходимо создать проект в Unreal Engine используя следующие настройки:

- Выбрать пункт Film, Television and Live Events в меню выбора проекта

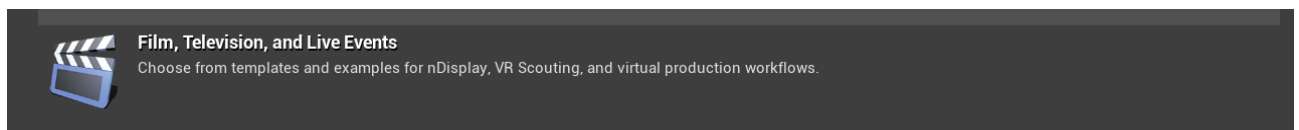


Рисунок 3. Тип проекта Film, Television and Live Events

- Выбрать пункт Blank в следующем меню
- В последнем меню надо выбрать нужен ли в проекте стартовый контент и ray tracing, а также расположение файла проекта



Рисунок 4. Пункт Blank

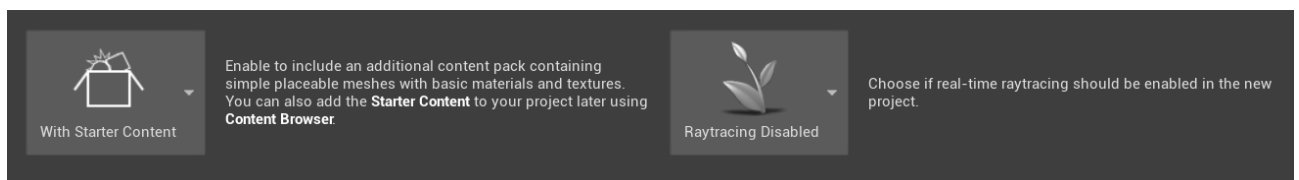


Рисунок 5. Выбор последних настроек



После того как проект был создан можно импортировать самого персонажа из Quixel Bridge в Unreal Engine. Для этого необходимо открыть созданный нами ранее проект и запустить Quixel Bridge. Далее нам необходимо зарегистрироваться в самом приложении, чтобы можно было осуществить импорт. Осуществить регистрацию можно как создав новый аккаунт в самом Quixel так и используя уже готовый аккаунт Epic Games Store.

После регистрации необходимо открыть приложение. При запуске должна открыться главная страница.

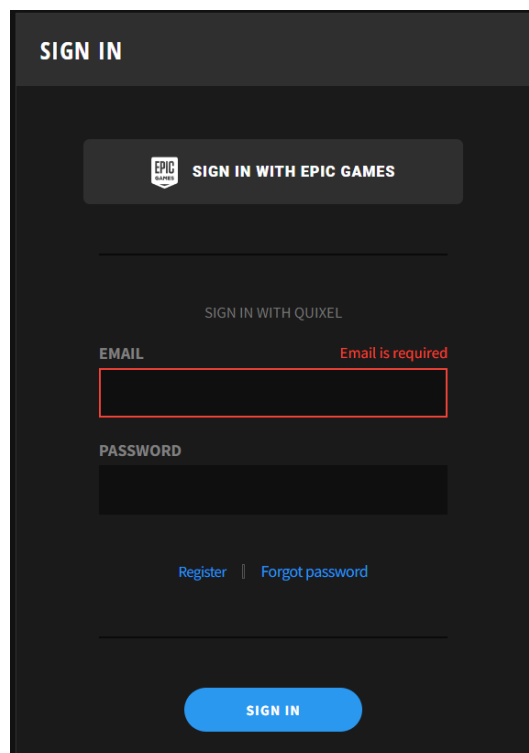


Рисунок 6. Меню регистрации в Quixel Bridge.

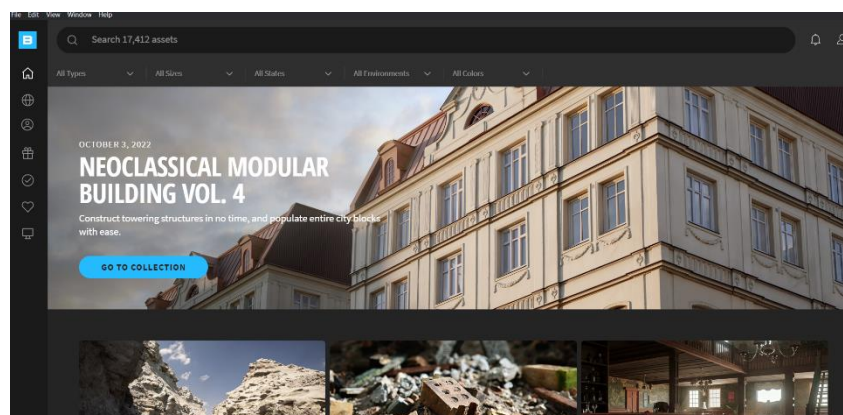


Рисунок 7. Меню Quixel Bridge.

После открытия нужно перейти на вкладку MetaHumans. Она расположена на левой панели и является третьей сверху вкладкой.

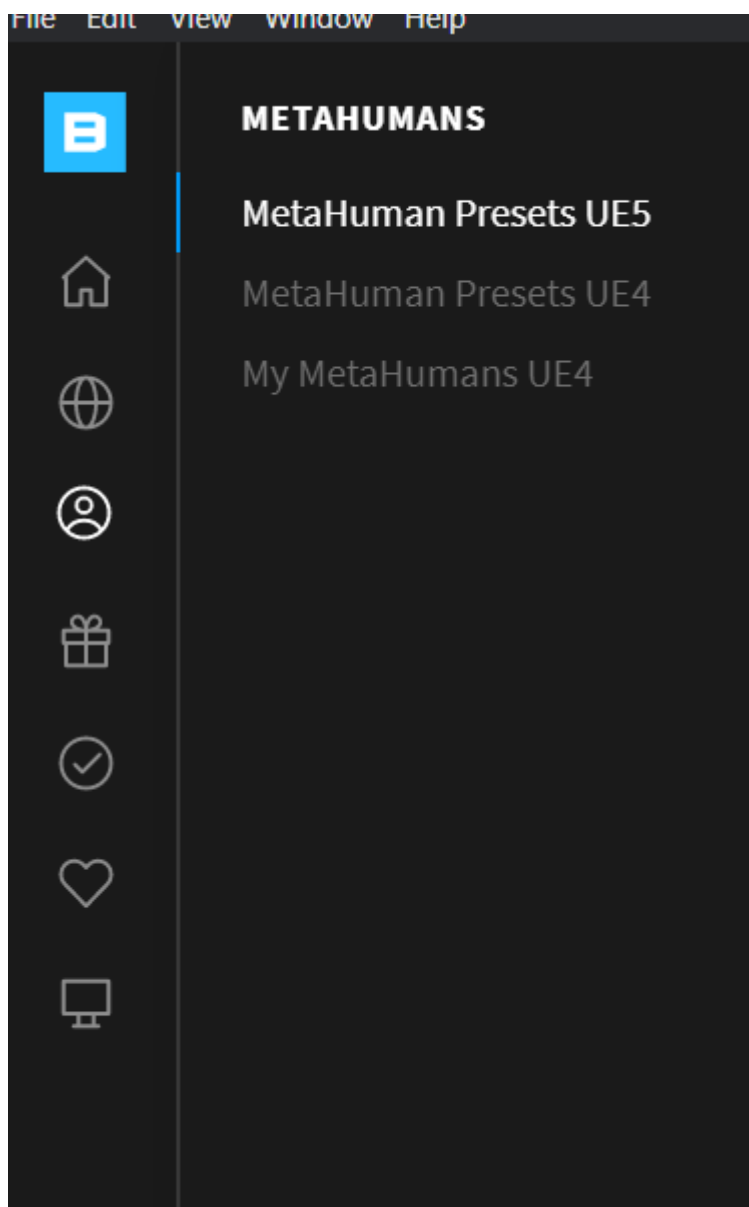


Рисунок 8. Вкладка MetaHumans

Далее надо выбрать версию MetaHumans в зависимости от версии Unreal Engine. После выбора версии можно перейти к выбору 3D-модели персонажа, которого мы хотим перенести в свой проект Unreal Engine.

После выбора нужного нам персонажа надо нажать кнопку “Export” в правом нижнем углу. Когда модель экспортируется в проект можно будет приступить непосредственно к анимации лица персонажа.

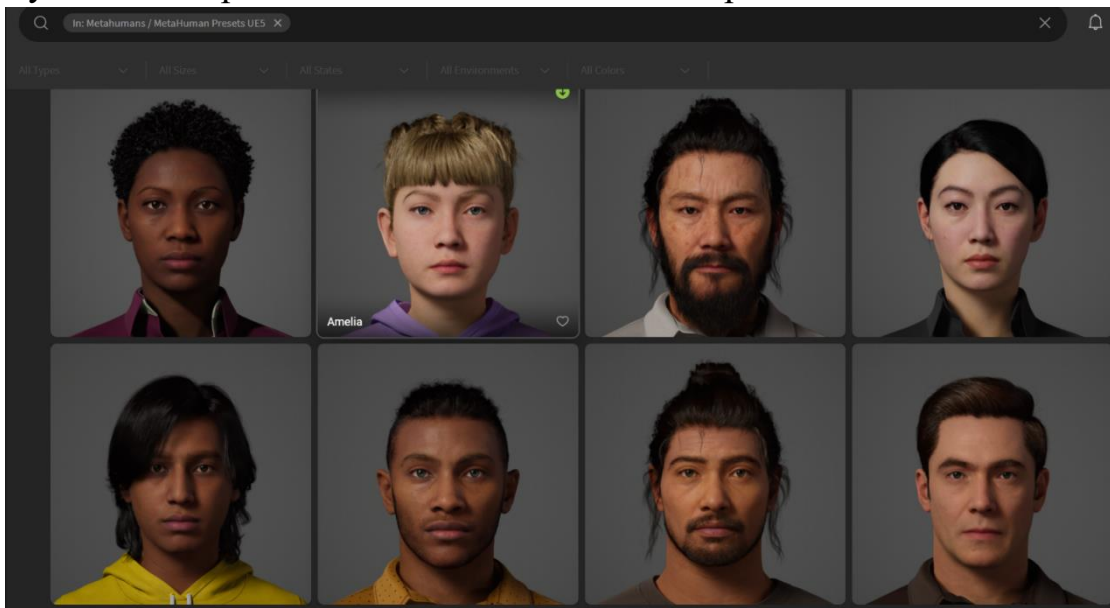


Рисунок 9. Меню выбора модели для импорта в проект

## 4. Наложение мимики на Mesh персонажа

Когда все плагины подключены, проект создан, и персонаж экспортирован в Unreal Engine, можно перейти непосредственно анимации мимики лица персонажа. Для осуществления этой задачи будет использовано приложение Live Link Face по ранее упомянутым причинам. Итак, приступим к реализации задачи.

### 4.1 Подключение iPhone к Unreal Engine

Для начала надо подключить iPhone к самому Unreal Engine. Для этого необходимо открыть Live Link Face в iPhone и зайти в настройки приложения нажав на шестерёнку в левом верхнем углу. Далее нужно зайти в графу Live Link для последующего создания пары между iPhone и Unreal Engine 4.[3]

В графе Subject Name необходимо дать имя нашему iPhone, которое будет видеть Unreal Engine при поиске нашего устройства. После этого нужно указать IPv4 адрес ПК, с которым необходимо сделать пару. Для того, чтобы узнать IPv4 устройства нужно зайти в настройки ПК и перейти во вкладку “Сеть и интернет” и там выбрать либо Wi-Fi, либо Ethernet в зависимости от того, к чему подключен компьютер. В этих вкладках можно узнать IPv4 адрес. После этого можно указать адрес в графе “Targets” в Live Link Face на iPhone. После выполнения всех этих операций можно перейти к наложению мимики. Необходимо обратить внимание, что связь может не сработать при включённом фаерволе или антивирусе, поэтому при возникновении данной проблемы необходимо отключить фаервол или антивирус.

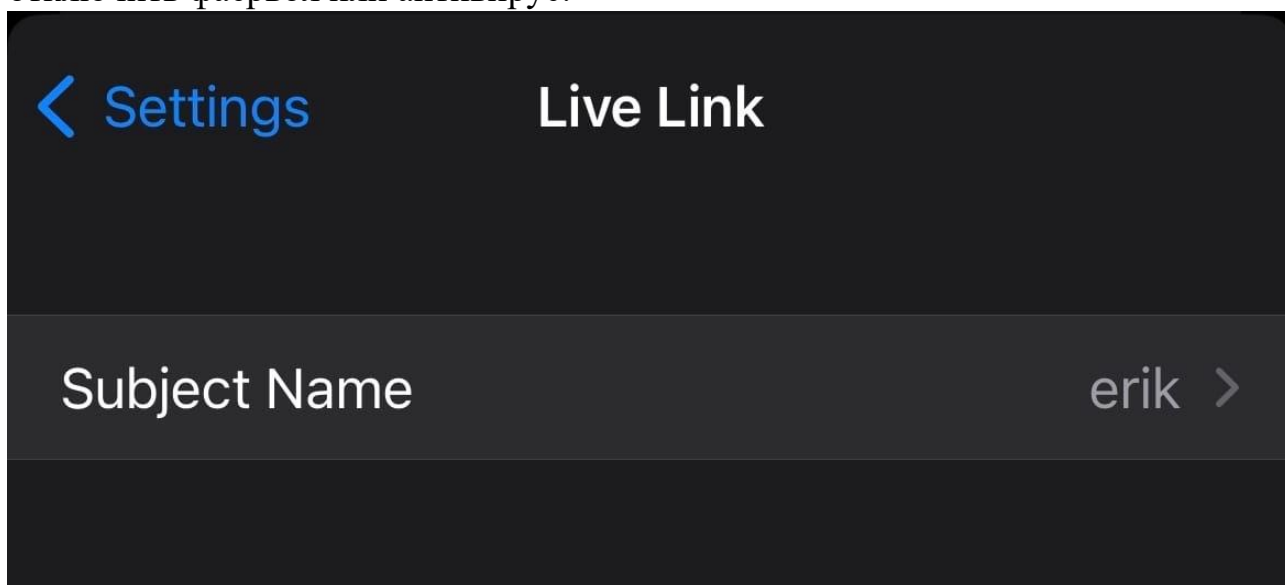


Рисунок 10. Графа для выбора имени iPhone

## **4.2 Анимация мимики лица**

Первым делом нужно перейти в нужную директорию в Unreal Engine – это директория Content-> Metahumans-> Common-> Face. После перехода необходимо открыть файл Face\_AnimBP. После открытия этого файла должен открыться Viewport [2].

В этом окне необходимо произвести подключение iPhone к Unreal Engine. Делается это в графе LLink Face Subj в котором необходимо указать найденное устройство.

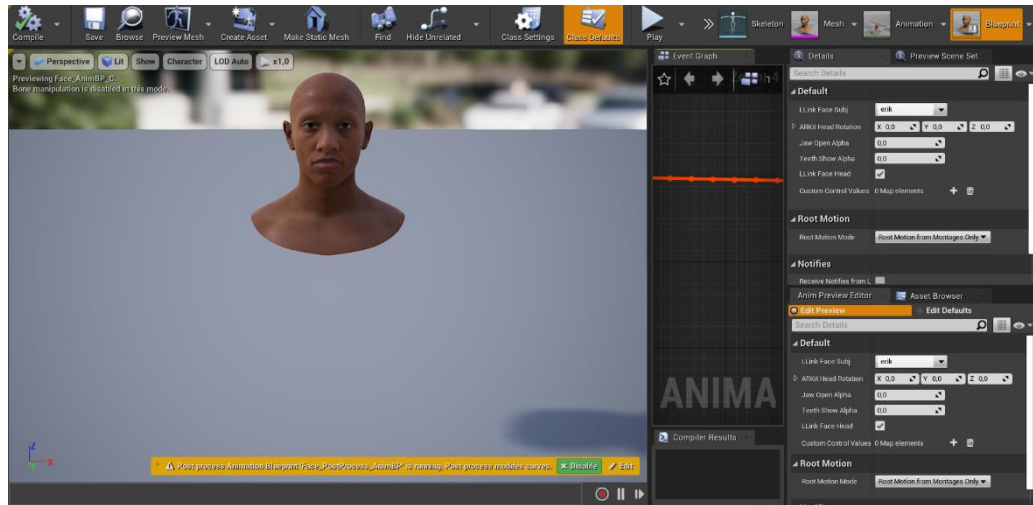


Рисунок 11. Viewport Face\_AnimBP

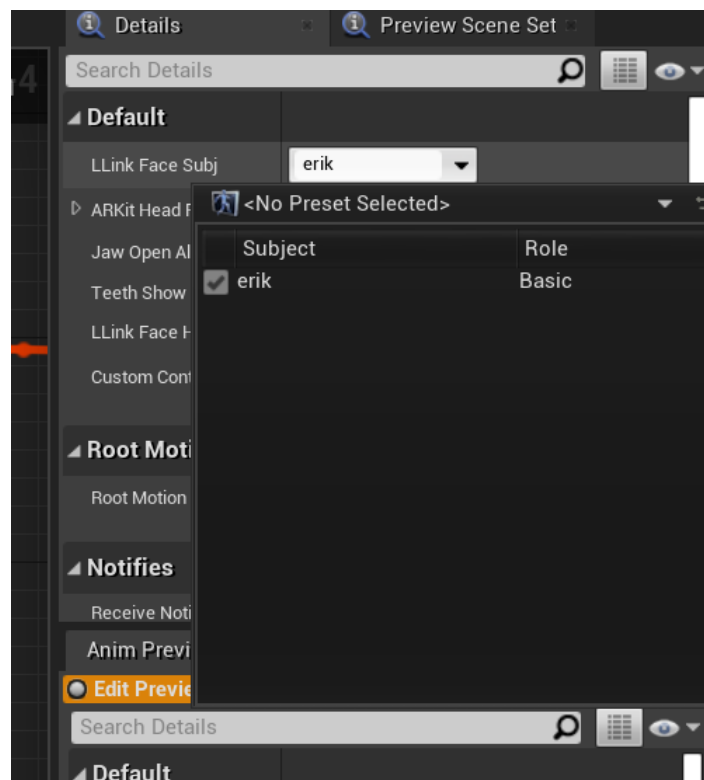


Рисунок 12. Выбор iPhone для подключения

После подключения устройства должна накладываться анимация на Mesh персонажа.

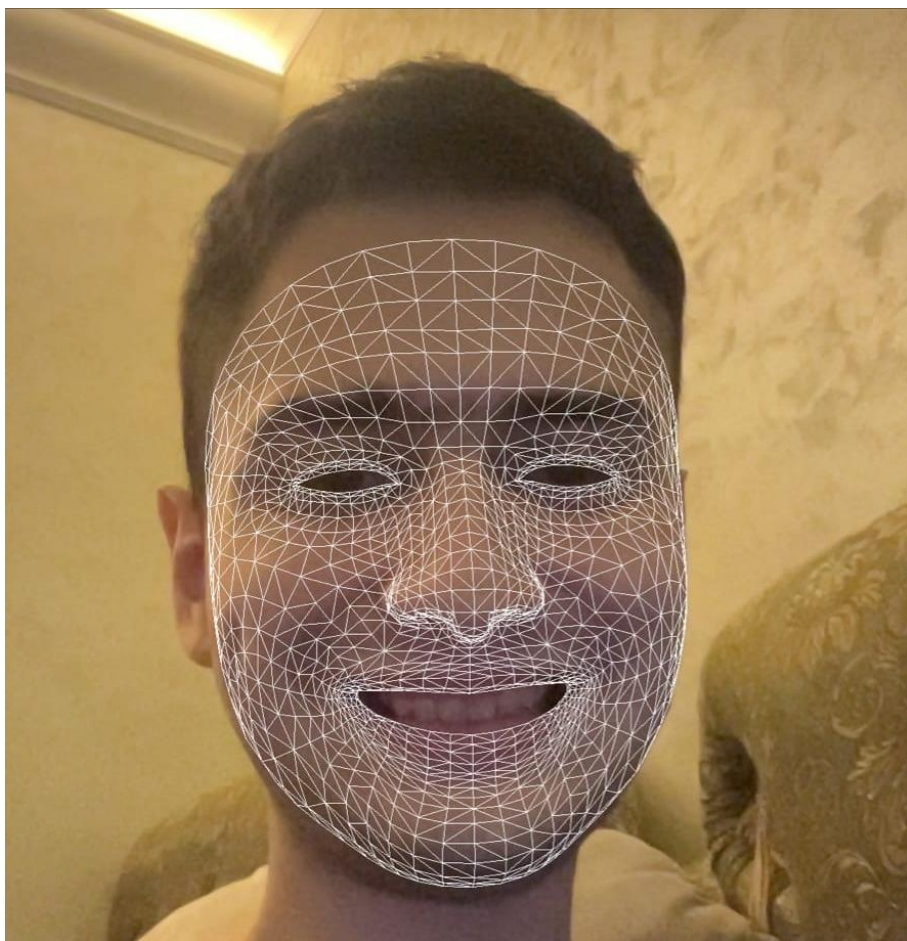


Рисунок 13. Лицо, с которого считывается мимика



Рисунок 14. Mesh, на который переносится мимика



После того как мы убедились, что мимика считывается можно перейти к переносу персонажа на сцену

### 4.3 Перенос персонажа на сцену

После переноса мимики можно уже перенести непосредственно персонажа на сцену. Сначала нужно перейти в директорию Content в Unreal Engine.

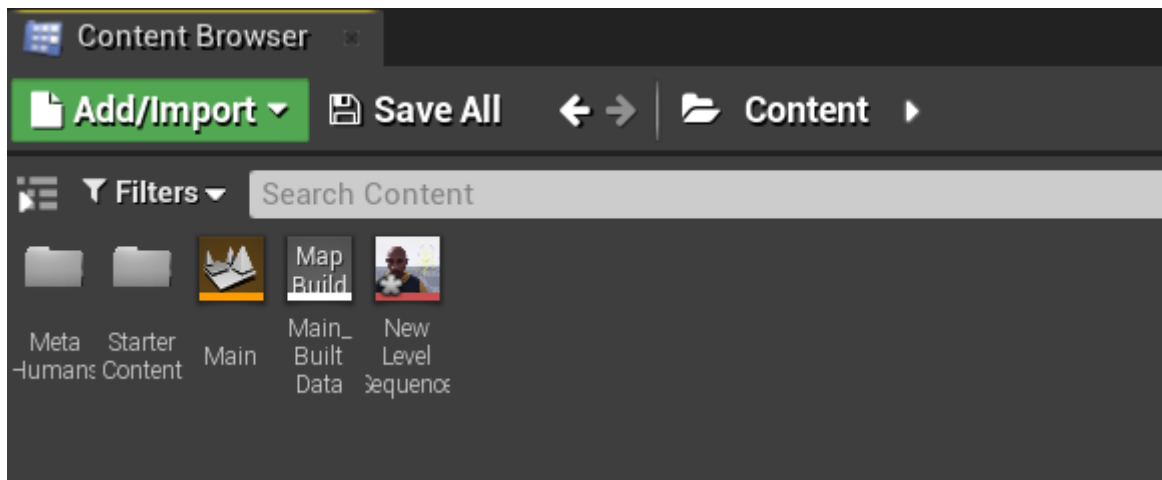


Рис.15. Директория Content

Далее необходимо создать Level Sequence. Для этого нужно нажать правой кнопкой мыши по пространству в директории Content и перейти по пути Animation -> Level Sequence.

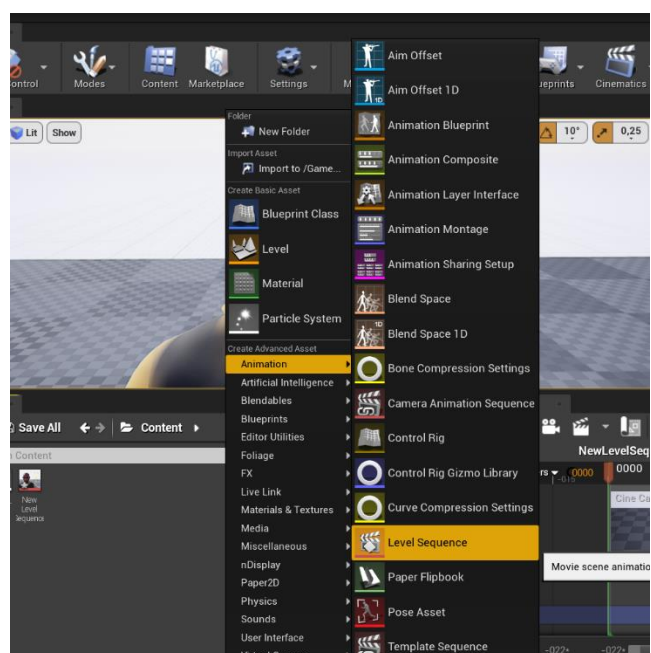


Рисунок 16. Создание LevelSequence

Далее необходимо перенести Blueprint персонажа из директории Content -> MetaHumans -> Bryan (стоит отметить, что Bryan – это имя непосредственно того персонажа, который был перенесён из Quixel Bridge. Если будет перенесён другой персонаж, то имя будет другим).

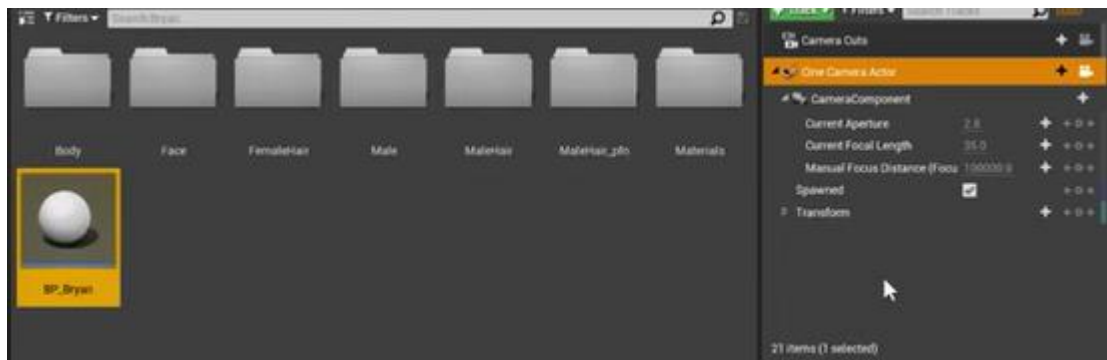


Рис.17. Перенос BP\_Bryan из директории в New Sequence

Перенос осуществляется простым перетягиванием файла. После этого нужно в BP\_Bryan в меню Details добавить компонент Live Link Skeletal Animation. После этого нужно точно так же LLink Face Subj подключить iPhone, и персонаж уже будет анимирован на сцене.



Рисунок 18. Анимированный персонаж перенесённый на сцену



## 5. Исправление ошибки с отсутствием волос на лице персонажа

При переносе персонажа из Quixel Bridge в Unreal Engine 4 может возникнуть такая ошибка как отсутствие волос на лице персонажа. Эта ошибка может возникнуть по разным причинам и в данном параграфе будут разобраны методы её решения.

### 5.1 Отсутствие волос из-за не прорисовки

Волосы могут просто не прорисоваться на персонаже из-за того, что им нужны хорошие аппаратные возможности ПК. Такие модели обозначены предупреждением в самом Quixel Bridge, скачивать их нужно на свой страх и риск, так как растительность на них доступна только на LODs 0 & 1.

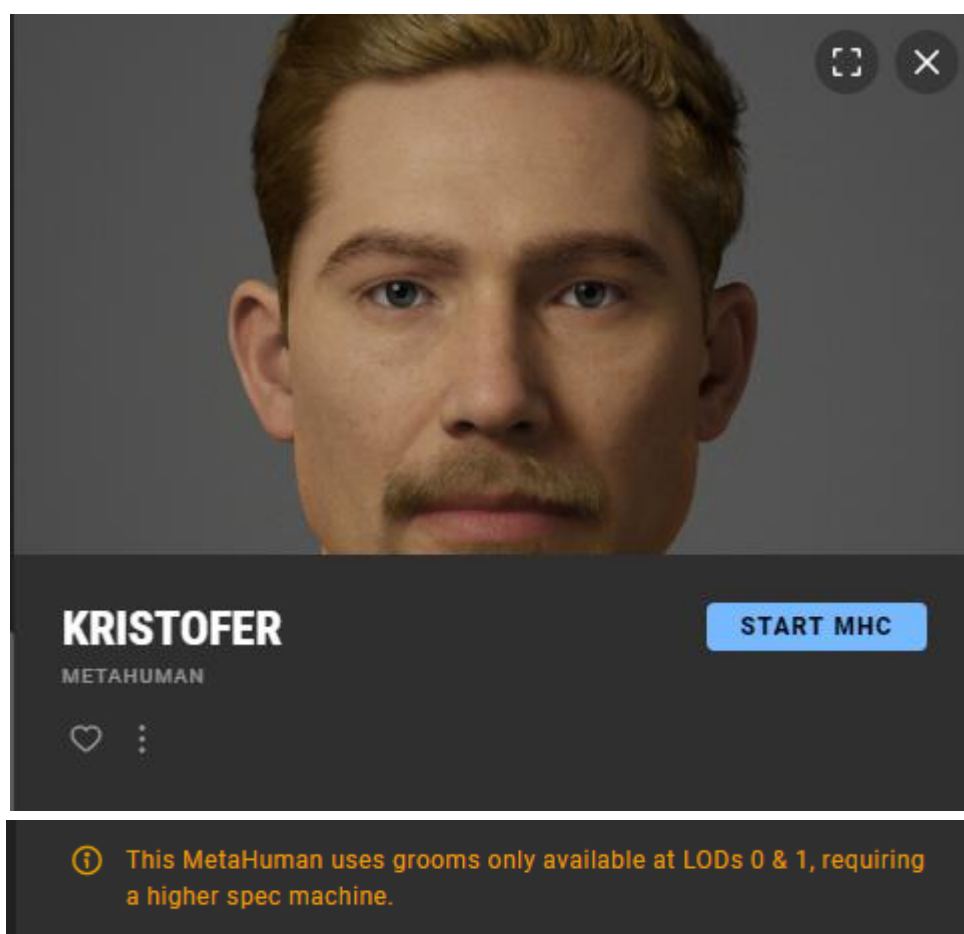


Рисунок 19. Предупреждение на модели персонажа

### 5.2 Отсутствие волос из-за отсутствия нужных плагинов

Следующей причиной данной проблемы может быть отсутствие плагинов для поддержки растительности на персонаже. Чтобы узнать подключены-ли они к проекту нужно зайти по следующим пунктам: Setting -> Plugins. В окне подключения плагинов нужно ввести Groom и, если он отключён, подключить его поставив галочку. Если понадобится, можно будет подключить и Alembic Groom Importer.

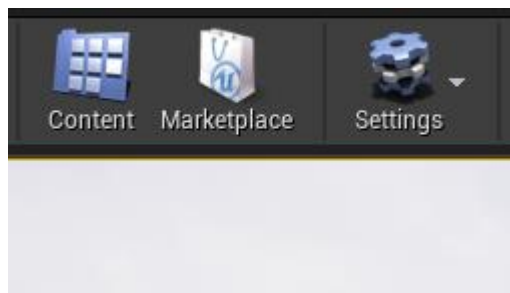


Рисунок 20. Меню настроек

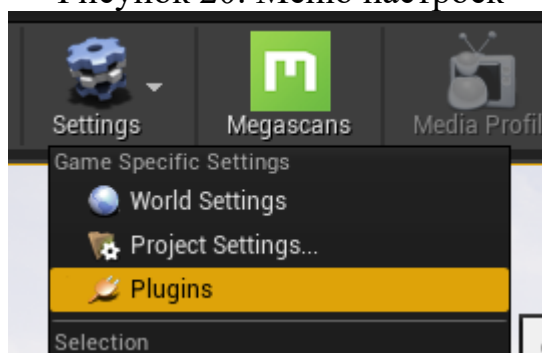


Рисунок 21. Меню Plugins

После добавления плагина в проект движок предложит перезапустить проект, для задействования плагинов в проекте. После перезапуска растительность должна появиться.

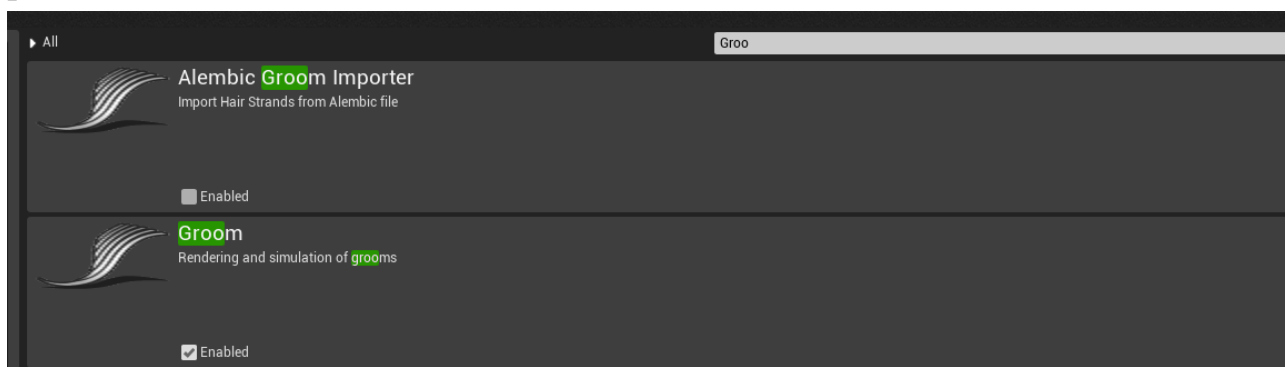


Рисунок 22. Plugin Groom

### 5.3 Отсутствие волос из-за ошибки прорисовки

В данном подпункте будет рассмотрена возможность ошибки прорисовки, которая может привести к пропаданию волос на персонаже. Для её решения необходимо запустить ВР файл персонажа, который обычно находится по пути Content/MetaHumans/CharacterName. Пройдя по этому пути нужно открыть файл BP\_CharacterName.

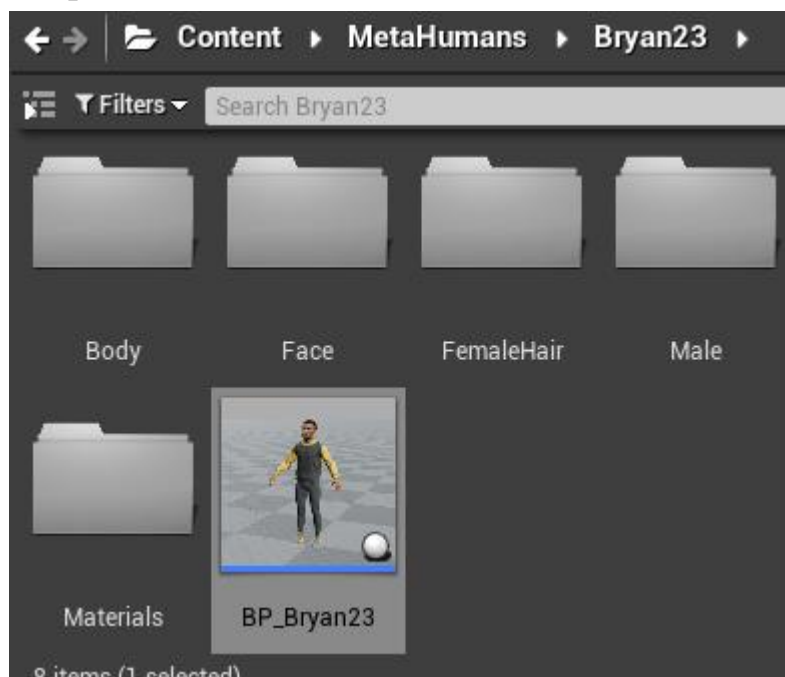


Рисунок 23. Файл ВР

После нажатия на файл должно открыться окно с настройками персонажа, в нём нам нужно перейти во вкладку Viewport:

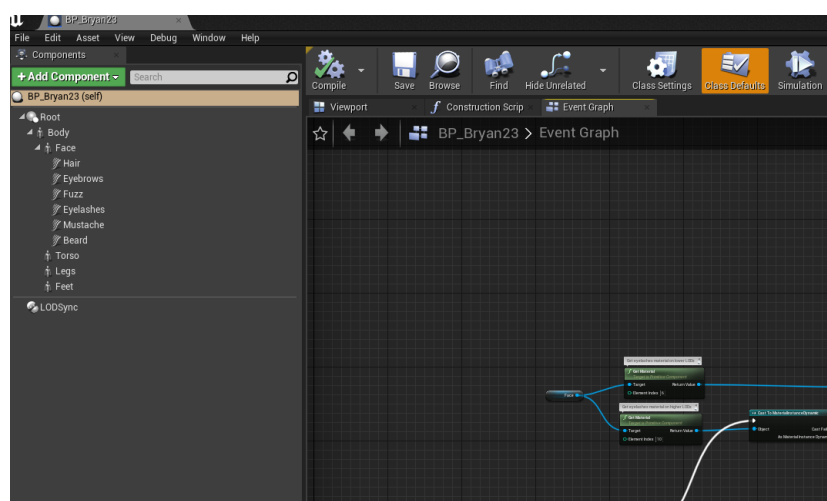


Рисунок 24. Окно настроек персонажа

В этом окне слева в дереве нужно перейти в ветку LODSync:

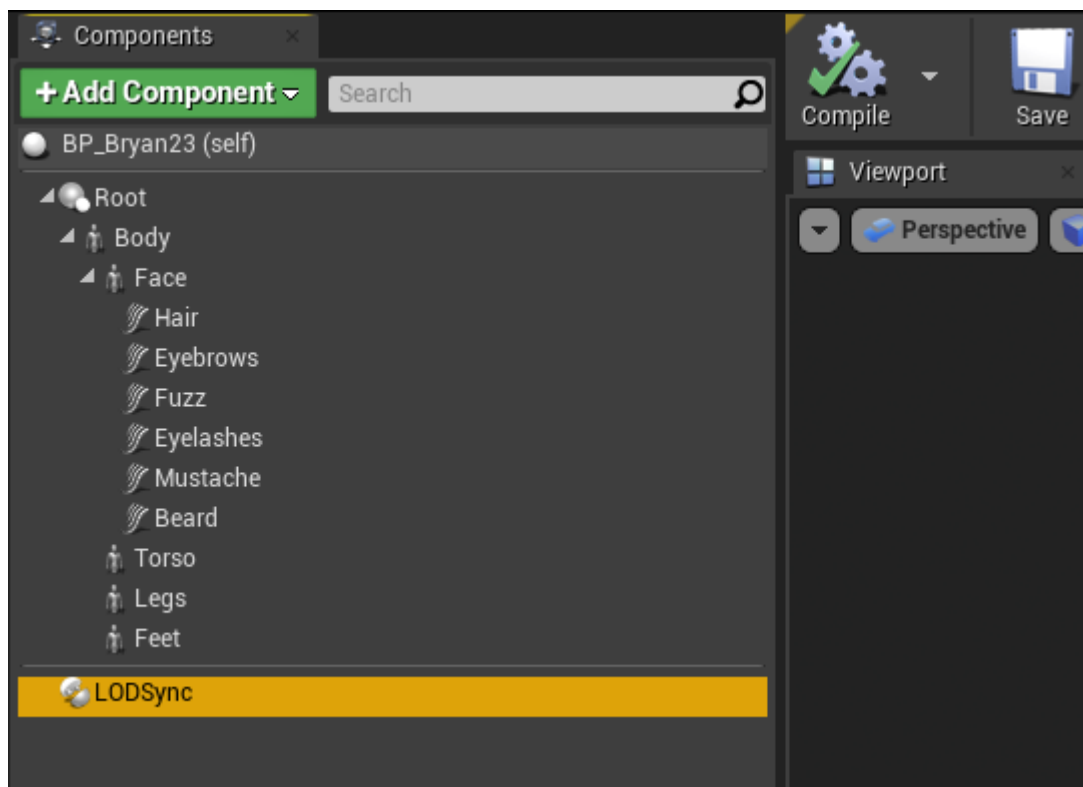


Рисунок 25. Ветка LODSync

В этой ветке во вкладке Details справа необходимо в пункте LOD поставить настройку Forced LOD на 0 вместо -1(-1 подразумевает использование всех доступных LOD, если изменить это значение на 0, то всегда будет использоваться только LOD 0) в результате чего у персонажа перестанут пропадать волосы.[6]

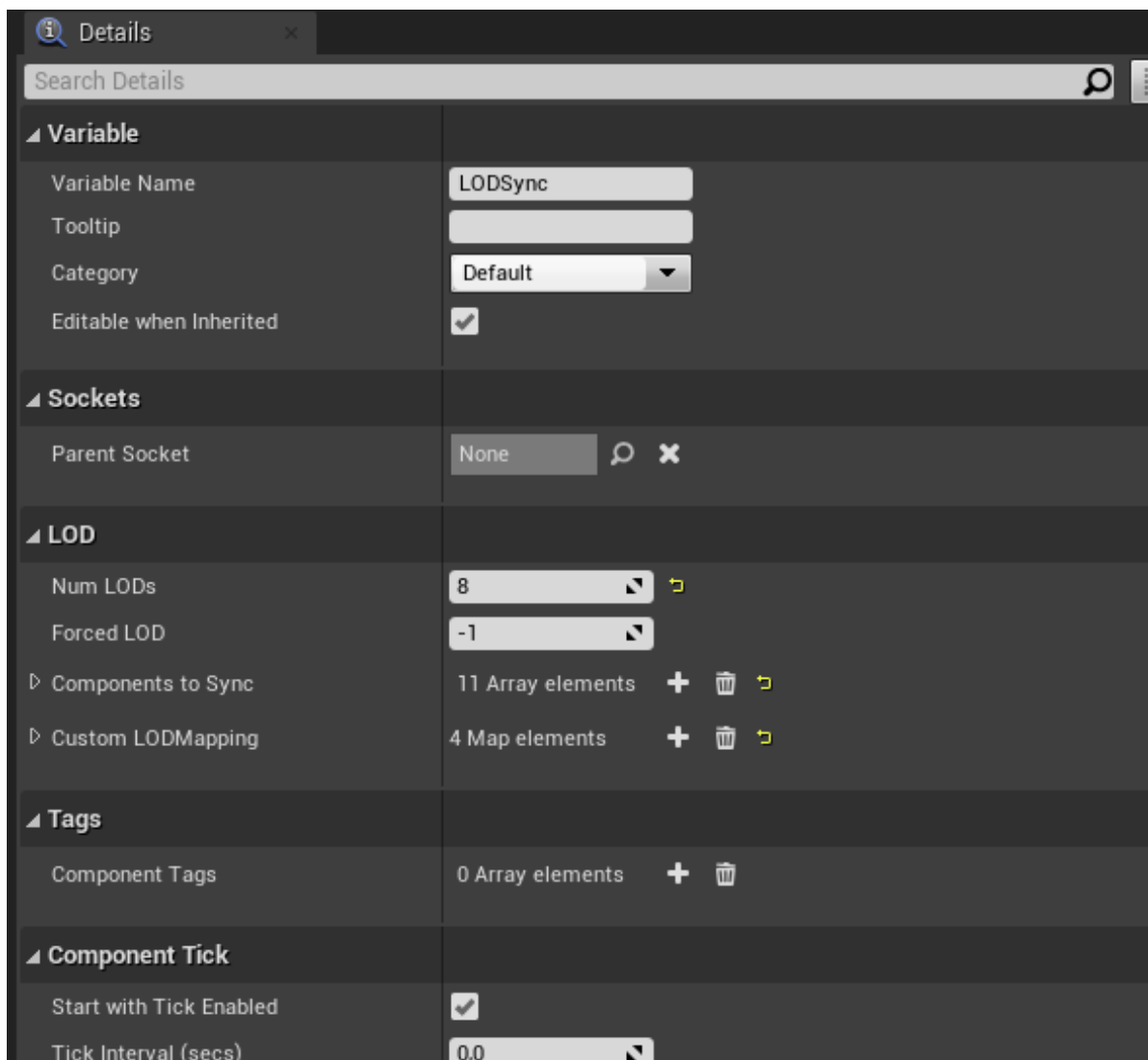


Рисунок 26. Окно Details

## 5.4 Отсутствие волос из-за ошибки импорта файлов

Растительность на персонаже может также не появляться, потому что ассетов для неё в проекте просто нет. Такая ошибка может случиться при экспортировании персонажа из Quixel Bridge в Unreal Engine 4. Чтобы исправить данную ошибку необходимо провести следующие манипуляции:

1. Сохранить модель для экспорта как свою.

Для того, чтобы сохранить модель как свою собственную необходимо нажать кнопку Start МНС под иконкой персонажа в Quixel Bridge.

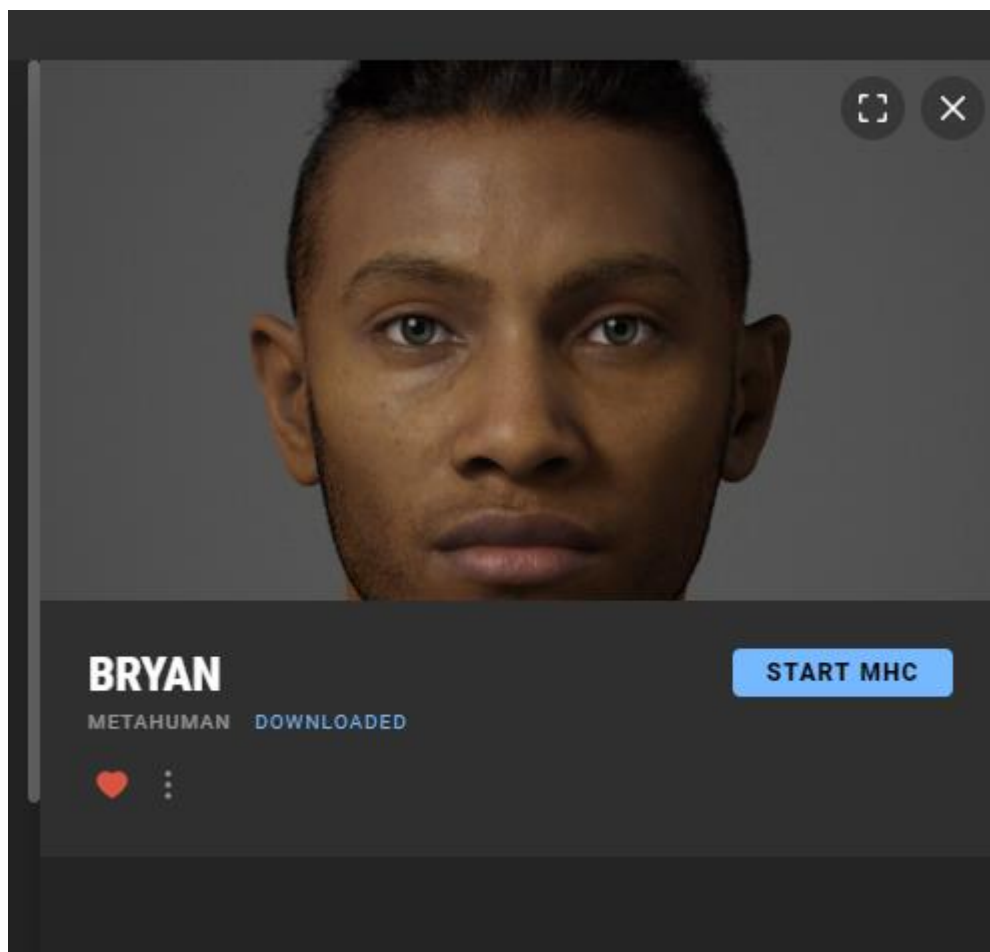


Рисунок 27. Кнопка START MHC

После нажатия на кнопку нас перенесёт на сайт редактирования персонажа Meta Human Creator. На нём нам необходимо перейти в редактор персонажа в зависимости от того какая версия Unreal Engine используется: Launch latest MetaHumanCreator, если используется версия UE 5 и позднее или Launch original MetaHumanCreator, если используется версия UE 4.27 и ранее.

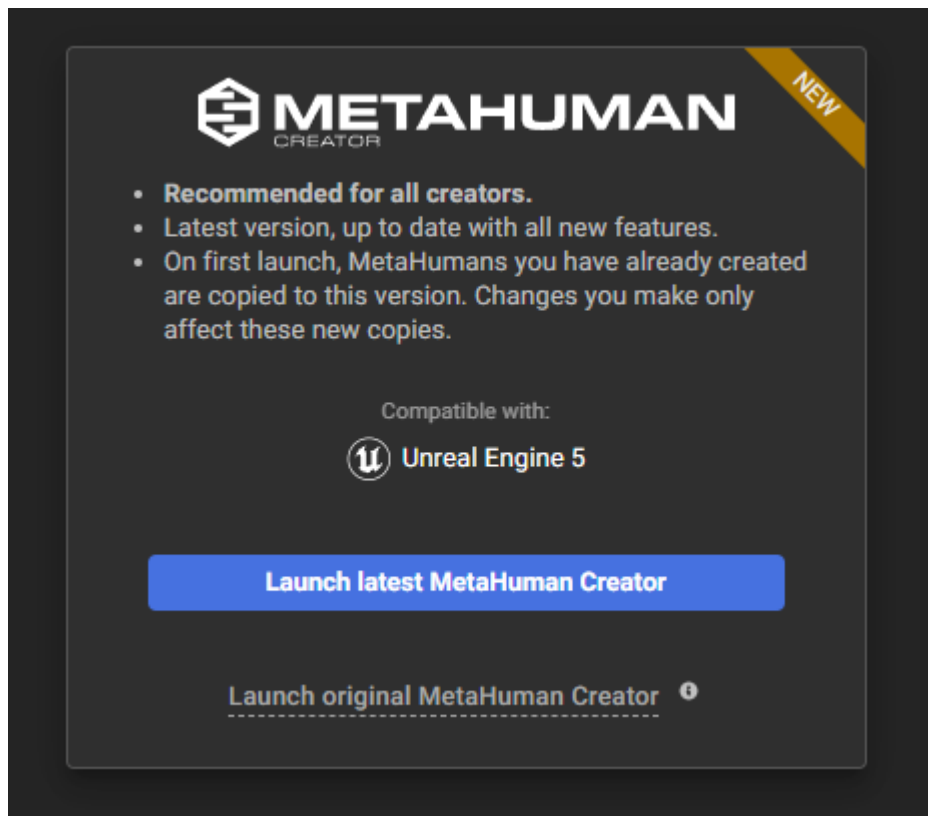


Рисунок 28. Выбор версии редактора МНС

После выбора редактора будет открыто меню выбора персонажа. Выбираем персонажа и открываем редактор. После открытия редактора просто сохраняем модель, не внося изменений под другим именем и зайдя в меню Quixel Bridge входим в меню My MetaHumans:

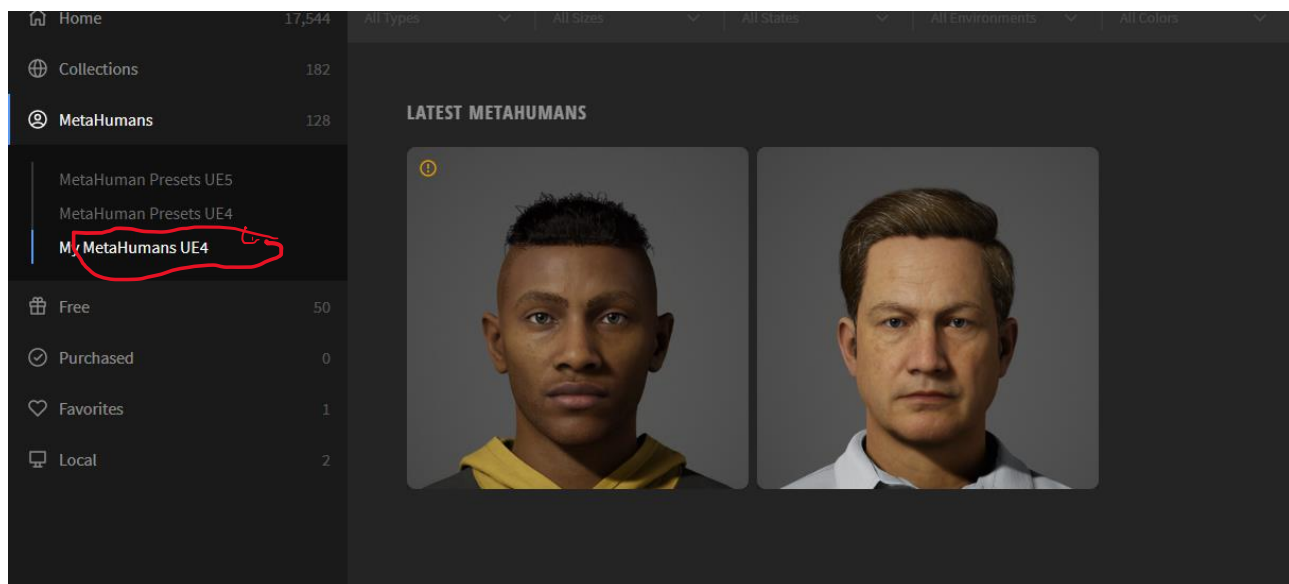


Рис.29. Меню персонажей My MetaHumans в Quixel Bridge

После этого экспортируем свою модель в UE 4. Растительность должна появиться.



## **6.Озвучка персонажа**

Итак, после создания персонажа и переноса его непосредственно на саму сцену необходимо выполнить важную и не менее важную работу – озвучку персонажа. Для многих это может стать весьма трудоёмкой задачей, так как это сопряжено с такими трудностями как недостаток средств и отсутствия навыков актёров озвучки. Для больших студий это будет гораздо легче, потому что у них будет намного больше возможностей как в наличии средств и материальной базы (микрофон, зубообрабатывающее оборудование, дорогое ПО и т.д.), так и в поисках актёров озвучки, оплачивание которых требует больших денег. Одинокие разработчики не имеют таких возможностей, поэтому им необходимо использовать такие обходные пути как преобразование текста в речь или обработка голоса. В этой главе будут рассмотрены методы озвучки персонажа.

### **6.1. Преобразование текста в речь с помощью Replica**

Одним из методов озвучки персонажа является преобразование текста в речь. Не каждое ПО, выполняющее подобные функции, может подойти для создания игр, фильмов и других продуктов, которые требуют чёткой речи с хорошей интонацией и наличием выражения эмоций. Это обусловлено «роботизированной и безэмоциональной речью» таких программ, вследствие чего озвучка получается слабой и неестественной. К счастью, Unreal Engine существует плагин, позволяющий использовать такую программу как Replica. Это ПО предназначено для использования голосов, присутствующих в библиотеке самой Replica, в целях озвучки написанного текста.

Для того, чтобы начать работу с Replica сначала нужно зарегистрироваться. Если у вас нет аккаунта, то его нужно будет создать.

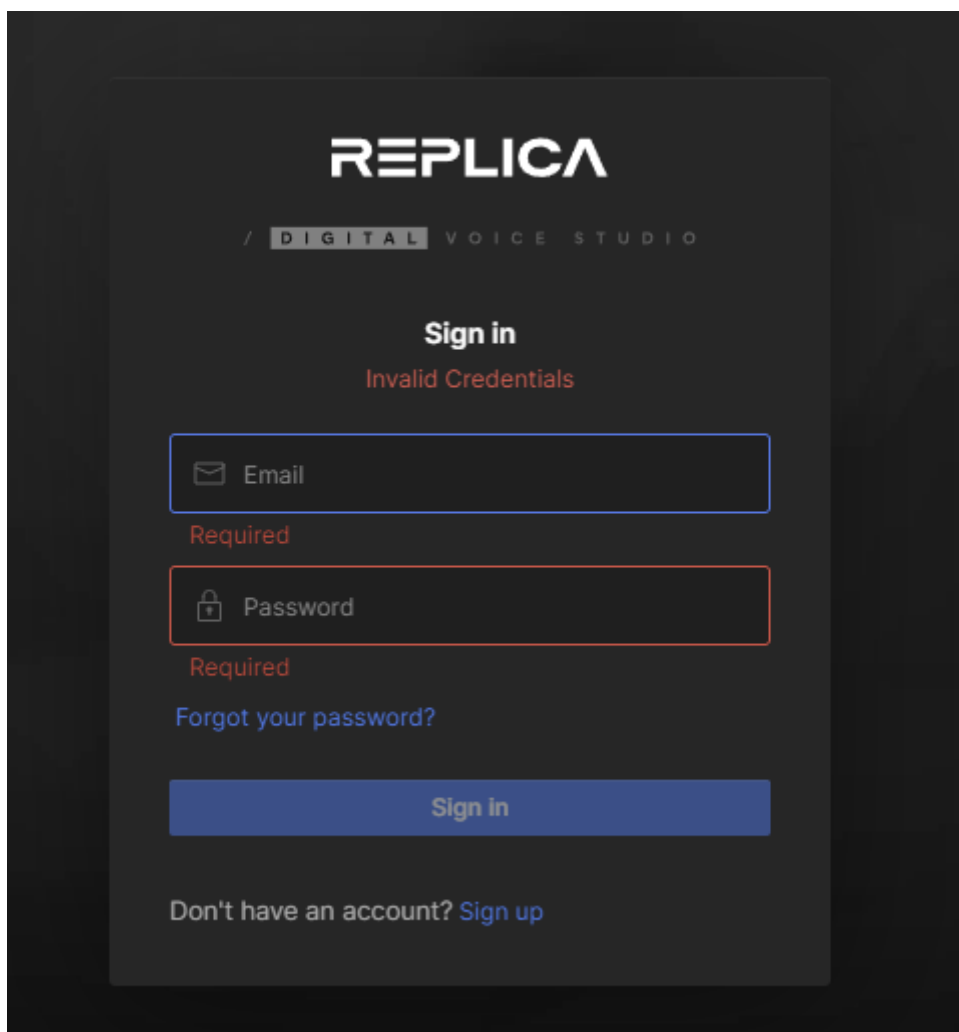


Рис.30. Окно входа в Replica

После этого вам будет доступна главная страница Replica. На ней нужно выбрать пункт Projects.

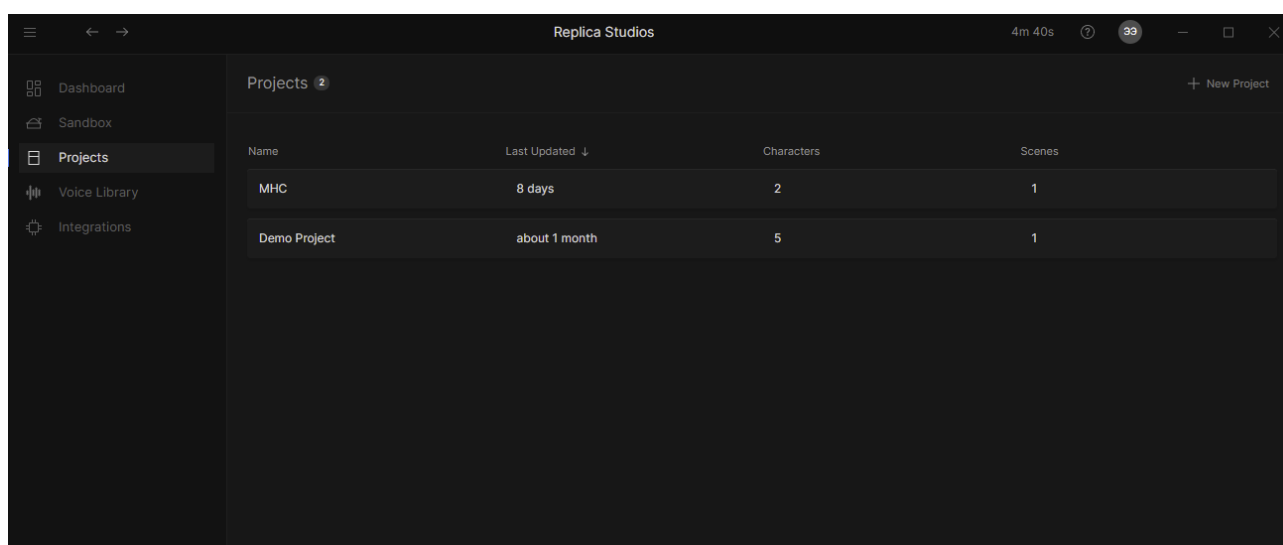


Рисунок 31. Меню Projects

В данном меню изначально будет присутствовать вкладка Demo Project. Удалять её не обязательно, но в ней хранится пример работы программы. Чтобы создать свой собственный проект нажать на кнопку New Project в правом верхнем углу (см Рис. 30). Далее необходимо войти в свой проект кликнув на его название. После этого будет доступно возможность добавить нового персонажа и создать уровень.

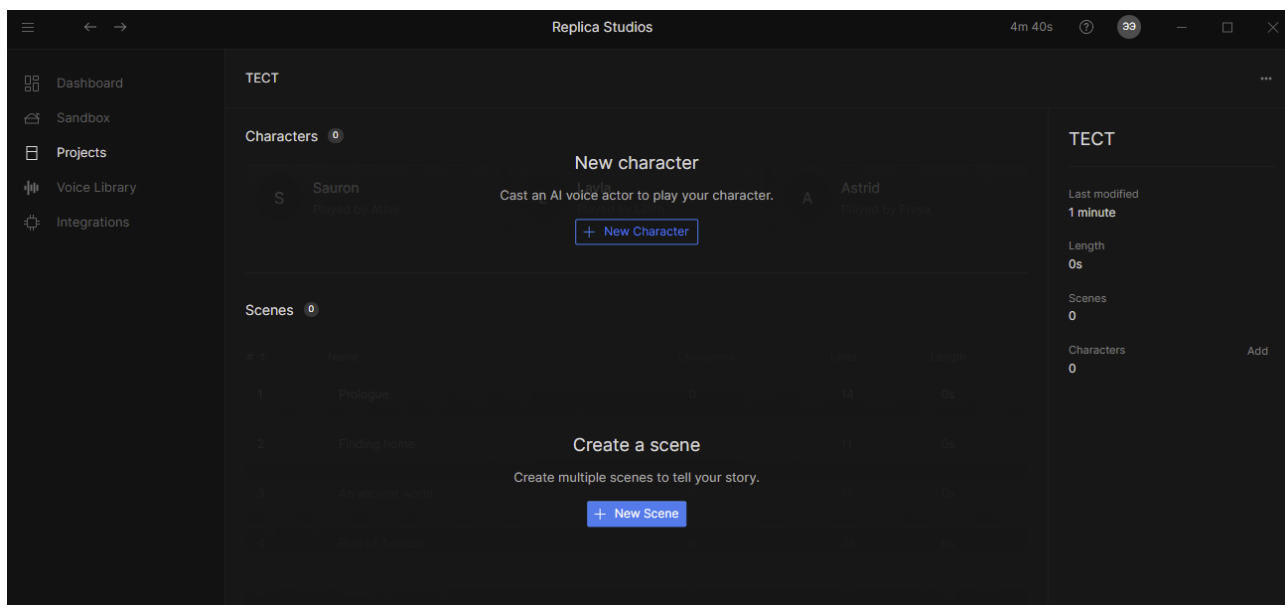


Рисунок 32. Меню редактора проекта

Чтобы добавить нового персонажа, необходимо нажать на кнопку +New Character отмеченной синей рамкой. После этого откроется окошко для имени и описания персонажа.

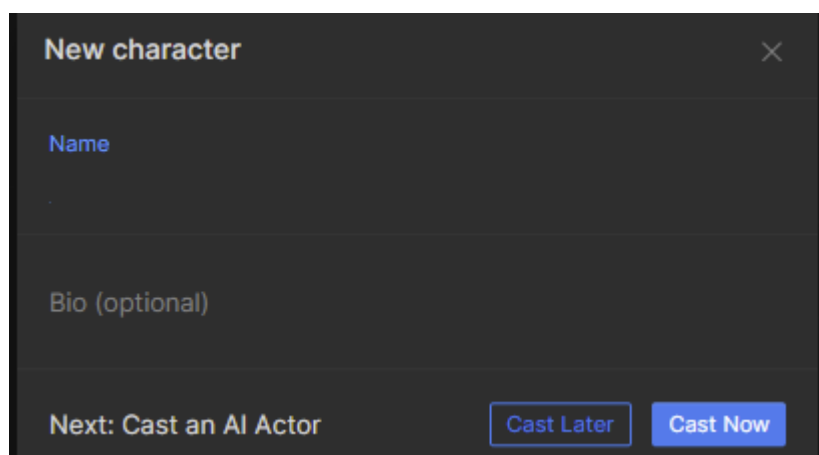


Рисунок 33. Меню описания персонажа

Кнопки Cast Later и Cast Now отвечают за создание персонажа и создание с его последующей озвучкой соответственно. Если же была нажата кнопка Cast Now, то откроется меню выбора голоса для озвучки персонажа.

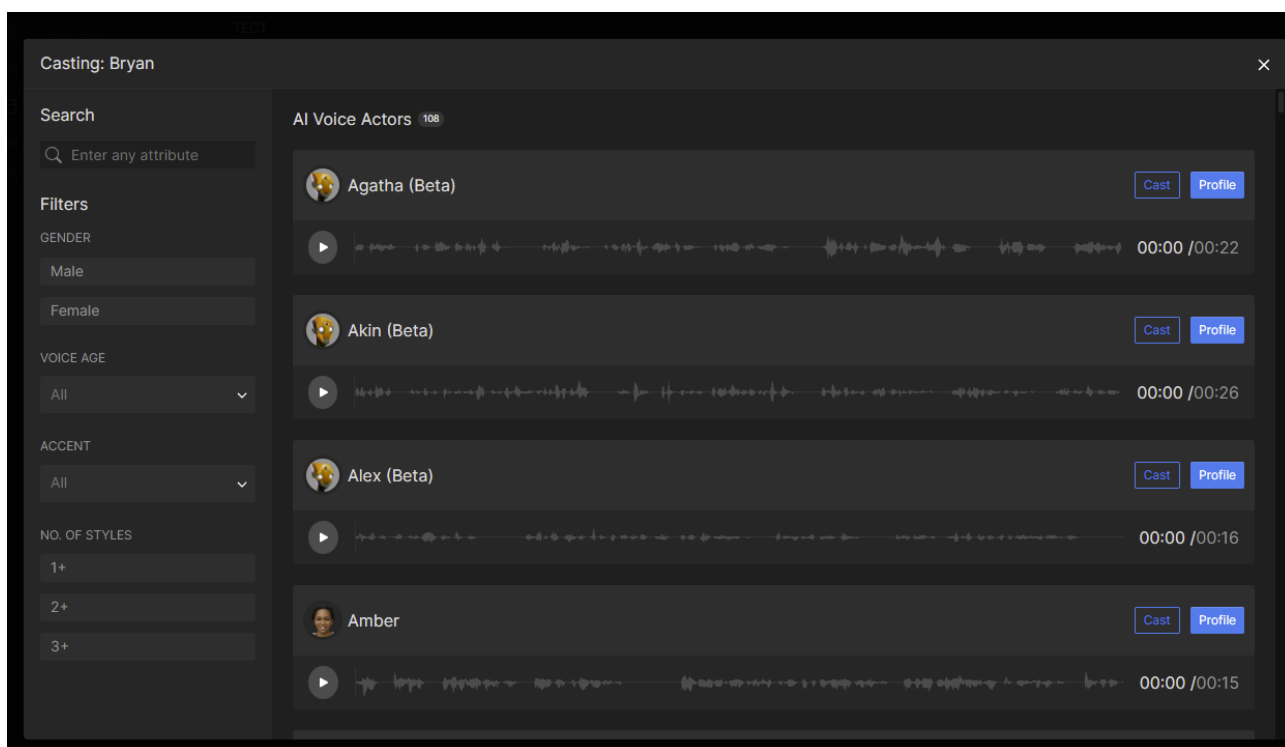


Рисунок 34. Список доступных голосов

Далее откроется меню доступных голосов. В левой панели находятся фильтры по полу, возрасту, акценту и стилю речи. Выбрав необходимый голос необходимо нажать на кнопку Cast в правой части звуковой дорожки. После этого необходимо создать Scene (Сцену), в который будет помещён созданный персонаж. Для этого необходимо нажать синюю кнопку +New Scene, после чего выбрать и зайти в созданную сцену. Чтобы преобразовать текст в речь нужно выбрать персонажа, голос которого был выбран заранее и набрать текст. К сожалению, пока что Replica способна проводить озвучку только на английском языке. В графе Style можно выбрать тон речи персонажа: Serious(серьёзный), Angry(злой), Scared(напуганный). Выбрав необходимые параметры необходимо нажать на кнопку записи, она находится слева от Direct и обозначена стрелкой в кружочке. Важно обратить внимание, что после преобразования текста в речь будут списаны доступные секунды в соответствии с длиной записи. Доступной время отображается в правом верхнем углу.

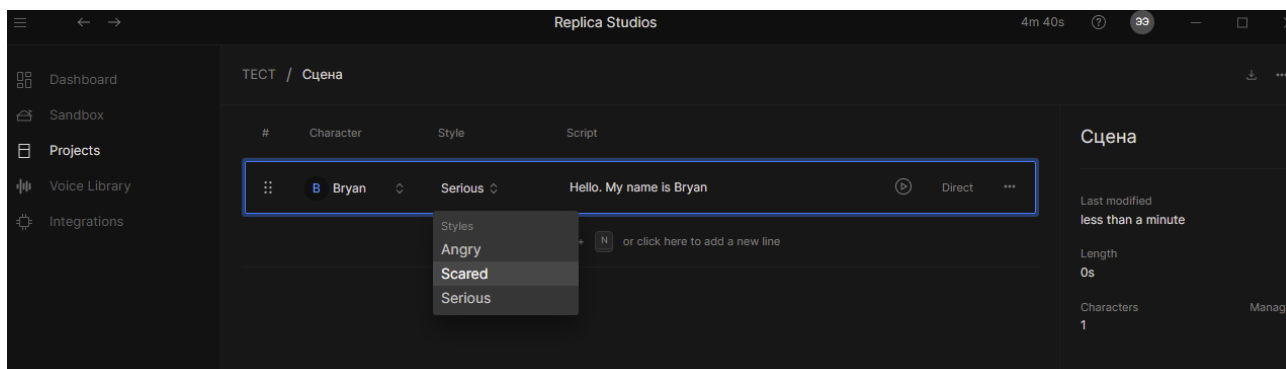


Рисунок 35. Меню озвучки персонажа

После записи можно будет скачать созданную аудиозапись нажав на три точки справа от Direct и сохранить запись в нужном формате.

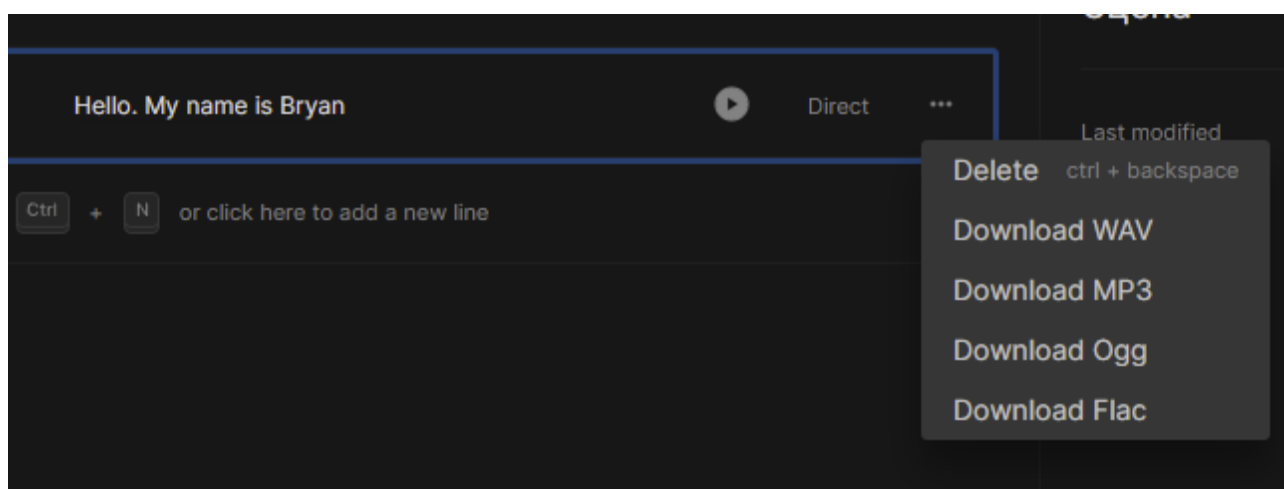


Рисунок 36. Варианты формата аудиозаписи

Скачанную аудиозапись можно будет перенести в сам Editor

## 6.2 Преобразование текста в речь с помощью xVASynth

xVASynth- это полностью бесплатное приложение, которое позволяет создавать диалоги как методом text-to-speech, так и speech-to-speech. Оно позволяет делать озвучку голосами персонажей из разных игр. Приложение можно как с площадки Nexus Mods, так и через Steam. Все голоса, с помощью которых будет производится озвучка также можно скачать с площадки Nexus Mods.

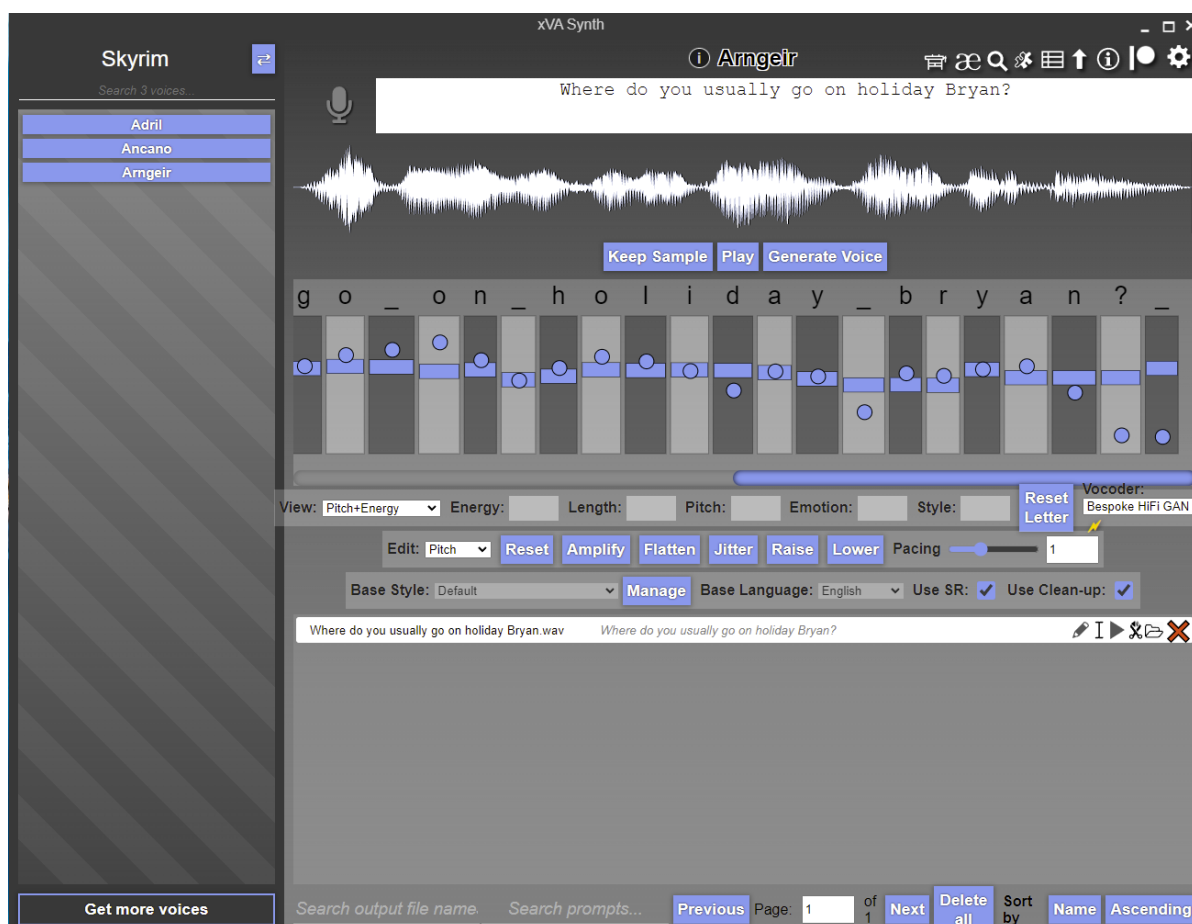


Рисунок 37. Панель управления xVASynth

Итак, чтобы создать аудиозапись необходимо для начала набрать в верхней панели текст, который должен быть преобразован в речь

### 6.3 Lip sync персонажа с помощью MetahumanSDK

Для того чтобы осуществить синхронизацию речи персонажа с движениями его губ (Lip sync) необходим плагин MetaHumanSDK, который автоматически генерирует синхронизацию для персонажа. Для этого его необходимо скачать из Epic Store Marketplace. Данный плагин абсолютно бесплатен. Необходимо обратить внимание на то, что его лучше добавлять на свой аккаунт Epic Store через браузер, так как в самом приложении он может не отобразиться. Если же плагин будет отображаться как Not For Sale можно попробовать скачать добавить его с телефона или подключить любой VPN и через него зайти на страницу плагина [9].

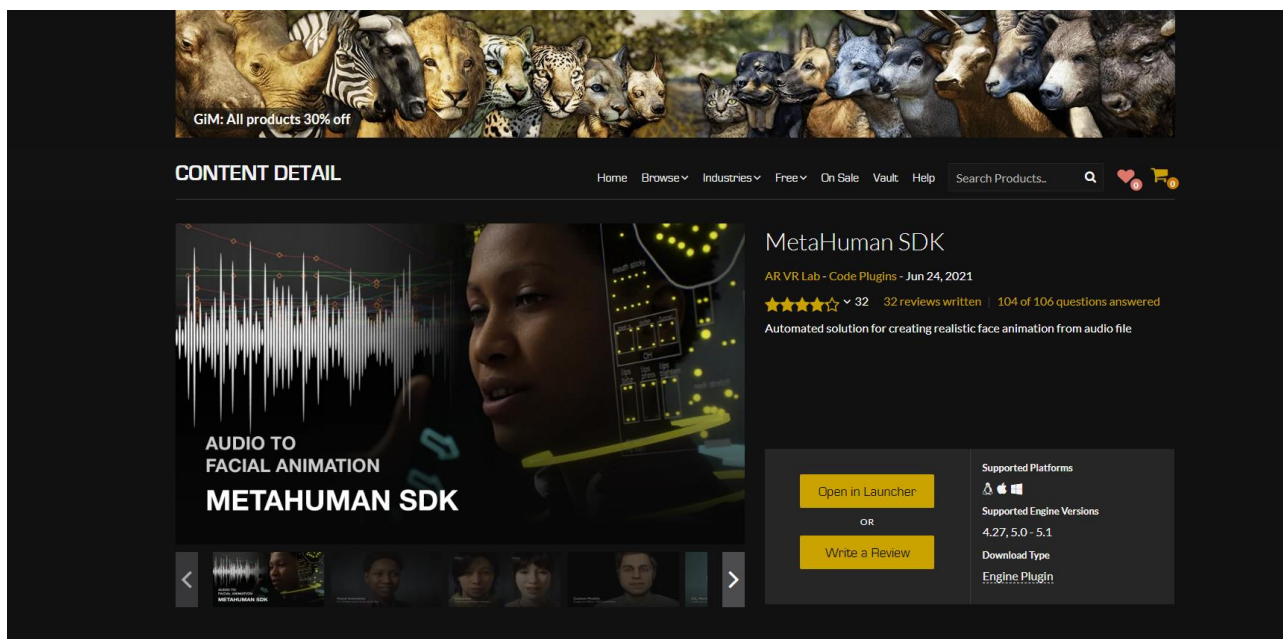


Рисунок 38. Страница плагина в Epic Games Store

После приобретения плагина его надо подключить к своему проекту через библиотеку.

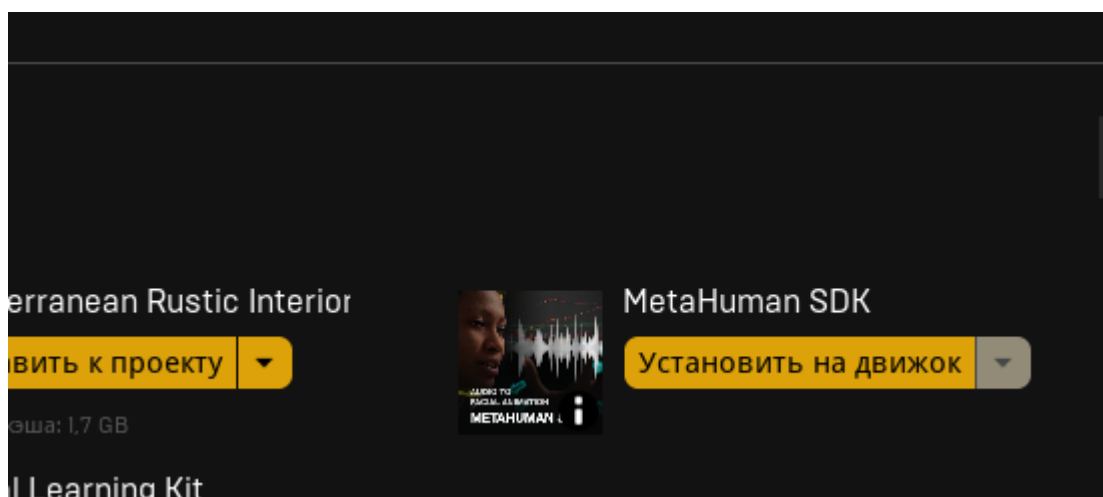


Рисунок 40. Плагин MetaHuman SDK

Необходимо лишь нажать кнопку “Установить на движок” и выбрать версию Unreal Engine, к которому нужно подключить плагин.

## 6.4 Создание анимации по аудиозаписи

Для генерации Lip sync необходимо нажать правой кнопкой мыши. По аудиозаписи и нажать Create Lipsync animation после чего будет открыто меню настройки анимации.

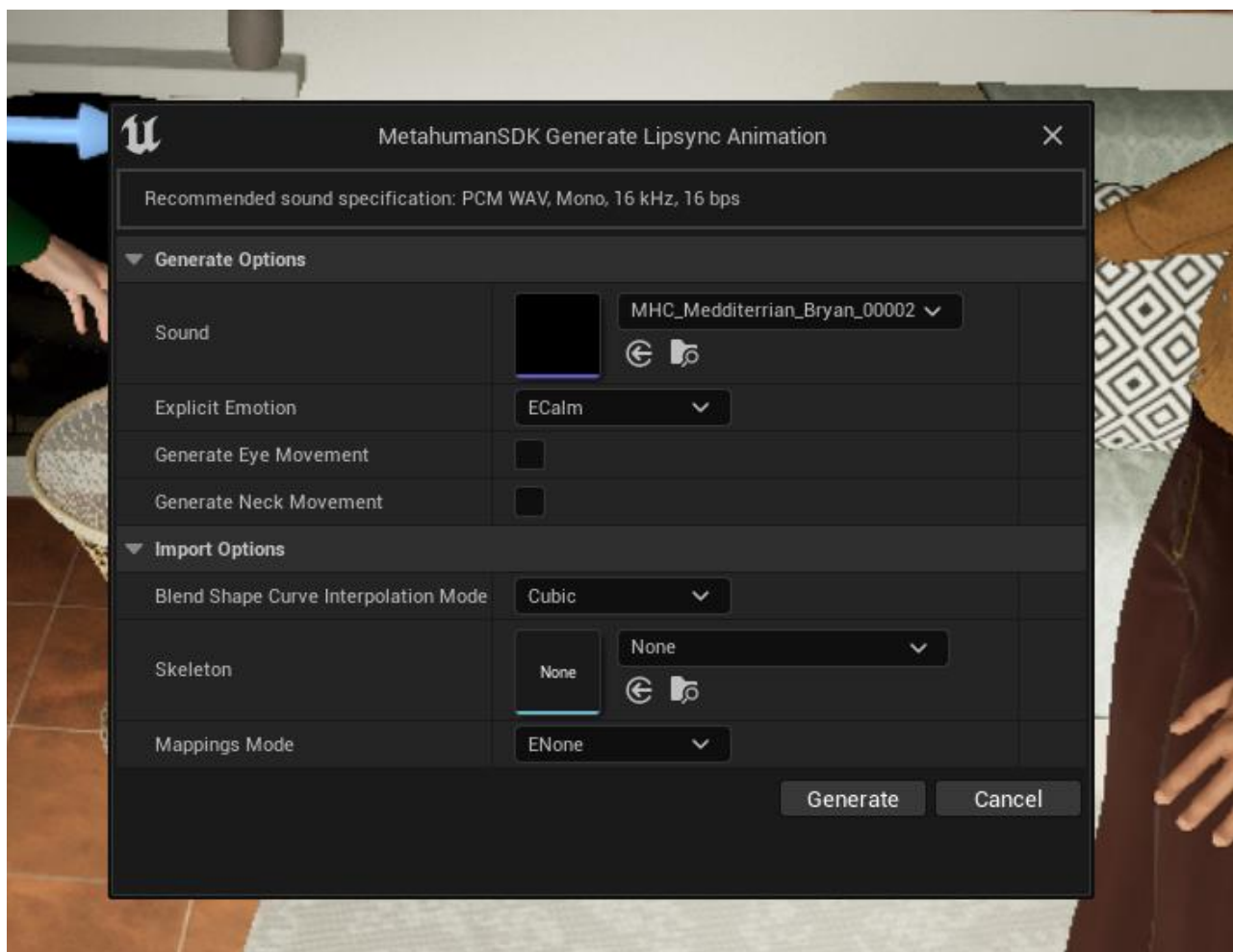


Рисунок 41. Меню настроек анимации

Ниже указаны назначения каждой кнопки:

- Sound – звуковая дорожка, по которой будет создаваться анимация. Необходимо выбрать необходимую звуковую дорожку.
- Explicit Emotion – эмоция, с которой будет воспроизводиться озвучка. Если выбрана ECalm, то голос персонажа будет со спокойной интонацией.
- Generate Eye Movement – генерация движений глаз.
- Generate Neck Movement – генерация движений шеи.
- Blend Shape Curve Interpolation Mode – режим интерполяции.
- Skeleton – выбор архетипа скелета, который будет использован для анимации. Необходимо выбрать Face\_Archetype\_Skeleton (Архетип скелета лица).



- Mappings Mode – режим маппинга. В данном случае нужен EMetaHuman

После выбора необходимых настроек нужно нажать кнопку Generate. Анимация появится в том же файле, что и аудиозапись.

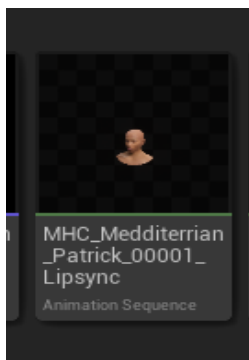


Рисунок 42. Созданная анимация

Если вдруг вместо анимации появилась ошибка LogMetaHumanSDKAssetFactory: Error: Request error (code:0), то необходимо прописать следующий код в файлах DefaultEditor.ini и DefaultEngine.ini, которые находятся в папке Config в папке проекта:

[HTTP]

FlushSoftTimeLimitDefault=120

FlushSoftTimeLimitBackground=120

FlushSoftTimeLimitShutdown=120

FlushSoftTimeLimitFullFlush=120

FlushHardTimeLimitDefault=240

FlushHardTimeLimitBackground=240

FlushHardTimeLimitShutdown=240

FlushHardTimeLimitFullFlush=240

После этого перезагружаем проект и повторяем предыдущие пункты.

## 6.5 Наложение анимации на персонажа

Для наложения анимации на лицо необходимо создать Level Sequence. Далее необходимо нажать +track и выбрать пункт “Actor to sequencer” и выбрать нужный нам ВР. В данном случае это ВР\_Bryan.

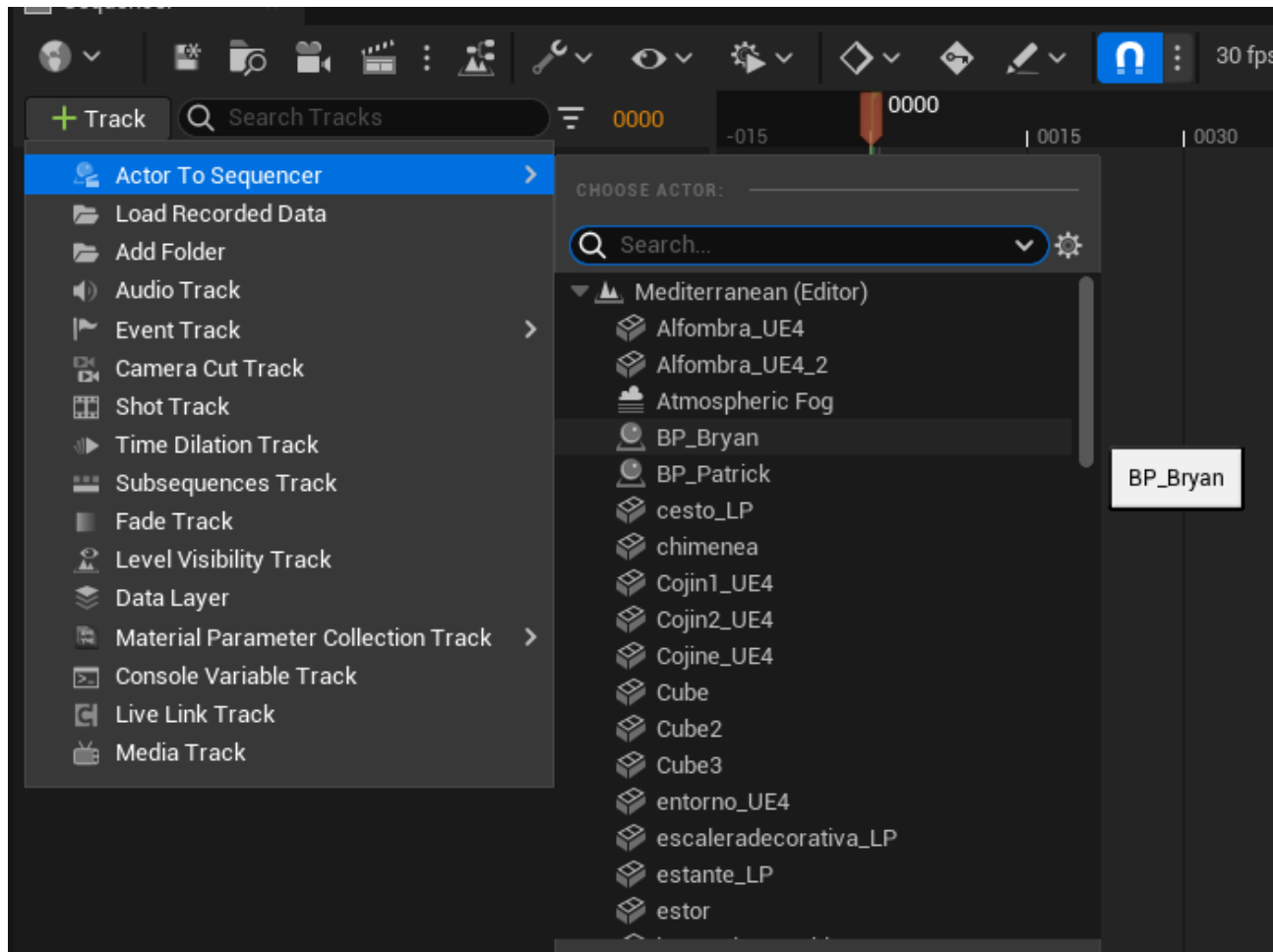


Рисунок 43. Добавление нового актёра

Теперь можно приступить к наложению Lipsync. Для это нам нужно открыть вкладку Face нажать +Track-> Animation и выбрать созданный ранее Lipsync(MHC\_Medditerian\_Bryan\_00002\_Lipsync). После этого появиться дорожка с анимацией персонажа

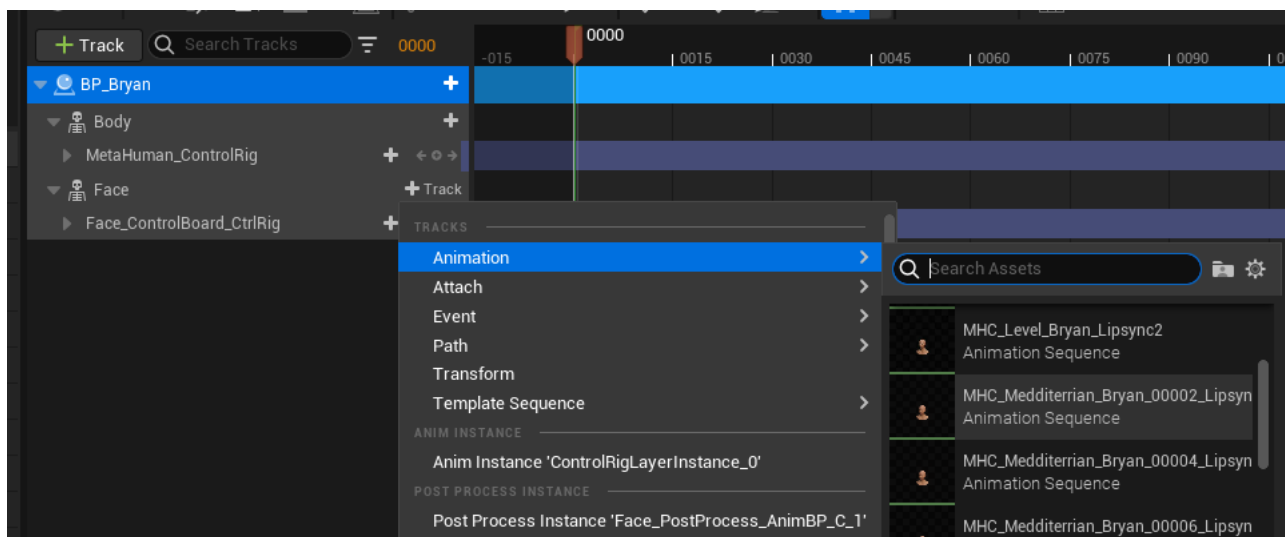


Рисунок 44. Добавление анимации для персонажа

Но этого недостаточно так как надо также добавить и звуковую дорожку, по которой создавалась анимация. Если этого не сделать анимация сработает, но при этом персонаж будет просто шевелить лицом. Для добавления аудиозаписи необходимо также как и при добавлении актёра нажать кнопку +Track в левом верхнем углу, но при этом нажать Audio Track. Потом нажать +Audio в появившемся окне Audio и добавить ту звуковую дорожку, по которой создавалась анимация.

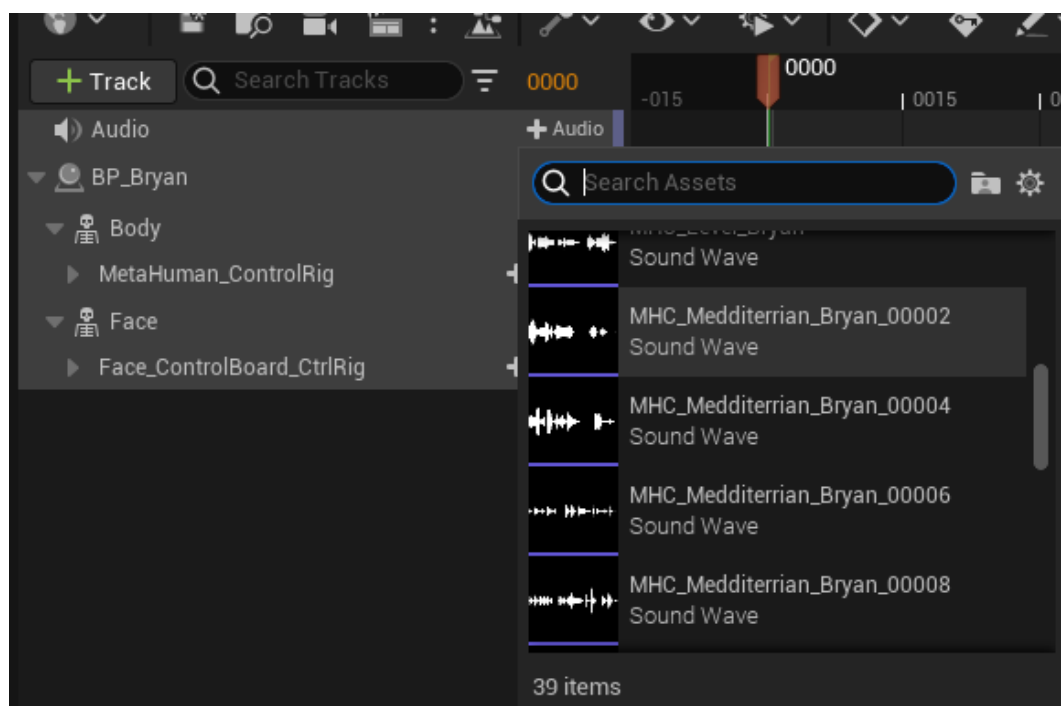


Рисунок 45. Добавление аудиозаписи для персонажа

Выполнив все указанные пункты, можно нажать кнопку старт и посмотреть результат проделанной работы.

## 7.Rigging персонажа

Rigging (риггинг) — это подготовка 3D-модели персонажа к анимации, при которой внутри заранее отрисованной заготовки размещается риг — набор виртуальных суставов и костей, устанавливаются закономерности его функционирования и возможные трансформации.

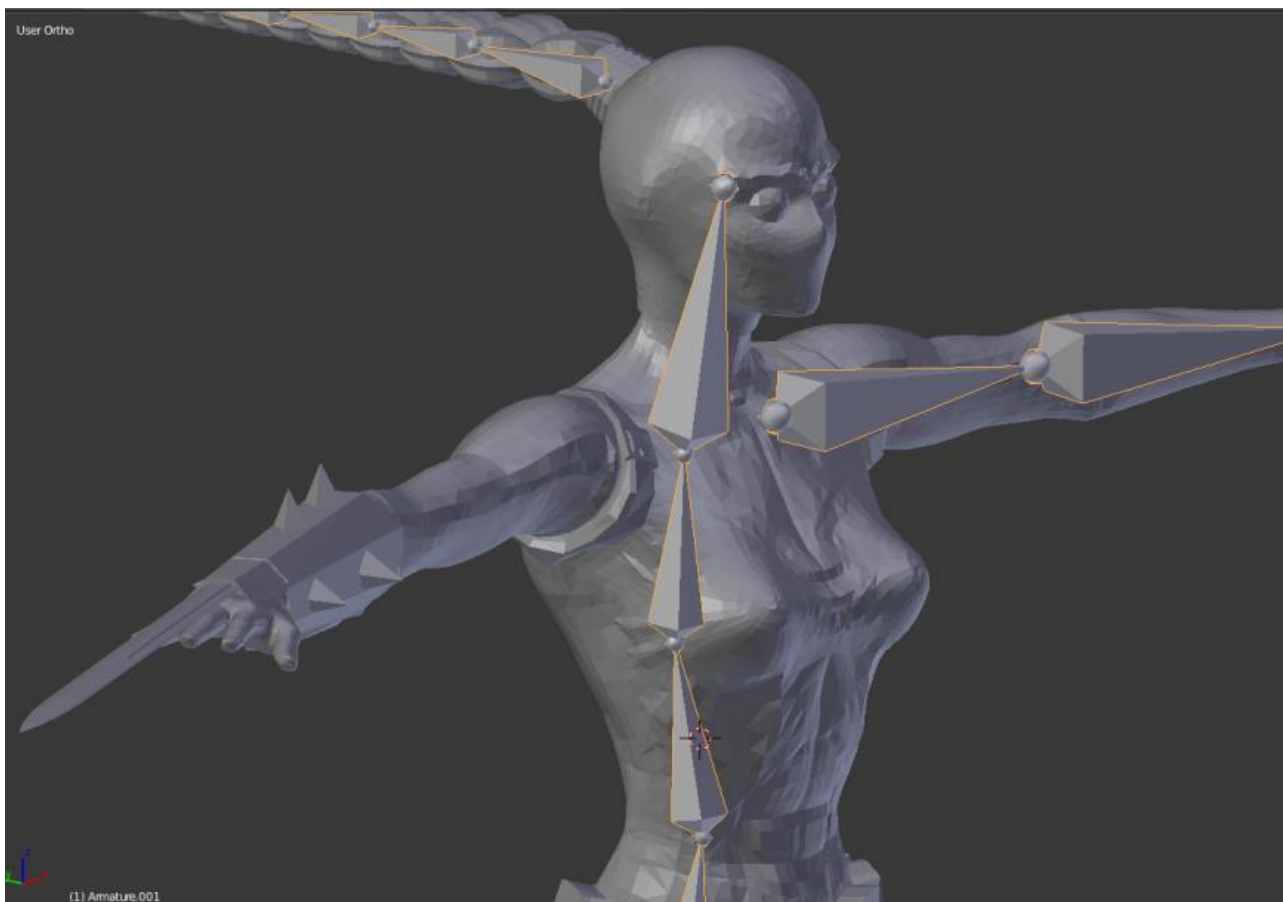


Рисунок 46. Пример скелета

К счастью, в Meta Human Creator у персонажей уже проставлены все кости, поэтому проставлять их вручную нет необходимости. Для этого нужно использовать встроенный MetaHuman\_Control\_Rig. Он появляется тогда, когда актёр добавляется в Level Sequence. Находится он во вкладке Body.

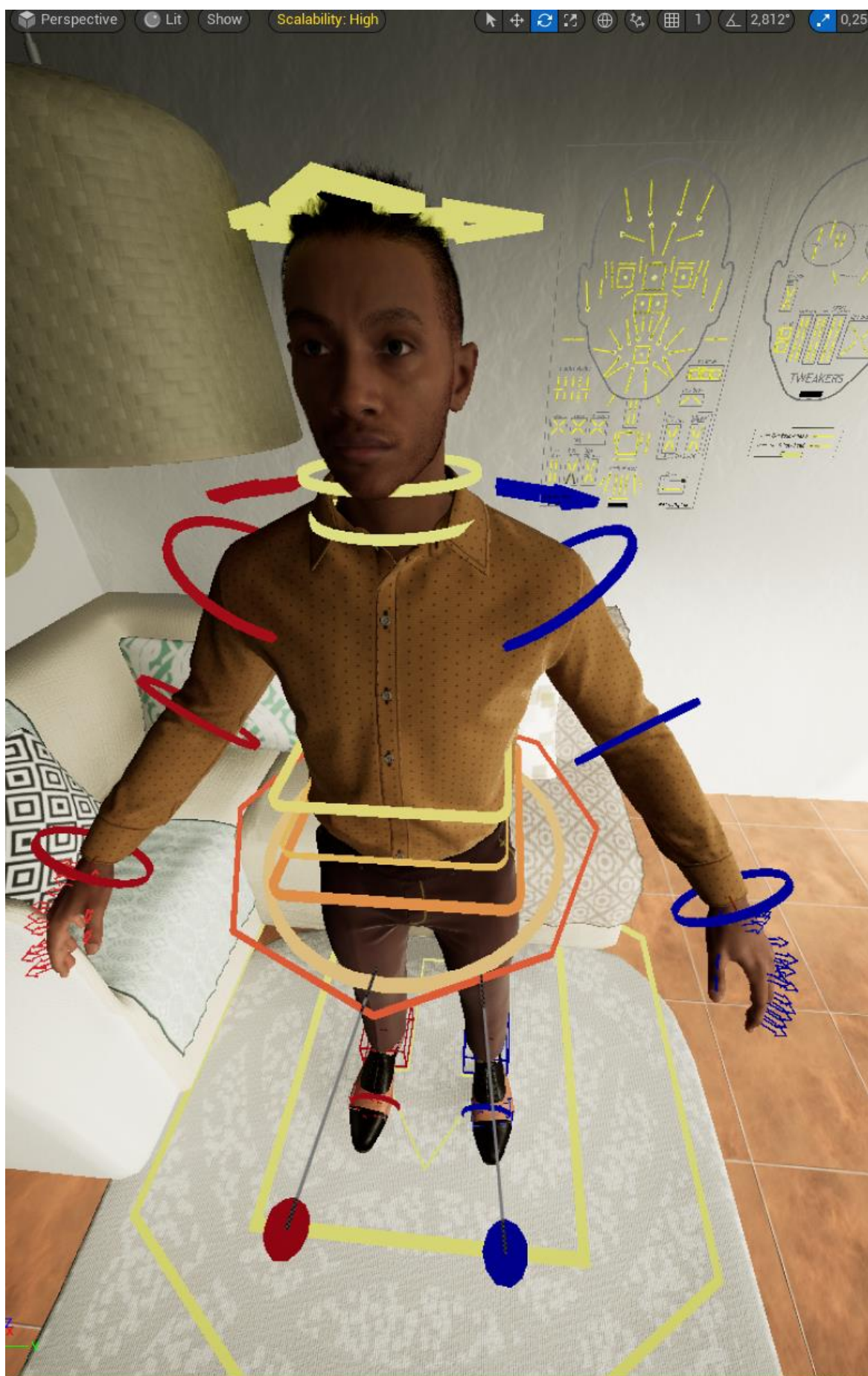


Рисунок 47. Расположение костей персонажа

## 7.1 Расположение персонажа с помощью Meta Human Control Rig

Итак, теперь можно расположить нашего персонажа так будто он сидит на диване. Все персонажи изначально имеют Т-образную позу [см. рисунок 42], это необходимо для более комфортного риггинга и детализации персонажа, так как все участки его тела находятся в доступности. Для начала нужно открыть MH\_CR\_Picker. Обычно он находится по пути All->Content->MetaHumans->Common->Common->Utilities. Кликаем на файл после чего должна открыться панель управления костями персонажа. Её можно перетащить в любой удобный участок экрана [5].

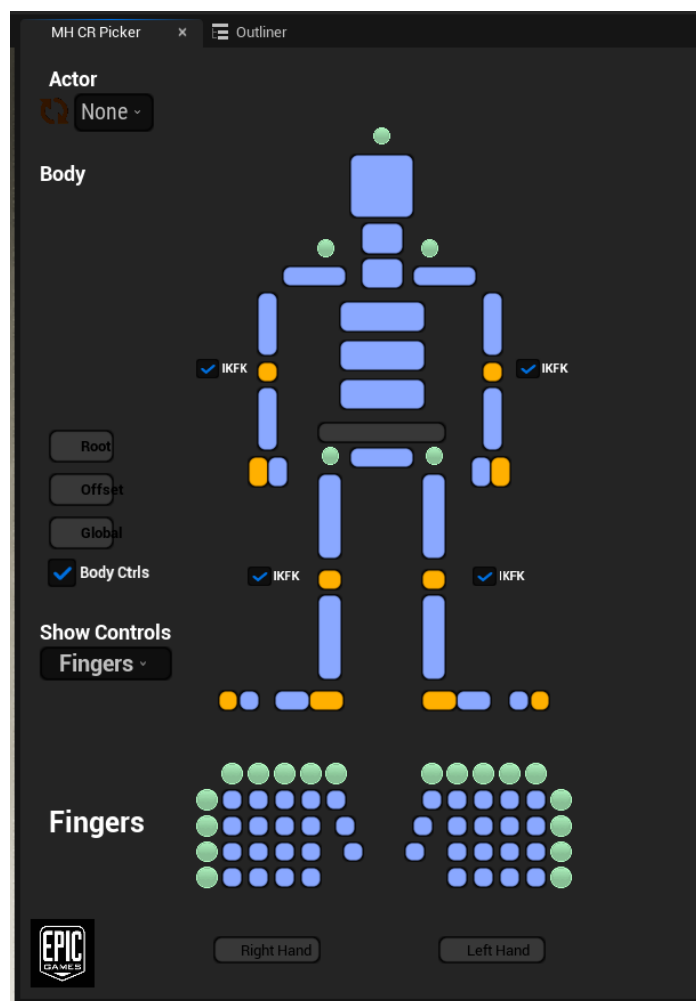


Рисунок 48. Панель управления костями

При нажатии на каждую из кнопок будет показано какая часть персонажа была выбрана и будет перемещена. Это крайне удобно так как нет необходимости вручную перебирать все кости в фитчерсе, а также это даёт понимание о том какой частью тела будет осуществлено перемещение.



Количество костей у персонажей Bryan и Patrick составляет 101, что весьма затрудняет ручной перебор. Все кости были у персонажей уже при скачивании, поэтому добавлять их вручную нет необходимости.

Для того, чтобы сохранить новое положение тела персонажа нужно нажать на кнопку “Add a new key at the current time” рядом с вкладкой MetaHuman\_ControlRig. Поза сохранится для текущего момента времени.

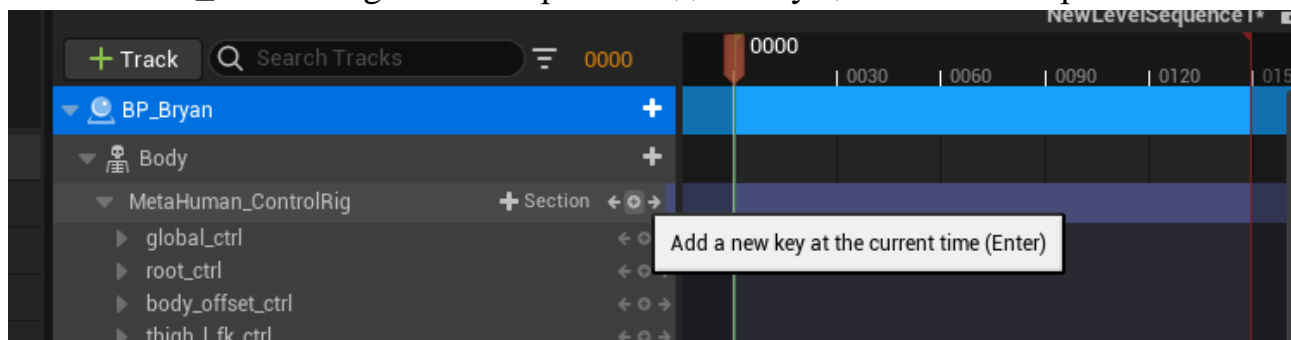
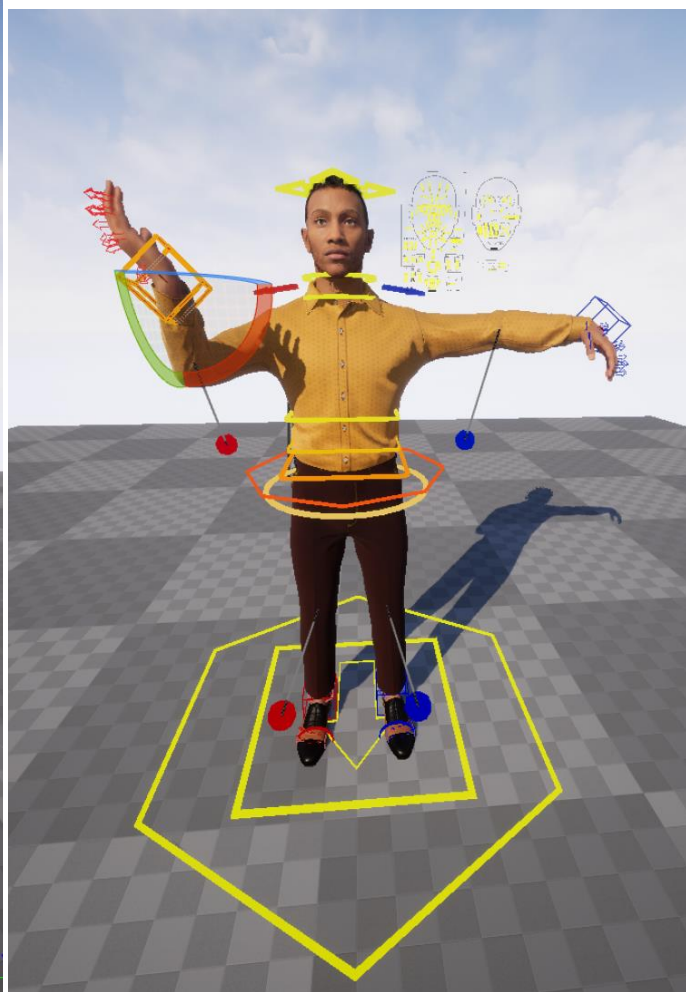
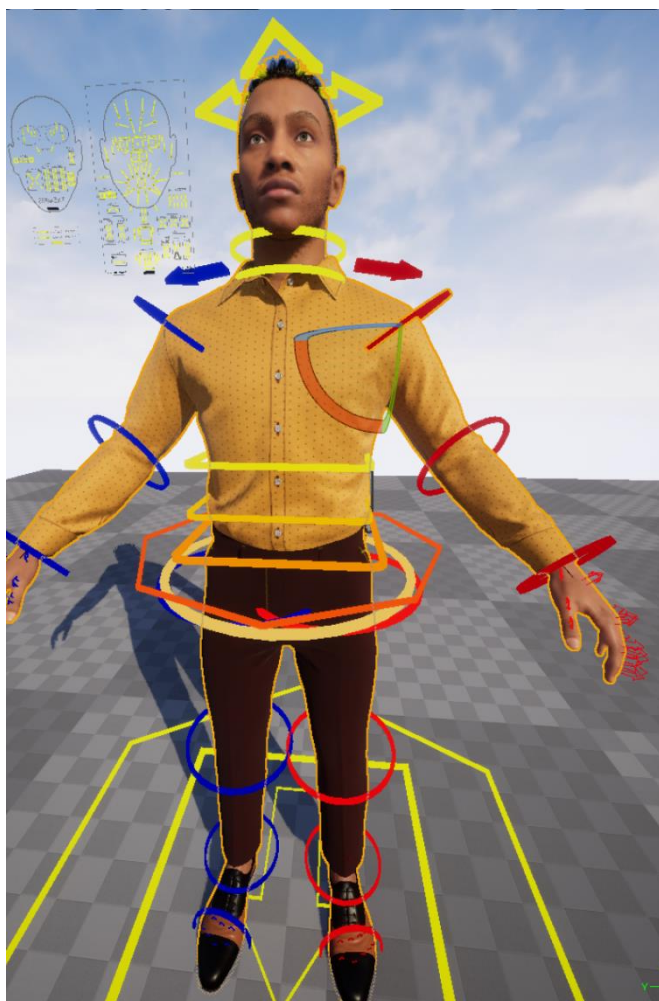


Рисунок 49. Сохранение нового положения персонажа



Рисунки 50, 51. Расположения персонажей до и после

## 8. Оборудование для работы с анимацией

Прежде чем начать работу с Unreal Engine, Meta Human Creator и анимацией в целом надо составить список необходимой материальной базы, которая будет использована в течение работы. Помимо программного обеспечения в данном деле также важны и инструменты, которыми работа производится и будет. Для точного считывания мимики необходима хорошая камера. Сама камера должна быть прикреплена на длинный шест, прикреплённый к шлему, который в свою очередь должен быть на человеке, с которого будет считываться мимика. Шлем необязательно должен быть заводского производства, он может быть сделан вручную.

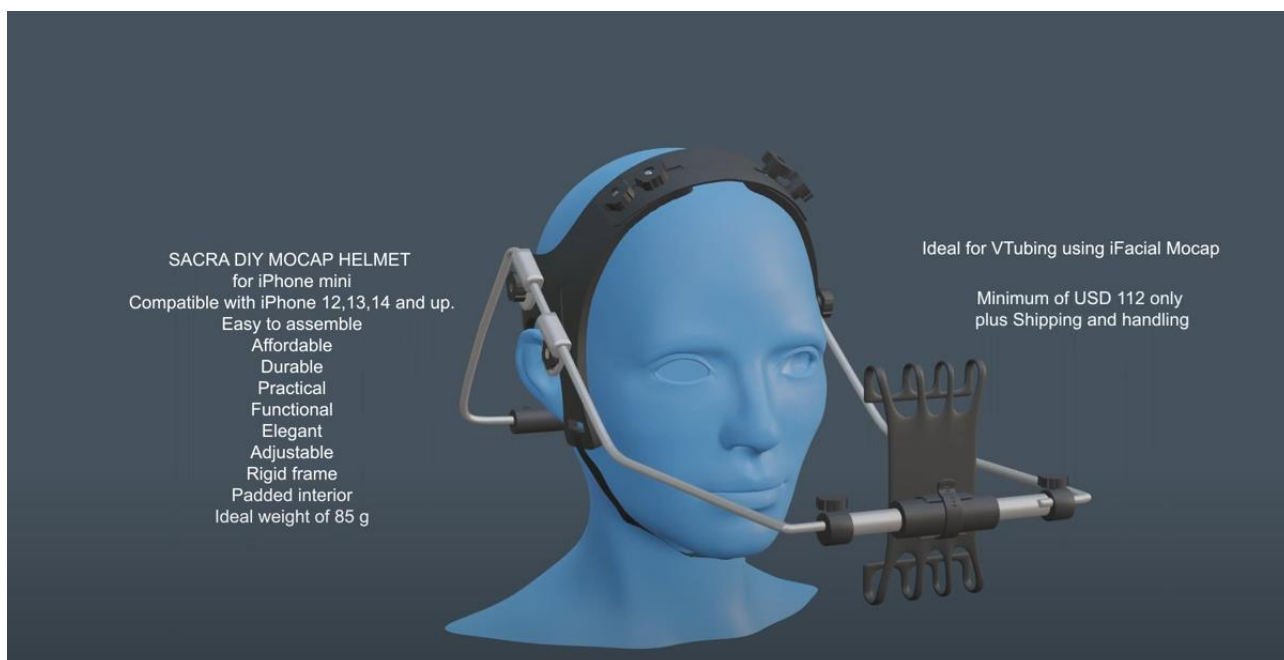


Рисунок 53. Самодельный шлем для считывания мимики

Поскольку в данной работе используется Live Link Face, то вместо камеры можно использовать iPhone прикрепив его на саму подставку. Также есть маркерный метод захвата мимики. Этот метод является довольно дорогостоящим, так как нуждается в камерах большого разрешения. Также нужно учесть факт того, что маркерный метод использует специальные датчики на лице человека[4]. Эти датчики как раз и называются маркерами. Движение маркеров считывается камерой и уже в виде скелета переносится на персонажа.





Рисунок 54. Считывание мимики с актёра маркерным методом

Итак, можно составить минимальный необходимый список оборудования для работы с лицевой анимацией:

- Камера высокого разрешения
- Крепление для камеры
- Шлем

## 9. Сравнение производительности разных моделей

Не мало важным параметром в анимации является оптимизация. Каким бы мощным не было бы железо на устройстве, на котором она создавалась на устройстве потребителя кинематографической продукции оно будет скорее всего слабее. Таким образом, игра или фильм могут воспроизводиться хуже или не воспроизвестись вообще. Поэтому очень важно при разработке игры учесть фактор оптимизации и до биться наибольшей производительности при наименьшей потере качества изображения. В Unreal Engine и Meta Human Creator эта тема также важна, так как они используются в разработке игр, а значит потребляют много вычислительных ресурсов и нуждаются в мощном железе, что далеко не все себе могут позволить. На ниже указанных рисунках представлены производительность модели Bryan качество Low, Medium и High и график изменения этой производительности в процентах.



Рисунок 55. Модель Bryan уровня Low. FPS = 65

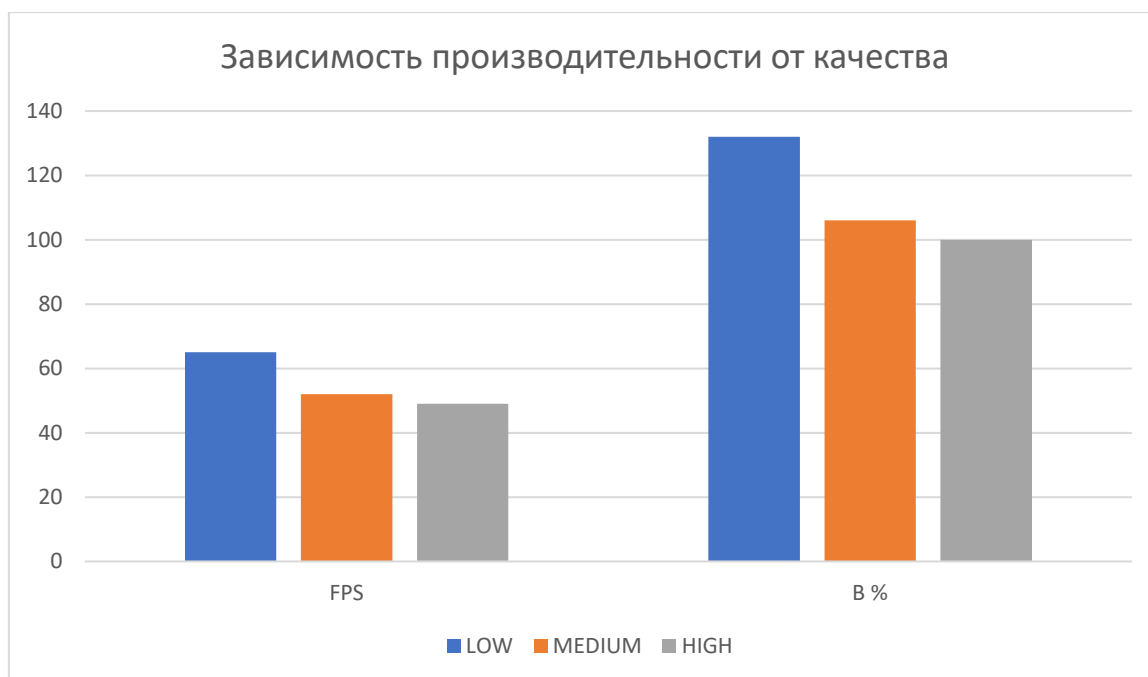


Рисунок 56. Модель Вгуап уровня High. FPS = 49



Рисунок 57. Модель Вгуап уровня Medium. FPS = 52

Как видно с улучшением качества падает число FPS, а значит на более слабом железе результат будет ещё хуже. Разница оказалась незначительной в данной случае, потому что устройство, на котором производилось сравнение обладает хорошими техническими показателями, но если бы видеокарта была намного слабее, то и результат был бы намного хуже, и разница была бы намного больше, чем в представленном эксперименте.



Также необходимо учесть зависимость производительности от LOD. Этот параметр влияет на количество детализации модели через изменение количества полигонов. Детализация меняется в зависимости от дальности камеры от персонажа. Чем ближе камера, тем лучше детализация. Уровень LOD можно установить как самый детальный при любых дистанциях, но тогда это будет сильно загружать систему. Графики зависимости производительности от уровня детализации представлены ниже вместе с фотографиями персонажа при LOD 0, 4, 7.

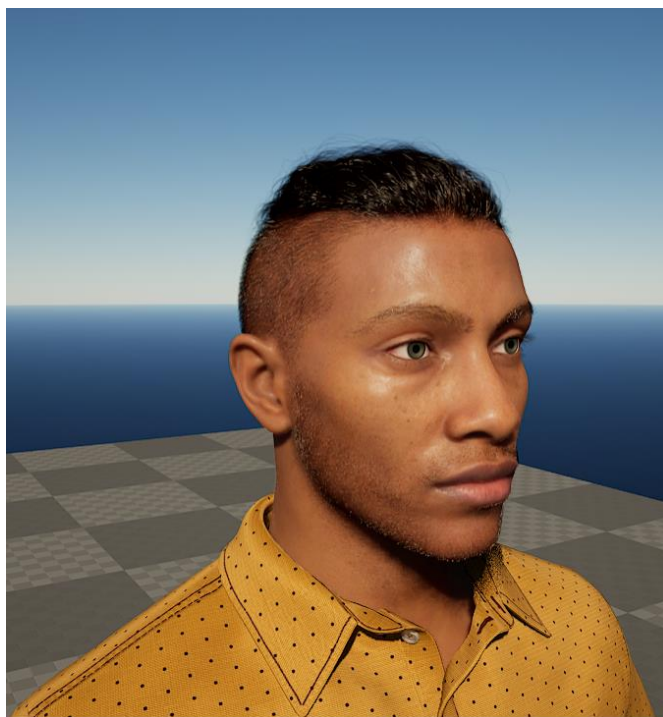


Рисунок 58. Модель Вryan при LOD 0. FPS = 70, 14.16 ms



Рисунок 59. Модель Вryan при LOD 4. FPS = 96, 11.56 ms

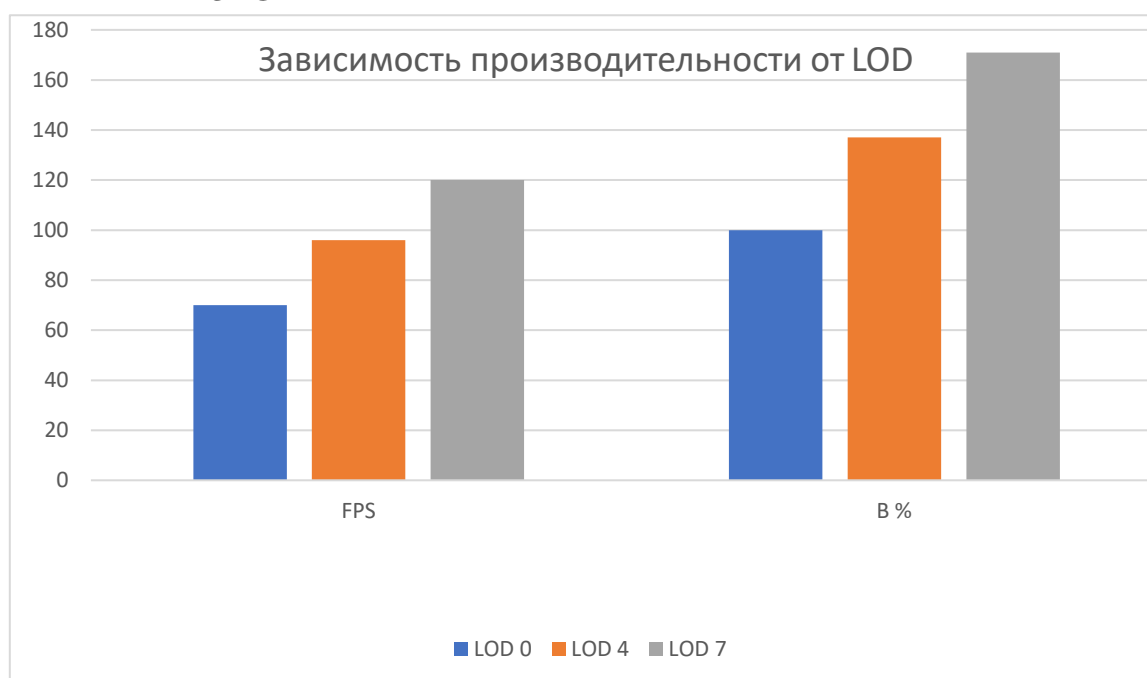




Рисунок 59. Модель Брайан при LOD 8. FPS = 120, 9.16 ms

Все проведённые эксперименты были проведены на устройстве со следующими комплектующими:

- Видео карта Nvidia GeForce RTX 3070 ti
- Процессор Intel Core i7-11700F
- 32GB RAM



## **10. MetaHuman для Vtube**

Vtube (Virtual Youtube) – это вариант прямой трансляции, при которой показывается не лицо живого человека, а 3D-персонажа, использующего считываемую в реальном времени с лица человека мимику. Такой метод трансляции может использоваться не только в развлекательных целях, но и в том случае, если человек не имеет желания или не может показать своё лицо. MetaHuman может использоваться для таких целей соответственно. Это повысит качество 3D-модели, а также процесс считывания мимики. К сожалению, данные технологии не имеют сильного развития, из-за чего имея хорошо сделанную 3D-модель не всегда получается получить хороший перенос мимики. Это и является причиной почему MetaHuman является одним из лучших инструментов для использования в Vtube. 3D-модель MetaHuman можно сделать собственноручно, а перенос мимики у него намного лучше, чем у любых других аналогов. Прежде чем начать заниматься Vtube необходимо установить OBS Studio (Open Broadcaster Software). Это ПО предназначено для ведения прямых трансляций и записей и имеет большую популярность среди многих стримеров благодаря простоте использования и широким спектром функций.

### **10.1 Установка OBS и необходимых плагинов к Unreal Engine**

Перед началом работы необходимо скачать сам OBS Studio. Это приложение абсолютно бесплатное и его спокойно можно скачать на официальном сайте. Далее после установки необходимо скачать Unreal Engine Toolkit от разработчика OFFWORLD на его сайте, где также находится Spout Plug-In for OBS Studio. В случае Unreal Engine Toolkit необходимо версию движка, на которую необходима установка и версию самого плагина. Toolkit устанавливается через инсталлер. При скачивании необходимо поставить галочку на пункт NDI 5 Runtime в случае его отсутствия, он необходим для работы плагина.

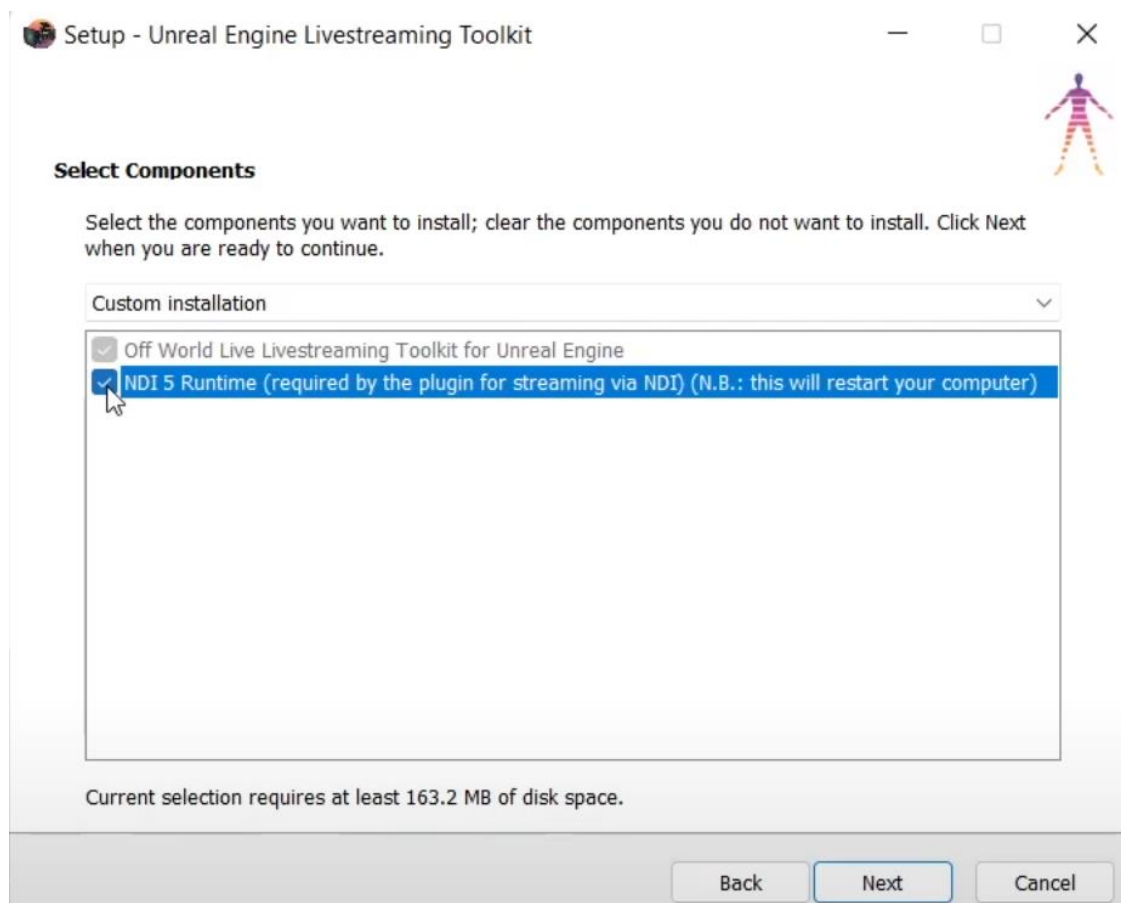


Рисунок 60. Пункт NDI 5 Runtime

После установки будет предложена возможность активировать ключ для использования плагина в коммерческих целях или же завершить установки для использования в личных целях. Отказ от активации не лишает каких-либо возможностей. Выбор делать по личному желанию.

SPOUT плагин для OBS устанавливается через инсталлер соответственно.

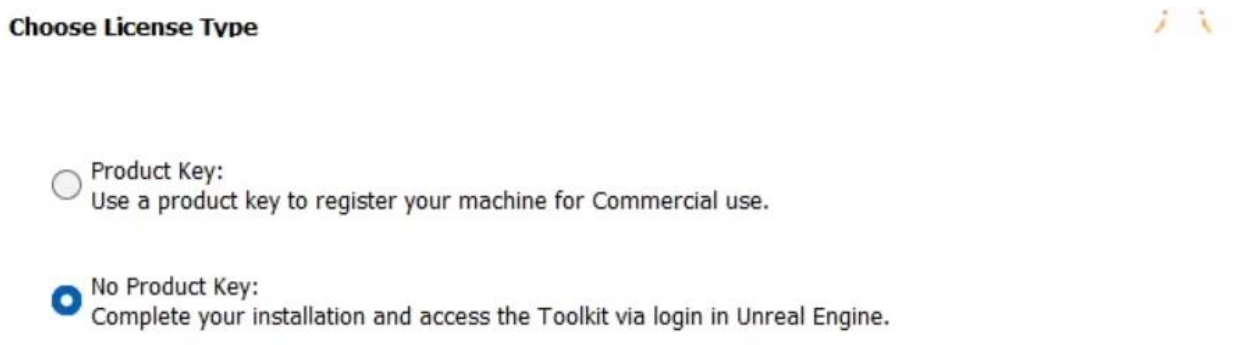


Рисунок 61. Выбор между активацией ключа и завершением установки



В случае возникновения проблем на сайте во вкладках скачивания плагинов есть видео-инструкция по установке[8].

## 10.2 Подключение трансляции из Unreal Engine к OBS

Для того чтобы начать трансляцию MetaHuman в OBS необходимо запустить Unreal Engine и открыть проект своего цифрового аватара и запустить Level, в котором находятся аватар в режиме New Editor Window.

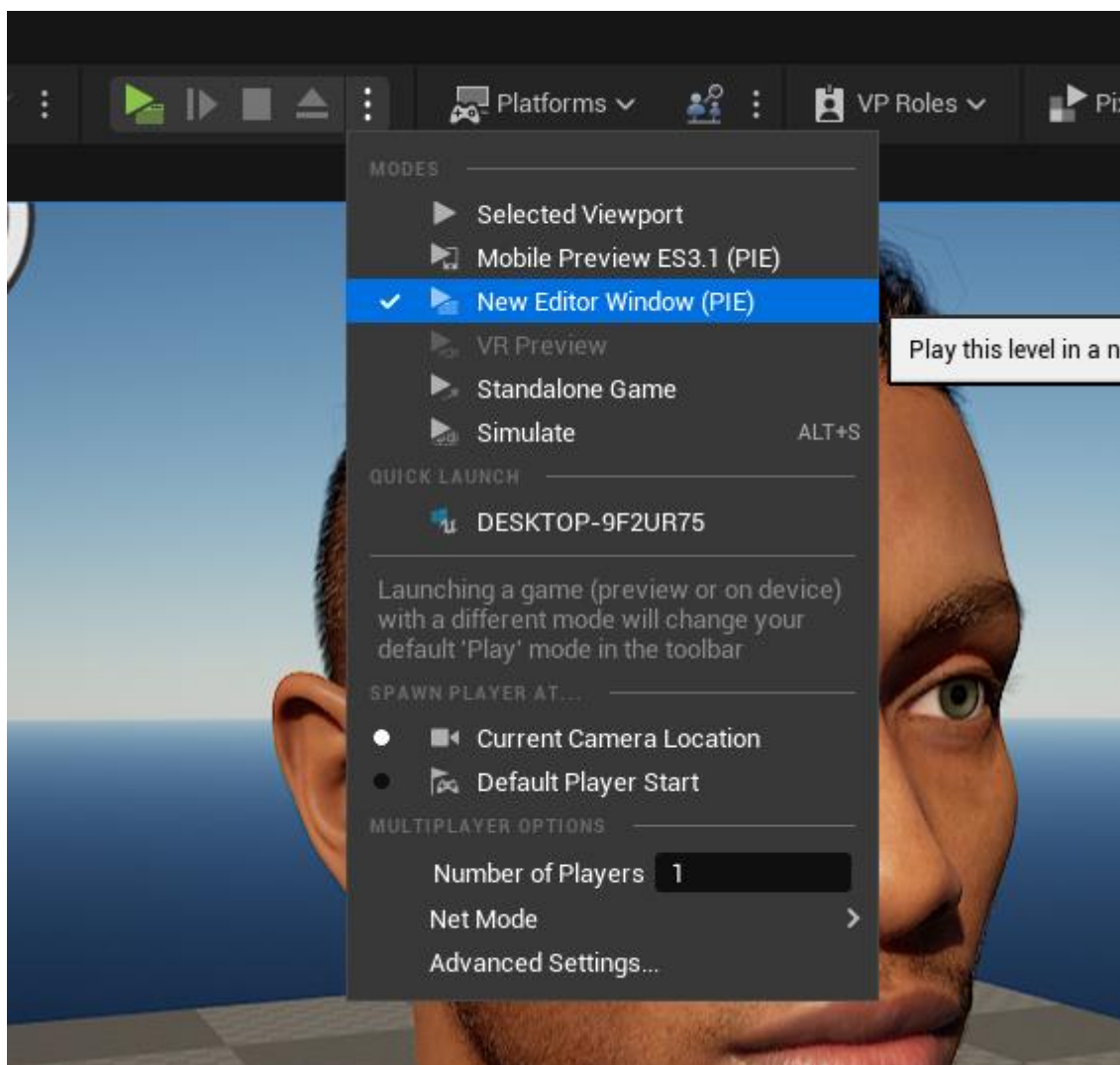


Рисунок 62. Запуск в режиме New Editor Window

Теперь можно открывать сам OBS. В нём нужно в графе Sources нажать на кнопку Add Sources в левом нижнем углу после чего откроется меню выбора. В нём нужно выбрать Window Capture и Game Capture, если целью является трансляция игры. Window Capture можно дать название Unreal, оно будет использоваться для трансляции цифрового аватара. Чтобы выбрать трансляцию

персонажа необходимо графе Window выбрать название окна, которое появилось после запуска level в режиме New Editor Window в Unreal Engine. В графе Capture Method нужно выбрать режим в зависимости от операционной системы. Выполнив все указанные пункты, в OBS должна появиться трансляция персонажа.

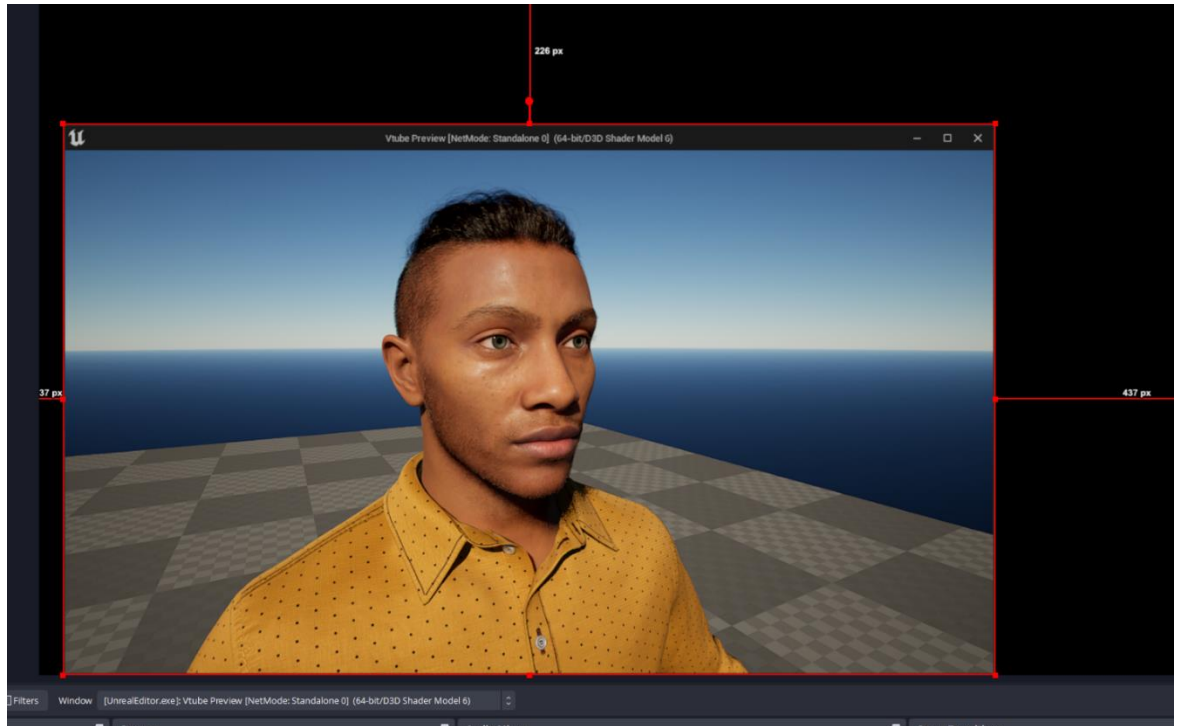


Рисунок 63. Трансляция персонажа в OBS Studio



Рис.64. Трансляция персонажа в OBS Studio вместе с игрой Elden Ring.

Для того, чтобы начать транслировать игру вместе с аватаром на фоне необходимо добавить Game Capture игры.

### 10.3 Трансляция Цифрового аватара в Twitch

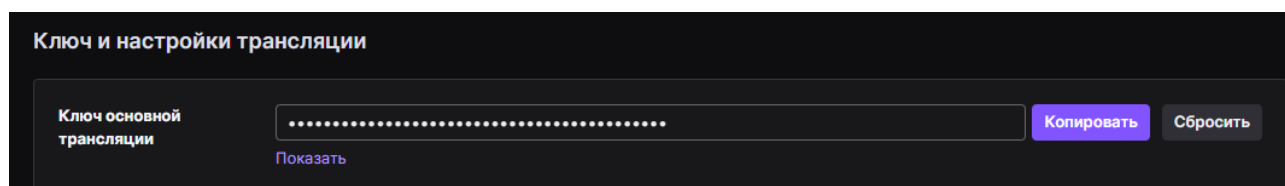


Рис.65. Streaming Key в настройках трансляции.

Чтобы начать трансляцию в Twitch необходимо в OBS Studio необходимо в настройках указать Streaming Key Twitch-аккаунта. Это ключ не должен попадать другим людям, так как он позволяет вести трансляцию с аккаунта в любом месте, в любое время и с любого устройства [10].

## ЗАКЛЮЧЕНИЕ

В результате работы был разработан метод автоматизирования процесса создания 3D-фильма. Также был изучен цикл создание модели Meta Human и перенос её в Unreal Engine. Приобретены знания об организации проектов.

Была разработана система позволяющая накладывать мимику на 3D-персонажа Meta Human, находящегося в Unreal Engine, в режиме реального времени и транслировать это персонажа в прямой эфир в Twitch и иные приложения с помощью OBS Studio.

Проведён анализ сравнения зависимости производительности FPS от качества текстур и LOD 3D-модели.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. MetaHuman – Unreal Engine // MetaHuman URL: <https://www.unrealengine.com/en-US/metahuman/> Дата обращения: 18.10.2022;
2. Youtube – Digital School Creator // Digital School Creator URL: <https://www.youtube.com/@DigitalCreatorSchool> Дата обращение: 18.10.2022;
3. Youtube – JSFILMZ // JSFILMZ URL: <https://www.youtube.com/@Jsfilmz> Дата обращения: 03.11.2022;
4. Playground – Как создать лицевую анимацию в играх // Как создать лицевую анимацию в играх URL: [https://www.playground.ru/misc/news/kak\\_sozdayut\\_litsevuuyu\\_animatsiyu\\_v\\_igrah-356274](https://www.playground.ru/misc/news/kak_sozdayut_litsevuuyu_animatsiyu_v_igrah-356274) Дата обращения: 01.10.2022;
5. Youtube – Cinematic // Cinematic URL: [https://www.youtube.com/@cinematic\\_yari](https://www.youtube.com/@cinematic_yari) Дата обращения: 19.11.2022;
6. Gaelle Morand, Charles de Rousiers, Michael Forot Unreal Engine Hair and Fur – GROOMING FOR REAL-TIME REALISM: HAIR AND FUR WITH UNREAL ENGINE: учебное пособие // Gaelle Morand, Charles de Rousiers, Michael Forot; Дата обращения: 07.04.2023;
7. Unreal Engine 4 Documentation // Unreal Engine Documentation URL: <https://docs.unrealengine.com/>. Дата обращения: 07.04.2023;
8. Youtube – OFF WORLD LIVE // OFF WORLD LIVE URL: <https://www.youtube.com/@offworldlive> Дата обращения 27.05.2023;
9. Youtube – MetaHuman SDK // MetaHuman SDK URL: <https://www.youtube.com/@metahumansdk> Дата обращения: 29.05.2023.
10. Youtube – RAZI da BULLY // RAZI da Bully URL: <https://www.youtube.com/@RAZIdaBULLY> Дата обращения: 20.05.2023