



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 (РК)

КАФЕДРА

Системы автоматизированного проектирования (РК6)

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## *К КУРСОВОМУ ПРОЕКТУ*

*НА ТЕМУ:*

*«Методы считывания мимики с помощью Meta Human  
Creator для анимации персонажей в Unreal Engine»*

Студент РК6-72Б

\_\_\_\_\_  
(Подпись, дата)

**Эвоян Э.Б.**

И.О. Фамилия

Руководитель курсового проекта

\_\_\_\_\_  
(Подпись, дата)

**Витюков Ф.А.**

И.О. Фамилия

2022 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  
А.П. Карпенко

«\_\_\_\_\_» \_\_\_\_\_ 2022 г.

## ЗАДАНИЕ на выполнение курсового проекта

по теме: Методы считывания мимики с помощью Meta Human Creator для анимации персонажей в Unreal Engine \_\_\_\_\_

Студент группы РК6-72Б

Эвоян Эрик Борисович  
(Фамилия, имя, отчество)

Направленность КП (учебная, исследовательская, практическая, производственная, др.) учебная  
Источник тематики (кафедра, предприятие, КП) кафедра

График выполнения КП: 25% к 5 нед., 50% к 11 нед., 75% к 14 нед., 100% к 16 нед.

**Техническое задание:** Разработать метод для считывания мимики лица человека и наложения этой мимики на mesh в Unreal Engine 4.

### **Оформление научно-исследовательской работы:**

Расчетно-пояснительная записка на 47 листах формата А4.

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

---

---

---

---

Дата выдачи задания «10» сентября 2022 г.

Руководитель КП

\_\_\_\_\_  
(Подпись, дата)

**Витюков Ф.А.**

И.О. Фамилия

Студент

\_\_\_\_\_  
(Подпись, дата)

**Эвоян Э.Б.**

И.О. Фамилия

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

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1. Обзор Meta Human Creator .....	5
1.1 Инструменты для считывания мимики лица .....	5
2. Необходимые плагины для Unreal Engine 4 с использованием Live Link Face .....	6
2.1 Необходимые плагины для Unreal Engine 4 с использованием Faceware Studio .....	7
2.2 Плюсы и минусы Live Link Face и Faceware Studio .....	8
3. Обзор Quixel Bridge .....	9
3.1 Перенос 3D-модели из Quixel Bridge в проект Unreal Engine .....	10
4. Наложение мимики на Mesh персонажа .....	13
4.1 Подключение iPhone к Unreal Engine .....	13
4.2 Анимация мимики лица .....	14
4.3 Перенос персонажа на сцену .....	17
5. Исправление ошибки с отсутствием волос на лице персонажа .....	19
5.1 Отсутствие волос из-за не прорисовки .....	19
5.2 Отсутствие волос из-за отсутствия нужных плагинов .....	19
5.3 Отсутствие волос из-за ошибки прорисовки .....	21
5.4 Отсутствие волос из-за ошибки импорта файлов .....	23
6. Озвучка персонажа .....	27
6.1. Преобразование текста в речь с помощью Replica .....	27
6.2 Lip sync персонажа с помощью MetahumanSDK .....	31
6.3 Создание анимации по аудиозаписи .....	32
6.4 Наложение анимации на персонажа .....	35
7. Rigging персонажа .....	37
7.1 Расположение персонажа с помощью Meta Human Control Rig .....	39
8. Оборудование для работы с анимацией .....	41
9. Сравнение производительности разных моделей .....	43
ЗАКЛЮЧЕНИЕ .....	46
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	47

## ВВЕДЕНИЕ

Фильмы, мультипликация, игры, реклама и еще пред десятком других направлений, использующих виртуальных персонажей, стоит задача выражать эмоции не только по средствам движений, но и мимики лица.

При анимации персонажа одним из самых сложных элементов является его лицо. Чтобы сделать искреннюю улыбку надо вложить достаточно много сил так как за счёт выражения эмоций идёт восприятие персонажа человеком. Но анимация любит точность. Есть большая разница между искренней и наигранной улыбкой, и разница эта происходит за счёт мелких деталей на лице человека, и как раз именно эти детали труднее всего реализовать на персонаже вручную.

Существует Множество различных программ, которые предназначены для анимации персонажа такие как Autodesk Maya, Blender и 3D Max. Но для того, чтобы сделать персонажа с помощью этих программ нужно обладать определёнными навыками, которые получить не так-то и просто. На освоение уходят годы, но помимо времени необходимы также ещё и деньги для покупки этих ПО. Хотя Blender и является бесплатной программой, она не может предложить то, что могут предложить Autodesk Maya и 3D Max, но эти программы являются платными по подписке. Помимо прочего затраты уходят ещё и на покупку необходимой аппаратной составляющей для работы с этими программами так как они довольно затрачивают много ресурсов. Также надо нанимать людей, которые умеют обращаться с этими программами, что, естественно, требует денег.

Не все компании могут позволить себе такие затраты. Поэтому существует Meta Human Creator. Это инструмент, который позволяет создавать 3D персонажей без лишних затрат и не требует навыков обращения со сложным 3D-графическим ПО.

Целью данной работы является изучение методов считывания мимики лица человека и перенос этой мимики на 3D-персонажа в Unreal Engine . В рамках достижения данной цели предполагается решение следующих задач:

## **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.

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 4 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.

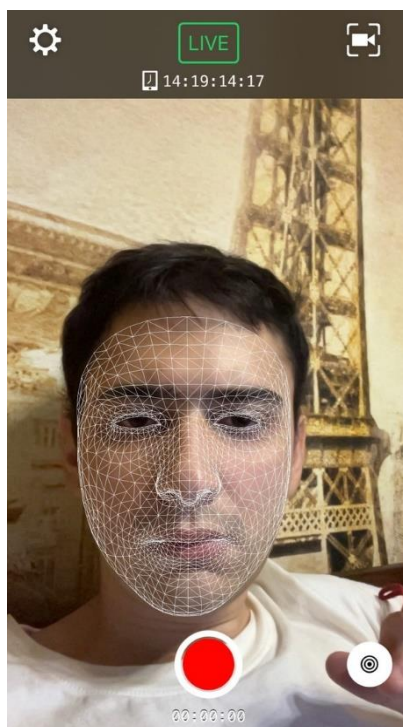


Рис.1. Считывание мимики в Live Link Face

## **2.1 Необходимые плагины для Unreal Engine 4 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.

Для того, чтобы можно было использовать Faceware Studio нам необходимо подключить лишь один плагин в проекте Unreal Engine 4. Это плагин Faceware Live Link Plugin.

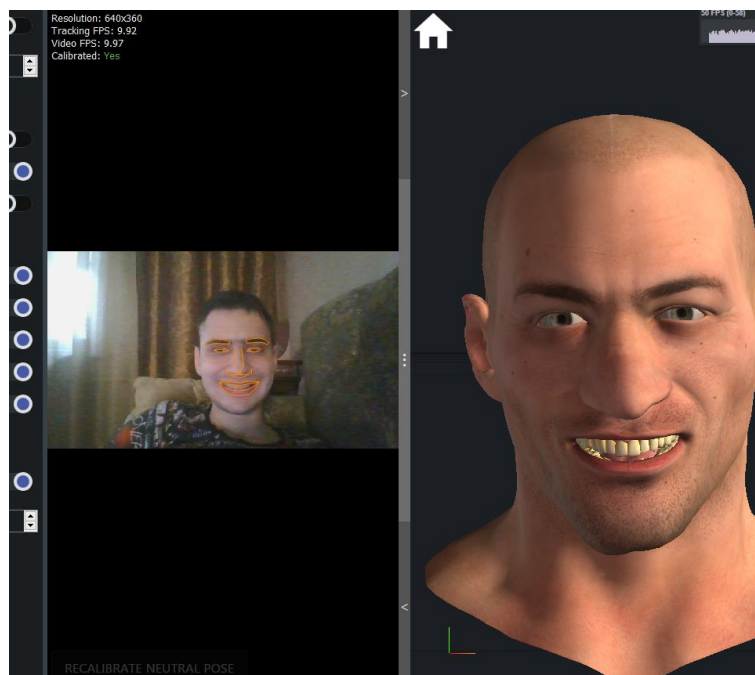


Рис.2. Считывание мимики в Faceware Studio

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.

## 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

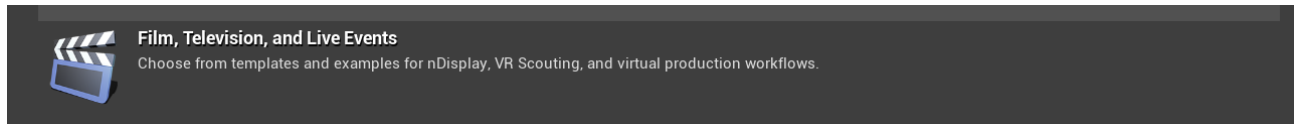


Рис.3. Тип проекта **Film, Television and Live Events** в меню

- Выбрать пункт Blank в следующем меню



Рис.4. Пункт Blank

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

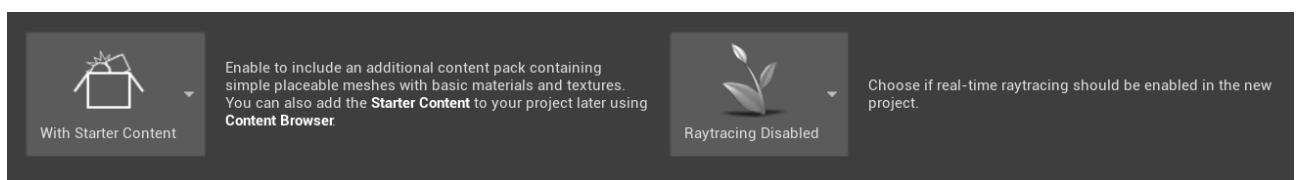


Рис.5. Выбор последних настроек

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.

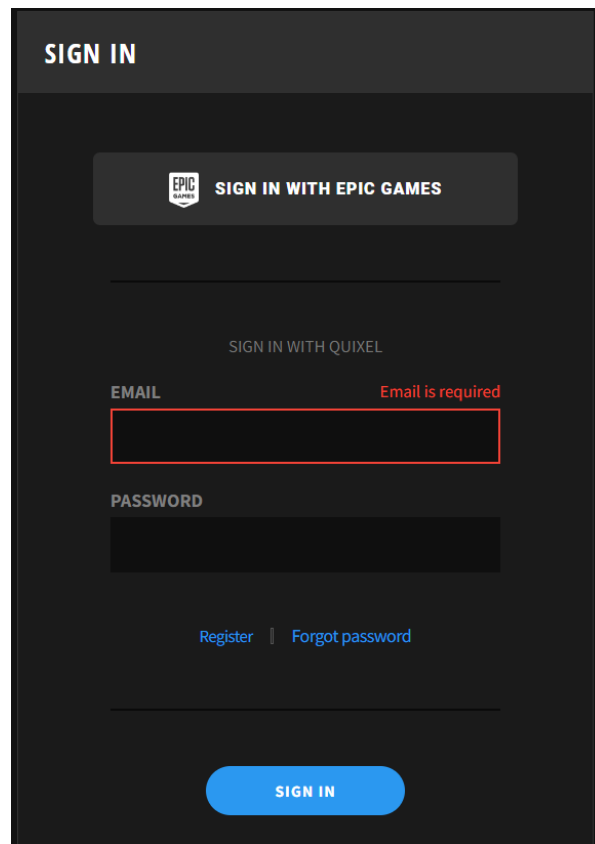


Рис.6. Меню регистрации в Quixel Bridge

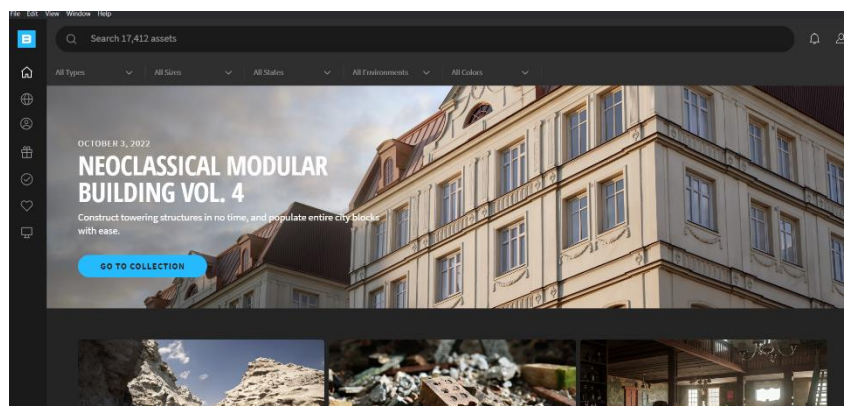


Рис.7. Главная страница Quixel Bridge

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

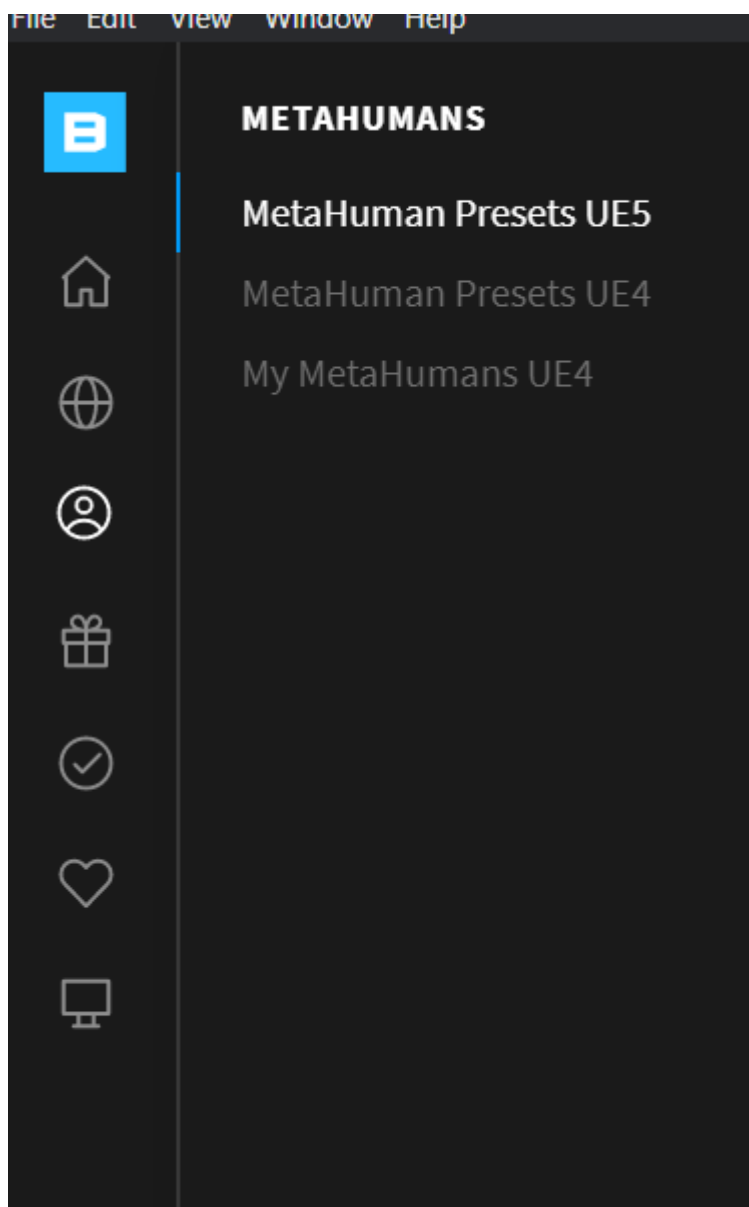


Рис.8. Вкладка MetaHumans

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.

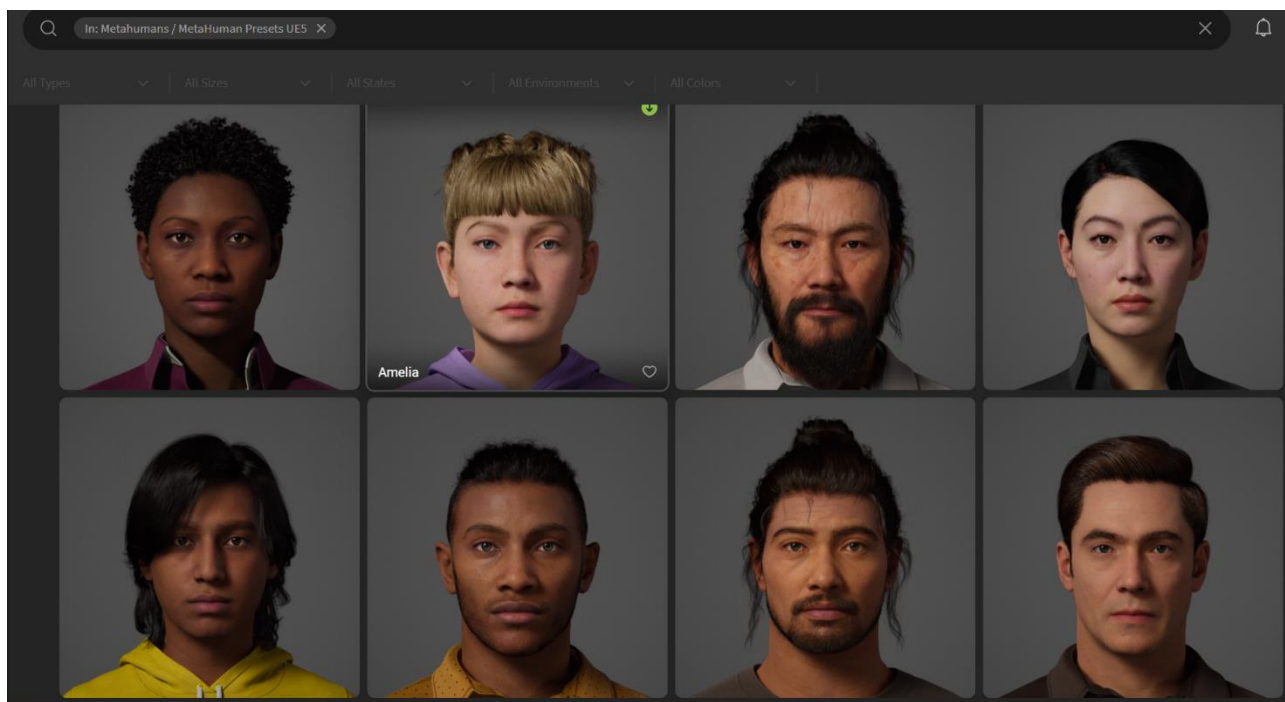


Рис.9. Меню выбора 3D-модели для импорта в проект

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.

#### 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 это 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.

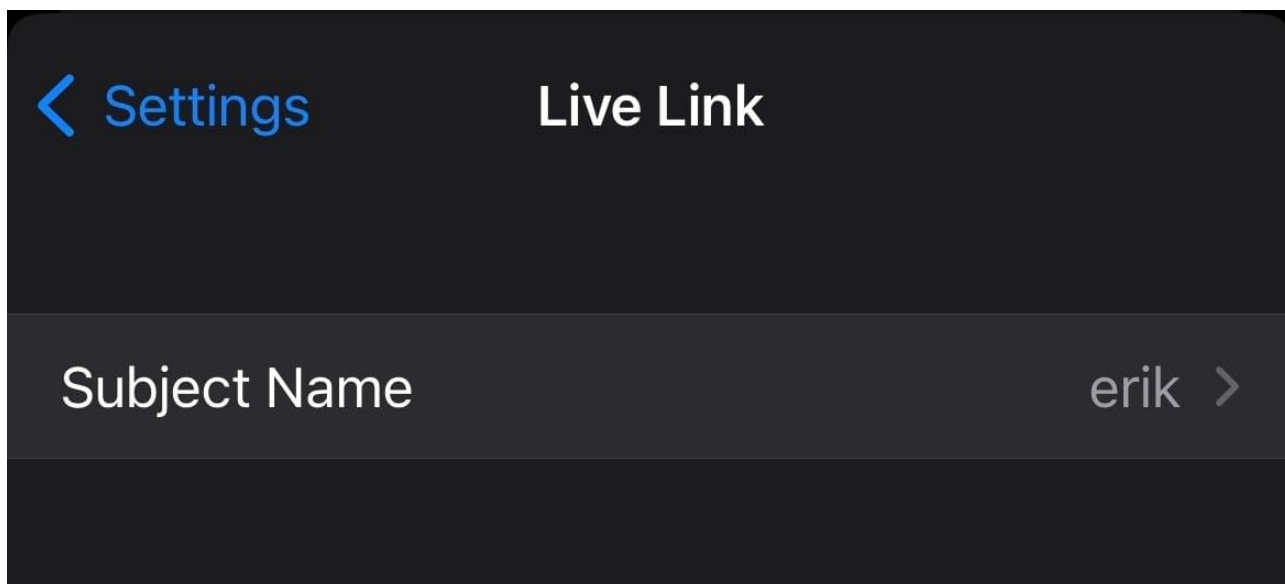


Рис.10. Графа для задания имени для iPhone

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.

#### **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.

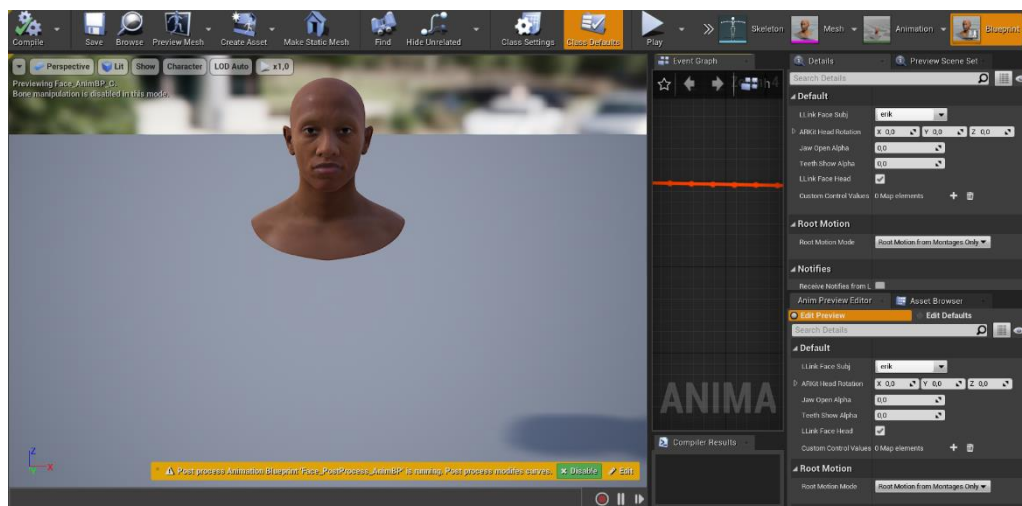


Рис.11. Viewport Face\_AnimBP

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

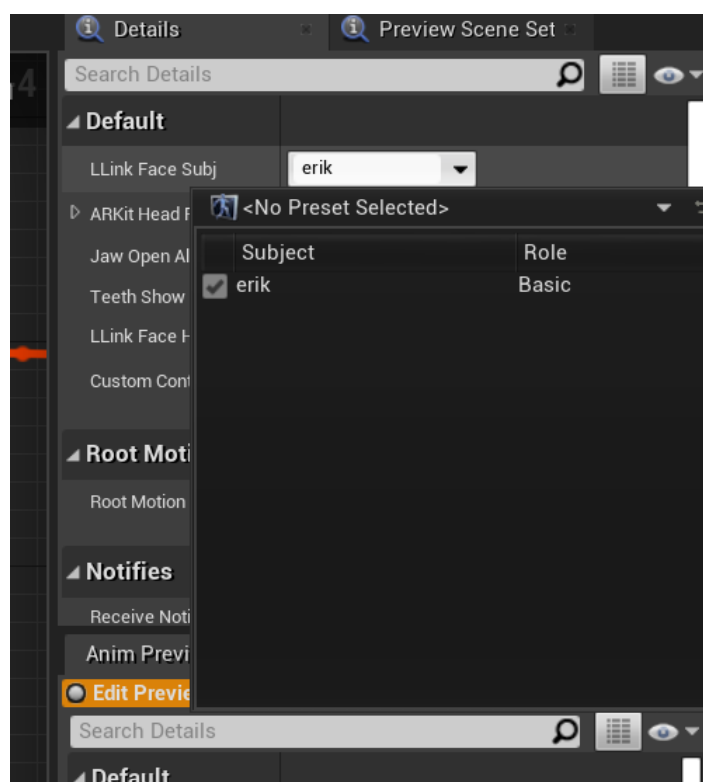


Рис.12. Выбор iPhone для подключения

После подключения устройства должна накладываться анимация на Mesh персонажа.



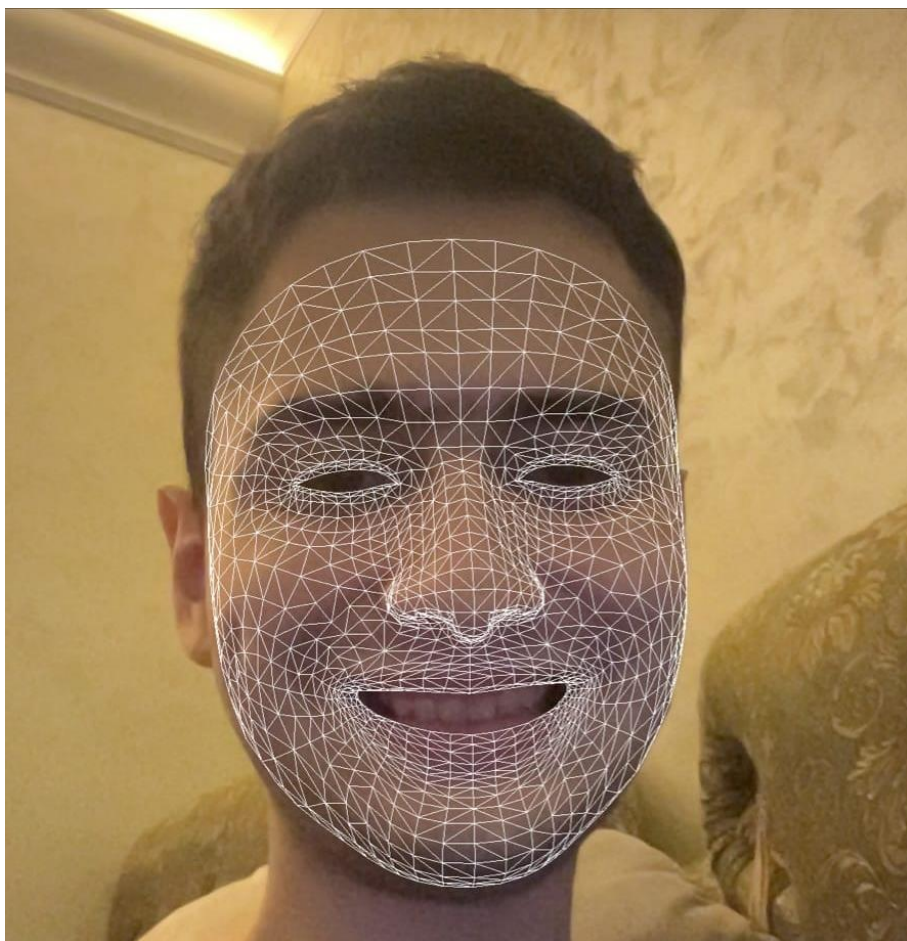


Рис.13. Лицо, с которого считывается мимика



Рис.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.

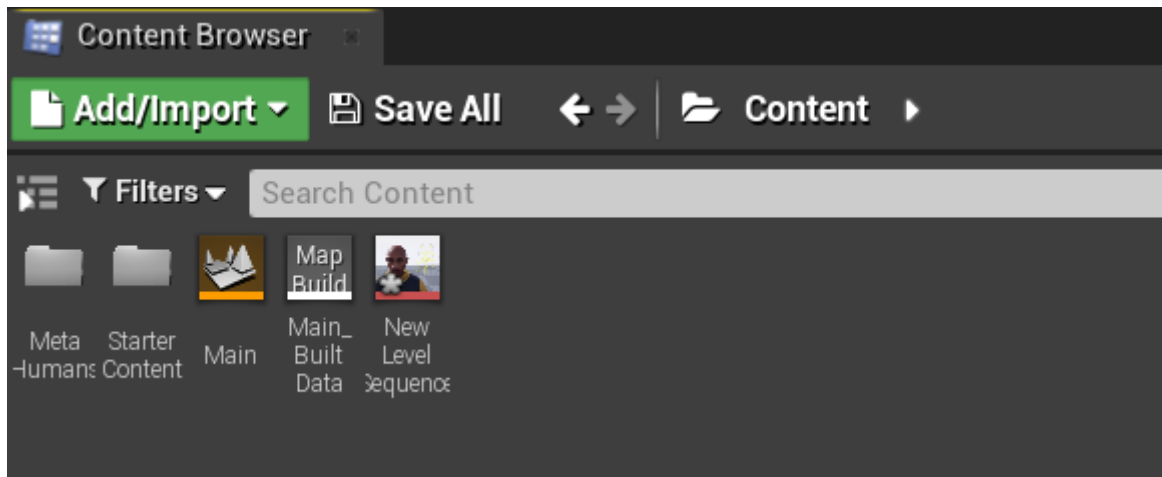


Рис.15. Директория Content

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

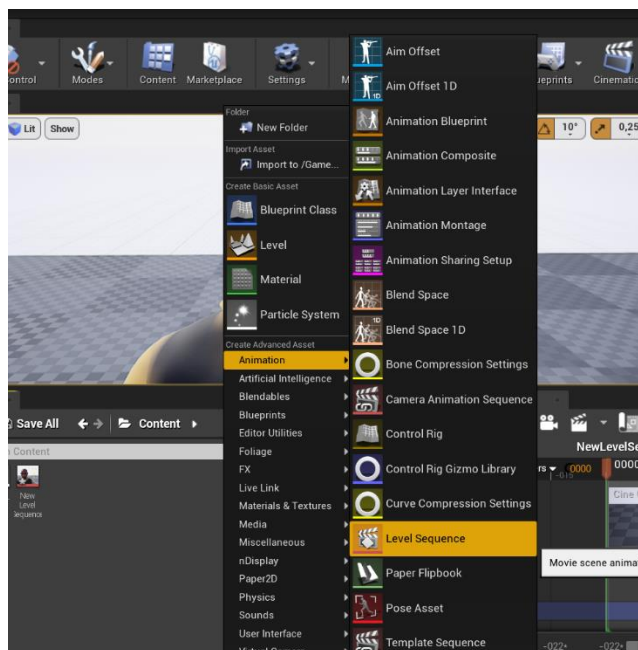


Рис.16. Создание Level Sequence

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).

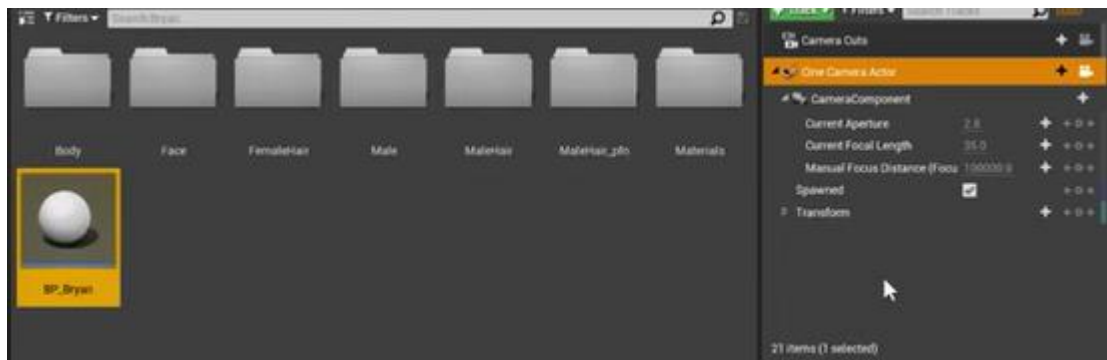


Рис.17. Перенос BP\_Bryan из директории в New Sequence

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.



Рис.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.

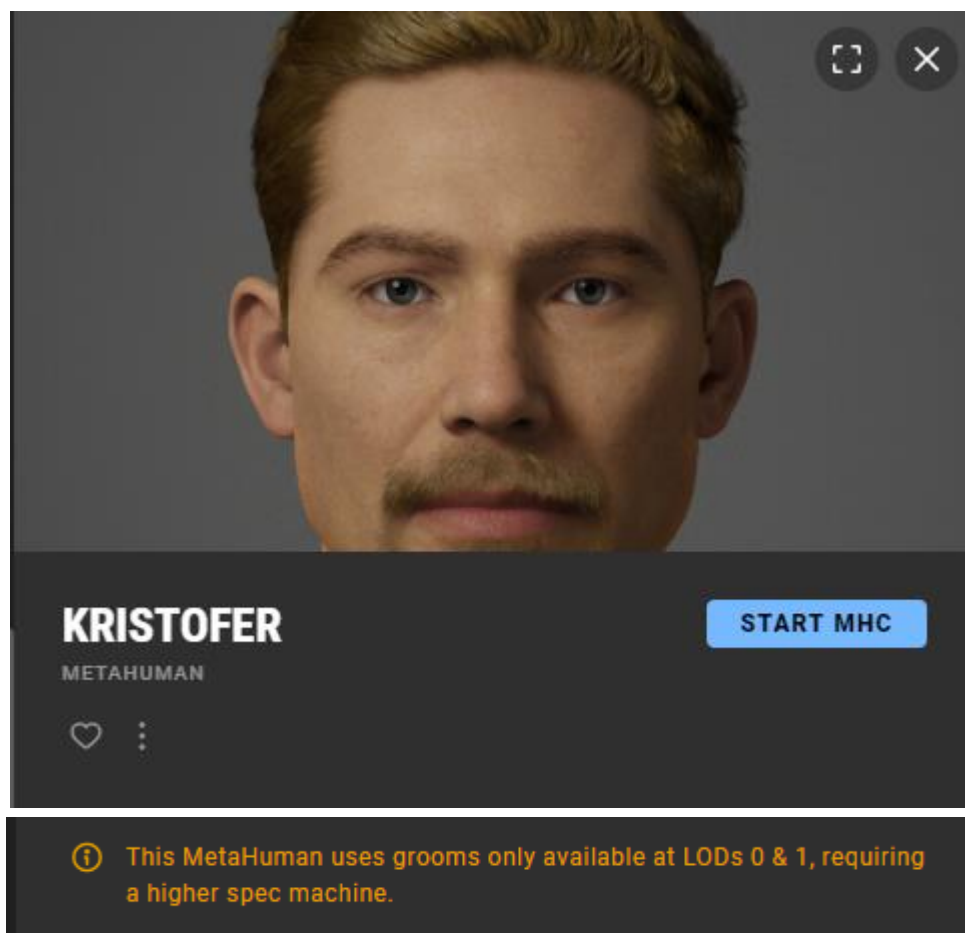


Рис.19. Предупреждение на модели персонажа

### 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.

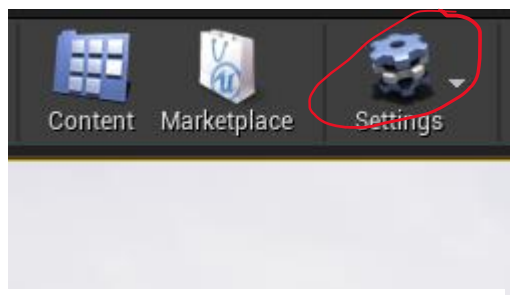


Рис.20. Меню настроек

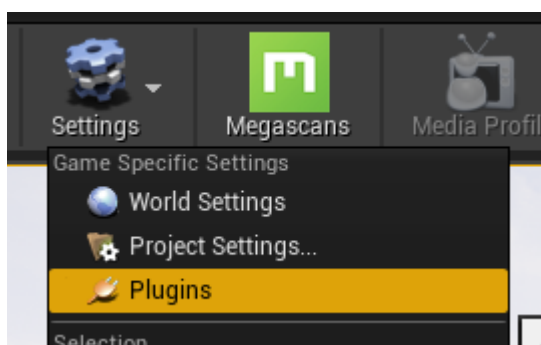


Рис.21. Меню plugins

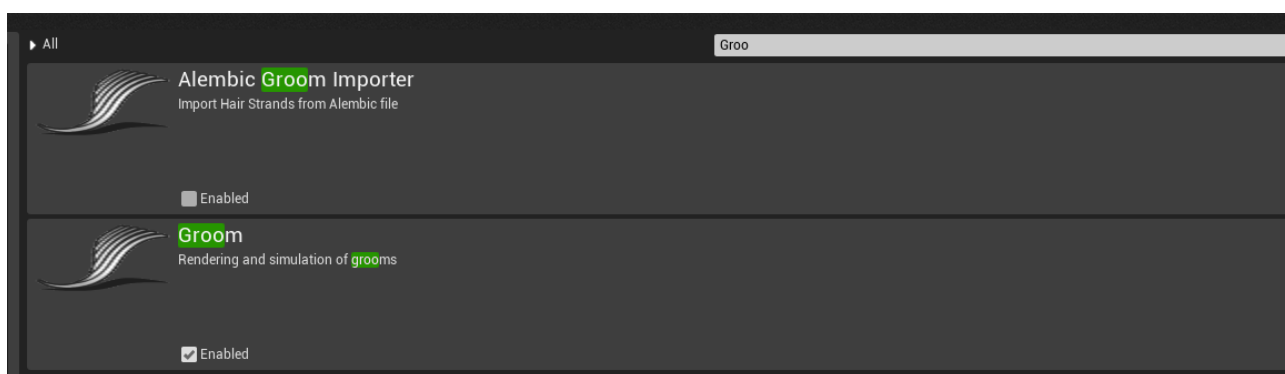


Рис.22. Плагин Groom, который необходимо подключить к проекту

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

### 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 BP файл персонажа, который обычно находится по пути Content/MetaHumans/CharacterName. Пройдя по этому пути нужно открыть файл BP\_CharacterName.

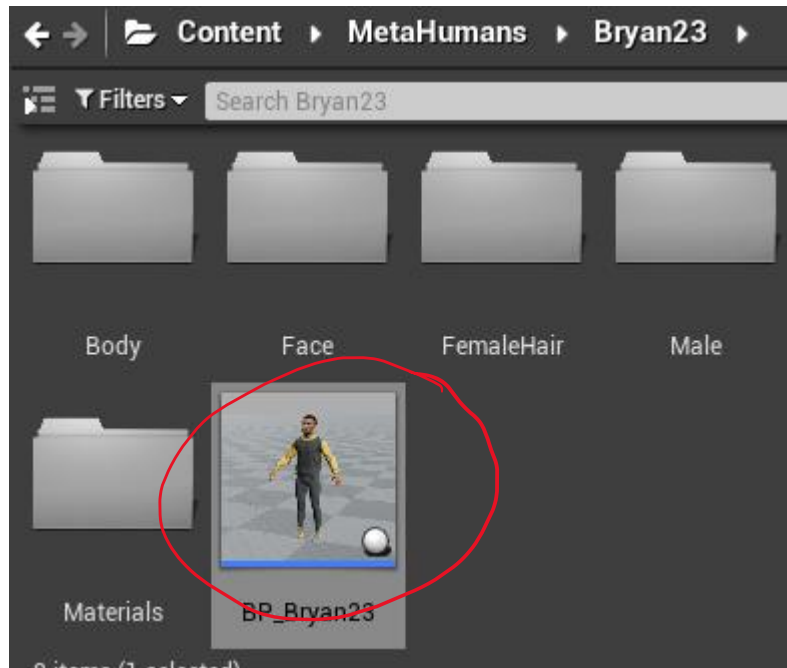


Рис.22. Файл BP персонажа Bryan23

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

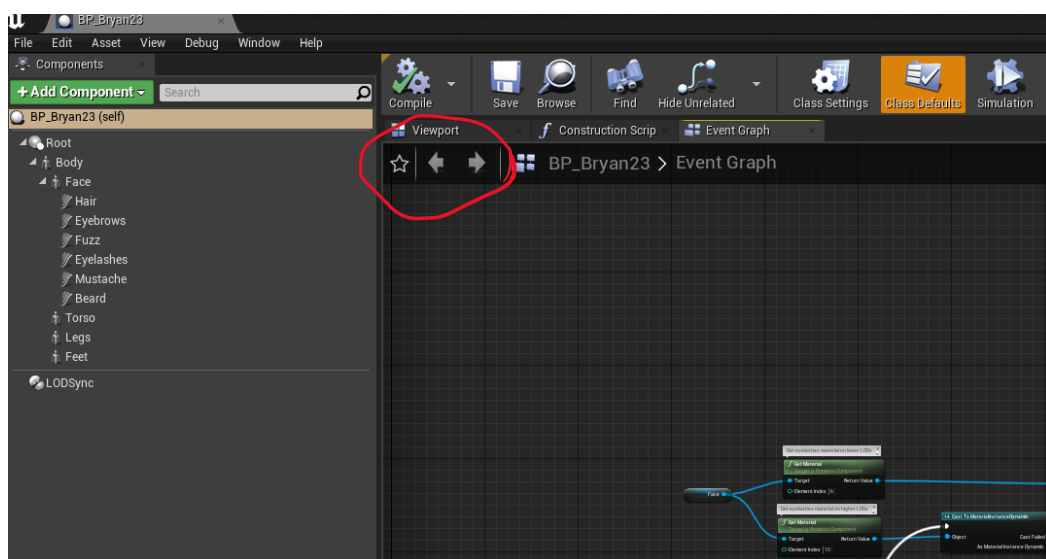


Рис.23. Окно настроек персонажа

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

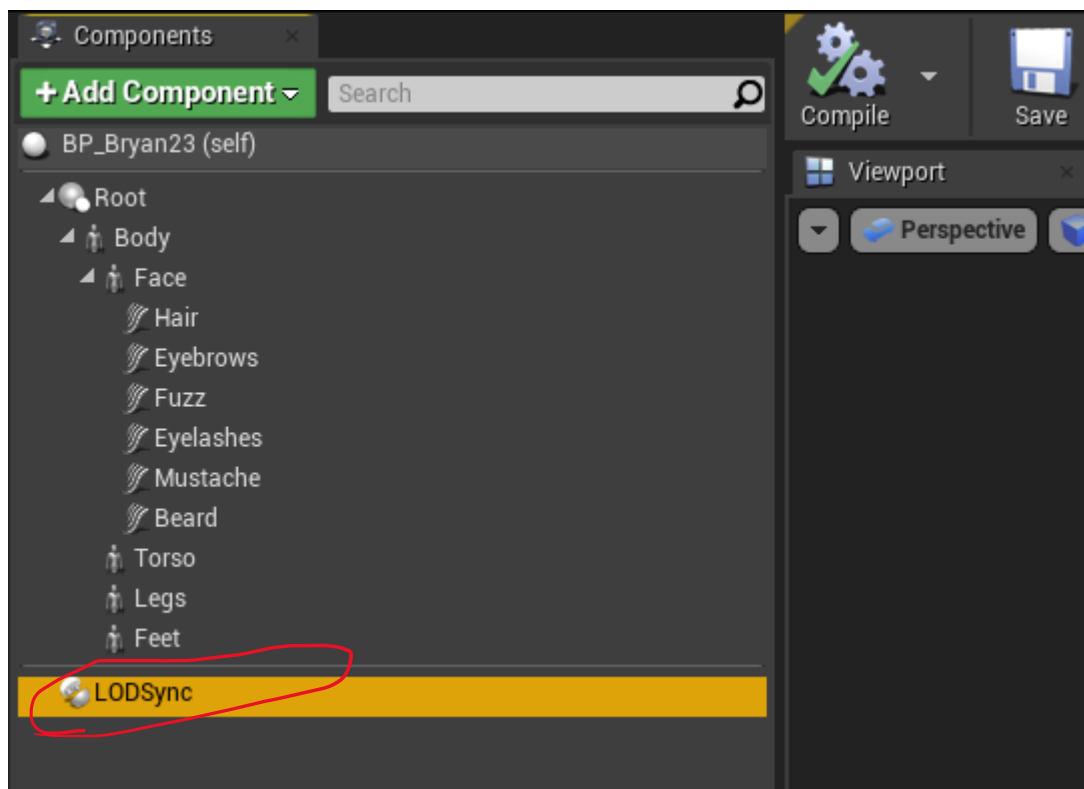


Рис.24. Ветка LODSync

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.

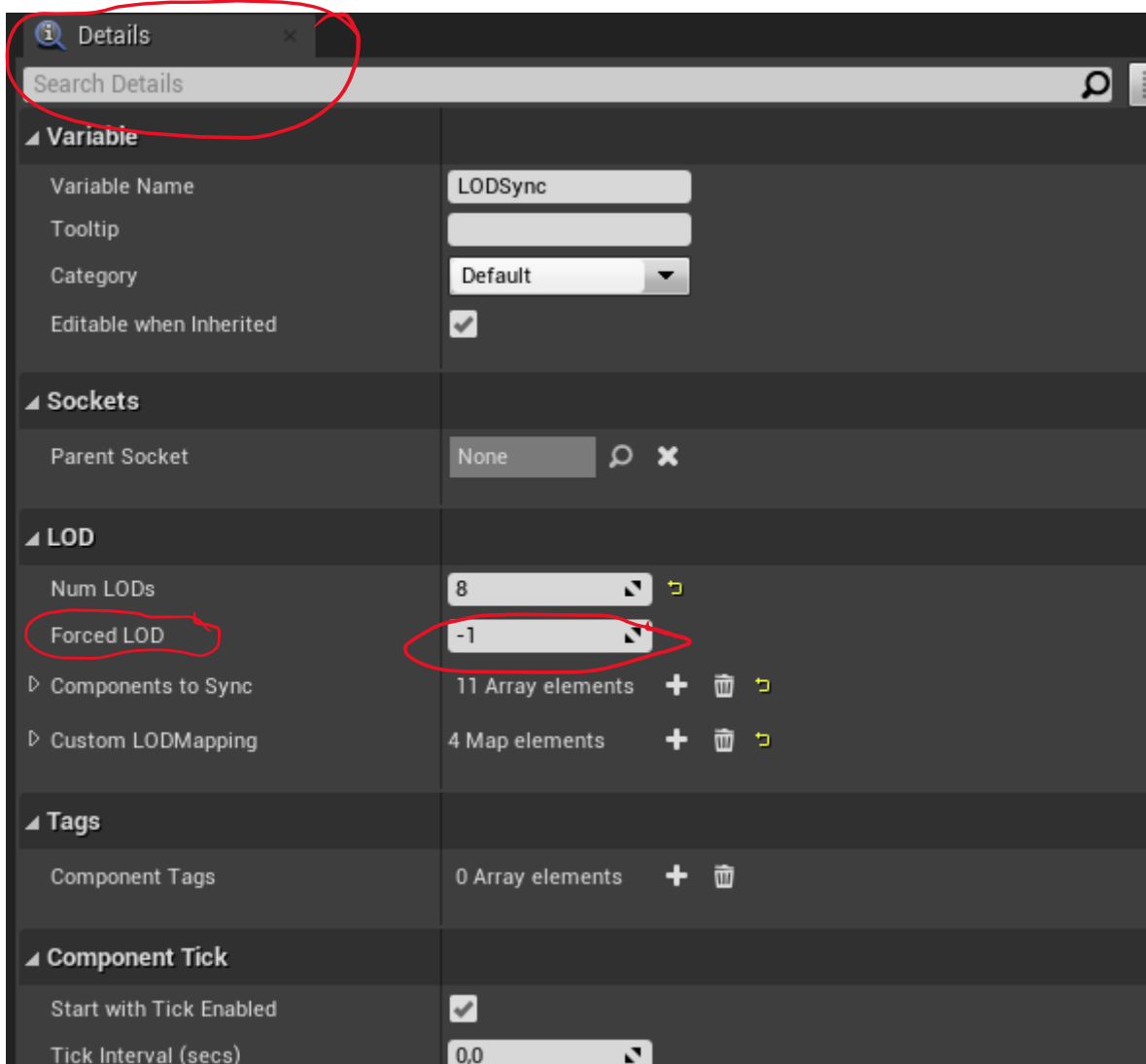


Рис.25. Окно Details

## 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 MNC под иконкой персонажа в Quixel Bridge.

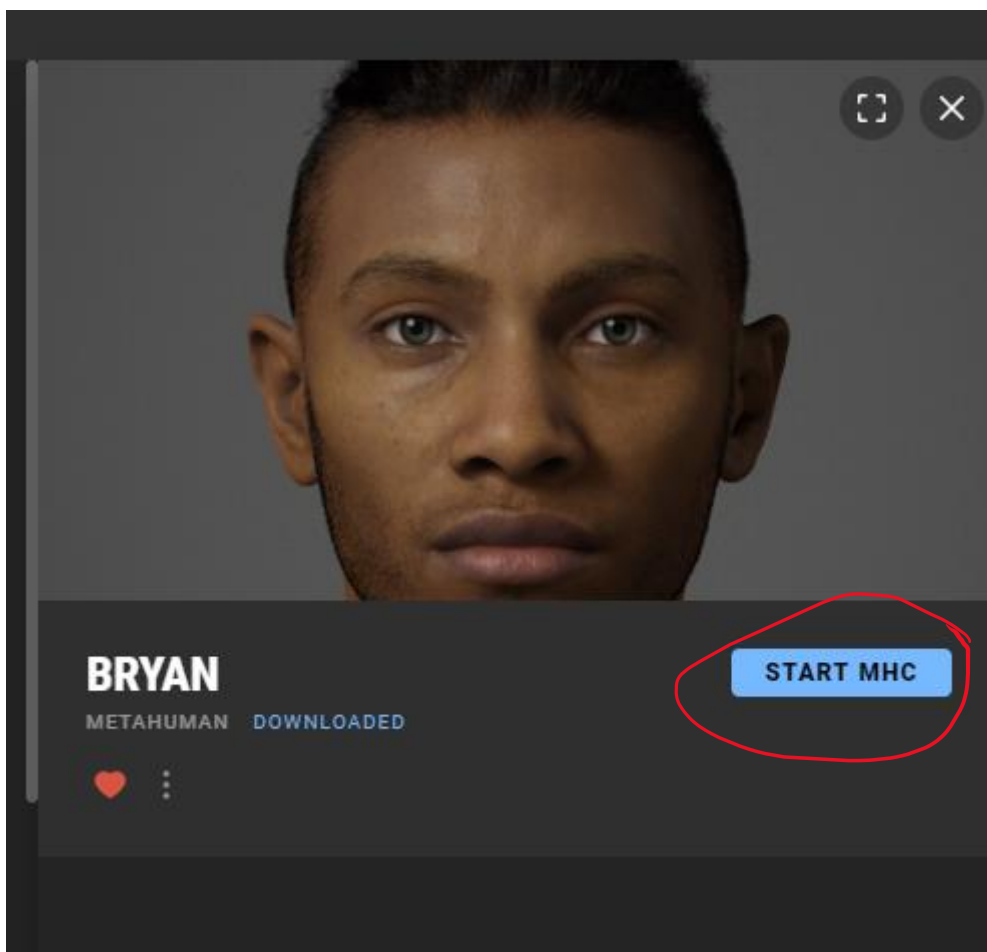


Рис.26. Кнопка Start MHC

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.



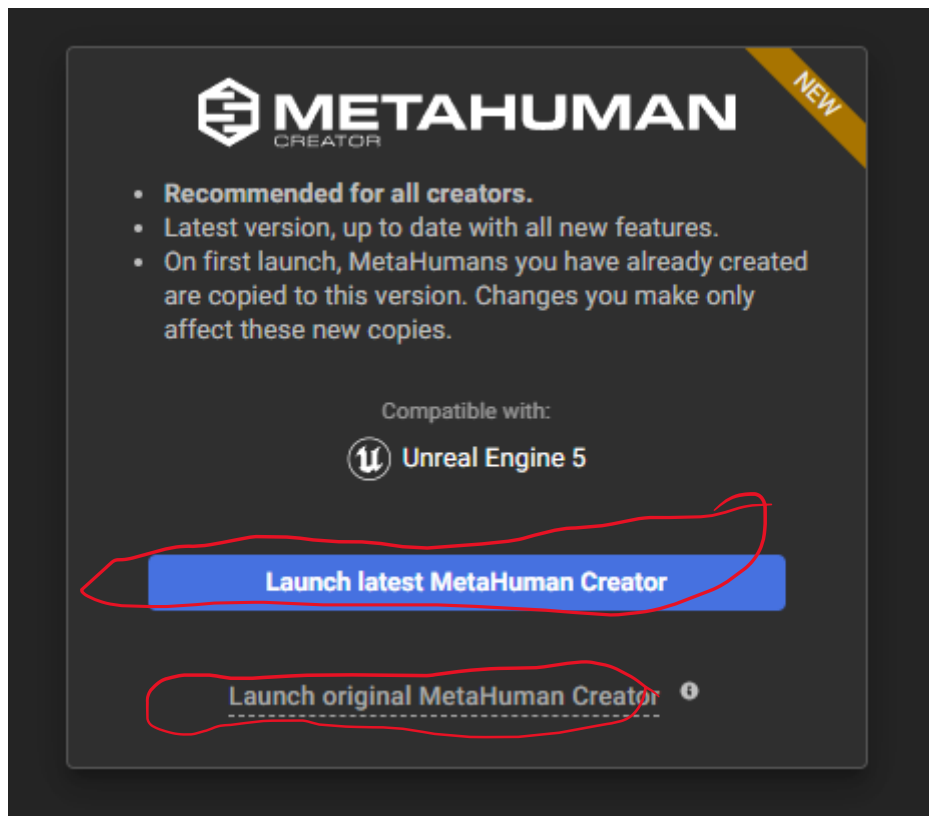


Рис.27. Выбор версии редактора МНС

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:

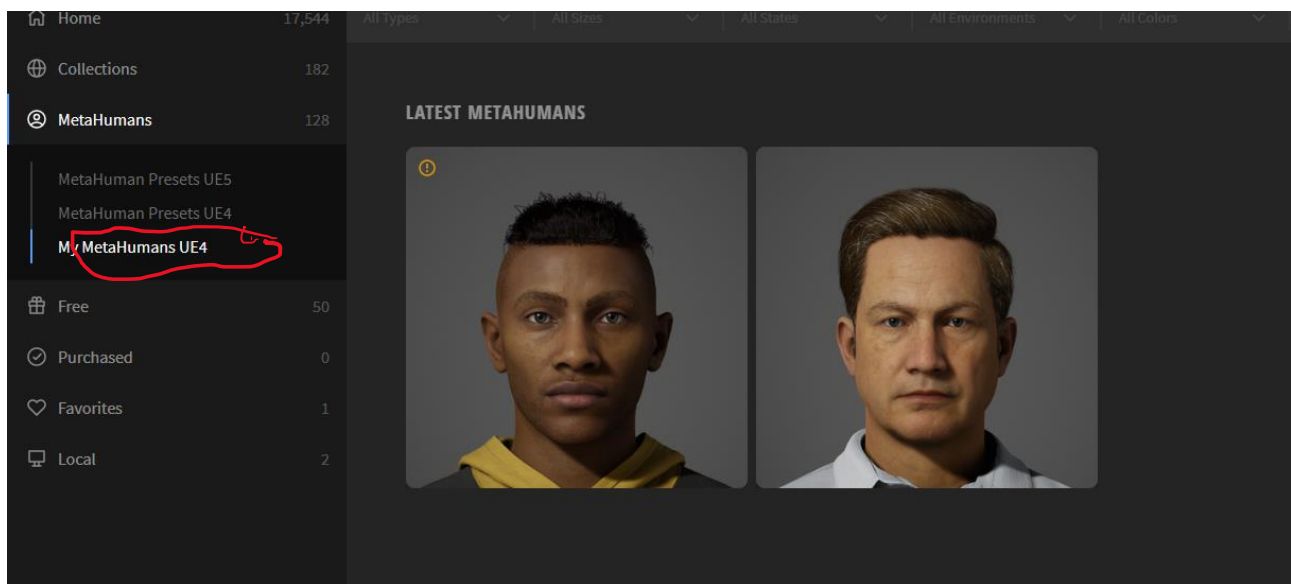


Рис.28. Меню персонажей My MetaHumans в Quixel Bridge

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.

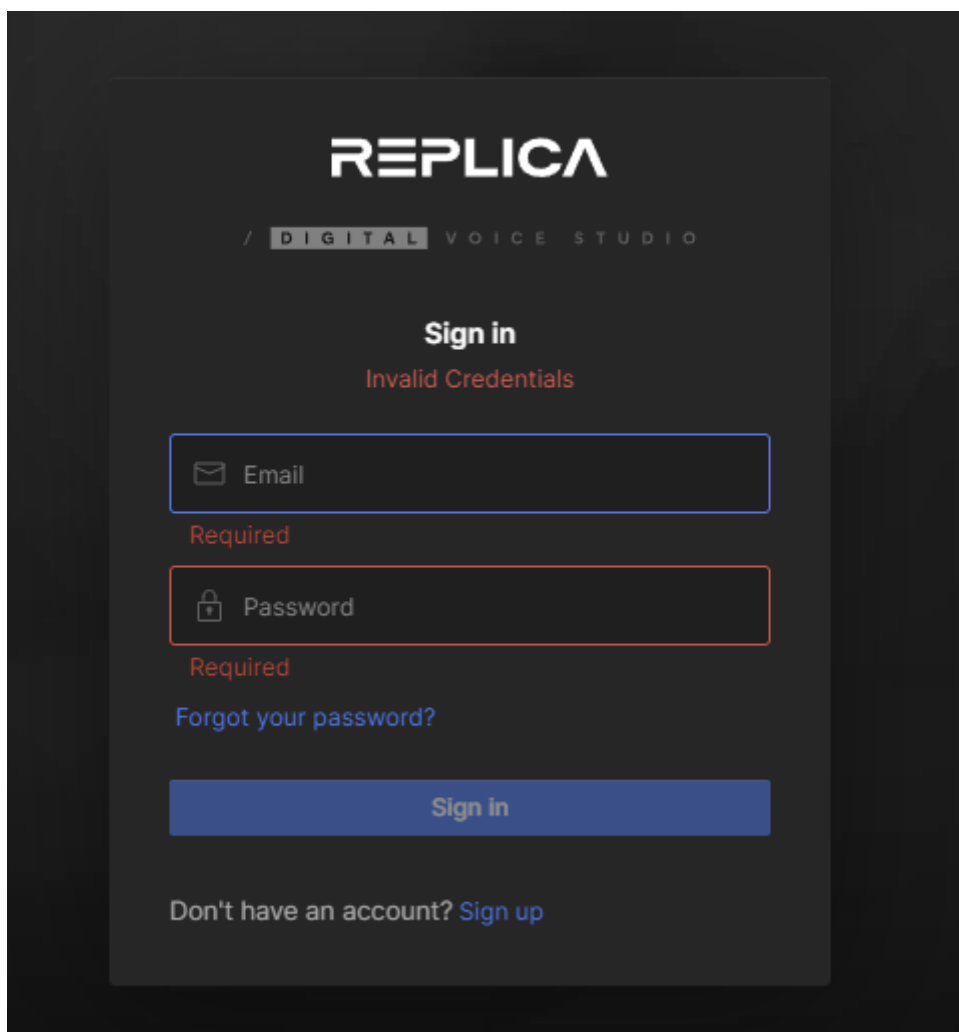


Рис.29. Окно входа в Replica

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

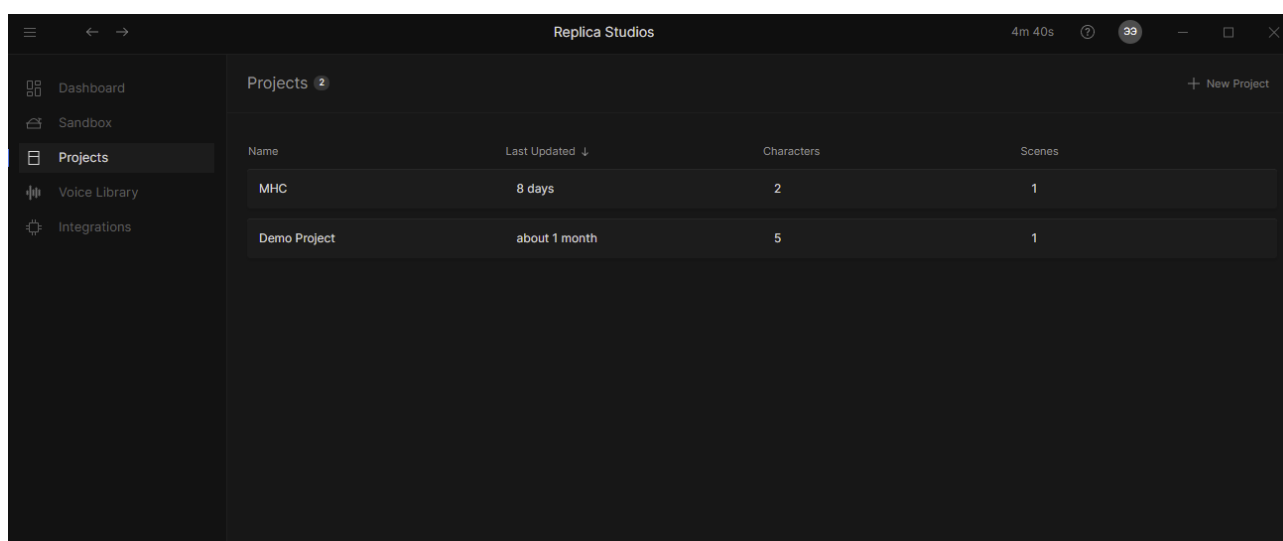


Рис.30. Меню Projects

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.

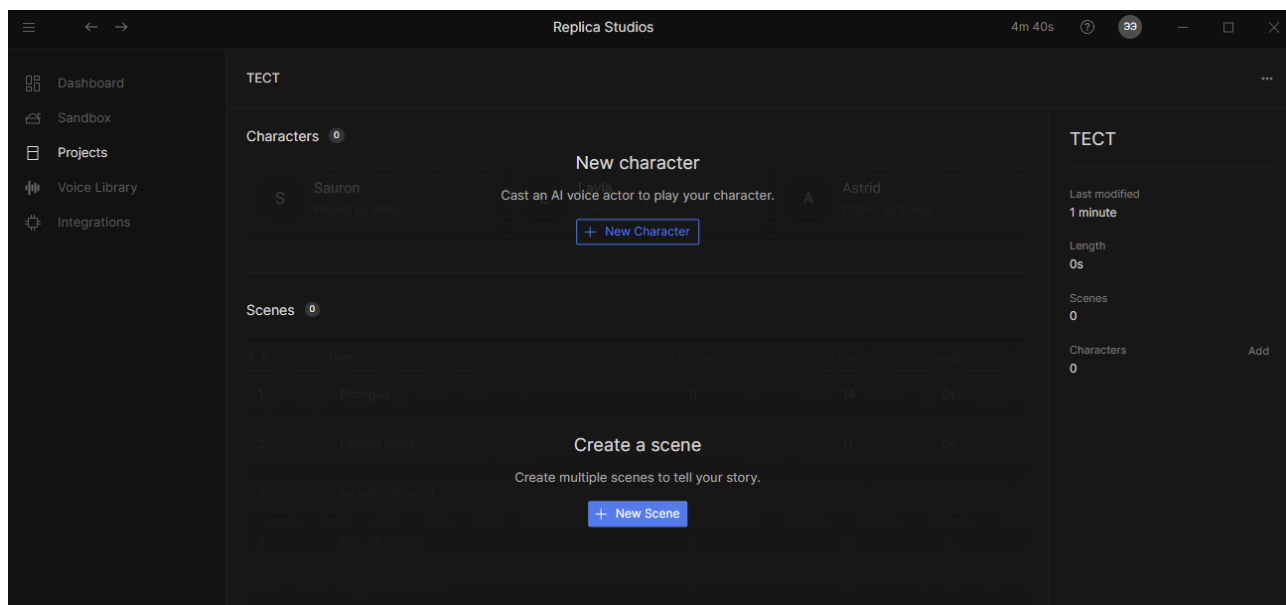


Рис.31. Меню редактора проекта

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

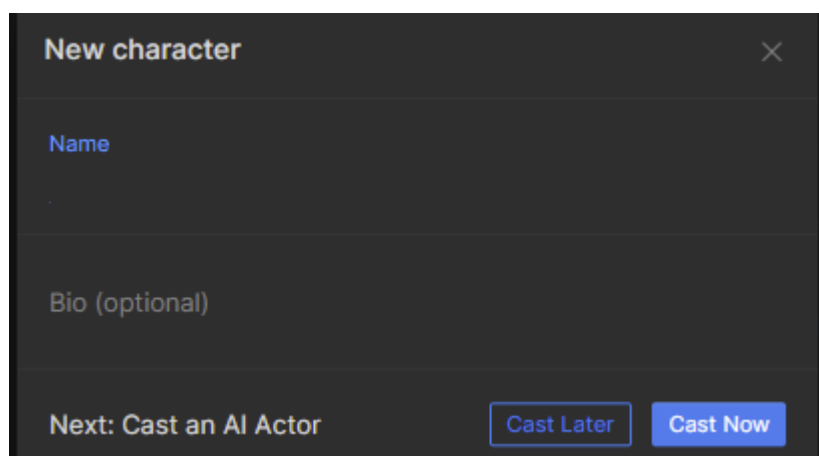


Рис.32. Меню описания персонажа

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.

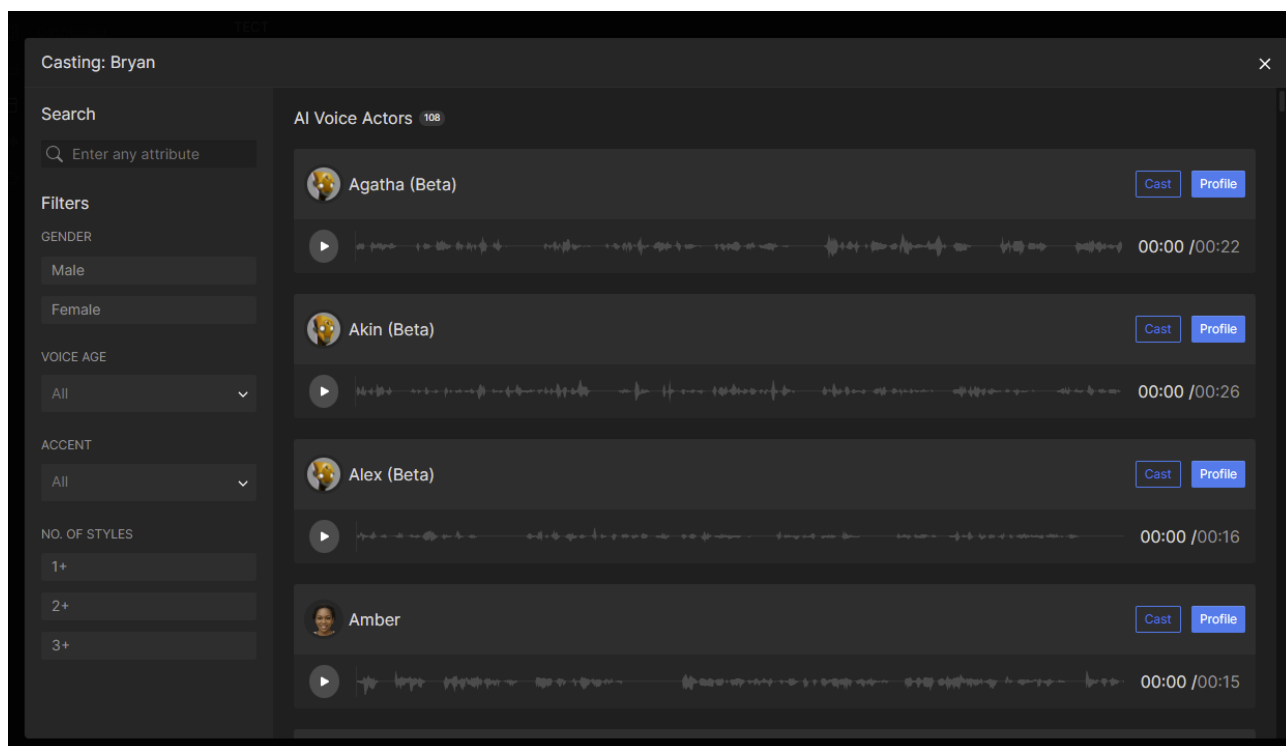


Рис.33. Список доступных голосов

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.

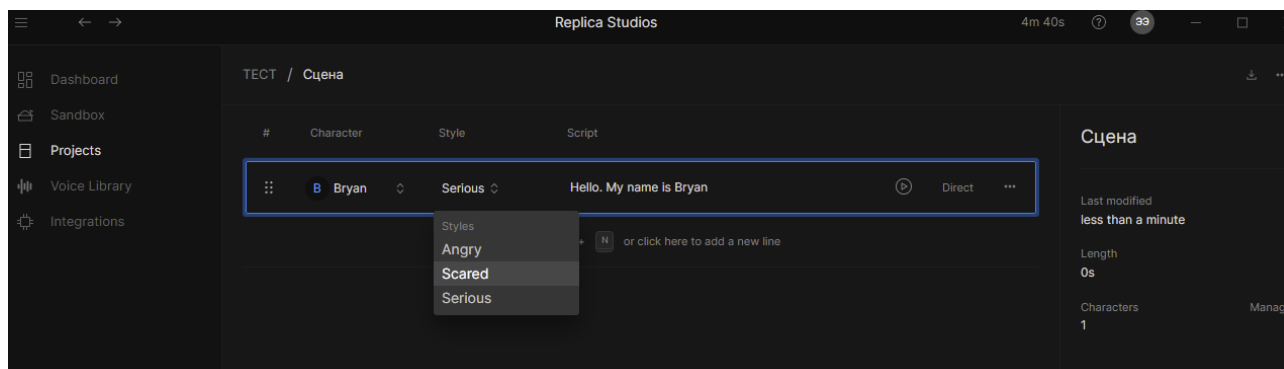


Рис.34. Меню озвучки персонажа

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

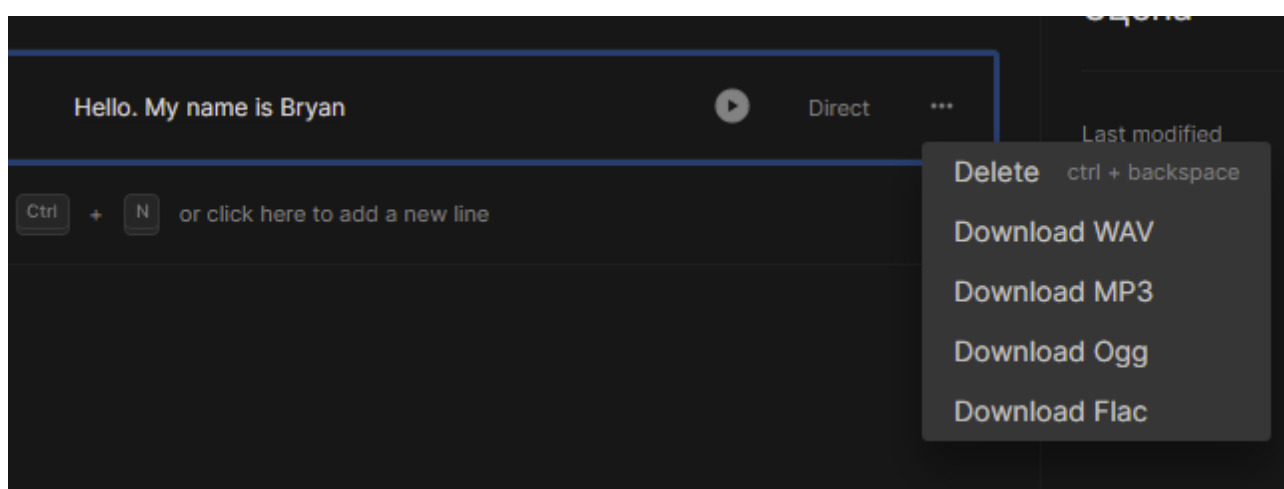


Рис.35. Варианты форматов аудиозаписи

Скачанную аудиозапись можно будет перенести в сам Editor

## 6.2 Lip sync персонажа с помощью MetahumanSDK

Для того чтобы осуществить синхронизацию речи персонажа с движениями его губ (Lip sync) необходим плагин MetaHumanSDK, который автоматический 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.

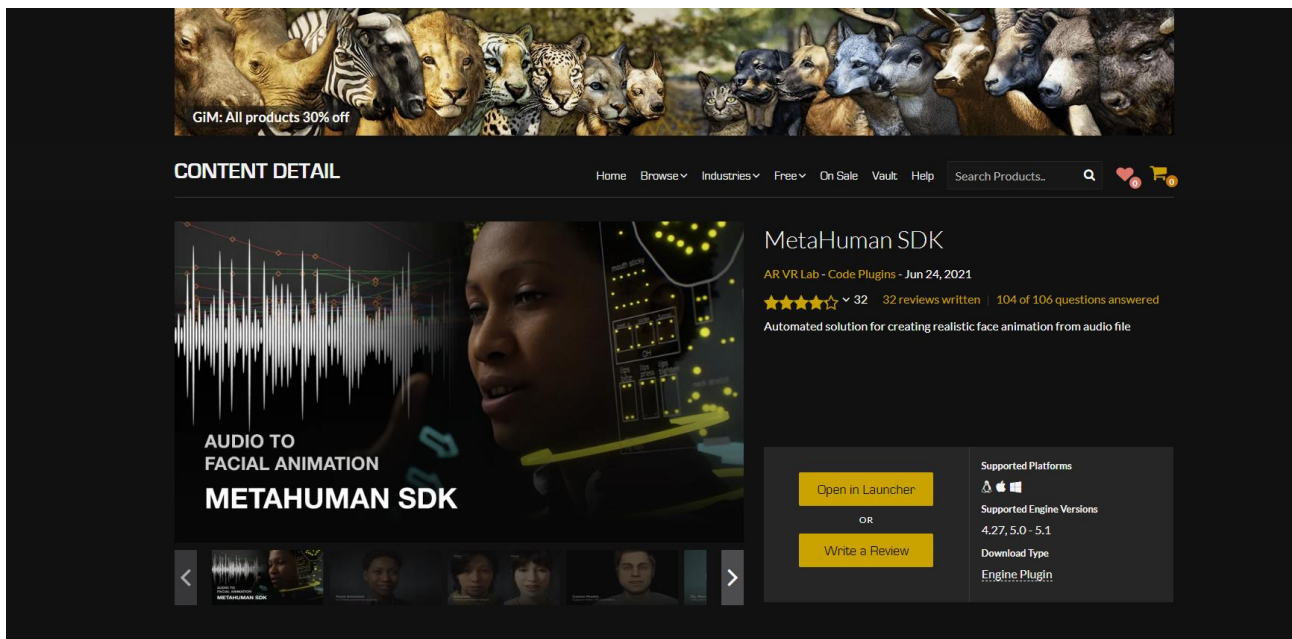


Рис.36. Страница плагина в Epic Games Store

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

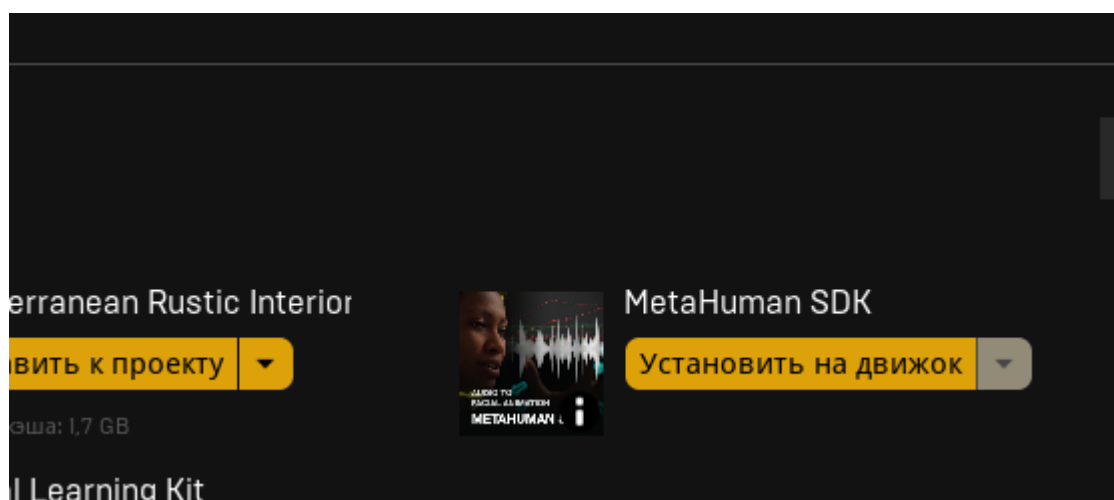


Рис.37. Плагин MetaHuman SDK в библиотеке.

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

### 6.3 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.



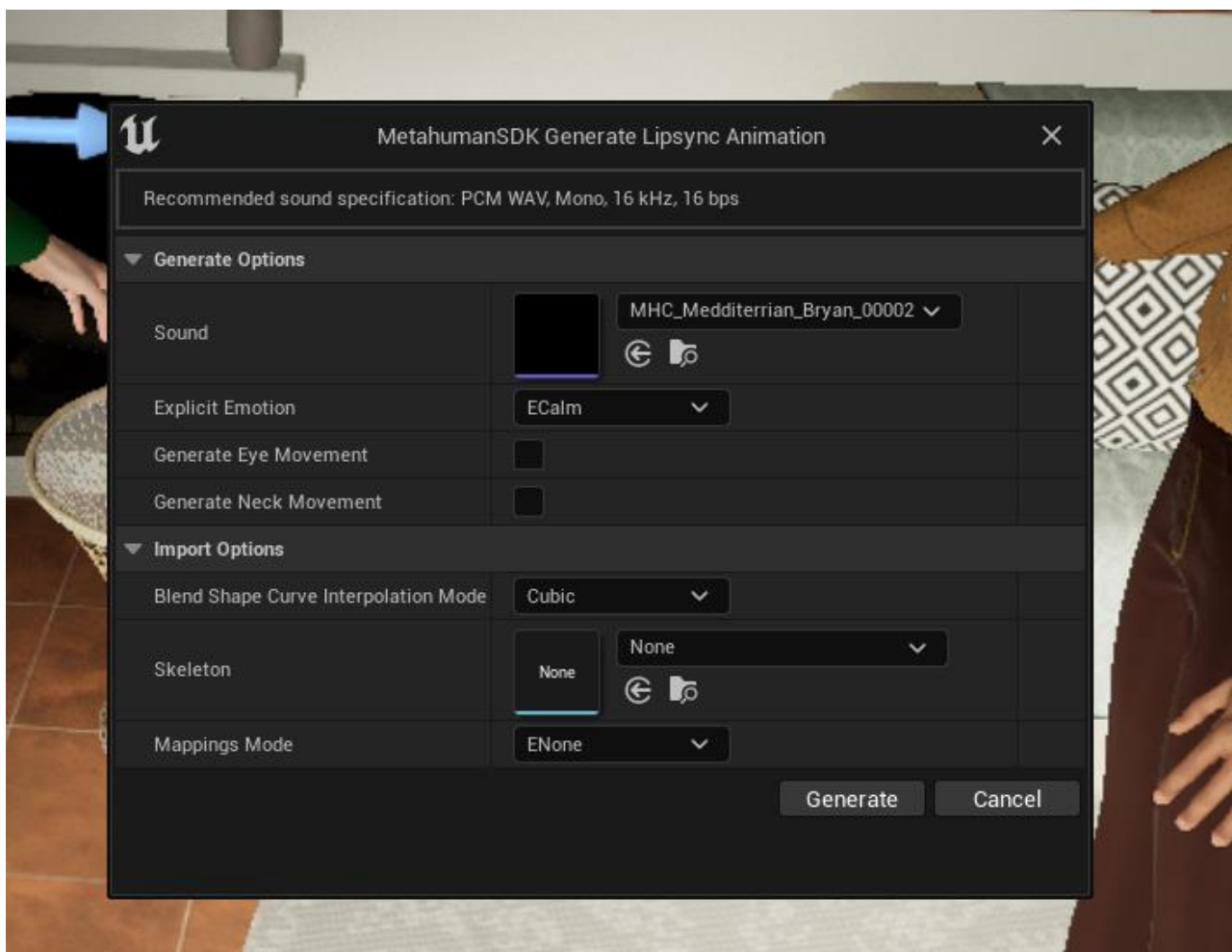


Рис.38. Меню настроек анимации.

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.

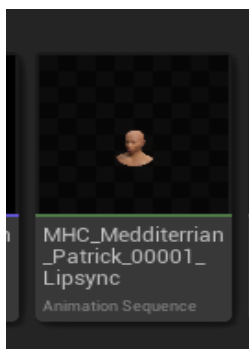


Рис.39. Созданная анимация.

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.

## 6.4 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 BP. В данном случае это BP\_Bryan.

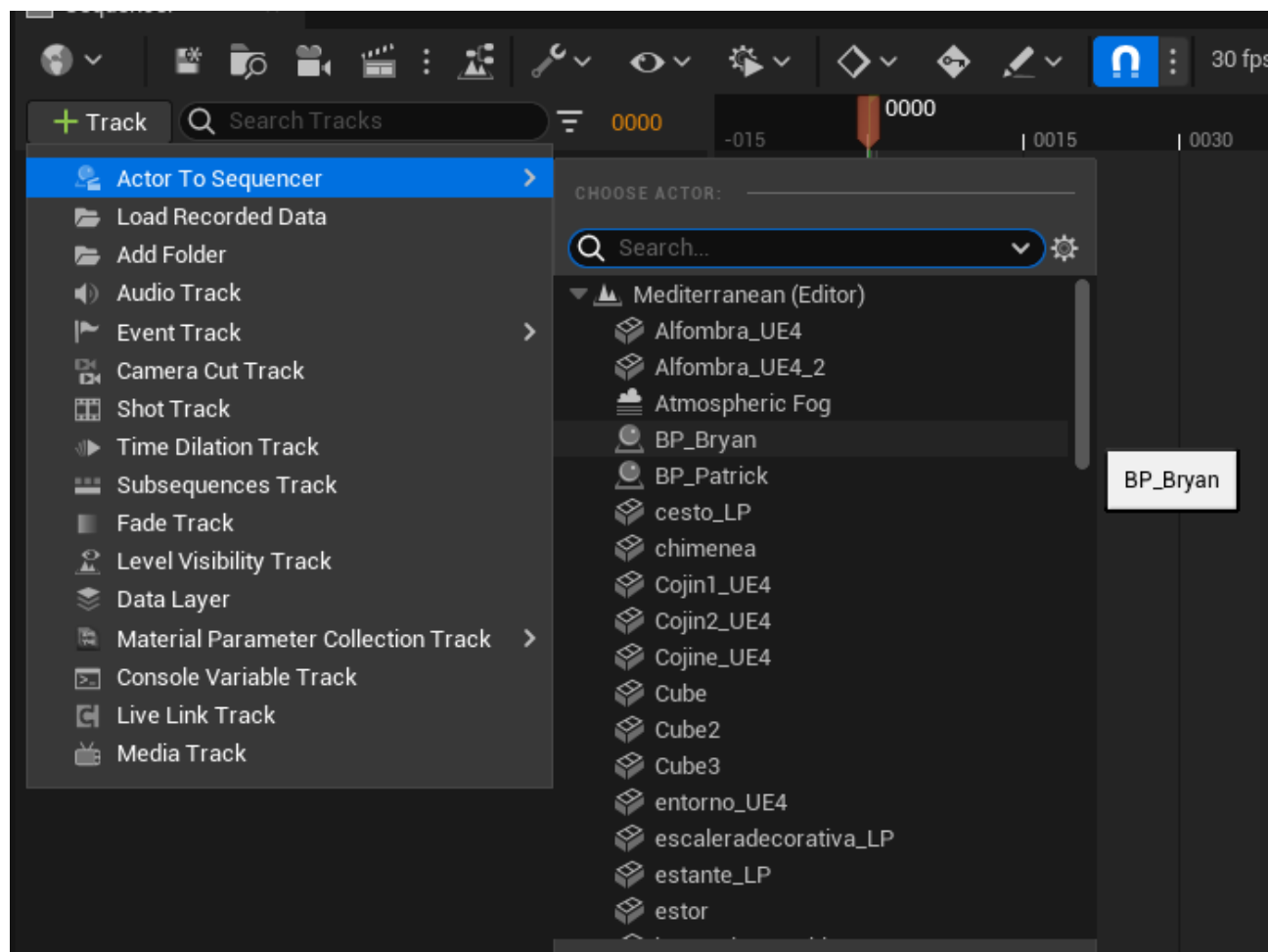


Рис.40. Добавление нового BP актёра.

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

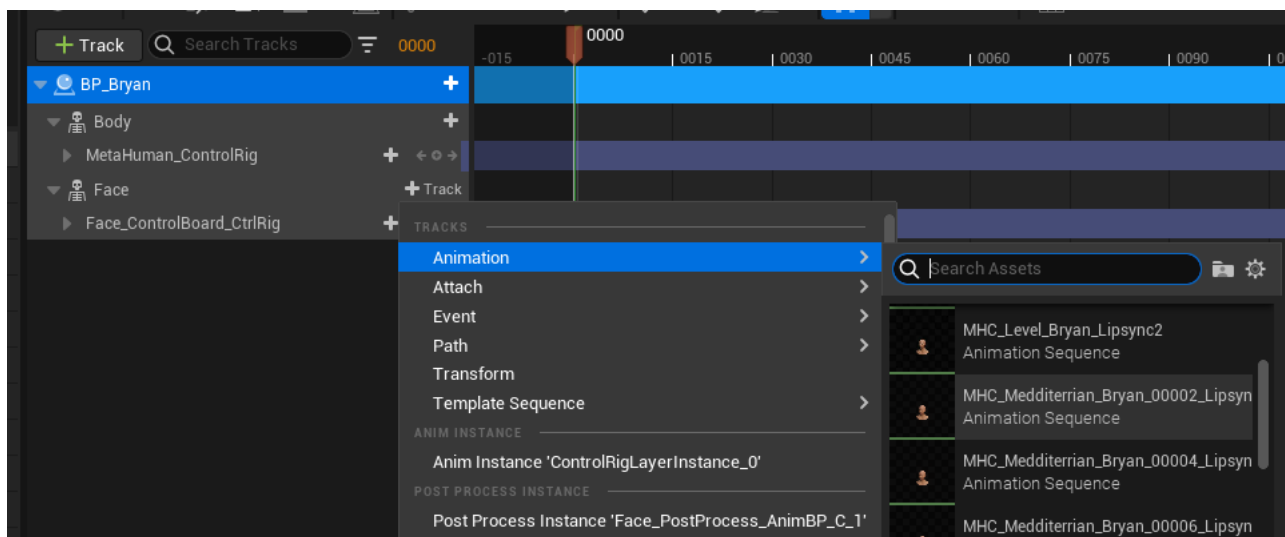


Рис.41. Добавление анимации для персонажа.

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.

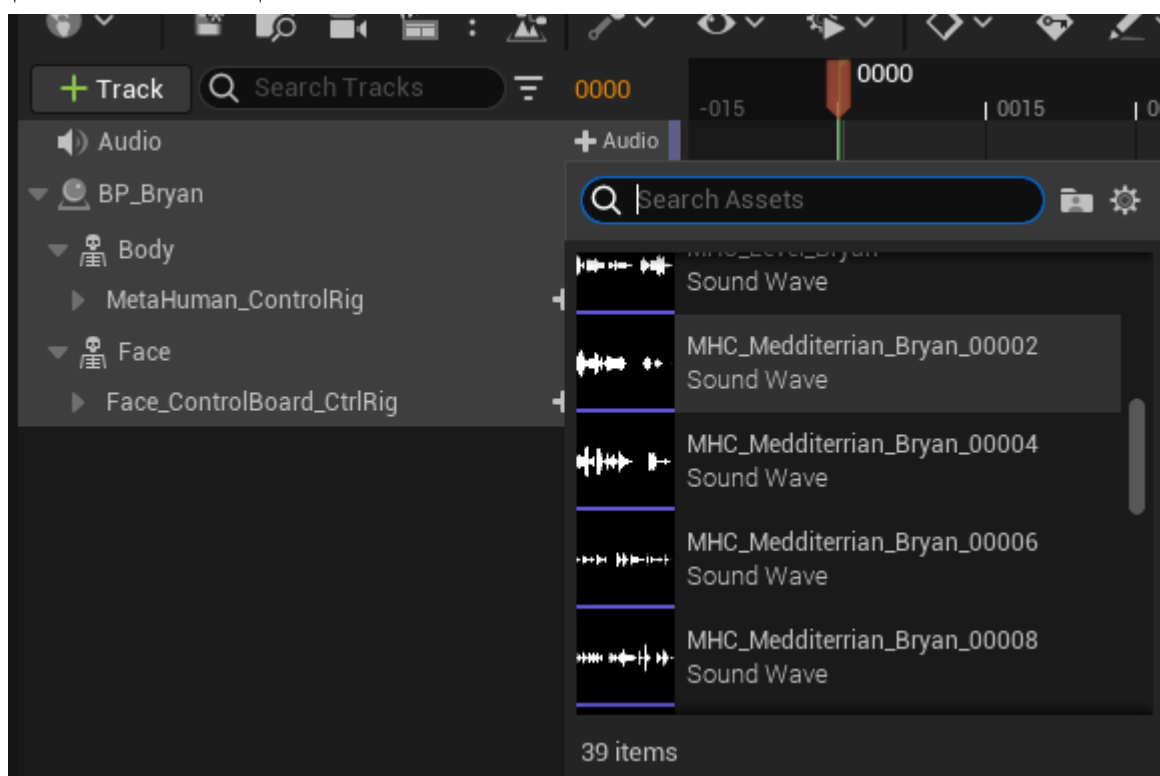


Рис.42. Добавление аудиозаписи для персонажа.

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.

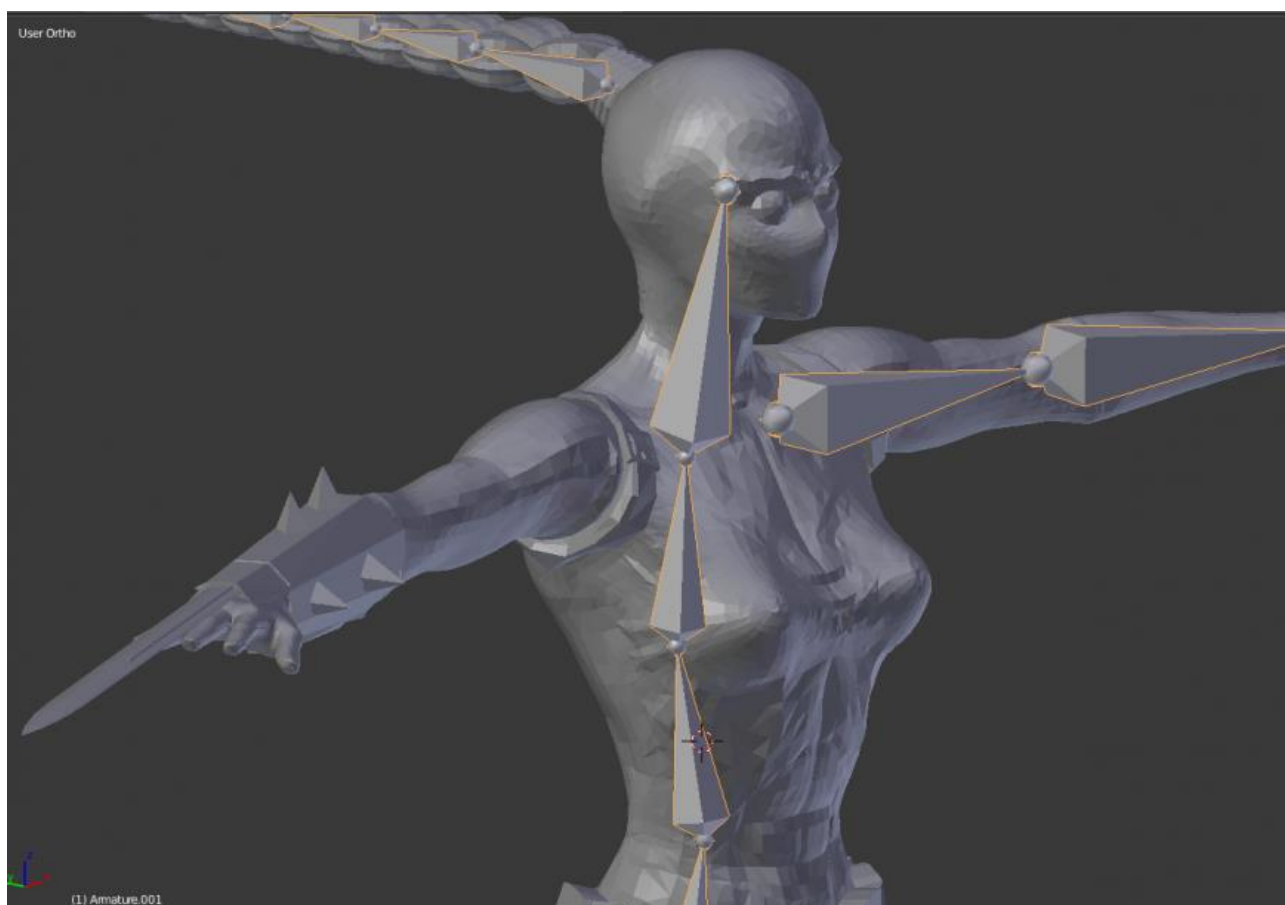


Рис.43. Пример создания скелета и настройки расположения костей модели.

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.



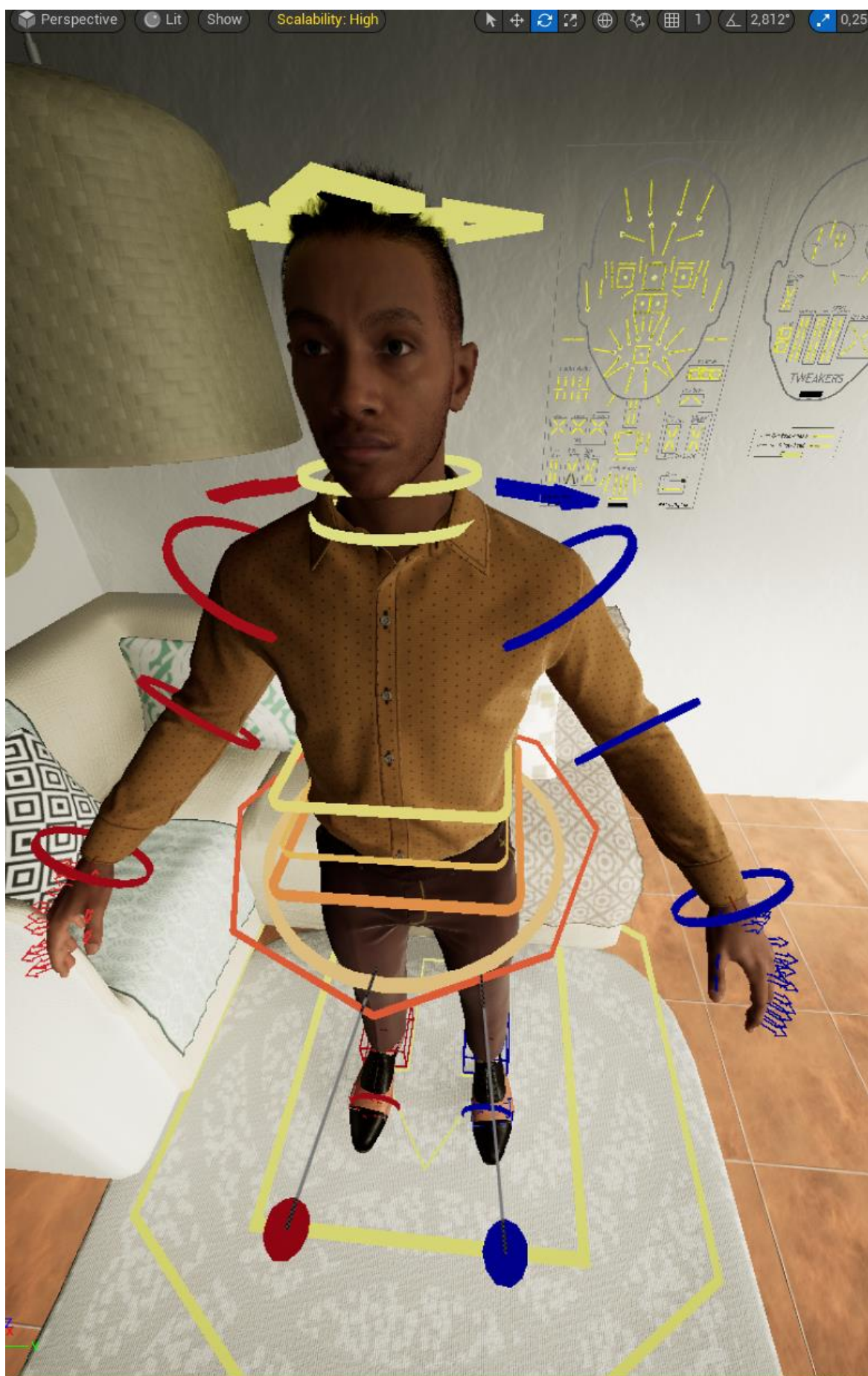


Рис.44. Расположение костей персонажа.

## 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.

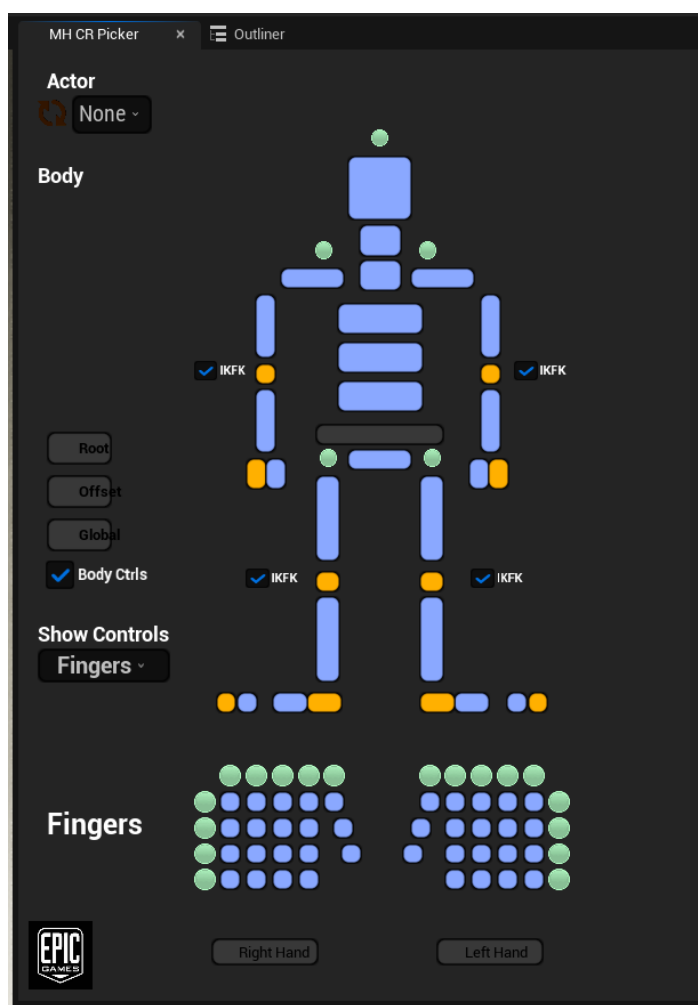


Рис.45. Панель управления костями персонажа.

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.

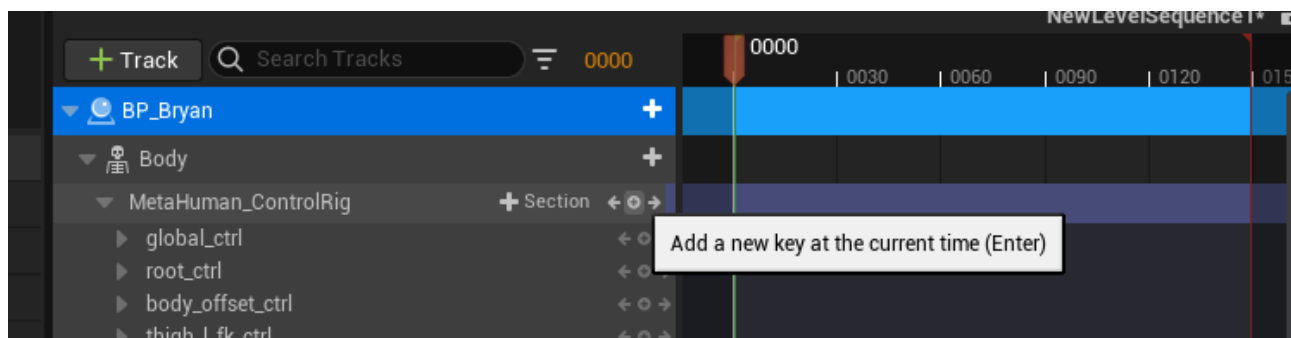


Рис.46. Сохранение нового положения персонажа.

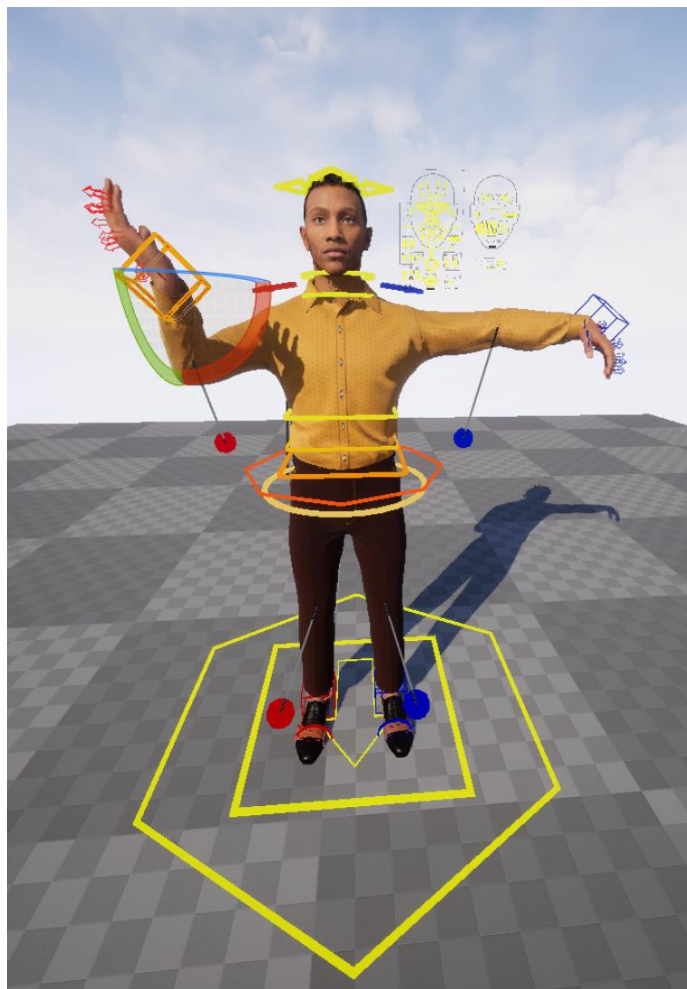
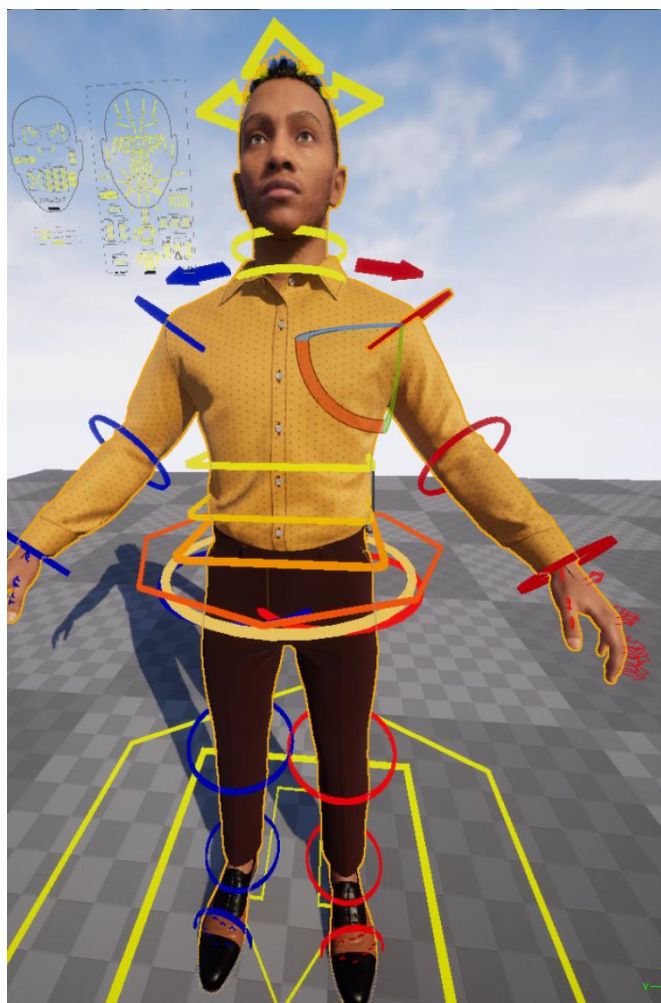


Рис.47. Положение персонажа до и после



## 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.

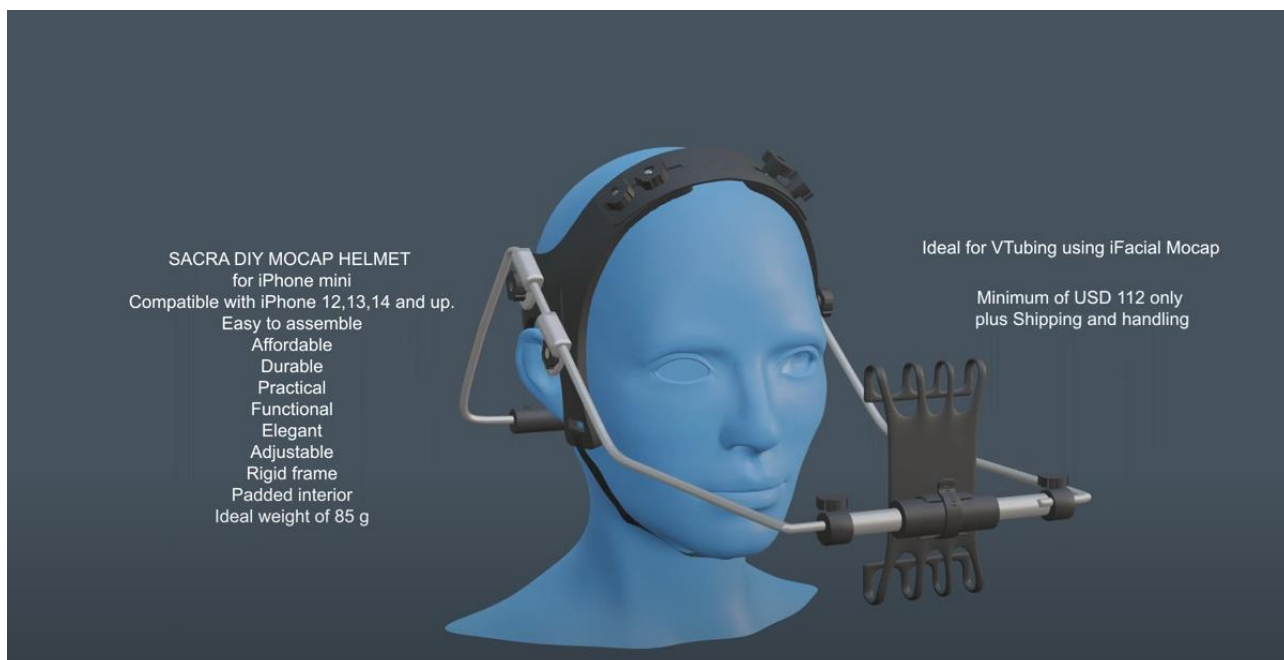


Рис.48. Самодельный шлем для считывания мимики

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. Эти датчики как раз и называются маркерами. Движение маркеров считывается камерой и уже в виде скелета переносится на персонажа.



Рис.49. Считывание мимики с актёра с применением маркерного метода.

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.

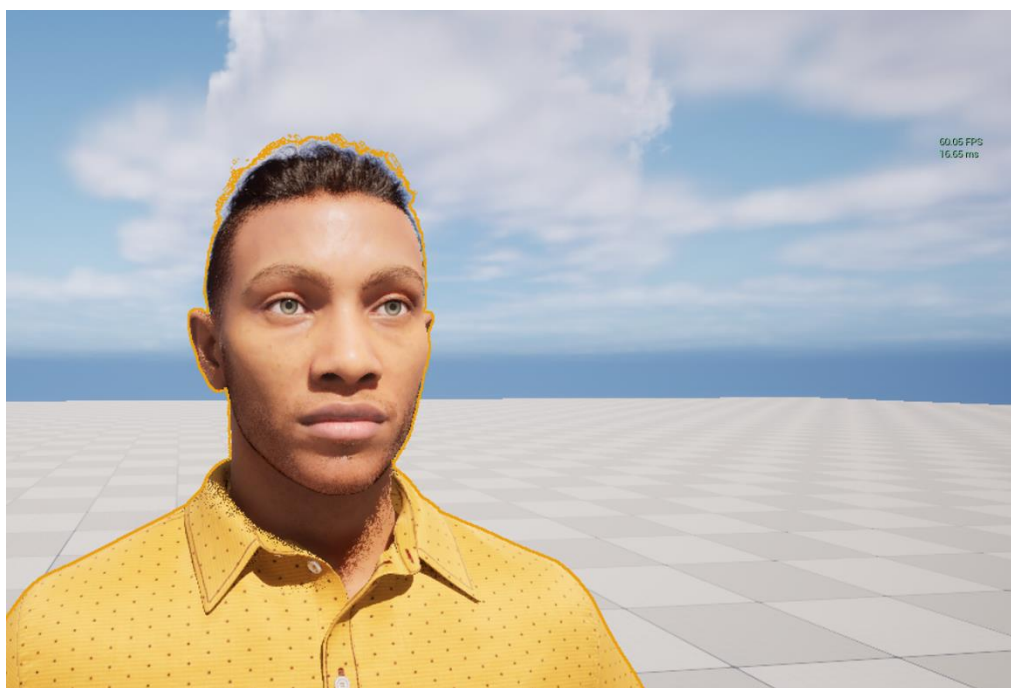


Рис.50. Модель Bryan уровня Low. FPS = 65



Рис.51. Модель Вryan уровня High. FPS = 49



Рис.51. Модель Вryan 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.

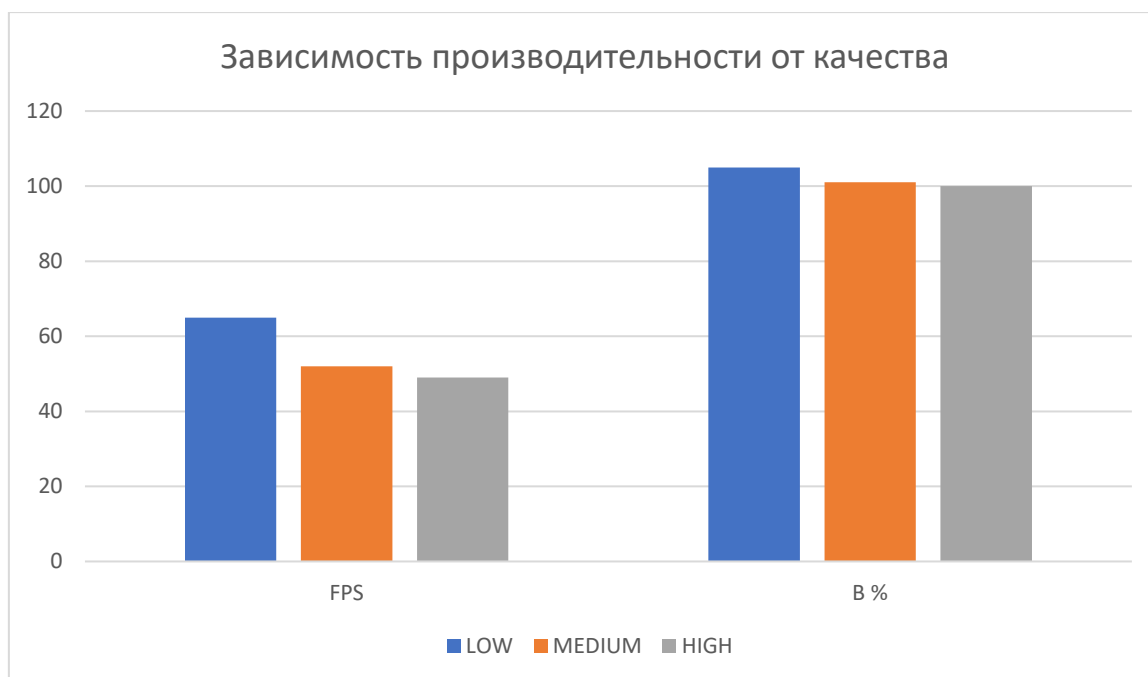


Диаграмма 1. Зависимость производительности от качества

## ЗАКЛЮЧЕНИЕ

Итак, в результате данного курсового проекта были рассмотрены инструменты, которые позволяют создать 3D-персонажей без навыков использования сложного ПО и без лишних затрат на оборудование и людские ресурсы, получены навыки работы в Unreal Engine и Quixel Bridge.

В результате работы были выполнены следующие задачи:

- 1) Создан проект Meta Human Creator с в Unreal Engine 4.
- 2) Подобраны необходимые инструменты для считывания мимики с лица человека.
- 3) Разобраны плюсы и минусы двух программ для считывания мимики с лица: Faceware Studio и Live Link Face.
- 4) Перенесена 3D-модель из Quixel Bridge в Unreal Engine 4.
- 5) Перенесена мимика на Mesh персонажа
- 6) Персонаж был помещён в интерьер вместе с новым персонажем Patrick.
- 7) Был осуществлён риггинг персонажей в результате чего они сидят.
- 8) Была добавлена и наложена Lipsync анимация для обоих персонажей вместе с озвучкой.

## 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 обращения: 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\\_litsevuyu\\_animatsiyu\\_v\\_igrah-356274](https://www.playground.ru/misc/news/kak_sozdayut_litsevuyu_animatsiyu_v_igrah-356274) Дата обращения: 1.10.2022;
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