D3.4 - Background Agent Environment Creation Demonstration

Grant Agreement nr	856879
Project acronym	PRESENT
Project start date (duration)	September 1st 2019 (36 months)
Document due:	August 31st 2021
Actual delivery date	August 31st 2021
Leader	Framestore
Reply to	Richard.Ollosson@framestore.com
Document status	Submission Version

Project funded by H2020 from the European Commission

Project ref. no.	856879
Project acronym	PRESENT
Project full title	**P**hotoreal **RE**altime **S**entient **ENT**ity
Document name	Background Agent Environment Creation Demonstration
Security (distribution level)	Public
Contractual date of delivery	31/08/2021
Actual date of delivery	30/08/2021
Deliverable name	D3.4 - Background Agent Environment Creation Demonstration
Type	Demo
Status & version	Submission Version
Number of pages	24
WP / Task responsible	Framestore
Other contributors	all partners
Author(s)	Theo Jones, Richard Ollosson
EC Project Officer	Ms. Diana MJASCHKOVA-PASCUAL Diana.MJASCHKOVA-PASCUAL@ec.europa.eu
Abstract	This deliverable focuses on the methodology employed to create the background agents for Present and to add support for those agents into the shared project structure or reference implementation. The document also presents the case for a pivot in focus away from the compositing of realtime content into offline content that was included in the original proposal. An alternative approach is presented that involves including all content in a single interactive scene.
Keywords	digital human, real-time, Unreal Engine
Sent to peer reviewer	Yes
Peer review completed	Yes
Circulated to partners	No
Read by partners	No
Mgt. Board approval	No

Document History

Version and date	**Reason for Change**
1.0 15/07/2021	Document created by Theo Jones
1.1 15/08/2021	Version for review including inputs from other partners
1.2 27/08/2021	Final version for submission

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

This deliverable focuses on the methodology employed to create the background agents for Present and to add support for those agents into the shared project structure or reference implementation. Thanks to Cubic Motions incorporation as part of Epic Games the Unreal MetaHuman architecture has been targeted as the approach for building background characters in Present. This offers enormous advantages to the project in terms of usability, diversity and scalability and these advantages are outlined and discussed.

This deliverable also presents the case for a pivot in focus away from the compositing of realtime content into offline content that was included in the original proposal. An alternative approach is presented that involves including all content in a single interactive scene. This approach holds a number of key advantages, which are presented in detail.

A number of demonstrations are also included to show the feasibility of this alternative approach, as well as an example in which a background MetaHuman agent is composited into a live action feed via the Brainstorm InfinitySet software.

Finally a number of 'next steps' are outlined that show the road map for expanding the reference implementation and background agent integration to include crowd work and other multi agent scenarios. This work is already underway and will be presented in future deliverables.

2. BACKGROUND AGENT CREATION

2.1 INTRODUCTION

The need for multiple agent resolutions was identified early in the project. A hero hi-res agent that targets the best possible visual quality along with a full range of emotional expression is required for one-on-one user interaction in order to facilitate a high level of user engagement. However this resolution of agent and its associated computational cost is not suitable for all use cases.

Alongside the one-on-one user interaction there are a number of scenarios outlined in the project that require more than one agent or are restricted in their computational overhead, for example because they target VR hardware. For these scenarios a lower resolution agent is required that has a correspondingly lower computational cost.

At the early planning stage of the project Cubic Motion were identified as the partner best placed to deliver this lower resolution agent, while Framestore focussed on the highest fidelity version. This choice has allowed Cubic Motion to build the lower resolution agent on top of rigging technology, which has gone on to form the basis of Epic MetaHumans.

https://www.unrealengine.com/en-US/digital-humans

Adoption of the MetaHuman technology for the lower res agent has many benefits to the project. The open framework allows for partners to author original agents easily using well supported and widely used tools, whilst the wide spectrum of ethnicities, ages and genders available within MetaHumans helps to greatly increase diversity options, a key concern of the original single actor focus.

The focus for Framestore has therefore shifted from creating the lower resolution agent to providing support within the wider reference implementation for MetaHuman implementation. Alongside this Framestore has provided all the scanning, modelling

and texturing undertaken for the high resolution agent to Cubic Motion in order to allow them to create a MetaHuman build of our principal actor.

2.2 BACKGROUND AGENT CREATION METHODOLOGY

Cubic Motion utilised its position as a part of Epic Games to give the project pre-release access to the build and rigging technology behind Epic's MetaHumans. This has provided the Present consortium with a well supported and widely used methodology for new agent creation - the MetaHuman Creator application.

MetaHuman Creator is a free cloud base application that empowers anyone to create digital humans, complete with hair and clothing, in minutes. MetaHumans come fully rigged and ready to animate in Unreal Engine.

MetaHuman Creator is intuitive and easy to use. Users can simply select a starting point from the diverse range in the database, choose several more to contribute to the MetaHuman, and blend between them. Then, refine the character with sculpting tools and control guides, just by dragging on the asset.

With near-infinite variations of facial features and skin complexions, plus an array of different choices for hair, eyes, make-up, and teeth, you can create a huge variety of faces for your projects that cover a wide spectrum of ethnicities, gender and age. Pick the body type that is required for the character and dress them with different clothing sets in the tool.

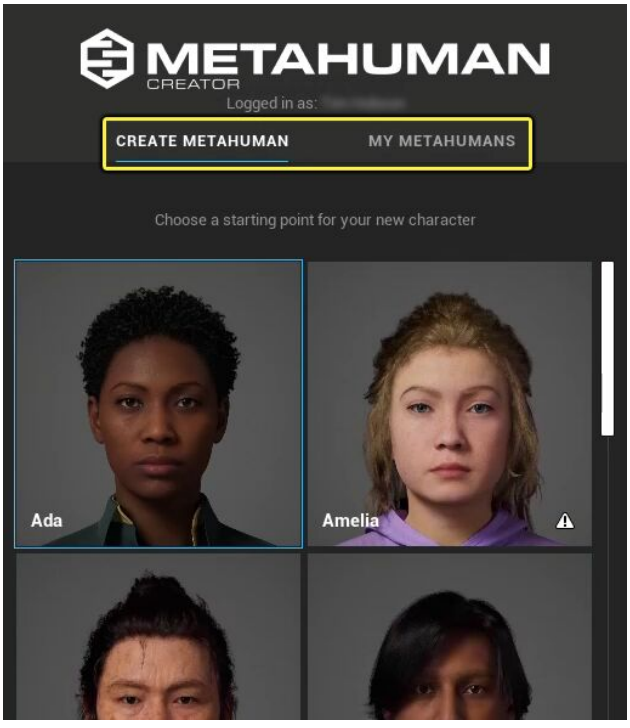
MetaHuman Creator derives its data from real-world scans and any adjustments are constrained to fit within the limits of the examples in its database, so it's simple to make physically plausible MetaHumans. Other factors, such as the carefully selected range of skin tones and hair colors, also help ensure accuracy.

MetaHumans can run in real time on high-end PCs with RTX graphics cards, even at their highest quality with strand-based hair and ray tracing enabled. Assets come with eight LODs, some of which employ hair cards, which makes it ideally suited for the scenarios required of the Present project's background agents.

The MetaHuman Creator provides a way to create new agent representations in an intuitive and easy-to-use interface. Creating the look of a digital human consists of editing broad and fine details that are divided into key attributes, such as the face, hair, eyes, and body.

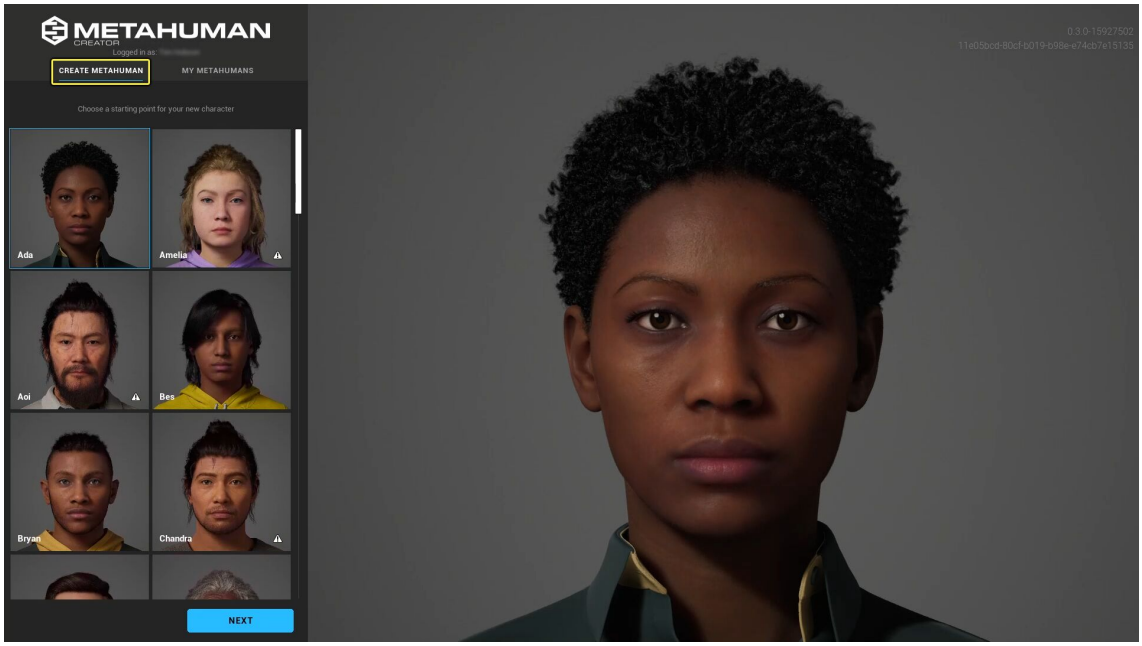
MetaHuman Presets Selection and Setup

When initially opening the MetaHuman Creator, the user is presented with two tabs: Create MetaHumans and My MetaHumans. Select a MetaHuman Preset to work from, or continue working on one previously selected and started modifying.



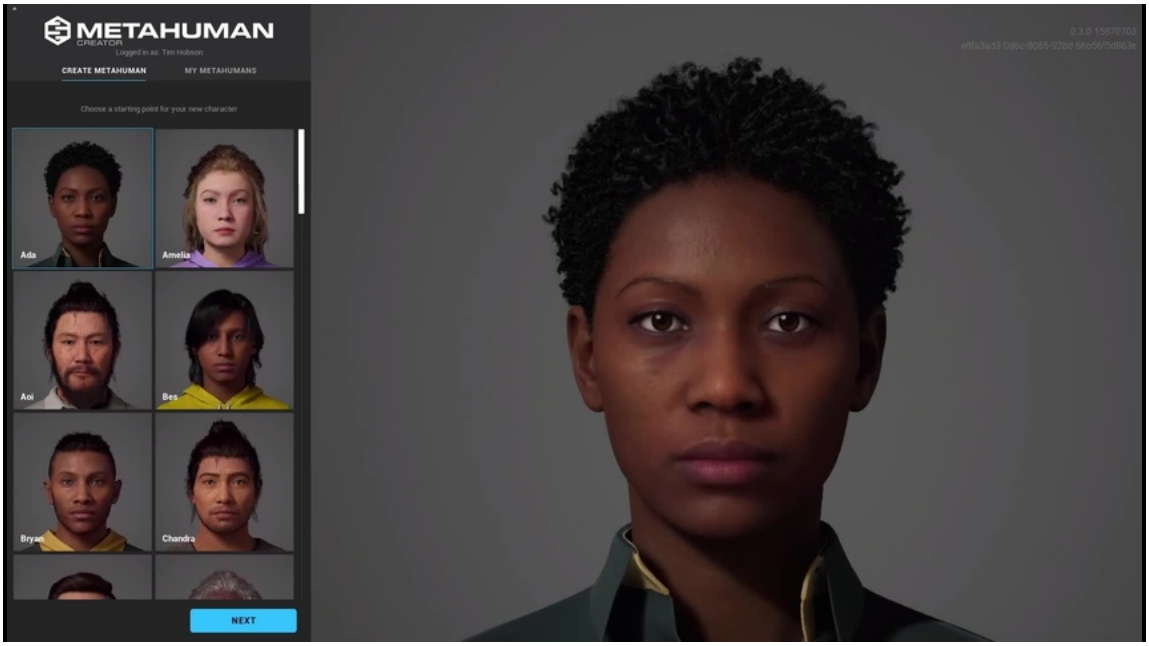
Choosing a MetaHuman Preset

The MetaHuman Creator provides a number of MetaHuman Presets as starting points to choose from. They include a diverse group of starting points for different body figures, genders, and ethnicities.



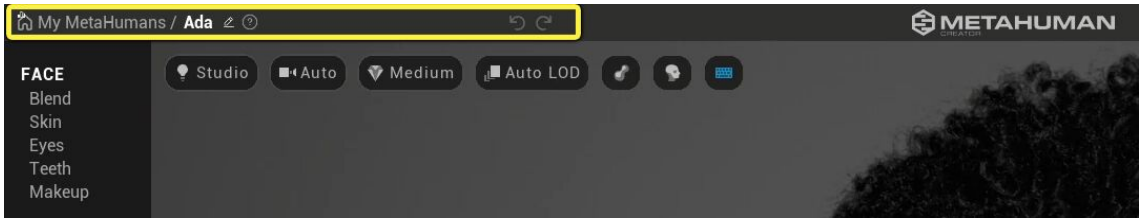
Once a character is selected the preview can be used to inspect that character using the following controls:

- Left-click and drag to rotate the character.
- Mouse wheel to zoom in and out.
- Middle mouse button to pan or rotate the camera around the character.



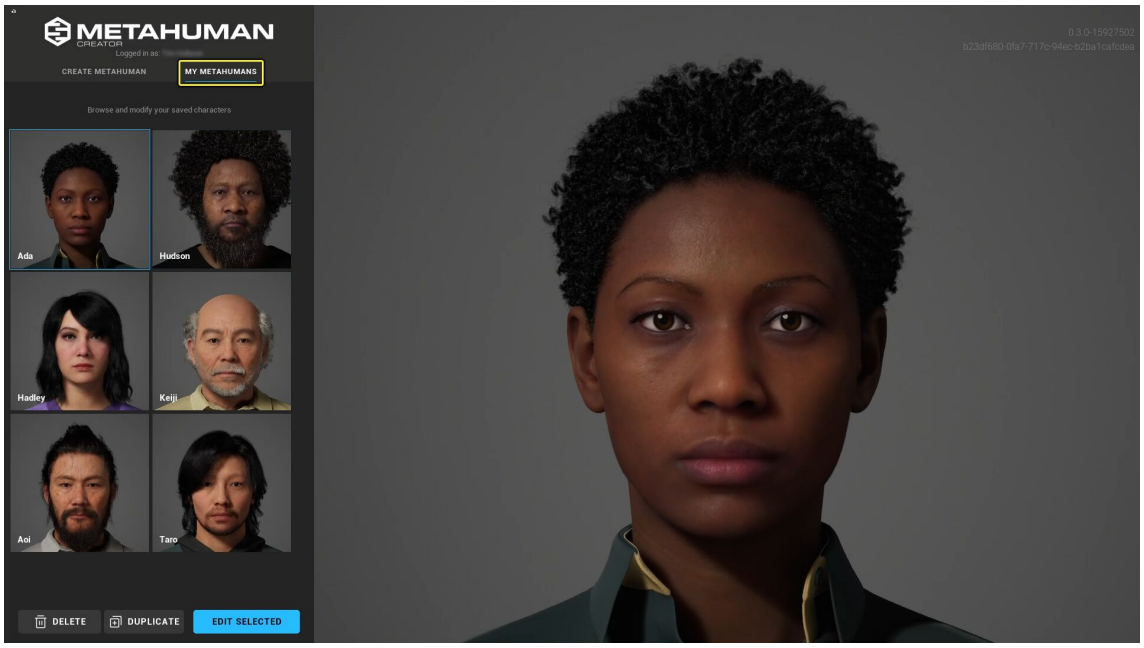
Once a selection is made, click the Next button to create an instance of this Preset to start editing. All edits will be automatically saved to the My MetaHumans gallery.

Presets will load in the creator tool where you can start editing and changing properties. The top-most toolbar is the MetaHuman Title Bar. Here you can go back to the home screen and select a different Preset, or choose one you've previously worked on. The name of the MetaHuman can be edited, which carries over to Quixel Bridge when you export to another application. You can also use the arrows to undo / redo an previous action.

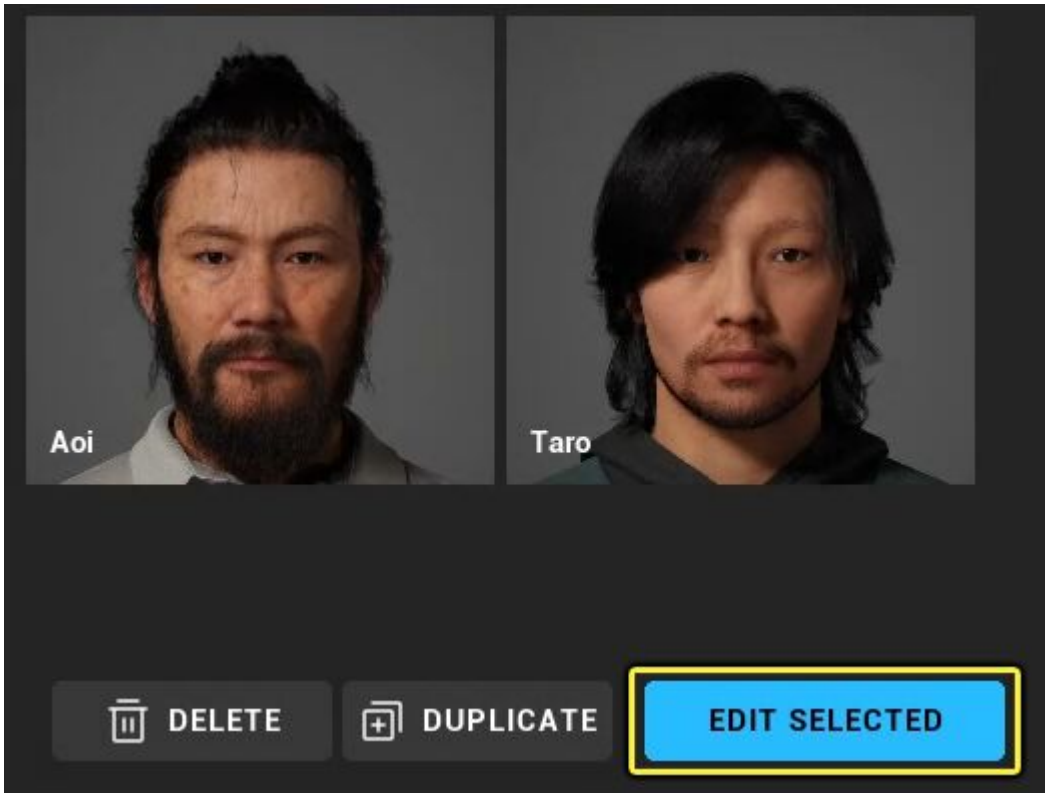


Modifying a Previously Created MetaHuman

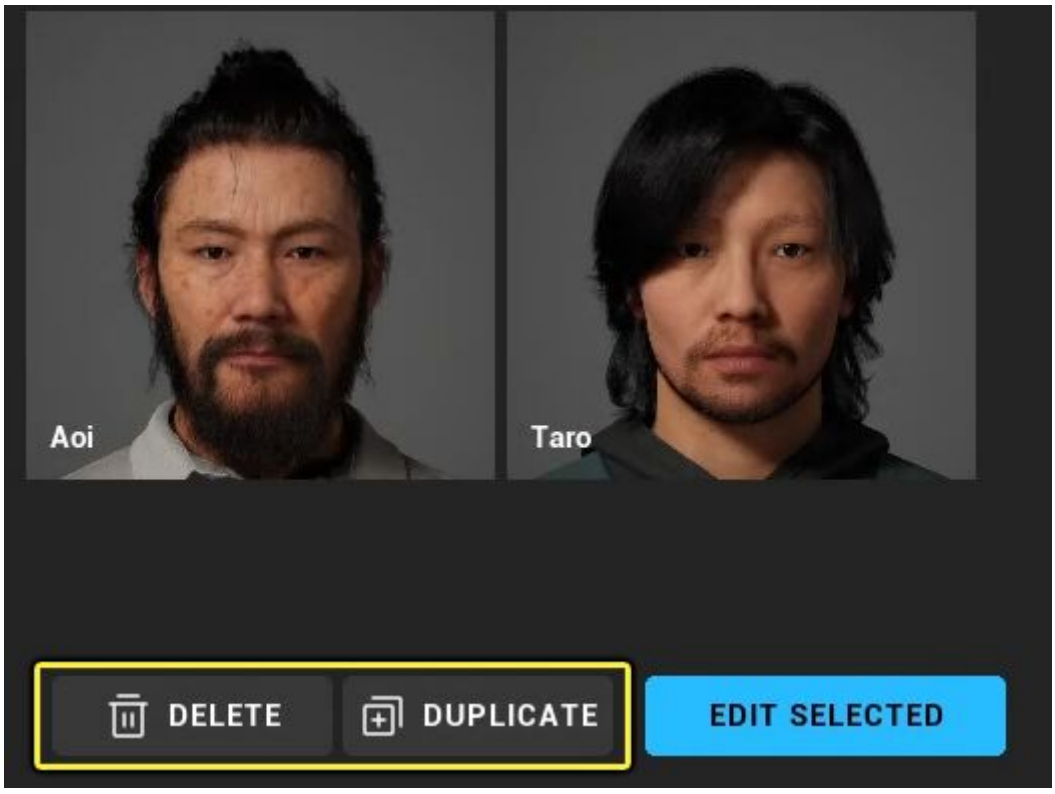
Any MetaHuman Presets that are edited automatically get saved to the My MetaHumans gallery. From there you can choose to edit, duplicate, or delete them from the creator.



When you want to continue work on a MetaHuman in your gallery, select them and choose Edit Selected.

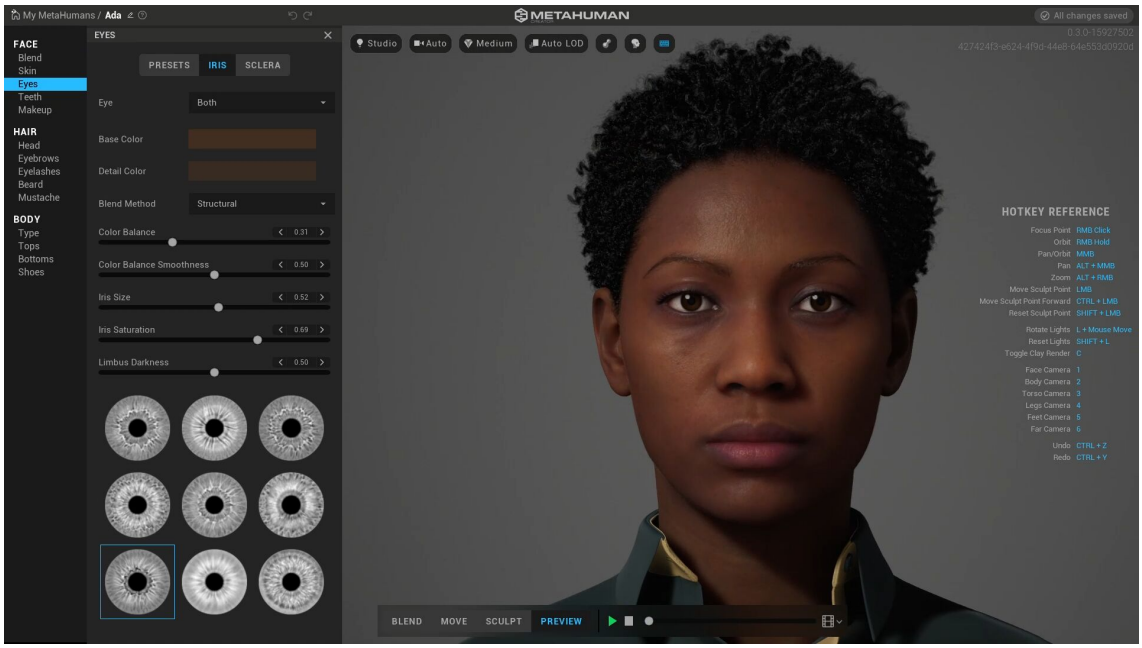


At the bottom of the My MetaHumans gallery are the Delete and Duplicate buttons. These can be used to remove any MetaHumans you no longer want or need, or choose to duplicate ones that you want to iterate on or create a variation of.



MetaHuman Face Controls

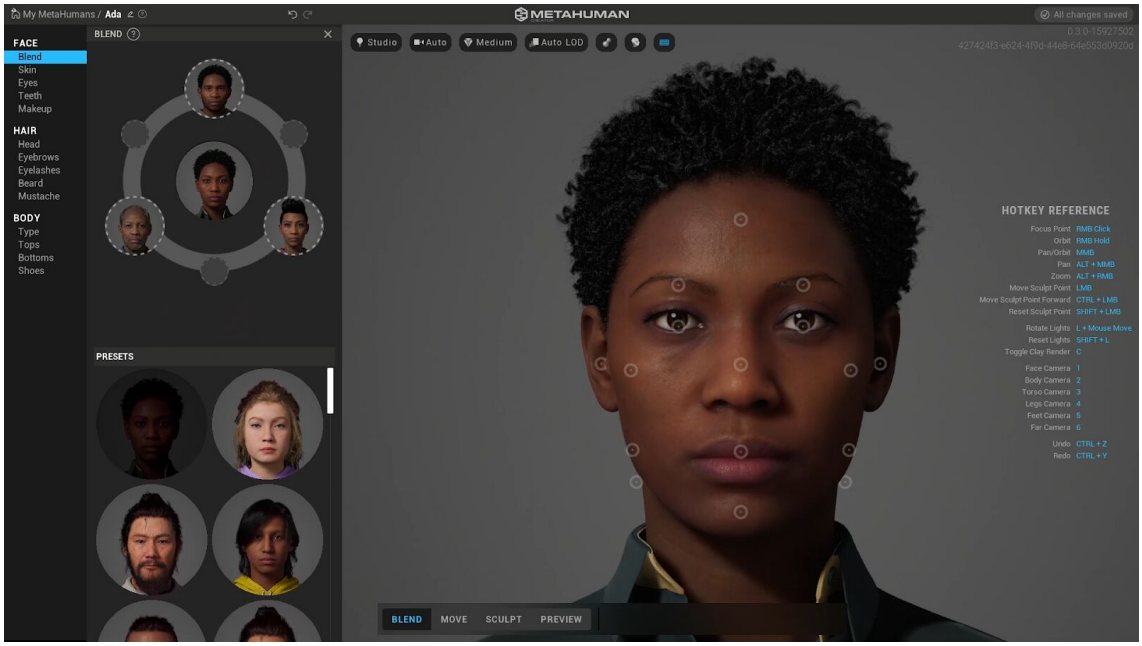
The Face controls provide properties relevant to the attributes of the face. You have control over regional facial characteristics, skin tone and accents, eye color and look, teeth shape and style, and a selection of makeup styles to apply to the MetaHumans you create.



MetaHuman Viewport Controls and Toolbar

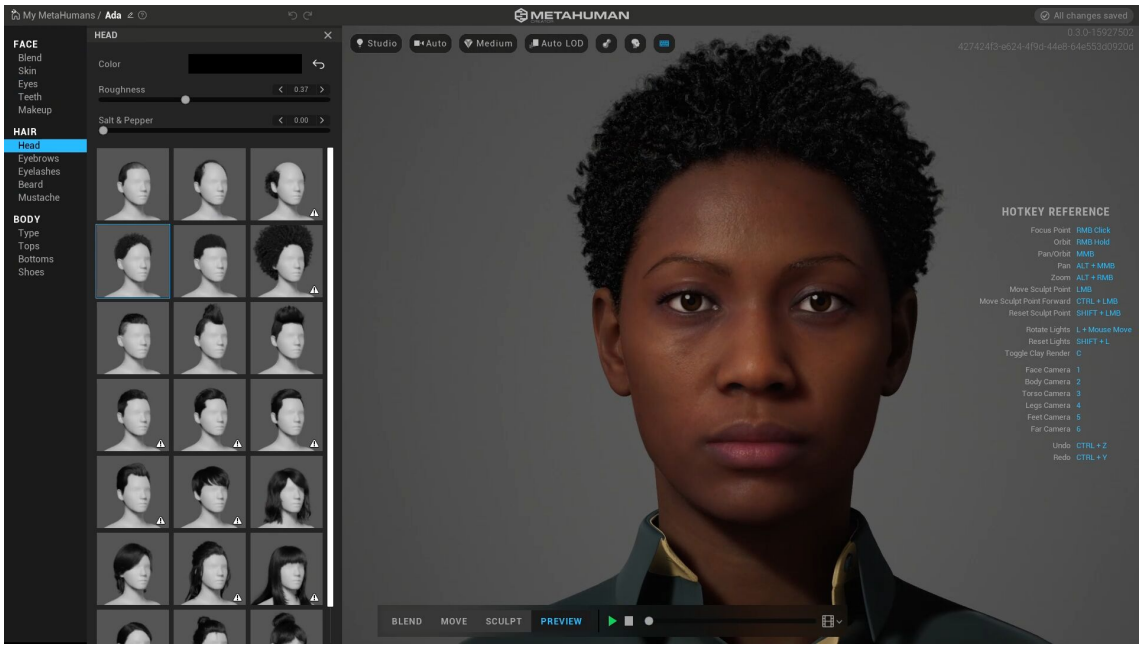
The Viewport Controls and Toolbar provides access to lighting and quality options along with a dedicated toolbar to sculpt and play animations on your MetaHuman characters.

The lighting and quality controls can be used to test a MetaHuman in differently lit environments with different quality settings for lighting Level Of Detail (LOD) of the MetaHuman character mesh. The viewport toolbar contains various editing controls for MetaHuman to add regional influence through control points and blend targets, sculpt and move parts of the facial topography for large and small changes, and play animations on a MetaHuman.



MetaHuman Hair Controls

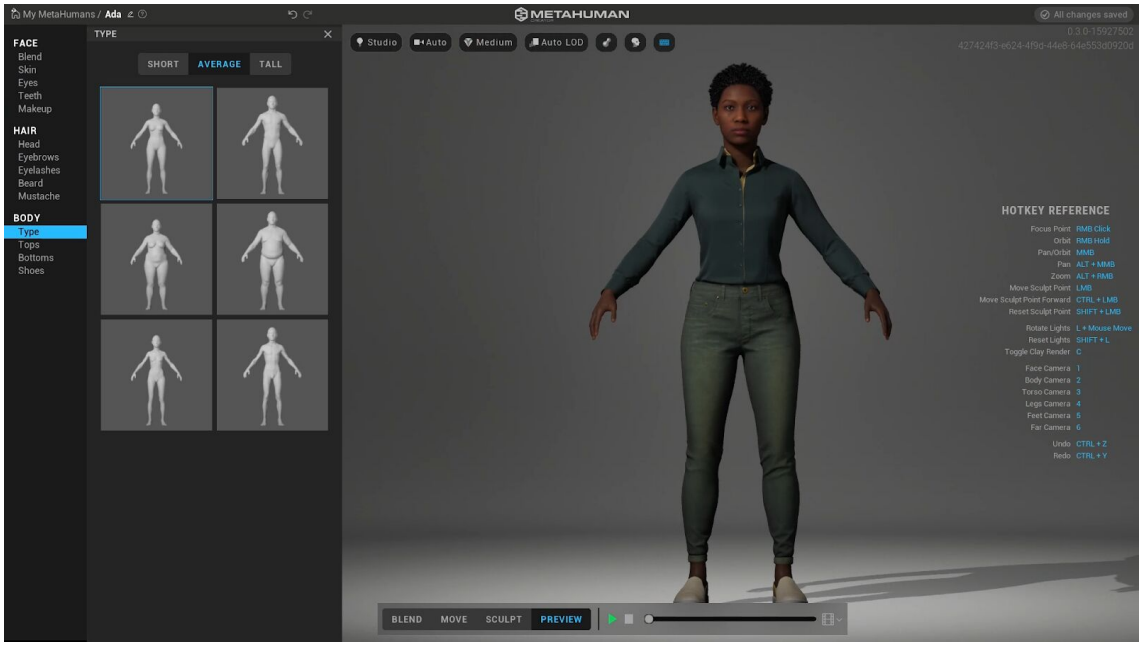
The Hair controls provide a selection of styles to choose from and properties relevant to the attributes of hair on a character's head. The user has control over defining the type of hair style for head, eyebrows, eyelashes, beard, and mustache for the MetaHumans you create.



For details about these properties and their usage, see the [MetaHuman Creator Hair Selections and Controls].

MetaHuman Body Type and Clothing Selections

The Body controls provide a selection of body types and heights, as well as clothing choices for MetaHumans.



Level of Detail

MetaHumans are made up of a handful of different components within a Blueprint. Some of the components share the same number of **Levels of Detail** (LODs) while some have fewer levels of detail or use different types of geometry to represent them..

MetaHuman Blueprint LODSync Component

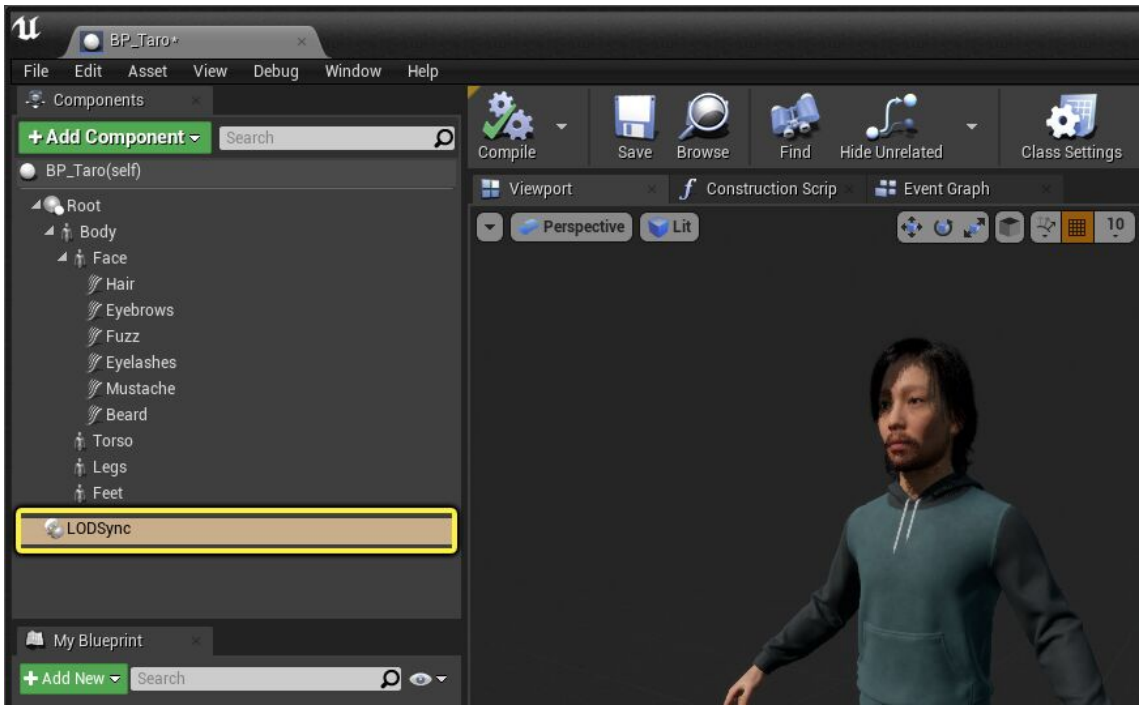
A MetaHuman's Blueprint brings together all of the different components that make up its visual representation. Not all of these components share the same type of geometry or number of levels of detail. So, there needs to be a way to manage and synchronize them so that they switch at the same time and not individually based on their own screen size, like normal LODs do.

The **LODSync** Component manages the individual components that make up the MetaHuman for the body, head, hair, and clothes. It maintains a visual quality and uniformity of them to one another even when they don't share the same number of LODs.

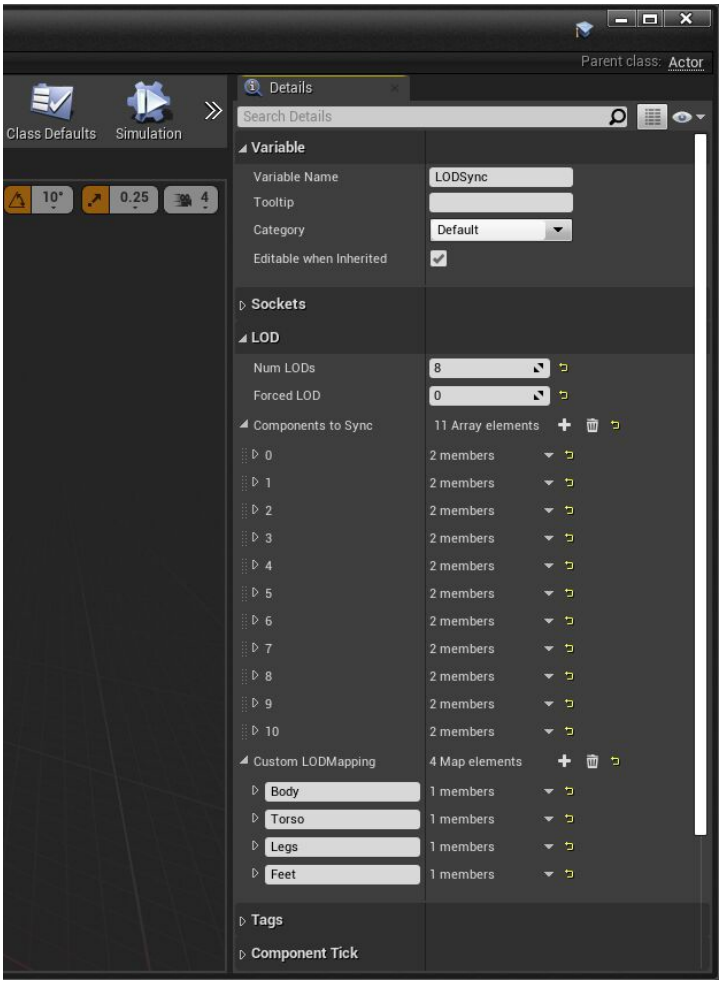
For example, MetaHumans do not all have the same number of LODs for each component that they have. Components above the neck can use up to eight individual LODs. For components below the neck, there are only four LODs. Since there is a

disparity of more LODs with some components than others, the LODSync Component enables some of them to tell others when it's good to change. This means that for every two LOD quality changes that happen for components above the neck, only one LOD quality change happens below the neck.

The LODSync component is located at the bottom of the **Components** panel stack in a MetaHuman Blueprint.



When the component is selected, the Blueprint **Details** panel displays the configurable settings under the **LOD** category. By default, it's set up to work well for MetaHumans, but you can change settings as needed to fit your needs.



The **LODSync** component contains the following:

property	Description
Num LODs	Sets the maximum number of available LODs that any component in the list can use. By default, MetaHuman Blueprints already have their value set to 8 for the maximum number of LODs used. If -1 is used, all subcomponents' LODs are calculated to determine the maximum number of LODs to use.

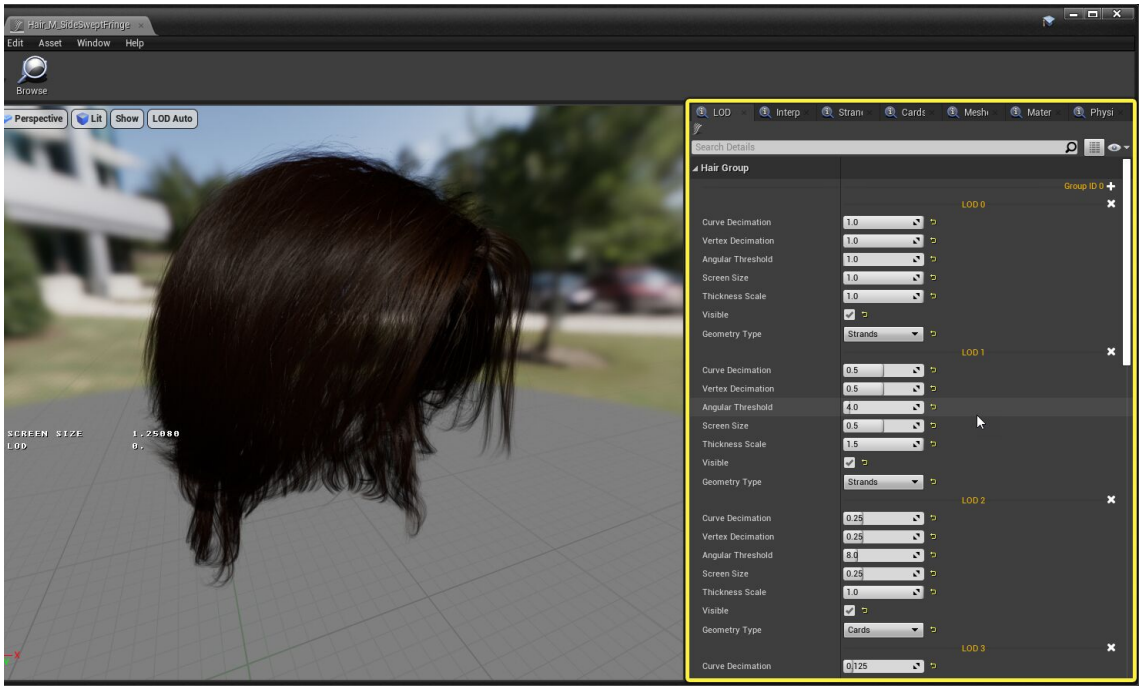
Forced LOD	Select a LOD to use across all subcomponents. If you want to have the highest quality LOD used across all components, enter 0. If you want the lowest quality display, use 8. If -1 is used, LODs will automatically switch based on settings configured by the LODSync component.
Components to Sync	This is an array of components whose LOD may drive or be driven by another component. Components are given a **Name** a **Sync Option**. Sync Options provide three ways for how this component syncs with other components: **Drive:** Causes this LOD to contribute to the change of level of detail. **Passive:** Causes this LOD to follow what is currently driven by other components. It doesn't contribute to the changing of level of detail. **Disabled:** It is disabled and does nothing. Components that are flagged as **Drive** are treated as being in priority order with the last component having highest priority. The highest priority visible component sets the LOD for all other components. If no components are visible, then the highest priority non-visible component sets the LOD.
Custom LOD Mapping	This is an array of components by name and body part. It provides more control over how LODs switch. For MetaHumans, each mapping contains 8 LODs and maps them accordingly.

MetaHumans Groom Asset Level of Detail

MetaHumans, like their real-life counterparts, have varying amounts of hair that covers their head and face. This can include hair on top of the head to the eyelashes and eyebrows, mustaches and beards, and the vellus (also called "peach fuzz") hair.

All of these types of hair are stored as Groom Assets in Unreal Engine. Each of them is managed by their configuration in the Groom Asset Editor. And, because Grooms can be made up of different types of geometry from individual strands to cards to a low-poly mesh, it's important that they be configured and set up to display when they're needed.

The **LOD** panel, in the Groom Asset Editor, manages all the individual levels of detail there are for this Groom. Each can be configured for the amount of decimation they have, the screen size they should switch to the next LOD, and the type of geometry they should support. The LOD panel pulls all these geometry types into a single panel for ease of editing and viewing what each level of detail looks like.



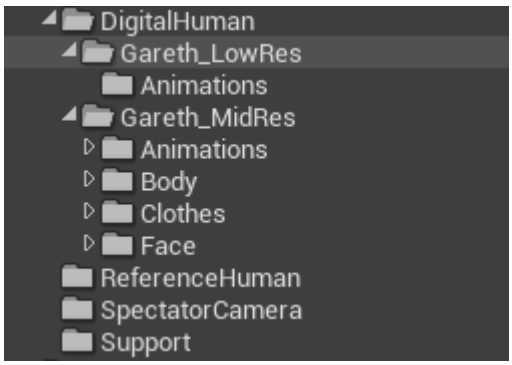
The geometry types for **Strands**, **Cards**, and **Meshes** each have their own panel where you can configure their individual properties. The examples below demonstrate the highest LOD for each:

(Left to Right) Strand, Cards, and Mesh geometry types

When targeting specific platforms or wanting to stay within a set performance budget, you can specify in the Groom Asset Editor under the LOD panel a **Minimum LOD** to use. For example on your MetaHuman, if you don't need to use strand-based hair, you could set the minimum LOD to 3, which is the highest quality LOD for the card-based hair geometry.

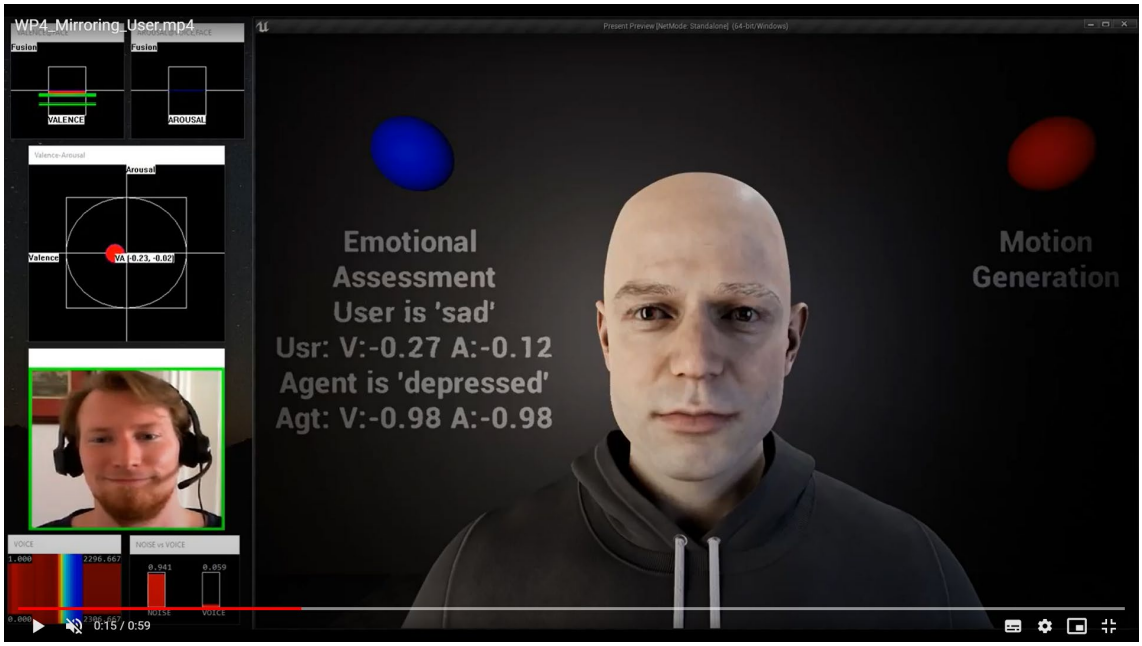
2.3 DEMONSTRATION

In order to leverage the power of the MetaHuman framework support for the MetaHuman agent has been added to the Present reference implementation. This agent is referred to within the reference implementation as Gareth_MidRes in reference to Gareth, the name of the subject actor, and MidRes referring to the resolution of agent that the MetaHuman is meant to fulfil on the project - HiRes being the Framestore agent and LowRes being a stub agent implementation used for testing purposes.



- MidRes agent (since reference implementation version ***3.2.0***)
  - Produced and maintained by Cubic Motion.
  - Use third party plugins ***Control Rig*** and ***Rig Logic***. These need to be enabled in the Unreal project for the face animations to work. This is done by default in the Reference Implementation project but if any partner wants to copy the assets to another project those plugins need to be enabled before migrating the assets!
  - Body animations, face control rig animations and audio files are synced up in *level sequences* which are produced offline. Sequences are associated with a specific map and are added as data properties to the DigitalHuman blueprint. Sequences are then identified by index when playing them.

In order to demonstrate the support which has been added to the reference implementation for the MetaHuman based mid-res agent a demonstration was conducted as part of the M18 POC. This demonstration shows the mid-res agent being driven within the reference implementation by a user whose emotional state is being mirrored by the agent.



https://drive.google.com/file/d/11BIkpx7PP_QzhOUYcUqaTiaU6kaZG33e/view?usp=sharing

A full video of this demonstration is included as part of this deliverable - D3.4.

3. COMPOSITING AS LAYERS

3.1 INTRODUCTION

The graphics that game engines, such as Unreal Engine, are capable of producing has advanced considerably over the last few years. There has also been significant progress made in the power of the hardware on which these game engines run, specifically GPU raytracing implementations such as RTX.

This combination of factors means that the methodology, as set out in the original Present proposal, of rendering a character in realtime and then compositing this into an offline rendered background no longer represents the most efficient approach to creating a high-quality output. The combination of realtime and offline rendered content set out in the original proposal was always likely to be complex and non-optimal in terms of flexibility. Thanks to a number of key factors we now feel that all aspects of the various use-cases of the project can be achieved in realtime within the game engine without the need for offline rendering.

3.2 METHODOLOGY

The approach for all the use cases on Present, with the exception of the Broadcast Sports Analysis, is to render all agents and backgrounds within a single game engine project.

The key ingredients that have made this possible are the multiple advanced levels-of-detail present in the MetaHuman implementation, which make them highly computationally efficient for multi-agent crowd scenarios. Although the running of multiple crowd agents within the reference implementation is yet to be tested we feel, based on current experience, that this should be achievable at interactive framerates.

For the hi-res agent, advances in game engine technology, combined with the increased computational efficiencies achieved by utilising the cutting edge machine learning character rigging research carried out by Framestore as part of WP6, mean that even this hi-fidelity digital human can be animated and rendered in the same scene as a highly realistic background on a hi-end consumer workstation at interactive framerates.

Despite the use of offline rendered content no longer being considered for the project there is still one use case that requires a compositing step. This is the Sports Broadcast Analysis use-case where the agent is being composited into a live-action camera feed. The compositing for this use case is achieved using Brainstorm's Infinity Set software and is detailed in D7.1 - Interim Virtual Studio Pipeline Report.

3.3 DEMONSTRATION

As there is no business case for compositing realtime and offline rendered content in layers, there is no demonstration in this deliverable at M24. The work connected with this work package has been focussed on integrating metahumans into the project and

optimising the machine learning aspects of the high-res agent to enable all content to run in a single interactive scene.

This being said a business case does exist for the agent to be composited into a live-action feed for broadcast purposes. The use case that requires this capability is the Broadcast Sports Analysis use case led by Brainstorm. As part of the M18 Present review a POC of this use case was created. This POC clearly demonstrates the mid-res Meta-Human agent running within the reference implementation and being composited into a live action broadcast using the Brainstorm InfinitySet Software.

Some images from the POC are shown below along with a link to the video of the demonstration.

https://drive.google.com/file/d/1EjFG8mp1PvpVYM7c_LzSDBKsoAc1me26/view?usp=sharing

3.4 NEXT STEPS

Multi-agent support has recently been added to the reference implementation (version 3.2.2). Additional support has also been added for the crowd software UMANS, which is being used by Inria for the multi-agent crowd simulations required of the Greek Chorus and Complex Social Situation use cases being implemented by Crew.

Work is also ongoing between Cubic Motion, INRIA and Brainstorm to allow the background agents to interact with their environment, both virtual and live-action. This work consists of passing data to the agents to allow them to look-at and point-to targets, as well as avoid obstacles.

Reference Implementation - MultiAgent (Inria)

- *Purpose:* Integration to Inria's UMANS software.
- *Input:* Crowd simulation data from the UMANS API.
- *Output:* Position, orientation, trajectories and gaze targets for each of the simulated agents.
- *Connections:* Motion Generation, Agents (Digital Humans)
- *Integration:* Locally built DLL.

Due to the Covid19 travel and shoot restrictions the motion capture sessions required to supply the animation for these crowd simulations have only just been carried out.

The support for these features has only been added just prior to this deliverable and so no active demonstrations of these crowd simulations driving multiple background MetaHuman agents is yet available. However this is a current area of focus and is expected to be achieved by mid/late September.

And finally, another aspect worth noting is that, although the diversity of ages, genders and body types provided by MetaHumans offers enormous gains to the project in terms of inclusivity it also presents a challenge in terms of animation retargeting and audio. The original project scope only included motion capture and audio record using a single actor - Gareth. A limitation is therefore that when retargeting Gareth's performance and audio it is inherently male despite what character it goes on to.

To address this Cubic Motion have captured a limited set movements and audio of a second female actor that may be able to be used depending on project time constraints. This should help boost the underlying diversity.

4. CONCLUSIONS

The technology and software on which the Present project is based, ie game engines and hi-end graphics workstations, have advanced significantly since the project was conceived. As such certain approaches that seemed logical when the project was proposed no longer represent the optimal methodology.

Equally exciting new possibilities have opened up that were not foreseen at the project's outset. This deliverable D3.4 is an example of both of these and has therefore changed significantly in scope.

The methodology of combining realtime and offline rendered content was always a complex workaround necessitated by the lack of realtime computing power and the reduced visual quality output from game engines. Since this approach was put forward 3 years ago both computing power, game engine graphics and machine learning techniques have advanced to the point where rendering entire scenes, inclusive of multiple agents and backgrounds or a single hi-fidelity agent and background, are now possible within a single game engine project on a high end graphics workstation. This provides a simpler, more efficient and flexible approach and has therefore become the sole focus of the project.

Equally the MetaHuman framework for diverse background character generation did not exist when the Present project was proposed. Thanks to Cubic Motions integration within Epic Games the Present project has been able to take advantage of this significant advancement in the production of background agents far in advance of the public release.

By adding MetaHuman support within the reference implementation all of the project partners are now able to take advantage of this technology to produce highly efficient, bespoke background agents that can be used within the framework of the Present project for any of the use-cases. This represents an enormous advantage for the project as a whole and helps to answer one of the key diversity concerns presented by the previous single actor focus.

The Covid19 restrictions have delayed the motion capture sessions required to supply the necessary animation for the background agents. This has meant that a demonstration of multiple background agents has not been completed in time for this deliverable. However, with this motion capture data having been recently gathered and support for the UMANS crowd software now added to the reference implementation multi-agent all the ingredients are in place for this demo and results are expected shortly.