**3GPP TSG-SA WG4 Meeting #131S4-250227-r1**

**Geneva, Switzerland, 17 - 21 February 2025**

**Source: Tencent**

**Title: [FS\_AVATAR] pCR on Metahuman representation format**

**Agenda item: 9.7**

**Document for: Agreement**

**1. Introduction**

Metahuman is a popular avatar representation format that is already documented in the permanent document. In order to address the interoperability challenges of dealing with different avatar formats, it is proposed to document this representation format into the TR.

**2. Proposal**

It is proposed to agree the following changes to the 3GPP draft TR 26.813 v1.0.0

\* \* \* First Change \* \* \*

[XX] "MetaHuman assets overview" https://dev.epicgames.com/documentation/en-us/metahuman/metahuman-assets-overview

\* \* \* Second Change \* \* \*

## 6.4 Metahuman

### 6.4.1 Metahuman Avatar Representation

Metahuman [XX] is a set of avatar tools that were developed by Epic Games and integrated into their Unreal Engine 5 game engine. These tools are meant to ease for developers the creation of animatable realistic-looking avatars and use them in their games and applications.

Metahuman Creator is one of these tools that allow the creation and personalization of realistic Avatars. The user may start from pre-built Avatars and customize each characteristic of the Avatar to match a particular looks.

The Metahuman avatar consists of a set of common assets and personalized assets. The common assets include:

- Body skeletal meshes which are used for the purpose of retargeting.

- Clothes and shoes meshes: these are split by gender by can be reused by all avatars of the same body type.

- Facial pose library: describes the facial expressions that are common to all Avatars. These store the blendshapes that are used to animate the Avatar’s face.

Custom assets of the Avatar include the following:

- Head and Body meshes that correspond to the user and reflect their gender, body shape, and height,

- Hair meshes that represent the user’s hair,

- A collection of materials and textures representing the face, body, clothes, and hair. The textures may include albedo, color, normal, and bump maps.

The format supports multiple levels of detail to provide flexibility based on available processing capabilities. The various assets of the Metahuman Avatar may come in different levels of detail.

### 6.4.2 Metahuman DNA Storage Format

The DNA format is a file format that is used to store all the details of the shape and rig of the Metahuman Avatar head. DNA format currently does not support the storage of the complete Avatar.

DNA file format comes in both a JSON representation and a binary representation and there is a tool to convert between both.

The structure of the DNA file may be analysed by inspecting some of the sample DNA files. The following is the rough hierarchy of the file:

- Signature: an identifying signature of the file

- Version: contains version information

- Sections: lists the included sections in the file

- Descriptor: provides some metadata about the file, such as gender and age. It also indicates the number of levels of detail stored in the file

- Definition: includes the following information:

- mappings between assets (joints, blendshapes, animatedmaps, meshs) and the levels of detail

- control information describing the supported controls.

- joint and mesh names that describe the head and facial joints that can be controlled.

- joint mappings and hierarchy as well as their transforms to the neutral T-pose

- Behaviour: provides definitions of the Avatar behaviour, including:

- Controls: that assign transform paths for the control elements upon an animation

- Joints: provides transforms for the corresponding joint groups, identified by their joint indices

- Blendshape channels: provides transforms for blendshapes during a specific behaviour

- Animated maps: provides transforms for the references maps during a specific behaviour

- Geometry: contains all mesh descriptions for the Avatar, where each mesh may have:

- Positions: provides the coordinates of all vertices

- Texture coordinates: provides the UV coordinates for the textures

- Normals: provides the vertex normal

- Faces: includes the face definitions for the mesh

- Skin weights: provides the weight associated with each specific joint, which are used to control the skinning

- Blendshape Targets: links the mesh to the corresponding blendshapes

- End of file: a terminating element to indicate the end of the file

An example of Metahuman DNA format is documented in Annex A.1.

\* \* \* Third Change \* \* \*

Annex A (informative):  
Example Avatar Representation and Animation Streams

# A.1 Example of Metahuman DNA file

The following JSON example illustrates a Metahuman DNA syntax as defined in clause 6.4.

{

"dnaVersion": "1.0",

"meta": {

"name": "MetaHuman\_Sample",

"author": "Epic Games",

"description": "High-fidelity MetaHuman character"

},

"geometry": {

"meshCount": 3,

"meshes": [

{

"name": "HeadMesh",

"vertexCount": 40000,

"indices": [0, 1, 2, 2, 3, 0]

},

{

"name": "BodyMesh",

"vertexCount": 50000,

"indices": [4, 5, 6, 6, 7, 4]

}

]

},

"rig": {

"joints": [

{

"name": "Head",

"parent": "Neck",

"position": [0.0, 1.5, 0.0],

"rotation": [0.0, 0.0, 0.0, 1.0]

},

{

"name": "Jaw",

"parent": "Head",

"position": [0.0, 1.4, 0.2]

}

]

},

"blendShapes": {

"targets": [

{

"name": "MouthSmile",

"weights": [0.0, 0.5, 1.0]

},

{

"name": "EyeBlink",

"weights": [0.0, 0.3, 1.0]

}

]

},

"animationCurves": [

{

"name": "JawOpen",

"keys": [0.0, 0.5, 1.0],

"values": [0.0, 0.8, 1.0]

}

]

}

\* \* \* End of Changes \* \* \*