**3GPP TSG-SA WG4 Meeting #131S4-250228-r1**

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

**Source: Tencent**

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

**Agenda item: 9.7**

**Document for: Agreement**

**1. Introduction**

VRM 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] VRM Consortium. VRM Specification 1.0 - Technical Documentation. GitHub Repository, 2023. Available at: https://github.com/vrm-c/vrm-specification

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

## 6.5 VRM Representation Format

The VRM specification [xx] is a platform-independent file format designed for use with 3D characters and avatars in the modern VR landscape. It is an open-source file format specifically designed for humanoid character avatars. The VRM specification is based on glTF 2.0, a cross-platform format that has been gaining a lot of traction in recent years.

The VRM specification provides a unified way to store and exchange avatar data, including the 3D model, textures, skeletal rigging, facial expressions, and even additional metadata information, such as licensing and copyright. By integrating into glTF 2.0, VRM aims to make 3D avatars more usable on all platforms, such as distribution and games.

The VRM specification is an extension of the glTF 2.0 format, designed specifically for humanoid avatars. The following list describes the VRM avatar representation and components:

- VRM Humanoid Structure: The VRM specification defines a humanoid bone structure as a glTF extension. This structure allows for the standardization of bone names and locations, making it easier for applications to control avatars in a uniform manner.

- VRM Emotion: To support anime-like styles, expressions are defined as changes of blend shapes (morph) and materials with respect to vowels and preset expressions (angry, fun, etc). These preset patterns can be extended.

- VRM SpringBone: This is a simple secondary animation system intended to be used for non-realistic hairs and clothes. It prioritizes simplicity to ensure portability rather than being feature-rich.

- VRM MToon: This is an anime look material definition provided as part of the default implementation. It has also been ported to WebGL for deployments in Web environments.

- VRM License: The EULA/License of distributing avatars can be defined and included. This is important to protect user’s appearance in the virtual world.

The VRM specification defines a “model space” for each VRM model, that observes relative transforms from the root of the glTF scene. This model space is distinct from the world space defined in the application that uses the VRM model.

VRM Animation is a format for describing animations of humanoid models defined in VRM. The same VRM animation file can be used for any VRM Avatar file. The format is described in glTF and is a cross-platform format. A standard implementation for importing and exporting VRM animations in Unity is provided through UniVRM.

An example of the VRM format is documented in Annex A.2.

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

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

# A.2 Example of VRM file

The following JSON example illustrates a VRM syntax as defined in clause 6.5.

{

 "extensions": {

 "VRM": {

 "exporterVersion": "VRM1.0",

 "specVersion": "1.0",

 "meta": {

 "title": "MetaHuman\_Sample\_VRM",

 "version": "1.0",

 "author": "Epic Games",

 "contactInformation": "",

 "reference": "",

 "allowedUser": "onlyAuthor",

 "violentUsage": false,

 "sexualUsage": false,

 "commercialUsage": "personal",

 "licenseType": "CC0",

 "otherLicenseUrl": ""

 },

 "humanoid": {

 "humanBones": {

 "hips": { "node": 0 },

 "spine": { "node": 1 },

 "head": { "node": 2 },

 "jaw": { "node": 3 },

 "leftEye": { "node": 4 },

 "rightEye": { "node": 5 }

 }

 },

 "firstPerson": {

 "firstPersonBone": 2,

 "meshAnnotations": []

 },

 "blendShapeMaster": {

 "blendShapeGroups": [

 {

 "name": "Smile",

 "presetName": "joy",

 "binds": [

 {

 "mesh": 0,

 "index": 1,

 "weight": 0.9

 }

 ]

 },

 {

 "name": "Blink",

 "presetName": "blink",

 "binds": [

 {

 "mesh": 0,

 "index": 2,

 "weight": 0.5

 }

 ]

 }

 ]

 },

 "secondaryAnimation": {

 "boneGroups": [

 {

 "comment": "Jaw movement",

 "bones": [3],

 "stiffness": 0.5,

 "gravityPower": 0.2

 }

 ]

 }

 }

 },

 "nodes": [

 {

 "name": "Root",

 "translation": [0.0, 1.5, 0.0]

 },

 {

 "name": "Spine",

 "parent": 0

 },

 {

 "name": "Head",

 "parent": 1

 },

 {

 "name": "Jaw",

 "parent": 2

 },

 {

 "name": "LeftEye",

 "parent": 2

 },

 {

 "name": "RightEye",

 "parent": 2

 }

 ],

 "meshes": [

 {

 "name": "HeadMesh",

 "primitives": [

 {

 "attributes": {

 "POSITION": [

 [0.1, 1.2, 0.3],

 [0.2, 1.3, 0.4]

 ],

 "NORMAL": [

 [0.0, 1.0, 0.0],

 [0.1, 1.0, 0.1]

 ],

 "TEXCOORD\_0": [

 [0.5, 0.5],

 [0.6, 0.6]

 ],

 "JOINTS\_0": [0, 1],

 "WEIGHTS\_0": [0.8, 0.2]

 },

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

 }

 ],

 "blendShapes": {

 "Smile": {

 "deltaPositions": [

 [0.02, 0.01, 0.0],

 [0.03, 0.02, 0.0]

 ]

 }

 }

 }

 ]

 “skins”: [

 {

 “joints”: [0, 1, 2, 3, 4, 5]

 }

 ]

}

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