**3GPP TSG SA WG 4 Meeting #123-e S4-230577**

**e-meeting, 17 - 21 April 2023**

**Source: Tencent**

**pCR Title: Signalling available visual space**

**Draft Spec: Permanent document**

**Agenda item: 9.5 (MeCAR)**

**Document for: Approval**

**1. Introduction**

This contribution proposes to address the configuration in which a visual 3D scene is required to be rendered within a specific visualization area. Such an approach is required to ensure that visual objects will fit into the available space surrounding the user. Primarily addressing a need for AR games, the proposed approach is meant to be applicable for any type of service such as AR streaming, AR conversational and other applications.

**2. Identified problem**

An AR experience is achieved by the integration of visual objects into the user environment. Depending on the available space around the user, the AR experience may be unreal by the perceived collision between the virtual 3D objects and the real environment.

**3. Proposed solution**

In order to appropriately render an AR scene into the real environment, the system needs to know the available space in which visual objects can be rendered. The visual area capability defines the available volume in size and coordinates in space inside which the 3D objects can be easily rendered.

Such a rendering space may be defined with a simple shape (e.g., cylinder, cube, sphere) for which the size and coordinates can be easily signaled.

A more complex and precise space may also be defined thanks to the device capabilities to capture accurately the shape of objects (e.g., with LiDAR sensors). In this case, the available rendering space may either be calculated in the device itself of by the server if the LiDAR scene is uploaded to the server. But this second case may cause some privacy issues. For many applications, defining the free space with a simple shape is adequate.

**4. Potential implementations**

Assuming the visualization space is represented by a simple shape, the UE needs to signal the type and dimensions of the shape.

In the example of a cylinder as illustrated below the radius, the height (if not infinite) and the angle (if only part of the cylinder is covered, as shown in green below) is signalled.



In the example of a cube, only the distance from the center of the cube to any face of it (half the distance of an edge) is signaled, assuming that the cube is centered on the viewer’s viewpoint. An extension of cube signalling is when the 3 sizes of a cubic space (depth, width and height) are signalled.



In the case of a sphere, only its radius is signalled, also assuming that the center if the sphere is the user’s viewpoint.



For the above examples here is the required signalling:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Shape | Cylinder | Partial cylinder | Cube | Sphere |
| Measures | * Height (if not infinite)
* Radius
 | * Height (if not infinite)
* Radius
* Angle
 | * Side
 | * Radius
 |

**5. How the initial problem may be solved**

With the knowledge of the available rendering space, a server can ensure that the virtual objects fit into it. **What the server decides to do remains out of scope of 3GPP and is the responsibility of the service provider**. The following is just an illustration of the possible content adaptations at the server side:

* The server may decide to downscale the 3D scene so that all objects fit into the rendering volume.
* The server may decide to clip the scene and only send the virtual objects fully present in the rendering volume.
* The server may decide to deny the service due to incompatibility between the immersive experience and the available space around the user.
* In case of multiple users in different spaces, the server may decide to downscale the 3D scene for some or upscale for others to create a symmetric experience for all users.

As an example, the following scene with 3 objects has one object within the viewing recanble (orange cylinder), one object partially inside the cube (yellow rectangle) and one object out of the rendering volume (green cylinder).



Example of clipping:



Example of downscaling:



**6. Conclusion**

This contribution presented the principle of rendering space that would need to be signalled by the client so that the server can adapt the scene to be rendered accordingly. The decision on how to adapt the virtual content to the available space remains the decision and responsibility of the service provider.

It is proposed to document this contribution into a permanent document and further analyze the potential solutions.