Interactive 360 VR

Luiz Velho

Technical Report TR-19-1 Relatório Técnico

January - 2019 - Janeiro

The contents of this report are the sole responsibility of the authors.
O conteúdo do presente relatório é de única responsabilidade dos autores.
Interactive 360 VR
Luiz Velho - IMPA

Abstract: In this document we describe a platform for creation of interactive 360 cinematic content for Virtual Reality. The system is implemented as a Unity package.

1 - 360 Cinematic VR

The modality of Virtual Reality called 360 Cinematic VR corresponds to an experience based on panoramic 360 degrees video that is played on a Headset. In this setting, the display is equipped with sensors for spatial orientation, therefore the viewer can look around the scene using head rotation.

1.1 - Video 360

The 360-degree video is captured with a special purpose panoramic camera. There are various types of omnidirectional video cameras. The most common type combines a few regular video cameras for sampling the directions on a sphere. Subsequently to the capture stage, these videos are stitched together in a post-processing stage to create a 360-degree panoramic video, usually in a Latitude-Longitude parametrization, also known as Equirectangular format.

1.1.1 - Mono and Stereoscopic Video Clips

Panoramic 360 videos can be monoscopic or stereoscopic. In the second case, the video format encodes information for both eyes (Left and Right). The way that this information is stored in the video frame is arbitrary and it is determined by the video format layout. The basic layouts are Side-by-Side and Top-Bottom. Figure 1 shows the different format layouts for monoscopic and stereoscopic video.

Fig. 1 - Format layouts for monoscopic and stereoscopic 360 video.

Figure 2 shows one frame of a stereoscopic panoramic 360-video encoded using the Equirectangular format and the Top-Bottom layout.
1.1.2 - Playing a 360-Video in Unity

In order to show a 360-video on a VR headset using Unity it is necessary to follow a two-step process. First, the video clip is rendered to a dynamic texture. Next, the texture is in turn mapped to a sphere or to the scene skybox. The process can be summarised as follows:

- Render Video to Texture;
- Mapping Texture to Sphere

1.1.2.1 - Unity Video Player

The first step of the process is accomplished through the Video Player Game Object that reads the video clip and renders it to the dynamic texture. This is illustrated in the diagram below.

![Diagram of Video Player Process]

The Video Player has the relevant parameters that control the render process, such as: playback speed; looping; and audio-visual output modes. In particular, it contains the specification of the source media, i.e., the video clip and additional audio clips. For the context of 360-degree video, as mentioned above, the Render Mode is to a target texture. The texture defines additional attributes related to rendering.
1.1.2.2 - Unity 360 Video in Sphere

The second step of the process entails the embedding of the Render Texture as a visual attribute of a sphere or the source data for the scene Skybox. The Skybox is used when there is only one 360-video in the scene. Alternatively, a Video Sphere is preferred in the case of interactive, non-linear 360-video because it makes easier to combine and cut between different video clips during runtime. The solution in this context is to associate each clip with its own video sphere that are activated to play the clip.

The video sphere encapsulates the 360-video clip through its Material attribute. We define a Pano Material which employs a Panoramic Shade that is designed specifically for the purpose of mapping a 360-degree Render Texture to the Skybox or Sphere.

Unity has developed the Skybox Panoramic Shader to display 180 and 360 degree videos in either equirectangular (latitude-longitude) or cube-map (6 frames) layouts as a scene backdrop (see ref [1]).
Figure 4 shows a Sphere Game Object with a 360-video mapped as a Render Texture using the Pano Material and Panoramic Shader.

1.2 - 3D Sound / Ambisonic Audio

Another important component in an interactive 360-degree film is the audio. In that respect, the spatial perception of tridimensionality of the omnidirectional video should be complemented by the ambient aural sensation of 3D sound.

The standard VR equipment for experiencing a 360-degree film is a Headset that has stereo audio output. The effect of 3D sound depends greatly on a binaural audio configuration, such as the one provided by headphones. Low-end headsets, such as the Cardboard, can plug external headphones while high-end headsets, such as the Oculus Rift, have integrated headsets.

The 3D sound plays a critical role in the narrative of a 360-degree film because it can be used to direct the viewer's focus of attention to the relevant points of the image where the action is happening. For example, a voice sound on the left side might indicate to the viewer that a character outside his current field of view is communicating something and he/she should turn the head to see that.

In order for the above scenario to work effectively, the Audio needs to be authored specifically for providing the directional aural cues to the viewer. There are various ways to encode the appropriate information in the audio channels. Unity features a number of tools for Game Audio and Sound Design, including Spatial Audio in a 3D scene and support for spatial audio source.

One of the most popular spatial audio format is the Ambisonic Audio. In Unity, audio clips recorded in this format can be imported and associated with 360 audio sources.
Figure 5 shows an audio source component incorporated to a Video Sphere. The Inspector panel reveals the settings for 3D sound and the associated audio clip is in the Ambisonic format.

Fig. 5 - Audio Source and Ambisonic Audio Clip.

2 - Gaze-Based Interaction

The interactive mechanisms for 360-degree VR films, because of its very nature, demand a gaze-based interaction, such that the viewer’s center of interest guides the narrative.

The basic mechanism can be divided into two elements:
- Gaze detection; and
- Gaze Action.

The development of Gaze-Based Interaction for 360 VR was inspired in the Unity package Interactive 360 Sample Project [2].

2.1 - Gaze Detection

Gaze detection provides support for tracking the viewer direction and identifying the object in the scene which the user is looking at.
2.1.1 - VR Camera

The implementation of Gaze-Detection resorts to a VR Camera. The camera should be located at the center of the video sphere. Figure 6 shows the VR Camera and Video Sphere.

![Fig. 6 - VR 360 Camera and a Video Sphere.](image)

Figure 7 shows the Unity VR 360 Camera Game Object and its associated Reticule Object.

![Fig. 7 - VR 360 Camera Game Object and a Reticule Object.](image)
As can be seen in Figure 7, the VR 360 Camera incorporates various components that, together, implement the gaze-detection functionality and feedback. These components are:

- Tracked Pose Driver;
- VR Eye Ray-Caster;
- VR Camera UI;
- VR Tracking Reset; and
- Camera Manager.

The Tracked Pose Driver ensures that only the rotation from the Headset sensors will affect the camera. The VR Eye Ray-Caster traces a probe ray from the eye through the view center in order to detect which object is currently being viewed. The VR Camera UI is the support for visual feedback, such as the Reticule. The VR Tracking Reset ensures that head tracking behaves correctly when the application is paused. The Camera Manager controls the activation of the VR Camera.

**Reticule**

The Reticule is an UI Image that indicates the view center direction and gives feedback for user selection of objects during interaction. Figure 8 shows the VR Camera Reticule.

![Fig. 8 - Image of the VR Camera Reticule.](image)

2.1.2 - VR Interactive Item

The VR Interactive Item implements the general Unity component that is used for communication between Gaze Detection and Gaze Actions in the interaction mechanism. Figure 9 shows a VR Interactive Item that is a Hotspot with its associated Region of Interest (ROI) area. When the ray probe from the VR Camera intersects the ROI Rectangle, a Gaze Action is triggered.

![Fig. 9 - VR Interactive Item and Associated ROI Area.](image)
2.2 - Gaze Action

The Gaze Action is the element that implements the interaction events for 360 VR. It is used by the Hotspot and integrates several components as shown in Figure 10.

They are as follows:
• VR Interactive Item;
• Hotspot Button;
• Box Collider;
• Button;
• Canvas; and
• GUI Image.

Fig. 10 - Gaze Hotspot Game Object
2.2.1 - Hotspot

As mentioned above, the Hotspot uses the Gaze Action mechanism in order to provide interactivity to the VR narrative. Essentially, it defines a region of the viewing sphere that, when looked at, potentially creates an interaction event. Figure 11 shows a Gaze Hotspot with its ROI Rectangle corresponding to a character in the scene.

![Gaze Hotspot in the Scene.](image)

Fig. 11 - Gaze Hotspot in the Scene.

2.2.2 - Trigger

The Trigger initiates the interaction event of a Gaze Action. The Hotspot Button monitors the time interval that the viewer direction is inside the ROI Rectangle. When this wait time expires the Gaze Action is triggered. The VR Camera Reticle gives a feedback of this process by a circular counter that animates. Figure 12 illustrates the triggering process, note the Red Circle visually showing the countdown.

![Trigger Timer.](image)

Fig. 12 - Trigger Timer.
3 - Timeline

Interactive 360-degree films feature a narrative-based audio-visual experience. Moreover, they rely on a non-linear storytelling structure to allow the viewer explore the content. In that respect, the time dimension is the fundamental ground for building the film.

The 360 Cinematic VR platform employs the Unity Timeline as the basis to create interactive narratives for virtual reality.

3.1 - Structure

The Unity Timeline is a structure containing a set of Tracks that control various Game Objects through time. A Track, in turn, is composed of Clips that span a time interval and guide the events and actions which are in effect during that particular period of time.

Figure 13 shows an example of a Timeline for an 360 Cinematic VR interactive film. It contains four tracks, as follows from the top down: two activation tracks; one audio track and a gaze interaction track.

Fig. 13 - Main Timeline for Interactive 360 VR.

The Timeline Game Object uses the Unity Playables API as a mechanism for controlling animation, audio, scripts and interaction in the Game Engine. For this purpose it encapsulates a Playable Director and bindings to the Timeline tracks.

Figure 14 shows a Timeline Game Object with its associated components and respective attributes.

Fig. 14 - Timeline Game Object.
3.2 - Interaction Track

The interactivity for Cinematic 360 VR narrative is mainly built on top of a Gaze Interaction Track, whose development was inspired on examples in the Default Playables Unity package [3].

The Gaze Interaction Track has as its primary purpose to provide means for creating non-linear narratives based on the viewer’s gaze direction.

Figure 15 highlights an instance of the Gaze Interaction Track in the Main Timeline of Figure 13.

3.3 - Clip Types

The elements for controlling the non-linear flow of the narrative are track Clips. They are the building blocks to alter the sequential time-evolution based on interaction events. As such, they collectively implement the language idioms for interactive storytelling.

The Gaze Interaction Clip types are listed below. They are described by a diagram related to their respective functionality and the parameters they require. The clip names are intuitive and auto-explicative.

Note that some Clip types have markers as parameters. These markers correspond to particular time-instants in the Timeline and they provide the basic mechanism for jumping through time in a non sequential manner.

Other Clip types are associated to a Unity Game Object. These objects are responsible for the implementation of interaction events. The fact that they are general Game Objects make possible to develop different kinds of interaction based on content and narrative.

3.3.1 Marker
3.3.1 Player

Fig. 17 - Player Clip.

3.3.1 Jumper

Fig. 18 - Jumper Clip.

3.3.1 Spotter

Fig. 19 - Spotter Clip.
3.3.1 Trigger

Fig. 20 - Trigger Clip.

3.3.1 Timer

Fig. 21 - Timer Clip.

3.3.1 Choicer

Fig. 22 - Choicer Clip.
3.4 - Interaction Objects

As mentioned in the previous section, the interaction game objects connect with the clips of the gaze interaction track to perform actions based on narrative interactivity.

The interaction objects usually have as one of its components a VR Interactive Item. This is the general API for controlling the actions of the object.

So far, we have implemented three interaction objects: Audio Source; Gaze Rotation and Gaze Marker Selector.

3.4.1 Audio Source

![Audio Source](image1.png)

Fig. 23 - Audio Source Interaction Game Object.

3.4.2 Gaze Rotation

![Gaze Rotation](image2.png)

Fig. 24 - Gaze Rotation Game Object.
3.4.2 Gaze Marker Selector

![Gaze Marker Selector Game Object]

3.5 - Interaction Idioms

The combination of Gaze Interaction Clips defines the idioms in the language of non-linear interactive narratives.

Below we list some of the idioms that have been created within the context of Cinematic 360 VR.

3.5.1 Jumper + Marker

![Jumper + Marker Diagram]

3.5.2 Spotter + Player + Maker

![Spotter + Player + Marker Diagram]

3.5.3 Trigger + Player

![Trigger + Player Diagram]
3.5.4 Timer + Jumper + Marker

3.5.5 Choicer + Marker

5 - References

