Introduction

Virtual reality and immersive video

Getting bored is a quintessential part of human nature. It is one of the main driving forces behind humanity's rapid technological development. It also makes us long for new and exciting experiences, and thus invent increasingly sophisticated forms of entertainment. It is no wonder then, that virtual reality, or VR, has become increasingly popular in recent years.

The goal of VR experiences is to fully immerse the user in a virtual environment, almost completely erasing the gap between the real and the virtual world. This is achieved using advanced human-computer interfaces. Most prominently - VR headsets, that allow the user to see into the virtual world, and various motion tracking systems, enabling the user to interact with it. Of course, like any new form of human-computer interaction, VR has found applications in most industries, but the application most closely related to the goal of this thesis, and by extension the one where most adoption has been seen so far, is indeed entertainment.

While fully interactive experiences, such as games are prevalent, the applications of VR are not limited to them. Immersive video, for example, (also referred to as "360 video" or "360° video") is a mostly noninteractive form of VR content. It benefits from the advantages of VR, but is closer to traditional film and television than most immersive experiences. While creation of interactive VR experiences shares more similarity with game development than traditional video production, immersive video is largely linear, allowing to use creative software initially developed without VR content in mind.

Compared to specialized immersive content creation programs, such software is often more powerful, costs less, and has more available learning resources regarding it's usage. These factors would make it a great choice for creators, if not for the lacklustre VR content support. Thankfully, such creative software is usually extendable via plugins, allowing developers to fill in the gaps.

The role of sound in immersive experiences

There is another key ingredient in achieving immersion in a virtual environment - audio. Although it's importance is easily outshined by the importance of sight, hearing is an essential part of human perception, and getting audio wrong in a VR context can easily ruin the whole experience. While simulating the perception of touch - another vital sense - is a very difficult task, doing the same for hearing is fortunately already within our reach. To do so, as with video, the extension of audio into the third dimension is required. Because of that, creating audio for an immersive experience is quite different to normal audio production. In addition to the usual recording, sound design and mixing work, the spatial position of sounds must be defined and animated.

An improved workflow

In order to utilise the audio production capabilities of specialised software, a workflow that allows to define the position of sounds using a DAW (digital audio workstation) is seemingly required, but a different approach may be taken. Blender - a free and open source 3D software - already provides all the required tools and an established workflow for 3D animation. The only challenge is to utilise it's capabilities to control the spatial position of sounds while still using the DAW for sound design and mixing. This can be achieved by extending both Blender and the DAW via plugins, allowing the user to use Blender's 3D animation workflow for spatial audio. An existing 3D animation can also be used, providing significant time savings in case of animated immersive video productions. This thesis aims to design and implement such a solution.

The goals of this thesis

1. Analyze available solutions for spatial audio production while focusing on immersive video production.

2. Research digital audio workstation and Blender plugin programming. (Most DAWs use a common plugin format.)

3. Design a user-friendly solution that will synchronize the spatial position of sounds in a DAW with the respective positions of objects in a Blender scene in real time, and then allow to export the resulting audio.

   I should note that implementation of spacial audio encoding algorithms will not be a part of this thesis.

4. Implement said solution, test it, and document it's functionality.