Listening_Spot: Spatialized classical music in virtual reality

During 2019 I've been working on a VR-conductor simulator called Kinisia. The game is made in co-operation with Copenhagen Philharmonic as part of their educational material. They wanted the game to feature a concert practice they call *Open Orchestra*. The idea is to spread the orchestra out and let the audience move freely around the musicians. This is an untraditional and learning experience. But how is this done in VR?

Rethink the traditional way
Normally music is non-diegetic and running through one instance, controlled through states and switches. But in our case we want to create the open orchestra feeling, so all music needs to be diegetic and each musician needs to have its own emitter, in order to create the 3D environment necessary. This means that each instrument needs to be its own instance, as shown with the string quartet in *Figure 1*.

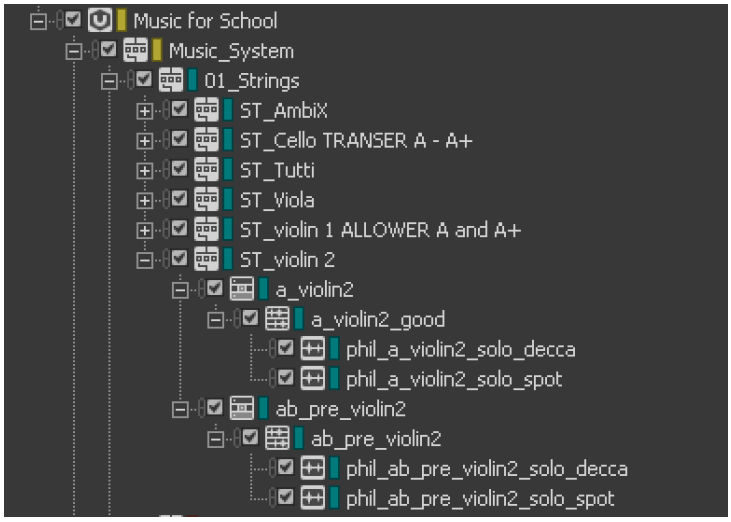

Figure 1

Having each musician be its own instance and at the same time be able to have the music change parts as the player progresses, is bound to create synchronization problems. To combat this, we created a callback system between Unity and Wwise.

When the player enters a new area and triggers the music, the Unity system receives information about what unity-state to set. In this case, it is the state called *music_trans_a*, which means that "music should transition to part *a"*. Wwise receives this as an event. See *Figure 2.*

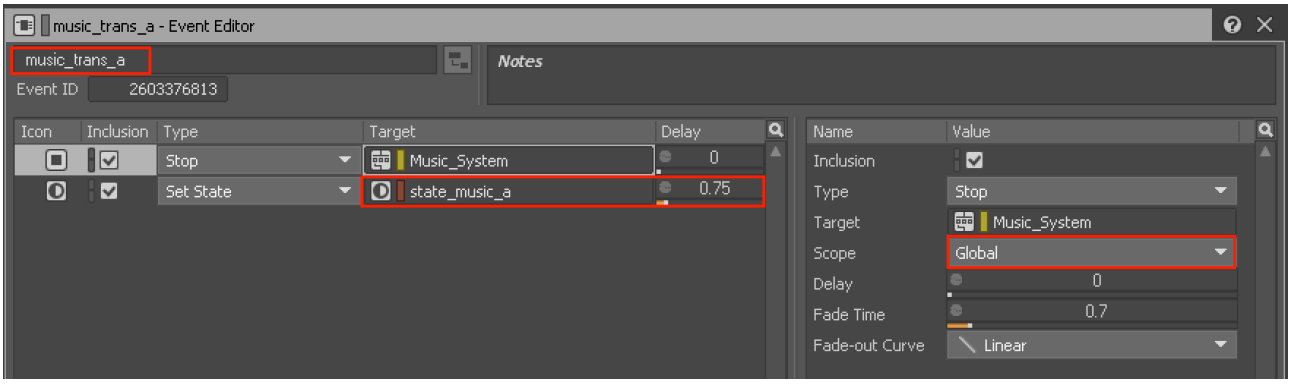

Figure 2

Wwise fades all music and then sets the wwise-state to *state_music_a*. Notice the delay of 0.75 seconds, which matches 80 bpm.

In Unity, all musicians in the level receives the same state change. They then equally wait 0.75 seconds and then tell Wwise to play the event attached to their name. See *Figure 3.*

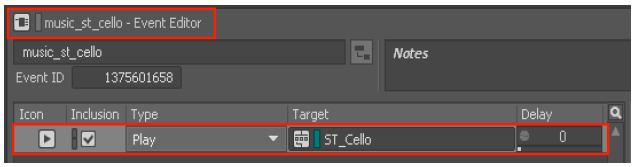

Figure 3

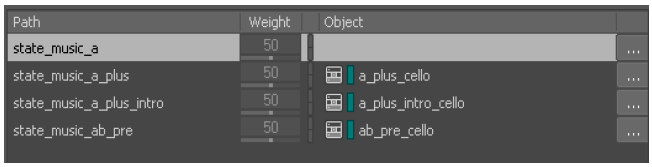

Figure 4

Wwise then checks if the instrument should play or not, depending on the wwise-state. See *Figure 4.* The cello doesn't have anything to play in *state_music_a* and therefore it remains silent.

When the music needs to transition to a new part, there is two different conditions. If the current part is looping, the system goes through a similar process of player behaviour triggers new state. In the other case, the music can't loop and will transition directly. To make sure this transition is smooth, a cue is placed on the last beat, hence the delay of 0.75 seconds. Unity receives this cue. Activate the next unity-state, stops the music, set the wwise-state, tell musicians to play, musicians play, everything stays in sync and nobody noticed a damn thing. See *Figure 5.*
The cue name is used to trigger additional content in the game.

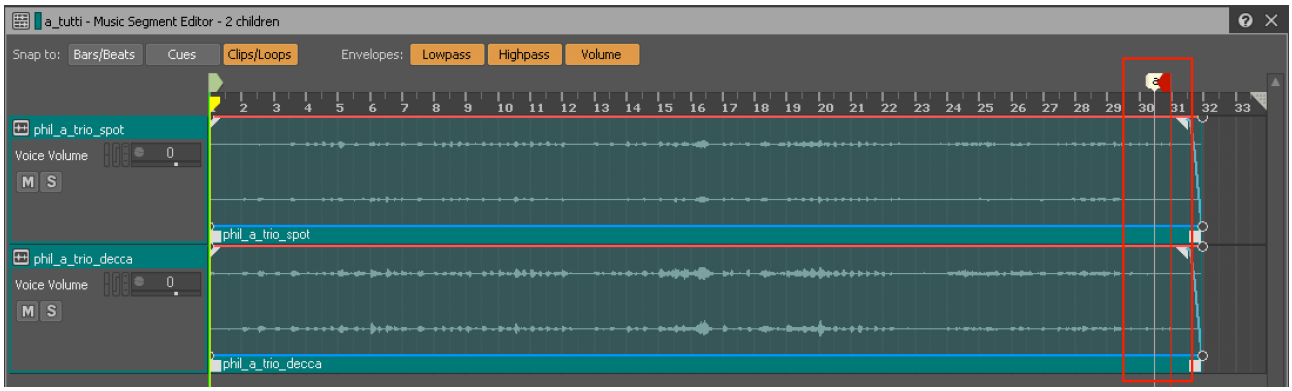

Figure 5

Realism and the use of multiple recordings

The recordings I received from the CPH Phil consisted of two types of solo recordings (spot & decca), three types of tutti recordings (spot, decca & 1st order ambix), two types of group recordings for the finale (spot & decca). With around 40 musicians in the orchestra, this would sum up to around 100 voices. This is before ambience, sfx and all other voices.
Running all of these recordings at the same time is of course impossible. Especially since the game is developed for the oculus quest, which is an android based system.
By having two musicians share the same number I quickly reduced the number of voices to 52, but I needed to reduce it even further. So first of all I had to make a system that could prioritize what voices to have active and which to run virtual, without making compromises with the quality.
Secondly I needed to find a good way of mixing the voices. Using distance attenuation was out of question. Balancing that many voices by distance and still have it sound good was impossible.

Listening spot mixing

To fix these problems I had to come up with a solution. This resulted in the Listening_spot system. Like most VR games, in Kinisia we are using teleportation as our way of movement. We use predetermined teleportation. Which means that we limit the player to only teleport to certain spots, our so-called stepping stones. This limited movement meant that I had could create a system that would recognize at what location you were and then mix the music according to that. I decided to categorize these stepping stones into three types. See *Figure 6*

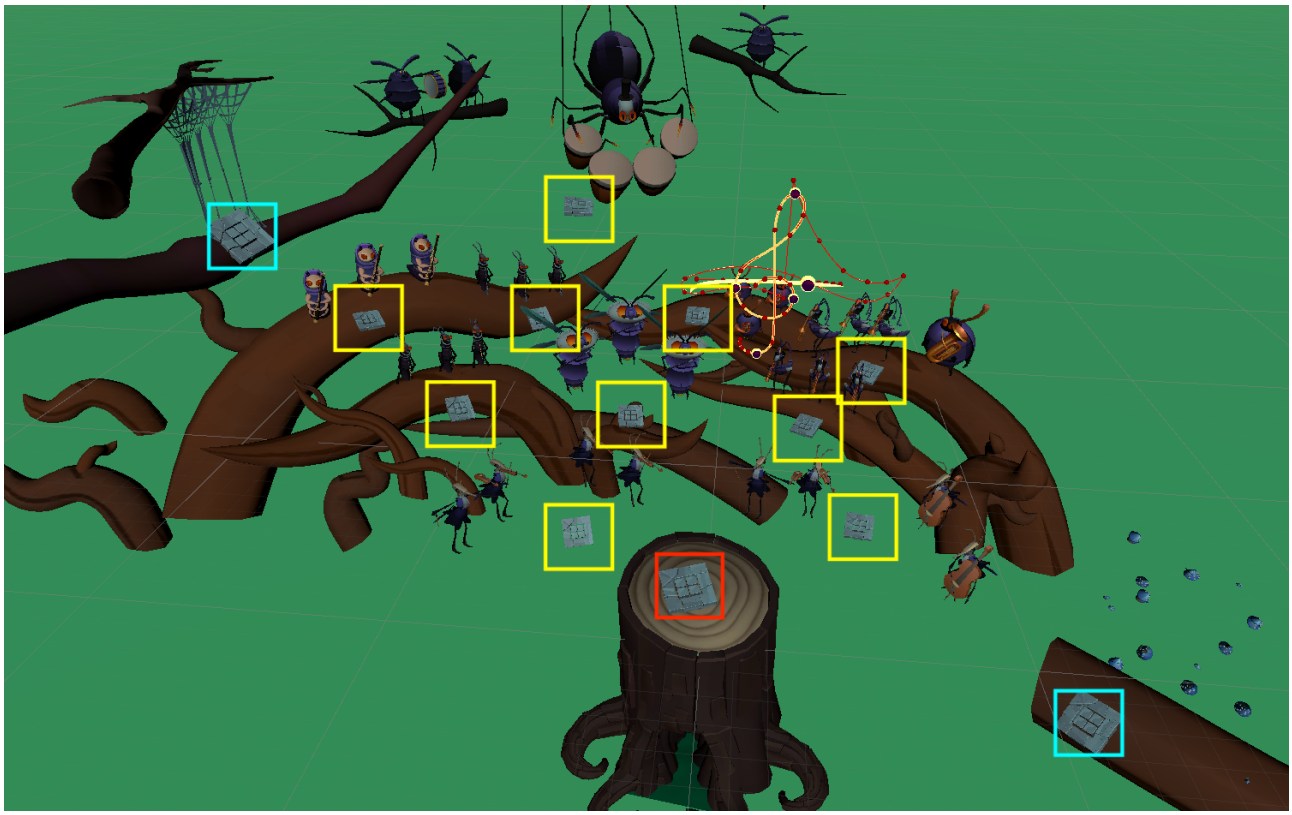

Figure 6

Music_location_tutti – The conductors podium (red)
Music_location_solo – Locations next to musicians (yellow)
Music_location_share – Locations with interactive features (blue)

The player enters the level by travelling to the *Music_location_tutti*. When doing so the music starts, the listening_spot is called and the following setting will be set. See *Figure 7.* The location state will be set to *tutti*. By doing so, all solo recordings will be sent to virtual and the player will only hear the *tutti*, group and ambix recordings.
Leaving the *Default Volume* to *Yes*, means no volume changes will be made.

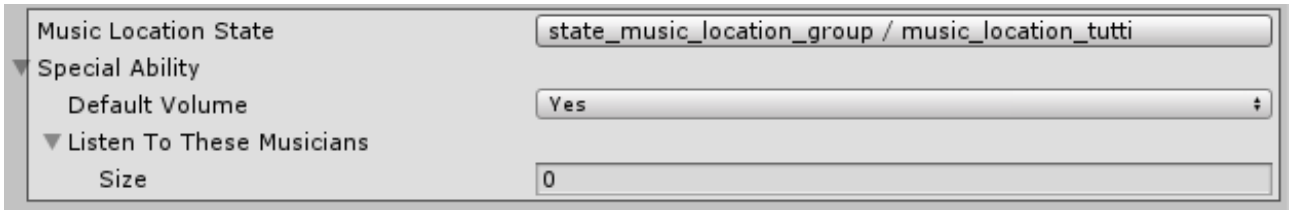

Figure 7

When travelling to a *Music_location_solo,* these setting will change. See *Figure 8.*

Figure 8

In this case the player travels to the listening spot, between the first and second violins. These musicians will be our listening focus. First of all we are changing our location state to *music_location_spot*. This means that we are sending all *tutti* recordings to virtual voices. At the same time we are activating all solo voices. Notice that we are keeping the group recordings.
The next thing we do, is change the *Default Volume* to *No.* What this does, is that it is turning the volume on all solo voices down to -200 dB, which makes them all virtual voices again. See *Figure 9.*

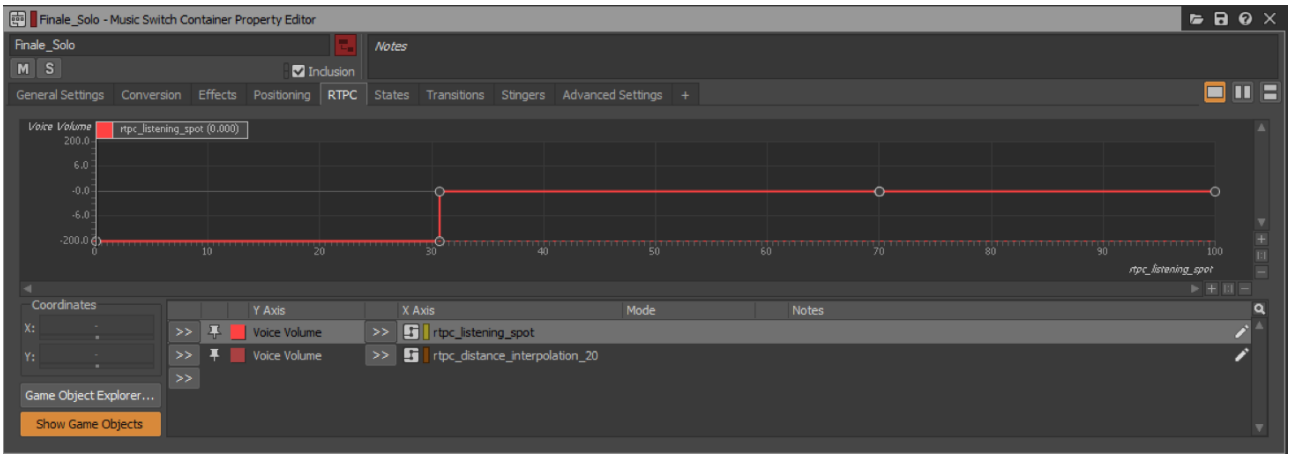

Figure 9

Then we check what musicians is connected to the *music_location_spot*. In this case it is the first and the second violins. We then change the rtpc_listening_spot value to 100, locally on their emitter, which puts them back at 0 dB.
This way we only hear the closest solo voices and the group recordings. We are also adding a little distance attenuation, so that you feel a difference between the violins close to you, and the violins further away.

This way I'm creating a mix, where we experience the open orchestra feeling through the presence of the musicians, while still hearing the whole orchestra.
The result of this is a reduction of the number of active voices from 52, down to 15-20.
Consistency in the quality of the recordings no matter your location.
A balanced, present and spatial feeling of the Open Orchestra experience.

Visit https://youtu.be/cDhYpL_FBvw for a gameplay example of the system in action.