Adam Short

Abpp113

Digital Signal Processing

Task 1:

Implemented a controllable delay using the dsp\_custom example and creating a circular buffer to store the samples. I store the value of the read buffer into the delay buffer. I then subtract a delay from the current position in the delay buffer and write this out to the outbuffer to play the sound with delay. This can be found in lines 12-17 and 58-73 in the dsp\_custom.cpp file.

Task 2:

I have attempted to implement a 3D sound by passing the camera’s properties, such as the position and velocity, to FMOD through FMOD vectors. I call a self-made method, CreateListener() which takes 4 FMOD\_VECTOR variables, inside the main game update method to constantly provide FMOD with the respective values. I also set the 3D attributes of the EventSound and the FMOD system itself. The code I changed can be found in the Update method, line 277 of Game.cpp and line 53 of Game.h. Also the LoadEventSound and PlayEventSound methods of Audio.cpp

Line 144 and the SetCameraPositionInfo was my initial attempt at setting the FMOD vectors but this worked incorrectly.

In order to see my demonstration, once you launch the Game, press “1”. The sound is positioned at 0,0,0 which is under the horse. My implementation is buggy as it’s entirely dependent on where the camera is looking. The sound has roll off once you can locate it, but again it doesn’t roll off smoothly if you’re not looking in the right direction. There also is some small form of occlusion through the horse if you position yourself above it.

Ideally I would have liked to structure this better and let the Audio.cpp handle the setting of the vectors as was attempted with the SetCameraPositionInfo method, rather than calling and setting things related to Audio in the main game file which isn’t very OO.