INFO6046 Media Fundamentals

Project 2: 3D Sounds, channel groups and DSP effects

Group members:

Euclides Araujo Jr. and Jorge Amengol

For this assignment, we have integrated the FMOD lower level API to our custom OpenGL engine. The special classes implemented were:

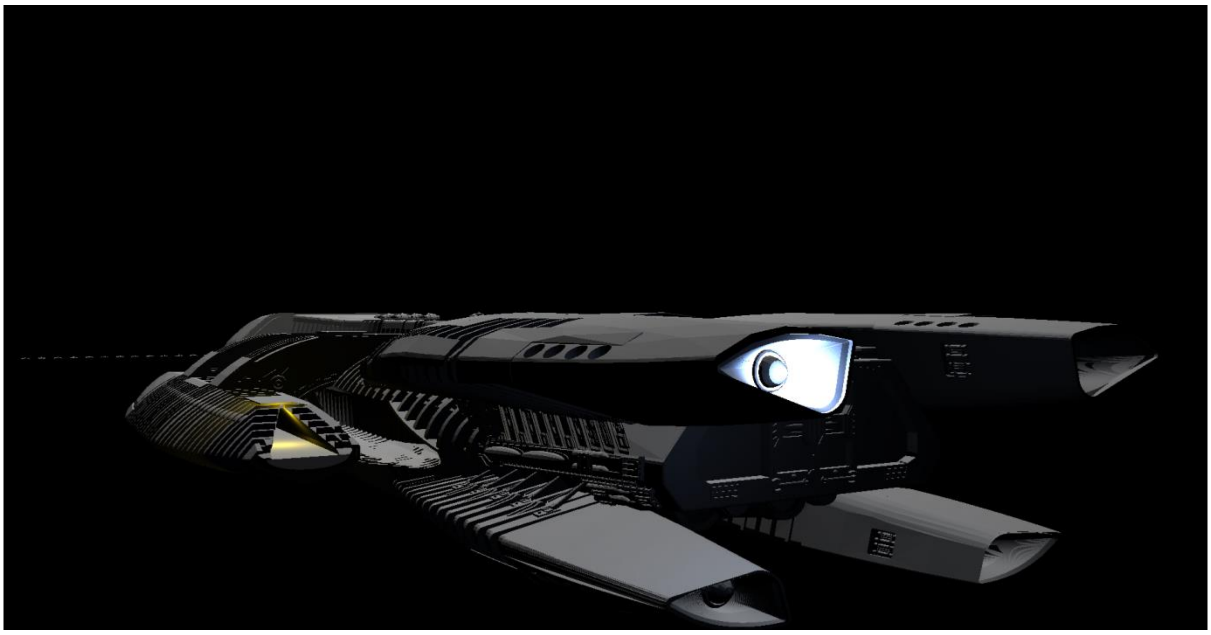
cSoundObject – Holds some states of an sound object like volume, friendly name, source of the sound, position (if 3D), etc.

cSoundManager – Controls all sounds in the scene, as well as update cSoundObject states.

cConsoleState – A state machine for controlling sound aspects and effects.

In the solution folder there is a “sounds” folder with all the sounds used for this project and some more. Also there is the sound.config that is the file to load the sounds into the engine. The format of the file is described in comments.

The scene depicts a space battle using some models from the Galactica TV series.



There are two Galactica models. The one that is shown in the image is very high resolution. For the scene as you sending it, the model is low resolution. The models can be changed in the ModelUtilites.cpp at the lines 164 and 165 (comment one and uncomment the other).

To “travel” through the scene, please use the following controls:

A/D – Move the camera sideways;

W/S – Move the camera forwards and backwards;

Q/E – Move the camera Up and Down

Arrows LEFT/RIGHT – Turn the camera left and right

Arrows UP/DOWN – Pitch the camera up and down

To see the sound and channels information, there is a console that opens up together with the scene. It is pretty much self-explanatory. Press 1, 2, 3, etc. according what is shown in the screen. To change values, press ‘,’ or ‘.’ in the keyboard too.

The questions for the assignment:

1. 20 points - Have a number of 3D sounds (at least 3)

There is 3D sounds in the Galactica engine, 4 in the lights of its landing area in the landing. All the Raiders have sounds too.

1.1. Allow user to manipulate listener position

The position of the camera is passed to the listener position.

1.2. Allow user to manipulate sound position of each 3D sound.

All Raiders in the scene (the alien line formation in front of the Galactica (after the formation of the Vipers) all have an alien sound. To start a movement of the Raider formation, hit ‘R’ while in the OpenGL window. They will start moving forward, and their position updates their sound position.

2. 20 points - Have at least 3 channel groups, with 3 streamed (usually music or ambient, “continuous” sounds) sounds each. There should be some interaction with all channel groups (using the keyboard) with the application to change aspects of the sounds. (You could press a specific key (tab or similar) to select 1 sound at a time)

3 Channels groups are created and if the program finds 9 sounds in the configuration file, each channel gets 3 sounds. If the program finds less than 9, the first channels are filled first. If there is more than 9 stream sounds, the spare sounds stay out of a channel group.

3. 45 points – Create at least 9 DSP effects. By pressing keys 1 to 9 enable/disable each effect. DSP effects must be applied to your channel groups. (at least 3 effects per channel group). i.e. effects 1 to 3 affect channel group 1, effects 4 to 6 affect channel group 2, effects 7 to 9 affect channel group 3.

9 different effects were created and can be verified though the console window, selecting the DSP effects option.

4. 5 points - Load all information from an external file (i.e. XML, JSON, txt, etc.) to allow for changing the sounds without recompiling the application. The names of the sound files should not be “hard coded”

Done through the sound.config file, inside the sounds folder.

5. 5 points - Your implementation must be “sensible” in the sense that there is some “point” to the application/demo, not just a bunch of random events. You can choose a story, or a simple game, but it has to “make sense”

It is a space battle scene.

6. 5 points - Use your own audio files; you could probably simulate/replicate an existing videogame?

We choose royalty free files.

5 points - Use OpenGL to implement this project.

Integrated with OpenGL