Audio Programing Coursework – Report

Michael Little – 1103677

This report is to coincide with the application “Audio Coursework - Soundscape” that was submitted with this document. This report will go into greater detail about the aforementioned application and how its various components were developed/used to create the final product.

Overview

A brief description of the application submitted, is a Soundscape that allows the user to wander around a virtual Battlefield and find the goal.

To allow the user to find goal, the user can freely walk around the virtual space and turn the camera/head of the character in the game. This allows the user to use the direction of the audio to determine the goal’s location by listening for the sound emanating from it.

The soundscape makes use of static sounds for the background noise, footsteps and voice work. 3D dynamic sound emitters for the various sound effects located around the map and a sound cone for the Goal sound that the player has something to head towards.

Code Structure

During the early planning stages, the application was originally going to utilise the coursework from last terms DirectX module, this would be useful as it would show that the two modules could be brought together to allow visuals and audio to complement each other and make a beautiful or though provoking scene. Unfortunately due to the coursework specifications and time, it seemed like unnecessary over complication. So to cut back the project the DirectX elements except for the camera were dropped and the scene was to consist entirely of audio.

After a few ideas were thrown around the final design for the project was to be a Soundscape that fitted an appropriate theme which would allow the player to wander around and soak in the atmosphere, while having to complete a simple objective.

To create a base to begin working from, the “Framework” classes created by Allan Milne were implemented, this allowed access to the XASound and updateable classes which allow the scene to utilise 3D sounds from the get go.

After the Framework was implemented, the soundscape itself was the next step; to create this the various components were to be added in as separate classes. The first to be added was the “SoundscapeControl” class, this is the class that would utilises all the other classes and is where the player controls and sounds would be declared. The next class after that would be the “Sounds” class this would be where the 3D sounds that would be created and the maths to allow them to detect where the player is and change their settings appropriately. Then the “Camera” class, this was essentially the player class as it would control the listener’s position and would utilise D3DXVector math. And finally the “Map” class, this is where the virtual environment for the player to walk around would be created and the goal would be set.

As mention before the “SoundscapeControl” class is where the application comes together, it contains instances of the Map and Camera class, as well as where various sound elements are declared. Using the aforementioned classes, it creates an instance of the camera to serve as the player; it then uses the map class to set up the boundaries for the player to walk around and the location of the goal. The Sound files are then loaded in and the emitters given initial locations. After everything has been set up and initialised the next step is to get the player movement and collision detection going, this is handled in the “ProcessFrame” and “DetectCollision” function. Utilising if statements the application constantly monitors the player input and calls the appropriate functions in the Camera class, (e.g. MoveFWRD to move forward) and if the player exceeds the boundaries it stops the player moving forward any further. The final task the Control class performs is calling the specific sounds to play; these include the footstep sounds whenever the player moves and the “goal reached” sound when the player enters the proximity of the Goal.

The “Sounds” class is based on the “My3DSound” class by Allan Milne; the purpose of the class is to set up the 3D sound elements that are used to populate the soundscape. In this application they are used to create the sounds of the battlefield (e.g. explosions, gunfire). Using random number generators the emitters are randomly positioned around the map, and after playing their assigned sound they move to a different location and repeat the process. The class features an updateable interface which allows the sounds to be started, stopped or moved around on command, as well as XASound and X3DAudio objects to encapsulate the sounds and maintain their position.

The “Camera” class uses D3DXVector maths to calculate vectors that are used to control the position/movement of the camera. It also contains several functions that are used to allow the player to move the camera in any direction as well as turn. The class creates the vectors of the camera by taking its position and rotation and using them to calculate the forward, up and left vectors, the camera is then moved along these vectors depending on the player’s inputs.

The “Map” class is based of the “Room” Class by Allan Milne; the class is used to create a virtual map for the player to move around and goal for them to reach. The map itself is fairly simple as it takes 2 positions: which represent opposite corners of the map, if the player ever passes these they are stopped in the Soundscape class. The other purposes of the class it to create a goal for the player to reach to end the game, and a sound cone that plays a sound to lead the player towards it. The sound cone consists of an inner and outer cone, the outer cone is larger than but not as loud as the inner cone, and this is directed down the map towards the vicinity of the player’s initial starting position and serves as a beacon for them to follow.

Implementation

The theme of the Soundscape is a battlefield, the user take control of a soldier wandering through the dark, able to hear the horrors around them as they try to head to the bomb shelter. To help create the feel of the location sound emitters randomly spawn across the map and play the various sounds of the battlefield, then repeat the process somewhere else, this gives off the feeling that several things are happening all around the player, similar to a real battlefield. The “Goal Sound” to lead the player to the goal is an air raid siren; this is implemented with a sound cone that can be heard throughout the battlefield. The use of a cone allows the player to tell if he’s heading in the right direction and gives the impression that the sound is coming out of a funnel/loudspeaker, like it would in real life. The soundscape also features voice acting to help continue the theme and help the player understand what’s happening, (e.g. when they hit the wall or reach the objective).

If the Project had to be repeated, there would be a few features that may be changed but for the most part the project would be similar. One feature that was planning to be implemented as to be a form of impending danger, so unless you reach the goal in the allotted time the player would be killed. The only problem that came with this feature was how to convey the sense of danger and how much time the player had left through only audio. In other games this could be done with piece of music designed to convey a sense of danger or the screen slowly being coloured red, but due to the design of the game and coursework specs this would subtract from the game more that add anything as it would make it harder to find the goal, and the game cant contain any graphical elements that are required for interpretation.

Another Feature that could have been added is obstacles for the player to get around; this would have worked well with the previously mentioned feature and would have overall contributed to the sense of urgency and danger of the situation being simulated. Unfortunately due to time constraints and possibly having to completely rethink the Map class it seemed like it was unnecessarily complex.

Conclusion

In conclusion I feel that the submitted soundscape not only fulfils the necessary requirements but makes good use of the audio features it has and manages to convey a sense of atmosphere. The coursework specifications stated that we needed to create an application that presents an audio scene, and then have it make good use of various audio elements, user interaction and have it all work together to create an atmospheric and engaging scene. The submitted application utilises static, 3D and dynamic sound effects; puts them to good use, and has them convey the atmosphere of an actual battlefield at night. And the theme even carries over to the design of the games mechanics, with a sound cone being used for the air raid siren and the constantly spawning sound effects helping to convey a feeling of being surrounded and confused. So in the end while I personally feel the application isn’t perfect and could have been given more depth, I feel it meets the requirements and the techniques learned this year will help me in my future as a developer.