**Project Commentary**

In my A level computing class, we learn about coding but not game development and my project idea of creating a game came about when I realised that there was a very small number of computer games for partially sighted people. This audience is estimated to be two million people in the UK alone. That's two million people without access to a major form of entertainment, so I knew there was a potential market for an action game for partially sighted people. I decided to mimic my favourite type of game, the first person shooter, only make it purely audio based.

I quickly realised that my idea was incredibly ambitious. I was originally planning on a 3D world and online multiplayer. Looking over this, I decided to make the work more manageable; I decided to make the game 2D instead of 3D in the style of *DOOM*, widely regarded to as “The most influential video game of all time” - *PC Gamer,* which shows that the format works. I also decided to reduce the game down to a basic single player mode, where zombies would chase the player. This would allow me to focus on the gameplay mechanics and features instead of advanced networking, which was too ambitious at this stage.

As writing code is a mainly self-taught, tutorials and sources are comprehensive guides to very specific subjects. This means that the number of useful or accurate sources is very small, let alone appropriate to my level.

After deciding on a more simple game, I realised that the first thing that needed to be done was to connect myself to my target audience. For this I set up a forum profile for *AudioGames.net* and asked for feedback about my design. The forum post is shown in Appendix 2, pg. 11. I had a large amount of positive feedback, as well as some constructive. Most notably there was some discussion about the feasibility of radar, after testing audio games without it though I deemed it necessary. I also decided to create an account for a few coding forums, most notably *StackOverflow.*  These were useful as I managed to get help when stuck on particular errors. This covered a variety of topics from simple syntax errors to more complicated logic errors. The most helpful thread was about a Null Pointer Exception error I kept on getting. I found out that it was because I hadn't correctly imported the art assets, so the game just crashed on launch.

To create the foundation of the game a board and a sprite for the player was needed. For this I researched tutorials for 2D games in Java. Before this point I had only really worked with console applications, displaying graphics and using the JFrame was a new skill for me. After creating movement for the player, I realised that I need collision boxes to stop the player running all over the map.

This is where the first major problem came up. I had four different attempts to write the code for this, ranging from amazingly overcomplicated to incredibly basic and useless. I eventually decided on a simple approach that can be edited at a moment’s notice. This way works by checking the coordinates of the player, and seeing if they are out of predetermined bands and if so in which direction. It would then restrict the players velocity it that direction but allow it to keep it's velocity in other directions, allowing you to scrape along the wall.

From this point on, development slowed down massively. The next major error was trying to create a gun and get the player to be able to fire it. I started modelling this in an extremely complicated and performance heavy way, modelling each bullet as a projectile. This created the error as I couldn't get it to create the bullet entity. This halted progress as I tried to grapple with the problem, until I eventually discussed the problem with a classmate and developer at Cambridge Software Jason Mashinchi. We discussed solutions for a while, and it was suggested that I use vectors instead of projectiles. Meeting with someone really helped the development of my project, even though it did earn my the nickname of “Projectile Peran” which is still used to this day.

During this time my laptop died. It was at this point where I learned the advantage of backup and started questioning the viability of my project. I was sure that my game was completely lost. To fix this, I researched, using my phone as my primary computer, how to recover data off of a broken PC. It took a while, but I managed to recover my project. While scary, this has taught me a lot about computer hardware as well as the importance of planning and preparing for unpredictable errors. From this point on, I have learnt to keep the past three builds of the project stored online.

After this the main thing left to do was to implement the sound effects for the game to get it to be actually playable. I decided to create the sounds myself. For this I had to learn how to use sound manipulation software, to remove noise from a recording, and to edit its stereo options. This was another new skill for me. The most important sounds to create were the radar sounds. The first thing I needed to do was to download an audio editor. I choose Audacity because it was open source, had a good community, and was free. I first synthesised a beep sound and then modified it individually for the twenty one different radar sounds used in the game. This taught me many skills about sound synthesis and filing, as it was important to name the exported sound files recognizable names as they all sound quite similar. Because of this, my knowledge of how sounds work has been significantly increased.

Finally, I had to design a menu for the game to be replayed. I settled for a basic affair which should show all the information needed. I then wrapped up the project by debugging it, compiling it, and set it all into a singular folder. At this point, where the game was playable, I decided to trail it on a small group of people. This feedback can be found in Appendix 2: pg. 17. This small brush with user interface was actually important, as my first rendition people didn't know how to start the game. Based on their feedback, I proceeded to add in tooltips and instructions which clarified the menu significantly, as well as an outside ReadMe.

I recorded a set of voice overs and descriptions to add to the game to help audio navigation, however, the quality of the sounds were not good enough to be introduced into the game.

Despite the numerous setbacks, I believe I have completed the aim of my project, to create an action oriented audio game. I have had to strip back features of the final build, most notably sound based on the direction of the enemy, but the game is fully playable and many testers have also said that they found it enjoyable.

I have gained many skills during the creation of this game. The most important one is that this is the first time I have managed a project all the way through from conception to execution, I have completely taught myself a new coding languages and practices. I have learnt about sounds, and game design, and communicating with potential clients. I have taken feedback from early testers and started work on implementing the improvements suggested by them.

After the project hand in I plan to continue working on my game, starting with a tutorial as that will help introduce people to the game without the for some one to explain the mechanics. This is the points that arose the most from feedback. I will then work on directional sounds for enemies as well as a better voice over for the menu. I will expand the game play with added mechanics such as ammo, reloading, and secondaries. Eventually, I will also implement the original mulitplayer vision for the game and then release it to the partially sighted community to add to the list of game that they can play. Hopefully, they will enjoy it as much as I have done in creating it.