**COMP 452 TME 1**

**Jason Bell 3078931**

**June 26, 2016**

**Overview**

This game is a 2D space shooter where you have to survive attacks by erratically moving saucers attacking your civilian transports. The saucers spawn at random intervals and random times: the interval may be very short or a few seconds.

The saucers will target the transports, and if all transports are destroyed before the time runs out, the game is lost. If at least one survives, the game is won.

The game is kept centered on the player ship, which follows the mouse cursor. The transports follow behind in formation. Pressing the left mouse button fires a projectile weapon: have fun trying to hit the saucers with it.

The game executable can be found in the /dist folder

The sprites were drawn myself, and are clear evidence of my complete lack of artistic talent.

The sound files are free, Creative Common files downloaded from www.freesound.org.

**Implementation**

This assignment was implemented in Python, with permission from Dr. Esmahi, to gain experience with that language: I already have extensive experience with Java.

There were two technical hurdles resulting from using pygame, the main Python 2D game engine.

1. One of the requirements of submitting a python game was that it be executable as a standalone program. This requires building the scripts into an exe file using py2exe. However, creating the script to perform the build process is maddeningly difficult and poorly documented. Fortunately, before giving up on using Python, I found a script provided by the pygame developers.
2. The vector maths provided by pygame are broken, to the point of being unusable. Not only are there bugs, but operations are performed in unexpected and nonstandard ways. For that reason, I implemented my own vector class. Fortunately, I had my own c# vector code and was able to port it to Python.

The only thing that I would describe as a bug (though it’s more of a glitch) is that pygame seems to do a poor job of managing audio channels. It supposedly handles audio channels automatically, but if it does, it does not do it well. You can see this while playing: often you don’t hear sounds when you should. This is most evident when UFOs kill one of the transports: the audio of the explosion often doesn’t play. I assume that this is because the UFO’s laser sound is already running. This is quite disappointing to me, as they should play in separate channels, but it’s not really something that breaks the game.

This game has a number of entities:

1. Fighter is the player ship. It uses the seek and arrive behaviors to follow the mouse cursor. Pressing the left mouse button fires it’s projectile weapon, a railgun. It employs an actuator to ensure that movement is always in it’s forward direction, as would be the case with rocket propulsion.
2. Transport are the vessels that follow the player in a linear formation, and which must be protected from attack. They also use the seek and arrive behaviors, but target a position behind the vessel in front of them in the formation. If one is destroyed, the one behind it moves up to fill the gap. It uses the same rocket actuator as Fighter.
3. Saucers are the antagonists: if outside a range, they use the seek behavior to approach their target. Once in range, they start using the wander behavior to move in an erratic, difficult to target, way. When in weapons range, they use a laser to do slow, but continuous, damage to their target. Saucers do not use an actuator: they have a greater degree of freedom of movement because, well, they’re UFOs.
4. Slugs are the player ship’s projectiles. They move in a straight line at a high speed, and do a substantial amount of damage if they hit their target.
5. Splosion (called that for whimsical reasons, honestly) is a temporary explosion entity. When it is spawned, it is rendered at that position while it counts down. Once the count reaches zero, the explosion despawns.

If you wish to run from the scripts, you will need to install 32-bit Python 2.7.

Then, install any version of Visual Studio 2015. Full versions are available through DreamSpark, and the community and express versions will also work. You’ll need to install python tools for visual studio: you’ll see an option to do this in the “new project” window.

Once python tools have been installed and this project is opened, you may need to install pygame. To do this, you should see a “Python Environments” tab in the upper right next to the solution explorer. Click it, then select Python 2.7. Use the pip option to search for and install pygame (pip is Python’s equivalent to Maven).

I do not recommend trying to use py2exe to build an executable: I suggest removing BuildExe.py from the project. Then, start the game with COMP452TME1.py as the startup script.