**Term Project Proposal – Joseph Sandler, jsandler, Section B**

**Project Description:** Singing Space Invaders – A game which has the same general mechanics as the classic 8-bit game *Space Invaders,* except to fire bullets the player must sing specific pitches into a microphone. The bullet fired would correspond to a specific pitch based on C major scale. Aliens can only be killed by certain pitches, which will be likely indicated by their color. As time passes in the game, the aliens will move from the top of the screen towards the player who is at the bottom of the screen. If the aliens reach the bottom of the screen, the game will be over, and the player loses. The aliens will also be firing bullets at the player. If the player is hit, they lose a life. If the player loses all their lives, the game is over. The player will have barriers on their end of the screen to hide behind, but these can be destroyed by alien bullets. If the player kills all the aliens, the screen will repopulate with aliens and the player will be able to play again, but at a greater difficulty, most likely with the aliens moving faster towards the player.

**Competitive Analysis:** Inherently this game is similar to games like *Space Invaders* and *Galaga*, save for the pitch specific bullet mechanic. This project was loosely inspired by “screaming Flappy Bird,” where the player must make noise to get the bird sprite to fly, rather than by tapping the screen. That is another example of a well-known game which was adapted to be controlled with sound rather than standard controls.

Additionally, this project could provide an atypical way for musicians to practice sight-reading skills, since each alien must be hit with a specific pitch, and players would have to produce these pitches on the fly without any sort of melodic reference.

**Structural Plan:** The project will primarily be organized into different files containing the necessary objects, e.g. sprites, pygame framework (created by Lukas Peraza), object which manages pitch detection. The primary reason for multiple files is to help keep things easy to find rather than having to search through hundreds of lines of code. Any imported images or sound will be stored in folders “Photos” or “Sounds” respectively.

**Algorithmic Plan:** One issue that, at first, I assumed was going to be tricky was having the game constantly take in audio input to fire bullets while also running the game animation. However, this was quickly solved after close reading of the Pyaudio documentation and use of the audio stream and pitch detection within a callback function. This effectively operates in a separate thread and is constantly running in the background while the stream is open. Insertion of the Aubio pitch detection object into the function allows pitch detection to take place without interrupting game play.

Another potential challenge could be the NPC mechanics and managing the synchronized movement of all the aliens, but not precisely identical behavior. For example, having the aliens appear initially as a grid and move together, but still respond to bullet hits individually, and only have aliens that have a line of sight to the player, unobscured by other aliens, fire bullets. This could be managed by initially drawing all the aliens with coordinates based on nested for-loops. If a zig-zag pattern of movement is employed as in the original game, the aliens would travel a set number of steps across the screen for horizontal travel, with the step size being relative to the screen, to ensure that the aliens on the far left or right never went off screen. This would not change even if the aliens in the outmost columns had been killed.

**Timeline plan:**

4/16 (TP1) – Space ship which fires bullets based on specific pitches

4/18 – Basic Alien sprite and barrier sprites complete

4/20 – Aliens appear and move as grid

4/22 – Splash screens for start of game and pause/help screen; game difficulty increases with each level cleared, hard mode implemented

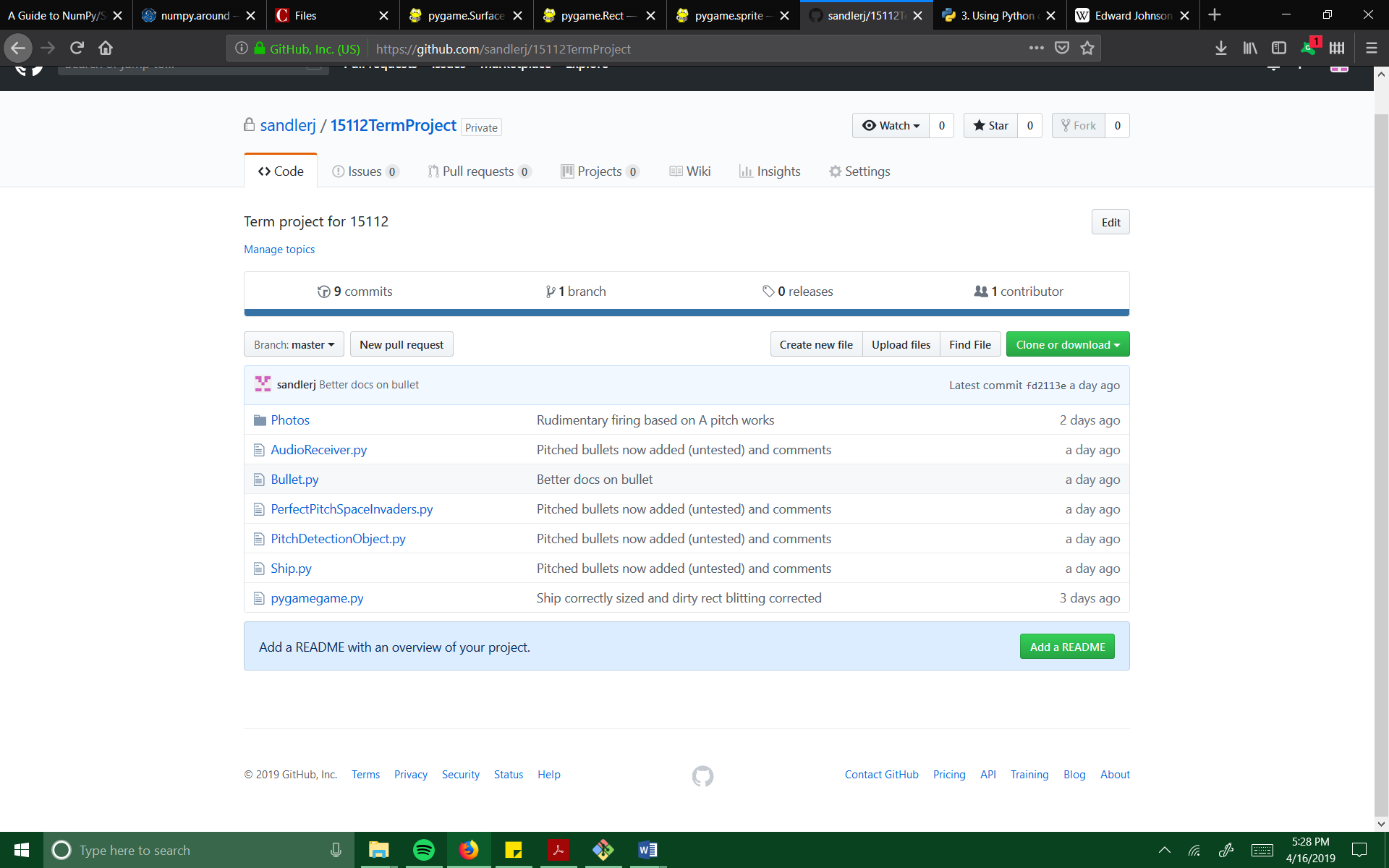
4/24 TP2 - MVP

4/26 – Cleaner graphics, if necessary.

4/30 – Possible random alien map generation

**Version control:**

I plan to back up each stage of this project using github and have already begun doing so. Repo is private but screenshot provided below.



**Module list:**

External modules used: Pygame, Pyaudio, Aubio