

Project Description: My TP is an aerial view survival game. The player will be able to choose from several classes, which all have unique abilities. They will then be sent into the middle of a room/box and have to fight against waves of enemies that approach from the edges of the screen, approaching progressively faster.

Competitive Analysis: I looked at some survival games coded up in GitHub and although there is a lot of variation in them, my design differs in the fact that none of the similar codes I found had different subclasses for the main player. There is usually one single character with a fixed set of abilities. However, my design uses inheritance and gives the game-player a choice of several classes to pick from (right now I only have Gunner and Wizard). In terms of the enemy spawn, other games also use some type of random location generation, but while these games do generation on the visible game board itself, my game finds a random location along the edge of the visible screen and then has the enemy approach into the playing field. Similar to other designs, my game also implements classes of abilities and projectiles and creates instances of them based on events, although the specific details and structure obviously differ.

Structural Plan: As far as my structure is going right now, I will have a main class for the running game itself. Then I will have a player class with multiple subclasses to represent the “classes” in the game. The enemies will also be structured similarly with different types. I will also need to create projectile or spell classes and also a function class where I put some of the functions that I use in my other classes (such as those involving randomization or vector calculations. It will look something like this:

- Game Class
 - Player Class
 - Superclass for Main Character Classes:
 - Gunner
 - Wizard
 - ...
 - Projectile Class
 - RegBullet Class
 - Superclass for other special bullets
 - Various Spell Classes (Depends on character classes and abilities)
 - Enemy Class
 - Superclass for different types of enemies
 - Functions Class (Contains static methods for random generation, calculating vectors, etc.)

Algorithmic Plan: As far as complexity goes, the two most difficult parts of my project will be using sprite sheets effectively and also generating a navigable player-friendly user interface (homescreen, selection screen, game over screen, etc). For looping through sprite sheets, I will be looking at some online youtube videos and using a similar structure adapted for my own project.

Timeline Plan:

Before TP1 (4/16): Finish at least 2 main character classes, 1 enemy class, generate a template for sprite sheet looping, add a selection and game over screen)

Before TP2: Add 2 more classes, 1-2 enemy classes, add more visual effects, add random location item drops

Before TP3: Balance testing, mostly focus on visual and gameplay improvements, potentially add more classes of player or enemies

Version Control Plan: Every few days, I will upload the most recent version of my files onto Google Drive

Drive

Search in Drive

Shared with me > Stanley Chen (stanley2)

Name	Owner	Last modified
112 TP Project Proposal	me	6:44 PM me
enemy.py	me	Apr 13, 2020 me
functions.py	me	Apr 13, 2020 me
game.py	me	Apr 13, 2020 me
mainClasses.py	me	Apr 13, 2020 me
player.py	me	Apr 13, 2020 me
projectiles.py	me	Apr 13, 2020 me
Stanley Chen (stanley2)	Emily Zhang	Apr 13, 2020 Emily Zhar

Storage: 1.8 GB used

Module List: Pygame

*****TP2 Update***:** For TP2, I changed the character basic attack from mouse click to mouse held with a set fire rate, so players can't take advantage of fast clicking. I also added enemy tracking for the Wizard Class bullets, in which I just looped through and found the closest enemy to a bullet on the screen and adjusted bullet velocity toward that direction.

*****TP3 Update***:** I tried to make the smart bird tracking better and it is slightly better than before, but now it always swings around the left side of

the rock no matter how close it is to the right side. I also tried doing a leaderboard, but the way I found online using "import sys" lagged my program too much so now I just have a txt file keeping track of the single highest score of all time that displays after the game. Game testing mode can be used if the character is selected while holding down the "C" key and you can use the 1234 buttons to spawn respective birds. (This won't update high score). And also background music.