**PROJECT PROPOSAL**

PROJECT DESCRIPTION:

Title: Spooky Situations (possibly to be changed later?)

Description: As a player, you play as a ghost trying to get out of the “land of death” to haunt the living. To do so, you need to solve three puzzles, and get three keys. Each of these keys unlock part of a riddle that you need to answer to get to the world of the living.

COMPETITIVE ANALYSIS:

Overall, this game is similar to a lot of mini-rpg type games. In terms of similarities, I think the overall structure of how to play and “win” the game is typical of the class. I think the puzzles in particular a little different than perhaps the average game, but certainly they are very similar to some games. For the puzzle mechanics, I mostly took inspiration from the puzzles in games like The Legend of Zelda, because I thought spatial type puzzles would be more fun for a player to interact with than brainteasers, because as a player can move things around until they “work”. I think once I get past my MVP and the puzzle randomization (which would be past technically, but I really want that part to work so my game is more fun), and possible implement open cv, then it will be a more unique game.

STRUCTURAL PLAN:

The final structure will include the “pygamegame” template class, the main game class, the three puzzle minigame classes, classes for the wall sprite (probably extended into a “box” sprite as well for the second puzzle) and ghost/ghost reflection sprites. There is also a “settings” class to define colors and sizes of things for easy access.

ALGORITHMIC PLAN:

The most difficult part of my program will be implementing the “large map”. I already have a method of reading/displaying txt files as a game map, however I still have to figure out how to use this in terms of the larger map. What I plan on doing is having the large map “shift” between squares when the character gets to a certain place on the screen (with each square being a block of the larger map file). When the player gets to the minigame tile, the minigame puzzle will begin. When the minigame is finished, I will overwrite the larger file and replace the minigame tiles with regular tiles, effectively erasing the “portal tile” to the minigame.

TIMELINE PLAN:

Sunday, 11/25: The three puzzle “minigames” are completed and can be run separately

Tuesday, 11/27: The larger Map is implemented with sidescrolling, and the minigames can be entered

(Tuesday-Thursday: polishing style/upgrading main elements so it runs smoothly, randomizing puzzles, fixing last minute errors/organization issues)

Saturday, 12/01: OPENCV Riddle key experimentation with text (this is EXTRA)

Sunday, 12/02: OPENCV continued implementation, audio implementation (EXTRA)

Tuesday, 12/04: All main/extra components finished!

VERSION CONTROL PLAN:

Currently I am using Google Drive to backup my program files (just copying/pasting into some text documents), and am in the process of getting my code into Github. Once I figure out Github, I will most likely use both methods.

MODULE LIST:

-Pygame

**(past mvp)**

-Open CV

-Audio