**Project Description**

*‘Who Wants To Be A Leader’ create-your-own adventure interactive activity*

Going through a narrative and filling in parts of it with user input information, responding to situations, playing some games, and based on Organizational Behavior (OB) theory (70-311) to learn something about OB and about themselves (i.e. what type of personality based on Big 5 Personality traits, leadership style, strengths, and weaknesses).

**Competitive Analysis**

There doesn’t seem to be any activity similar to this from past projects, but the ‘Hogwarts’ and ‘Lights Out’ games come close in the idea that there are multiple different ‘mini-games’ that are throughout the general narrative. My project will be differentiated with its focus on the organizational behavior side of things.

**Structural Plan**

So I think I’ll organize the final project by several .py files – one ‘master’ file that’s delineated by different modes (code for the framework taken from the animation demo notes) and each mode to be a different stage in the entire narrative (i.e. home screen, game 1, game 2, scenario, results screens). The code some of the individual games that go in those modes will go in separate .py files that I will import in order to be called in the ‘master’ file.

* Home Screen
* Balloon Game (Balloon Analog Risk Test – behavioral test of risk assessment)
* ~~Either a simple ‘Icy Tower’ or ‘Tower Match’ game (to advance to next steps of the narrative)~~ *implementing game of Othello with game AI*
* 1-2 situational questions (user to click to respond)
* A personality test
  + One of which to be Conflict Resolution Style Assessment
    - From “Conflict and Negotiation Processes in Organization,” Thomas, K.W. (1992)

**Algorithmic Plan**

~~Either the icy tower/tower match or the balloon games are the most algorithmically tricky aspects of my project.~~ Othello will be the most algorithmically complex aspect of my project.

*~~Icy Tower~~*

~~A character that can go left/right (bounded by walls on two sides) or up (probably some sort of ‘upscrolling’ instead of ‘sidescrolling’). Have platforms of random lengths appear. Character stays on the platforms, they’re good. If they fall, they restart. Probably going to have user go up for a bit and hit a ‘next step’ platform to proceed to the next step. Since I’m implemented this as one thing in the narrative of my project, there won’t be too many advanced features (i.e. moving platforms, boosts…etc) unless I end up having time for them.~~

*~~Tower Match~~*

~~No character, just side-by-side moving platforms. Random speeds, but would start slow and move faster. Side-by-side movements remind me of wraparounds; there’ll also be ‘upscrolling’ involved. Platforms stick when user places them ontop another. User’s goal is to match the pieces as best as possible. Once user places pieces~~ *~~imperfectly~~*~~, the next piece will also decrease in size. Game ends either when player completely misses and/or the length of the new piece reaches 0. Again, won’t add any fancy graphic designs like the pieces looking like Empire State Building or something, they’ll just be regular rectangles/shapes. The goal would be for user to complete something (or see how far they can go) as a way to move on to the next part of the narrative.~~

~~Regardless of which game I pick I’ll probably look at previous class notes and cite any code I use, particularly with the upscrolling aspect.~~

**Version Control Plan**

Github!

**Module List**

Good old Tkinter and game AI

**TP 2 Updates made in red**

Othello (also known as Reversi) is a two-player game each denoted as ‘black’ or ‘white’ chips. In a 8x8 grid-like board, each player takes turn moving, where one can only place a chip in a position adjacent to another chip. Particularly, one must do so in a way that a piece must ‘flank’ opposite color pieces so to ‘flip’ those colors. Game ends when either entire board is filled or if both players have no moves to play. Winner is determined by the player with the most of their color chips on the board remaining.

Algorithmically, I start out representing the board as a 2-D list where elements can be ‘None’, ‘white’, or ‘black.’ The board gets updated when chips are being placed and flipped. In the folder I have two files relevant to Othello (othello.py and WIP-othello.py) with the former being the ‘last working version’ (manual alternating between players) and the latter work-in-progress (currently not functional yet) version where I’m working on implementing game AI. I’m starting out with implementing just a random strategy for the AI. Once it works, I’ll work on a more complex AI strategy (i.e. minimax or minimax with alpha-beta pruning).