Sean Pereira
15-112
Term Project

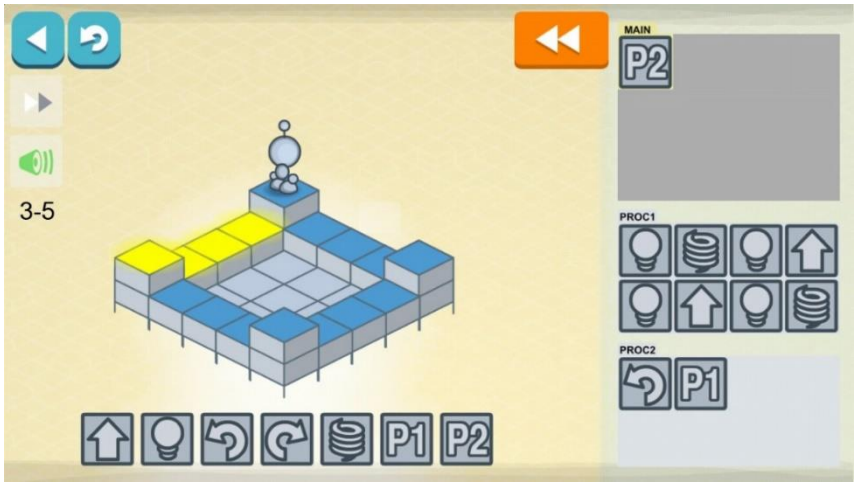
Project Description:

- Title: 112Bot
- Short Description:

  This project will implement scratch code where users select commands that are selected and will run either recursively or in a loop, verbatim. These commands must light up all the blue tiles for the user to complete the level.

Competitive Analysis:

Lightbot is an app or program from online, its purpose is to teach kids the concept of programming. The game works by a kid inputting commands into a "function" in which these commands will be called. When these commands are done the bot will follow them. The objective of the player is to light all the blue tiles. An example picture of this game is shown below:



The game has various levels of increasing difficulty. The game initially teaches you basics, but it makes concepts of programming such as loops, conditionals and recursion into a fun, challenging game. My game 112Bot will still implement the core aspects of lightbot but will have different features. These other features include a part of the game where a player can design a board that they can play and create their own solutions. If time permits, multiplayer could be added.

Structural Plan:

This game will be broken down into different classes/files in a very similar way the tech demo game asteroids was made. These different files are stated below:

- The files being written will be:
  - Game.py:
    - This will have init and control the modes
  - SplashScreen.py:
    - Will have the splashscreen redraw, when games starts

- Instructions.py:
  - This will have the instructions.
- PlayGame.py
  - Will have two modes create world or play built in worlds, this is where most of the important things are done.
- Bot.py
  - This will be a class file consisting of the bots properties, position of x,y and its image, and what happens when it takes commands.
- Board.py
  - This will be a class for creating different instances of board from a 2D matrix, and will be converted to an isometric projection consisting of x,y coordinates.
- Commands.py
  - This will have all the commands happening in the game, and it will be stored in this class, which the bot will use.

Algorithmic Plan:

The main parts of the project consist of designing the boards, implementing the commands, and running the scratch code the user inputs.

- Designing the Board:
  - ***The board needs to be an isometric projection to incorporate this there will be a 2D-List of True and False which will define that instance of the board, and the row and col position will be found through a nested loop. These x, and y coordinates will be found and finally converted into isometric points and finally placed in the game.
    *This is no longer being done, however it will be done on a 2D board with tkitnter
- Implementing Commands and Running these Commands/scratch code:
  - The commands the use clicks based on if its main function, function/procedure 1 or function 2/procedure 2 will be stored in a 2D – List. Then when the user clicks run, all the commands will be run on the list. The first list to loop thru will be the main function, if it encounters function 1 or 2, it will then call that function, and if its recursive it will keep calling itself. If this recursive function doesn't meet the win conditions of the game (base case), it will be recursing till the user hits stop which is the other base case.
- Saving Leaderboards and creating world:
  This will be done using BasicFileIO to save files

Timeline Plan:

Below is a calendar of the import deadlines, and it descriptions on the next page.

April

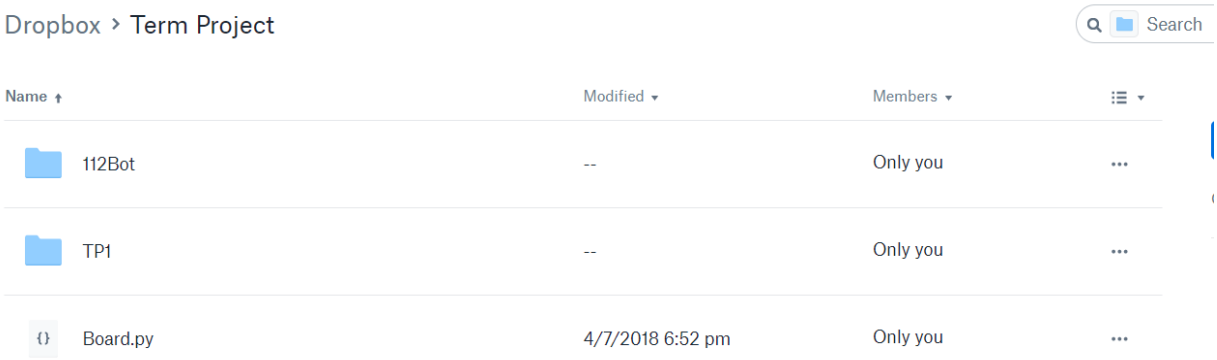
Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14:

						Meeting with Lizzy
15: Step 1	16: Step 2	17	18: Step 3	19: Step 4	20	21: Step 5
22: Step 6	23: Step 7	24: Step 8	25	26	27: Step 9	28: Step 10
29:	30	1: Step 11	2: Step 12	3		

1. Step 1:
    a. First Create all the basic modes given by 15112 and create layout for game.
    b. Then design the board, using ideas stated before.
2. Step 2:
    a. Finish coding the board, including visuals
    b. Create different game modes, main menu: instructions, play game, and create world modes.
3. Step 3:
    a. Work on designing the class of the bot
4. Step 4:
    a. Code the bot to be able to move on the board
    b. Create commands that when clicked the bot will move
5. Step 5:
    a. Create procedures and store these commands for the user to use in lists.
    b. Design a way to run these commands
6. Step 6:
    a. Allow the user to run these commands when they click run
7. Step 7:
    a. Create “create world” mode part of the game
8. Step 8:
    a. Polish the game, add features of the game, music
9. Step 9:
    a. Continue to polish the game, and add features
10. Step 10:
    a. Implement multiplayer if time is left but complete tech demo first
11. Step 11:
    a. Fix style, make comments
12. Step 12:
    a. Make the video for you to submit

Version Control Plan:

I have stored all my term project files and term project documents in a folder. This folder will be placed in the dropbox desktop app, and anytime the files are edited they will be synced to dropbox’s cloud. Proof is show below:



Module List:

The modules used in this term project are Pygame: https://www.pygame.org. Other modules that could be used are sockets, however this will be additional feature that will be added after the all the desired features are implemented. A tech demo will be completed for sockets if it is to be added.

**The module pygame is no longer being used however, was kept here to demonstrate the old files.

TP2 Update:

I have redone my bot class where it utilizes sprites. It utilizes sprites to move the bot around. Its better for animations and helps with design/user interface.

TP3 Update:

I have added a leaderboard, loading bar and create your own world feature which can be saved. These features utilize tkinter built in features.