Lisa Lo llo1

1. Description of Project: A role-playing 2D game where you start as an architect with a small firm and then make your way up the power ladder to become one of the influential architects and leader of your world. However, as you make your way up, beware the costs of the journey.
2. Problem To Solve Of Project: Create an engaging game that also teaches about awareness of people who are forgotten and wronged due to societies careless and need to move “forward”
3. Algorithmic Problems To Solve:
   - a. Main Game:
     - i. Continuously creating a world of the rpg such that it does not constantly look exactly the same/ it varies hence making it an infinite world
     - ii. Storing the data so that games, resources, and all data is saved
     - iii. Generating a dynamic main menu, and backtrack
   - b. SubGame
     - i. Creating 3D graphics in 2D
     - ii. Detecting by sound whether what someone is saying/ singing is benevolent or malign
     - iii. Continuously recording and playing music throughout game without freezing
     - iv. Casting sound to a location
4. Modules:
   - a. Pygame
   - b. Pyaudio
     - i. Speech Recognition
     - ii. Pocket Sphinx if it’ll install
     - iii. SoundAnalyzer if it’ll install
5. How To Solve:
   - a. Main Game
     - i. Continuous World:
       1. When on area full, unlock ability to go to new area
       2. Store a variety of options for different aspects of the area such as biome, size, empty plots of land, already existent buildings, etc
       3. Randomly choose some of those options which will be stored in classes
       4. Draw classes that are chosen for the new area
       5. Store to file of map
     - ii. Storing Data
       1. Have a file folder
       2. Have file with world information, resources/ inventory information, player information, basic game information such as time/date played
       3. In init function of classes, check if file corresponding exists and load data, otherwise load preset parameters
       4. When save option is pressed or game exited, update files or create file if it does not exist yet
     - iii. Dynamic Main Menu
       1. Load file with basic game information

2. Check for existence of subworlds
3. Parse through file to get location of crack relating to subworld and draw on canvas
4. Pass information of crack into Crack class that is subclass of Button Class. Draw crack with width dependent on time of the start creation of subworld which is located in the file.
5. Check for any broken worlds, change volumes of the two soundtracks dependent on if any broken worlds

b. SubGame
    i. 3D Graphics
        1. Math
    ii. Casting Sound
        1. Math and lots of research into pyaudio- not sure yet how to do this
    iii. Sound Tone
        1. For Speech:
            a. Find keywords relating to negative things and positive things
            b. More positive means higher scale, more negative lowers the scale
            c. Brighten area focused from centered out depending on scale
                i. Use another surface and change the color
            d. When negative/ goes negative, bad things happen (not determined yet)
        2. Sound
            a. Detect if in normal range of pitch= good
            b. Too low pitch= negative
            c. Too loud =negative
            d. Too soft= neutral/ no effect
            e. Add numbers to scale and lighten room as a result
    iv. Continuous recording/ playing Music
        1. Game is on one thread
        2. Recording on another thread
        3. Music on last thread

6. How Game Works Basics:
    a. Start Menu
        i. Start menu, Options, Credits- buttons get x,y
        ii. Check button list for locations of cracks
        iii. Access file relating to crack
    b. Main Game
        i. Create file recording information
        ii. Start in room
        iii. Move around up down left right- know direction facing
        iv. Enter to interact with objects in block in front
        v. Go downstairs

vi. Get requests from customer
vii. Leave building
viii. Go to empty block land
ix. Buy pieces/ build request
x. Add to text file recording world
xi. Success=customer happy, money, etc
xii. Gain money through continuous gameplay- unlock new pieces, etc
xiii. After certain time pass-places start breaking down
xiv. Can go back to fix or choose not to- notification pop up
xv. After certain time passed- permanently broken

   1. Add location to list for cracks
   2. Create file for subworld
   3. Generate situation
   4. Add time

c. Sub Games

   1. Crack clicked- find corresponding file
   2. Find code
   3. Create world relating corresponding file
   4. Class for child- include location
   5. Run code for relating subgame

ii. SubWorld options:

   1. Dark world- melancholy lonely music playing
   2. You singingly gently/ softly/ comfortably= in front glow for few seconds
   3. Move closer- music uplifts
   4. Save child- return to main game= loss resources, but reference in future to child through song or appearance
   5. Take out of list for cracks- appearance faded
   6. Also record time- time spent away= progression in timeline
   7. Too long time spent away=downward spiral

7. Update 1:

a. After TP2, I’ve decided to not do 3D subworld and instead use 2D graphics for the world. Instead I decided to use 3D for the building portion in which the building must be build in a 3D cube. Other design changes include decreasing the scope of the project. Whether or not I will manage to be able to do an infinite world depends on time. Instead, I plan primarily on polishing graphics, fixing bugs, and incorporating NPCs to the current world I have instead of adding a lot more.

8. Update 2:

a. I added customers to the game and people and made it so that you could infinitely go around and have new lands. The processing for the buildings isn’t really based on shape anymore, instead it is based on items which people request and then you provide in the building however you wish.