Adithi Rao

Section F

Proposal

**Project Description:**

The name of my project is “Speedy Diner!”. This project is a game where the user has to go back and forth between the kitchen and the dining room to seat customers and maximize how many points you get by giving them their food in the least amount of time. The customers will come in sets of 4 and each group has a separate name. When you seat a group at a table, they're order will appear in the ticket bar at the bottom of the screen and ticket will only vanish once you have completed their order.

**Competitive Analysis:**

This game is based on two games that I really loved playing as a child: Papa’s Pizzeria and Penguin Diner. Papa’s Pizzeria was a game where a player takes individual customer’s orders and prepares them to the best of their ability in order to maximize the payment that the chef receives. Penguin Diner is an online game where a waitress controlled by the player has to seat the customers at tables and take their orders to the kitchen, where a chef character (over whom the player has no real control) makes the food, which the waitress serves to customers.

My game is different than both of these games in the sense that it is a combination of the best aspects of each of the games. My game lets the user both prepare the meals for the customer in the kitchen and serve them- our player plays the role of both a cook and a chef! In general, games stay interesting when you give the player multiple types of things to do, so by adding in more responsibilities makes the game, the game is more fun to play!

**Structural Plan:**

My whole game depends heavily on classes. I have a runGame class which contains the run function for the entire game and this class uses features from the board classes which are board1 and board2. Board1, board2 and the runGame class are all instances of PygameGame. The board1 and board2 correspond to the kitchen and the dining room respectively and both of these create instances of other classes (such as the character class, which represents the waitress, as well as the customer class, table class, etc.). I have other files where these classes are initialized to make my Board1 and Board2 code as clean as possible.

**Algorithmic Plan:**

The trickiest part to my project involves creating a priority queue and the AI that will predict what move is the most strategic for the player to make (this decision will be made using the priority queue).

For the priority queue, I will need to come up with a way to assign a priority to each of the customers. This will be based on how long they have been waiting, whether or not the customer looks wealthy and what their starting happiness value is.

For the AI, because there are so many moves that a player can make, there are a lot of possible moves that the AI needs to consider. I plan for this part to involve backtracking because I will try all possible moves and store in a list the set of moves that would create the most money for the player. The AI will also be using the values from the priority queue because all of the values in the queue are important to what move is being suggested.

**Timeline Plan:**

By the first TP, I had planned to finish most of the graphics of the game, which I did, and have all of the basic functionalities that I wanted working, which includes moving our waitress to different locations, seating customers and the spawning of customers every “x” number of seconds. Additionally, I got both of the background music files to play with the appropriate boards and found a way to pass information between the two boards.

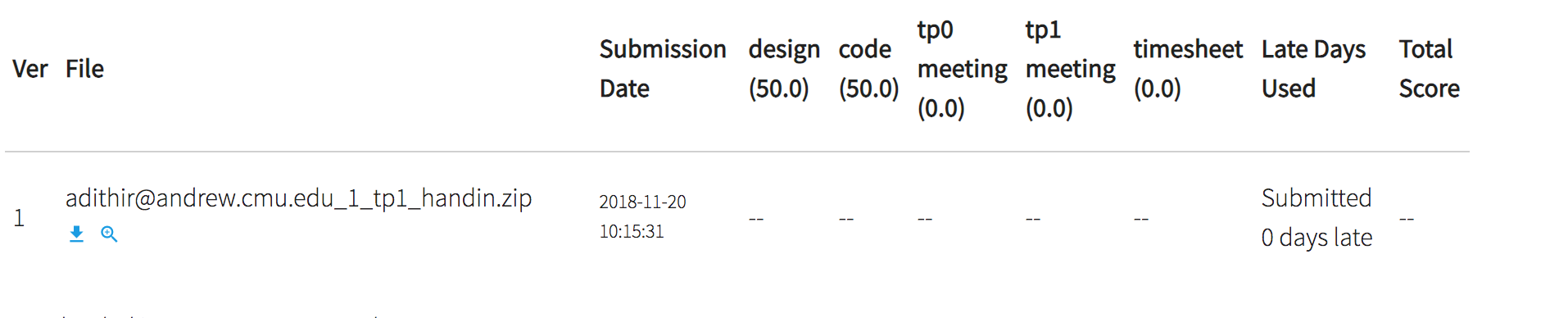
By the second TP, I plan to have all of the timer features working, in specific the time cooking in the kitchen takes and the times that each of the customers have been waiting. Additionally, I will finish implementing how each of the customers will order and keep track of their happiness levels (each a function of time) in a list. This will involve the priority queue. I also want to add in a different type of customer here (a richer one which will be recognizable by the nicer clothes the customers will wear) and these customers will be placed higher on the priority queue.

By the third TP, I will be implementing an AI that tells what move is most profitable for the user to make. This will be displayed as suggestions at the top of the player’s screen. I will also just be polishing up the game to make the User Interface a lot better looking and more interactive.

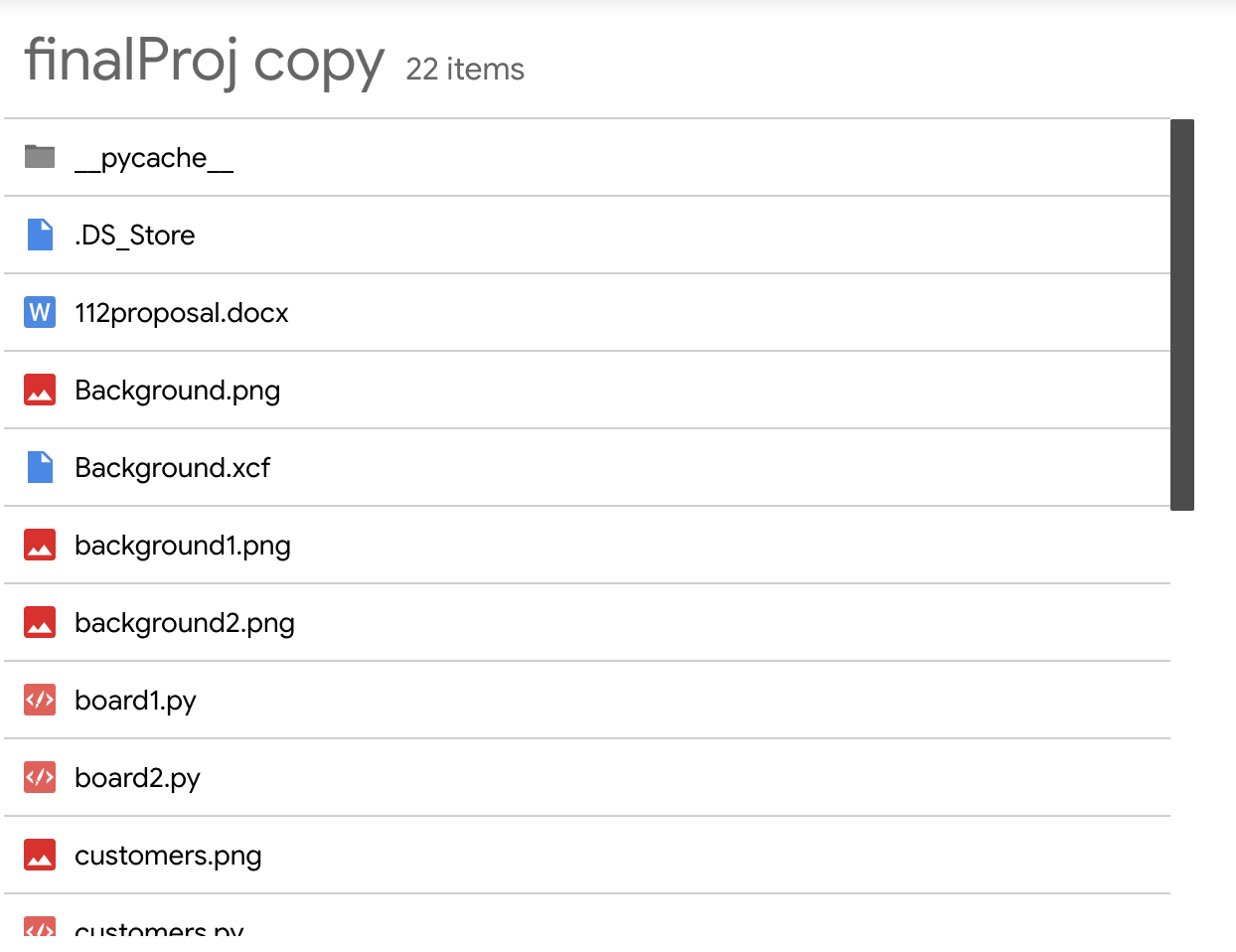
**Version Control Plan:**

To back up my code, I submitted a zip file with all of my code on Autolab (I could not zip the whole file because it contains images and therefore is larger than 10 MB). I have also been backing up my final project on Google Drive (this is my primary backup method).

Autolab Backup:



Google Drive Backup:



**Module List:**

The only new module that I am using is Pygame.

**TP2 Update**

I changed a lot of my goals for my project because adding in all the functionalities that I wanted to work in the game took a lot longer than I anticipated, so I cut out some of the additional functionalities I wanted to have (different group sizes, richness of customers). I have mostly all the functionality of the game itself at this point so all I need is to fix a few buys and figure out the priority queue. I also had a new file.IO idea (where the best score gets saved to an outputted file) that I need to implement.

**TP3 Update:**

A lot of the goals of my project changed based on how much time I had to do graphics, so my current project is quite different than what I started out with. For this TP, I focused on more algorithms and cleaning up bugs. In specific, I accomplished the following goals:

* Made a start screen with a button and the graphics for that and made the music play over that part- had to adjust all other variable states for that
* Gave each of the customers names to differentiate them
* Calculates score as a function of the time it takes for you to seat
* Makes customers leave after they finish eating (a few seconds) and allows the user to collect money from the table, creating a new empty table
* Implemented a priority queue to tell the player which table they should be focusing on
* Uses file IO to update the high score by creating a text file that is updated whenever the user quits the game
* Using a weighted system, comes up with a suggestion of a specific food for the user to make based on the priority customer and which item will take the most time using backtracking
* Fixed bugs involving cooking the food (there were lots)
* Implemented the astar algorithm and added a table in the center of the room in the kitchen that the character will always go around. Additionally, this algorithm will always return the shortest path that the player can take to reach another location. Note that we only added one table because otherwise the computational time that would be necessary to avoid all of the tables would be far too long for an actual game
* Figured out how to incorporate the astar algorithm into my game and control the speed of the player using it
* Added the game over state