TP0

Project Name: Overcooked!: The Cook Off

Project Description:

Players will prepare to make burgers by grabbing ingredients, chopping vegetables, and cooking meat. They will then assemble the ingredients to make the burger and serve it on a plate to be given to the customer. Afterwards, they must wash dishes before they can be used again. They will be given orders which will expire after a certain amount of time. There will be three gameplay modes: one player, two players, and competitive mode. The one player and two-player modes will consist of players trying to cook and serve dishes as quickly as possible to gain money which will determine how many stars they receive on that level. The competitive mode consists of one player competing to make more profit than a bot.

Similar projects:

The gameplay of my project will be similar to the actual gameplay of Overcooked. The recipes and actions will be similar. The term project Overcooked2Py! contains an AI chef for the player to compete against which I will be including as well. I will also have a two-player mode like Overcooked With Friends. However, my graphics will look more similar to the term project Litely Overcooked as I want the visuals to match that of the actual game. My project will also incorporate animations regarding the walking and chopping actions like seen in the term project Undercooked.

Structural Plan:

My project will utilize object-oriented programming where there will be different classes for the chef, customer, and ingredients. Each of the different ingredients will be a subclass of the ingredient class. Within each class will contain methods that will outline what can be done with that object or what the object can do. I plan to separate my classes into different files. There will be a main file that will create objects of these classes and code the actual gameplay. The redrawAll function used to draw graphics will contain helper functions to separate each large item that needs to be drawn, such as the chef and the ingredients.

Algorithmic Plan:

The trickiest part of my project will be implementing the AI bot for players to compete against. To determine the most optimal movements and actions of the bot, I will use backtracking. The bot’s movement path to each ingredient should be optimized, with its focus on the order that will expire sooner. This will be done by looping through all possible paths, having the robot make a move, and then using recursion to make the next move. These moves will be undone if that path is not the optimal path. To make it easier for the player to go against, I will have the bot randomly make an nonoptimal move.

Timeline Plan:

TP0: Nov 14

Nov 19 - Have OOP structure set up along with variables and working functions for each class. Basic features of each object should be present and working. Basic graphics (shapes) will be used.

TP1: Nov 20

Nov 23 - Implement 2 player mode (players should be able to collide with each other)

Nov 26 - Improve graphics (using images)

Nov 29 - Implement AI/Competitive mode

TP2: Nov 30

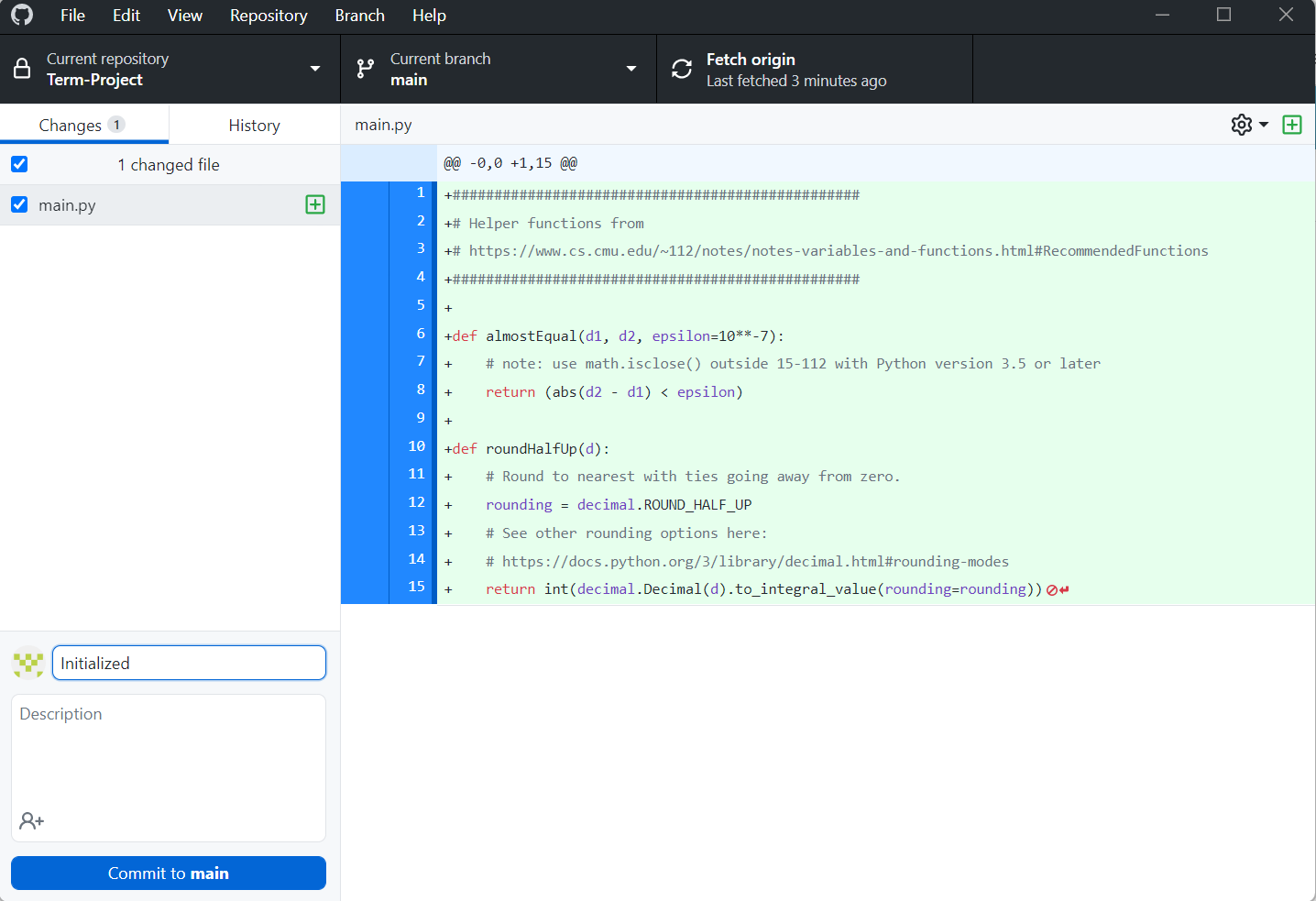
Dec 3 - Implement animations and improve UI

Dec 5 - Video demo + other TP3 write up requirements

TP3: Dec 5

Version Control Plan:

I will be using GitHub to back up my code. I will create a repository that will contain the code and other files my project will utilize and publish the repository online so that it will be accessible if I am unable to access my computer and locally stored files. After modifying any files, I will need to commit my changes and push it to my remote repository.



Module List:

No additional modules planned to be used

TP1 Update

Project will only consist of one player mode and competitive mode. Project will also have rats spawn throughout the game in a random location. These rats will target the food that was last left on the counter, if there is food on the counter, it will move towards the food and steal it unless the player moves the food away.

TP2 Update

The competitive mode will not be implemented. The path planning algorithm for the rats uses the A\* algorithm. Recipes are randomly generated between three different types. The point values for completing an order depends on the type of order. The game ends after a certain amount of time has passed. If the player’s score is greater than half of the max amount of points they could have gotten, they win. Otherwise, they lose.