Name: Lucy Wang
AndrewID: lexinw

Project Description:

The term project is called “112° Cooked” and it would be simulating the game “Overcooked”. The project would be a game where you can control a chef to make food in a kitchen. There would be multiple types of food that you could make and different processes required to make each type of food. Each order passed in to the chef would need to be completed under a certain amount of time, or else the order would be considered as failed.

Similar Projects:

The project would be similar to the original game of “Overcooked”. In the original game, multiple players would each take a role as a chef in the kitchen. Each player would have to collect ingredients, prepare the ingredients, create the food in a certain order, then serve the food. In later levels, players are also required to wash dishes as the customers finish their dishes. There would be a time limit to complete as many orders as possible. Each order would also be required to be completed under a short amount of time. When creating the order, the chefs would have to work together to maximize the efficiency. The number of orders completed, combined with time bonuses for speed in completing the order (tips) would earn points and coins. The points would then be converted to a number of stars. The cooking process changes each level and each station for a certain process is usually scattered across the kitchen, therefore making it harder for players to complete the orders on time. There may also be obstacles and challenges when completing the orders, such as a crosswalk in the middle of the kitchen with pedestrians crossing, the kitchen separated in two halves which each half on a moving truck, as well as the kitchen set on an iceberg, making it more difficult for players to move the chefs. The original game can be single player or multiplayer.

In this term project, most of the cooking aspect of the game would be preserved, such as collecting the ingredients, preparing the ingredients, and serving the food. However, rather than having multiple players control, there would only be one chef in the kitchen. Similar to the original game, there may be obstacles when completing the orders. A potential obstacle design may be that the platform that the kitchen is on may tilt at an angle at some point during the game, therefore making the plates slide along the counter while it is also harder for the player to move their character, both depending on the angle of the tilt and the direction. The time limit for completing each order would also be implemented in this project, as well as a total time limit and the later score calculations. The animations of the chef chopping food, serving the food, and moving around would be simulated as close as possible to the original game.

Structural Plan:

The finalized project would include cmu graphics and would load images of the characters as well as the background of the kitchen. The project would also utilize the keypressed

function (up, down, left and right) to move the character around the kitchen while they are cooking. Certain movements may be assisted by other keys, such as “h” for hold plate or “r” for release plate. In moving the character, the project would include timerFired and a spritestrip (or separate images) to make the character’s image change when they are moving. Further, the timerFired function and a variable timePassed would be used to calculate the time passed between when each order is passed in and is completed (or failed if not completed). Another timePassed variable would be used for the entire level and would stop the game after a certain amount of time. There would be a function for chopping food and it would change the animations of the character while it chops food.

The different classes included in the project may include food, which has subclasses of different types of food that could be made. The food, or ingredients, would behave similarly when being chopped, but the subclasses would include different images. Additionally, there would be a class for orders, which would include the timePassed variable. Another class would be the plates, which would move accordingly to the tilt of the platform that the kitchen is on.

Algorithmic Plan:

The trickiest part of the project would be managing the tilt of the platform and therefore the kitchen. As the kitchen tilts, the plates and the food on them would slide to one side accordingly. Further, it would be harder and/or slower for the chef to move to one side (up the hill) than to the other (down the hill). The speed of the objects moving would need to be calculated based on the angle of the tilt. This may be set by a variable (“left”, “right”, “up”, “down”, or None) and another variable based on the level of tilt. The former would decide which side is harder for the character to move as well as the direction of the sliding of plates and the food. The level of tilt would change the speed of the character and that of the sliding of plates and food, which could be achieved through changing the amount of pixels the character is moving and/or by using the timerDelay function.

Additionally, keeping track of the time of each order would also be algorithmically difficult. Each order object would pass in a different amount of time needed to complete and begin a stopwatch. The order object would also have a boolean to signify whether the order is completed (completed or “True”, failed or “False”, or incomplete or “None”). If the stopwatch exceeds the time limited, the order would be recorded and the state of completion would also be recorded. The completion of the orders and the time of completion would both need to be stored for later score calculations.

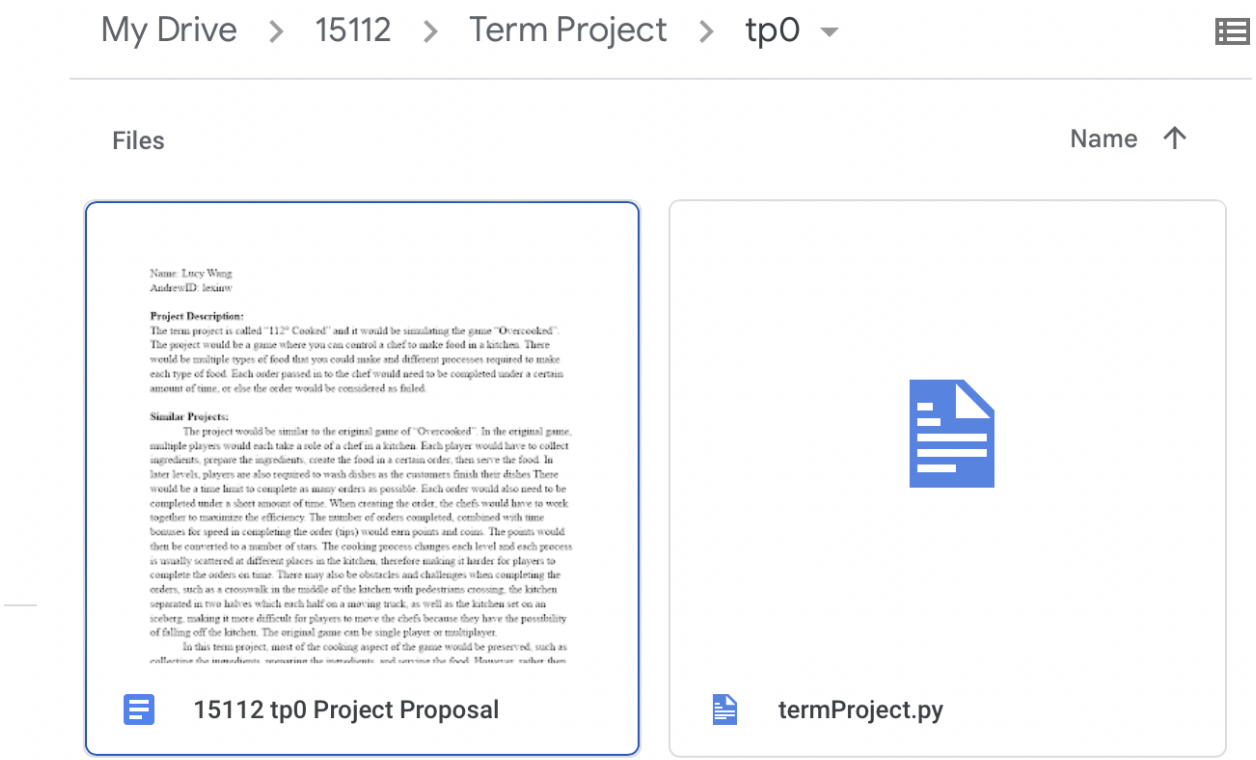
Timeline Plan:

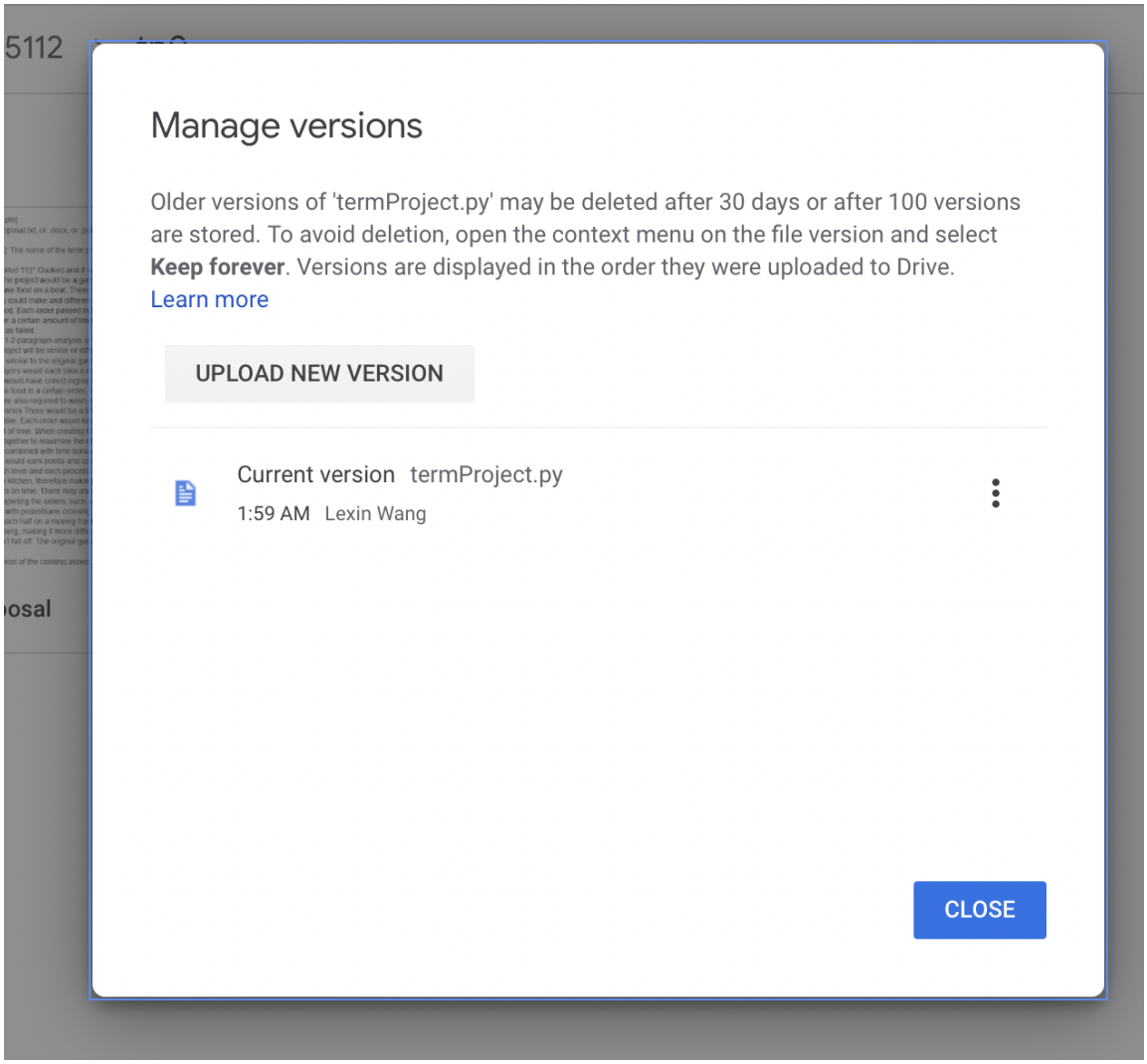
- For tp1 (11/20 5pm):
    - Create the different classes required for the project: order, food, plates and define the basic characteristics of each class
    - Set up the background of the game and draw the characters in the window

    - Set up the spriteStrip of the character and use the keyPressed function to move the character around
    - Complete the chopping function and the respective animations
    - Complete the time tracking aspect of the game and also the time limit feature of each order
- For tp2 (11/30 5pm):
    - Complete the tilting algorithm and adjust the animations accordingly
- For tp3 (12/7 5pm):
    - Possibly include multiplayer feature or incorporate sound and/or music

Version Control Plan:

The version control used to back up code would be Google Drive. All documents and files related to the term project would be stored in the same folder, which then subdivides into folders for tp0, tp1, tp2, and tp3. The design documents would be put in google docs and the different versions of the code would be uploaded onto google drive. The "manage versions" function on google drive is shown in the second image below. After uploading a certain file, the user is able to update the file by uploading a new version of the file. This could be a way of updating code files. Another simpler way of storing each file would be to upload different versions each time as separate files, noting when they were completed or what changes were made.





Module List:
I do not plan on using any additional modules.

TP1 Update:

- I decided to attempt to make the design 2.5D for tp1 and tp2 and potentially change to 3D graphics after MVP.
- I also decided to include AI, which would be mice that use pathfinding to steal ingredients and would have them run around to interfere with the chefs.

TP2 Update:

- I put everything in the kitchen into a 2d list board and created a counter class and counter objects are put into the floor 2d list.
- I made the chef walk in 2.5d.

- The second step of all preparation is putting all the ingredients onto the preparation counter and when all the ingredients are present, the preparation counter would automatically generate the food.

TP3 Update:

- No changes were made to the design of the project.