Kate Zhao
Design Proposal

Project description: I will be recreating Galaga. There will be a player that is controlled by the user and the main objective is to hit as many enemies as you can without getting hit. The player has 3 lives and enemies can have multiple lives depending which type they are.

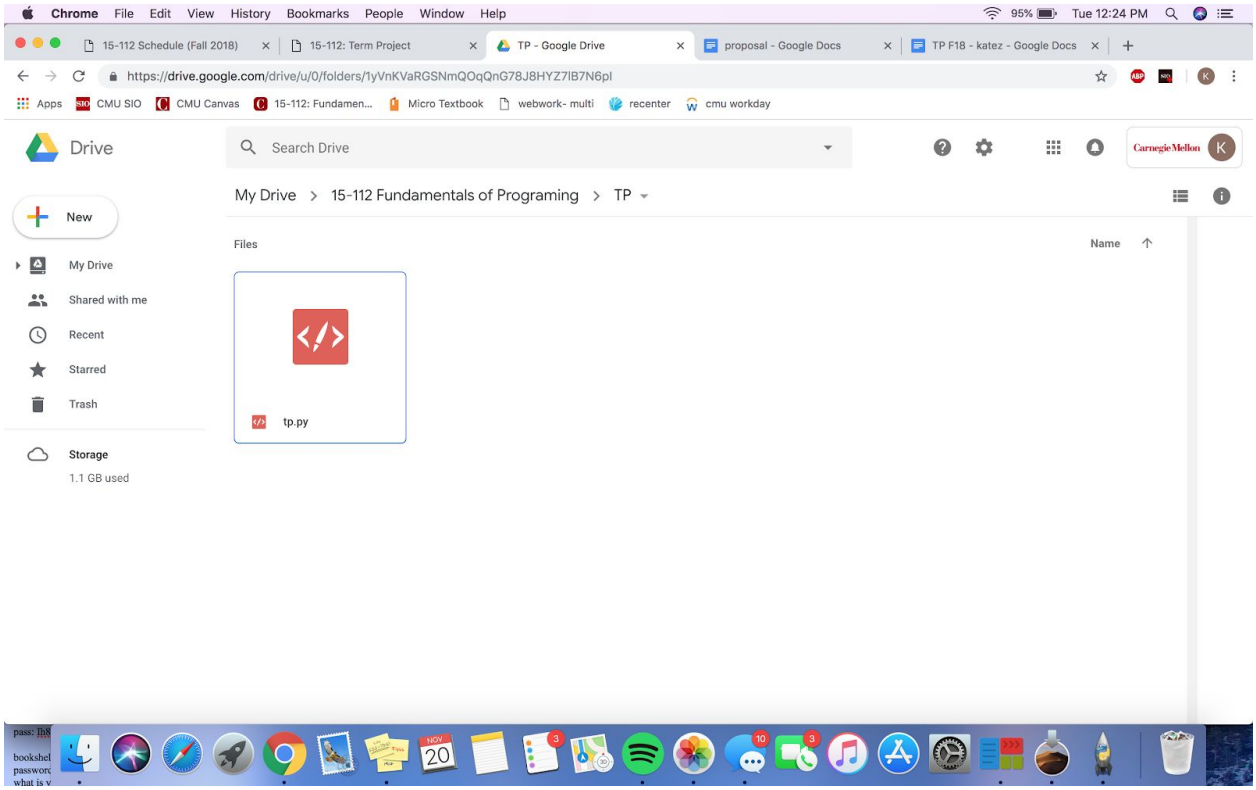
Competitive analysis: For my term project, I will be coding a game very similar to the Japanese arcade game, Galaga. I will be implementing most of the features in the real game, like having different types of enemies shooting at the main player. The player will be able to move left and right along the bottom of the screen shooting at the enemies as well. In the game, there are 255 stages, but I will probably only have one or a couple stages. The enemies have different attributes like the ability to move around the screen, shoot at the player, etc.

Structural plan: I am using OOPy animation. I will have a Player class, an Enemy class, and a Bullet class. There will be multiple enemy classes that inherit from the main Enemy class (ex. groupEnemy, individualEnemy, trackingEnemy). There will be a Bullet class for every type of enemy (ex. groupBullet, individualBullet, trackingBullet). I have an init(), mousePressed(), keyPressed(), redrawAll(), and timerFired(). Most of my code will be written inside the timerFired().

Algorithmic plan: I think the trickiest part of the project for me right now is figuring out how to represent the group and individual enemies. I plan to create a 2D list of True/False boolean variables. I will randomly select a [row][col] or multiple (creating some sort of formation within the 2D list). Looping through the 2D list, I will draw only the ones that are True. Each group of enemy will have a different "timer" attached to them. Every x seconds, they will shoot x bullets. Basically, all the enemies are shooting different amounts at different times so it is not just constant shooting of bullets. Each enemy timer will be based on when the enemy is created.

Timeline plan: By TP1, I plan to have the basic outline of my code done and minimal functionality. I am implementing OOPy animation, so I plan to have all the player, enemy, and bullet classes and have collision functions written up. By TP2, I plan to implement the different types of enemy classes that I wrote for TP1. For example, a tracking enemy (that can track the player), enemies with multiple lives, enemies that can capture the player, etc. I will also have the group enemies be able to move towards the player and around the screen together in a group. By TP3, I want to make it look nicer (adding a background that keeps looping through the image so it looks like it's constantly move; replacing shapes for actual images to represent player and enemies) and add extra features to gain more points for complexity.

Version control plan: I am using my Google Drive to back up my code. I plan to back up my code whenever I make big changes to it.



Module list: I'm not planning on using any modules as of right now. All of my graphics will be done using Tkinter!

TP2 UPDATE

I have not made any changes to my design plans as stated above, but I will be adding a leaderboard feature. In the beginning of the game, I will have a pop-up window that asks users to type in their name. Then, I will save their name and score at the end of the game to a .txt file. After sorting through the file, the endGameScreen will display the top 3 names and corresponding scores.

TP3 UPDATE

Instead of doing a pop-up window, users will input their name directly on the start screen. There will be a leaderboard page where the top 5 scores will displayed. The highest score will be saved to the .txt file with the leaderboard so that the highest score of all time will be printed on the screen, not just within the same game. After user reaches a score of 100, group enemies will start moving across the screen back and forth. Some individual enemies will also become "kamikaze" enemies, where they will track the player and try to hit the player themselves. If the kamikaze enemy hits the player, they both die. I also added some background music.