Project Description [2.5 pts]:

Name: Escape
Player's character will have to try to make it through as many rooms (or levels) without dying. Ghosts will approach the player and each time the player gets caught by a ghost, they lose a life. To kill ghosts, the player must type the proper spell (located above the ghost's head). In order to escape a room and move onto the level, the player must collect enough points (they look like floating orbs).

Competitive Analysis [2.5 pts]:

Escape is based off of the Google doodle game from Halloween 2016. The similarities that my game shares with the google game are that there are ghosts that approach the player and the player loses a life if they get caught, and the goal is to make it through all the levels. However, the games are different: in my game, the player must type the correct phrase to get rid of the ghosts instead of drawing the correct shape like in the google game, the player is also allowed to move to try and escape the ghosts in my game instead of being stationary like in the google game, in order to move onto the next level the player must collect enough points in my game instead of staying alive long enough like in google's game.

Because the player can move around, must avoid obstacles in the map, must collect points, and must avoid ghosts, it is also similar to pacman.

Structural Plan [2.5 pts]:

Will use OOP; a class for the player, each ghost, and obstacles. I was planning on organizing things based on the type it is in animation (so initialization will be broken up, then drawing will be broken up and key pressed will be broken up).

Algorithmic Plan [2.5 pts]: A detailed algorithmic plan for how you will approach the trickiest part of the project. Be sure to clearly highlight which part(s) of your project are algorithmically most complex, and include details of the algorithm(s) you are using in those cases.

Trickiest part is probably the ghosts tracking where the player is so they can move towards the player. I'm planning on using nodes and trees (from the graph mini-lecture). So each possible move of the ghost will be a state/node and each edge will be weighted by the change in distance from the player. Then will have to find the path with the lowest weight to move the ghost along that path.

Timeline Plan [2.5 pts]: A timeline for when you intend to complete the major features of the project.
By Sunday (11/21):

- At least one ghost class (with tracking function)
- Change it from moving from cell to cell to just moving around (only the obstacles will be confined to cells, some ghosts and the player can move anywhere that doesn't have an obstacle)
- Start level generator function and different screens (home screen, instruction screen)

TP2:

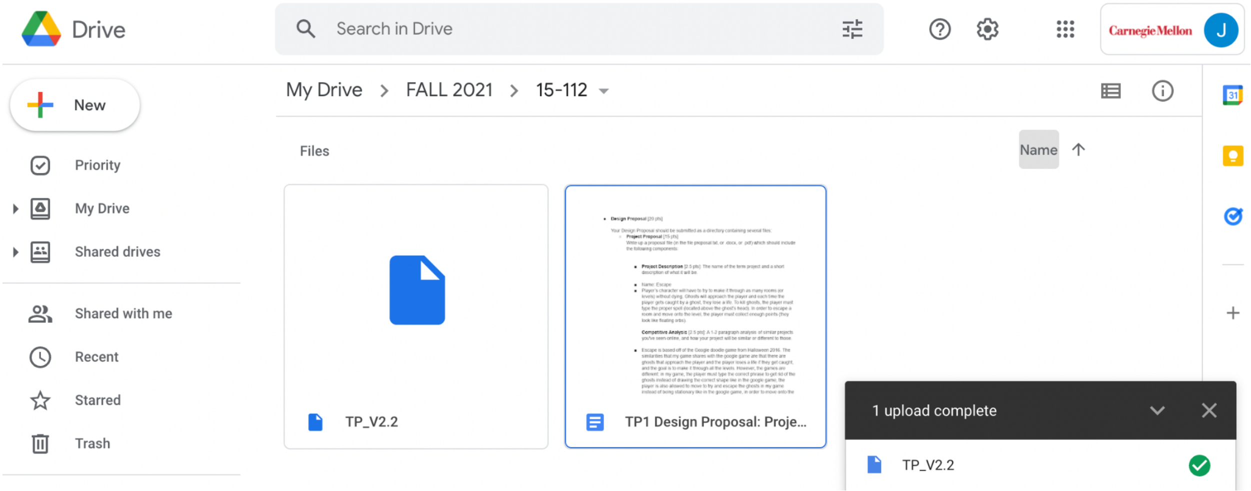
- Finish level generator

- Add at least 1 additional levels (so like how to continue on to a level and make the level progressively harder)
- Add at least 1 more ghost class
- Start animations/sprites/backgrounds and images

TP3:

- Finish all the animations and putting images in
- Add more levels potentially

Version Control Plan [1.5 pts]: Save each TP checkpoint and any other versions I want to a google drive folder.



Module List [1 pts]: I'm not planning on using any additional modules.