Subhei Shaar
CS 162
3/16/15

FINAL PROJECT: REFLECTION / DESIGN

Understanding the Problem

Create a maze game with 3 levels from text files given (FloorA, FloorB, FloorC)

- Use fstream to read in text file and save each character into a grid array in the class Floor
- Analyze the array for characters specific to the game
    - o # = wall
    - o A = ladder to floor above
    - o B = ladder to floor below
    - o C = cherry (set invincibility counter to 20)
    - o D = door (need key to open)
    - o K = key
    - o P = player
    - o G = ghost (AI)
- Allow players to use w,a,s,d to move P about the array
    - o Implement function to get character without pressing enter to create better flow of the game
- Ghost moves toward player every other frame
    - o Uses distance between player and ghost to calculate best path to choose
    - Ghost moves randomly until it is a distance of 4 or less away from the player.
- Restart player to entrance when ghost position is player's position.
- Implement player interactivity with the objects on the level (door, wall, key, etc.).
-Player has a permanent fog of war
    - Only 3 spaces around the player are visible until player discovers more of each level.

Devising a Plan

Class hierarchy

Game class: has levels player, ghost, and floors

Character class (abstract)

Player, Ghost derived from character class

Floor class (holds the levels)

Possible Challenges:

Getting AI of ghost to transfer to different level when player goes through a ladder.

Not deleting ladders after stepping over them.

Path finding from ghost to player

Reflection:

I was able to do what the program asked for with the use the suggested class hierarchy but I did not see where I could have used linked lists and queues. I used an array as the grid which was imported from reading the text files of the 3 given floors. The game class had an array of floors with 3 elements, each containing a map. If the player used a ladder to get into another level the floor array was incremented by 1 and the player teleported to where the ladder position on the new level was located. I had function to scan the array for special characters in the array to make them do different things. Every time a player moved depending on where it was going it had a different set of instructions. These different move outcomes were written in the play function in game in switch case style. For example if you were on a ladder and u press U it will send the player to the next level. However if you were not on a ladder U wouldn't move the player. The ghost would move every other turn the player did to make it not as difficult. The ghost moves toward the player as best as it can without calculating a path between the two. The biggest challenge for me in this assignment was replacing some characters after the player stepped over them and leaving some deleted. Maybe it was the way I designed it but the player P moves around and replaces its previous P on the grid with a space character. This makes it so a bunch of P's aren't copied to the grid.

Testing:

I was constantly testing my game as soon as I could move a player around on the grid. I also printed out some location numbers of the ghost and player when the ghost wasn't properly killing the player when they met. I also had my brother play the game and I received a bunch of feedback to make modifications to the ghost movement pattern. At first I just had the ghost move back and forth in a small area. Then I wrote the algorithm that let the ghost hunt down the player, which was much more fun. This way I didn't have to create 3 separate patterns for the ghost to move in which would have been a pain.