Vlad Predovic

CS 162

Due 3/17/2015

Winter 2015

Assignment #5 Design Doc

* Changes to implement:

Player Sight: Implement a change to how much the player can see:

This change seems straightforward. As I am using a print function accessed directly from the maze class, the class in which my 2d array map is formed, I plan on accessing a modified version of this. To do this I plan on using the current location of the player each turn and feeding it as input into this new function. His location will then be the parameters from which the graph will be displayed 5 spaces in all directions (north south east west) in a box like vision shape for simplicity’s sake.

Ghost Movement:

My ghost movement function is separate from my game function as it is located in its own class. So for this part I have the advantage that I can make my changes for the most part free of the worry of messing up some of my main game code. In order to get the ghost to move randomly I plan on selecting a random number out of 4 (so 0, 1, 2, or 3) and depending on which appears, the ghost will move in one of the four cardinal directions.

In order to account for chasing the player I will create a quick function that calculates the sum of the horizontal and longitudinal differences in spaces between the player and the ghost. To do this I will have to slightly alter the ghost movement function to take in as parameters the current location of the player within the game. Once this is done, I will have movement instructions that will be dictated by this computed distance. If the distance is low enough (5 spaces), then the movement of the ghost will change.

The new movement of the ghost will consist of a few if logical statements as follows:

* Check whether the player is farther away vertically or horizontally.
* If vertically, is he to the left or the right (is his x loc more or less than the ghosts(yours))
* Move in that direction.
* Else repeat the check in the horizontal direction and voila!

Now to implement and hope nothing explodes…..

I began this assignment by examining examples to understand the structure of the game. Since I have no experience designing interactive games, pacman code provided me with a good structure to understand how I would begin manipulating the changes in the structure of the game. I then programmed all the initial classes into a skeleton of the final product. I knew I would need one main class ‘Game’ that would hold all the other classes together and so on. One of the things it really helped me learn was just how critical object oriented programming can be to making coding games easy. I tried to keep my program conceptually as simple as possible in order to reduce the amount of complexity. This was done because I initially assumed I would have a lot of variables since a lot of different situations can occur within the game.

However one my biggest issues I ran into was keeping the amount of variables I had floating around to a minimum. As I implemented more and more changes I began to lose track and had to do quite a bit of cleaning up. I tested my program primarily by using the ostream built in function “cout”.

Being able to output the coordinates of all necessary implementations or characters throughout the project when necessary proved very useful for asserting what were the root causes of bugs. Apart from this, I didn’t really run into many other issues. It is also 10 pm on Tuesday night and this is my last final due holy cow I really want to be done.

THE POSITION OF MY GHOST DOES NOT RESET IF YOU DIE. THIS WAS IMPLEMENTED BECAUSE I THINK IT IS MORE FUN THIS WAY THAN HAVING IT RESET WHEN YOU DIE. YOU HAVE NO IDEA WHERE HE IS BE SCARED HE IS COMING FOR YOU.