|  |  |
| --- | --- |
| Program #1 | Initial Planning Document  James Scott  Colin Riley  Stephen Belden  Shaya Wolf  Neil Carrico  COSC 3011  02/26/2016 |

**Project 1**

First, we created a UML diagram, consisting of 2 classes. This diagram is included in this planning document. The first class is the tile class. This class implements the behavior of the tile game pieces. These pieces are squares with sections of a maze image on the top. Each tile has a location and an orientation. Each of these can be accessed and changed using the methods. The second class is the gameWindow. This holds the game board, the ‘sidelines,’ the buttons and the tiles.

The locations are points on a grid. There are 16 tile locations on the game board. Each tile also has one, and only one, correct space on the game board. This is represented by the Boolean value ‘correctLoc’ and returns True if the tile is in the correct location on the game board. Each tile must also be able to be in any spot on the ‘sidelines’ and therefore be completely off of the game board as well.

The orientation of each tile will be entered as an integer in degrees in the counterclockwise direction. This will use ‘side’ in the rotate function. Tiles are instantiated at zero degrees, which is the tile fully in quadrant I. The ‘correct orientation’ for every tile will be this zero degrees. The other three possibilities are 90 (quadrant II), 180 (quadrant III), and 270 (quadrant IV).

As a group, we were able to use the gameWindow code that was provided and create a game window with three interactive buttons, blue ‘sidelines’ and a game board. The three interactive buttons include:

* “New Game” Button -> Creates a new game with new tiles,
* “Reset” Button --------> Puts the same tiles back at the start position and
* “Quit” Button ----------> Exits the game.

Our next step after this assignment, is to start working on the tile class. We would like to have a clean and efficient tile class to ensure that the overall game is looking good. Our plan is to work on this while we await the next assignment.

**Scope = Next Two Weeks**

After we get our next assignment we will decide our meeting times for sure. However, in general, we are looking at meeting on Friday’s at 3:00. This time is flexible of course and subject to change. These meetings will last until all of our work is complete or until each person can effectively finish the assignment on their own.

We will also be communicating using Slack, given that it has a better notification system then WyoCourses. Further, we would like to suggest that we find a better way to share code with one another. Already, version control is getting out of hand.

Once the next assignment is given out, we will determine if we need to meet twice a week instead of once. We will be adding to this document as the project continues.

**UML**

GameWindow

+ newGame() : void

+ reset() : void

+ quit() : void

- button : button

- grid : tile [] []

Tile

+ setLoc(p : Point) : void

+ getLoc() : Point

+ setOrient(v : int) : void

+ getOrient() : int

- blockPic : Image

- loc : Point

- correctLoc : Boolean

---

- side : (Point, Point)

- orient : int

- correctOrient : Boolean