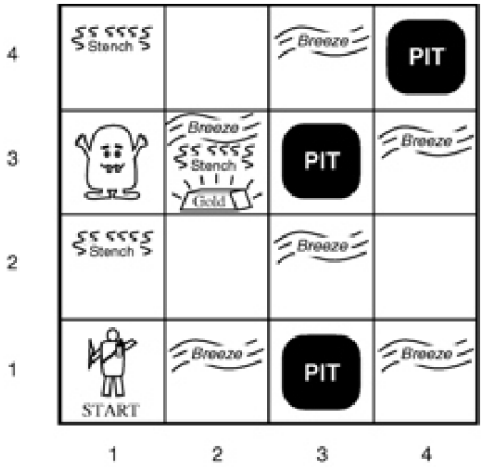
Wednesday, September 26, 2018

CS2261 Project 3 - Wumpus World

Due date: Oct 10th, 2018

Overview:
In this project we will be writing code to allow a user to play a “game” where they explore a 2-dimensional world while trying to find gold and avoid the evil Wumpus.

Description:
In the world of the Wumpus, you are a brave explorer setting out to find gold. This world is in 2D and contains two primary dangers, several pits and a wumpus. Below is a graphical depiction of this world.



In this picture we see the “physics” of this world. That is, that PITs cause surrounding squares to have breezes and the evil Wumpus puts a stench in surrounding squares. The gold on the other hand is only noticeable if you are in the current square with it. We, as the brave adventurer, must explore this grid to find the gold. Unfortunately, our brave adventurer can ONLY see what is in his or her square, so when playing this game, the player will only get told what is in their current square. The player will need to explore this world, only being allowed to go North, South, East or West. As the person can only see their current square, they will need to use the information of breezes and stenches to hopefully avoid the Wumpus and find the gold.

Our Objects:
In this project you will be creating 3 classes (one, a main driver, and the two others that will represent objects that we use).

Square
This class stores one square of the Wumpus world. So it must represent if the square is breezy, has a stench, has a PIT, has a person, has a rumpus or has gold. It should have appropriate getters and setters and some constructor that creates at least a default “empty” square.

WumpusWorld

This class stores the "gameboard" if you will, so stores a single instance of the WumpusWorld game.This class must contain a 2-dimensional array of the Square class. The constructor for this class should create a "random" wumpus world of 4x4 Squares, with 3 pits in random locations, a Wumpus in a random location and ourselves in the bottom left. It should then propagate breezes and stenches around appropriate objects. It should only be able to change Square objects by using the appropriate setters from Square. This class will also need appropriate methods that allow changing of the object state to play the game. It should also contain two different methods for display. One method that would in text display the entire game board (this should be done in text, but it should be laid out in some form of grid). Another display method should simply display the state where the human is located. Note that WumpusWorld should never directly work with Square private attributes but should use getters and setters to do so.

Project3

Our last class is what should allow us to play the game (so it should be the only one with public static void main). It should start by creating an object of Wumpus World, who's constructor should create the game. Then users should be shown the state of the current square that they are in. At which point, they are given a menu and can choose to go N, S, E or W or C(for display cheat). Once they do a move, then using appropriate WumpusWorld methods update the state of the game and then display their current game location. If they moved onto a PIT or a Wumpus, they lose. If they move over the gold, they win. If they try and go off the board, tell them they cannot do that and let them continue to play. If they press C(for cheat) then you should display the entire game board and then let them make another selection. This is so I can actually see if you created the board and if the boards are being updated correctly. Note that project3 should NEVER directly work with a Square object, but should do this only through working with WumpusWorld methods.

Documentation:

For this project I want you to turn in as documentation UML class diagrams of WumpusWorld and Square. You should also, in your README, describe the type of relationship between WumpusWorld and Square and their multiplicity in regards to each other.

Hints:

You should approach this project incrementally. I would suggest starting it as follows:

1) Visualize in your mind how the game should be played. It might be useful at this time to even write on a piece of paper what you think the output of the game would be to get the task laid out in your head.
2) Implement a simple Square, WumpusWorld that has a 2d array of Squares and a Project3 class (so only default constructors), then make sure Project3 can create that default object.
3) Write some preliminary UML's for Square and WumpusWorld. If you aren't sure, go back to step one and think about what each object might have to do.
4) Implement a constructor for Square that takes in arguments and sets it to a particular configuration. Also implement some preliminary display method that displays that square. You can test this in Project by creating a square object and then displaying it.
5) Implement a constructor for WumpusWorld that creates the 2D squares by calling the appropriate Square constructors using the new operator. Test this in main by creating a WumpusWorld object and displaying the board.
6) Only at this point should you start to try and think about creating your game.