
CS231: Project 9

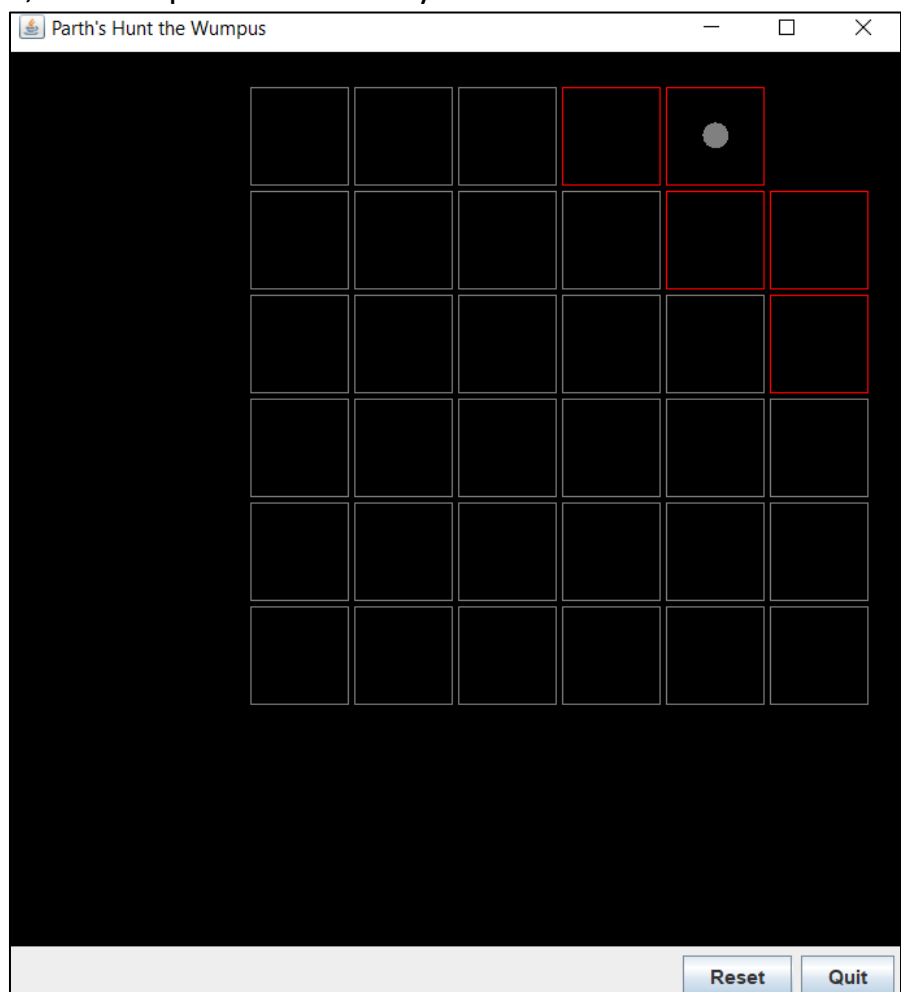
Hunt The Wumpus

Parth Parth | CS231L-A | Dr. Alan Harper

Abstract

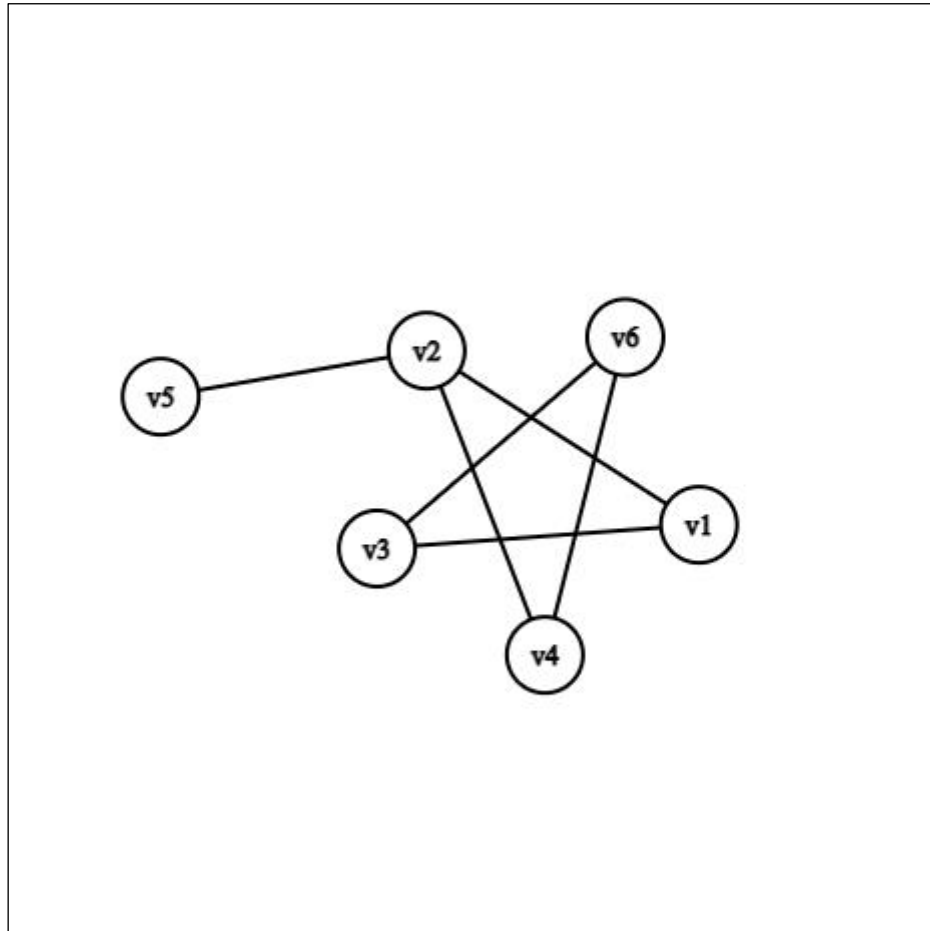
In the project, a graph has been created. A graph is a data structure that contains a finite number of nodes (called vertices) joined by edges which can be uni or bi directional.

The graph has then been used to implement the game called “*Hunt The Wumpus*.” In the game, a “*Wumpus*” is hunted by the “*hunter*.” If the hunter is within two blocks of the Wumpus, the blocks turn red. The hunter is armed with one arrow that, if shot from a room adjacent to the Wumpus in its direction, will kill the Wumpus. If the hunter enters the room where the Wumpus is, the hunter will be eaten. The aim of the game is to figure out where the Wumpus is and kill it with the arrow of the hunter, without getting eaten by it.



Results

The following picture shows the pictorial representation of the graph and the results after running shortest path on it using *Graph.java*.

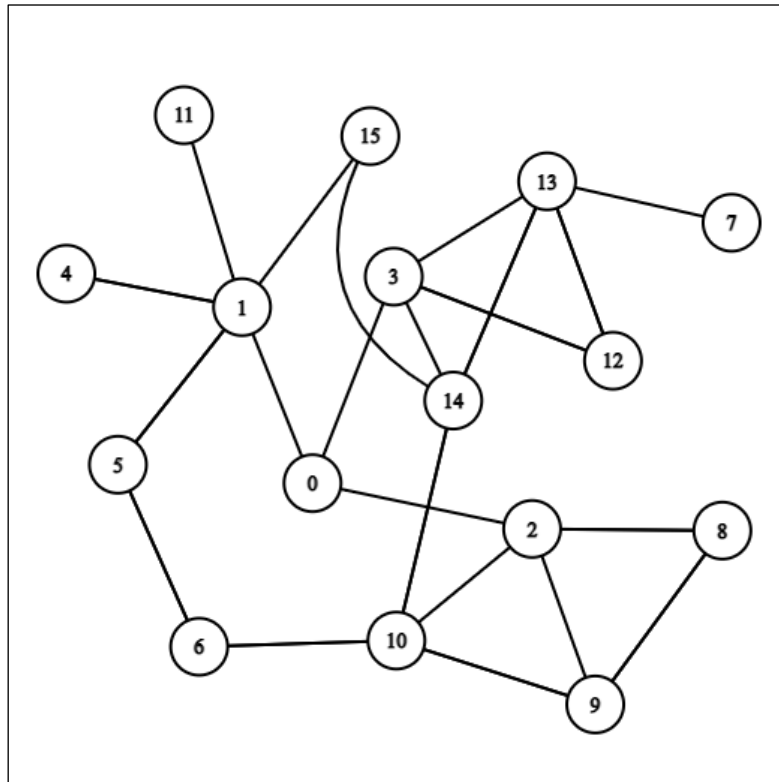


Then, *shortestPath* was called on v3. The screenshot below is pre and post algorithm call.

```
(1, 0) Neighbors: 2 Cost: 0.00 Visited: false
(2, 0) Neighbors: 3 Cost: 0.00 Visited: false
(3, 0) Neighbors: 2 Cost: 0.00 Visited: false
(4, 0) Neighbors: 2 Cost: 0.00 Visited: false
(5, 0) Neighbors: 1 Cost: 0.00 Visited: false
(6, 0) Neighbors: 2 Cost: 0.00 Visited: false

(1, 0) Neighbors: 2 Cost: 2.00 Visited: true
(2, 0) Neighbors: 3 Cost: 3.00 Visited: true
(3, 0) Neighbors: 2 Cost: 0.00 Visited: true
(4, 0) Neighbors: 2 Cost: 5.00 Visited: true
(5, 0) Neighbors: 1 Cost: 6.00 Visited: true
(6, 0) Neighbors: 2 Cost: 3.00 Visited: true
```

The following picture shows the pictorial representation of the graph and the results after running shortest path on it using *GraphTest.java*.



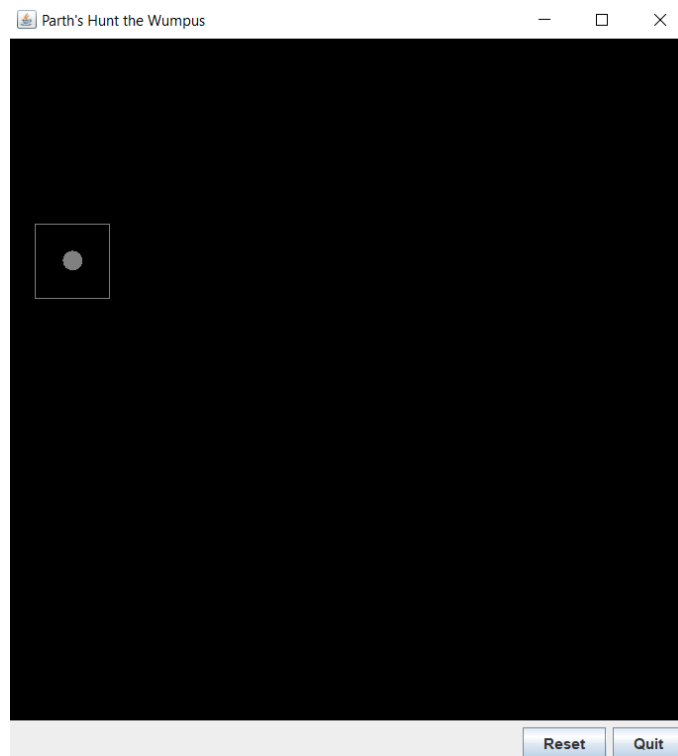
```
Pre-test
0.0 -- 0.0 -- 0.0 -- 0.0
|      |      |      |
0.0 -- 0.0 -- 0.0 -- 0.0
|      |      |      |
0.0 -- 0.0 -- 0.0 -- 0.0
|      |      |      |
0.0 -- 0.0 -- 0.0 -- 0.0

Running on grid[1][1]

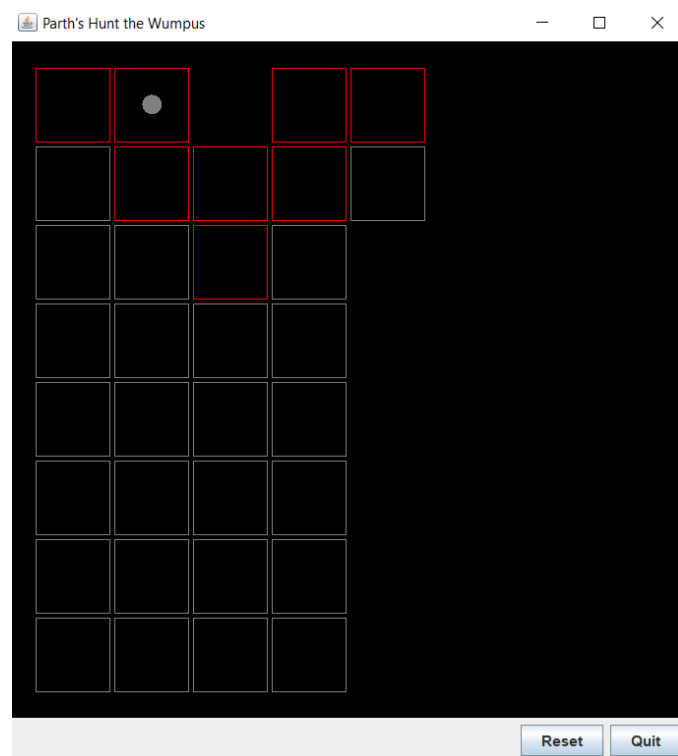
2.0 -- 1.0 -- 4.0 -- 5.0
|      |      |      |
2.41 -- 0.0 -- 1.0 -- 6.82
|      |      |      |
4.0 -- 3.0 -- 2.0 -- 3.82
|      |      |      |
5.0 -- 4.0 -- 3.0 -- 4.0
```

The 0 node in the picture above corresponds to [0][0], 1 node to [0][1], 2 node to [0][2] ... 4 node to [1][0] ... 15 node to [3][3]. All nodes are bi-directional.

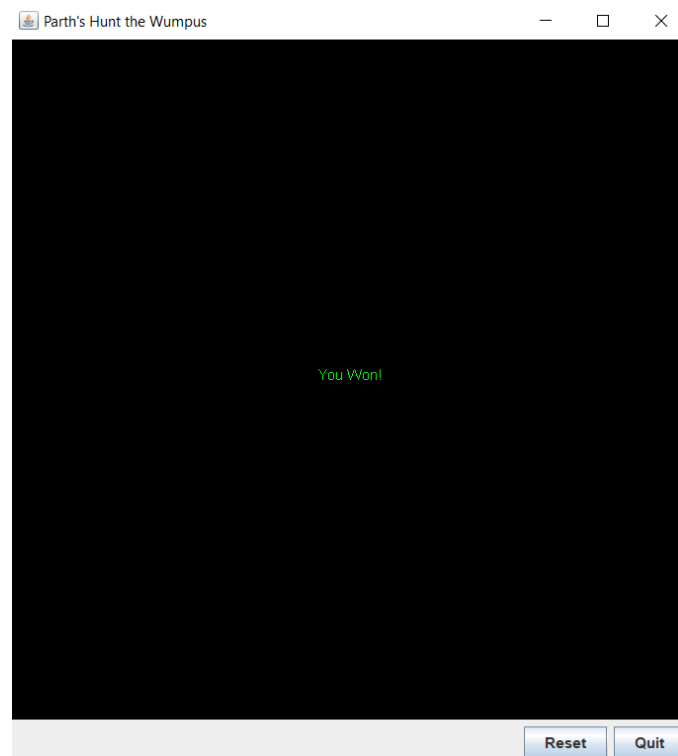
The game starts off like this with the hunter and Wumpus in random positions (such that the hunter is always more than 2 rooms away from the Wumpus):



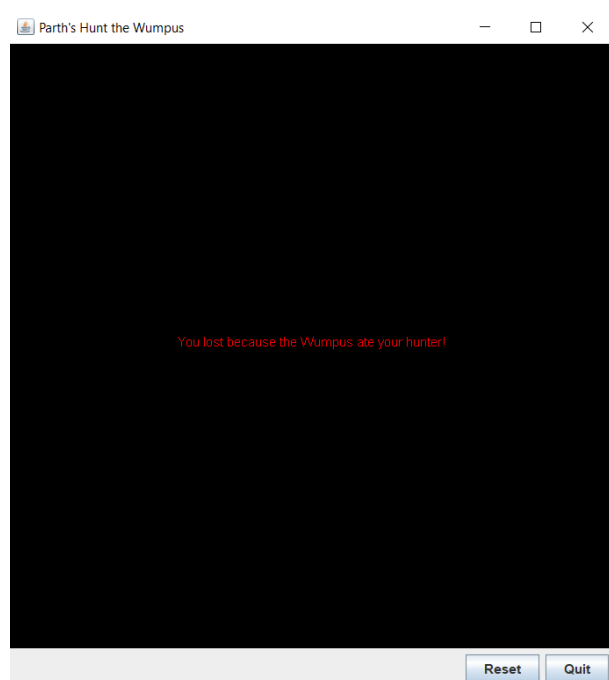
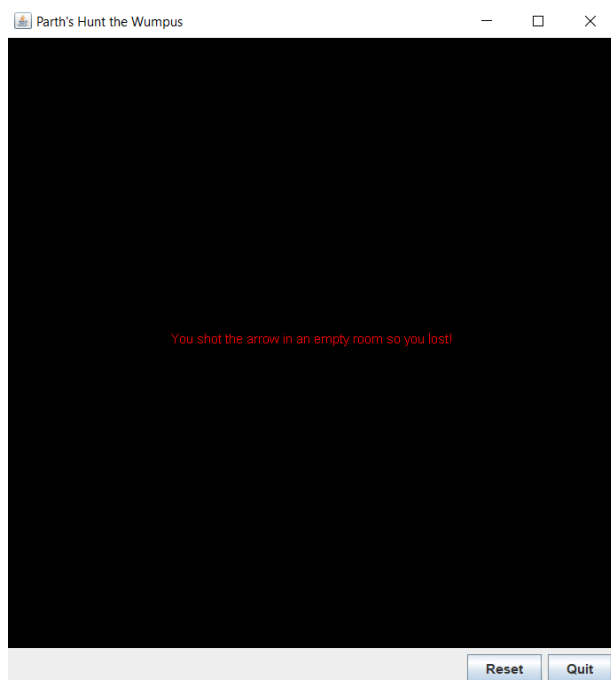
Then, using WASD keys, the hunter can move. If the room is more than two blocks away from the Wumpus, it is gray, if not, it is red:



Then, using IJKL keys, an arrow can be shot. On winning, the following splash screen is shown:



On losing, one of the following screens will appear, depending on the reason of the loss:



Extension 0

For this extension, I have added a reset button to the game. Every time the button is pressed, a new random game is generated.

Extension 1

For this extension, I have added sound to the game. Every time the hunter moves, or the game ends, sounds play (depending on what just happened: win, loss because the hunter got eaten, or loss because hunter shot arrow in an empty room).

References / Acknowledgements

I consulted Prof. Harper for help with understanding how the game was to be implemented. The code for playing the sounds was from [this Stack Overflow](https://stackoverflow.com/questions/6045384/playing-mp3-and-wav-in-java) article (<https://stackoverflow.com/questions/6045384/playing-mp3-and-wav-in-java>). The sounds have been retrieved from <https://www.freesound.org> .