Hunt the Wumpus

The final project was to emulate the game called Hunt the Wumpus, a game where the player is a hunter trapped within a system of caves and the goal is to kill the wumpus before it kills you. To implement the game I used a graph of connected vertices(individual interconnected rooms within the larger cave system), within which a hunter and wumpus agent exist. Additionally certain methods of the game required the usage of other data structures, which are explained later on. Overall there was a positive response regarding the game from friends and peers.

Controls:

	Player 1	Player 2
Movement	↑ - Up ↓ - Down ← - Left → - Right	I - Up K - Down J - Left L - Right
Fire	Spacebar	,

The player moves around the map using the movement controls and once she wants to fire she simply must engage fire and use the regular movement keys to indicate the direction in which she should shoot.

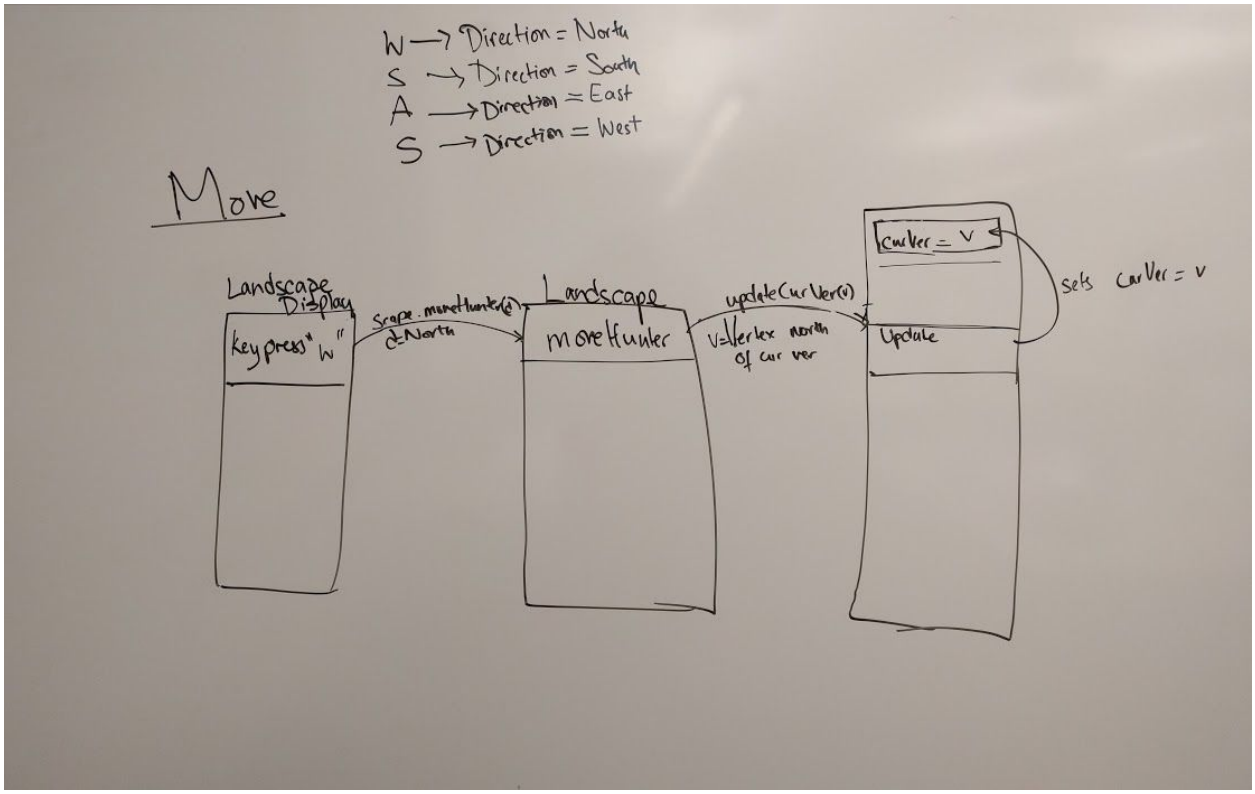
Game Characters:

1. **Hunter:**
   The hunter is a user controlled character which has the ability to move through the vertices of the cave, all the caves are interconnected to any neighbors. The hunter can move four directions on a 2D plane (North, South, East and West), additionally the hunter can also fire his gun with one bullet in the same four directions.
   Extension: As an extension for this project I made the game multiplayer where up to two players can hunt for the wumpus together.
2. **Wumpus:**
   The wumpus is the protagonist of the game, unfortunately within this game he is only stationary. Since the player can only see the sections of the graph they have already explored they are unable to see the wumpus, however they can “smell/hear” the wumpus. This is represented through a red border on vertices that are within 2 vertices of the wumpus.

Gameplay Elements:

The goal of the game is for the hunter to kill the wumpus before it accidentally walks into the same cave(represented as a vertex) as the wumpus. Therefore the main tasks the program needs to handle are hunter movement, hunter firing a gun, checking whether the hunter is dead.

1. **Hunter Movement**:
   The hunter movement is controlled using the updateCur() method of the hunter. Since the hunter's position is determined through a field within the hunter agent which holds a vertex as the hunter's current vertex. Therefore to move the hunter the landscape simply passes a new vertex to the hunter and the updateCur() not only updates the current vertex but also the row and col fields to match the new vertex.

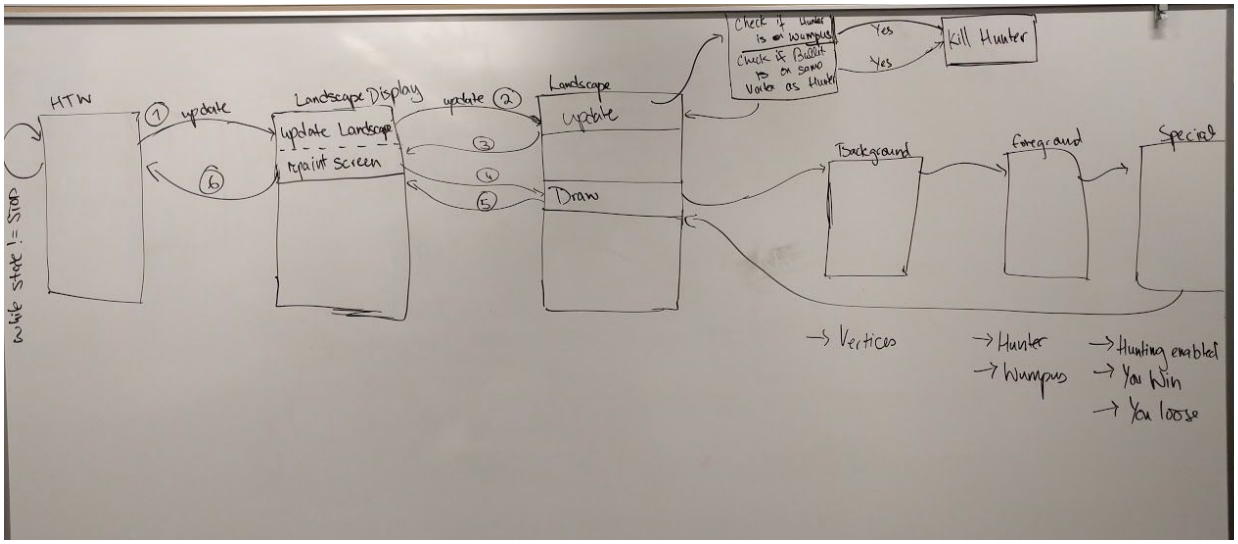


2. **Hunter Life Status:**
   An important part of the game is ensuring the hunter does not get eaten by the wumpus. Thus the landscape's update method constantly checks to see if the hunter has not accidentally walked into the wumpus, or that the bullet fired by the hunter has not missed the wumpus and instead should kill the hunter itself.

   If the hunter walks into the wumpus they die. However if the hunter fires and aims incorrectly then the bullet ricochets off the wall and ends up killing the hunter itself. Killing an agent is simply a matter of updating the agent's life status field.

3. **Firing the Gun:**
   The hunter has a gun with one bullet. The hunter must enable the shoot mode to be shoot the gun. If the bullet does not hit the Wumpus then it should come back and hit the hunter itself and kill the hunter, set life status to false.

Game Loop:



The main game exists within a while loop, the image above shows one cycle within the while loop. The while loop checks to see if the game is in stop mode and until the game doesn't reach stop mode keeps updating the landscape and repainting the display. Continuously repainting the display even while the landscape is in stop mode allows GUI elements to still work after the landscape has been paused. This allows the player to reset the game even after they have won/lost.

Game Video: https://www.youtube.com/watch?v=Ayr2RGHrUlU&feature=youtu.be

Extensions:

1. **Multiplayer:**
   In this extension I allowed a maximum of two players to team up and kill the wumpus, each hunter looks different and has one bullet. Together the game should be easier to play.
2. **Randomly generating levels:**
   Instead of individually creating multiple levels, levels can be randomly generated allowing interesting game play every turn
3. **Difficulty levels:**
   There are three different difficulty levels, 1,2,3 in levels of increasing difficulty. Difficulty is measured through how long you need to travel the map before it you can find the wumpus. Easier game levels are instances where the wumpus is close to you whereas harder levels requires the player to explore the map a little more.
4. **Better Graphics:**
   Instead of having a blank white graph I made the background all black to continue the narrative of being within a system of caves. Additionally the caves are coloured a light grey and connections are white, providing a contrast to the background. Lastly the hunter and wumpus are no longer simple rectangles rather they are images of a hunter and an angry monster if the wumpus wins, otherwise it is an image of a dead wumpus.

5. **Quit and Replay Button:**
   An essential for any game is the ability to open the game once and be able to re-open the game each time you would like to play the game again. Additionally the quit button is just an addition to GUI controls.
6. **Adding Cheat Codes:**
   No game would really be complete without cheat codes, as cheat codes are what ensure you do not give up on the game when it gets too hard. Within this game the cheat code is the character H. The H button reveals the entire map to the user for any duration in which the H key is held down, when H is released the map returns to caves already explored by the player. The H key was chosen as it is likelier to be pressed when 2 player mode is enabled due to its close proximity to the second player's controls.

Special Thanks: Stephanie, Tatsuya, Luis, Jason, Kayla.