How to Play Hunt the Wumpus

> *You have been training for this for as long as you can remember. Finally, as the shadowy caves appear in front of you, a vile stench crawls up your nose. You ready yourself one last time as you prepare to descend into the dark complex of caverns, ready to hunt for the notorious Wumpus that has terrorized the villagers for generations. In exchange for the beast's head, the elders have offered the village's riches as a reward.*
>
> *As you walk, you review the plan once again:*
>
> - *You will move through caves by inputting N, S, E, or W.*
> - *The Wumpus is asleep in one of the 20 caves.*
> - *Each cave has 3 tunnels leading from it to other caves.*
>   - *Two of the caves are bottomless pits: if you enter, you die…*
>   - *Two other caves have super bats: if you enter, they will carry you to another random cave.*
> - *The Wumpus only wakes up if you wander into its lair or if you shoot your weapon and miss your shot.*
>   - *If you manage to wake a Wumpus, there is a 75% chance it moves to an adjacent cave, and a 25% chance it goes back to sleep where it is.*
>   - *If it moves into the same cave you are in, you die…*
> - *In your leather pouch, you have a map, a lantern with a limited amount of oil (starting from 100), and a bow with 5 arrows.*
> - *Each time you may move to an adjacent cave, in one of the 4 standard compass direction (N, S, E, W).*
> - *You can also choose to shoot an arrow into a cave [shoot] to hopefully kill the Wumpus.*
>   - *Moving or shooting will reduce the amount of oil you have (by a random value between 2 and 4 each time).*
> - *You also brought your secret weapon with you: an Environmental Sensor, which allows you to look at the location of the Wumpus, pits, and super bats [cheat].*
>   - *When you are one cave away from the Wumpus or a hazard, it will vibrate and give you a clue what lies ahead but not its direction.*
> - *You win if you successfully shoot and kill the Wumpus.*
> - *You lose if your lantern runs out of oil, or if you run out of arrows, or if you fall into a pit, or if you end up in the same cave as the Wumpus.*
>
> *You take one last deep breath of fresh air as you start to descend down into pitch black darkness… Good luck, you'll need it."*

The above is my introduction to my Hunt the Wumpus game. This will all be stored in a text file called "huntTheWumpusIntro.txt" and is loaded at the start of the game for the player to read. The rules (those points in bullets) will be saved into another text file called "huntTheWumpusRules.txt" and will be used to display help if the player asks for it.

Development outline for Hunt the Wumpus

Outlining the Functionality of the Game Classes

I have decided to use 3 classes in my game: a **Player class**, a **Location class**, and a **Application file**.

The **Player class** will hold the player's information such as their name, the equipment they are carrying (bow, arrow, map, lantern with amount of oil remaining.) The player will be able to look at their equipment at any time. The Player class will be the class that contains the current room the player is in (index of the current room). By having the player in a separate class, it makes it easier for me to add more variables and functions in case I decide to add any.

The **Location class** will hold all of the details of the various locations in the game including a name, exits, if it has a hazard or not. For this project, there will be a total of 4 possible types of location: room with Wumpus, room with pit, room with bats, and empty room. By having the location in a separate class, it makes it easier for me to add more variables and functions in case I decide to add any.

The **Application file** holds the main() function and controls the flow of the game. Using the best practices taught to me will ensure that the game is well-designed and easy to debug and test. How this is structured will be covered in the breakdown below.

The Game Setup

- Load and display an overview of the game rules and introduction so that the player knows what is going to happen and what to do
    - Information will be loaded from a text file (huntTheWumpusIntro.txt)
    - "Press 'c' to continue" will be included after (no clear screen since I will be using XCode)
- Initialise the game variables:
    - Seed randoms
    - Adding the player – ask for the player's name, set default variables
    - Set number of rooms(caves) to 20 using array
    - Set starting lantern oil to 100
    - Set no. of arrows to 5
    - Set 2 random rooms to bottomless pit room and 2 to super bat rooms
    - Set 1 random room to Wumpus's lair
    - Set index of starting room

The Player's turn

- Every turn, the player will have the choice to move or shoot into one of the three adjacent rooms
- Displaying results of moving/shooting
    - Display remaining amount of arrows and oil
    - Check if player shot cave with Wumpus in it → if yes, you win
    - Check if Wumpus moved into same room as player, or lantern ran out of oil, or player ran out of arrows, or player entered room with bottomless pit → game over screen

At any point, player is able to look at location (which exact cave) of Wumpus, pit, and bats by inputting "CHEAT", quit the game inputting "QUIT", look at game rules inputting "HELP".

Processing Player Input and Examples of Player Feedback

There will be two ways which the player's input will be processed:

- Asking for a string input: asking for the player's name, "move" or "shoot", and also for the "CHEAT", "QUIT", and "HELP" functions
    - `string askForString (string question);`
- Asking for a single letter input: 'N', 'S', 'E', 'W' to move from cave to cave, 'a', 'b', 'c' for which room to shoot into
    - `char askForLetter (string question);`

Depending on user input:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
You are now in cave ___. Do you want to [move] or [shoot]?  _
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

- move:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Do you want to move [N] to cave ___, [W] to cave ___, or [S] to cave ___?  _
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

- After moving, display if player entered the Wumpus's lair, or entered room with bottomless pit, or room with super bats.
- Player will be hinted accordingly if danger is nearby:
    - If Wumpus is in adjacent cave:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                  You smell the stench of the Wumpus.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

- If bats are in adjacent cave:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                  You hear the flapping of wings.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

- If pit is in adjacent cave:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                              You feel a chilling breeze.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

- shoot:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Do you want to shoot into [a] cave ___, [b] cave ___, or [c] cave ___?  _
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

    - After shooting, display results (if player killed Wumpus or missed etc)

Based on the player’s response, the game will perform the required action:

- The amount of arrows and oil will be deducted accordingly
- Index of the current room will be updated if player chose to move
- Display locations of Wumpus/bats/pits if “CHEAT” is inputted
- Terminate the game if “QUIT” is inputted
- Display rules from huntTheWumpusRules.txt if “HELP” is inputted

The End Game Conditions

This game only has two end game conditions – first being if the player successfully kills the Wumpus, second being if the player gets eaten/falls into bottomless pit/runs out of lantern oil or arrows

- After every action (moving/shooting) the `checkGameOver()` function is called to check if any of the end game conditions are met.
- If yes, the `gameOver()` is called and it displays the player stats (inventory items left) and the program terminates.
- Asks player if they want to play again