Assignment #1: Getting Started with Your Object

Name: _____

Background:

Hunt the Wumpus is a project that is broken up into a number of different objects that can each be worked on independently. Each of you will "own" one or more of these objects for Hunt the Wumpus. To be able to properly create your object, you will need to know how your object fits into the full game, including how it interacts with the other objects in the game and the tasks that it performs.

Additionally, you will be writing your code in Visual Studio, which may be new to you. You'll need to make sure you are set up to run Visual Studio and know how to create the classes and UI objects you'll be using to build your game.

This assignment will give you the opportunity to understand both how the game works and how your object will do its part. You will also create the file where you'll begin your coding work on Hunt the Wumpus.

Assignment Task Part 1: Scenario Walkthrough

1. Read the Hunt the Wumpus Specification.
2. Read the Object Description for your object.
3. For each of the scenarios in the next page, give up to three sections of the Wumpus Spec that control what your object does in that scenario.
    a. If you are unsure of what your object would do, put that in the list as, "I think my object will…" and make a best guess.
    b. If you don't think your object will do anything for the scenario, put in "Does not apply to my object" and don't put in any sections.
    c. Your lists do not need to be perfect: you will be graded on how much thought you've put into this, not on how correct your lists are.

A partial example for the Player Object:

Scenario	Spec Section	What My Object Does
User is in game, moves forward	Money and Trivia	Increases player's gold inventory by 1.
	Scoring	Adds 1 to the number of turns user has taken.
	User Interface	Gives the score, number of arrows, and gold count to be displayed.

Scenario	Spec Section	What My Object Does
User wants to see High Score		
User wants to start a New Game		
User is in game, moves forward		
User encounters Wumpus		
User encounters bats		
User falls into bottomless pit		
User defeats Wumpus		

Assignment Task Part 2: Creating the start of your object

1. Make sure you've cloned your team's repo to a local folder – get help from your team and mentors if you haven't done this.
2. Open your repo folder inside VSCode
   - Option 1: On the command line, go to that folder and type "`code .`"
   - Option 2: Launch VSCode and then choose File → Open Folder
3. Create a file for your object
   - In the VSCode window, right-click the **`scripts`** folder and choose "New File"
   - Name it after your object, e.g. CaveConnections.js.
4. **FOR EVERY OBJECT EXCEPT GAME CONTROL:**
   Create the beginnings of your object code, with an **`init`** method and a **`test`** method that you will fill in later. Make sure to have a way to turn testing on and off. Something like:

```
import { world } from "@minecraft/server";

export default class CaveConnections {
	static init() {
		world.sendMessage("Initializing CaveConnections");
	}
	static test() {
		if (CaveConnections.testing) {
			world.sendMessage("Testing CaveConnections");
		}
	}
}
CaveConnections.testing = false;
```

   NOTE: Minecraft already has a concept of "Player" so you might need to name your Player class WumpusPlayer.

5. **DIFFERENT STEPS ONLY FOR THE GAME CONTROL OBJECT:**
   Create a file for the GameControl object as described in step 4, but instead of an **`init`** method, you'll create a **`tick`** method. So your object will have **`tick`** and **`test`**. Find the **`gameTick`** method in main.js, and change it to call your **`tick`** method.

```
	static tick(curTick) {
		if (curTick === GameControl.startTick) {
			world.sendMessage("Initializing GameControl");
		}
	}

GameControl.startTick = 100;
```

6. Create a commit with your changes in it, and push the commit to your team's repo. Make sure that your teammates get your changes, and that you get theirs!