## 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 a Visual Studio project to build on for the remainder of your work on Hunt the Wumpus, and get started creating your C# object as well as a form to use for testing your code.

### 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.
   1. 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.
   2. 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.
   3. 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 a Windows Project in C#

1. Open Visual C#
2. Create a new project
   1. Select File à New à Project…
   2. Make sure “Visual C#” is selected on the left
   3. Select “Windows Forms Application”
   4. For the name of your project, use “WumpusTest”
3. You now see a Windows Form
4. Use the Toolbox window to drag a button onto your form
   1. The button will call your object constructor.
   2. After adding the button, right-click it and choose “Properties” to display the item’s properties in the Properties window.
   3. In the Properties window, change the name of the button from “Button1” to something descriptive, like “ConstructorButton”
5. Create a class for your object
   1. In the Solution Explorer window, right-click your project (Should be “WumpusTest”
   2. Hover over “Add” and click “Class”
   3. This will bring up a window that lets you name your new class. Name it after your object, e.g. Cave.cs.
   4. Create a constructor function. For now it should not take any arguments, though you may decide to change that in the future. So for example you might type:  
       public Cave ()  
       {  
       }
6. Now go back to your form. Double click on the button. This will take you to the C# function that will be run when the button is clicked. For example a method called “ConstructorButton\_Click.”
7. Add code in this function to call your object constructor.
   1. Add a global variable in this form whose type is your object class. This will be the object that your form operates on. For example:  
       private Cave \_Cave;
   2. The button click function can then call the constructor and assign the result to the global variable. For example:  
       \_Cave = new Cave();

It is important that you bring your code to class each week ready to exchange with others. If you are going to miss class, email it to the student who is assigned to be your team’s project manager, or pass it to them by flash key on an earlier day. **You will not receive credit for any assignment unless it is in your project manager’s hands!**