Final Summative: The Zork Adventure Game

**Who**

The assignment will be done in partners and you will use GitHub to allow for efficient collaboration.

**The Game**

Your task is to design and implement an adventure game. You have been given a simple framework (Zork1 shell) that lets you walk through a couple of rooms. You can use this as a starting point.

**1 Read the Code**

Reading code is an important skill that you need to practice. You first task is to read some of the existing code and try to understand what it does. By the end of the assignment, you will need to understand most of it.

**2 Make small extensions**

As a little exercise to get warmed up, make some changes to the code. For example:

* change the name of a location to something different.
* change the exits or pick a room that currently is to the west of another room and put it to the north

• add a room (or two, or three…). These and similar exercises should get you familiar with the game.

**3 Design Your Game**

First, you should decide what the goal of your game is. It should be something along the lines of: You have to find some items and take them to a certain room (or a certain person?). Then you can get another item. If you take that to another room, you win.

Once you have chosen a story and a goal you will need to start thinking about some of the essential things you will need in your game:

* What rooms will you need?
* What items would you like to incorporate into your game (if any)?
* Do you need doors that can be closed or locked?
* Are there any other characters in your game other than you?
* What can your character do (you will need to handle these commands)?
* What will make items special – do they have weight, special purpose?

The assignment can be as much or as little as you would like it to be. So you will need to make decisions on what you will want your game to be.

The code that you are given contains the room information within a file. You will want to create a visual map of the rooms so that it is easier for you to create your text file and for debugging purposes.

­

**4 Implement the Game**

Before you create your classes you will need to create a class diagram (UML Diagram) to depict what behavior your classes will have and what relationships you will be between your classes. **Before you begin any development you must have your design approved by me.**

**5 Levels**

The **base functionality** that you have to implement is:

* The game has several locations/rooms.
* The player can walk through the locations. (This was already implemented in the code you were given.)
* There are items in some rooms. Every room can hold any number of items. Some items can be picked up by the player, others can‘t.
* The player can carry some items with him. Every item has a weight. The player can carry items only up to a certain total weight.
* The player can win. There has to be some situation that is recognised as the end of the game where the player is informed that he/she has won.
* Implement a command —back that takes you back to the last room you‘ve been in.
* Add at least eight new commands (in addition to those that were present in the code you got from us).

**Optional tasks**:

* Add characters to your game. Characters are people or animals or monsters œ anything that moves, really. Characters are also in rooms (like the player and the items). Unlike items, characters can move around by themselves. Think aobut how you can use Inheritance or Interfaces to accomplish this task.
* ***Extend*** the parser to recognise three-word commands. You could, for example, have a command

give bread dwarf

to give some bread (which you are carrying) to the dwarf.

* Add a magic transporter room so every time you enter it you are transported to a random room in your game.
* You can invent additional challenge tasks yourself. You have to discuss those with your tutor and get his/her approval before you implement them. Your tutor will advise you if you have picked something that is too difficult or too much work.

**6 Collaboration**

Since this project is to be completed as a team you will need to use GitHub as a repository for sharing your codebase. You will also use the ticket tracking system to allocate the work that needs to be done.

You should create a new GitHub Repository specifically for this project.

Since the GitHub tracks which user commits code and write tickets it will be apparent who completed what work. You will need to ensure that all tickets are assigned to a user and that they contain descriptive information about the tasks and issues.

In addition, you will need to create a wiki as part of the assignment that must have the following sections:

* General Information about the game
* The map for the game
* The UML (Class) Diagrams for the game
* What you learned from doing this project. Both members must include a section on this.
* You’re testing plan.
* Once you have completed all testing you can then create a ticket and assign it to me for marking.
* special features of your game
* known bugs or problems (Note: for a bug in your code that you document yourself, you may not lose many marks œ maybe none, if it is in a challenge task. For bugs that we find that you did not document you will probably lose marks.)

**7 Submission and Assessment**

You will submit your project by creating a ticket in the ticket tracking system and assigning it to me. All code must be professionally written (using industry standards) and will be marked for

* correctness
* appropriate use of language constructs
* style (commenting, indentation, etc.)
* difficulty (extra marks for difficult extensions/tasks)