# Checklist for self-evaluation

Self-evaluation of  Moria Mines (Textual adventure game)

**Section A. Evaluation of the quality of the program code.**

Please write in the box and/or ✓ or X against each of the points in second column below.

***Evaluation of the quality of the code:***

| Criteria | Values | Tick (✓) |
|---|---|---|
| What are your test data?<br>And does the program run on these test data? | | ✓ We have tried to run the game and it worked.<br>There was no breakdown. |
| Did you check that all your identifiers(variables, constants, classes and objects are appropriately | a. Name (e.g variables names are descriptive and should start with lowercase letters and class should start with uppercase etc.)<br>b. Defined/declared (e.g check that variables(local, instance and static) are declared the right place, methods have parameters, return value type and body that reflects the method name)<br>c. Initialized where appropriate value<br>d. Invoked appropriately<br>e. All identifies are used in your program to contribute to fulfill the program specification or have an appropriate role in the | ✓ OK<br>✓ OK<br>✓ OK<br>✓ OK<br>✓ OK |

| | | |
|---|---|---|
| | program<br>f. Access modifiers (private, projected and public)<br>g. Scope and visibility of the identifiers understood | ✓ OK<br>✓ OK |
| Did you modularize your code so it is easy to understand? | a. Are there same pieces of code that are appropriate for method abstraction (redundant code)<br>b. If a method is too long, it may be good idea to think about method modularity using method abstractions | ✓ OK<br>✓ We have tryed not to make the methods too long. |
| Control flow | a. All loop should terminate at some point in the program<br>b. Switch statements should have a default case<br>c. Avoid using multiple exit from a loop. Rethink about your algorithm if you think you need to do this<br>d. Are there too many nested loops/conditions? Rethink about your algorithm if you think you need to do this. | ✓ OK |
| Input/Output | a. Does the program cater for all types of input?<br>b. Are exceptions handles so that the program ends gracefully?<br>c. Does the program run without breaking? | |
| Boolean expressions | a. Are Boolean expression is short and easy to understand with regard to the program logic? | ✓ OK |
| Documentation of the code | a. Is it clear from the comments that what the each segment of code will do?<br>b. Do the codes do what the comments say for each appropriate segment? | ✓ Yes<br>✓ Yes |

| | | |
|---|---|---|
| | c. Do the comments in the beginning of the methods explain what the method will actually perform?<br>d. Do all the declarations(variable, class, methods) have appropriate comments?<br>e. Are critical algorithms explained in plain language? | ✓ Yes<br>✓ We have made some comments in english |
| Program layout | a. Indentation style is consistent.<br>b. Code within a bloc (e.g. inside a loop ) should be indented<br>c. If a block is nested within another block the inner block's body should be indented relative to the enclosing block.<br>d. Avoid excessive “stairstep” indentation. If problem reduce the number of spaces per indentation or switch to vertical style temporarily. | ✓ OK<br>✓ OK<br>✓ OK<br>✓ OK |
| Data encapsulation | a. Proper use of visibility modifiers and getters/setters<br>b. Are local variables are visible only within the declared method, constructor, or block<br>c. Access modifiers can be given for instance variables<br>d. Instance variable are declared private<br>e. Instance variables are decla in a class, but outside method, constructor or a block. | ✓ OK<br>✓ OK<br>✓ OK<br>✓ OK<br>✓ OK |
| Object oriented design | a. Does each class have distinct role e.g. controller class and entity class | ✓ OK |

**Section B. Evaluation against the program requirements.**
Please write small note against each of the requirements below.

| Requirements | Your comments/notes |
|---|---|
| Is you game to read user input from the console and also output text to it ? | |
| Does your show all elements used in the program. i.e.: all attributes, methods including parameters and return types and associations including multiplicity and navigation direction. | |
| Did you create a player class that holds the amount of gold picked up so far | ✓ Yes |
| Did you create a maze of rooms that the player can navigate?  Did you populate it? | ✓ Yes |
| Did you create a room class. Does each room object have a text description and some gold that can be picked up? Does the room have four tunnels? Which one is your starting room? | ✓ Yes |
| How do you keep track of the rooms? | |
| How does your program end? What conditions makes it end of the game? | ✓ A text with "You won" |
| Does the user get a menu of options to choose from once they enter a room? Do you have error-handling based on user input? Are there appropriate message for the wrong input? Do you have a mechanism for user to ask for help? | |