Unit:
Analysis, Design and Implementation

Assignment title:
Text Adventures

September 2015

Important notes

- Please refer to the Assignment Presentation Requirements for advice on how to set out your assignment. These can be found on the NCC Education *Campus*. Click on Policies and Advice in the left-hand menu and look under the Advice section.
- You must read the NCC Education documents 'What is Academic Misconduct? Guidance for Candidates' and 'Avoiding Plagiarism and Collusion: Guidance for Candidates' and ensure that you acknowledge all the sources that you use in your work. These documents are available on *Campus*. Click on Policies and Advice in the left-hand menu and look under the Policies section.
- You **must** complete the **'Statement and Confirmation of Own Work'**. The form is available on *Campus*. Click on Policies and Advice in the left-hand menu and look under the Policies section.
- Please make a note of the recommended word count. You could lose marks if you write 10% more or less than this.
- You must submit a paper copy and digital copy (on disk or similarly acceptable medium). Media containing viruses, or media that cannot be run directly, will result in a fail grade being awarded for this assessment.
- All electronic media will be checked for plagiarism.

Scenario

One of the most popular forms of entertainment software in the early days of the home computing era was the **text adventure game**. A text adventure game involves a player typing in commands that the game would interpret and implement in the game’s story. Text adventure games were often characterised by the **parser** – an input system that took in written instructions and turn them into a change in game state.

Many text adventure games were made up of numerous **rooms**, which the player could explore. Rooms come with **descriptions** that were shown when the player first entered the room. Players could also examine various nouns within the description to get a more detailed view of the area. For example, a room may have a ’flower’ noun defined in it, and players can ‘look flower’ to get some follow-up text such as ‘It’s a rose’. Players could move between rooms by issuing instructions in the form of directions, such as ‘up’ and ‘down’ or ‘north’ and ‘south ’.

Rooms would sometimes contain **items**, and players would be able to collect these items for use later in the world. This was done through the use of a text instruction (known as a command) with an identifier for the object. An example transcript of a short play session of such a game is outlined below. Lines beginning with > indicate a command typed by the player.

```
You are standing in a forest.  It’s full of trees.  There are
two exits:  north and west.
> Look tree
The trees are made of wood and leaves.
> look leaf
The leaf is green.
> get tree
You can’t get the tree.
> west

You are standing by a river.  There is a key here.  There is
one exit:  east.
> get key
You get the key and place it in your pocket.
> get key
There is no ‘key’ here.
> east

You are standing in a forest.  It’s full of trees.  There are
two exits:  north and west.
> Look key
It almost certainly fits a lock somewhere.
> drop key
You drop the key.
> look
You are standing in a forest.  It’s full of trees.  There is
a key here.  There are two exits:  north and west.
```

Tasks continue on next page

Your task for this assessment is to create a system which allows for simple versions of a text adventure game to be created. Your system should include rooms, commands, items, and a player with an inventory.

Your game for this should include at least four rooms, linked to each other through the directions ‘north’, ‘south’, ‘east’ and ‘west’. It should contain at least one item that a player can pick up. For example, a room may contain a ‘key’ or a ‘newspaper’ or a ‘cushion’ – it doesn’t matter what the item may be, just that it is something that can be picked up. The player should begin in a room at random and be able to move from room to room through the issuing of commands in the form of directions.

Task 1 – **26 Marks**

Candidate class list and Diagrams
The candidate class list should incorporate justifications and discussion as to why each class was selected for inclusion, and how its relationship to other classes was derived. The class diagram should show attributes, operations, scope and relationship of classes to each other.

.

Task 2 – **25 Marks**

Activity diagram
The activity diagram should incorporate the classes involved in a playing through the game world. Here, neatness of the flow of logic is important.

Task 3 – **8 Marks**

Use case diagrams
The use case diagram should incorporate each of the user activities indicated in the brief.

Task 4 – **15 Marks**

Code architecture
This involves creating a code architecture that shows an appropriate level of coupling and cohesion, along with the necessary amount of inheritance and encapsulation to express the system.

Task 5 – **26 Marks**

System implementations
This is for implementing the system as described and providing the completed Java code.

Submission requirements

- Your program must be submitted as a zip file of the full project. Whatever IDE you use, it should be possible to open and run the project directly from the extracted archive.
- Diagrams and materials associated with the tasks above should be presented in a word-processed document.
- All references and citations must use the Harvard Style.

Candidate checklist

Please use the following checklist to ensure that your work is ready for submission.

Have you read the NCC Education documents ‘What is Academic Misconduct? Guidance for Candidates’ and ‘Avoiding Plagiarism and Collusion: Guidance for Candidates’ and ensured that you have acknowledged all the sources that you have used in your work? □

Have you completed the ‘Statement and Confirmation of Own Work’ form and attached it to your assignment? **You must do this.** □

Have you ensured that your work has not gone over or under the recommended word count by more than 10%? □

Have you ensured that your work does not contain viruses and can be run directly? □