IJP Assignment 2: Adventure game

November 1, 2010

Introduction

The first (design) task of this assignment is due at 16:00 on Friday, 5th November 2010.

The rest of the assignment (implementation and evaluation tasks) is due at 16:00 on Friday, 3rd December 2010.

The aim of this assignment is to implement a simple adventure game. The exercise will extend your programming experience further, introducing you to:

- the decision-making process inherent in designing and building more complex object-oriented systems;
- a bigger and less guided programming task than you experienced in Assignment 1;
- the development of graphical user interfaces;
- the idea of soliciting and analysing user feedback on your work.

Adventure games

Computer games have been around since the creation of the first computers. Adventure games place the player in an interactive story in an artificial world — possibly fantastical, possibly mundane. Usually there is an ultimate goal, ranging from traditional ones such as "escape" or "rescue the helpless prince from the scary dragon" to more surpring ones such as "die" or "don't lose your mind". There are often puzzles to solve and game characters to interact with: game characters have varying motivations — some are there to help you, others to get in your way, some just don't seem to be there for any reason at all, and all of them might change their role according to your behaviour torward them. You might find the Wikipedia article on Adventure Games an interesting read, particularly since this assignment takes you back to the early days of computing, and such titles as *Colossal Cave Adventure*.

Overview

By the end of the exercise you will have written a simple adventure game with the following features:

- The user can interact with the game through a series of pre-defined commands.
- The user executes these commands through a graphical user interface.
- The execution of commands causes the graphical interface to be updated.

Marking Scheme

Marks will be awarded according to the following scheme. Instructions on how to submit each task are given at the end of this document.

Task 0: Game design	10%
Task 1: Game implementation	75%
Task 2: Game evaluation	15%

Preparation

You will find it helpful to refer to Chapter Seven of *Objects First with Java: A Practical Introduction using BlueJ*. You will also need to download the code on which you will base your work from the IJP course web page. Instructions for importing this code into NetBeans are to be found on the course web page, should you wish to start using NetBeans (a professional develompent environment). The Swing Tutorial (`http://java.sun.com/docs/books/tutorial/uiswing/`) will also be useful.

Sanity Warning: *Don't lose sleep over this exercise! If you are having difficulty, ask your lab demonstrator or lecturer for help.*

Task 0: Game design

[*10 marks*]

The first step in this assignment is to design the game scenario. You should decide on a theme for the game, where the action takes place, and a goal for the player. There are some ideas to help you on page 204 of *Objects First with Java* (Fourth Edition; page 192 of the Third Edition), but you'll probably find it more rewarding to develop your own scenario.

Write a design document describing the following components of the game:

Storyline/theme: What is the game about? Will the user take part in a story?

Goals: What is the user's goal when playing the game?

Who is the target user group?

Setting: Where does the game take place? List and describe the locations. You may draw a map if you prefer (hand drawn sketches are fine). Find a picture to represent each location. You could draw your own picture, or use your own photos if you prefer.

Characters: Does the player encounter any characters during the game? What do they look like? What kind of people are they? You may choose to include your chat bot character from Assignment 1. Find or draw a picture to represent each character.

User Actions/Rules: What actions can the user take in the game world? The "go", "help" and "quit" commands are the basic commands. Are there any other rules? Please specify at least **three** additional commands.

User interface design: The user interacts with the game world through a graphical user interface. Every location should be represented by an image. In rooms which have characters, a picture of the character should also appear. The user should be able to talk to the character. Sketch this interface, and indicate how the interface widgets relate to the actions and rules you have specified.

Task 1: Game Implementation

[*75 marks*]

You will build on the adventure game from Chapter Seven of *Objects First with Java* in this exercise. You should make sure that you document your code using JavaDoc comments. Credit will be given for use of good software engineering principles, including but not limited to high cohesion, low coupling, responsibility-driven design, commenting and documentation.

You may find that you want to revise your design when you start to implement it. It's OK to change your design, but you should note such changes in your documentation.

- Implement the locations you chose for your game. Each location should contain an image which represents it visually. You may build on the `Room` class in the example code. Set up your locations in the `GameEngine` code. [*5 marks*]
- Implement the actions for your game by augmenting the existing `Command` and `CommandWords` classes, and updating the `GameEngine` code. [*15 marks*]

- Implement a class called `Character` to represent the characters in your game. One way to do this is to integrate your chat bot code from Assignment 1. Each character will be confined to a room. A room can contain only one character. The character should have a picture associated with it for display. If a player is in the room with a character, they can interact with the character by typing the command "talk X Y", where X is the name of the character and Y is the remark the player wishes to make (or you may be able to design an easier way to start a conversation). [*10 marks*]

- Implement the user interface which you designed in Task 0. The GUI code should be contained in a class called `UserInterface`. This class should interact with the `GameEngine` class in order to have commands executed. [*45 marks*]

Task 2: Evaluation

[*15 marks*]

There's no point in making a game if nobody plays it! User evaluation is an important part of game development and good software engineering. In this part of the assignment you will gather feedback about your game from three players, and reflect on their comments.

Hand in evaluation forms (you can download them from the course web page) completed by three different players. You can choose anyone to evaluate your game, from any age group and any level of computer literacy. If possible, try to ask members of your target user group to evaluate the game. Hand in a one page report summarizing the results of your evaluation. List the features which you would like to improve, and bugs which you would like to fix.

Plagiarism

Informatics takes plagiarism seriously, and has automatic plagiarism detection systems for code submissions. Penalties apply, and Head of School and Head of College will be notified if plagiarism is detected.

A useful method of learning is discussing and sharing ideas with other students. However, you **MUST NOT** present others' work as your own:

- Don't share code with other students.

- Acknowledge ideas and help you've had from other people.

You must take steps to ensure that your work is not copied by other students:

- Don't give people your code.

- Make sure your directories are read protected.
- Don't leave print outs of your work lying around.

If you are not sure exactly what is meant by plagiarism, see: `http://www.inf.ed.ac.uk/teaching/years/msc/courseguide10.html#plag`

Submission

You should submit paper copies of your game design and evaluation report to the Informatics Teaching Office. Include the game evaluation forms for Task 2. Please write your name, matriculation number and the course name (IJP) on any paper you hand in.

These instructions assume that your project for this assignment is in the folder `IJPAssignment2/` contained in the directory `/home/sXXXXXXX/IJPAssignments/` (where `XXXXXXX` is your matriculation number). You'll need to adjust them appropriately if you've kept the assignment somewhere else.

You will submit the folder `IJPAssignment2/`. Make sure that this folder (or one of its subfolders) contains any image files you used in your game.

Go to the folder `/home/sXXXXXXX/IJPAssignments/` by issuing the following command at a DICE shell prompt:

```
cd /home/sXXXXXXX/IJPAssignments/
```

Now, create an archive of the `IJPAssignment2` folder (and all of its subfolders) by typing

```
tar cjvf IJPAssignment2-complete.tar.bz2 IJPAssignment2
```

You will use the `submit` command to send us this archive. At the DICE shell prompt, type

```
submit msc ijp 2b IJPAssignment2-complete.tar.bz2
```

You will be asked to confirm that you wish to submit your work. Once you say yes, you should see a message that your submission was successful. Ask a lab demonstrator or lecturer for help if you foresee having trouble with the submission process.

You can submit your work more than once. Each time you do so, *the previous version will be overwritten.* **Only the final submission will be marked.**