



## Unit: Analysis, Design and Implementation

### Assignment title: Board Games

December 2016

#### 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

A difficulty associated with playing board games is organisation. It can be hard to get enough people together in a room at the right time. Moreover, even if you can get a group together with the right number of people, there can be difficulties regarding the preferences people have. On one hand, some people may want to play a single, long strategy game that can be played with four but works best with six people. On the other hand, some may want to instead play several small games where the optimal number of players is four and they will not deviate from what is 'most fun'.

A local board game club has asked you to develop a software solution to deal with this problem by allowing people to find compatible groups in the local area. The idea is to allow people to register an account that stores their availability in terms of specific weekly time slots, their location, and the category of games they would be happy to play. Each account should also permit users to record preferences for specific game preferences, in addition to categories of games.

For an example of the kind of games people play and game categories, please see: <http://boardgamegeek.com/>

The minimum set of data stored for each player is as follows:

1. Name
2. Email address
3. Location
4. Preferred categories
5. Preferred games
6. Preferred designer
7. Available time slots
8. True or false – will only play optimal numbers

In addition to user accounts, the board game club will also maintain a central database of games, which is broken down into the following data elements:

1. Name of game
2. Game designer
3. Minimum players
4. Maximum players
5. Optimal players
6. Category of game

Upon registering on the site, a user can choose both a time-slot and a number of players, and also generate a list of compatible players available in that slot. Compatibility is defined as all possible players meeting the following criteria:

1. All players are in the same location
2. All players share a game style, game designer or game name preference

The tool will also output a list of compatible games based on a user's preferred categories, designers, and games chosen by compatible players. While not all players will be willing to

play all games, clicking through this list of games should show a list of players that are linked to the game title.

Upon receiving a list of players in the area, the player doing the search can choose a specific game, and then a number of players based on the minimum and maximum players for that game. Upon a press of a button, it will then generate a suggested group of players that might be interested in playing, along with their email addresses. Having generated a group, the player should be able to bookmark it for later browsing. If the preferences of players in a bookmarked list changes, it should be updated when the player next views it.

Your application then needs to provide the following functionality:

- Allows for users to register themselves and edit their profile.
- Allows for new games to be added and edited.
- Allows users to search by location and timeslot.
- Generates an appropriate list of players in an area based on game compatibility.
- Generates a candidate list of players based on gaming preferences.
- Allows the bookmarking of generated groups.
- Allows a player to look at previously bookmarked.
- Updating bookmarked lists every time the player views it.
- Saves the entire state of the current wiki in a suitable data format.

Your solution will consist of a class diagram, a use-case diagram, and an activity diagram for the process of generating a list of gamers based on timeslot, location, game choice and player number preferences. Moreover, you should also submit the completed program code in Java.

## **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 user generating a list of gamers based on games and player numbers. 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.

**Tasks continue on next page**

## Task 4 – 15 Marks

### Code architecture

This involves 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? ☐