CO2001 – Mini-project

Assessment Information

Assessment	**Mini-project (Sit)/Resit**
Total available marks	**100 (55% of Final Mark)**
Number of Parts	**4**
Submission deadline	**Sunday 9 December, 23:59**

This is an **individual** project; however you have **an option to do it in a group of two**. If you are doing in a group, then only one submission is required from either one of the group members.

Learning Outcomes

This assessment is aimed at assessing all the following module learning outcomes:

- Develop a GUI (Graphic User Interface) making use of basic HCI concepts.
- Demonstrate understanding of Object Oriented Concept and general programming construct.

Overview

This mini-project aims at developing a Java Desktop Application to simulate Dice Throwing Game; a simple game for 2 players, where they take turn to each roll a dice once, and score points according to the results of the dice rolls. The winner is the one who accumulates a pre-defined maximum score first.

Part 1 (20 Marks)

Welcome Simulation of the Dice Roll:

The game begins with an appropriate well-designed welcome message.

Dice Roll:

Your application simulates the dice roll operation for both players. Computer generates 3 random numbers between 1-6 (simulating a 6 sided dice), for each player.

The players' scores are updated every time dice roll button clicked for each player. The scoring rules for each round is as follow:

- If it generates a pair, then the player score is the sum of that pair. For example if it generates 3 & 3 & 1, the player scores will be the sum of 3 & 3 which is 6.
- if all the 3 dice rolls have the same values, then the score will be 18.
- Otherwise, for any other combination, they just get 1 score.
- if both players reaches a score which is more than the pre-defined maximum, the game's result is a **Draw**
- a player is considered a **winner** if he/she accumulates a score which is more than the predefined maximum.

At every state of the game, both players' score must be displayed together with the name of the player who is leading the game.

Program Design:

Your program design should include at least the following classes: player, game and dice or any other similar classes.

Part 2 (20 Marks)

Basic Graphical User Interface

Using JavaFX, develop a basic GUI to simulate the above listed functionalities.

Part 3 (40 Marks)

Advanced Graphical User Interface

- Your application should display the state of each player (the scores and highlights if they are heading) every time the dice is rolled.
- Your application should have the proper user instruction to explain the game and its rules; such as a short clip or animation

- Your application needs to have at least 3 different layout managers.

- You are encouraged to propose and implement new features different from the above listed items including extra visual features, animation. You can gather ideas by looking at similar applications. You are expected to consider the usability principles and produce an application that takes good care of them.

- Your source code should be well organised and documented.

Part 4 (20 Marks)

A document discussing the importance of usability principles in designing software user interfaces and how these principles were approached in the development of your software application. Therefore, choose and discuss any **three** design principles from the following link that apply to your application.

https://tfa.stanford.edu/download/TenUsabilityHeuristics.pdf

How to submit

Submit a single zipped file named as (yourUserID_CO2001_Miniproject.zip) or if you are doing it in a group, then name your zipped file as (yourGroupNumber_CO2001_Miniproject.zip) through Blackboard by the specified deadline. Your zipped file should contain the followings:

1. A JavaFX Desktop Software (Java source code + file resources needed + executable jar) implementing Part 1, 2, and 3.

2. A Discussion document (in PDF form) on how you have used basic HCI concepts in your GUI development.

Provided Resources

- https://www.tutorialspoint.com/javafx/index.htm
- http://www.playonlinedicegames.com/

Marking Scheme

The 100 marks of this coursework are distributed as follows:

- Part 1: 20 marks
- Part 2: 20 marks
- Part 3: 40 marks
- Part 4: 20 marks

The weightings of the assessed learning outcomes are given above. The following is a set of guidelines for the marking:

- First (70 to 100 marks): Excellent to Professional level Application/Document. Software functionality is complete and even additional features listed above have been implemented. The code and system structure are such that the system will be highly maintainable. Object-oriented design quality. This will be assessed on appropriate implementation of classes, fields, constructors, methods .The GUI shows excellent use of HCI and Usability techniques; it consists of various layout managers, controls and animation in different parts of the application as described in part 3. The document is rich in discussion and references to the software developed with a presentation and organisation to high standards.

- Second Upper (60 to 69 marks): Very Good Application/Document. Software functionality is complete with no extra features. The code and system structure are such that the system is maintainable. Object-oriented design quality. This will be assessed on appropriate implementation of classes, fields, constructors, methods. The GUI consists of 1-2 different layout managers, limited number of controls. The document presents a good discussion and some references to the software developed with a good presentation and organisation.

- Second Lower (50 to 59 marks): Good Application/Document. Most of the software functionality has been achieved. Object-oriented design quality. This will be assessed on appropriate implementation of classes, fields, constructors, methods. The GUI consists of a minimal number of controls. The document presents an average discussion and references to the software developed with an average presentation and organisation.

- Third (40 to 49 marks): Average Application/Document. The basic functionalities of the software are working but perhaps without proper use of OO technique and exceptions and no proper use of GUI. The document

presents some discussion and some references to the software developed with a minimum presentation and organisation.

- Fail (< 39 marks): Minimal Application/Document. There is an attempt to implement part of the software functionality but with big gaps in most of the key elements such as OO Technique. There is no GUI developed. The document presents a brief or no discussion with minimal or no references to the software developed.

- No-submission: A mark of 0 will be awarded.