GRIFFITH COLLEGE DUBLIN

COMPUTING ASSIGNMENT TITLE SHEET

Course:	HDC
Stage/Year:	1
Module:	HCI & GUI
Semester:	[Semester II]
Assignment Number:	2
Date of Title Issue:	Thursday 18th. April 2019
Assignment Deadline:	Wednesday 08th. May 2019
Assignment Submission:	Upload using Moodle link provided on HCI-GUI page
Assignment Weighting:	25/50

Assignment Title

Please state the assignment title / brief. Please specify details such as:

'CardsHiLoGUI' Card game application. This is introduced on page 2.

Learning Outcomes

Please state the programme and related module learning outcomes that this assignment is assessing.

1,2,3,4,5,6

Assessment Criteria

Criteria Include:
Application correctness. Originality of solution. Presentation, including adherence to the required file formats. Commenting and layout of code. Adherence to accepted GUI standards. Correct treatment of sources and general academic approach.

HCI & GUI Programming
Assignment 2: CardsHiLoGUI Card Game Application

HiLo is a card game where a card is dealt and the player has to guess whether the next card that is dealt will be higher or lower than the first card. In the application shown (Figure 1), a card is first dealt on the left hand side. In this case, the card is the King of Clubs. Then, a guess as to whether the next card dealt will be higher or lower is indicated, using the radio buttons in the centre. On clicking the 'Deal Second Card' button, another card is dealt on the right hand side. If the card dealt is in accordance with the guess of the user, then the user wins. Five consecutive, correct guesses constitute an overall game win. A new game is begun using the menu by selecting the 'New Game' item. Note that for the purpose of comparing cards, only the face of the card matters. The suit, i.e, whether Hearts, Spades, Clubs, etc., is unimportant. A second card of equal face value is considered a lose for the purposes of this game.

Figure 1: CardsHiLoGUI main interface

(a) Create a project called 'CardsHiLoGUI'. Create a main interface for the application similar to that shown in Figure 1. Remember to create a ToggleGroup for the radio buttons and set the ToggleGroup for each button to ensure mutual exclusivity. At application startup, the card images are populated with default cards. Use comments in your code to explain your approach in relation to the GUI construction.

(5 marks)

(b) Add a menu bar and 'File' menu. The 'File' menu should contain the items 'New Game', 'Shuffle' and 'Exit'. Implement these menu items fully. Add a 'Help' menu with an 'About' item to show your student name and number in a dialog.

(5 marks)

(c) Create classes for Card and DeckOfCards to help manage the game. The Card class should include methods such as isEqual( ), isLower( ), isHigher( ) and toString( ). The class DeckOfCards should include methods such as dealTopCard( ), isEmpty( ) and shuffle( ). Provide one or more constructors as appropriate for each class.

(5 marks)

(d) Implement dealing of each card, determination of a win/lose and user feedback in a label as shown in Figure 1. Store wins and inform the user of an overall game win on achieving 5 consecutive, correct guesses.

(5 marks)

(e) Add a progress bar and progress indicator to indicate progress towards 5 consecutive wins leading to an overall game win. Provide appropriate feedback on achieving an overall game win. Use a stylesheet to apply an attractive look and feel to the application.

(5 marks)
(Total 25 marks)

Assignment Weighting:
This is the second of two assignments in this module. This assignment has a weighting of 25 marks out of a total of 50 marks for both assignments.

Deliverables:

- 1 signed and dated assignment submission sheet. Save the assignment submission sheet as a .pdf file and include it in the zipped archive with the project.

- 1 zipped archive file containing the completed project created with Java FX, to be uploaded using the Moodle link provided for Assignment 1. Please note that only .zip files are acceptable. Any other format can not be opened/corrected and will result in a mark of 0. Please be warned that files created in any other format, or using tools or tool versions other than those stipulated can not and will not be corrected. As an aspiring software professional, you will understand that multiple versions of software can not be supported in a task such as this. This is non negotiable. If you supply a file that can not be corrected you will receive a mark of 0.

File Name
The zipped archive should be named with your student number and name eg:

123456_johnsmith_assign1.zip

Due Date
Upload the assignment before Wednesday 08th. May 2019 11:55pm.

Keep a Backup Copy
Ensure that all necessary components are present in the zipped archive. The assignment is submitted electronically. It is extremely important that you keep a backup copy. As a student of Computer Science, you can not easily defend the position that you did not understand the necessity to keep a backup copy.

Sources

In line with good academic practice, you must carefully acknowledge all sources including any material sourced from other students. Unreferenced material pasted from websites or where the source is unclear, is of little academic value. As a rough guide, if you include material that takes an alternative approach to that taught in lectures/labs, you must show in your comments that you understand the alternative approach thoroughly. Further, you must fully reference the source(s) of such work so that examiners may assess the value of your contribution. You may be required to explain/present your code in class as part of the work. For that reason it is important that you understand what you present.

Marking & Feedback

Detailed marks and extensive feedback will be issued in due course. Once this process is complete, please refrain from asking for further feedback, explanation or clarification. As is standard practice, marks are non-negotiable and are a matter only for the presiding academic in the module, overseen by external examiners.