COMP2396 Object-oriented programming and Java
Assignment 3: GUI Card Game

Due date: 21ˢᵗ April 2023 (Friday), 23:59

This assignment tests your understanding of GUI programming in Java and your programming skills You should develop the program using Eclipse and submit an Eclipse project to Moodle.

In this assignment, you are going to implement a simple card game. The computer will simulate the dealer. When the game starts, 52 playing cards will be shuffled and the player is given $100. At each round of the game, both the player and the dealer will be given 3 cards (they are drawn from the top of the deck). Player will place his/her bet (you can assume the player always place a positive integer bet). Then before the dealer opens the cards, the player has a chance to draw AT MOST two cards from the top of the deck to replace any two of the cards on hand and each card on hand can only be replaced ONCE. Player will lose the bet if the dealer got a better hand (see below for explanation), otherwise the player wins the same amount of money as his/her bet.
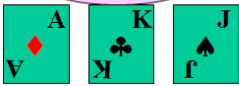
Rules to determine who has better cards:
J, Q, K are regarded as special cards.
Rule 1: The one with more special cards wins.
Rule 2: If both have the same number of special cards, add the face values of the other card(s) and take the remainder after dividing the sum by 10. The one with a bigger remainder wins. (Note: Ace = 1).
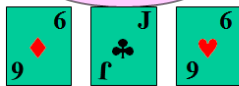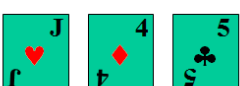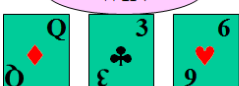Rule 3: The dealer wins if both rule 1 and rule 2 cannot distinguish the winner.

User interface components:

Your user interface should contain the following components:

1) 6 JLabel components for holding 6 ImageIcon components. These ImageIcon components are used to display the cards of the dealer and the player. The image files can be downloaded from Moodle.

2) 3 JButton components for the player to replace card 1, card 2 and card 3 respectively. Note that a player can replace at most 2 cards on hand and each card can only be replaced ONCE (Note: 1) the JButton for replacing the corresponding card should be disabled if that card has already been replaced; 2) After 2 cards have already been replaced, the 3 JButton should be disabled).

3) 1 JLabel component showing the string "Bet: $" to guide the player to input his/her bet.

4) 1 JTextField component for the player to input his/her bet (in the form of a positive integer).

5) 2 JButton components for the player to start the game (i.e. drawing 3 cards from the dealer's pack) and to evaluate the result.
(Note: The JButtons for evaluating the result, replacing the cards should be disabled before the player indicates his/her bet and start the game, they should only be enabled after the player indicates his/her bet and start the game; The JButton for starting the game should be disabled after the player places his/her bet and starts the game, it should only be enabled after the result is displayed, evaluated and the player still has money.)

6) 2 JLabel components for displaying important messages and the remaining budget that the player has. They should be placed below the JLabel component showing the string "Bet: $", the JTextField component for the player to input his/her bet and the 2 JButton components for the player to start the game and evaluate the result.

7) 1 JMenuBar contains the following:
-1 JMenu "Control" contains 1 JMenuItem to quit the game.

A sample user interface is shown on the next page. You can have your own design but you must include the above mentioned GUI components and all the functionalities described in this document should be implemented. To ensure your program has implemented all necessary functions, please refer to the screen capture and the marking scheme below for your reference.

Initial setting:
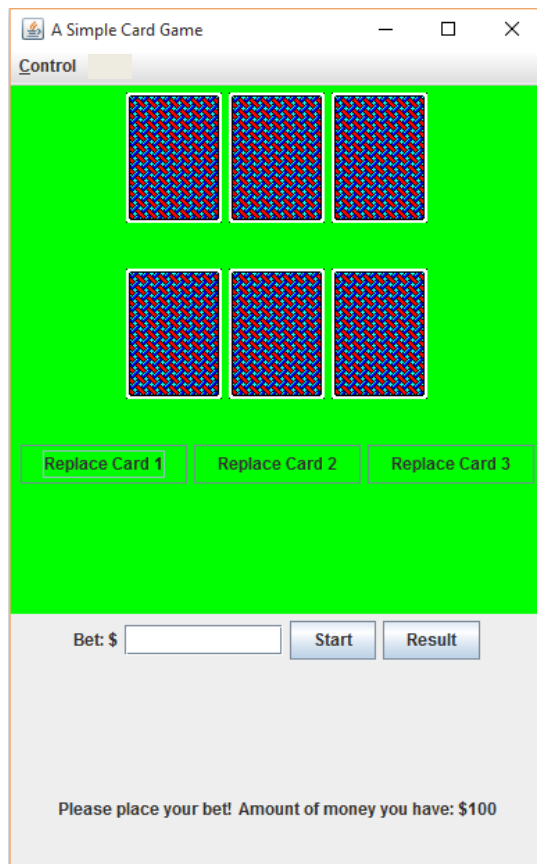
After player places his/her bet and click Start button:
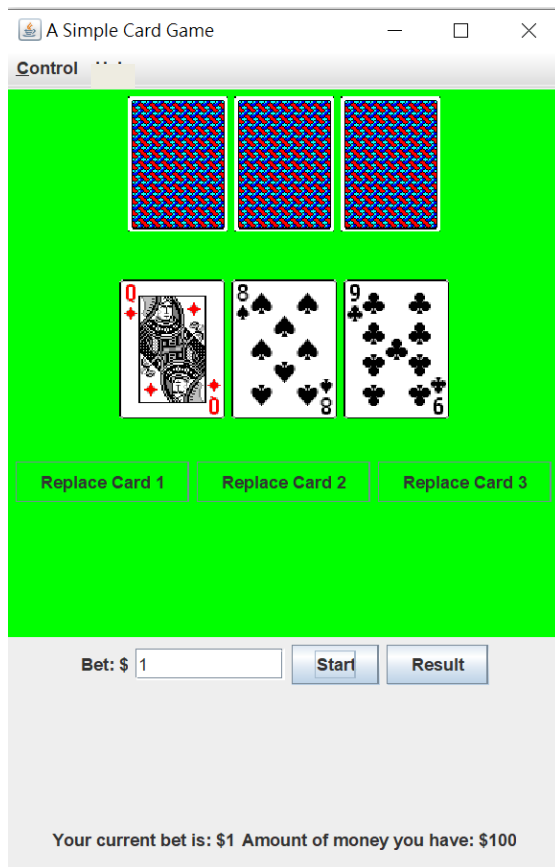
When the player clicks Replace Card button, the corresponding card should be replaced:

(the following screen capture shows an example when Replace Card 1 button is clicked)
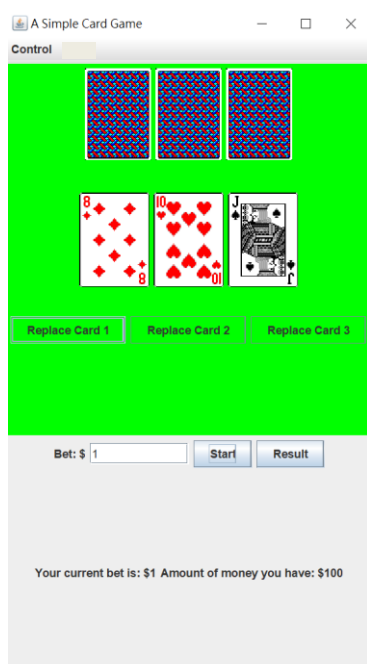
When the player loses all his/her money, a message notifying game over should be shown, in addition, the 3 JButton for replacing cards, the JButton for Start and the JButton for Result should be disabled:

When the player clicks Control → Exit, the game windows should be closed.
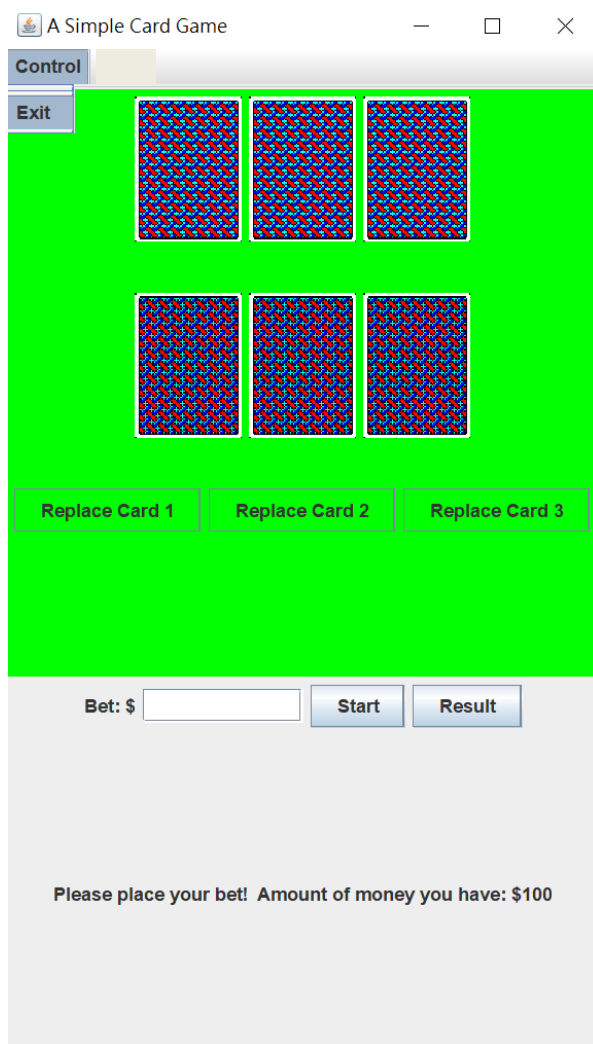
Marking Scheme

Correct implementation of GUI components: (Total 12%)

[6%] - 6 JLabel components for holding 6 ImageIcon components.

[1%] - 3 JButton components for the player to replace card 1, card 2 and card 3

[1%] - 1 JLabel component showing the string "Bet: $"

[1%] - 1 JTextField component for the player to input his/her bet

[1%] - 2 JButton components for the player to start the game and to evaluate the result

[1%] - 2 JLabel components for displaying important messages and the remaining budget that the player has

[1%] - 1 JmenuBar that consists of 1 JMenu "Control" which contains 1 JMenuItem to quit the game.

Correct functionality of the game: (Total 78%)

[8%] - Implementation of the Start button (i.e. drawing 3 cards from the top of the dealer's pack and display the 3 cards after clicking the Start button)

[4%] - Implementation of the disable of "Result" button, and the replace card buttons before the game starts (4%)

[18%] - Implementation of Replace Card buttons to replace a particular card with the card that is at the top of the dealer's pack and the disable of these buttons after the card is replaced or there have already been 2 cards replaced.

(6% each)

[4%] - Implementation of the disable of "Start" button after the game starts and the re-enable of the button after the result is displayed, evaluated and the player still has money.

[18%] - Implementation of evaluating the result with correct rules to determine who has better cards after clicking the Result button

[5%] - Implementation of showing the dealer's cards after clicking the Result button

[3%] - Implementation of messages when the player wins/loses

[3%] - Implementation of money incremental/deduction when the player wins/loses

[7%+3%] - Implementation of shuffling the cards and correct amount of player's money when a new game is started

[3%] - Implementation of messages and the disable of Replace Card buttons, the Start button and the Result button when the player loses all his/her money

[2%] - Implementation of exiting the game

JavaDoc (Total 10%)

Submission

After completing the assignment, export the project from Eclipse. Please submit the exported project in a single compressed file (in .zip) to Moodle. **Late submission is not allowed**. **Do not submit** *.class* **files.**

After submission, you are advised to download your submission from Moodle to try in Eclipse to ensure all necessary files are included in the submission.

You will get 0 mark if:

- you submit .class files instead of .java source files, or

- you submit java source files that cannot be compiled in Eclipse, or

- you submit java source files that are downloaded from the Internet, or

- you submit java source files from your classmates, or

- you submit java source files from friends taken this course last year.

For this assignment, you are required to write **JavaDoc** for all non-private classes and non-private class member functions. Programs without JavaDoc comments will lead to mark deduction. However, you don't need to generate JavaDoc HTML. Just write comment blocks in your source program.