EECS 1720
ASSIGNMENT 2 :: Java AWT/Swing/GUI application – Interactive Application

The overarching goal of assignments 1 & 2 in this course is to: (1) design and construct a supporting set of classes that model necessary components of a simple interactive application/ game, and (2) design/integrate this into a functional graphical user interface (GUI) that can be used to configure and launch/demonstrate various features of the application/game

(please refer to assignment 1 description for the types of application – you do not need to build upon your previous assignment, however, obviously it should be easier to build upon the classes you have already started with).

In **Assignment 2**, the classes (from assignment 1, or a new set of classes) will be extended and integrated with UI elements & event handling to support both basic configuration of the (card) game/application and the specific chosen elements of gameplay/operation (the assignment may not necessarily offer a complete/polished game/app experience, however, the submission should be able to ultimately demonstrate certain key INTERACTIVE elements.

Assignment 2 (minimum) deliverables:

1) **Process a minimum of 2 action-based events that result from interaction with a UI control or events spawned automatically via a thread), that directly result in some visual change in the scene, and/or a modification to the state of your application**

    a. The application should have at least one handler to process **at least 2 kinds of UI triggered action event (or similar)**

    b. The two action events **cannot be from the same type of UI control** (they must be from different types of controls.. i.e. handling two events triggered by two different buttons does not satisfy this requirement in full. You should have at least one other UI control that is not of the same type. If you have an animated or thread-triggered action event, this could count as one of the above (but there should also be at least one other input-driven action event in your working demo).

    For instance (in the context of a scene from a game), clicking on a button or other UI control would create one of the types of cards/or game states – e.g. to display/or shuffle a deck and display a single hand for a user (with the implemented visual/rendered states completed in assignment 1 (positioned on a Canvas/JPanel or similar situated within a JFrame).

    Another control might be used to set a variable (or set of variables/configuration) for the app/game, restart/implement an action. Or clicking on a card/player/object (set up as a UI control) might trigger its use in the application (which would run a method to update the state of the game/app)

2) **Process a minimum of 2 key-based/mouse-based events that result in some visual change in the scene, and/or a modification to the state of your application**

   a. The application must have at least one handler for each of the above types of events to process – this can be either on a graphical object or in the window (e.g. drag event or mouse down/release, key press/release, etc.)

      For example (card game), you could have a hand of cards, and use a number key to select and play a particular card. You could have a key that triggers a game to pause/quit, or prompts the user to provide some input through a dialogue window (like the JOptionPane class discussed in lectures, or a similar dialog for selecting Files from the local file system to load from / save to).

      In another context (record based application), a click into a canvas area could provide input for the choice/modification of an item (e.g. play or discard a card, cut the deck, use a resource, etc), or to move items around (perhaps they are JLabels of images/text labels that snap to different locations like a scrapbook).

      *** these are examples only. We are not expecting AAA games or Photoshop quality tools. Keep it as simple as possible. Ensuring that the minimum specifications are achieved.*

3) **The submission should include a signed version of the AcademicIntegrity statement (from assignment 1), a README.txt and a UML diagram describing your program (with its gui components and event handling components) linked with any classes that you have built that are part of/used by the application in its most recent form (this should be an extension of the diagram you had in assignment 1, or a new one if not using the classes developed in assignment 1).**

   Academic Integrity Statement (see attached at the end of this file). Sign and submit this page with your final assignment 2. Include all names of the students in the group (remember max two students per group).

   README.txt:

   a. Include all the names and student ID's of the people in your group
   b. Include a brief description of the project, **what it does** (not what you had hoped it would do), and **how to use it**
   c. Include any references to resources/inspiration used for this project

   UMLDiagrams.pdf:

   d. Include one or more UML class diagrams outlining the classes you have built for this project, what including all HAS-A and IS-A relationships used
   e. **Bonus marks if you use inheritance or interfaces (OR you are able to organize your classes to follow a model-view-controller (MVC) design pattern -** within any of the classes you build (not including extending JFrame or using any Listener interfaces).

RUBRIC (Marking Scheme) - 50 points total = worth 5% toward final grade

- Meets Goal 1 (10 points)
- Meets Goal 2 (10 points)
- Meets Goal 3 (10 points)
- Aesthetics/Creativity (10 points)
- Logical Design (10 points)

Assignment 1 & 2 Resources:

Java API:	https://docs.oracle.com/javase/8/docs/api/
Java AWT/Swing:	https://www.javatpoint.com/java-awt
Java Swing:	https://docs.oracle.com/javase/tutorial/uiswing/components/index.html
Java FX:	https://docs.oracle.com/javase/8/javafx/api/toc.htm
Processing:	http://processing.github.io/processing-javadocs/core/

SUBMISSION- DUE 11:59pm, April 10, 2023

Submit the following files to assignment link: **"a2" on web-submit**
https://webapp.eecs.yorku.ca/submit/

- **Academic Integrity Statement** (+ group members - 2 MAX per group)
  - *see end of this document*
- **Pdf/Docx** files for task 3 (project description) & 2 (uml diagram)
- **Exported project file** (*.zip) including your java source files for tasks 1&2

ACADEMIC INTEGRITY STATEMENT

We (the undersigned) hereby confirm that this assignment represents the sole work of the individuals listed below.

We (the undersigned) confirm that this work has been completed in adherence to the Senate Policy on Academic Honesty, without unapproved collaboration or the use of unpermitted aids or resources.

We recognize the importance of academic integrity and understand that there is no tolerance towards academic dishonesty within the Lassonde School of Engineering. We are aware that any suspected breaches will be reported to the Academic Honesty unit within the Student Welcome and Support Centre, and may result in additional penalties in accordance with the Academic Honesty Policy.

Name: First, Last	**Login Name:** (indicate the login used for submission)	**Student No.:**	**Signature/Date:**