Institute of Computer Science

CMSC 22: Object-Oriented Programming

CHECKPOINT JOURNAL 05

Instructions: Accomplish* this journal every checkpoint so we can monitor your progress and improve everyone's learning experience. Answer and submit as Google Doc (PDF is only for those who have limitations in working online.)

**Accomplish this when you are done (or almost done) with the lecture and lab requirements for the week.*

Name: **Gabinete, Keith Ginoel S.** Checkpoint Topic/s: **Graphical User Interface I**

Student Number: **2020-03670** Date: **May 06, 2023**

1. What problem/confusion did you encounter about the lesson/s or requirement/s?

   Explain the specifics of the problem (Minimum of **2** sentences).

   *Note that even If no problem was encountered in understanding the lesson, it's certain that at least a minor issue will be faced while doing the requirements, especially the **lab exercise**. Discuss at least one challenge faced.*

   I was getting frustrated on how would I be able to properly construct a 2d array for the game board of the memory game in which each number from 1-8 needs to appear twice. My first approach was to loop through the contents of the 4x4 2d array first then generate a number from 1 to 8 and assign it to the current element being traversed; then, Ill loop through the contents of the 2d array again to check if the randomly generated number has appeared twice already. However, I realized that this method of mine is not only inefficient but is actually very confusing to implement as well. Therefore, I need to think/search for a better way to construct this 2d array I wanted.

2. How did you solve it and what became your solution? **Explain the specific solution found.**

   Include **references** and **code snippets** when applicable (Minimum of **3** sentences).

   As I was skimming through online sites on how I would be able to implement the logic of creating a 2d array of size 4x4 in which each number from 1-8 should appear twice on it, I came across this youtube video <https:// www.youtube.com/watch?v=4Fvs0KI_Eqc> showing a tutorial on building a memory game in Java as well. Though the game is image-based (pairing images instead of numbers), I still earned an idea of how I would be able to fix this problem of mine about building a 2d array applying the logic of the memory game. Here, in the video provided, the host (instead of looping through the contents of the 2d array (game board)) chooses to loop through the given images (needed to be compared) then generates two sets of random indices for the array to store the current image being traversed. This way, since we have two indices to store the image into, we can assure that the current image will appear exactly twice in the 2d array (board game). This method will repeat until all images have been traversed already and all indices of the 2d array are filled with values. Same with the logic of the number-based memory game (1-8), if we traverse through the numbers 1-8 first then generate two indices to store each number into, then we can assure that all the contents of the 2d array (4x4) will be filled and each number (from 1-8) will appear at exactly two times.

3. Choose at least one of the things discussed that you understood the most. Imagine explaining it to a classmate.

   Explain it in your own words (Minimum of **4** sentences - can be 2 sentences for each of the week's topics).

JavaFX is a java library that contains many classes and interfaces to build GUI applications that are written in native Java code. JavaFX allows developers to design, create, test, debug and deploy applications that can run on multiple platforms such as Mobile, Web and Desktops. It is intended to replace Swing as the standard GUI library for Java as it has more features than Swing. With JavaFX, you can also style with CSS, just like in web applications.
To create a program with JavaFX, one must need to extend a class into the JavaFX Application class. A JavaFX application consists of stage, scene and nodes. Stage refers to the application window for JavaFX. A scene is attached inside the stage. Scene handles the visual components that are to be displayed on the stage. JavaFX components, also referred to as nodes, can be categorized into two: branch nodes and leaf nodes. Branch nodes can have children while leaf nodes can't.

Please communicate urgent concerns to your instructor via Discord. Do not write them down here so that they can be addressed immediately. (Ex: Installation problems, health concerns).