

GUI Programming 2024 to 2025 – Year 2

Labwork 6: (10% overall or 100 points of 600 points for labwork this semester)

Topic: Internal frames and menus, mouse listeners and choosers, option dialogs, 2D Graphics on a JFrame

IMPORTANT NOTES:

- **NO COPYING PERMITTED AND ZERO MARKS WILL APPLY TO COPIED WORK. FURTHER ACTION MAY BE TAKEN AGAINST STUDENTS THAT HAVE BEEN FOUND TO COPY WORK.**
- **ASSESSMENT WILL INVOLVE ONE-TO-ONE QUESTIONS ABOUT YOUR SUBMITTED WORK. A COMPLETED SELF-ASSESSMENT SHEET WILL BE USED TO GUIDE THE ASSESSMENT. USE COMMENTS IN YOUR CODE TO ENSURE YOU DON'T FORGET WHY YOU WROTE CODE YOU MAY LATER BE ASKED ABOUT.**
- **ALL WORK MUST BE SUBMITTED TO MOODLE BY DATES SPECIFIED (SUBMISSION DEADLINES WILL BE POSTED ON MOODLE).**
- **MANY OF THE TASKS ASSIGNED BELOW CAN BE COMPLEX AND\OR THE DESCRIPTIONS MAY REQUIRE FURTHER CLARIFICATIONS. PLEASE USE THE AVAILABLE LAB TIMES TO ASK FOR CLARIFICATIONS AND ADVICE\HINTS ON THE TASKS BELOW.**
- **YOU CAN USE A SIMPLE JAVA ENABLED TEXT EDITOR IF YOU WISH, e.g., TEXTPAD or NOTEPAD. HOWEVER, I SUPPORT THE MOVING ON TO A MORE ADVANCED IDE AT THIS POINT ALSO (e.g., Eclipse or IntelliJ or NetBeans).**
- **CHATGPT and other similar AI tools that can code simple solutions are not permitted. THEY DO NOT TEACH YOU HOW TO BECOME A GOOD PROGRAMMER EITHER!**

NOTE: COMPLETING THIS LAB EARLY WILL LIKELY REQUIRE STUDYING AHEAD!!!

Part 1 – Internal frame launched from a JMenu (20 points)

Create a class called **Lab6Part1**. Make this class a JFrame that is large (the exact size is up to you). Add a menu to the JFrame called 'Games Menu' that contains two menu items called 'Launch Chess Game' and 'Launch Snakes and Ladders'. When the user clicks the 'Launch Chess Game' menu item show a **JInternalFrame** within the JFrame with a JLabel showing a picture of a chess board (it doesn't have to work as a game just show the image). When the user clicks the 'Launch Snakes and Ladders' menu item make an internal frame appear in the JFrame with a label in it showing an image of the game of Snakes and Ladders. You may choose to show different board games if you like: none of them have to work as games but please have the menu's named correctly to reflect the game you are launching. Each game image must be launched in a label within an **internal frame** inside of the JFrame. Modularization in the form of methods or classes will be rewarded.

- Create the JFrame (1 point)
- Create the menu and menu items (5 points)
- Add the listeners and handlers (4 points)
- Launch each game in an internal frame [2 minimum games] (6 points)
- Modularize code using methods or classes (4 points)

Part 2 – Mouse with JFileChooser and JColorChooser (20 points)

Create a class called **Lab6Part2**. Make this class a JFrame. Create a panel inside the JFrame to hold TWO bordered and stacked JLabels (they can be stacked horizontally or vertically). Set the first JLabel with the text 'Choose an image here' and the second label should have the text 'Choose a colour here'. Implement the frame to respond to **mouse movements** so that when the user's mouse **enters** the first label a **JFileChooser** pops up and allows the user to choose and display an image of their choice in the label. Make the image chosen appear in the label. When the user **exits** the first label make the image **disappear**. When the user mouse **enters** the second label make a **JColorChooser** appear and allow the user to pick a colour to set the background of the label, i.e., if the user picks green from the JColorChooser pallet then the background of the label background changes to green (Note: You may have to use setOpaque on labels and you may need to add additional panels and layouts to make this work and you could have a small test-image in a specific folder to use when testing the file chooser).

- Create the JFrame (1 point)
- Create the labels (bordered) (2 points)
- Stack the labels in the frame (2 points)
- Add mouse listeners and handlers (4 points)
- Launch the JFileChooser when entering the first label (3 points)
- Set the label to show the chosen image and exit removes image (3 points)
- Launch the JColorChooser when entering the second label (3 points)
- Set the label to show the chosen colour (2 points)

Part 3 – A small adventure game using only Dialogs (30 points)

Create a class called **Lab6Part3**. Create a JFrame that mimics a space adventure game using only dialogs (you can choose a different theme other than space but you will have to come up with a way to launch the game to get it started!). The user will be presented with portals (represented as dialog options) to enter at the beginning of the game (use your imagination here but you must use **Dialogs**). Once a user chooses a portal present at least **ONE** other decision for them to make using dialogs before the game ends (e.g. You are presented with a cup of liquid to drink by an alien, do you drink it? Yes No etc.; does drinking the cup kill you or make you stronger? etc.). The options **MUST** be controlled by **Dialogs**. The end of game status should be shown using a **dialog**. Use a least **ONE** custom image added to a dialog in the dialog sequence (this means you override the default image in the option dialog by passing a custom image!!).

- Present the initial game option using a dialog (with three 'portals') (3 points)
- Create PATH 1: Another decision\option leading to an outcome (6 points)
- Create PATH 2: Another decision\option leading to an outcome (6 points)
- Create PATH 3: Another decision\option leading to an outcome (6 points)
- Flow of game\adventure well controlled (meaningful decisions) (3 points)
- Inclusion of a customized image in a dialogs (4 points)
- Overall game scenario complete\good quality (2 points)

Part 4 – Draw seasonal character on a JFrame with 2D Graphics (30 points)

Create a Java program called **Lab6Part4**. Create a JFrame that will be used to draw **2D graphics** of Santa Claus or Missus Claus or both if you wish! (the entire character(s) should be drawn not just a face). The drawing may be a little "rough" but try and put some shapes together that at least resemble the character(s)! Split up the drawing using methods, e.g., drawFace(Graphics g), drawHead(Graphics g), drawLimbs(Graphics g), drawBody(Graphics g) – (Note: you will need to *pass the graphics context as a parameters to the methods* when drawing with your own custom methods.)

- Create the JFrame with the paint method (2 points)
- Include Graphics g parameter and call to super.paint(g) (3 points)
- Draw head and face (with method – 2 points for method use) (8 points)
- Draw body (with method – 2 points for method use) (6 points)
- Draw limbs (arms\legs) (with method – 2 points for method use) (6 points)
- Use colour to make it look better (2 points)
- Overall look\design (i.e. how much does it look like a snowman) (3 points)