

# COMP3609: Game Programming 2023/2024

## Lab#1: Saturday January 27, 2024

### Summary of Lab Session

#### Quick Recap

The lab discussed various topics related to writing code for games. The first topic involved the introduction of Ms. Zainab Hanif as the tutor for the course. The agenda for the day included learning how to work with the screen and the Java 2D API. The lecturer also proposed a poll to determine interest in the face-to-face labs. The topics covered in the lab session included creating and customizing an empty window viewer in Java, manipulating colours in a program, creating a new window to draw on, overriding the *paint* method, and responding to keyboard and mouse inputs. Towards the end of the lab session, the names of three individuals who volunteered to be class representatives for the course were presented. Part of the lab session involved a coding session led by Ms. Hanif, who discussed responding to mouse clicks in a Java AWT game, the keyboard's *KeyListener* interface, and the structure and components of the game window.

---

#### Screen and 2D API

The lecturer discussed the agenda for the day which involved learning how to work with the screen and the Java 2D API. He introduced Ms. Zainab Hanif as the tutor for the course and mentioned that she would be taking over more of the lab sessions in the coming weeks. The lecturer also mentioned a significant increase in the course enrolment and proposed a poll to determine interest in the face-to-face labs. He showed how to use the MS-DOS window and Notepad for coding but recommended that students should use whatever Java IDE they are comfortable with.

#### Customizing Empty Window Viewer in Java

The lecturer showed how to create and customize an empty window in Java. He walked through the process of compiling and running the code, setting the window's size and location, and changing the background colour. The lecturer emphasized the importance of saving the code before compiling and recompiling to ensure that all the .class are up to date. He also discussed the possibility of further enhancements to the window viewer, such as repositioning the JFrame. He explained that the JFrame could be moved to any location and suggested using "set location relative to null" to centre the window on the screen.

#### Colour Manipulation in Java Programming

The lecturer showed how to manipulate colours in a program. He explained how to create a Color object using the Color class in Java, which takes three parameters for the red, green, and blue components. He highlighted that with a byte, which is eight bits, the program can store up to 255 combinations for each colour. He also suggested

experimenting with different colour combinations and using online resources such as colour charts for inspiration. Towards the end, he showed how to set the background colour of a window used a Color object.

### **JPanel Creation and Compilation Process**

The lecturer discussed the creation of a new drawing window, referred to as a JPanel, which is a drawing area on the JFrame. He outlined the process of inheriting from JPanel to create a new panel for drawing an alien. He stressed the importance of file organization and the need to recompile if changes are made. The lecturer also explained the process of compiling and running the FaceViewer application, emphasizing the need for various import statements. He discussed how the JPanel is automatically redrawn whenever the window needs to be refreshed.

### **Drawing Shapes and Overriding Methods in JPanel Class**

The lecturer discussed the method in the JPanel class that instructs the operating system to paint on the screen. He explained the process of overriding this method, emphasizing the importance of maintaining the same method signatures in object-oriented programming. He demonstrated how to draw shapes like ellipses, lines, and rectangles using the Java 2D API, explaining the parameters for each shape. He also discussed setting font attributes and the parameters involved in displaying strings on the screen. Towards the end, he clarified some misconceptions about the parameters involved in drawing shapes (mostly, mixing up the x and y parameters, as well as the width and height parameters).

### **Painting Process and Object Encapsulation Discussion**

Vance inquired about the use of polymorphic Graphic and Graphics2 objects which the lecturer acknowledged as a good question. However, the lecturer deferred answering Vance's question, promising to provide a more detailed explanation in the following week. After a short discussion about the modification of a class and the addition of constructor parameters, the lecturer tasked the students with implementing changes in the *paint* method to shift everything relative to a certain x and y coordinate. He also emphasized the importance of object encapsulation in programming in response to a comment from a student.

### **Drawing Adjustments and Input Handling**

The lecturer discussed the need for changes in the current setup, specifically regarding the drawing and adjustment of the coordinates. He emphasized the importance of keeping the width and the height the same and maintaining the same distance between the two end points of a line. He instructed the class to think about these adjustments and reconvene after a break. He also mentioned the need to shift certain parameters by the x and y coordinates. Lastly, he indicated that after the break, they would focus on responding to keyboard and mouse inputs and on modifying the JPanel.

### **Coding Issue and Visual Representation Discussion**

Jardel Mitchell discussed a coding issue he encountered in changing the code in *FaceViewer* to place it anywhere on the screen, and the lecturer clarified the correct approach, which involved adding to every coordinate the x and y coordinates of the point relative to which everything is drawn. Also, the width and height parameters should not be modified. The class observed the varying outcomes resulting from adding different values to these coordinates. The lecturer suggested starting with hard coding before moving to parameters once a functional system was established. He also advised trying larger numbers for better visual representation (due to pixel measurement being used for the parameters).

### **Volunteers for Class Representative, Coding Session: Java AWT Game**

The lecturer announced that three individuals, including Mr. Jardel Mitchell, Mr. Gilliano Agard, and Mr. Vance Ragbirsingh, volunteered to be class representatives for the course. He emphasized the role's importance and shared some past challenges. The meeting then transitioned into a coding session led by Ms. Hanif, who presented on responding to mouse clicks in a Java AWT game. She explained the process of creating an event every time an action is performed and how the precise location of the mouse click can be used to trigger specific responses in the game.

### **Keyboard Key Listener Interface Discussion**

Ms. Hanif discussed the keyboard's *KeyListener* interface, including methods for key press and release. She explained that the keys have specific codes, such as `VK_1` for the number one key. Jardel Mitchell asked if it was possible to add more keys, but Ms. Hanif clarified that the keys are already preset and cannot be changed. The lecturer added that all keys on the keyboard are already mapped to constants. The class then discussed running and compiling the code. Ms. Hanif proceeded to discuss the game window and its components.

### **Application Functionality and Coding Discussion**

Ms. Hanif explained the functionality and coding of an application, emphasizing the use of action listeners and key listeners. She demonstrated the identification of clicked buttons using an example of a 'start' button. Ms. Hanif and the lecturer also discussed the functionality of an exit button in the demonstration code, confirming that it does not perform any action. They also touched on the 'focus on key' feature, which returns the focus to the main window after a button press to ensure proper functionality.

## **Game Window Structure and Challenges**

Ms. Hanif and the lecturer discussed the structure and components of the game window, including the information panel, button panel, and main panel. The lecturer suggested specifying parameters for these panels and using default values. They also discussed the placement of these panels within the main panel and the potential challenges in object-oriented programming. Ms. Hanif experienced a coding issue, which the lecturer assisted with, without any success. They also discussed the concept of instance variables and the potential challenges in object-oriented programming in deciding in which class methods should be placed. Towards the end of the lab session, the lecturer reminded the students about the upcoming election for the class representative and indicated that a working version of the code where the FaceViewer JPanel is inserted on the window with the Information panel and the Button panel, will be uploaded to myeLearning.

## **Next steps**

- The lecturer will conduct a poll on myeLearning to gauge interest in the face-to-face labs. He will provide information on the three class representatives soon.
- Students should work on modifying the FaceViewer code to shift everything so that the alien's face can be drawn correctly at any (x, y) location.
- Students should download the final version of the game window and review the code carefully.