**GitHub Repository: https://github.com/kkkfc5/cs449\_sprint0\_KennethKakie**

**Sprint #0 Report**

Instructions

**Objectives**

* Make decisions on the SOS software development project.
* Learn unit testing and GUI programming in the language of your choice.

**Deliverables and Grading Policy**

Read the “CS 449 Homework Overview” document **carefully** and make the key decisions for the software development. Use the following template to complete your report.

1. **Key Decisions of the SOS Project (2 points)**

|  |  |
| --- | --- |
| Object-oriented programming language | Java |
| GUI library (strongly encouraged) | Java Swing / AWT |
| IDE (Integrated Development Environment) | Eclipse |
| xUnit framework (e.g., JUnit for Java) | Junit |
| Programming style guide (must read it carefully) | Google Java Style |
| Project hosting site | Github.com |
| Other decisions if applicable |  |

Sample programming style guides:

* Google Java Style Guide: <https://google.github.io/styleguide/javaguide.html>
* Google C++ Style Guide: <https://google.github.io/styleguide/cppguide.html>
* Google Python Style Guide: <https://google.github.io/styleguide/pyguide.html>

1. **Unit testing (4 points)**

Find a tutorial on the unit test framework you have chosen and write at least two xUnit tests of a program you have written or found elsewhere. Attach here (1) the screenshot of your program execution and (2) the source code of your program *[via github]*.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

1. **GUI programming (4 points)**

Write a GUI program in the language you have chosen for your SOS project. The GUI of your program must include text, lines, a check box, and radio buttons. While you are recommended to consider the GUI for the SOS game board, it is not required. In this assignment, any GUI program of your own work is acceptable.

Attach here (1) the screenshot of your program execution and (2) the source code of your program.

(1)

A screen shot of a computer screen

Description automatically generated

(2)

MainWindow.java

package app;

import java.awt.GridLayout;

import javax.swing.\*;

public class MainWindow extends JFrame {

JPanel mainPanel = new JPanel(new GridLayout(4,1));

public MainWindow() {

// set size of the window

setSize(500,500);

// Specify an action for the window close button.

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

// create panel for the text label

JPanel text = new JPanel();

text.add(new JLabel("LABEL"));

// create panel for the checkbox

JCheckBox box = new JCheckBox();

box.setText("CHECKBOX");

// create panel for the radio button

JRadioButton button = new JRadioButton();

button.setText("BUTTON");

// attach all objects to the grid

mainPanel.add(text);

mainPanel.add(add(new JPanelGraphicsTest()));

mainPanel.add(box);

mainPanel.add(button);

// attach grid to the window

add(mainPanel);

setVisible(true);

}

public static void main(String[] args) {

new MainWindow();

}

}

JPanelGraphicsTest.java:

package app;

import java.awt.\*;

import javax.swing.\*;

public class JPanelGraphicsTest extends JPanel {

public JPanelGraphicsTest(){}

// this is called on initialization of the panel

*@Override*

public void paintComponent(Graphics g) {

super.paintComponent(g); // lets the superclass paint what we aren't updating ourselves?

// https://stackoverflow.com/questions/28724609/what-does-super-paintcomponentg-do

Graphics2D g2d = (Graphics2D) g; // cast the graphics to 2d because 2d has more options

g2d.setStroke(new BasicStroke(8, BasicStroke.***CAP\_ROUND***, BasicStroke.***JOIN\_ROUND***)); // set brush options

// not the best way, but proof of concept of making a line.

g2d.drawLine(this.getX() + 300, this.getY() - 50, this.getX() + 100, this.getY() - 10);

}

}