**Lab 7**

Josue Ponce

Montgomery College

3/27/2018

Author’s Note

This lab report was prepared for CMSC 203 CRN #30672, taught by professor Ahmed Tarek

**Table of Contents**

Task #1 Code……………………………………………………………………………………2

Task #1 Screenshot of Running GUI……………………………………………………………3

Task #2 Code……………………………………………………………………………………4

Task #2 Screenshot of Running Application……………………………………………………6

Task #3 Code……………………………………………………………………………………7

Task #3 Running Application…………………………………………………………………...9

Task #4 Code……………………………………………………………………………………10

Task #4 Screenshot of Running Application…………………………………………………….14

**Task #1 Code**

/\*\*

\* Driver class that launches GUI for hello world application

\* @author Josue Ponce

\* @date 3/27/18

\* @version 1.0

\*/

**import** **java.io.IOException**;

**import** **javafx.application.Application**;

**import** **javafx.scene.Parent**;

**import** **javafx.scene.Scene**;

**import** **javafx.stage.Stage**;

**public** **class** **FXDriver** **extends** Application {

/\*\*

\* The main method for the GUI example program JavaFX version

\* @param args not used

\* @throws IOException

\*/

**public** **static** **void** **main**(String[] args) {

launch(args);

}

**@Override**

**public** **void** **start**(Stage stage) **throws** IOException {

//student Task #1:

// instantiate the FXMainPane, name it root

FXMainPane root = **new** FXMainPane();

// set the scene to hold root

stage.setScene(**new** Scene(root, **500**, **100**));

//set stage title

stage.setTitle("Hello World GUI");

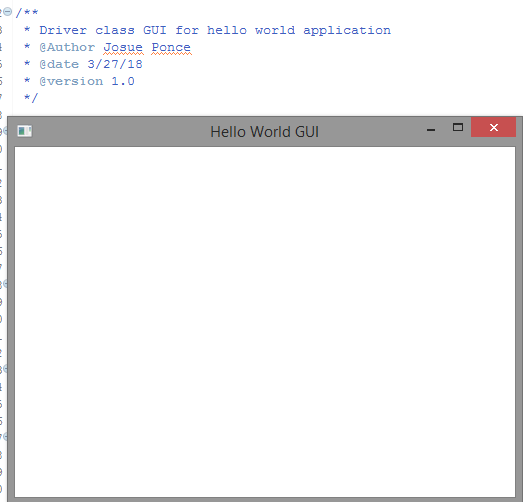
//display the stage

stage.show();

}

}

**Task #1 Screenshot of Running GUI**

****

*Note.* Screenshot of running GUI after compiling the line of codes.

**Task #2 Code**

/\*\*

\* @author Josue Ponce

\* @date 3/27/18

\* @version 1.0

\*/

**import** **javafx.application.Platform**;

**import** **javafx.event.ActionEvent**;

**import** **javafx.event.EventHandler**;

**import** **javafx.geometry.Insets**;

**import** **javafx.geometry.Pos**;

**import** **javafx.scene.control.Button**;

**import** **javafx.scene.control.Label**;

**import** **javafx.scene.control.RadioButton**;

**import** **javafx.scene.control.TextField**;

**import** **javafx.scene.control.ToggleGroup**;

**import** **javafx.scene.control.Tooltip**;

**import** **javafx.scene.layout.BorderPane**;

**import** **javafx.scene.layout.GridPane**;

**import** **javafx.scene.layout.HBox**;

**import** **javafx.scene.layout.Pane**;

**import** **javafx.scene.layout.VBox**;

//make the main panel's layout be a VBox

**public** **class** **FXMainPane** **extends** VBox {

//student Task #2:

// declare five buttons, a label, and a textfield

TextField textField;

Button helloButton;

Button howdyButton;

Button chineseButton;

Button clearButton;

Button exitButton;

Label labelVal;

// declare two HBoxes

HBox hb1;

HBox hb2;

//student Task #4:

// declare an instance of DataManager

/\*\*

\* The MainPanel constructor sets up the entire GUI in this approach. Remember to

\* wait to add a component to its containing component until the container has

\* been created. This is the only constraint on the order in which the following

\* statements appear.

\*/

FXMainPane() {

//student Task #2:

// instantiate the buttons, label, and textfield

labelVal=**new** Label("Feedback");

textField = **new** TextField();

helloButton = **new** Button("Hello");

howdyButton = **new** Button("Howdy");

chineseButton = **new** Button("Chinese");

clearButton = **new** Button("Clear");

exitButton = **new** Button("Exit");

// instantiate the HBoxes

hb1 = **new** HBox();

hb2 = **new** HBox();

//student Task #4:

// instantiate the DataManager instance

// set margins and set alignment of the components

//student Task #3:

// add the label and textfield to one of the HBoxes

// add the buttons to the other HBox

// add the HBoxes to this FXMainPanel (a VBox)

}

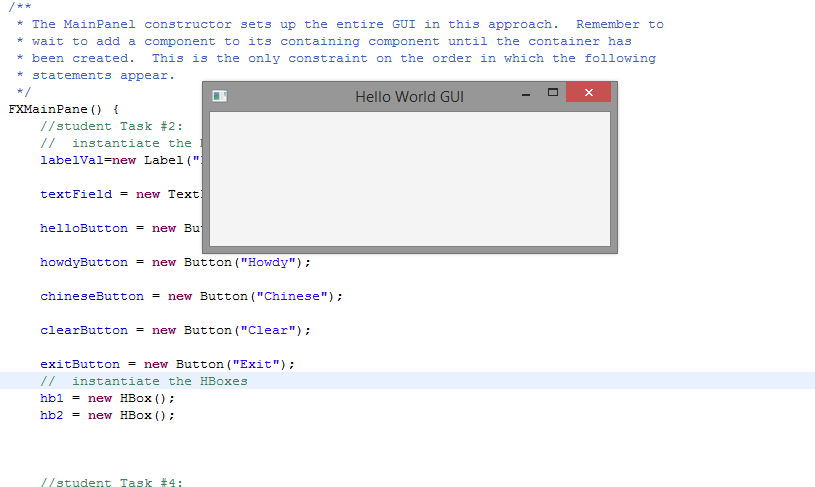
//Task #4:

// create a private inner class to handle the button clicks

}

**Task #2 Screenshot of Running Application**

The screenshot down below demonstrates that the application’s GUI still runs with the written code for this task. However, the buttons can’t be seen in the GUI because the next set codes that will be implemented will display the buttons.



*Note.* Screenshot displays the application running after the codes for task #2 have been implemented.

**Task #3 Code**

/\*\*

\* @author Josue Ponce

\* @date 3/27/18

\* @version 1.0

\*/

**import** **javafx.application.Platform**;

**import** **javafx.event.ActionEvent**;

**import** **javafx.event.EventHandler**;

**import** **javafx.geometry.Insets**;

**import** **javafx.geometry.Pos**;

**import** **javafx.scene.control.Button**;

**import** **javafx.scene.control.Label**;

**import** **javafx.scene.control.RadioButton**;

**import** **javafx.scene.control.TextField**;

**import** **javafx.scene.control.ToggleGroup**;

**import** **javafx.scene.control.Tooltip**;

**import** **javafx.scene.layout.BorderPane**;

**import** **javafx.scene.layout.GridPane**;

**import** **javafx.scene.layout.HBox**;

**import** **javafx.scene.layout.Pane**;

**import** **javafx.scene.layout.VBox**;

//make the main panel's layout be a VBox

**public** **class** **FXMainPane** **extends** VBox {

//student Task #2:

// declare five buttons, a label, and a textfield

TextField textField;

Button helloButton;

Button howdyButton;

Button chineseButton;

Button clearButton;

Button exitButton;

Label labelVal;

// declare two HBoxes

HBox hb1;

HBox hb2;

//student Task #4:

// declare an instance of DataManager

/\*\*

\* The MainPanel constructor sets up the entire GUI in this approach. Remember to

\* wait to add a component to its containing component until the container has

\* been created. This is the only constraint on the order in which the following

\* statements appear.

\*/

FXMainPane() {

//student Task #2:

// instantiate the buttons, label, and textfield

labelVal=**new** Label("Feedback");

textField = **new** TextField();

helloButton = **new** Button("Hello");

howdyButton = **new** Button("Howdy");

chineseButton = **new** Button("Chinese");

clearButton = **new** Button("Clear");

exitButton = **new** Button("Exit");

// instantiate the HBoxes

hb1 = **new** HBox();

hb2 = **new** HBox();

//student Task #4:

// instantiate the DataManager instance

// set margins and set alignment of the components

//student Task #3:

// add the label and textfield to one of the HBoxes

hb1.getChildren().addAll(labelVal, textField);

// add the buttons to the other HBox

hb2.getChildren().addAll(helloButton, howdyButton, chineseButton, clearButton, exitButton);

// add the HBoxes to this FXMainPanel (a VBox)

getChildren().addAll(hb1, hb2);

}

//Task #4:

// create a private inner class to handle the button clicks

}

**Task #3 Running Application**

****

*Note.*Running application after task #3 was completed. The GUI now displays buttons.

**Task #4 Code**

/\*\*

\* GUI for hello world application.

\* @author Josue Ponce

\* @date 3/27/18

\* @version 1.0

\*/

**import** **javafx.application.Platform**;

**import** **javafx.event.ActionEvent**;

**import** **javafx.event.EventHandler**;

**import** **javafx.geometry.Insets**;

**import** **javafx.geometry.Pos**;

**import** **javafx.scene.control.Button**;

**import** **javafx.scene.control.Label**;

**import** **javafx.scene.control.RadioButton**;

**import** **javafx.scene.control.TextField**;

**import** **javafx.scene.control.ToggleGroup**;

**import** **javafx.scene.control.Tooltip**;

**import** **javafx.scene.layout.BorderPane**;

**import** **javafx.scene.layout.GridPane**;

**import** **javafx.scene.layout.HBox**;

**import** **javafx.scene.layout.Pane**;

**import** **javafx.scene.layout.VBox**;

//make the main panel's layout be a VBox

**public** **class** **FXMainPane** **extends** VBox {

//student Task #2:

// declare five buttons, a label, and a textfield

TextField textField;

// Extra button for fun

Button frenchButton;

Button helloButton;

Button howdyButton;

Button chineseButton;

Button clearButton;

Button exitButton;

Label labelVal;

// declare two HBoxes

HBox hb1;

HBox hb2;

//student Task #4:

// declare an instance of DataManager

DataManager data;

/\*\*

\* The MainPanel constructor sets up the entire GUI in this approach. Remember to

\* wait to add a component to its containing component until the container has

\* been created. This is the only constraint on the order in which the following

\* statements appear.

\*/

FXMainPane() {

//student Task #2:

// instantiate the buttons, label, and textfield

labelVal=**new** Label("Feedback");

textField = **new** TextField();

helloButton = **new** Button("Hello");

howdyButton = **new** Button("Howdy");

chineseButton = **new** Button("Chinese");

// For fun button.

frenchButton = **new** Button ("French");

clearButton = **new** Button("Clear");

exitButton = **new** Button("Exit");

// instantiate the HBoxes

hb1 = **new** HBox();

hb2 = **new** HBox();

//student Task #4:

// instantiate the DataManager instance

data = **new** DataManager();

// set margins and set alignment of the components

HBox.setMargin(helloButton,**new** Insets(**10**));

helloButton.setOnAction(**new** ButtonHandler());

HBox.setMargin(howdyButton,**new** Insets(**10**));

howdyButton.setOnAction(**new** ButtonHandler());

HBox.setMargin(chineseButton,**new** Insets(**10**));

chineseButton.setOnAction(**new** ButtonHandler());

HBox.setMargin(frenchButton,**new** Insets(**10**));

frenchButton.setOnAction(**new** ButtonHandler());

HBox.setMargin(clearButton,**new** Insets(**10**));

clearButton.setOnAction(**new** ButtonHandler());

HBox.setMargin(exitButton,**new** Insets(**10**));

exitButton.setOnAction(**new** ButtonHandler());

HBox.setMargin(labelVal,**new** Insets(**10**));

hb1.setAlignment(Pos.CENTER);

hb2.setAlignment(Pos.CENTER);

//student Task #3:

// add the label and textfield to one of the HBoxes

hb1.getChildren().addAll(labelVal, textField);

// add the buttons to the other HBox

hb2.getChildren().addAll(helloButton, howdyButton, chineseButton, frenchButton, clearButton, exitButton);

// add the HBoxes to this FXMainPanel (a VBox)

getChildren().addAll(hb1, hb2);

}

//Task #4:

// create a private inner class to handle the button clicks

**private** **class** **ButtonHandler** **implements** EventHandler <ActionEvent>

{

**public** **void** **handle** (ActionEvent event)

{

Object source = event.getTarget();

**if** (source == helloButton)

{

String input = data.getHello();

textField.setText(input);

}

**else** **if** (source == howdyButton)

{

String input = data.getHowdy();

textField.setText(input);

}

**else** **if** (source == frenchButton) {

String input = data.getFrench();

textField.setText(input);

}

**else** **if**(source == chineseButton)

{

String input = data.getChinese();

textField.setText(input);

}

**else** **if**(source == clearButton)

{

textField.setText("");

}

**else** **if**(source == exitButton)

{

Platform.exit();

System.exit(**0**);

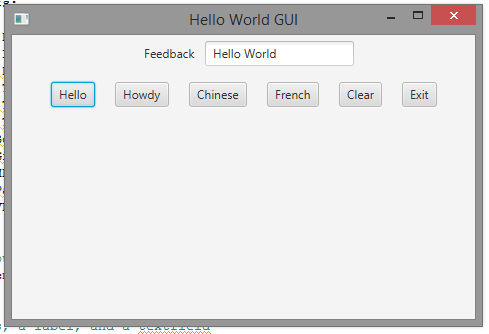
}

}

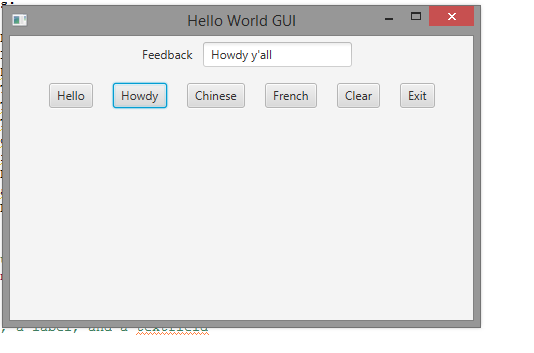
}

}

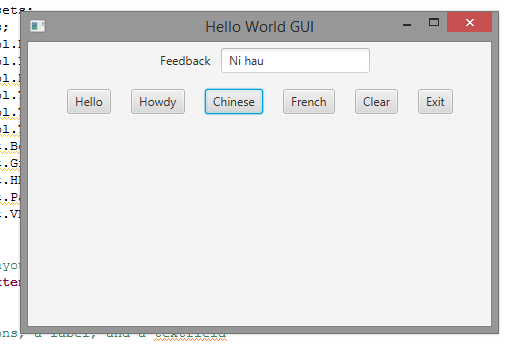
**Task #4 Screenshot of Running Application**



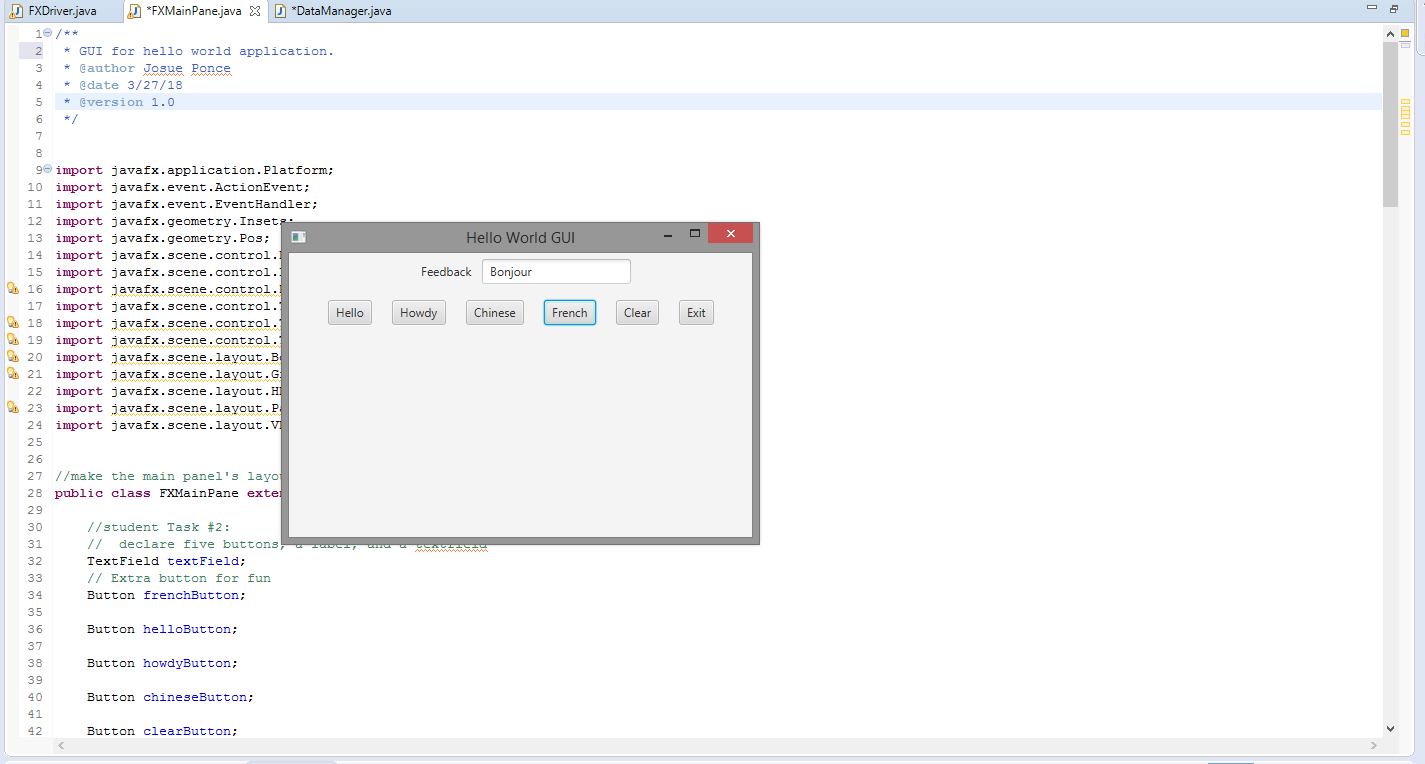
*Note.* Running application with the hello button working correctly.



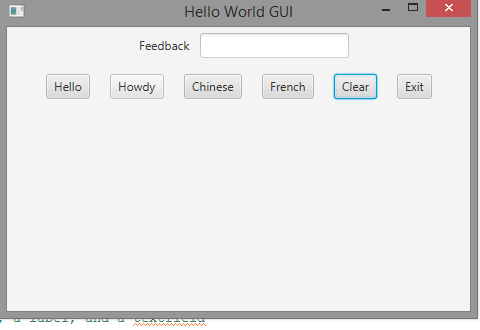
*Note.* Running application Howdy button working correctly.



*Note.* Running application with the Chinese button working correctly.



*Note.* Running application with the French button working correctly.



*Note.* Running application with the clear button working correctly.