JavaFX – Step 1

Student.java

```
public class Student {
    private int sid;
    private String firstName;
    private String lastName;
    private double gpa;

    public Student(int sid, String firstName, String lastName, double gpa) {
        this.sid = sid;
        this.firstName = firstName;
        this.lastName = lastName;
        this.gpa = gpa;
    }

    public int getSid() { return this.sid; }
    public String getFirstName() { return this.firstName; }
    public String getLastName() { return this.lastName; }
    public double getGpa() { return this.gpa; }

    public void setSid(int sid) { this.sid = sid; }
    public void setFirstName(String firstName) { this.firstName = firstName; }
    public void setLastName(String lastName) { this.lastName = lastName; }
    public void setGpa(double gpa) { this.gpa = gpa; }
}
```

DataSource.java

```
import javafx.collections.*;

public class DataSource {
    public static ObservableList<Student> getAllStudents() {
        ObservableList<Student> students = FXCollections.observableArrayList();

        students.add(new Student(100100100, "Janet", "Combes", 2.85));
        students.add(new Student(100100101, "Abichal", "Kaur", 1.71));
        students.add(new Student(100100102, "Cecile", "Lalonde", 3.60));
        students.add(new Student(100100103, "Pablo", "Rodriguez", 2.19));
        students.add(new Student(100100104, "Flora", "Ivanovic", 2.45));
        students.add(new Student(100100105, "Mahmoud", "Ashfaq", 3.15));
        students.add(new Student(100100106, "Stephen", "McCullough", 1.55));
        students.add(new Student(100100107, "Zhilong", "Fu", 1.35));
        students.add(new Student(100100108, "Sadiva", "Krupal", 2.64));
        students.add(new Student(100100109, "Carmine", "Dipaolo", 3.13));
        students.add(new Student(100100110, "Sarah", "Morrissey", 2.40));
        students.add(new Student(100100111, "Pavel", "Zakharov", 1.95));

        return students;
    }
}
```

Main.java

```
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.*;
import javafx.scene.control.*;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.*;
import javafx.scene.input.KeyCombination;
import javafx.scene.image.*;
import javafx.collections.*;
import javafx.event.*;
import javafx.scene.control.TableColumn.CellEditEvent;
import javafx.scene.control.cell.*;

import java.io.File;
import java.net.URL;
import java.net.URLClassLoader;

public class Main extends Application {
    private Stage window;
    private BorderPane layout;
    private TableView<Student> table;
    private TextField sidField, fnameField, lnameField, gpaField;

    @Override
    public void start(Stage primaryStage) throws Exception {
        primaryStage.setTitle("JavaFX Demo");

        /* create the menu (for the top of the user interface) */
        Menu fileMenu = new Menu("File");
        MenuItem newMenuItem = new MenuItem("New", imageFile("images/new.png"));
        newMenuItem.setAccelerator(KeyCombination.keyCombination("Ctrl+N"));
        fileMenu.getItems().add(newMenuItem);
        fileMenu.getItems().add(new SeparatorMenuItem());
        fileMenu.getItems().add(new MenuItem("Open...",
imageFile("images/open.png")));
        fileMenu.getItems().add(new SeparatorMenuItem());
        fileMenu.getItems().add(new MenuItem("Save", imageFile("images/save.png")));
        fileMenu.getItems().add(new MenuItem("Save As...",
imageFile("images/save_as.png")));
        fileMenu.getItems().add(new SeparatorMenuItem());
        MenuItem exitMenuItem = new MenuItem("Exit", imageFile("images/exit.png"));
        fileMenu.getItems().add(exitMenuItem);
        exitMenuItem.setAccelerator(KeyCombination.keyCombination("Ctrl+Q"));
        exitMenuItem.setOnAction( e -> System.exit(0) );

        Menu editMenu = new Menu("Edit");
        editMenu.getItems().add(new MenuItem("Cut", imageFile("images/cut.png")));
        editMenu.getItems().add(new MenuItem("Copy", imageFile("images/copy.png")));
        editMenu.getItems().add(new MenuItem("Paste", imageFile("images/paste.png")));

        Menu helpMenu = new Menu("Help");
        helpMenu.getItems().add(new MenuItem("About...",
imageFile("images/about.png")));
        helpMenu.getItems().add(new SeparatorMenuItem());
```

```
        helpMenu.getItems().add(new MenuItem("Help...",
imageFile("images/help.png")));

        MenuBar menuBar = new MenuBar();
        menuBar.getMenus().add(fileMenu);
        menuBar.getMenus().add(editMenu);
        menuBar.getMenus().add(helpMenu);

        /* create the table (for the center of the user interface) */
        table = new TableView<>();
        table.setItems(DataSource.getAllStudents());
        table.setEditable(true);

        /* create the table's columns */
        TableColumn<Student,Integer> sidColumn = null;
        sidColumn = new TableColumn<>("SID");
        sidColumn.setMinWidth(100);
        sidColumn.setCellValueFactory(new PropertyValueFactory<>("sid"));

        TableColumn<Student,String> firstNameColumn = null;
        firstNameColumn = new TableColumn<>("First Name");
        firstNameColumn.setMinWidth(200);
        firstNameColumn.setCellValueFactory(new PropertyValueFactory<>("firstName"));
        firstNameColumn.setCellFactory(TextFieldTableCell.<Student>forTableColumn());
        firstNameColumn.setOnEditCommit((CellEditEvent<Student, String> event) -> {

((Student)event.getTableView().getItems().get(event.getTablePosition().getRow())).setF
irstName(event.getNewValue());
        });

        TableColumn<Student,String> lastNameColumn = null;
        lastNameColumn = new TableColumn<>("Last Name");
        lastNameColumn.setMinWidth(200);
        lastNameColumn.setCellValueFactory(new PropertyValueFactory<>("lastName"));
        lastNameColumn.setCellFactory(TextFieldTableCell.<Student>forTableColumn());
        lastNameColumn.setOnEditCommit((CellEditEvent<Student, String> event) -> {

((Student)event.getTableView().getItems().get(event.getTablePosition().getRow())).setL
astName(event.getNewValue());
        });

        TableColumn<Student,Double> gpaColumn = null;
        gpaColumn = new TableColumn<>("GPA");
        gpaColumn.setMinWidth(100);
        gpaColumn.setCellValueFactory(new PropertyValueFactory<>("gpa"));

        table.getColumns().add(sidColumn);
        table.getColumns().add(lastNameColumn);
        table.getColumns().add(firstNameColumn);
        table.getColumns().add(gpaColumn);

        /* create an edit form (for the bottom of the user interface) */
        GridPane editArea = new GridPane();
        editArea.setPadding(new Insets(10, 10, 10, 10));
        editArea.setVgap(10);
        editArea.setHgap(10);

        Label sidLabel = new Label("SID:");
```

```
    editArea.add(sidLabel, 0, 0);
    TextField sidField = new TextField();
    sidField.setPromptText("SID");
    editArea.add(sidField, 1, 0);

    Label fnameLabel = new Label("First name:");
    editArea.add(fnameLabel, 0, 1);
    TextField fnameField = new TextField();
    fnameField.setPromptText("First Name");
    editArea.add(fnameField, 1, 1);

    Label lnameLabel = new Label("Last name:");
    editArea.add(lnameLabel, 0, 2);
    TextField lnameField = new TextField();
    lnameField.setPromptText("Last Name");
    editArea.add(lnameField, 1, 2);

    Label gpaLabel = new Label("GPA:");
    editArea.add(gpaLabel, 0, 3);
    TextField gpaField = new TextField();
    gpaField.setPromptText("GPA");
    editArea.add(gpaField, 1, 3);

    Button addButton = new Button("Add");
    addButton.setOnAction(new EventHandler<ActionEvent>() {
        @Override public void handle(ActionEvent e) {
            int sid = Integer.parseInt(sidField.getText());
            String firstName = fnameField.getText();
            String lastName = lnameField.getText();
            double gpa = Double.parseDouble(gpaField.getText());

            table.getItems().add(new Student(sid, firstName, lastName, gpa));

            sidField.setText("");
            fnameField.setText("");
            lnameField.setText("");
            gpaField.setText("");
        }
    });
    editArea.add(addButton, 1, 4);

    /* arrange all components in the main user interface */
    layout = new BorderPane();
    layout.setTop(menuBar);
    layout.setCenter(table);
    layout.setBottom(editArea);

    Scene scene = new Scene(layout, 600, 600);
    primaryStage.setScene(scene);
    primaryStage.show();
}

private ImageView imageFile(String filename) {
    return new ImageView(new Image("file:"+filename));
}

public static void main(String[] args) {
    launch(args);
```

```
    }
}
```

JavaFX – Step 2

Controller.java

```
package sample;

import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.event.ActionEvent;
import javafx.event.Event;
import javafx.event.EventHandler;
import javafx.fxml.FXML;
import javafx.scene.control.*;
import javafx.scene.input.MouseEvent;

public class Controller {
    @FXML private TextField usernameField;
    @FXML private PasswordField password1Field;
    @FXML private PasswordField password2Field;
    @FXML private TextField emailField;

    @FXML private TreeView<String> projectTreeView;
    @FXML private TextArea editor;

    // the initialize method is automatically invoked by the FXMLLoader - it's magic
    public void initialize() {
        TreeItem<String> rootItem = new TreeItem<>("Project");
        rootItem.setExpanded(true);

        TreeItem<String> src = new TreeItem<>("src");
        src.setExpanded(true);
        rootItem.getChildren().add(src);

        TreeItem<String> main = new TreeItem<>("main");
        main.setExpanded(true);
        src.getChildren().add(main);

        TreeItem<String> java = new TreeItem<>("java");
        java.setExpanded(true);
        main.getChildren().add(java);

        TreeItem<String> helloWorld = new TreeItem<>("HelloWorld.java");
        java.getChildren().add(helloWorld);

        TreeItem<String> gradle = new TreeItem<>("build.gradle");
        rootItem.getChildren().add(gradle);

        projectTreeView.setRoot(rootItem);
        projectTreeView.getSelectionModel().selectedItemProperty().addListener(new
ChangeListener() {

            @Override
```

```
            public void changed(ObservableValue observable, Object oldValue, Object
newValue) {
                TreeItem<String> selectedItem = (TreeItem<String>)newValue;
                if (selectedItem.getValue().equals("HelloWorld.java")) {
                    editor.setText("public class HelloWorld {\n"+
                            "    public static void main(String[] args) {\n"+
                            "        System.out.println(\"Hello, world!\");"+
                            "    }\n"+
                            "}\n");
                } else if (selectedItem.getValue().equals("build.gradle")) {
                    editor.setText("apply plugin: 'java'");
                }
            }

        });
    }

    public void register(ActionEvent e) {
        String username = usernameField.getText();
        String password1 = password1Field.getText();
        String password2 = password2Field.getText();
        String email = emailField.getText();

        // do something with this data

        System.out.println("Register:");
        System.out.println("\tUsername:   " + username);
        System.out.println("\tPassword1:  " + password1);
        System.out.println("\tPassword2:  " + password2);
        System.out.println("\tE-Mail:     " + email);
    }
}
```

register.css

```
.tab-pane .tab {
    -fx-background-color: linear-gradient(#d0d0d0, #b0b0b0);
}

.tab-pane .tab:selected {
    -fx-background-color: linear-gradient(#857DB1, #5D5393);
}

.tab .tab-label {
    -fx-alignment: CENTER;
    -fx-text-fill: #404040;
    -fx-font-size: 12px;
    -fx-font-weight: bold;
}

.tab:selected .tab-label {
    -fx-alignment: CENTER;
    -fx-text-fill: white;
    -fx-font-weight: bold;
 }

.bg {
```

```
    -fx-background-color: #D8D8EA;
}

.button {
    -fx-text-fill: white;
    -fx-font-weight: bold;
    -fx-background-color: linear-gradient(#857DB1, #5D5393);
    -fx-effect: dropshadow(three-pass-box, rgba(0,0,0,0.6), 5, 0.0, 0, 1);
}

.code {
    -fx-font-family: monospace;
}
```

sample.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.*?>
<?import java.lang.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TabPane?>
<?import javafx.scene.control.Tab?>
<?import javafx.scene.control.SplitPane?>
<?import javafx.scene.control.TextArea?>
<?import javafx.scene.control.TreeView?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.image.Image?>
<?import java.net.URL?>
<BorderPane xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="sample.Controller">
    <center>
        <TabPane>
            <tabs>
                <Tab fx:id="tab1" closable="false" text="SplitPane">
                    <content>
                        <SplitPane dividerPositions="0.25">
                           <items>
                               <TreeView fx:id="projectTreeView">
                               </TreeView>

                               <TextArea fx:id="editor" styleClass="code" />
                           </items>
                        </SplitPane>
                    </content>
                </Tab>
                <Tab fx:id="tab2" closable="false" text="GridPane">
                   <content>
                       <GridPane alignment="CENTER" hgap="10" vgap="10"
styleClass="bg">
                           <padding>
                               <Insets bottom="10" left="10" right="10" top="10" />
                           </padding>
```

```
            <children>
               <Label text="Username:"
                      GridPane.columnIndex="0"
                      GridPane.rowIndex="0" />

               <TextField fx:id="usernameField"
                          promptText="Your desired username"
                          GridPane.columnIndex="1"
                          GridPane.rowIndex="0" />

               <Label text="Password:"
                      GridPane.columnIndex="0"
                      GridPane.rowIndex="1" />

               <PasswordField fx:id="password1Field"
                              promptText="Your desired password"
                              GridPane.columnIndex="1"
                              GridPane.rowIndex="1" />

               <Label text="Password (again):"
                      GridPane.columnIndex="0"
                      GridPane.rowIndex="2" />

               <PasswordField fx:id="password2Field"
                              promptText="Repeat your password"
                              GridPane.columnIndex="1"
                              GridPane.rowIndex="2" />

               <Label text="E-Mail:"
                      GridPane.columnIndex="0"
                      GridPane.rowIndex="3" />

               <TextField fx:id="emailField"
                          promptText="E-Mail Address"
                          GridPane.columnIndex="1"
                          GridPane.rowIndex="3" />

               <Button onAction="#register"
                       text="Register"
                       GridPane.columnIndex="1"
                       GridPane.rowIndex="4"
                       styleClass="button" />

            </children>
            <columnConstraints>
                 <ColumnConstraints />
                 <ColumnConstraints />
            </columnConstraints>
            <rowConstraints>
                <RowConstraints />
                <RowConstraints />
                <RowConstraints />
                <RowConstraints />
                <RowConstraints />
            </rowConstraints>
         </GridPane>
      </content>
   </Tab>
```

```
            </tabs>
        </TabPane>
    </center>
    <stylesheets>
        <URL value="@register.css" />
    </stylesheets>
</BorderPane>
```

Main.java

```
package sample;

import javafx.application.Application;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.TreeItem;
import javafx.scene.control.TreeView;
import javafx.stage.Stage;

public class Main extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception{
        Parent root = FXMLLoader.load(getClass().getResource("sample.fxml"));

        primaryStage.setTitle("JavaFX - Demo 2");
        primaryStage.setScene(new Scene(root, 800, 600));
        primaryStage.show();
    }


    public static void main(String[] args) {
        launch(args);
    }
}
```

JavaFX – Canvas

Main.java

```
package sample;

import javafx.animation.KeyFrame;
import javafx.animation.Timeline;
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Group;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.canvas.Canvas;
import javafx.scene.canvas.GraphicsContext;
import javafx.scene.image.Image;
import javafx.scene.paint.Color;
```

```
import javafx.scene.paint.Paint;
import javafx.scene.shape.Arc;
import javafx.scene.shape.ArcType;
import javafx.scene.text.Font;
import javafx.stage.Stage;
import javafx.util.Duration;

public class Main extends Application {
    @FXML
    private Canvas canvas;

    @Override
    public void start(Stage primaryStage) throws Exception{
        Group root = new Group();
        Scene scene = new Scene(root, 800, 600, Color.LIGHTGRAY);

        canvas = new Canvas();
        canvas.widthProperty().bind(primaryStage.widthProperty());
        canvas.heightProperty().bind(primaryStage.heightProperty());

        root.getChildren().add(canvas);
        primaryStage.setScene(scene);
        primaryStage.show();

        draw(root);

        drawAnimation(root);
    }

    private void draw(Group group) {
        GraphicsContext gc = canvas.getGraphicsContext2D();
        System.out.println("width: " + canvas.getWidth());
        System.out.println("height: " + canvas.getHeight());
        gc.clearRect(0, 0, canvas.getWidth(), canvas.getHeight());

        // line
        gc.setStroke(Color.BLACK);
        gc.strokeLine(50, 50, 150, 250);

        // rectangles
        gc.setFill(Color.BLUE);
        gc.setStroke(Color.BLUE);
        gc.fillRect(250, 50, 100, 75);
        gc.strokeRect(250, 175, 100, 75);

        // rounded rectangles
        gc.setFill(Color.BEIGE);
        gc.setStroke(Color.BEIGE);
        gc.fillRoundRect(450, 50, 100, 75, 10, 10);
        gc.strokeRoundRect(450, 175, 100, 75, 20, 20);

        // ovals (ellipses)
        gc.setFill(Color.CORAL);
        gc.setStroke(Color.CORAL);
        gc.strokeOval(650, 50, 100, 75);
        gc.fillOval(650, 175, 100, 75);

        // arcs
```

```
        gc.setFill(Color.DARKCYAN);
        gc.setStroke(Color.DARKCYAN);
        gc.strokeArc(50, 350, 100, 75, 115.0, 45.0, ArcType.ROUND);
        gc.fillArc(50, 500, 100, 75, 45.0, 115.0, ArcType.ROUND);

        // polygons (one filled semi-transparent)
        gc.setFill(Color.color(0.8, 0.0, 0.3, 0.5));
        gc.setStroke(Color.HOTPINK);
        gc.strokePolygon(new double[] {250, 310, 300, 250}, new double[] {350, 360,
380, 400}, 4);
        gc.fillPolygon(new double[] {250, 310, 300, 250}, new double[] {500, 510, 530,
550}, 4);

        // text (with adjusted font)
        Font font = new Font("Arial", 24);
        gc.setFont(font);
        gc.setFill(Color.OLIVE);
        gc.setStroke(Color.OLIVE);
        gc.strokeText("CSCI2020u", 450, 400);
        gc.fillText("CSCI2020u", 450, 550);

        // image
        Image image = new Image("disk.png");
        gc.drawImage(image, 685, 400);
    }

    private Timeline timeline = null;

    private int frameOffsetX = 0;
    private int frameOffsetY = 0;
    private final int frameWidth = 128;
    private final int frameHeight = 128;
    private final int totalWidth = 768;
    private final int totalHeight = 1536;
    private final int numFrames = 6;
    private int frameNum = 0;

    private void drawAnimation(Group group) {
        Image sprites = new Image("sprites.png");
        GraphicsContext gc = canvas.getGraphicsContext2D();
        timeline = new Timeline();
        timeline.setCycleCount(Timeline.INDEFINITE);
        timeline.getKeyFrames().add(new KeyFrame(Duration.millis(20), new
EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent e) {
                gc.setFill(Color.LIGHTGRAY);
                gc.fillRect(685, 500, frameWidth, frameHeight);

                // draw the current frame
                gc.drawImage(sprites, frameOffsetX, frameOffsetY, frameWidth,
frameHeight, 685, 500, frameWidth, frameHeight);

                // proceed to the next frame of the animation
                frameNum = (frameNum + 1) % numFrames;

                // increment x offset and y offset
                frameOffsetX += frameWidth;
```

```
                if (frameOffsetX >= totalWidth) {
                    frameOffsetX = 0;
                    frameOffsetY += frameHeight;
                    if (frameOffsetY >= totalHeight) {
                        frameOffsetY = 0;
                    }
                }
            }
        }));
        timeline.playFromStart();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```