**National University of Computer & Emerging Sciences, Karachi**

**Computer Science Department**

**Spring 2022, Lab Manual - 08**

|  |  |
| --- | --- |
| **Course Code: SL3001** | **Course: Software Development and construction** |
| **Instructor(s):** | **Miss Nida Munawar, Miss Abeeha Sattar** |

**Lab # 08**

**How to Install Scene Builder**

**Step 1:**

1. visit [Scene Builder](http://gluonhq.com/products/scene-builder) (https://gluonhq.com/products/scene-builder/).
2. Click the Download now button.

When you do, a list of download options appears.

1. Click the button corresponding to your computer's operating system (Windows, Mac, or Linux).

As a result, the download begins. On a Windows computer, you get an .exe file. Double-click the file to begin the installation.

1. Follow the installation routine's instructions.

On a Windows computer, you accept a bunch of defaults.

Remember the path where you install scene builder

**Step 2:**

Open eclipse > now create a javafx project and setup user lib and VM arguments as we did in the last lab

Go to windows > preference >javafx > browse the path where you install the scene builder > select ..exe file > select apply > apply and close

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated



**Step 3:**

Right click on your project > new >others > new FXML document > name this file (e.g. main)

Right click on your main.fxml > open with Scene builder > youll pop up with scene builder GUI

A picture containing text, electronics, screenshot

Description automatically generated

Go to container > drag and drop AnchorPane to begin with > add buttons from controls dropdown

Graphical user interface

Description automatically generated

save your project > click on file > save

**Step 4:**

now head back to Main . java and add a line to your start method

replace the line

BorderPne root = new BorderPane();

With root node based on fxml file

Parent root = FXMLLoader.*load*(getClass().getResource("main.fxml"));

Make sure your main.fxml within your application package if this is not you will get a null pointer exception there are two fixes for this:

1. Put a forward slash with your resource name (it will look outside the application pckg)

Parent root = FXMLLoader.*load*(getClass().getResource("/main.fxml"));

OR

1. Drag your main.fxml to application pckg

Graphical user interface, text, application, chat or text message

Description automatically generated



**1st way:**

**Event Handling using SceneBuilder**

**Create a separate java class for events (e.g. controller)**

**package** application;

**import** javafx.event.ActionEvent;

//don’t include awt pckg

**public** **class** controller {

**public** **void** up(ActionEvent e) {

System.***out***.println("up");

}

**public** **void** down(ActionEvent e) {

System.***out***.println("down");

}

**public** **void** left(ActionEvent e) {

System.***out***.println("left");

}

**public** **void** right(ActionEvent e) {

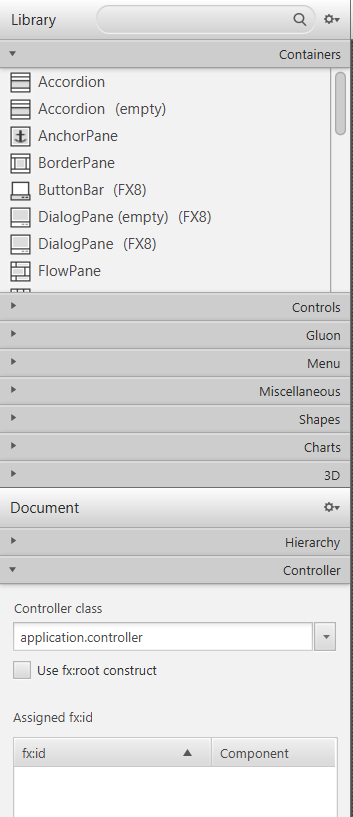
System.***out***.println("right");

}

}

Open scenebuilder navigate to left hand menu there should be a dropdown for controller and under controller there should be your controller class ,select application.controller

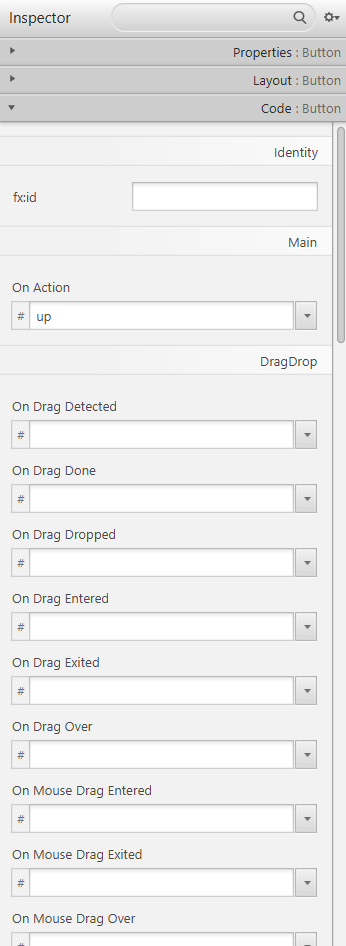






For each of the button click on the button there should be a right hand menu -> select code and add On Action

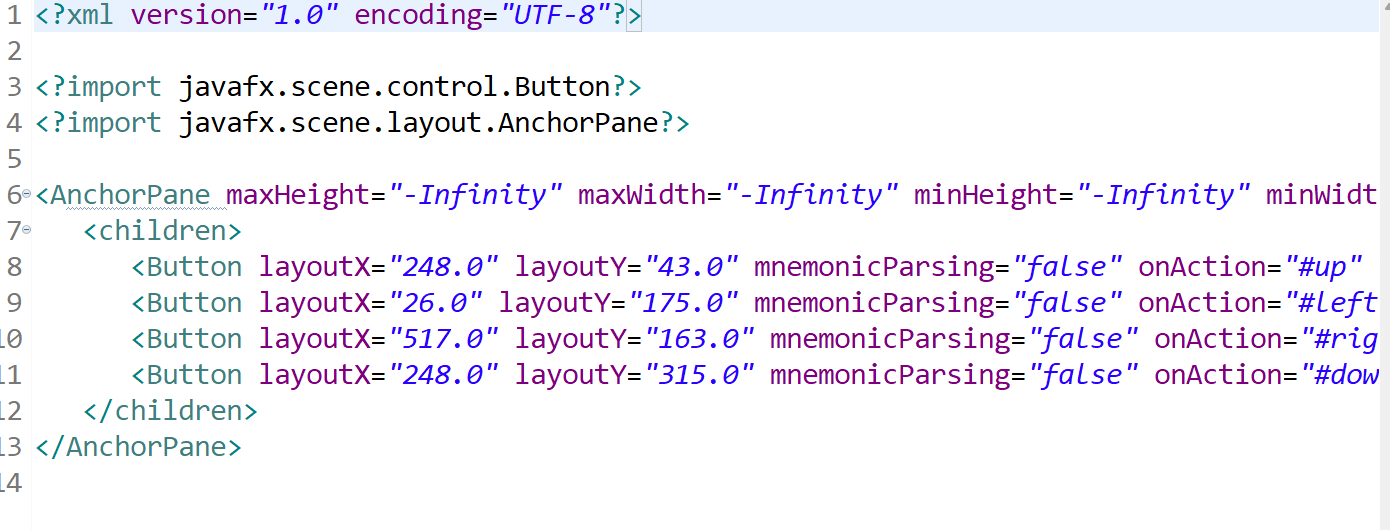
Select up method for up button and so on





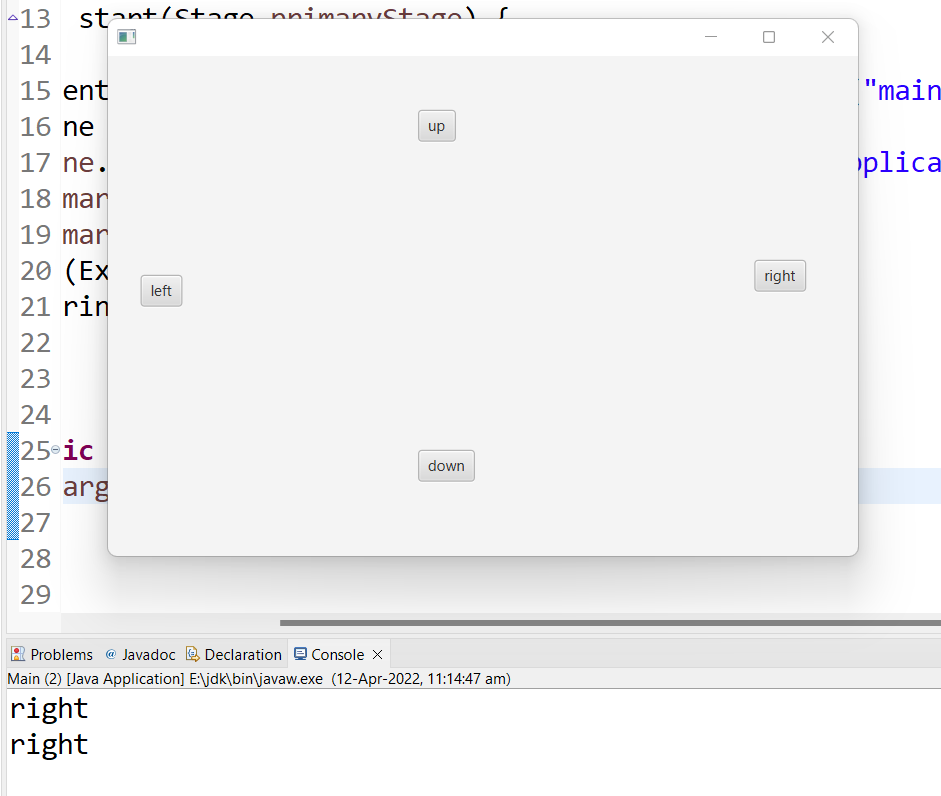
Add methods for each of the button and click on save

After this verify that your on action is added in your fxml file by opening the file in eclipse



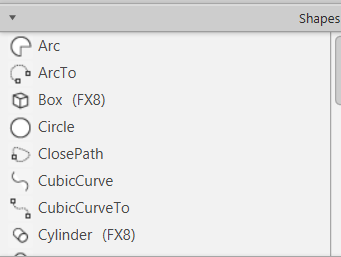


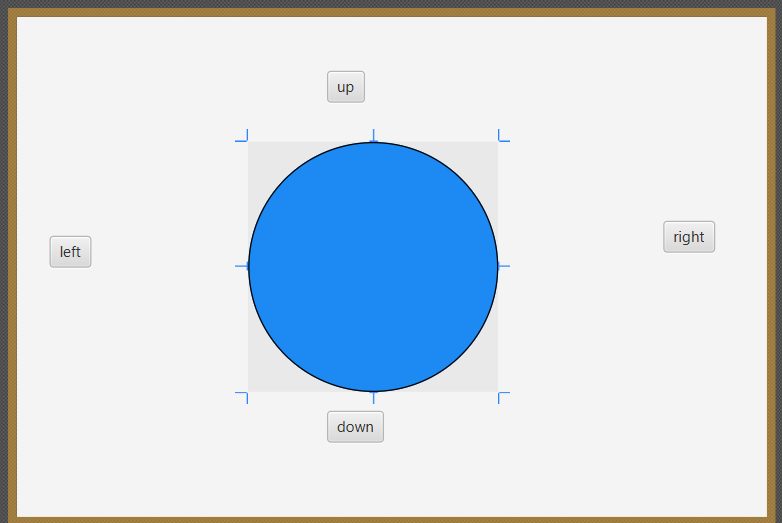
Run your main file if it is not working correctly than refresh your project don’t forget to add VM arguments



**Example for moving circle with buttons**

Add a circle go to left hand menu > under the shapes dropdown > select circle > drag and drop it to your scene





Save this and head back to your controller class

Within your controller class on the top add an annotation @FXML and **import** javafx.fxml.FXML;

It will link your FXML document with your controller class

Now create a cirlcle

**package** application;

**import** javafx.event.ActionEvent;

**import** javafx.fxml.FXML;

**import** javafx.scene.shape.Circle;

**public** **class** controller {

@FXML

**private** Circle myCircle;

//declare the coordinates of circle

**private** **double** x;

**private** **double** y;

**public** **void** up(ActionEvent e) {

myCircle.setCenterY(y=-1);

}

**public** **void** down(ActionEvent e) {

myCircle.setCenterY(y=+1);

}

**public** **void** left(ActionEvent e) {

myCircle.setCenterX(x=-1);

}

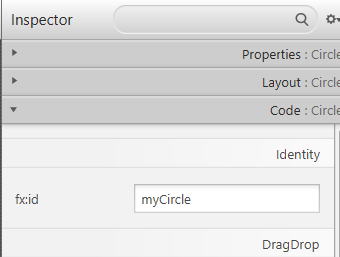
**public** **void** right(ActionEvent e) {

myCircle.setCenterX(x=+1);

}

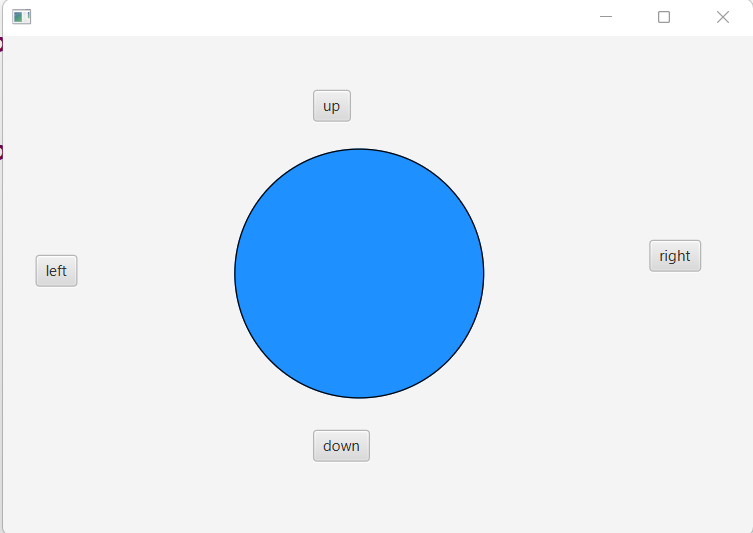
}

head back to scenebuilder > select circle > navigate to right hand menu > open code section > under fxid type the name of your circle reference(myCircle) that you declared in your controller class > save the changes





and run your main class

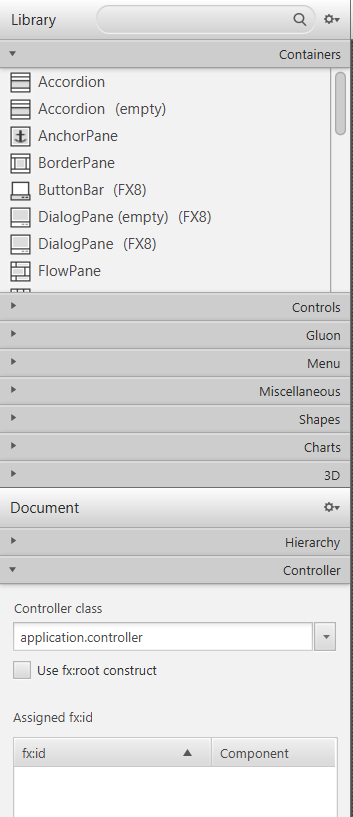


**2nd way:**

**Event Handling using SceneBuilder**

Navigate to Scenebuilder > For each of the button > click on the button there should be a right hand menu -> select code > fxid > give a name to your control object > add On Action method name to handle the event

Add controller that you will create in eclipse java



Go to view > show sample controller skeleton > copy the controller code

Head back to ecclipse > create a controller class> paste the code > add implementation for your events > run your main.java class