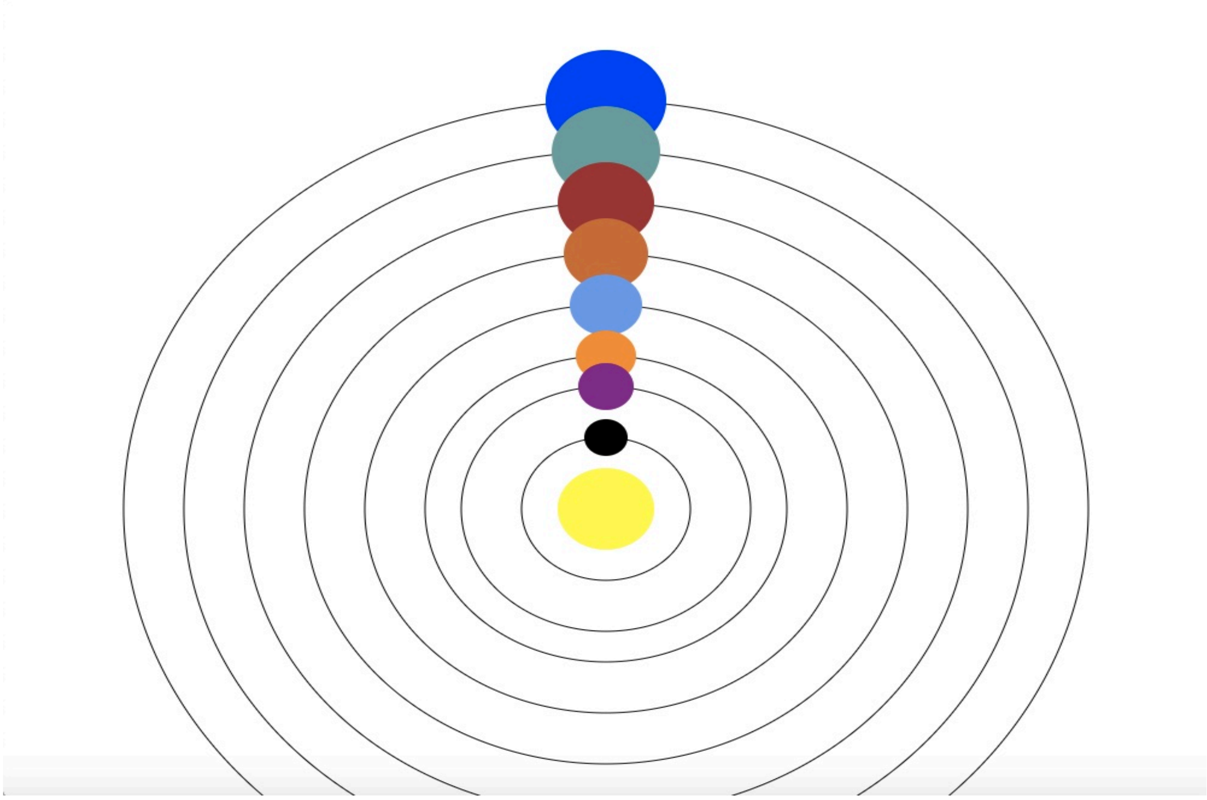
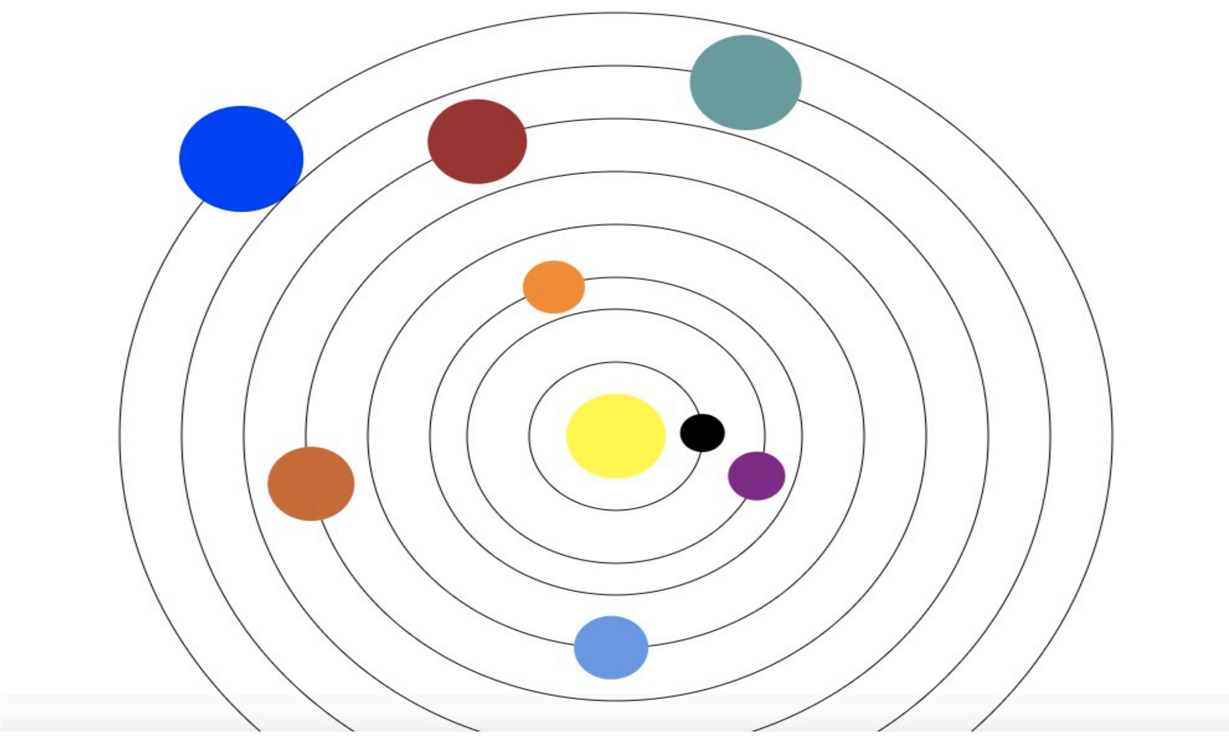
SOLAR SYSTEM USING JAVAFX

1) Press on the space button to play/pause all of the planets.
2) Press on the UP button to enlarge all of the circles , and the down button to shrink them.
3) Click on any circle to play or pause its movement.
4) I couldn’t add aa label on every circle that moves with it (SO I may need your help in this)
5) I also want to know If this code can be written in a more efficient way.
6) I also need some help if we can add the binding properties to this project.

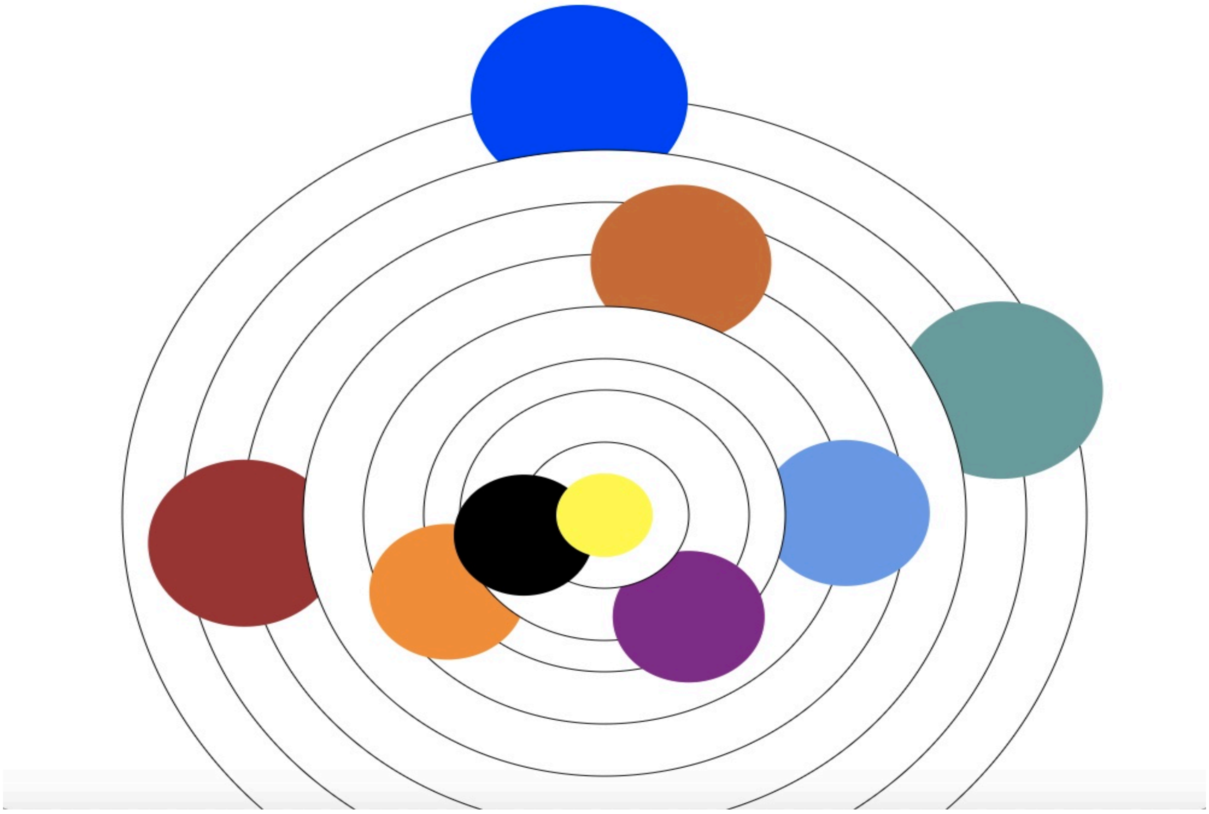
Case 1: Steady mode.



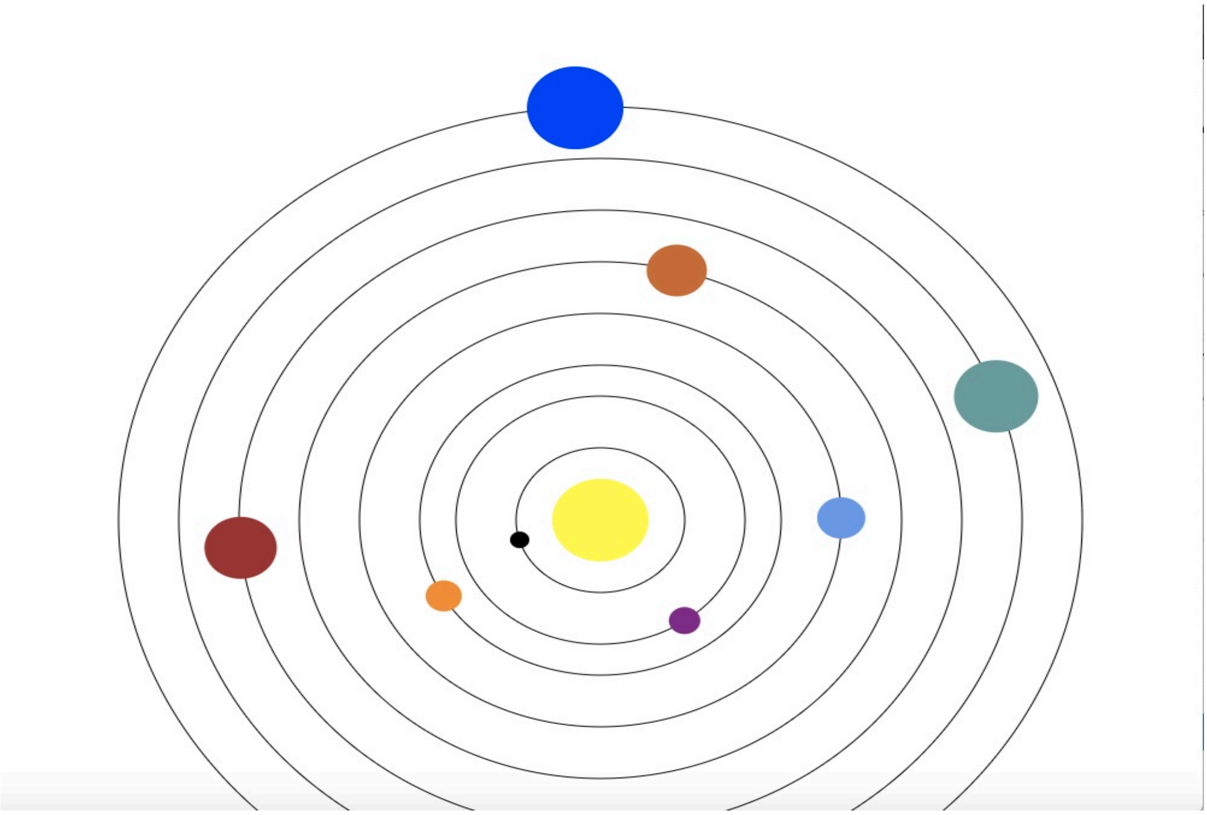
Case 2:Movement mode.



Case 3:Enlarge:



Case 4: Shrinking:



Kindly Find the Code below

```
package javafxapplication2;

import javafx.animation.PathTransition;
import javafx.animation.Timeline;
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;
import javafx.scene.shape.Circle;
import javafx.scene.paint.Color;
import javafx.scene.text.FontPosture;
import javafx.scene.text.FontWeight;
        import javafx.scene.text.*;
import javafx.scene.layout.HBox;
 import javafx.scene.layout.Pane;
 import javafx.scene.image.Image;
 import javafx.scene.image.ImageView;
import javafx.scene.layout.VBox;
import javafx.scene.shape.Line;

import javafx.scene.control.*;
import javafx.scene.layout.FlowPane;
import javafx.scene.shape.Rectangle;
import javafx.scene.control.TextField;
import javafx.scene.input.KeyCode;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.util.Duration;
```

```
/**
 *
 * @author al-farouksaleh
 */
public class JavaFXApplication2 extends Application {
   int x1=1,x2=1,x3=1,x4=1,x5=1,x6=1,x7=1,x8=1,x10=0;
  @Override
  public void start (Stage stage){
  Pane pane = new Pane();

  Circle sun = new Circle (40,Color.YELLOW);
  sun.setCenterX(500);
  sun.setCenterY(500);


  //*********************************
  Circle p11=new Circle (400,Color.WHITE);
  p11.setStroke(Color.BLACK);
  p11.setCenterX(500);
  p11.setCenterY(500);
  Circle p1 = new Circle (50,Color.BLUE);
  p1.setCenterX(500);
  p1.setCenterY(100);

  PathTransition pt1 = new PathTransition ();
  pt1.setDuration(Duration.millis(5000));
  pt1.setPath(p11);
  pt1.setNode(p1);
  pt1.setOrientation(
  PathTransition.OrientationType.ORTHOGONAL_TO_TANGENT);
  pt1.setCycleCount(Timeline.INDEFINITE);
  pt1.setAutoReverse(false);
  EventHandler<MouseEvent> eventHandler = new
EventHandler<MouseEvent>() {
  @Override
  public void handle(MouseEvent e) {
      if (x1%2==1)
     pt1.play();
      else pt1.pause();

      x1++;
  }
};
  p1.addEventFilter(MouseEvent.MOUSE_CLICKED, eventHandler);


//*****************************************************
**


   Circle p22=new Circle (350,Color.WHITE);
  p22.setStroke(Color.BLACK);
  p22.setCenterX(500);
  p22.setCenterY(500);
  Circle p2 = new Circle (45,Color.CADETBLUE);
  p2.setCenterX(500);
  p2.setCenterY(150);

  PathTransition pt2 = new PathTransition ();
  pt2.setDuration(Duration.millis(4000));
  pt2.setPath(p22);
  pt2.setNode(p2);
  pt2.setOrientation(
  PathTransition.OrientationType.ORTHOGONAL_TO_TANGENT);
  pt2.setCycleCount(Timeline.INDEFINITE);
  pt2.setAutoReverse(false);
  EventHandler<MouseEvent> eventHandler2 = new
EventHandler<MouseEvent>() {
  @Override
  public void handle(MouseEvent e) {
     if (x2%2==1)
    pt2.play();
     else pt2.pause();

     x2++;
  }
};
  p2.addEventFilter(MouseEvent.MOUSE_CLICKED, eventHandler2);


//**********************************************************
**
  Circle p33=new Circle (300,Color.WHITE);
  p33.setStroke(Color.BLACK);
  p33.setCenterX(500);
  p33.setCenterY(500);
  Circle p3 = new Circle (40,Color.BROWN);
  p3.setCenterX(500);
  p3.setCenterY(200);

  PathTransition pt3 = new PathTransition ();
  pt3.setDuration(Duration.millis(3000));
  pt3.setPath(p33);
  pt3.setNode(p3);
  pt3.setOrientation(
  PathTransition.OrientationType.ORTHOGONAL_TO_TANGENT);
  pt3.setCycleCount(Timeline.INDEFINITE);
  pt3.setAutoReverse(false);
  EventHandler<MouseEvent> eventHandler3 = new
EventHandler<MouseEvent>() {
  @Override
  public void handle(MouseEvent e) {
     if (x3%2==1)
    pt3.play();
     else pt3.pause();

     x3++;
  }
};
  p3.addEventFilter(MouseEvent.MOUSE_CLICKED, eventHandler3);


//************************************************************
**
  Circle p44=new Circle (250,Color.WHITE);
  p44.setStroke(Color.BLACK);
  p44.setCenterX(500);
  p44.setCenterY(500);
  Circle p4 = new Circle (35,Color.CHOCOLATE);
  p4.setCenterX(500);
  p4.setCenterY(250);

  PathTransition pt4 = new PathTransition ();
  pt4.setDuration(Duration.millis(2000));
  pt4.setPath(p44);
  pt4.setNode(p4);
  pt4.setOrientation(
  PathTransition.OrientationType.ORTHOGONAL_TO_TANGENT);
  pt4.setCycleCount(Timeline.INDEFINITE);
  pt4.setAutoReverse(false);
  EventHandler<MouseEvent> eventHandler4 = new
EventHandler<MouseEvent>() {
  @Override
  public void handle(MouseEvent e) {
     if (x4%2==1)
    pt4.play();
     else pt4.pause();

     x4++;
  }
};
  p4.addEventFilter(MouseEvent.MOUSE_CLICKED, eventHandler4);


//**********************************************************
**
  Circle p55=new Circle (200,Color.WHITE);
  p55.setStroke(Color.BLACK);
  p55.setCenterX(500);
  p55.setCenterY(500);
  Circle p5 = new Circle (30,Color.CORNFLOWERBLUE);
  p5.setCenterX(500);
  p5.setCenterY(300);

  PathTransition pt5 = new PathTransition ();
  pt5.setDuration(Duration.millis(1500));
  pt5.setPath(p55);
  pt5.setNode(p5);
  pt5.setOrientation(
  PathTransition.OrientationType.ORTHOGONAL_TO_TANGENT);
  pt5.setCycleCount(Timeline.INDEFINITE);
  pt5.setAutoReverse(false);
  EventHandler<MouseEvent> eventHandler5 = new
EventHandler<MouseEvent>() {
  @Override
  public void handle(MouseEvent e) {
     if (x5%2==1)
    pt5.play();
     else pt5.pause();

     x5++;
  }
};
  p5.addEventFilter(MouseEvent.MOUSE_CLICKED, eventHandler5);


//**********************************************************
**
  Circle p66=new Circle (150,Color.WHITE);
  p66.setStroke(Color.BLACK);
  p66.setCenterX(500);
  p66.setCenterY(500);
  Circle p6 = new Circle (25,Color.DARKORANGE);
  p6.setCenterX(500);
  p6.setCenterY(350);

  PathTransition pt6 = new PathTransition ();
  pt6.setDuration(Duration.millis(1300));
  pt6.setPath(p66);
  pt6.setNode(p6);
  pt6.setOrientation(
  PathTransition.OrientationType.ORTHOGONAL_TO_TANGENT);
  pt6.setCycleCount(Timeline.INDEFINITE);
  pt6.setAutoReverse(false);
  EventHandler<MouseEvent> eventHandler6 = new
EventHandler<MouseEvent>() {
  @Override
  public void handle(MouseEvent e) {
     if (x6%2==1)
    pt6.play();
     else pt6.pause();

     x6++;
  }
};
  p6.addEventFilter(MouseEvent.MOUSE_CLICKED, eventHandler6);


//***************************************************
**
   Circle p77=new Circle (120,Color.WHITE);
  p77.setStroke(Color.BLACK);
  p77.setCenterX(500);
  p77.setCenterY(500);
  Circle p7 = new Circle (23,Color.DARKMAGENTA);
  p7.setCenterX(500);
  p7.setCenterY(380);

  PathTransition pt7 = new PathTransition ();
  pt7.setDuration(Duration.millis(1200));
  pt7.setPath(p77);
  pt7.setNode(p7);
  pt7.setOrientation(
  PathTransition.OrientationType.ORTHOGONAL_TO_TANGENT);
  pt7.setCycleCount(Timeline.INDEFINITE);
  pt7.setAutoReverse(false);
  EventHandler<MouseEvent> eventHandler7 = new
EventHandler<MouseEvent>() {
  @Override
  public void handle(MouseEvent e) {
     if (x7%2==1)
    pt7.play();
     else pt7.pause();

     x7++;
  }
};
  p7.addEventFilter(MouseEvent.MOUSE_CLICKED, eventHandler7);


//********************************************************
**
  Circle p88=new Circle (70,Color.WHITE);
  p88.setStroke(Color.BLACK);
  p88.setCenterX(500);
  p88.setCenterY(500);
  Circle p8 = new Circle (18,Color.BLACK);
  p8.setCenterX(500);
  p8.setCenterY(430);

  PathTransition pt8 = new PathTransition ();
  pt8.setDuration(Duration.millis(1000));
  pt8.setPath(p88);
  pt8.setNode(p8);
  pt8.setOrientation(
  PathTransition.OrientationType.ORTHOGONAL_TO_TANGENT);
  pt8.setCycleCount(Timeline.INDEFINITE);
  pt8.setAutoReverse(false);
  EventHandler<MouseEvent> eventHandler8 = new
EventHandler<MouseEvent>() {
  @Override
  public void handle(MouseEvent e) {
     if (x8%2==1)
    pt8.play();
     else pt8.pause();

     x8++;
  }
};
  p8.addEventFilter(MouseEvent.MOUSE_CLICKED, eventHandler8);


//*******************************************************
**





pane.getChildren().addAll(p11,p1,p22,p2,p33,p3,p44,p4,p55,p5,p66,
p6,p77,p7,p88,p8,sun);




 Scene scene = new Scene(pane,1000,1000);



 scene.setOnKeyPressed(e->{
   if (e.getCode()==KeyCode.SPACE&& (x10%2==1)){
```

```
pt1.pause();
  pt2.pause();
pt3.pause();
pt4.pause();
pt5.pause();
pt6.pause();
pt7.pause();
pt8.pause();
x10++;
  }
  else if (e.getCode()==KeyCode.SPACE&& (x10%2!=1)) {
pt1.play();
  pt2.play();
pt3.play();
pt4.play();
pt5.play();
pt6.play();
pt7.play();
pt8.play();
x10++;

}
    else if (e.getCode()==KeyCode.UP){
p1.setRadius(p1.getRadius()+5);
  p2.setRadius(p2.getRadius()+5);
p3.setRadius(p3.getRadius()+5);
p4.setRadius(p4.getRadius()+5);
p5.setRadius(p5.getRadius()+5);
p6.setRadius(p6.getRadius()+5);
p7.setRadius(p7.getRadius()+5);
p8.setRadius(p8.getRadius()+5);
}
else if (e.getCode()==KeyCode.DOWN){
p1.setRadius(p1.getRadius()-5);
  p2.setRadius(p2.getRadius()-5);
```

```
p3.setRadius(p3.getRadius()-5);
p4.setRadius(p4.getRadius()-5);
p5.setRadius(p5.getRadius()-5);
p6.setRadius(p6.getRadius()-5);
p7.setRadius(p7.getRadius()-5);
p8.setRadius(p8.getRadius()-5);
}

});

stage.setScene(scene);
stage.setTitle("Student Form");
stage.show();

 }



  /**
   * @param args the command line arguments
   */
  public static void main(String[] args) {
     Application.launch(args);
  }

}
```