

Pendulum

The animation of a pendulum using JavaFx is interesting and a little challenging, because in this project we have to pull together all of the concepts learned in COP 2800. Class Main extends Application which provides all of the required code making the graphical user interface work. For this project we need to create two widgets, one is a circle and the other is a line. Since we want one end of the line to be stationary and the other end to move with the circle, we need to bind the (x, y) coordinates of one endpoint of the line with the (x, y) coordinate of the circle. The code snippet below performs that task for the x coordinates and equivalent code is required for the y coordinate. “l” is the line and “pendulum” is the circle. The pendulum.translateXProperty() method continuously calculates the new positions so that the line moves with the circle.

```
l.startXProperty().bind(pendulum.centerXProperty().add(pendulum.translateXProperty()));
```

CODE:

```
import javafx.animation.*;
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.*;
import javafx.scene.paint.Color;
import javafx.scene.shape.*;
import javafx.stage.Stage;
import javafx.util.Duration;
```

```
public class Main extends Application
{
```

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

```
    public void start(Stage stage)
    {
```

```
        Circle pendulum = new Circle(480, 220, 18);
        pendulum.setStroke(Color.rgb(190, 190,190));
        pendulum.setFill(Color.BLUE);
```

```
        Arc arc = new Arc(300, 260, 180, 100, -20, -130);
        arc.setOpacity(0.0);
```

```
        Line l = new Line(50, 50, 300, 100);
```

```

l.setStroke(Color.rgb(0, 0, 0));
l.setFill(Color.BLACK);

l.startXProperty().bind(pendulum.centerXProperty().
    add(pendulum.translateXProperty()));

l.startYProperty().bind(pendulum.centerYProperty().
    add(pendulum.translateYProperty()));

PathTransition pt = new PathTransition();
pt.setDuration(Duration.millis(5000));
pt.setPath(arc);
pt.setNode(pendulum);
pt.setCycleCount(Timeline.INDEFINITE);
pt.setInterpolator(Interpolator.LINEAR);
pt.setAutoReverse(true);

Pane pane = new Pane();
pane.getChildren().addAll(l, arc, pendulum);

Button startBtn = new Button("Start");
startBtn.setMaxSize(70, 50);
startBtn.setOnAction(e -> {
    System.out.println("hi");
    pt.play();
});

Button pauseBtn = new Button("Pause");
pauseBtn.setMaxSize(70, 50);
pauseBtn.setOnAction(e -> {
    pt.pause();
});

Button stopBtn = new Button("Stop");
stopBtn.setMaxSize(70, 50);
stopBtn.setOnAction(e -> {
    pt.stop();
});

Button exitBtn = new Button("Exit");
exitBtn.setMaxSize(70, 50);
exitBtn.setOnAction(e -> {
    System.exit(0);
});

HBox hb = new HBox();
hb.setPadding(new Insets(10,10,10,10));
hb.setSpacing(7);
hb.getChildren().addAll(startBtn, pauseBtn, stopBtn, exitBtn);

```

```
VBox vb = new VBox();  
vb.setSpacing(20);  
vb.getChildren().addAll(pane, hb);
```

```
Scene scene = new Scene (vb, 500, 500);  
stage.setScene(scene);  
stage.setTitle("Pendulum Animation");  
stage.setWidth(650);  
stage.setHeight(500);  
stage.show();
```

```
}  
}
```