1. **Transition Animations**

   Please modify the code in HW7 problem2. Write a JavaFX program to play the animations. The following requirements are needed:

   1) All animations in HW7 problem2 are divided into five parts:
      a) Parallel Transition
      b) Fade Transition
      c) Rotate Transition
      d) Path Transition
      e) Scale Transition
   2) This program contains two buttons: ”Next” button and “Back” button.
      a) “Next” button: When you press “Next” button, the program will play the next part of animation.
      b) “Back” button: When you press “Back button, the program will play the previous part of animation.

   Note:

   You can use what you learned in homework to implement this program or use the following method to complete it. For example:

```
ScaleTransition scaleTransition = new ……;
……// some setting code
scaleTransition.setNode(rectangle);  // apply the setting to rectangle
scaleTransition.play();
```

   Sample Output

2. Ball Animation

Please modify the code in HW7 problem3 or problem4. Write a JavaFX program to change the color of the circle. The following requirements are needed:

1) After clicking the screen, the color of the circle should be changed randomly and smoothly by using "FillTransition".
2) The color of circle should contain values of red, green, blue and alpha.

Note:

To apply the settings object of FillTransition to an object, you can use the following code:

fillTransition.setShape(c); (Not fillTransition.setNode(c))

3. Sample Game

Please write a JavaFX program with the following requirements:

1) Show one image (in png format) on the right half pane.
2) Add a "Start" button on the pane. After you click the "Start" button, set the "Start" button invisible and start the game.
3) Display the game time on upper left corner of the plane, and the time format is: "minute : second . millisecond".
4) Move the image up and down by pressing key "up" and key "down" correspondingly. The image can not exceed the boundary of the pane.
5) You have to build FXML file by Scene Builder.

Note:

1) To show the image on the screen by ImageView, you can use the following method to complete it:

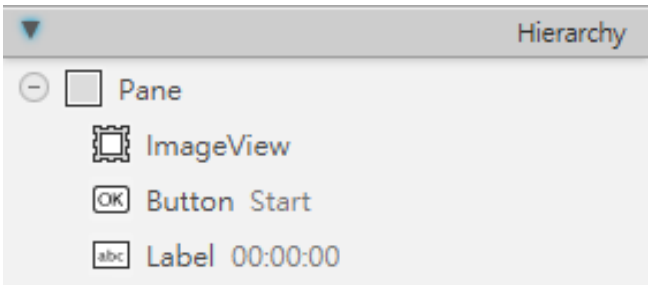
```
ImageView imageView;
Image image = new Image();
image = new Image(getClass().getResource(fileName).toExternalForm());
imageView.setImage(image);
```

2) To set the button invisible, you can use:
   Button button;
   button.setVisible(false);

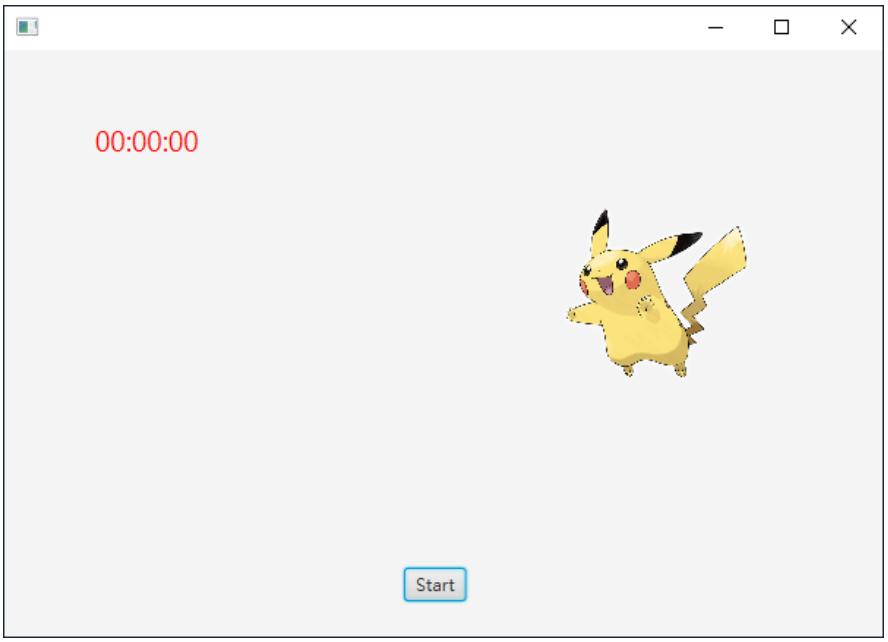
3) To obtain or set the location of the image, you can use
   ImageView imageView;
   Double x = imageView.getLayoutX();
   Double y = imageView.getLayoutY();
   imageView.setLayoutX(x+10);
   imageView.setLayoutY(y+10);

4) When the event function of key-series is invoked, you can use the following code to determine which key is pressed.
   If(e.getCode()==KeyCode.UP) …….

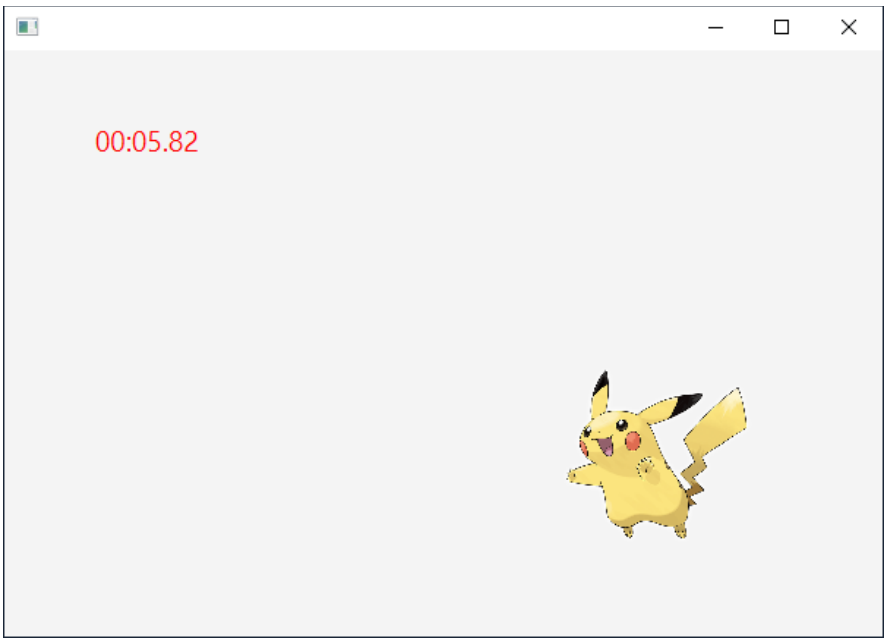
Hierarchy:



Sample Input and Output



After click “Start” button:

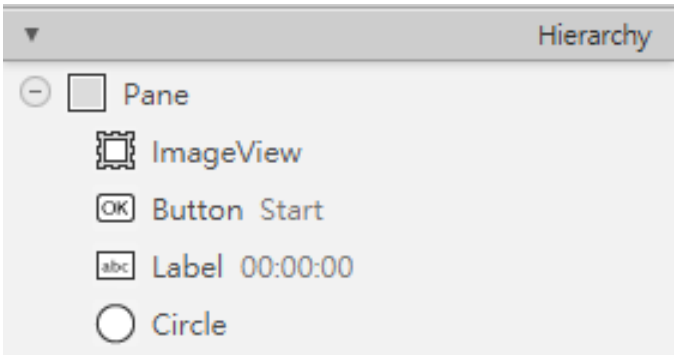


4. Simple Game (Advance)

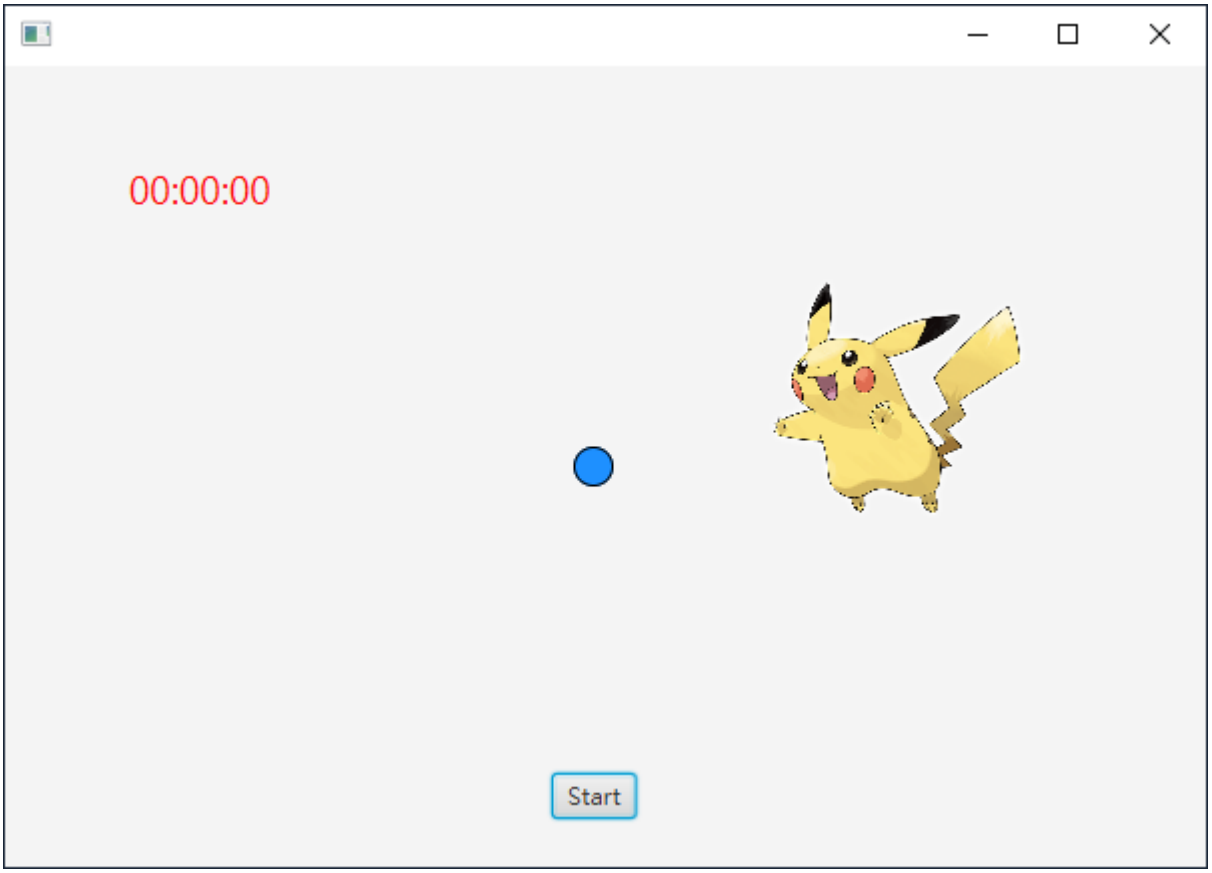
According to the previous problem, you should modify your code with the following requirements:

1) Add a circle on the screen. After clicking “Start” button, the movements of the circle should be the same as HW7 problem3 or problem4.
2) When the image touches the circle, the circle will move in the opposite x direction.

Hierarchy:



Sample Input and Output



Basic Sample code:

```
// Game Controller for student
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.animation.*;
import javafx.scene.input.KeyCode;
import javafx.scene.input.KeyEvent;
import javafx.scene.layout.Pane;
import javafx.scene.shape.Circle;
import javafx.geometry.Bounds;

public class GameController {
    // FXML
    @FXML Button startButton;   // Problem3-2
    @FXML Label clock;          // Problem3-3
    @FXML ImageView imageView;  // Problem3-1
    @FXML Pane pane;            // Problem3
    @FXML Circle c;             // Problem4

    long startTime;             // Problem3-3
    Image image;                // Problem3-1
    AnimationTimer timer;       // Problem3-3 & Problem4

    public void initialize() {
```

```
        // Problem3-1
        // Problem3-3
        // Problem4-1
        // Problem4-2

        // set panel be focusable
        pane.setFocusTraversable(true);
        pane.requestFocus();
    }

    // Problem3-2
    @FXML
    public void buttonClicked(ActionEvent e){

        // set panel be focusable
        pane.setFocusTraversable(true);
        pane.requestFocus();
    }

    // Problem3-4
    @FXML
    public void keyPressed(KeyEvent e){

    }

    // You may define other functions
}
```