```java
import java.util.Comparator;

import javafx.animation.KeyFrame;
import javafx.animation.Timeline;
import javafx.application.Application;
import javafx.beans.property.DoubleProperty;
import javafx.geometry.Pos;
import javafx.scene.Node;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.ScrollBar;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.util.Duration;

public class MultipleBounceBall extends Application {
      @Override // Override the start method in the Application class
      public void start(Stage primaryStage) {
        MultipleBallPane ballPane = new MultipleBallPane();
        ballPane.setStyle("-fx-border-color: yellow");

        Button btAdd = new Button("+");
        Button btSubtract = new Button("-");
        HBox hBox = new HBox(10);
        hBox.getChildren().addAll(btAdd, btSubtract);
        hBox.setAlignment(Pos.CENTER);

        // Add or remove a ball
        btAdd.setOnAction(e -> ballPane.add());
        btSubtract.setOnAction(e -> ballPane.subtract());

        // Pause and resume animation
        ballPane.setOnMousePressed(e -> ballPane.pause());
        ballPane.setOnMouseReleased(e -> ballPane.play());

        // Use a scroll bar to control animation speed
        ScrollBar sbSpeed = new ScrollBar();
        sbSpeed.setMax(20);
        sbSpeed.setValue(10);
        ballPane.rateProperty().bind(sbSpeed.valueProperty());

        BorderPane pane = new BorderPane();
        pane.setCenter(ballPane);
        pane.setTop(sbSpeed);
        pane.setBottom(hBox);

        // Create a scene and place the pane in the stage
        Scene scene = new Scene(pane, 250, 150);
        primaryStage.setTitle("MultipleBounceBall"); // Set the stage title
        primaryStage.setScene(scene); // Place the scene in the stage
        primaryStage.show(); // Display the stage
      }

      private class MultipleBallPane extends Pane {
        private Timeline animation;
        public MultipleBallPane() {
          // Create an animation for moving the ball
          animation = new Timeline(
            new KeyFrame(Duration.millis(50), e -> moveBall()));
          animation.setCycleCount(Timeline.INDEFINITE);
          animation.play(); // Start animation
        }

        public void add() {
          Color color = new Color(Math.random(),
            Math.random(), Math.random(), 0.5);
          getChildren().add(new Ball(30, 30, (double)(Math.random() * (20 - 2 + 1)
+ 2) , color));
        }

        public void subtract() {
          if (getChildren().size() > 0) {
            Ball largestBall = null;
            double maxRadius = 0;
            for(Node node : getChildren()){
              if(node instanceof Ball){
                Ball currentBall =(Ball)node;
                if(currentBall.getRadius() > maxRadius){
                  maxRadius = currentBall.getRadius();
                  largestBall = currentBall;
                  getChildren().remove(largestBall);
                }
              }
          }
        }
      }
          public void play() {
              animation.play();
          }

          public void pause() {
              animation.pause();
          }

          public void increaseSpeed() {
             animation.setRate(animation.getRate() + 0.1);
         }

          public void decreaseSpeed() {
              animation.setRate(
              animation.getRate() > 0 ? animation.getRate() - 0.1 : 0);
          }

          public DoubleProperty rateProperty() {
              return animation.rateProperty();
           }

          protected void moveBall() {
           for (Node node: this.getChildren()) {
            Ball ball = (Ball)node;
               // Check boundaries
              if (ball.getCenterX() < ball.getRadius() ||
                    ball.getCenterX() > getWidth() - ball.getRadius()) {
```

```
                    ball.dx *= -1; // Change ball move direction
                  }
              if (ball.getCenterY() < ball.getRadius() ||
                  ball.getCenterY() > getHeight() - ball.getRadius()) {
                  ball.dy *= -1; // Change ball move direction
                 }

                // Adjust ball position
                  ball.setCenterX(ball.dx + ball.getCenterX());
                  ball.setCenterY(ball.dy + ball.getCenterY());
                }
              }
            }

          class Ball extends Circle {
            private double dx = 1, dy = 1;

             Ball(double x, double y, double radius, Color color) {
                  super(x,y,radius);
                  setFill(color); // Set ball color
              }


  }

  /**
   * The main method is only needed for the IDE with limited
   * JavaFX support. Not needed for running from the command line.
   */
  public static void main(String[] args) {
    launch(args);
  }

}
```