Soluzione 1 – singola classe

```
package it.unitn.disi.ronchet.laboratorio4;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Polygon;
import javafx.scene.shape.Rectangle;
import javafx.scene.shape.Shape;
import javafx.stage.Stage;

import java.io.IOException;

public class MyApplication_self extends Application implements
EventHandler<ActionEvent> {
    int counter=0;
    Shape figure[]=new Shape[3];
    int currentFigure=2;
    int direction=1;
    Button b2=null;
    @Override
    public void start(Stage stage) throws IOException {
        figure[0]=new Rectangle(50,50);
        figure[1]=new Circle(25);
        figure[2]=new Polygon(0.,0.,50,0,25,45);
        for (Shape i:figure){i.setFill(Color.WHITE);
i.setStroke(Color.BLACK);}
        HBox figureBox=new HBox(figure);
        Button b1=new Button("Cambia colore");
        b1.setOnAction(this);
        b2=new Button("=>");
        b2.setOnAction(this);

        HBox buttonBox=new HBox(b1,b2);
        BorderPane root=new BorderPane();
        figureBox.setAlignment(Pos.CENTER);
        BorderPane.setAlignment(figureBox, Pos.CENTER);
        root.setCenter(figureBox);
        buttonBox.setAlignment(Pos.CENTER);
        BorderPane.setAlignment(buttonBox, Pos.BOTTOM_CENTER);
        root.setBottom(buttonBox);
        Scene scene = new Scene(root, 320, 240);
        stage.setTitle("Self");
        stage.setScene(scene);
        stage.show();
    }
    public void nextFigure() {
        System.out.println(currentFigure);
        currentFigure=(currentFigure+direction);
        if (currentFigure<0) currentFigure=2;
        else if (currentFigure>2) currentFigure=0;
        System.out.println(currentFigure);
    }
    public static void main(String[] args) {
        launch();
    }

    @Override
    public void handle(ActionEvent actionEvent) {
        Button b=(Button)actionEvent.getTarget();
        if (b.getText().equals("Cambia colore")) {
            Color randomColor=new
Color(Math.random(),Math.random(),Math.random(),1.0);
            nextFigure();
            figure[currentFigure].setFill(randomColor);
        } else {
            switch (direction) {
                case 1:
                    b2.setText("<=");
                    break;
                case -1:
                    b2.setText("=>");
                    break;
            }
            direction=-direction;
        }
    }
}
```

Soluzione 2 – Listener esterni

```java
package it.unitn.disi.ronchet.laboratorio4;
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Polygon;
import javafx.scene.shape.Rectangle;
import javafx.scene.shape.Shape;
import javafx.stage.Stage;
import java.io.IOException;

public class MyApplication_external extends Application  {
    int counter=0;
    Shape figure[]=new Shape[3];
    int currentFigure=2;
    int direction=1;
    Button b2=null;
    @Override
    public void start(Stage stage) throws IOException {
        figure[0]=new Rectangle(50,50);
        figure[1]=new Circle(25);
        figure[2]=new Polygon(0.,0.,50,0,25,45);
        for (Shape i:figure){i.setFill(Color.WHITE);
i.setStroke(Color.BLACK);}
        HBox figureBox=new HBox(figure);
        Button b1=new Button("cambia colore");
        b1.setOnAction(new Controller1(this));
        b2=new Button("=>");
        b2.setOnAction(new Controller2(this));
        HBox buttonBox=new HBox(b1,b2);
        BorderPane root=new BorderPane();
        figureBox.setAlignment(Pos.CENTER);
        BorderPane.setAlignment(figureBox, Pos.CENTER);
        root.setCenter(figureBox);
        buttonBox.setAlignment(Pos.CENTER);
        BorderPane.setAlignment(buttonBox, Pos.BOTTOM_CENTER);
        root.setBottom(buttonBox);
        Scene scene = new Scene(root, 320, 240);
        stage.setTitle("External");
        stage.setScene(scene);
        stage.show();
    }
    public void nextFigure() {
        System.out.println(currentFigure);
        currentFigure=(currentFigure+direction);
        if (currentFigure<0) currentFigure=2;
        else if (currentFigure>2) currentFigure=0;
        System.out.println(currentFigure);
    }
    public static void main(String[] args) {
        launch();
    }
}
```

```java
package it.unitn.disi.ronchet.laboratorio4;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.paint.Color;

public class Controller1 implements EventHandler<ActionEvent> {
    MyApplication_external app=null;
    Controller1(MyApplication_external app) {
        this.app=app;
    }
    @Override
    public void handle(ActionEvent actionEvent) {
        Color randomColor=new
Color(Math.random(),Math.random(),Math.random(),1.0);
        app.nextFigure();
        app.figure[app.currentFigure].setFill(randomColor);
    }
}
```

```java
package it.unitn.disi.ronchet.laboratorio4;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;

public class Controller2 implements EventHandler<ActionEvent> {
    MyApplication_external app=null;
    Controller2(MyApplication_external app) {
        this.app=app;
    }
    @Override
    public void handle(ActionEvent actionEvent) {
        switch (app.direction) {
            case 1:
                app.b2.setText("<=");
                break;
            case -1:
                app.b2.setText("=>");
                break;
            default:
                System.out.println("Non deve mai arrivare qui!");
                System.exit(1);
        }
        app.direction=-app.direction;
    }
}
```

Soluzione 3 – Classi interne

```
package it.unitn.disi.ronchet.laboratorio4;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Polygon;
import javafx.scene.shape.Rectangle;
import javafx.scene.shape.Shape;
import javafx.stage.Stage;

import java.io.IOException;

public class MyApplication_internal extends Application  {
    int counter=0;
    Shape figure[]=new Shape[3];
    int currentFigure=2;
    int direction=1;
    Button b2=null;
    @Override
    public void start(Stage stage) throws IOException {
        figure[0]=new Rectangle(50,50);
        figure[1]=new Circle(25);
        figure[2]=new Polygon(0.,0.,50,0,25,45);
        for (Shape i:figure){i.setFill(Color.WHITE);
i.setStroke(Color.BLACK);}
        HBox figureBox=new HBox(figure);
        Button b1=new Button("cambia colore");
        b1.setOnAction(new Controller1());
        b2=new Button("=>");
        b2.setOnAction(new Controller2());
        HBox buttonBox=new HBox(b1,b2);
        BorderPane root=new BorderPane();
        figureBox.setAlignment(Pos.CENTER);
        BorderPane.setAlignment(figureBox, Pos.CENTER);
        root.setCenter(figureBox);
        buttonBox.setAlignment(Pos.CENTER);
        BorderPane.setAlignment(buttonBox, Pos.BOTTOM_CENTER);
        root.setBottom(buttonBox);
        Scene scene = new Scene(root, 320, 240);
        stage.setTitle("Internal");
        stage.setScene(scene);
        stage.show();
    }
    public void nextFigure() {
        System.out.println(currentFigure);
        currentFigure=(currentFigure+direction);
        if (currentFigure<0) currentFigure=2;
        else if (currentFigure>2) currentFigure=0;
        System.out.println(currentFigure);
    }
    public static void main(String[] args) {
        launch();
    }
    class Controller1 implements EventHandler<ActionEvent> {
        @Override
        public void handle(ActionEvent actionEvent) {
            Color randomColor=new
Color(Math.random(),Math.random(),Math.random(),1.0);
            nextFigure();
            figure[currentFigure].setFill(randomColor);
        }
    }
    class Controller2 implements EventHandler<ActionEvent> {
        @Override
        public void handle(ActionEvent actionEvent) {
            switch (direction) {
                case 1:
                    b2.setText("<=");
                    break;
                case -1:
                    b2.setText("=>");
                    break;
            }
            direction=-direction;
        }
    }


}
```

Soluzione 4 – Classi interne anonime

```
package it.unitn.disi.ronchet.laboratorio4;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Polygon;
import javafx.scene.shape.Rectangle;
import javafx.scene.shape.Shape;
import javafx.stage.Stage;

import java.io.IOException;

public class MyApplication_anonymous extends Application  {
    int counter=0;
    Shape figure[]=new Shape[3];
    int currentFigure=2;
    int direction=1;
    Button b2=null;
    @Override
    public void start(Stage stage) throws IOException {
        figure[0]=new Rectangle(50,50);
        figure[1]=new Circle(25);
        figure[2]=new Polygon(0.,0.,50,0,25,45);
        for (Shape i:figure){i.setFill(Color.WHITE);
i.setStroke(Color.BLACK);}
        HBox figureBox=new HBox(figure);
        Button b1=new Button("cambia colore");
        b1.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent actionEvent) {
                Color randomColor=new
Color(Math.random(),Math.random(),Math.random(),1.0);
                nextFigure();
                figure[currentFigure].setFill(randomColor);
            }
        });
        b2=new Button("=>");
        b2.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent actionEvent) {
                switch (direction) {
                    case 1:
                        b2.setText("<=");
                        break;
                    case -1:
                        b2.setText("=>");
                        break;
                }
                direction=-direction;
            }
        });
        HBox buttonBox=new HBox(b1,b2);
```

```java
        BorderPane root=new BorderPane();
        figureBox.setAlignment(Pos.CENTER);
        BorderPane.setAlignment(figureBox, Pos.CENTER);
        root.setCenter(figureBox);
        buttonBox.setAlignment(Pos.CENTER);
        BorderPane.setAlignment(buttonBox, Pos.BOTTOM_CENTER);
        root.setBottom(buttonBox);
        Scene scene = new Scene(root, 320, 240);
        stage.setTitle("Anonymous");
        stage.setScene(scene);
        stage.show();
    }
    public void nextFigure() {
        System.out.println(currentFigure);
        currentFigure=(currentFigure+direction);
        if (currentFigure<0) currentFigure=2;
        else if (currentFigure>2) currentFigure=0;
        System.out.println(currentFigure);
    }
    public static void main(String[] args) {
        launch();
    }
}
```