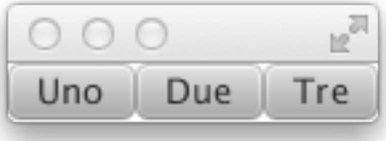
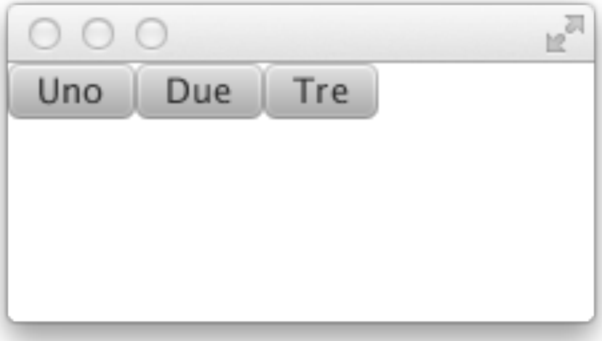
Posizionamento automatico: Layouts di base

http://docs.oracle.com/javafx/2/layout/jfxpub-layout.htm

Layout: HBox

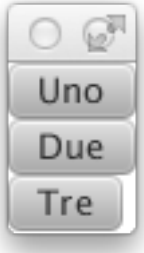
```
public class Layout1 extends Application {
    public void start(Stage stage) {
        Pane layout=new HBox();
        layout.getChildren().add(new Button("Uno"));
        layout.getChildren().add(new Button("Due"));
        layout.getChildren().add(new Button("Tre"));
        Group root = new Group(layout);
        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }}…}
```



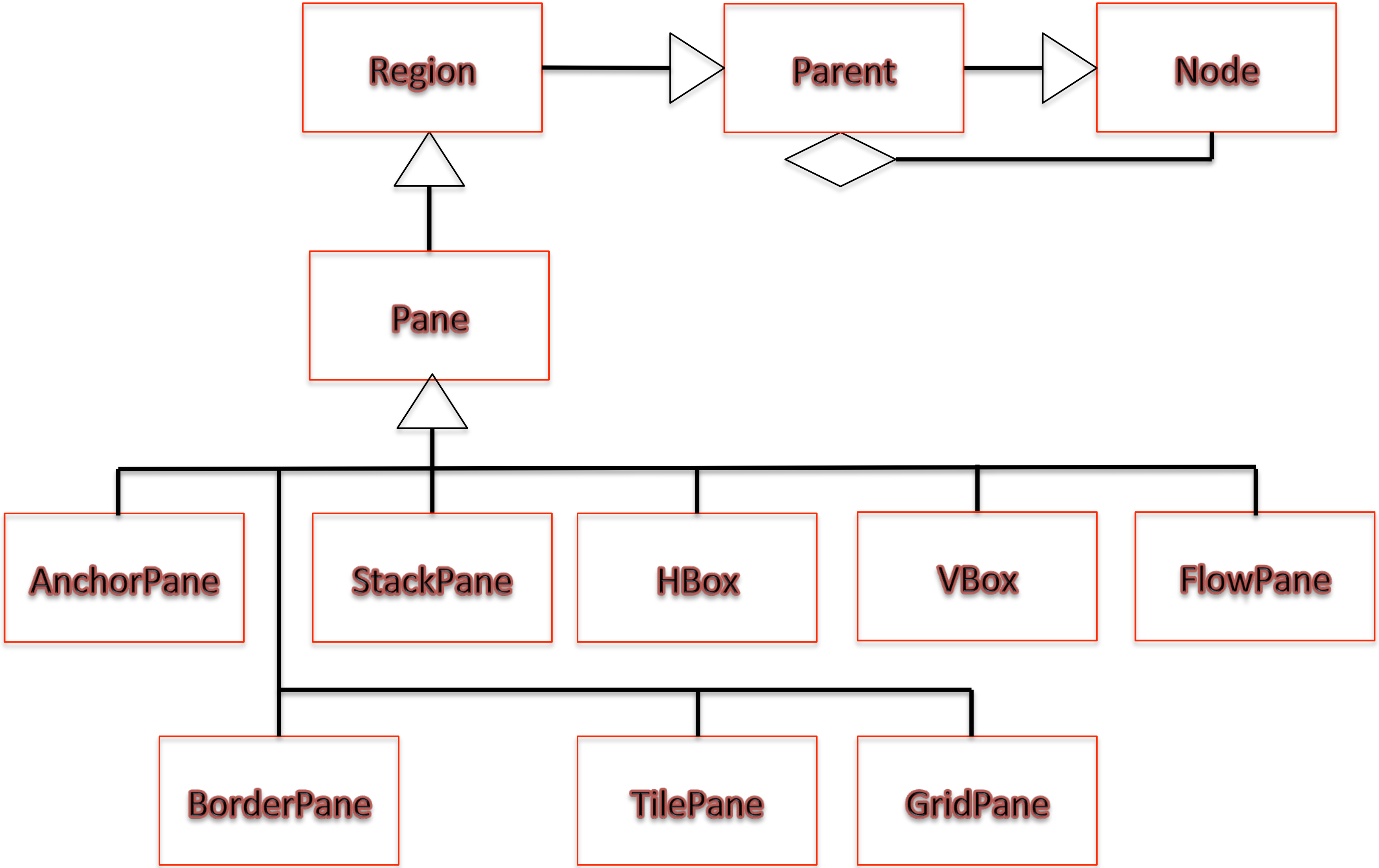


Layout: VBox

```
public class Layout1 extends Application {
    public void start(Stage stage) {
        Pane layout=new VBox();
        layout.getChildren().add(new Button("Uno"));
        layout.getChildren().add(new Button("Due"));
        layout.getChildren().add(new Button("Tre"));
        Group root = new Group(layout);
        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }}…}
```



MediaView - Media



Region

Parent

Node

Pane

AnchorPane

StackPane

HBox

VBox

FlowPane

BorderPane

TilePane

GridPane

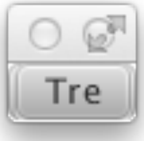
Container classes that automate common layout models

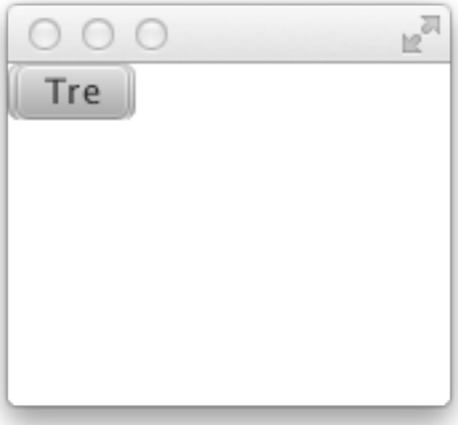
- The HBox class arranges its content nodes horizontally in a single row.
- The VBox class arranges its content nodes vertically in a single column.
- The StackPane class places its content nodes in a back-to-front single stack.
- The TilePane class places its content nodes in uniformly sized layout cells or tiles
- The FlowPane class arranges its content nodes in either a horizontal or vertical “flow,” wrapping at the specified width (for horizontal) or height (for vertical) boundaries.
- The BorderPane class lays out its content nodes in the top, bottom, right, left, or center region.
- The AnchorPane class enables developers to create anchor nodes to the top, bottom, left side, or center of the layout.
- The GridPane class enables the developer to create a flexible grid of rows and columns in which to lay out content nodes.

To achieve a desired layout structure, different containers can be nested within a JavaFX application.

Layout: StackPane

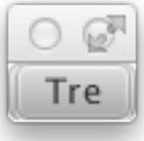
```
public class Layout1 extends Application {
    public void start(Stage stage) {
        StackPane layout=new StackPane();
        layout.getChildren().add(new Button("Uno"));
        layout.getChildren().add(new Button("Due"));
        layout.getChildren().add(new Button("Tre"));
        Group root = new Group(layout);
        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }}…}
```

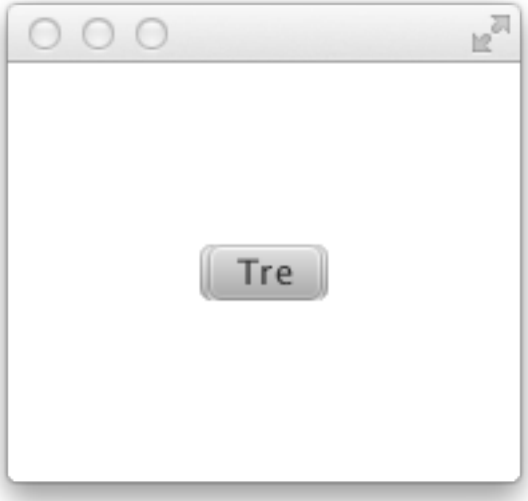




Layout: StackPane

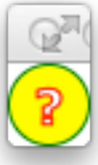
```
public class Layout1 extends Application {
    public void start(Stage stage) {
        StackPane layout=new StackPane();
        layout.getChildren().add(new Button("Uno"));
        layout.getChildren().add(new Button("Due"));
        layout.getChildren().add(new Button("Tre"));
        ~~//Group root = new Group(layout);~~
        ~~//Scene scene = new Scene(root);~~
        Scene scene = new Scene(layout);
        stage.setScene(scene);
        stage.show();
    }}…}
```

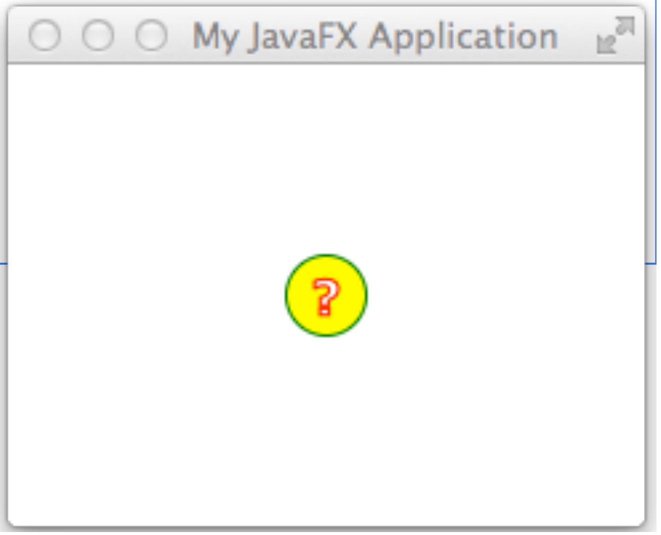




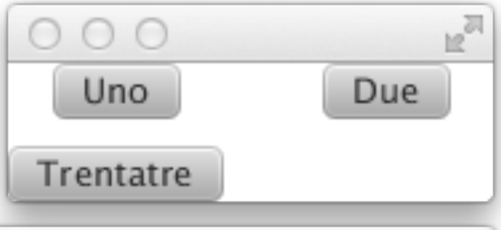
Layout: StackPane

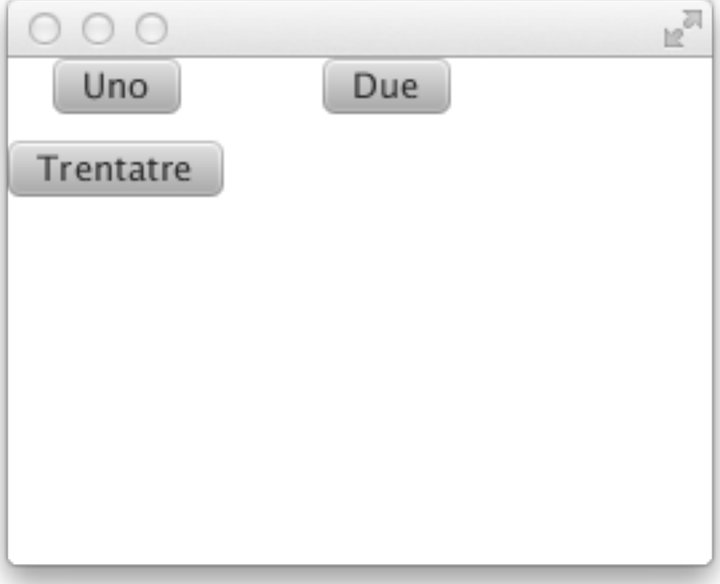
```
public class Layout1 extends Application {
    public void start(Stage stage) {
        StackPane stack = new StackPane();
        Circle helpIcon = new Circle(15, 15, 15);
        helpIcon.setFill(Color.YELLOW);
        helpIcon.setStroke(Color.GREEN);
        Text helpText = new Text("?");
        helpText.setFont(Font.font("Verdana", FontWeight.BOLD, 18));
        helpText.setFill(Color.WHITE);
        helpText.setStroke(Color.RED);
        stack.getChildren().addAll(helpIcon, helpText);
        stack.setAlignment(Pos.CENTER);
        Scene scene = new Scene(stack);
        stage.setTitle("My JavaFX Application");
        stage.setScene(scene);
        stage.show();
    }
    }}…}
```





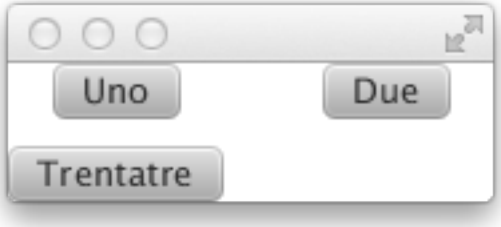
Layout: TilePane



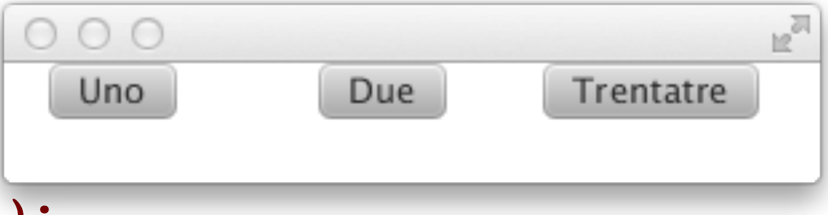


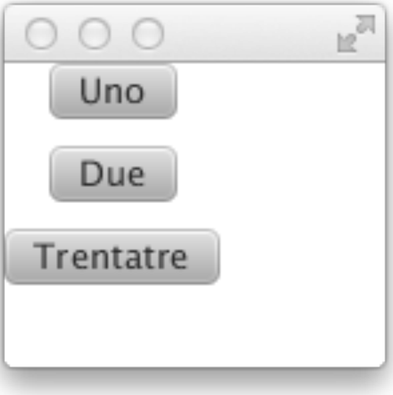
```
public class Layout1 extends Application {
    public void start(Stage stage) {
        //Pane layout=new HBox();
        //Pane layout=new VBox();
        //StackPane layout=new StackPane();
        TilePane layout=new TilePane();
        layout.setVgap(10);
        layout.setHgap(20);
        layout.setPrefColumns(2);
        layout.getChildren().add(new Button("Uno"));
        layout.getChildren().add(new Button("Due"));
        layout.getChildren().add(new Button("Trentatre"));
        Group root = new Group(layout);
        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }}…}
```

Layout: TilePane



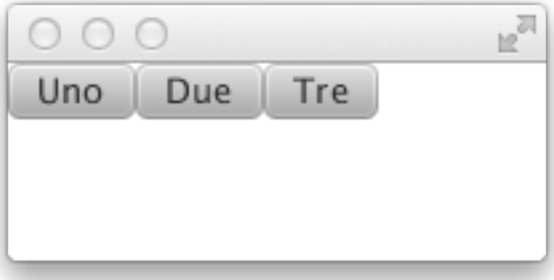
```
public class Layout1 extends Application {
    public void start(Stage stage) {
        //Pane layout=new HBox();
        //Pane layout=new VBox();
        //StackPane layout=new StackPane();
        TilePane layout=new TilePane();
        layout.setVgap(10);
        layout.setHgap(20);
        layout.setPrefColumns(2);
        layout.getChildren().add(new Button("Uno"));
        layout.getChildren().add(new Button("Due"));
        layout.getChildren().add(new Button("Trentatre"));
        ~~//Group root = new Group(layout);~~
        ~~//Scene scene = new Scene(root);~~
        Scene scene = new Scene(layout);
        stage.setScene(scene);
        stage.show();
    }}…}
```

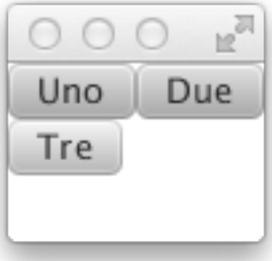




FlowPane

```
public class Layout1 extends Application {
    public void start(Stage stage) {
        final FlowPane layout=new FlowPane();
        layout.setPrefWrapLength(100);
        layout.getChildren().add(new Button("Uno"));
        layout.getChildren().add(new Button("Due"));
        layout.getChildren().add(new Button("Tre"));
        Scene scene = new Scene(layout);
        stage.setScene(scene);
        stage.show();
    }…
}
```

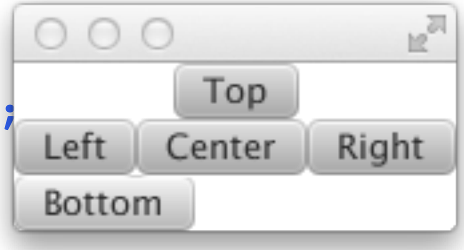


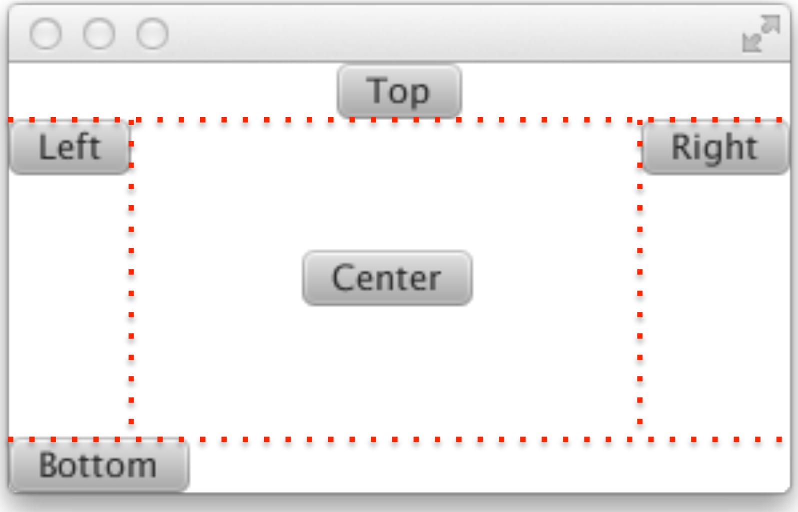


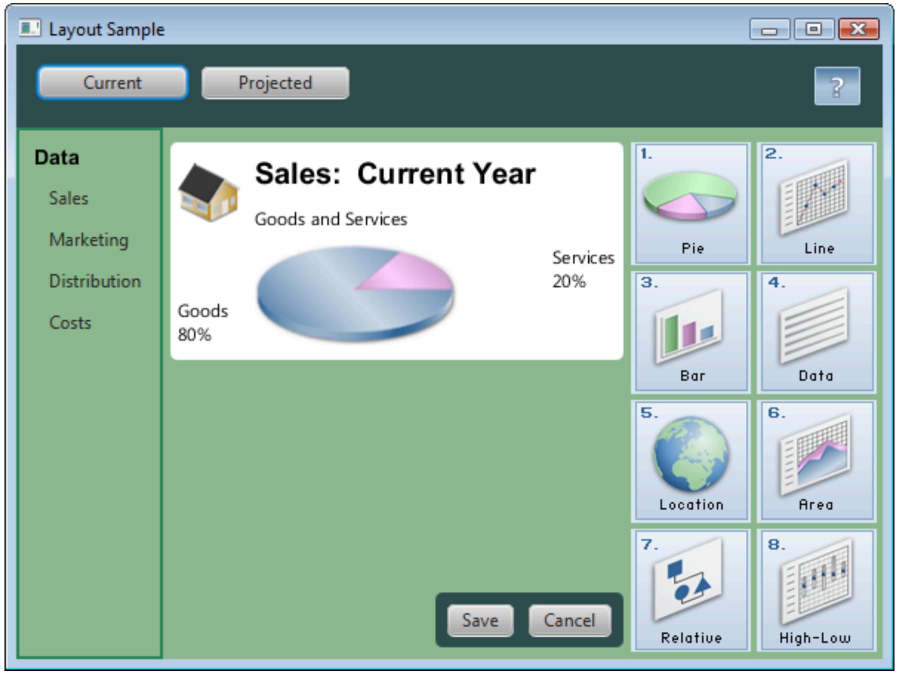
Posizionamento automatico:
Layouts avanzati

BorderPane

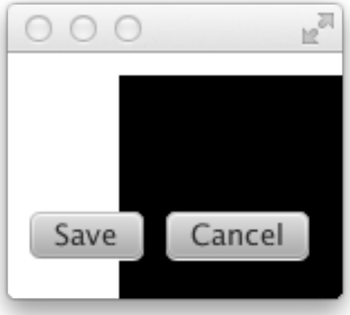
```
public class Layout1 extends Application {
    public void start(Stage stage) {
        BorderPane layout=new BorderPane();
        Button top=new Button("Top");
        BorderPane.setAlignment(top, Pos.TOP_CENTER);
        layout.setTop(top);
        layout.setBottom(new Button("Bottom"));
        layout.setLeft(new Button("Left"));
        layout.setRight(new Button("Right"));
        layout.setCenter(new Button("Center"));
        Scene scene = new Scene(layout);
        stage.setScene(scene);
        stage.show();
    }
…
}
```



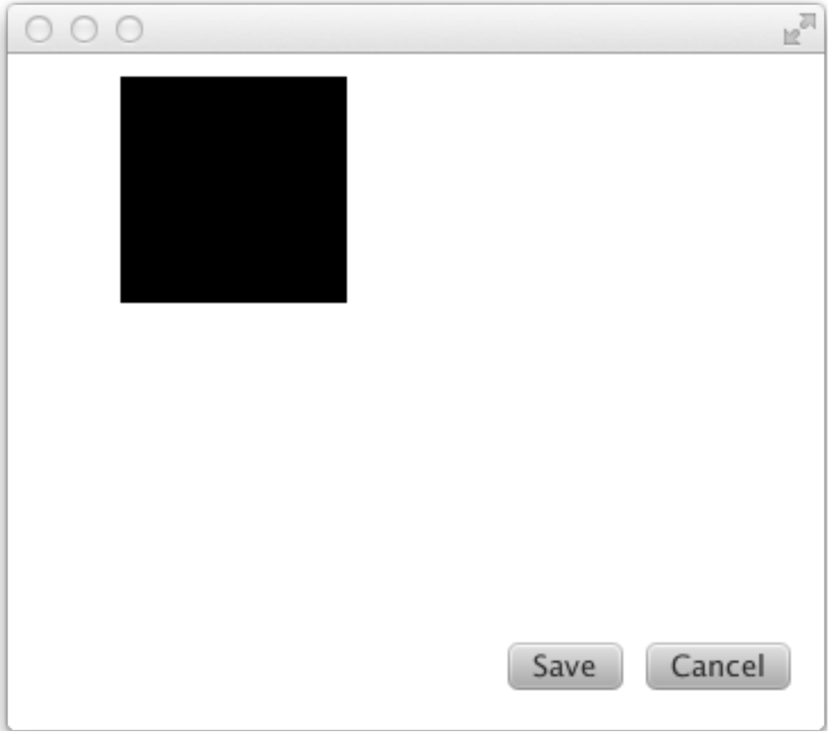




AnchorPane

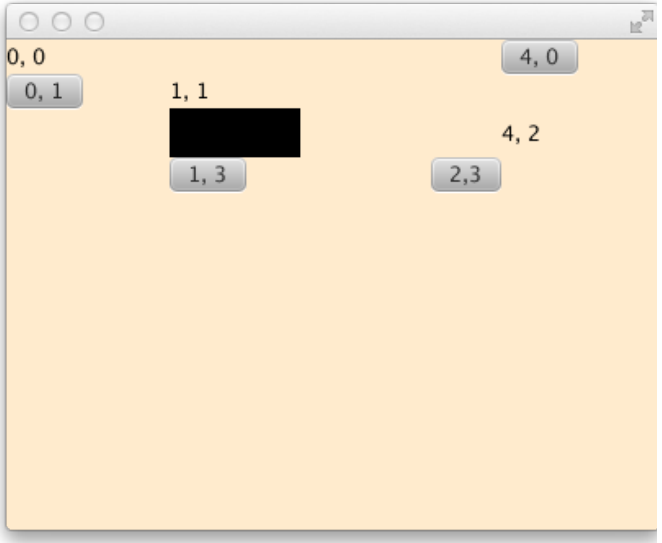


```
public void start(Stage stage) {
    AnchorPane anchorpane = new AnchorPane();
    Button buttonSave = new Button("Save");
    Button buttonCancel = new Button("Cancel");
    HBox hb = new HBox();
    hb.setPadding(new Insets(0, 10, 10, 10));
    hb.setSpacing(10);
    hb.getChildren().addAll(buttonSave, buttonCancel);
    Rectangle r=new Rectangle(100,100);
    anchorpane.getChildren().addAll(r,hb);
    AnchorPane.setBottomAnchor(hb, 8.0);
    AnchorPane.setRightAnchor(hb, 5.0);
    AnchorPane.setTopAnchor(r, 10.0);
    AnchorPane.setLeftAnchor(r, 50.0);
    Scene scene = new Scene(anchorpane);
    stage.setScene(scene);
    stage.show();
}
```

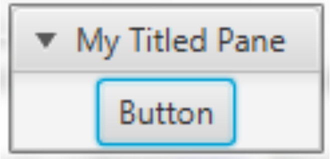


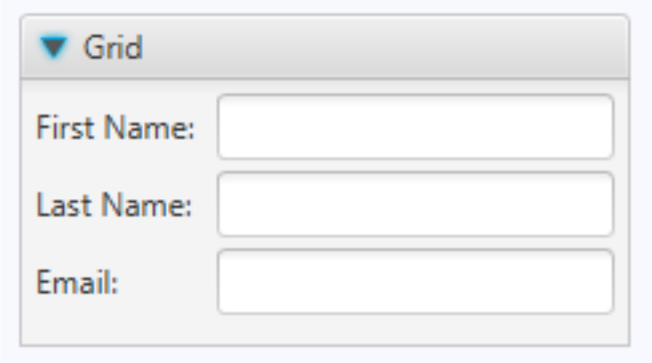
GridPane

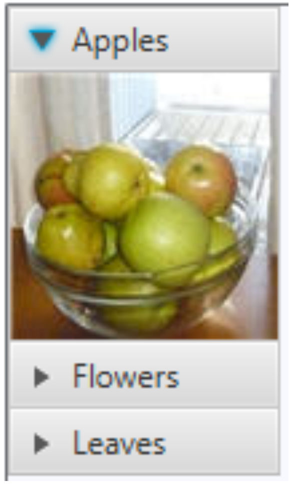
```
public void start(Stage primaryStage) {
        double width = 400;
        double height = 300;
        GridPane gridPane = new GridPane();
        Scene scene = new Scene(gridPane,
           width, height, Color.BLANCHEDALMOND);
        gridPane.add(new Text("0, 0"), 0, 0);
        gridPane.add(new Button("0, 1"), 0, 1);
        gridPane.add(new Text("1, 1"), 1, 1);
        Rectangle r=new Rectangle(80,30);
        gridPane.add(r, 1, 2);
        gridPane.add(new Button("1, 3"), 1, 3);
        gridPane.add(new Button("2,3"), 2, 3);
        gridPane.add(new Button("4, 0"), 4, 0);
        gridPane.add(new Text("4, 2"), 4, 2);
        ColumnConstraints column1 = new ColumnConstraints(100);
        ColumnConstraints column2 = new ColumnConstraints();
        column2.setPercentWidth(40);
        column2.setHgrow(Priority.ALWAYS);
        gridPane.getColumnConstraints().addAll(column1, column2);
        primaryStage.setScene(scene);
        primaryStage.show();
}
```



TitledPane



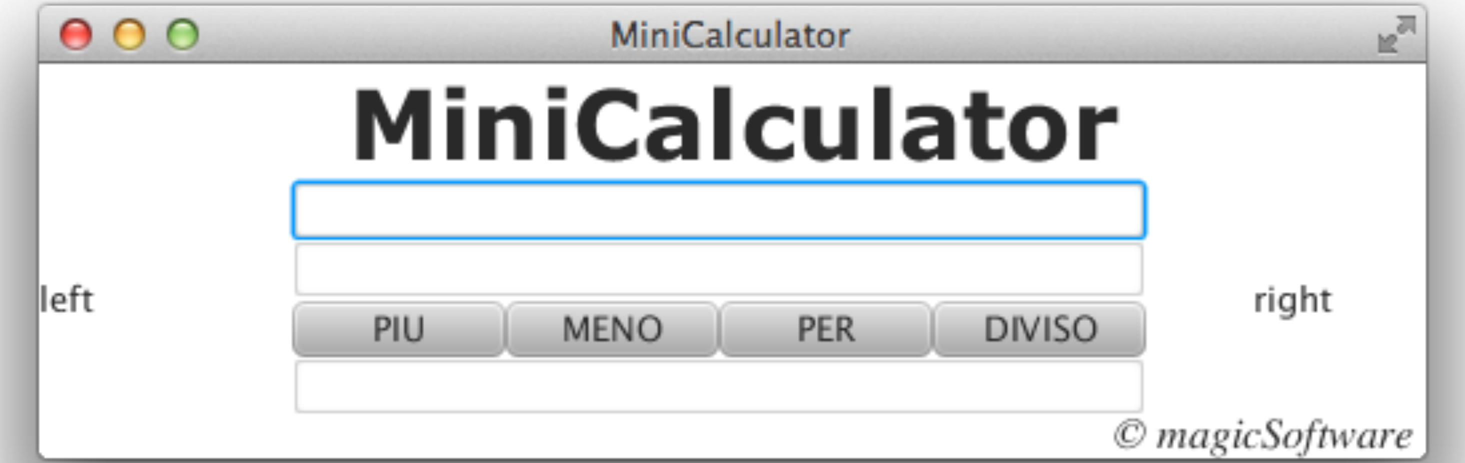




Accordion

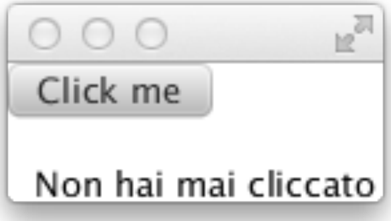
http://docs.oracle.com/javase/8/javafx/user-interface-tutorial/accordion-titledpane.htm#CACGBAHI

Esercizio

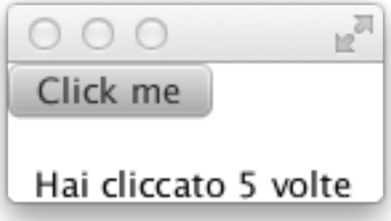

MiniCalculator
MiniCalculator
left
right
PIU
MENO
PER
DIVISO
© magicSoftware

Come organizzare gli ascoltatori/osservatori

Listener Esterno



Non hai mai cliccato



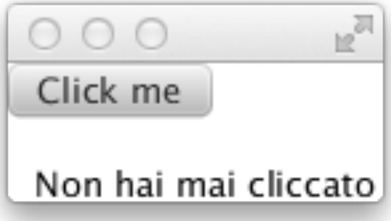
Hai cliccato 5 volte

```
public class AppWithEvents1 extends Application {
    Text text=null;
    public void start(Stage stage) {
      text=new Text(10,50,"Non hai mai cliccato ");
      Button btn = new Button();
      btn.setText("Click me");
      Listener a=new Listener(this);
      btn.addEventHandler(ActionEvent.ACTION, a);
      Group root = new Group(btn);
      root.getChildren().add(text);
      Scene scene = new Scene(root);
      stage.setScene(scene);
      stage.show();
    }
    public void updateText(int n){
      text.setText("Hai cliccato "+n
         +" volte");
    }
    public static void main(String[] args) {
      Application.launch(args);
    }}
```

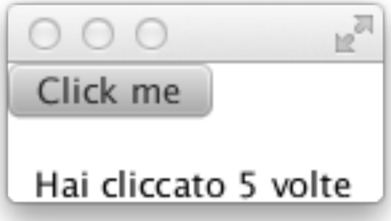
```
class Listener
     implements EventHandler{
  AppWithEvents1 awe=null;
  int counter=0;
  Listener1(AppWithEvents1 a){
       awe=a;
  }
  public void handle(Event t) {
      awe.updateText(++counter);
}
}
```

Listener Interno

```
public class AppWithEvents1 extends Application {
    Text text=null;
    public void start(Stage stage) {
      text=new Text(10,50,"Non hai mai cliccato ");
      Button btn = new Button();
      btn.setText("Click me");
      Listener a=new Listener(~~this~~);
      btn.addEventHandler(ActionEvent.ACTION, a);
      Group root = new Group(btn);
      root.getChildren().add(text);
      Scene scene = new Scene(root);
      stage.setScene(scene);
      stage.show();
    }
    public void updateText(int n){
      text.setText("Hai cliccato "+n
         +" volte");
    }
    public static void main(String[] args) {
      Application.launch(args);
    }}
```



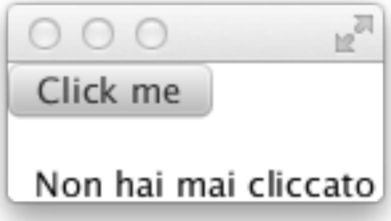
Non hai mai cliccato



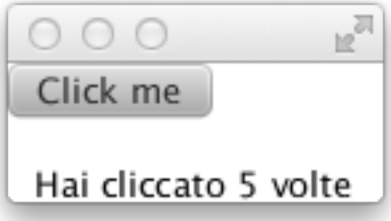
Hai cliccato 5 volte

```
class Listener
     implements EventHandler{
   ~~AppWithEvents1 awe=null;~~
   int counter=0;
   ~~Listener1(AppWithEvents1 a){~~
           ~~awe=a;~~
   ~~}~~
   public void handle(Event t) {
       ~~awe.~~updateText(++counter);
   }
}
```

Listener Interno



Non hai mai cliccato



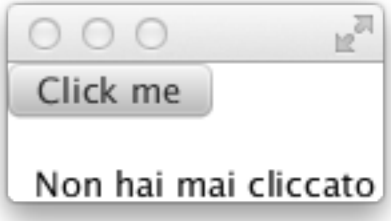
Hai cliccato 5 volte

```
public class AppWithEvents1 extends Application {
    Text text=null;
    public void start(Stage stage) {
      text=new Text(10,50,"Non hai mai cliccato ");
      Button btn = new Button();
      btn.setText("Click me");
      Listener1 a=new Listener();
      btn.addEventHandler(ActionEvent.ACTION, a);
      Group root = new Group(btn);
      root.getChildren().add(text);
      Scene scene = new Scene(root);
      stage.setScene(scene);
      stage.show();
    }
    class Listener
       implements EventHandler{
       int counter=0;
       public void handle(Event t) {
           updateText(++counter);
    }}
```

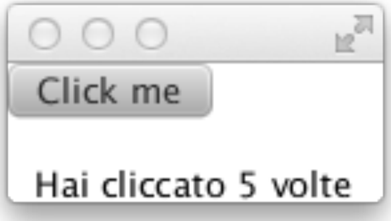
```
public void updateText(int n){
      text.setText("Hai cliccato"
         +n+" volte");
  }
  public static void main(
       String[] args) {
    Application.launch(args);
  }}
```

Self Listener

```
public class AppWithEvents
    extends Application implements EventHandler {
    Text text=null;
    int counter=0;
    public void start(Stage stage) {
      text=new Text(10,50,"Non hai mai cliccato ");
      Button btn = new Button();
      btn.setText("Click me");
      btn.addEventHandler(ActionEvent.ACTION, this);
      Group root = new Group(btn);
      root.getChildren().add(text);
      Scene scene = new Scene(root);
      stage.setScene(scene);
      stage.show();
    }
    public void handle(Event t) {
           updateText(++counter);
    }
```



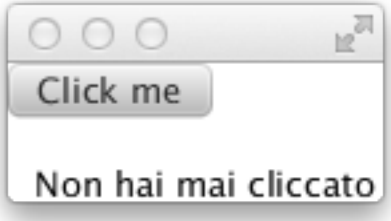
Non hai mai cliccato



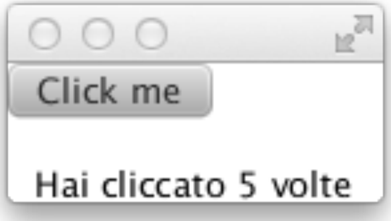
Hai cliccato 5 volte

```
public void updateText(int n){
      text.setText("Hai cliccato"
         +n+" volte");
    }
    public static void main(
         String[] args) {
      Application.launch(args);
    }}
```

Listener Interno Anonimo



Non hai mai cliccato



Hai cliccato 5 volte

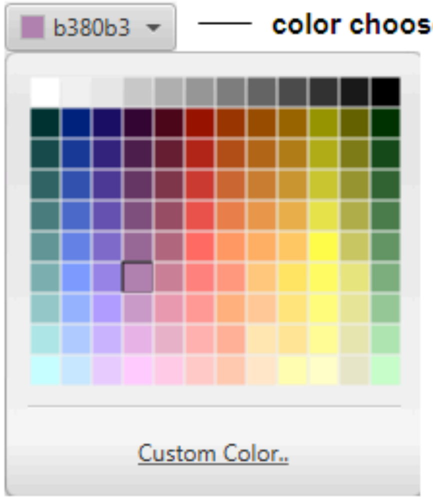
```
public class AppWithEvents1 extends Application {
    Text text=null;
    public void start(Stage stage) {
      text=new Text(10,50,"Non hai mai cliccato ");
      Button btn = new Button();
      btn.setText("Click me");
      Listener1 a=new EventHandler(){
       int counter=0;
       public void handle(Event t) {
           updateText(++counter);
      }};
      btn.addEventHandler(ActionEvent.ACTION, a);
      Group root = new Group(btn);
      root.getChildren().add(text);
      Scene scene = new Scene(root);
      stage.setScene(scene);
      stage.show();
    }
```

```
public void updateText(int n){
      text.setText("Hai cliccato"
         +n+" volte");
    }
    public static void main(
         String[] args) {
      Application.launch(args);
    }}
```

Due questioni:

- ColorPicker
- Convenience Methods

ColorPicker

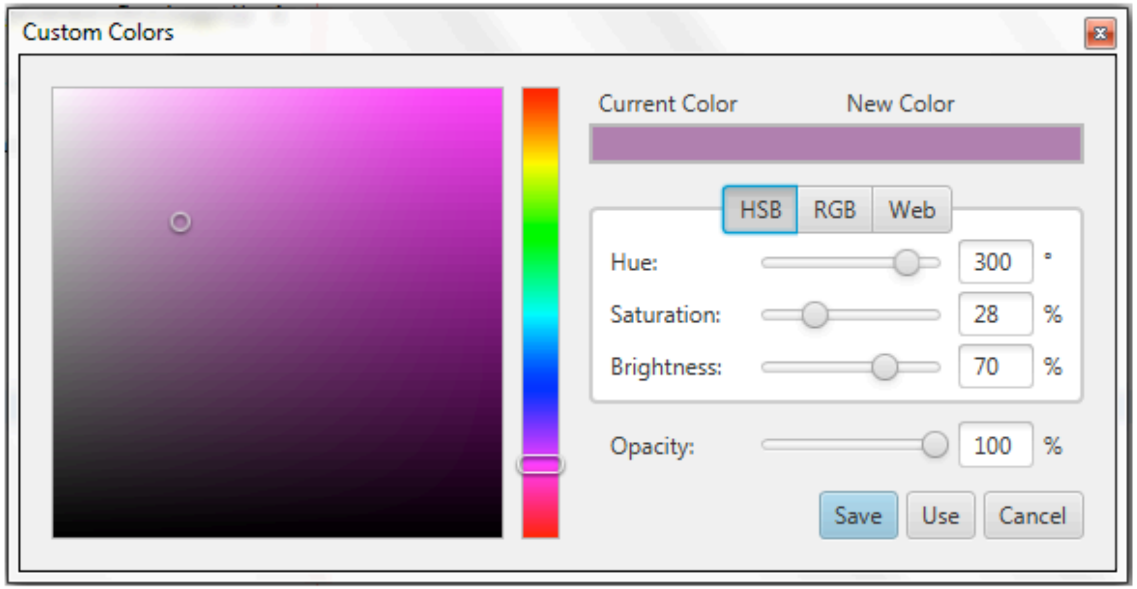


b380b3

color chooser

palette

custom color dialog window



Custom Colors

Current Color New Color

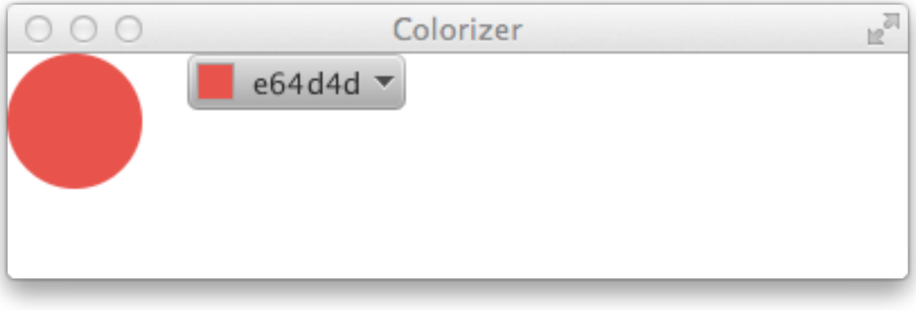
Hue: 300 °

Saturation: 28 %

Brightness: 70 %

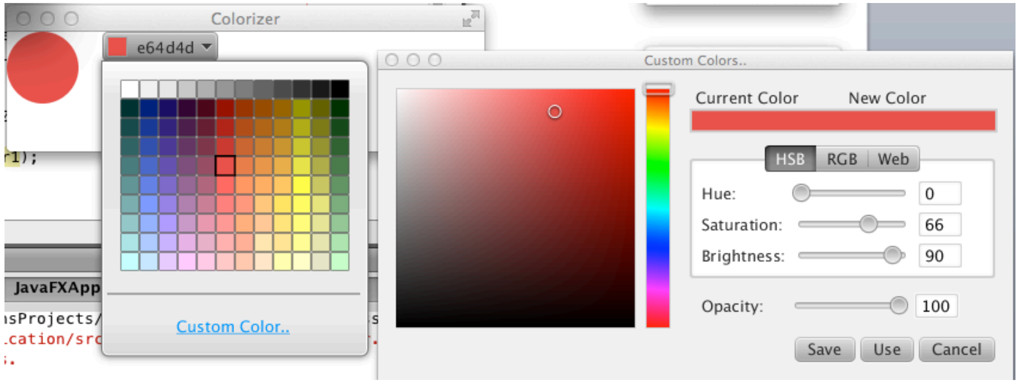
Opacity: 100 %

ColorPicker



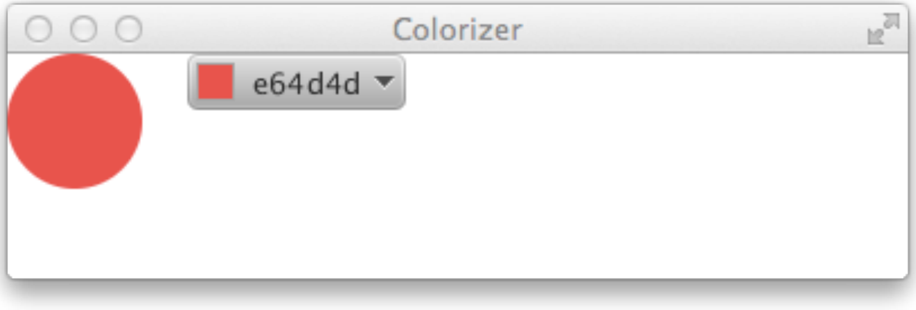
```
public class Colorizer extends Application {
  public void start(final Stage stage) {
    final Circle circ = new Circle(40, 40, 30);
    final ColorPicker colorPicker1 = new ColorPicker(Color.BLACK);

    colorPicker1.addEventHandler(ActionEvent.ACTION, new EventHandler() {
            @Override
            public void handle(Event t) {
                System.out.println(t.getEventType());
                circ.setFill(colorPicker1.getValue());
    }});
    Scene scene = new Scene(new HBox(20), 400, 100);
    HBox box = (HBox) scene.getRoot();
    box.getChildren().addAll(circ,colorPicker1);
    stage.setScene(scene);
    stage.show();
  }
  …
}
```

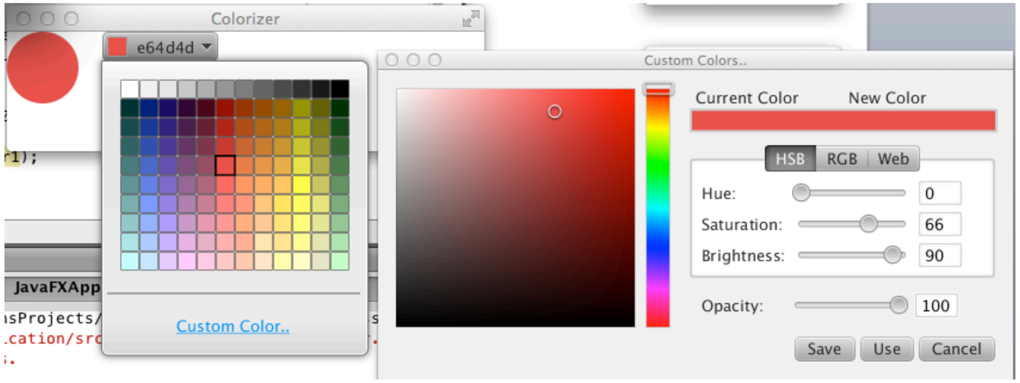


Hue: 0
Saturation: 66
Brightness: 90
Opacity: 100

Convenience Methods



```
public class Colorizer extends Application {
  public void start(final Stage stage) {
    final Circle circ = new Circle(40, 40, 30);
    final ColorPicker colorPicker1 = new ColorPicker(Color.BLACK);
    colorPicker1.setOnAction(new EventHandler(){
    // ~~colorPicker1.addEventHandler(ActionEvent.ACTION, new EventHandler() {~~
            @Override
            public void handle(Event t) {
                System.out.println(t.getEventType());
                circ.setFill(colorPicker1.getValue());
    }});
    Scene scene = new Scene(new HBox(20), 400, 100);
    HBox box = (HBox) scene.getRoot();
    box.getChildren().addAll(circ,colorPicker1);
    stage.setScene(scene);
    stage.show();
  }
  …
}
```



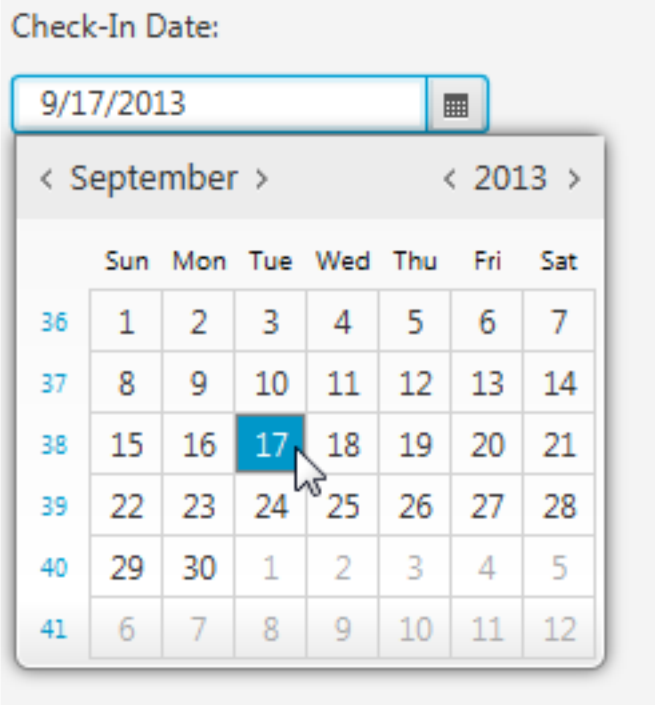
Hue: 0

Saturation: 66

Brightness: 90

Opacity: 100

DatePicker



Check-In Date: 9/17/2013

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
36	1	2	3	4	5	6	7
37	8	9	10	11	12	13	14
38	15	16	17	18	19	20	21
39	22	23	24	25	26	27	28
40	29	30	1	2	3	4	5
41	6	7	8	9	10	11	12

http://docs.oracle.com/javase/8/javafx/user-interface-tutorial/date-picker.htm#CCHHJBEA