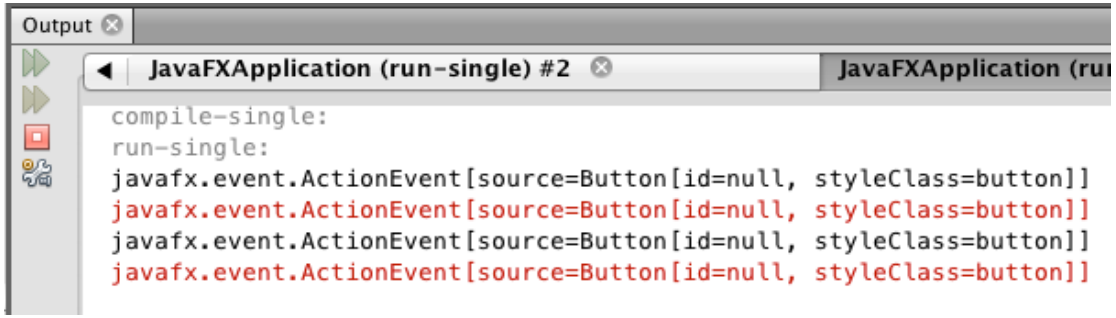
Gestione di base degli eventi

MultiListener

```
class Olistener
 implements EventHandler{
  public void handle(Event t) {
      System.out.println(t); }
}
```

```
class EListener
 implements EventHandler{
  public void handle(Event t) {
      System.err.println(t); }
}
```

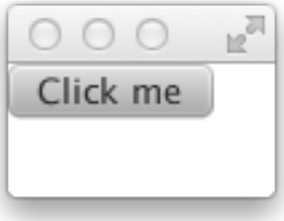
```
public class Event0 extends Application {
      public void start(Stage stage) {
        Button btn = new Button();
        btn.setText("Click me");
        Olistener o=new OListener();
        Elistener e=new EListener();
        btn.addEventHandler(ActionEvent.ACTION, o);
        btn.addEventHandler(ActionEvent.ACTION, e);
        Group root = new Group(btn);
        Scene scene = new Scene(root, 300, 250);
        stage (scene);
        stage ();
    }
    public static void main(String[] args){
      Application.launch(args);
    }
}
```



Basic Events

```
public class Event0 extends Application {
      public void start(Stage stage) {
        Button btn = new Button();
        btn.setText("Click me");
        Listener a=new Listener();
        btn.addEventHandler(Event.ANY, a);
        Group root = new Group(btn);
        Scene scene = new Scene(root, 300, 250);
         stage.setScene(scene);
        stage.sizeToScene();
        stage.show();    }
    public static void main(String[] args){
        Application.launch(args); }
}
class Listener implements EventHandler{
    int counter=0;
    public void handle(Event t) {
     System.out.println(++counter+" Ricevuto un evento di tipo "
        +t.getEventType()); } }
```

```
1 Ricevuto un evento di tipo
                INPUT_METHOD_TEXT_CHANGED
2 Ricevuto un evento di tipo MOUSE_ENTERED
3 Ricevuto un evento di tipo
                MOUSE_ENTERED_TARGET
4 Ricevuto un evento di tipo MOUSE_MOVED
…
12 Ricevuto un evento di tipo MOUSE_MOVED
13 Ricevuto un evento di tipo MOUSE_PRESSED
14 Ricevuto un evento di tipo ACTION
15 Ricevuto un evento di tipo MOUSE_RELEASED
16 Ricevuto un evento di tipo MOUSE_CLICKED
17 Ricevuto un evento di tipo MOUSE_MOVED
```



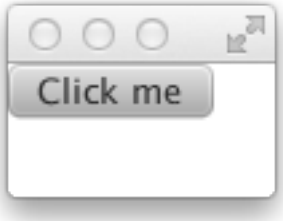
Warning

Note: /Users/ronchet/Downloads/JavaFX_001/src/javafx_001/Event0.java uses unchecked or unsafe operations.

Note: Recompile with -Xlint:unchecked for details.

Basic Events - fixed

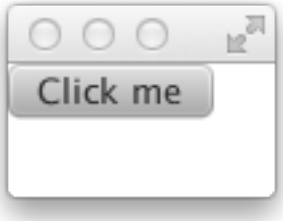
```
public class Event0 extends Application {
      public void start(Stage stage) {
        Button btn = new Button();
        btn.setText("Click me");
        Listener a=new Listener();
        btn.addEventHandler(Event.ANY, a);
        Group root = new Group(btn);
        Scene scene = new Scene(root, 300, 250);
         stage.setScene(scene);
        stage.sizeToScene();
        stage.show();    }
    public static void main(String[] args){
        Application.launch(args); }
}
class Listener implements EventHandler<Event>{
    int counter=0;
    public void handle(Event t) {
     System.out.println(++counter+" Ricevuto un evento di tipo "
        +t.getEventType()); } }
```



1 Ricevuto un evento di tipo ACTION

Basic Events

```
public class Event0 extends Application {
      public void start(Stage stage) {
        Button btn = new Button();
        btn.setText("Click me");
        Listener a=new Listener();
        btn.addEventHandler(ActionEvent.ACTION, a);
        Group root = new Group(btn);
        Scene scene = new Scene(root, 300, 250);
         stage.setScene(scene);
        stage.sizeToScene();
        stage.show();    }
    public static void main(String[] args){
        Application.launch(args); }
}
class Listener implements EventHandler{
    int counter=0;
    public void handle(Event t) {
     System.out.println(++counter+" Ricevuto un evento di tipo "
        +t.getEventType()); } }
```



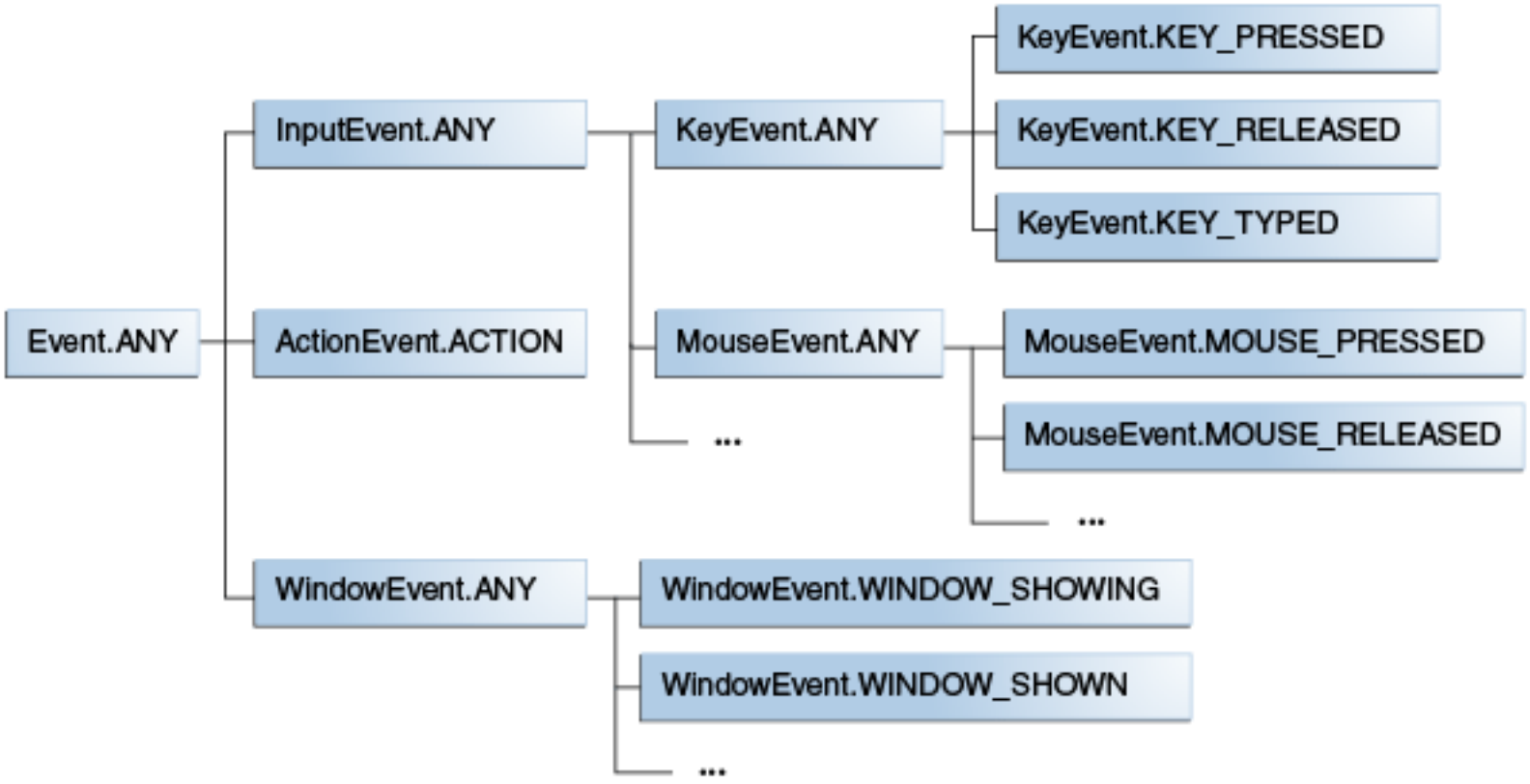
Basic Events - fixed

```
public class Event0 extends Application {
      public void start(Stage stage) {
        Button btn = new Button();
        btn.setText("Click me");
        Listener a=new Listener();
        btn.addEventHandler(ActionEvent.ACTION, a);
        Group root = new Group(btn);
        Scene scene = new Scene(root, 300, 250);
         stage.setScene(scene);
        stage.sizeToScene();
        stage.show();    }
    public static void main(String[] args){
        Application.launch(args); }
}
class Listener implements EventHandler<ActionEvent>{
    int counter=0;
    public void handle(ActionEvent t) {
     System.out.println(++counter+" Ricevuto un evento di tipo "
         +t.getEventType()); } }
```

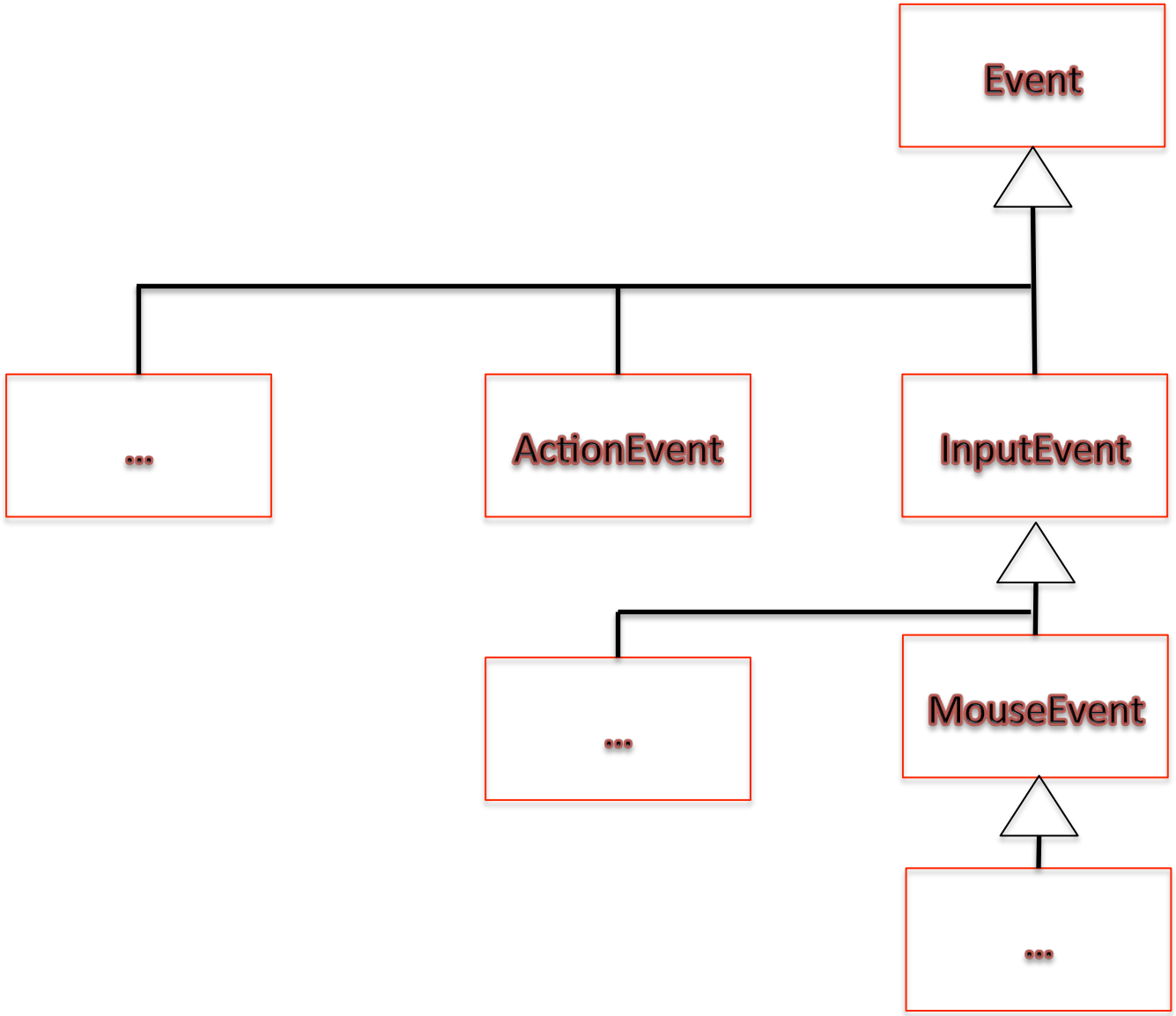
Basic Events – convenience method

```
public class Event0 extends Application {
      public void start(Stage stage) {
        Button btn = new Button();
        btn.setText("Click me");
        Listener a=new Listener();
        btn.setOnAction(a);
        Group root = new Group(btn);
        Scene scene = new Scene(root, 300, 250);
         stage.setScene(scene);
        stage.sizeToScene();
        stage.show();    }
    public static void main(String[] args){
        Application.launch(args); }
}
class Listener implements EventHandler<ActionEvent>{
    int counter=0;
    public void handle(ActionEvent t) {
     System.out.println(++counter+" Ricevuto un evento di tipo "
        +t.getEventType()); } }
```

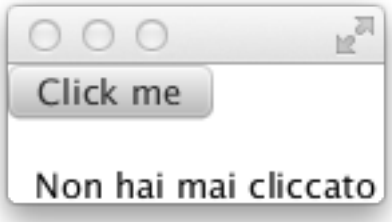
Event hierarchy

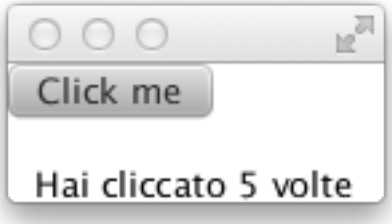


Event



Listener Esterno

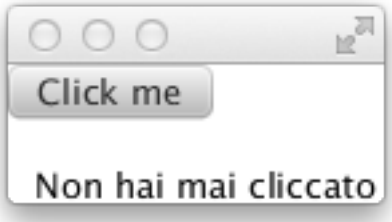


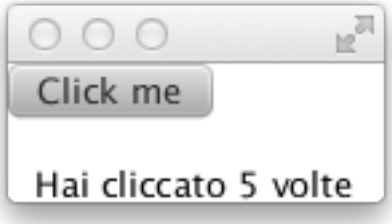



```
public class AppWithEvents1 extends Application {
    Text text=null;
    public void start(Stage stage) {
      text=new Text(10,50,"Non hai mai cliccato ");
      Button btn = new Button();
      btn.setText("Click me");
      Listener a=new Listener(this);
      btn.addEventHandler(ActionEvent.ACTION, 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<ActionEvent>{
  AppWithEvents1 awe=null;
  int counter=0;
  Listener(AppWithEvents1 a){
       awe=a;
  }
  public void handle(ActionEvent t) {
      awe.updateText(++counter);
}
}
```

Listener Interno

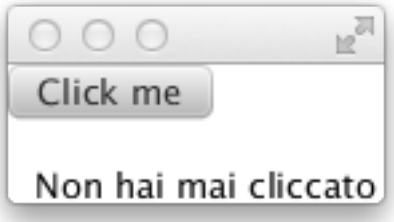


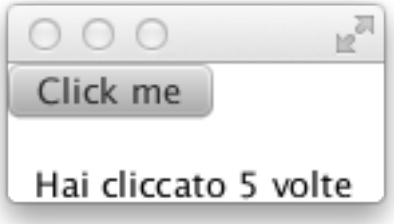



```
public class AppWithEvents1 extends Application {
    Text text=null;
    public void start(Stage stage) {
      text=new Text(10,50,"Non hai mai cliccato ");
      Button btn = new Button();
      btn.setText("Click me");
      Listener a=new Listener(this);
      btn.addEventHandler(ActionEvent.ACTION,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

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

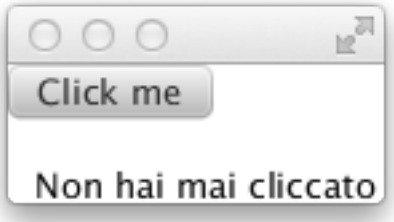
Listener Interno

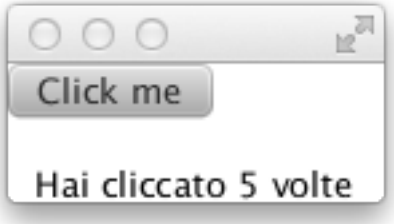





```
public class AppWithEvents1 extends Application {
    Text text=null;
    public void start(Stage stage) {
      text=new Text(10,50,"Non hai mai cliccato ");
      Button btn = new Button();
      btn.setText("Click me");
      Listener1 a=new Listener();
      btn.addEventHandler(ActionEvent.ACTION,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

Listener Interno Anonimo



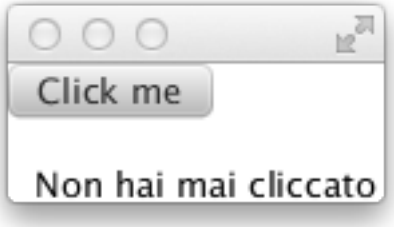


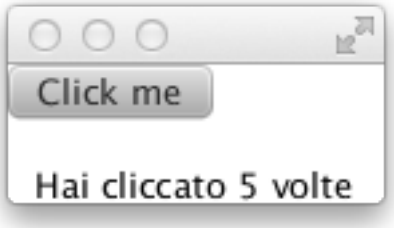

```
public class AppWithEvents1 extends Application {
    Text text=null;
    public void start(Stage stage) {
      text=new Text(10,50,"Non hai mai cliccato ");
      Button btn = new Button();
      btn.setText("Click me");
      EventHandler<ActionEvent> a
            =new EventHandler<ActionEvent>(){
       int counter=0;
       public void handle(ActionEvent 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);
}}
```

Self Listener

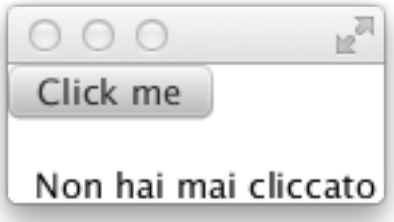
```
public class AppWithEvents
    extends Application implements EventHandler<ActionEvent> {
    Text text=null;
    int counter=0;
    public void start(Stage stage) {
      text=new Text(10,50,"Non hai mai cliccato ");
      Button btn = new Button();
      btn.setText("Click me");
      btn.addEventHandler(ActionEvent.ACTION,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

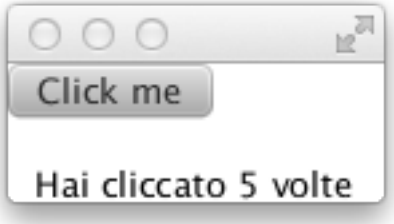





```
public void updateText(int n){
      text.setText("Hai cliccato"
         +n+" volte");
}
public static void main(
      String[] args) {
  Application.launch(args);
}}
```

Listener Interno Anonimo 2

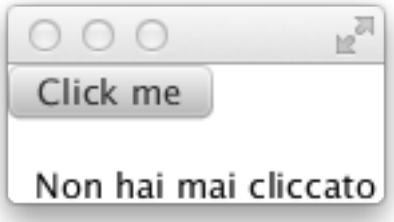


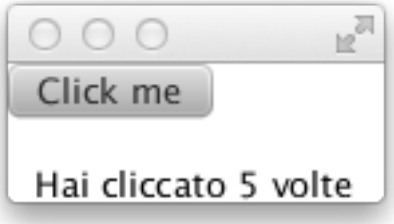



```
ublic class AppWithEvents1 extends Application {
    Text text=null;
    public void start(Stage stage) {
      text=new Text(10,50,"Non hai mai cliccato ");
      Button btn = new Button();
      btn.setText("Click me");
      btn.setOnAction(new EventHandler<ActionEvent>(){
       int counter=0;
       public void handle(ActionEvent t) {
           updateText(++counter);
      }});
      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);
  }}
```

Listener Interno Anonimo 2






```
ublic 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");
      btn.setOnAction(new EventHandler<ActionEvent>(){
       int counter=0;
       public void handle(ActionEvent t) {
           updateText(++counter);
      }});
      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);
  }}
```

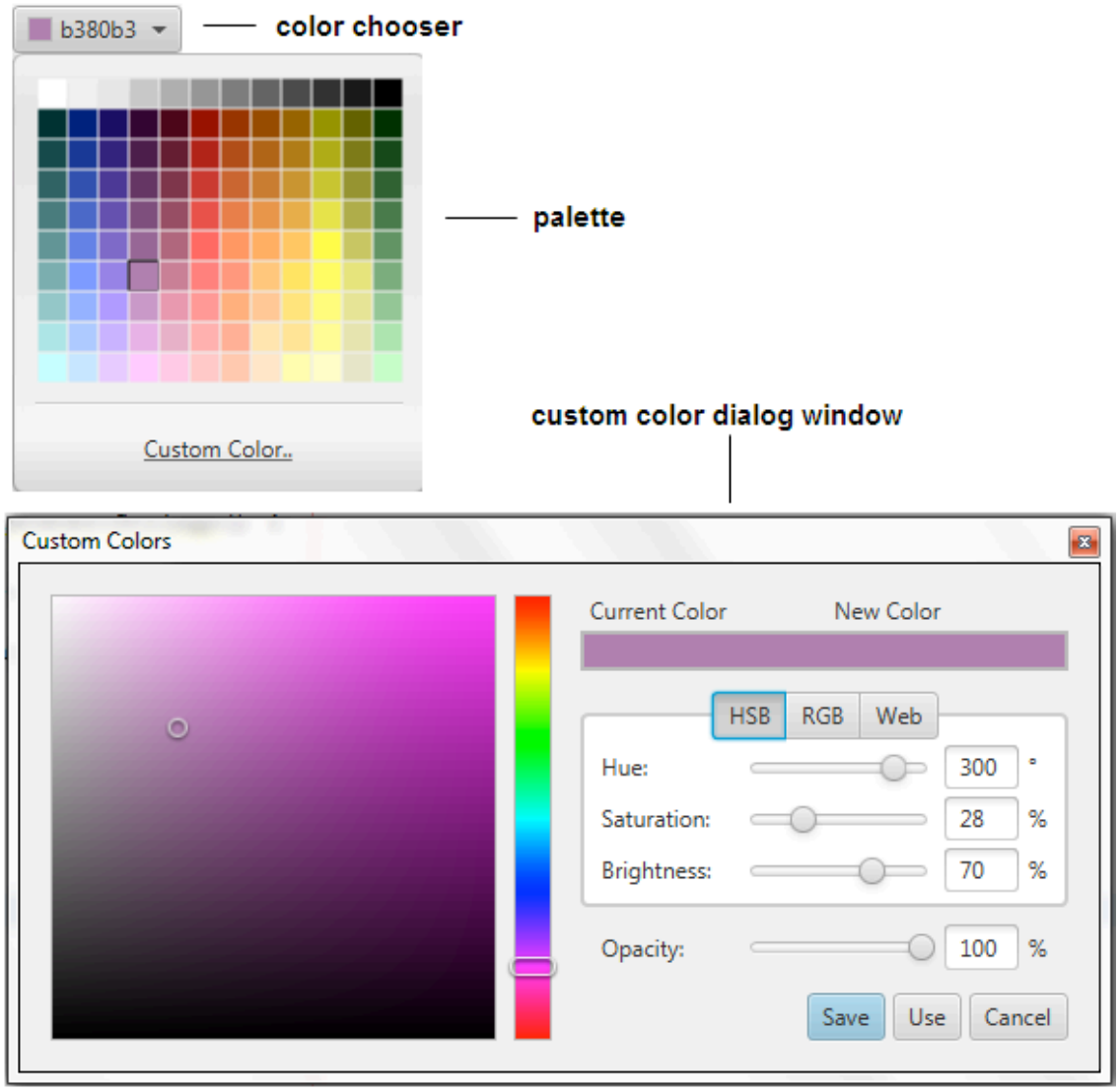
Questa slide è lasciata bianca per note

Questa slide è lasciata bianca per note

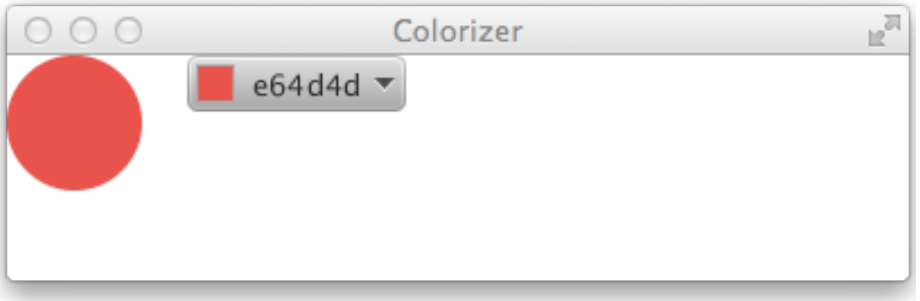
Due questioni:

- ColorPicker
- Convenience Methods

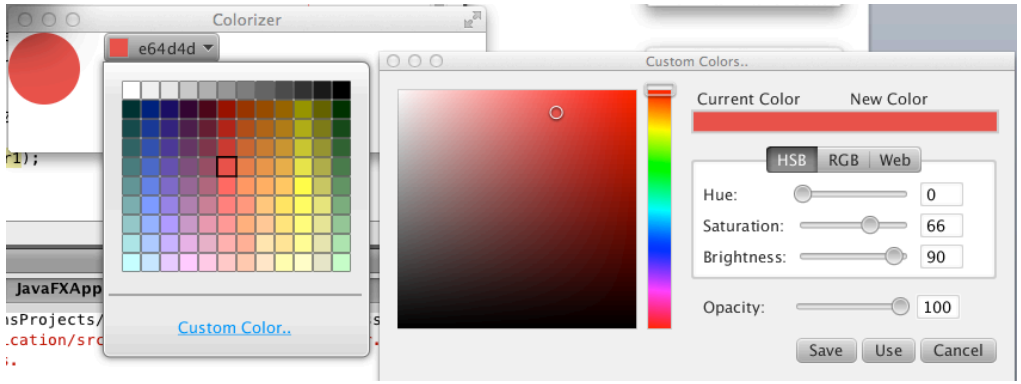
ColorPicker



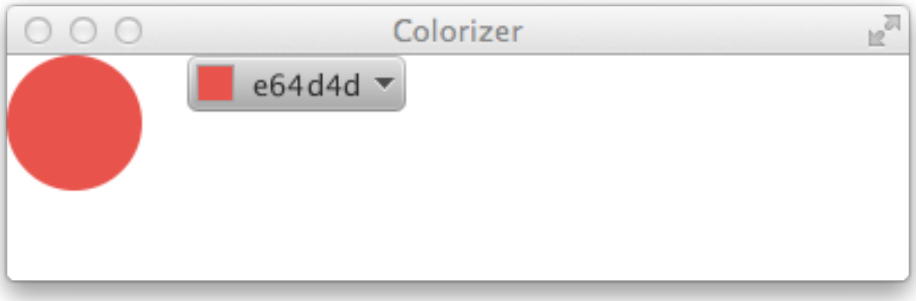
ColorPicker



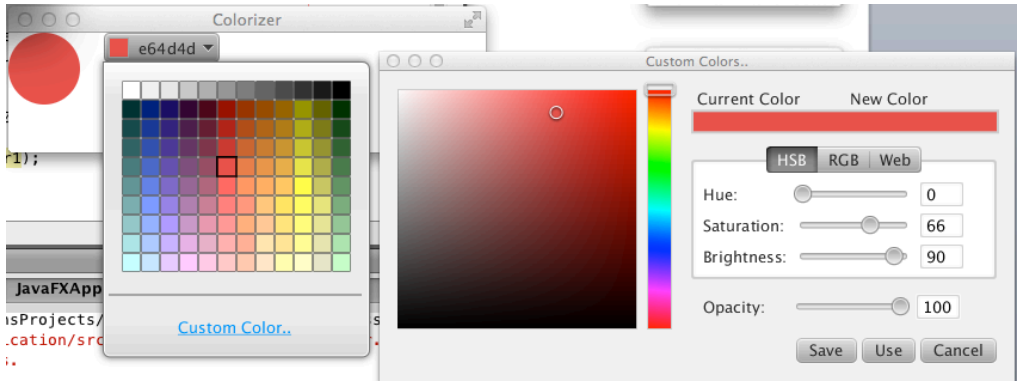

```
public class Colorizer extends Application {
  public void start(final Stage stage) {
    final Circle circ = new Circle(40, 40, 30);
    final ColorPicker colorPicker1 = new ColorPicker(Color.BLACK);

    colorPicker1.addEventHandler(ActionEvent.ACTION,
                                 new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent t) {
                 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();
  }
  …
}
```



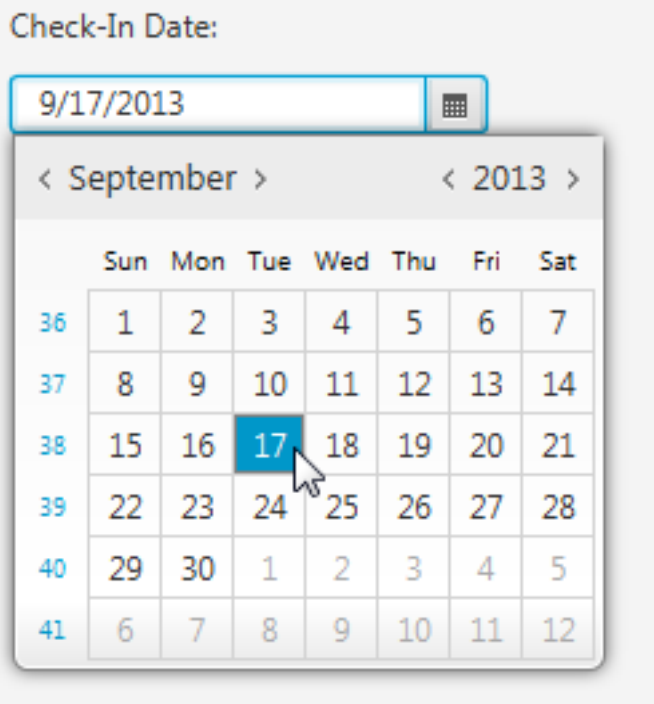
Convenience Methods




```
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<ActionEvent>(){
    // colorPicker1.addEventHandler(ActionEvent.ACTION, new EventHandler() {
            @Override
            public void handle(ActionEvent 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();
  }
  …
}
```



DatePicker



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