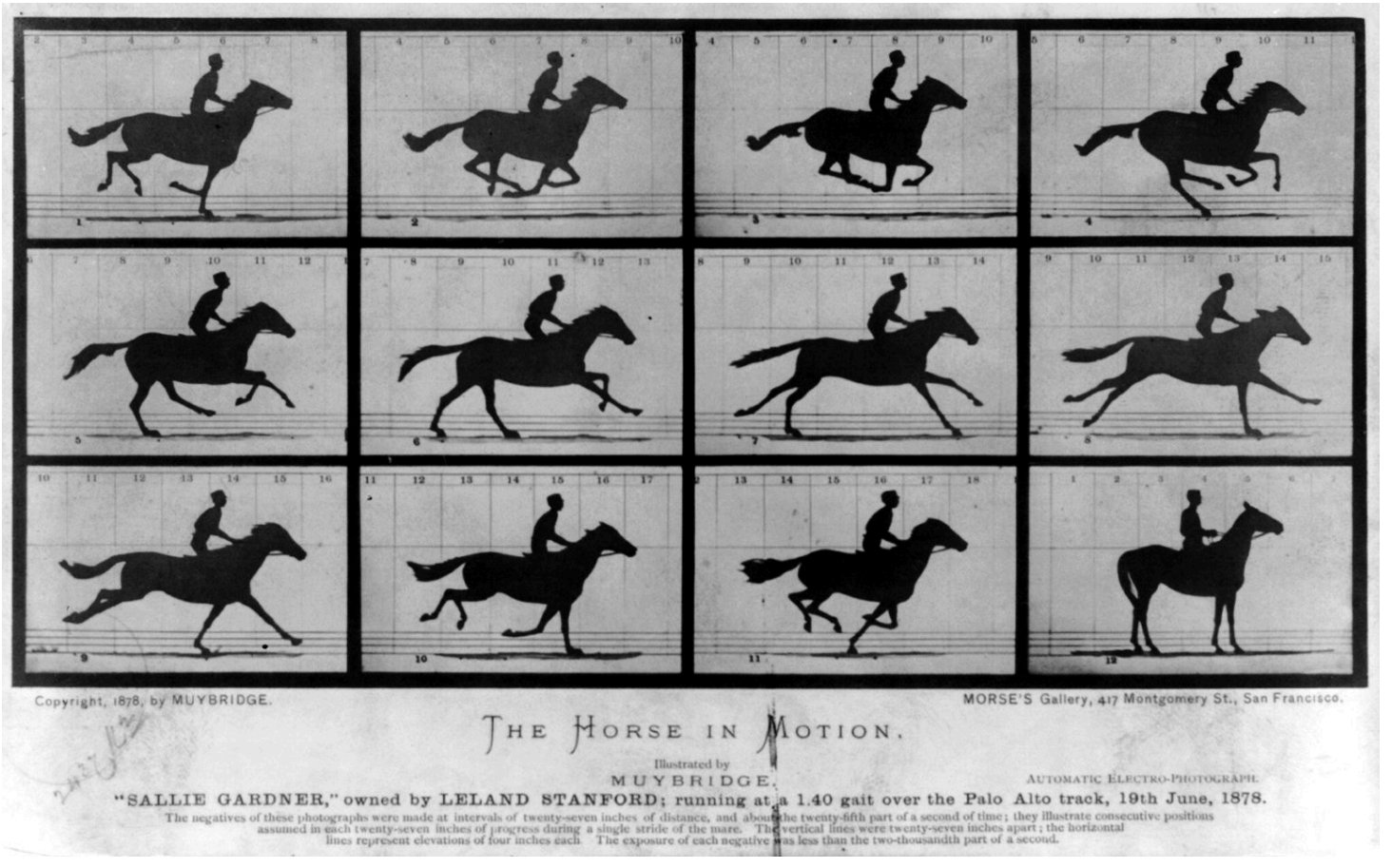
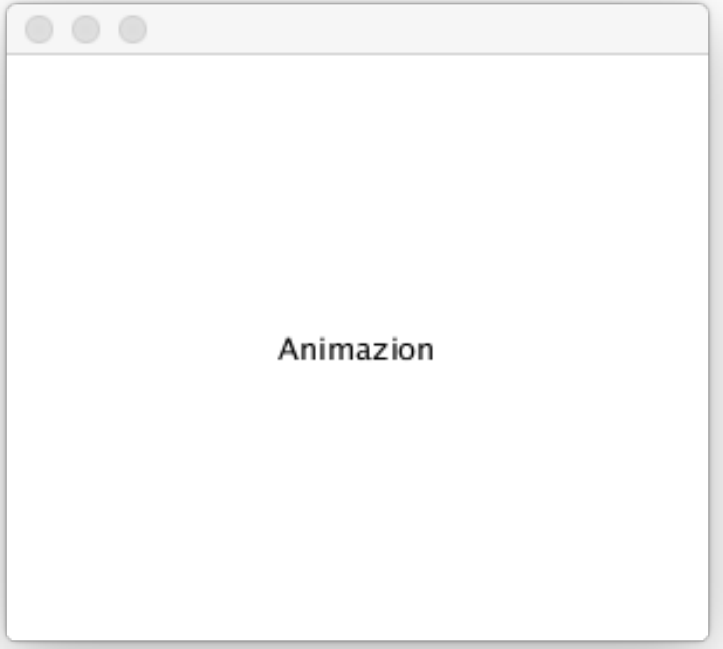
Animation in JavaFX

Horse in Motion: Frames



Esempio di Animation

```
public void start(Stage primaryStage) {
 final String content = "Animazioni Java FX";
 final Text text = new Text(10, 20, "");
final Animation animation = new Transition()      {
        { setCycleDuration(Duration.millis(2000)); } //constructor
        protected void interpolate(double frac) {
          final int length = content.length();
          final int n = Math.round(length * (float) frac);
          text.setText(content.substring(0, n));
        }
};
StackPane root = new StackPane();
root.getChildren().add(text);
Scene scene = new Scene(root, 300, 250);
primaryStage.setScene(scene);
primaryStage.show();
animation.play();
}
```



SimpleTransitionDemo

Animation

- provides the core functionality of all animations used in the JavaFX runtime.
- Methods:
  - play()
  - playFromStart()
  - pause() (keeps the play head to the current position)
  - stop() (reset its play head to the initial position)
- The Animation progresses in the direction and speed specified by rate
- An animation can run in a loop by setting cycleCount.
- An animation with indefinite duration (a cycleCount of INDEFINITE) runs repeatedly
- If the autoReverse –flag Is set, the animation run back and forth while looping

subclasses: **Transition** and **Timeline**

Transition

Offers a simple framework to define animation, and provides all the basic functionality defined in Animation.

- Transition requires:
  - the implementation of a method interpolate(double) which is the called in each frame, while the Transition is running.
  - to set the duration of a single cycle with Animation.setCycleDuration(javafx.util.Duration).

Esempio di Transition

```
public void start(Stage primaryStage) {
  final String content = "Lorem ipsum";
  final Text text = new Text(10, 20, "");
  final Animation animation = new Transition() {
    { setCycleDuration(Duration.millis(2000)); }
    protected void interpolate(double frac) {
       final int length = content.length();
       final int n = Math.round(length * (float) frac);
       text.setText(content.substring(0, n));
    }
  };
  StackPane root = new StackPane();
  root.getChildren().add(text);
  Scene scene = new Scene(root, 300, 250);
  primaryStage.setScene(scene);
  primaryStage.show();
  animation.play();
}
```

Come cambiare le coordinate di un oggetto posizionato da un Pane?

public final double getTranslateX()

Gets the value of the property translateX.

Property description:

Defines the x coordinate of the translation that is added to this Node's transform.

The node's final translation will be computed as layoutX + translateX, where layoutX establishes the node's stable position and translateX optionally makes dynamic adjustments to that position.

Layout e Translate properties

- getLayoutX
- setTranslateX
- getTraslateX

- stesso per Y

Timeline

- A Timeline can be used to define a free form animation of any WritableValue, e.g. all JavaFX Properties.

- A Timeline, defined by one or more KeyFrames, processes individual KeyFrame sequentially, in the order specified by KeyFrame.time. The animated properties, defined as key values in KeyFrame.values, are interpolated to/from the targeted key values at the specified time of the KeyFrame to Timeline's initial position, depends on Timeline's direction.

- Timeline processes individual KeyFrame at or after specified time interval elapsed, it does not guarantee the timing when KeyFrame is processed.

- If a KeyFrame is not provided for the time==0s instant, one will be synthesized using the target values that are current at the time Animation.play() or Animation.playFromStart() is called.

KeyFrame

Defines target values at a specified point in time for a set of variables that are interpolated along a Timeline.

The developer controls the interpolation of a set of variables for the interval between successive key frames by providing a target value and an Interpolator associated with each variable. The variables are interpolated such that they will reach their target value at the specified time.

An onFinished function is invoked on each KeyFrame if one is provided.

Esempio di animazione 1.1

```
public class BallAnimation1 extends Application {
    int dx = 1;
    int dy = 1;
    double frameDuration=0.05;
    @Override
    public void start(final Stage primaryStage) {
        primaryStage.setTitle("Animation");
        Group root = new Group();
        Scene scene = new Scene(root, 400, 300, Color.WHITE);
        primaryStage.setScene(scene);
        addBall(scene);
        primaryStage.show();
    }
    private void addBall(final Scene scene) { ... }
}
```

```
public static void main(String[] args) {
        Application.launch(args);
    }
```

Esempio di animazione 1.2 v-1

```
private void addBall(final Scene scene) {
    final Circle ball = new Circle(50, 50, 20);
    final Group root = (Group) scene.getRoot();
    root.getChildren().add(ball);
    Timeline tl = new Timeline();
    tl.setCycleCount(100); //Animation.INDEFINITE);
    KeyFrame moveBall = new KeyFrame(Duration.seconds(frameDuration),
            new EventHandler<ActionEvent>() {
                public void handle(ActionEvent event) {
                   ball.setTranslateX(ball.getTranslateX() + 2*dx);
                   ball.setTranslateY(ball.getTranslateY() + 0*dy);
                }
             });
    tl.getKeyFrames().add(moveBall);
    tl.play();
}
```

Esempio di animazione 1.2 v-2

```
private void addBall(final Scene scene) {
    final Circle ball = new Circle(50, 50, 20);
    final Group root = (Group) scene.getRoot();
    root.getChildren().add(ball)
    Timeline tl = new Timeline();
    tl.getKeyFrames().addAll(
       new KeyFrame(Duration.ZERO, // set start position
           new KeyValue(ball.translateXProperty(), 0),
           new KeyValue(ball.translateYProperty(), 0)),
       new KeyFrame(new Duration(2000), // set end position at 2s
           new KeyValue(ball.translateXProperty(), 200),
           new KeyValue(ball.translateYProperty(), 100)),
       new KeyFrame(new Duration(4000), // set end position at 4s
           new KeyValue(ball.translateXProperty(), 200),
           new KeyValue(ball.translateYProperty(), 0))
        );
        tl.play();
    }
```

Esempio di animazione – v-3

```
public void handle(ActionEvent event) {
    double scale=ball.getScaleX();
    if (scale>1.5) {
        ball.setFill(Color.RED);
        ball.setOpacity(ball.getOpacity()-0.005);
    }
    if (scale>2) ball.setVisible(!ball.isVisible());
    if (scale>2.5) {
        ball.setVisible(false);
        tl.stop();
    }
    ball.setScaleX(scale+0.01);
    //ball.setCenterX(ball.getCenterX() + 2*dx);
    //ball.setLayoutX(ball.getLayoutX() + 2*dx);
    ball.setTranslateX(ball.getTranslateX() + 2*dx);
    ball.setTranslateY(ball.getTranslateY() + 1*dy);
}
```

espansione del solo metodo handle delle slides precedenti!

- The node's final translation will be computed as layoutX + translateX, where layoutX establishes the node's stable position and translateX optionally makes dynamic adjustments to that position.
- If the node is managed and has a Region as its parent, then the layout region will set layoutX according to its own layout policy. If the node is unmanaged or parented by a Group, then the application may set layoutX directly to position it.

Fade Transition 1.1

```
public class FadeTransitionDemo extends Application {
    @Override
    public void start(final Stage primaryStage) {
        primaryStage.setTitle("Animation");
        Group root = new Group();
        Scene scene = new Scene(root, 400, 300, Color.WHITE);
        primaryStage.setScene(scene);
        addFadingRect(scene);
        primaryStage.show();
    }
    private void addFadingRect (final Scene scene) { ... }
}
```

```
public static void main(String[] args) {
        Application.launch(args);
    }
```

Fade Transition 1.2

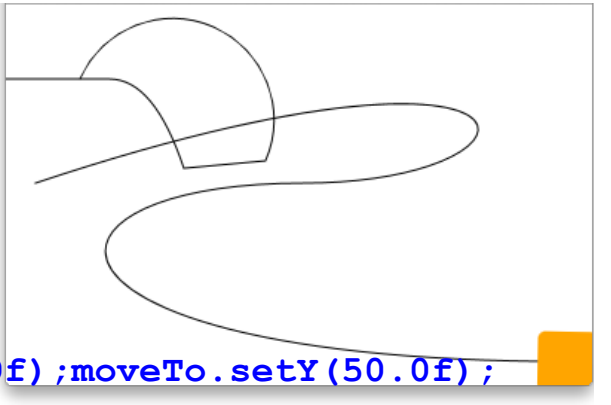
```
private void addFadingRect(final Scene scene) {
      final Rectangle rect1 = new Rectangle(10, 10, 100, 100);
      rect1.setArcHeight(20);
      rect1.setArcWidth(20);
      rect1.setFill(Color.RED);
     final Group root = (Group) scene.getRoot();
     root.getChildren().add(rect1);
     FadeTransition ft = new FadeTransition(Duration.millis(1000),
                 rect1);
     ft.setFromValue(1.0);
     ft.setToValue(0.0);
     ft.setCycleCount(Timeline.INDEFINITE);
     ft.setAutoReverse(true);
     ft.play();
}
```

PathTransition 1.1

```
public class PathExample extends Application {
    @Override
    public void start(Stage primaryStage) {
        Path path = drawPath();
        final Rectangle rectPath = new Rectangle(0, 0, 40, 40);
        rectPath.setArcHeight(10);
        rectPath.setArcWidth(10);
        rectPath.setFill(Color.ORANGE);
        Group root = new Group();
        root.getChildren().addAll(path,rectPath);
        Scene scene = new Scene(root, 400, 400);
        moveObjectOnPath(path, rectPath);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
    drawPath() {...}
    moveObjectOnPath (...) {...}
```

```
public static void main(String[] args) {
        Application.launch(args);
    }
```

PathTransition 1.2



```
public Path drawPath() {
    Path path = new Path();
    MoveTo moveTo = new MoveTo(); moveTo.setX(0.0f);moveTo.setY(50.0f);
    HLineTo hLineTo = new HLineTo(); hLineTo.setX(70.0f);
    QuadCurveTo quadCurveTo = new QuadCurveTo();
    quadCurveTo.setX(120.0f); quadCurveTo.setY(110.0f);
    quadCurveTo.setControlX(100.0f); quadCurveTo.setControlY(50.0f);
    LineTo lineTo = new LineTo();
    lineTo.setX(175.0f); lineTo.setY(105.0f);
    ArcTo arcTo = new ArcTo(); arcTo.setX(50.0f); arcTo.setY(50.0f);
    arcTo.setRadiusX(50.0f); arcTo.setRadiusY(50.0f);
    path.getElements().add(moveTo); path.getElements().add(hLineTo);
    path.getElements().add(quadCurveTo); path.getElements().add(lineTo);
    path.getElements().add(arcTo);
    path.getElements().add(new MoveTo(20, 120));
    path.getElements().add(new CubicCurveTo(380, 0, 380, 120, 200, 120));
    path.getElements().add(new CubicCurveTo(0, 120, 0, 240, 380, 240));
    return path;
}
```

PathTransition 1.3

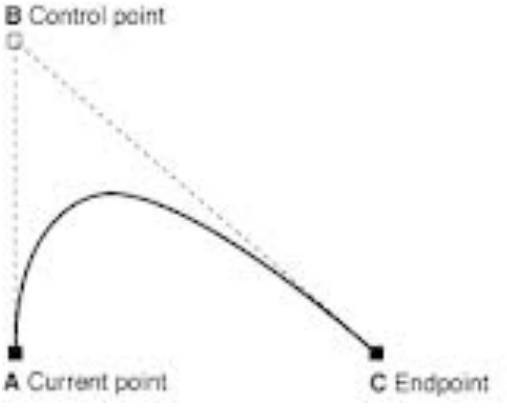
```
void moveObjectOnPath(Path path, Node node) {
        PathTransition pathTransition = new PathTransition();
        pathTransition.setDuration(Duration.millis(4000));
        pathTransition.setPath(path);
        pathTransition.setNode(node);
        pathTransition.setOrientation(
            PathTransition.OrientationType.ORTHOGONAL_TO_TANGENT);
        pathTransition.setCycleCount(Timeline.INDEFINITE);
        pathTransition.setAutoReverse(true);
        pathTransition.play();
    }
```

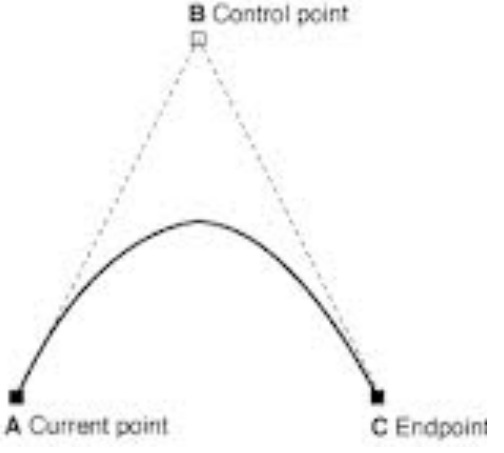
ORTHOGONAL_TO_TANGENT
The targeted node's rotation matrix is set to keep node perpendicular to the path's tangent along the geometric path.

QuadCurveTo

```
MoveTo moveTo = new MoveTo();
moveTo.setX(0.0f);
moveTo.setY(0.0f);

QuadCurveTo quadTo = new QuadCurveTo();
quadTo.setControlX(0.0f);
quadTo.setControlY(0.0f);
quadTo.setX(100.0f);
quadTo.setY(50.0f);
```

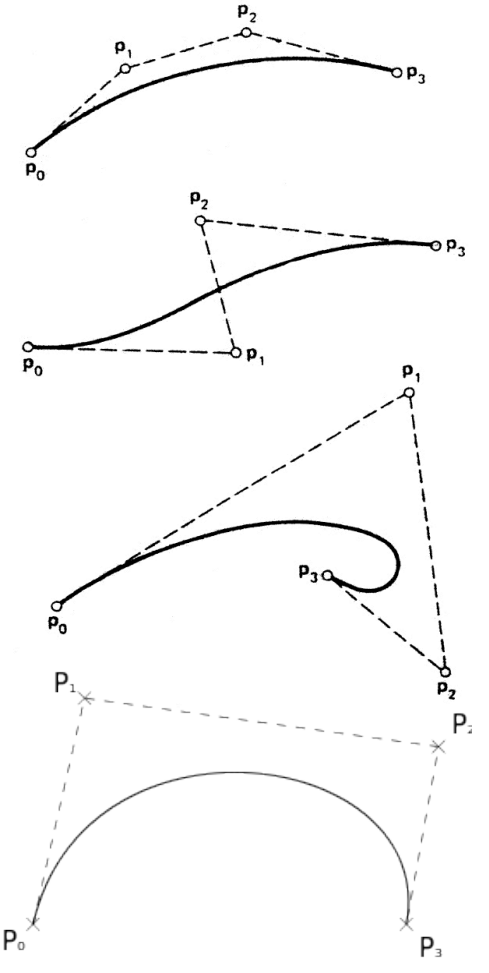




Curve di Bezier Quadratiche

$$\mathbf{B}(t) = (1-t)^2\mathbf{P}_0 + 2t(1-t)\mathbf{P}_1 + t^2\mathbf{P}_2 \ , \ t \in [0,1].$$

CubicCurveTo




```
MoveTo moveTo = new MoveTo();
moveTo.setX(0.0f);
moveTo.setY(0.0f);

CubicCurveTo cubicTo = new CubicCurveTo();
cubicTo.setControlX1(0.0f);
cubicTo.setControlY1(0.0f);
cubicTo.setControlX2(100.0f);
cubicTo.setControlY2(100.0f);
cubicTo.setX(100.0f);
cubicTo.setY(50.0f);
```

Curve di Bezier Cubiche

$$\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2 t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3 \ , \ t \in [0,1].$$

Altre Transition

- FadeTransition
- FillTransition
- PathTransition
- PauseTransition
- RotateTransition
- ScaleTransition
- StrokeTransition
- TranslateTransition

Transition composition

```
SequentialTransition sq= new SequentialTransition();
sq().addAll(fadeTransition, translateTransition, ...);

ParallelTransition pt= new ParallelTransition ();
pt().addAll(fadeTransition, translateTransition, ...);
```

Altri esempi

- BouncingBallAnimation
- Animation
- Tree