

Interfaces graphiques - JavaFX

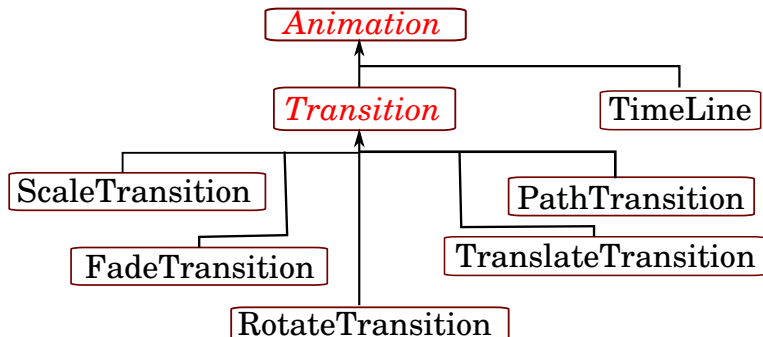
responsable : Wiesław Zielonka

`zielonka@liafa.univ-paris-diderot.fr`

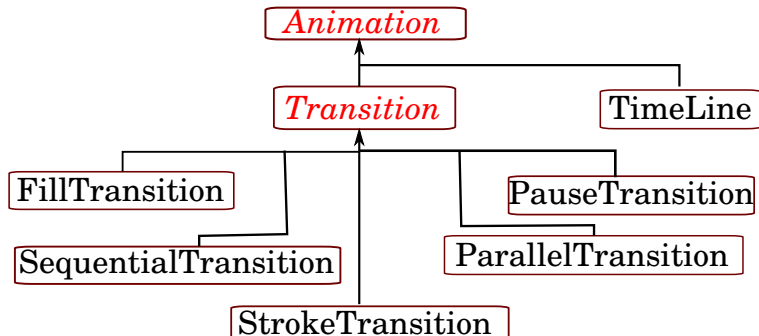
`http://liafa.univ-paris-diderot.fr/~zielonka`

March 23, 2016

Hiérarchie de classes d'animations



Hiérarchie d'animations (suite)



Animation

- ▶ exécuter dans une boucle pendant le temps spécifié
- ▶ contrôle par “cycle count”,
- ▶ contrôler la vitesse et la direction,
- ▶ l'accès direct à chaque position de l'animation.

Les états de l'animation :

- ▶ `Animation.Status.RUNNING` - l'animation s'exécute,
- ▶ `Animation.Status.PAUSED` - l'animation est pause,
- ▶ `Animation.Status.STOPPED` - l'animation arrêtée.

Animation – les propriétés

- ▶ `rate` – la vitesse et la direction,
- ▶ `cycleCount` – le nombre de cycles de l'animation, doit être soit positif soit `Animation.INDEFINITE` si le nombre de cycles infini,
- ▶ `autoReverse` – inverser la direction (`false` par défaut),
- ▶ `onFinished` - l'action à exécuter à la fin de l'animation,
- ▶ `status` – l'état de l'animation,
- ▶ `currentTime` – le temps depuis le début du dernier cycle de l'animation.

Contrôler l'animation

Les méthodes :

- ▶ `play()` - exécuter l'animation à partir de la position courante,
- ▶ `playFrom()` - exécuter l'animation à partir de la position spécifique,
- ▶ `playFromStart()` - exécuter depuis le début,
- ▶ `pause()` - mettre en pause, `stop()` -arrêter, le position courante remise au débute de l'animation,
- ▶ `jumpTo()` – sauter à la position spécifique.

La classe Transition

La propriété `interpolator` contrôle l'accélération et ralentissement de l'animation :

- ▶ `Interpolator.LINEAR` – vitesse constante,
- ▶ `Interpolator.DISCRETE` – saut discret, début → fin,
- ▶ `Interpolator.EASE_IN` – lentement au début et accélère ensuite,
- ▶ `Interpolator.EASE_OUT` – ralentit à la fin,
- ▶ `Interpolator.EASE_BOTH` – ralentit au début et à la fin.

FadeTransition

Modification de l'opacité du noeud.

Propriétés :

- ▶ `node` – le noeud qui subit la transition,
- ▶ `duration` – l'intervalle du temps,
- ▶ `fromValue` – valeur initiale,
- ▶ `toValue` – valeur finale,
- ▶ `byValue` – l'incrément.

Si `fromValue` n'est pas spécifié alors l'opacité initiale du noeud est utilisée comme l'opacité de départ.

Pour changer l'attribut pendant que l'animation s'exécute il faut d'abord mettre l'animation en pause.

FillTransition

`FillTransition` s'applique uniquement à `Shape` (et les classes dérivées).

`FillTransition` modifie la propriété `fill` d'un noeud (la couleur de remplissage du noeud).

Propriétés de `FillTransition` :

- ▶ `shape` – le “shape” qui subit la transition,
- ▶ `duration` – l'intervalle (la durée de la transition),
- ▶ `fromValue` – la couleur initiale,
- ▶ `toValue` – la couleur finale.

StrokeTransition

`StrokeTransition` s'applique uniquement à `Shape` (et les classes dérivées).

`StrokeTransition` modifie la propriété `stroke` d'un noeud (la couleur du trait).

Propriétés de `StrokeTransition` :

- ▶ `shape` – le “shape” qui subit la transition,
- ▶ `duration` – l'intervalle (la durée de la transition),
- ▶ `fromValue` – la couleur initiale,
- ▶ `toValue` – la couleur finale.

ScaleTransition

Modifie la taille de noeud en modifiant les propriétés

`scaleX scaleY scaleZ`

(modification d'échelle).

Propriétés de `ScaleTransition` :

- ▶ `node` – le noeud qui est l'objet de la transition,
- ▶ `duration` – la durée,
- ▶ `fromX`, `fromY`, `fromZ` – valeur initiales,
- ▶ `toX`, `toY`, `toZ` valeurs finales,
- ▶ `byX`, `byY`, `byZ` – l'incrément.

Pas besoin de spécifier les valeurs `Z` si la dimension `Z` ne change pas.

TranslateTransition

Modifie les valeurs des propriétés `translateX`, `translateY`, `translateZ` du noeud.

Propriétés de `TranslateTransition` :

- ▶ `node` – le noeud qui est l'objet de la transition,
- ▶ `duration` – la durée,
- ▶ `fromX`, `fromY`, `fromZ` – la position initiale,
- ▶ `toX`, `toY`, `toZ` la position finale,
- ▶ `byX`, `byY`, `byZ` – distance à parcourir.

RotateTransition

Modifie la propriété `rotate` du noeud.

Propriétés de `RotateTransition` :

- ▶ `node` – le noeud l'objet de la rotation,
- ▶ `duration` – le temps de la rotation,
- ▶ `fromAngle` – l'angle initial en degrés,
- ▶ `toAngle` – l'angle final,
- ▶ `byAngle` – l'incrément,ation,
- ▶ `axis` – l'axe de la rotation (l'axe Z par défaut).

PathTransition

`PathTranslation` déplace un objet le long d'une courbe.

Modifie les propriétés : `translateX`, `transalteY` et `rotate` du noeud.

Propriétés de `PathTransition` :

- ▶ `node` – le noeud l'objet de la rotation,
- ▶ `duration` – le temps de la rotation,
- ▶ `path` – le chemin de la transition,
- ▶ `orientation` – l'orientation du noeud le long de la courbe.

PauseTransition

La transition que fait rien (ne change pas les valeurs de propriétés d'un objet).

Propriétés `onFinished` indique l'action à faire à la fin de la pause.

`PauseTransition` utile dans les compositions parallèles ou séquentielles avec d'autres transitions.

SequentialTransition et ParallelTransition

`SequentialTransition` maintient une liste de transitions qui s'exécutent une après l'autre.

`ParallelTransition` maintient une liste de transitions qui s'exécutent une après l'autre.

Si une transition fille de `SequentialTransition` n'a pas de noeud spécifié alors elle utilise le noeud hérité de `SequentialTransition`. Même chose pour `ParallelTransition`.