Archived:Animating graphics item position and rotation simultaneously

Archived: This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Qt Quick should be used for all UI development on mobile devices. The approach described in this article (using C++ for the Qt app UI) is deprecated.

Overview

This series of snippets demonstrates how to implement nice animation effects into your Qt UI application. Qt 4.6 has Qt Animation Framework as a built-in feature. Documentation can be found here: [1]

This snippet shows how to execute parallel animations using `QParallelAnimationGroup`. Item position and rotation are animated simultaneously.

See full example code here: File:Animation fw.zip

Preconditions

Qt 4.6 has been installed.

Header file

```
// Animate position and angle
void animatePosAndAngle(const QPointF& start, const QPointF& end,
                        int startAngle, int endAngle);
};
```

Source file

Create animations in `QParallelAnimationGroup` to execute them simultaneously.

```
void PictureItem::animatePosAndAngle(const QPointF& start, const QPointF& end,
                                     int startAngle, int endAngle)
{
    // Create animation group
    // Animations in the group are executed in same time
    QParallelAnimationGroup *group = new QParallelAnimationGroup;

    // Create rotation animation
    QPropertyAnimation* anim1 = new QPropertyAnimation(this, "rotationAngleY");
    anim1->setDuration(3000);
    anim1->setStartValue(startAngle);
    anim1->setEndValue(endAngle);
    anim1->setEasingCurve(QEasingCurve::InOutBack);

    // Create position animation
    QPropertyAnimation* anim2 = new QPropertyAnimation(this, "pos");
    anim2->setDuration(3000);
    anim2->setStartValue(start);
    anim2->setEndValue(end);
    anim2->setEasingCurve(QEasingCurve::InOutElastic);

    // Add animations into the group
    group->addAnimation(anim1);
    group->addAnimation(anim2);

    // Execute group
    group->start(QAbstractAnimation::DeleteWhenStopped);
}
```

Postconditions

Parallel animation has been created using `QParallelAnimationGroup`.

(C) Copyright Nokia 2013. All rights reserved.

Animation Framework snippet series

Full example code package

- Enabling Qt Animation Framework in an application
- Archived:Animating graphics item position in Qt
- Archived:Animating graphics item rotation
- Archived:Animating graphics item fading
- Archived:Animating graphics item scaling
- **Archived:Animating graphics item position and rotation simultaneously**

 (C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.