Archived:Animating graphics item rotation

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 animate `QGraphicsPixmapItem` with a rotate effect.

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

Preconditions

Qt 4.6 has been installed.

Header file

Define `rotationAngleY` property and setter/getter function.

```
class PictureItem: public QObject, public QGraphicsPixmapItem
{
Q_OBJECT

    // Rotating picture property for QPropertyAnimation
    Q_PROPERTY(qreal rotationAngleY READ rotationAngleY WRITE setRotationAngleY)

public:
    // For rotationAngleY property
    void setRotationAngleY(qreal angle);
    qreal rotationAngleY() const;

    // Animate angle of this class
    void animateAngle(int startAngle, int endAngle);

private:
    qreal m_rotationAngleY;
};
```

Source file

Implement setter/getter for `rotationAngleY` property.

The animation is executed from the `animateAngle()` method.

(C) Copyright Nokia 2014. All rights reserved.

```
qreal PictureItem::rotationAngleY() const
{
    return m_rotationAngleY;
}

void PictureItem::setRotationAngleY(qreal angle)
{
    if (m_rotationAngleY != angle) {
        m_rotationAngleY = angle;
        QPointF c = boundingRect().center();
        QTransform t;
        t.translate(c.x(), c.y());
        t.rotate(m_rotationAngleY, Qt::YAxis);
        t.translate(-c.x(), -c.y());
        setTransform(t);
    }
}

void PictureItem::animateAngle(int startAngle, int endAngle)
{
    // Start animate this class
    QPropertyAnimation* anim = new QPropertyAnimation(this, "rotationAngleY");

    // 2 second duration animation
    anim->setDuration(2000);
    // angle to start animation
    anim->setStartValue(startAngle);
    // end angle of animation
    anim->setEndValue(endAngle);
    // easing curve
    anim->setEasingCurve(QEasingCurve::InOutBack);

    // Listen animation finished signal
    QObject::connect(anim, SIGNAL(finished()), this, SLOT(animationFinished()));

    // Start animation and delete QPropertyAnimation class on the end
    anim->start(QAbstractAnimation::DeleteWhenStopped);
}
```

Postconditions

`PictureItem` rotation can be animated by calling `PictureItem::animateAngle()`.

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 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.