Archived:How to create a simple animation using QTimeLine

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 (based on QWidget) is deprecated.

This (archived) code snippet demonstrates how to create a simple Qt animation using QTimeLine

Tip: The Qt 4.6 Animation Framework provides a more elegant solution for animation that using QTimeLine directly.

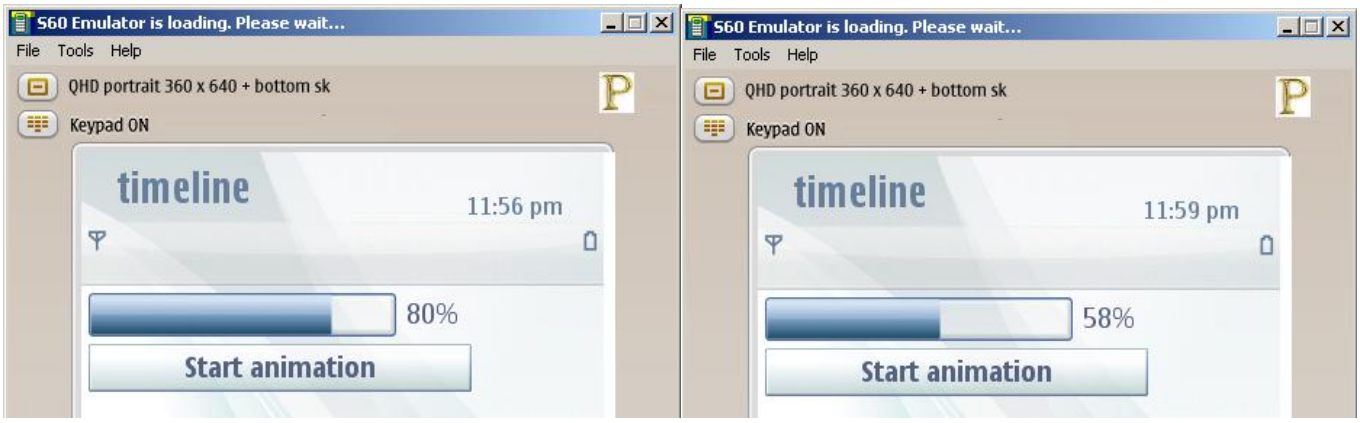
Preconditions

- Download and install the Qt SDK

Various Functions

- This property holds the direction of the timeline when QTimeLine is in Running state.The default is forward.

```
timeLine->setDirection(QTimeLine::Backward);
```



- This property holds the shape of the timeline curve.

```
timeLine->setCurveShape(QTimeLine::EaseInCurve);
```

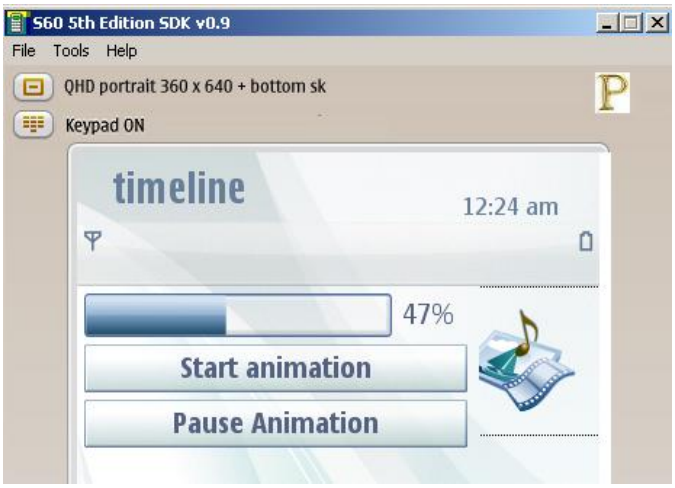
- To add a Pause to pause the TimeLine you have to add the following code in to your source code.

```
connect(pushButton1, SIGNAL(clicked()), this, SLOT(viral()));
```

Add This Function to your current code.

```
void timeline::viral(  timeLine->setPaused(1); }
```


(C) Copyright Nokia 2013. All rights reserved.



Code Snippet

Source File

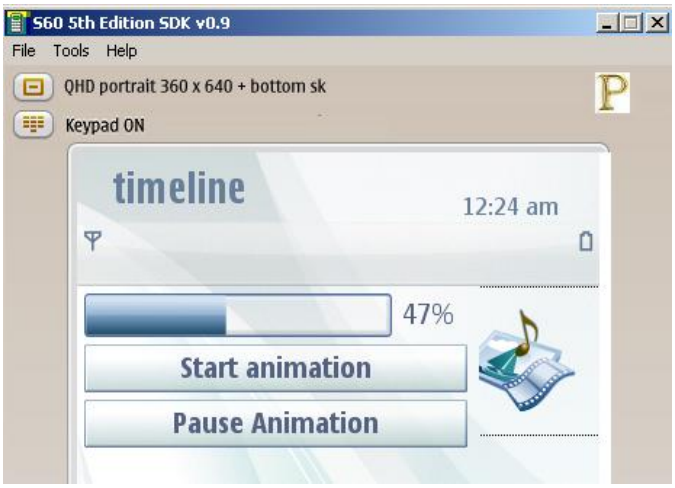
```
#include "timeline.h"

timeline::timeline(QWidget *parent)
    : QWidget(parent)
{
 layout= new QVBoxLayout(this);
 progressBar = new QProgressBar(this);
 progressBar->setRange(0, 100);

 // Construct a 5-second timeline with a frame range of 0 - 100
 timeLine = new QTimeLine(5000, this);
 timeLine->setFrameRange(0, 100);
 connect(timeLine, SIGNAL(frameChanged(int)), progressBar, SLOT(setValue(int)));

 // Clicking the push button will start the progress bar animation
 pushButton = new QPushButton(tr("Start animation"), this);
 connect(pushButton, SIGNAL(clicked()), timeLine, SLOT(start()));
 layout->addWidget(progressBar);
 layout->addWidget(pushButton);
 setLayout(layout);
}

timeline::~timeline()
{
   // No need to delete any QObject that got proper parent pointer.

}
```

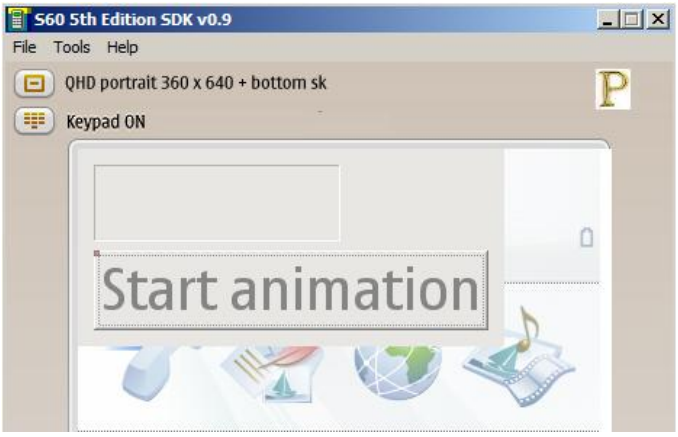
Header File

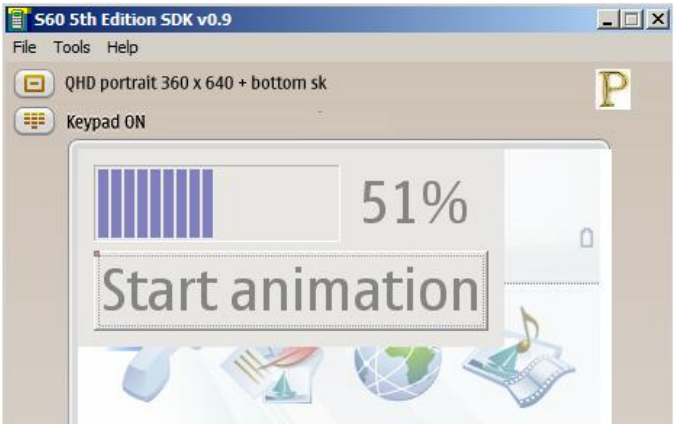
```
#ifndef TIMELINE_H
#define TIMELINE_H

#include <QtGui/QWidget>
#include "ui_timeline.h"
#include <QVBoxLayout>
#include <QTimeLine>
#include <QProgressBar>
#include <QPushButton>
class timeline : public QWidget
{
    Q_OBJECT

public:
 timeline(QWidget *parent = 0);
    ~timeline();

private:
 QPushButton *pushButton;
 QTimeLine *timeLine;
 QProgressBar *progressBar;
 QVBoxLayout *layout;
};

#endif // TIMELINE_H
```

ScreenShot

(C) Copyright Nokia 2013. All rights reserved.







(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.