Simple video player in QML

Introduction

Video playing is one of the most important requirement now a days in applications, be it playing a video player by storing the whole video file in the phone storage system or simply streaming a video.

This application takes the first use case, second will be explained in one of the coming articles.

Playing Video files in Qt

There are lots of option if one needs to play video files in Qt for Symbian or MeeGo devices.

First one can store the video file in phone storage system and then open phone's native video player using Qt::OpenUrlExternally function of Qt, this can be directly used with QML code.

Second you can use QMediaPlayer and QVideoWidget APIs to play Videos from Qt C++.

Third is using QMLs Video element which is nothing else but a QML Plugin class for `QMediaPlayer` and `QVideoWidget`.

First one works pretty smooth with some simple hacks, second and third approach are generally used if you require an video player of your self within your application context.

This article shows how to play videos using QML Video Element.

QML Video Element

QML Video Element is basically part if Qt Multimedia 1.1 package for Qt Mobility Framework. Its a simple QML component where you just need to give the source of Video along with the size you need to keep for this element.

This Component is still Buggy if you Load the video element in your main QML file at the Launch of the application then it crashes with Symbian native WSERV 10 Panic.

To Over come this Bug you need to Load the Video Element as and when required instead of just Loading it at the start of the application.

Code Snippet Below Will Show how that can be done with the simple Video Player Component.

Video Player Component

Component Below is a simple Video Player component, This component file has three main important Functions:

- playVideo
- stopVideo
- pauseVideo

QML Video Element has lots of properties using which we can get maximum of the metadata of a particulat video file, which has a lot of usage too.

```
import QtQuick 1.0
import QtMultimediaKit 1.1 // do not forgot to import this

Item {
    id: videoPlayerItem
    property bool isVideoPlaying: videoPlayer.hasVideo // property to know if videoPlaying id ongoing or not
    /*
      Functions Which starts Video Playing
      */
    function playVideo(){
        videoPlayer.play()
    }
    /*
      Function Which Stops Video Playing
      */
    function stopVideo(){
        videoPlayer.stop()
    }
    /*
      Function Which Stops Video Playing
      */
    function pauseVideo(){
        videoPlayer.pause()
    }
    /*
      Actual QML based Video Component
      */
    Video{
        id:videoPlayer
        anchors.fill: videoPlayerItem // never forget to mention the size and position
        source: "Video/Bear.wmv"
        focus: true
```


(C) Copyright Nokia 2013. All rights reserved.

```
    }
}
```

How To Use this Component

Following code snippet shows how to use the above developed QML VideoPlayer Component in other QML file, The important thing we need to keep in mind that, as mentioned above, we can load the Video Component at the start of the application, instead we need to load the Video Element as and when required. For this task a simple Loader element has been used.

To Accomplish this task , a simple toolbar component has been designed which contains three buttons play, pause and stop.

When user Clicks the play button the Loader Components Loads the VideoPlayer QML file and then call the playVideo Function to play the video.

When Stop Button is clicked first stopVideo function is called and then Loader unloads the VideoPlayer QML File.

Please have a look at this QML code to get a idea of how the things have been done.

```
import QtQuick 1.0

Rectangle {
    id: mainRect
    width: 360
    height: 640
    color: "black"
    TopTitleBar{
        id:topTopTitleBar
        x:mainRect.x; y:mainRect.y
        width: mainRect.width; height: mainRect.height/20+10
    }
    Rectangle{
        id:videoRect
        x:mainRect.x; y:topTopTitleBar.y+topTopTitleBar.height
        width: mainRect.width; height: mainRect.height-topTopTitleBar.height-70
        color: "black"

        /*
          Simple Loader is placed inside the Rect which will help to load the videoPlayer Component when
          user clicks the play button
          */
        Loader{
            id:videoLoader
            anchors.fill: videoRect
        }
    }
    /*
      Simple Toolbar Item which has play, pause and stop button
      */
    ToolBar{
        id:toolBar
        x: videoRect.x; y:topTopTitleBar.y+topTopTitleBar.height+videoRect.height;
        width: videoRect.width; height: mainRect.height-videoRect.height-topTopTitleBar.height
        onPlayButtonClicked: {
            console.log("Play Button Clicked")
            videoLoader.source = "VideoPlayer.qml" // sets the Loader to load our VideoPlayer Component
            videoLoader.item.playVideo()// starts the video
        }
        onStopButtonClicked: {
                console.log("Video Stopped"+videoLoader.source)
                videoLoader.item.stopVideo() // stops the video
                videoLoader.source = "" // unloads the QML Video Element component
                Qt.quit() // exits the application
        }
        onPauseButtonClicked: {
                console.log("Pause Suceessful"+videoLoader.source)
                videoLoader.item.pauseVideo()// pause the video
        }
     }
}
```

Sample Application

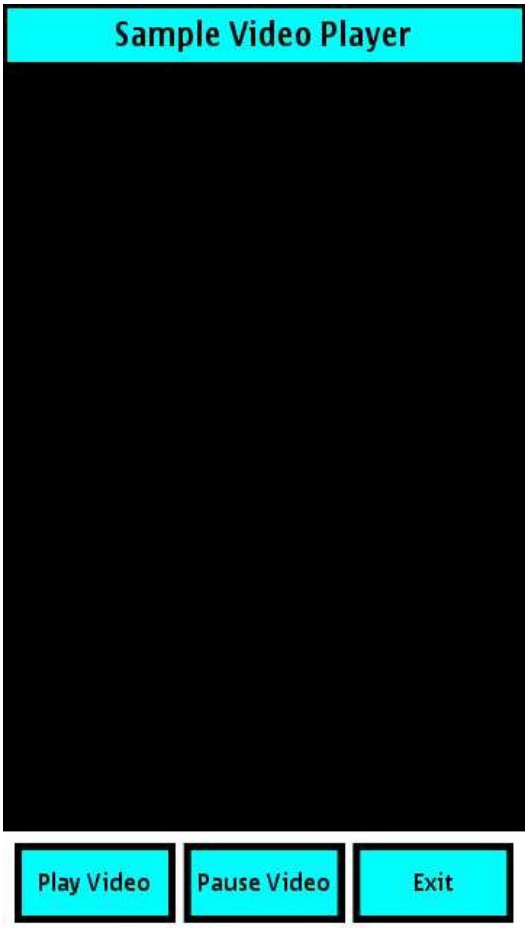
A Sample Application has been developed to show case the usability of QML Video Component. Screenshots for the application are showed at the bottom.

Where One can see the code of Toolbar and the ToolBarButton Components designed, these can be used in any other application too.

Source Code: File:VideoPlayer.zip

Screenshots

(C) Copyright Nokia 2013. All rights reserved.



References

- QML Video Element
- Launch External Applications in Qt

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.