> Author: dailey.dai@openthinks.com

Media

MediaPlayer

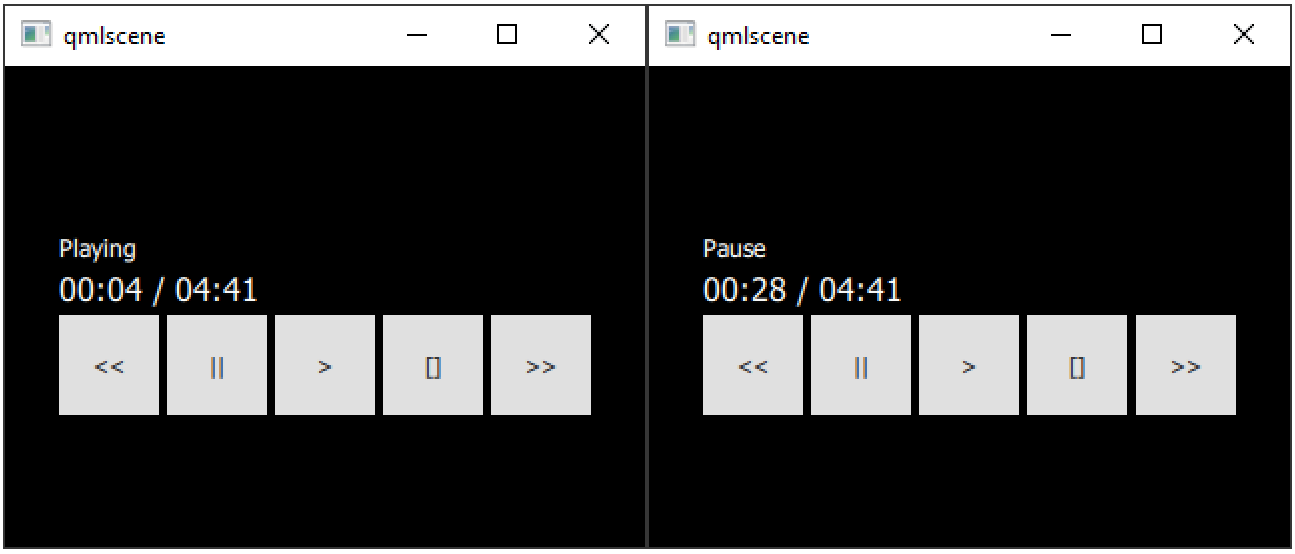
Music

```
//simple_music.qml
import QtQuick 2.2
import QtMultimedia 5.0
Rectangle{
    width: 200;
    height:100;
    MediaPlayer{
        autoPlay: true;
        source: "12.mp3";
    }
}
```

The media player source need add to resource qrc

```
//simple_music_player.qml
import QtQuick 2.0
import QtMultimedia 5.0
import QtQuick.Controls 2.2
Rectangle {
    width: 320;height: 240; color: "black";
    property var utilDate: new Date();
    function msecs2String(msecs)
    {
        utilDate.setTime(msecs);
        return Qt.formatTime(utilDate,"mm:ss");
    }
    MediaPlayer{
        id:player;
        source: "12.mp3";
        onPositionChanged: {
            progress.text = msecs2String(position) + progress.sDuration;
        }
        onDurationChanged: {
            progress.sDuration = " / "+ msecs2String(duration);
        }
        onPlaybackStateChanged: {
            switch(playbackState){
            case MediaPlayer.PlayingState:
                state.text="Playing";
                break;
            case MediaPlayer.PausedState:
                state.text="Pause";
                break;
            case MediaPlayer.StoppedState:
                state.text="Stop";
                break;
            }
        }
    }
    Row{
        id:controller;
        anchors.top:parent.verticalCenter;
        anchors.horizontalCenter: parent.horizontalCenter;
        anchors.topMargin: 4;spacing: 4;
        Button{
            width: 50;height: 50;
            text: "<<";
            onClicked: {
                if(player.seekable)
                    player.seek(player.position-5000);
            }
        }
        Button{
            width: 50;height: 50;
            text: "||";
            onClicked: {
                player.pause();
            }
        }
        Button{
            width: 50;height: 50;
            text: ">";
            onClicked: {
                player.play();
            }
        }
        Button{
            width: 50;height: 50;
            text: "[]";
            onClicked: {
                player.stop();
            }
        }
        Button{
            width: 50;height: 50;
            text: ">>";
            onClicked: {
                if(player.seekable)
                    player.seek(player.position+5000);
            }
        }
    }
    Text{
        id:progress;
        anchors.left: controller.left;
        anchors.bottom: controller.top;
        anchors.bottomMargin: 4;
        color: "white";font.pointSize: 12;
        property string sDuration;

    }
    Text{
        id:state;
        anchors.left: progress.left;
        anchors.bottom: progress.top;
        anchors.bottomMargin: 4;
        color:"white";font.pixelSize: 12;
    }
}
```



Video

```
import QtQuick 2.0
import QtMultimedia 5.0
Rectangle {
    width: 720;height: 480;
    MediaPlayer{
        id:player;
        source: "file:///xxx.mp4";
        onError: {
            console.log(errorString);
        }
    }
    VideoOutput{
        anchors.fill: parent;
        source: player;
    }
    MouseArea{
        anchors.fill: parent;
        onClicked: {
            player.play();
        }
    }
}
```

`VideoOutput` Render video or camera viewfinder

In Windows Platform, make sure install `DirectShow Filter` **LAVFilters**

Meta Data

`metaData` include all `MetaPlayer` meta info.

These properties hold the meta data for the current media.

- metaData.title - the title of the media.
- metaData.subTitle - the sub-title of the media.
- metaData.author - the author of the media.
- metaData.comment - a user comment about the media.
- metaData.description - a description of the media.
- metaData.category - the category of the media.
- metaData.genre - the genre of the media.
- metaData.year - the year of release of the media.
- metaData.date - the date of the media.
- metaData.userRating - a user rating of the media in the range of 0 to 100.
- metaData.keywords - a list of keywords describing the media.
- metaData.language - the language of the media, as an ISO 639-2 code.
- metaData.publisher - the publisher of the media.
- metaData.copyright - the media's copyright notice.
- metaData.parentalRating - the parental rating of the media.
- metaData.ratingOrganization - the name of the rating organization responsible for the parental rating of the media.
- metaData.size - the size of the media in bytes.
- metaData.mediaType - the type of the media.
- metaData.audioBitRate - the bit rate of the media's audio stream in bits per second.
- metaData.audioCodec - the encoding of the media audio stream.
- metaData.averageLevel - the average volume level of the media.
- metaData.channelCount - the number of channels in the media's audio stream.
- metaData.peakValue - the peak volume of media's audio stream.
- metaData.sampleRate - the sample rate of the media's audio stream in hertz.
- metaData.albumTitle - the title of the album the media belongs to.
- metaData.albumArtist - the name of the principal artist of the album the media belongs to.
- metaData.contributingArtist - the names of artists contributing to the media.
- metaData.composer - the composer of the media.
- metaData.conductor - the conductor of the media.
- metaData.lyrics - the lyrics to the media.

- metaData.mood - the mood of the media.
- metaData.trackNumber - the track number of the media.
- metaData.trackCount - the number of tracks on the album containing the media.
- metaData.coverArtUrlSmall - the URL of a small cover art image.
- metaData.coverArtUrlLarge - the URL of a large cover art image.
- metaData.resolution - the dimension of an image or video.
- metaData.pixelAspectRatio - the pixel aspect ratio of an image or video.
- metaData.videoFrameRate - the frame rate of the media's video stream.
- metaData.videoBitRate - the bit rate of the media's video stream in bits per second.
- metaData.videoCodec - the encoding of the media's video stream.
- metaData.posterUrl - the URL of a poster image.
- metaData.chapterNumber - the chapter number of the media.
- metaData.director - the director of the media.
- metaData.leadPerformer - the lead performer in the media.
- metaData.writer - the writer of the media.

See also QMediaMetaData.

Camera

TODO