

基于 QT 的视频播放器的设计与实现

谢杰 白俊鸽

四川大学锦城学院计算机与软件学院 四川 成都 610000

【摘要】随着当前社会对电子产品的使用逐渐增多,使得视频播放软件种类繁多。对于使用者,一个好的播放器需要具备画面清晰、播放不卡顿等具有一系列功能,而基于 QT 的视频播放器能够满足使用者的需求。本文主要讲述使用 QT 开发视频播放器的主要步骤,实现播放、暂停、快进、音量调节等功能。满足使用者需求,更好地适应市场。

【关键字】视频播放器; QT

引言

随着电子产品的发展和普及,播放器在电脑、手机、网络电视、手表等电子产品中出现,越来越多的播放软件开始争取市场。根据调查使用者对播放器的要求要有使用便捷、画质高清,所以目前播放器不断完善功能,满足需求。Qt 以其强大的功能、良好的可移植性逐渐成为一种被广泛使用的 GUI 系统^[1]。QT 后台语言是 C++ 使得开发的播放器在使用速度、图像的显示都具有一定优势,能够满足大部分使用者的需求。

1 开发环境、及播放器内容

文章开发播放器选择的操作系统 Windows 10, 开发工具选择的是 Visual Studio2017 上基于 Visual C++ 的 Qt5。Qt 以其强大的功能、良好的可移植性逐渐成为一种被广泛使用的 GUI 系统^[2]。Qt5 中有自带的 QMediaPlayer, Qt5 具有 Qt Multimedia 模块实现媒体应用,提供了丰富的接口,操作起来也较为简单,足以具备开发一个功能完全的视频播放器。在创建工程项目时,选择 QT 中的 Qt Widgets Application, 该工程是传统的 C++, 采用的是 QML 布局。下一步中我们需要选着添加的组件除了自选的 Core 和 GUI 还需要选着 Multimedia、Multimedia Widgets, 添加这两项之后才能正常使用 QMediaPlayer, 勾选之后就完成了项目的创建。

进入工程后,需要对项目的运行环境进行相关更改,根据自己的操作系统选择配置、平台。文章采用的配置是 Debug 平台选着的是 x64, 64 位操作系统。若为 32 为操作系统就需要进行修改选着 x86。在解决方案资源管理器中可以看到几个跟项目相关的程序,其中 Form

Files 中可以看到一个工程名 .ui 的文件,双击过后进入 Qt 设计师界面,就可以根据文章第 3 点界面设计, 根据自己设计播放器风格进行修改、添加组件,完成一个播放器的界面框架。

文章中播放器功能主要有, 导入一个文件中视频文件包括 mp4、avi 至播放列表、点击列表中播放目录播放视频文件、暂停播放视频、继续播放视频、播放当前视频的下一个播放文件、调节播放器音量大小、调节播放视频的进度。为了保证代码的整洁与结构性,需要对每个功能进行函数的封装。

2 界面设计

2.1 整体布局

界面设计需要使用到的是 Qt 自带的 Qt 设计。在添加组件之前需要在对象查看器中对整个界面进行一个布局,便于之后界面能够按比例进行缩放。在工具中可以添加需要使用到的组件,并为每个组件添加对象名称,便于之后编写程序对每个组件的使用不混乱。其中每个组件的位置、样式可以自由设置。使得播放器更加美化,便于用户能够有一个直观感受,实现与用户的交互。让播放器能够受到使用者的喜欢。

2.2 播放屏幕、播放列表

播放屏幕需要添加组件 QLabel 添加到主界面中,并设置编辑样式表将背景颜色设置成黑色。便于让使用者醒目当前无播放视频。播放列表添加组件 QListView 并可以根据自己的风格编辑样式表。

2.3 按钮、进度条布局

按钮是 QPushButton 组件,按钮的个数和样式可以根据每个按钮的功能来设定,例如本文具有播放、暂停、下一集、播放列表功能,则需要添加 4 个按

作者简介: 1. 谢杰 (1997-), 男, 汉, 四川省成都人, 本科, 四川大学锦城学院, 研究方向: C++ 编程。

2. 白俊鸽 (指导老师) (1980-), 女, 汉, 四川省成都市, 硕士, 讲师, 四川大学锦城学院, 研究方向: 网络安全。

钮, 按钮的样式可以提前画出使用编辑样式表中添加 `border-image` 来设定。进度条使用的是 `horizontalSlider` 组件。在此需要添加 2 个进度条, 一是播放视频进度, 二是音量大小。其中修改进度条的样式表 `QSlider` 中 `QSlider::groove:horizontal` 为进度条样式设置, `QSlider::sub-page:horizontal` 为进度条已经划过地方的样式设置, `QSlider::add-page:horizontal` 为进度条未划过地方的样式设置, `QSlider::handle:horizontal` 为滑动块样式设置。均可根据喜好随意改变进度条样式。

3 功能实现

3.1 将视频文件导入播放列表

在项目的 `.h` 文件中需要添加相关头文件 `QFileDialog` 用于获取播放文件目录、`QStandardItemModel` 定义数据项, 向 `listview` 传递数据项。示例使用的是点击按钮触发获取播放列表事件并将视频文件导入播放目录, 点击按钮触发事件就需要使用槽函数连接, `Qt` 中提供了 `connect` 接口。

例如示例代码:

```
connect(ui->pushButton_playlist,SIGNAL(clicked(
)),this,SLOT(pushButton_playlist_Get playback_path( ));
```

其中 `ui->pushButton_playlist` 是播放列表的按钮对象名称, `SIGNAL(clicked())` 是鼠标点击信号, `SLOT(pushButton_playlist_Get playback_path())` 是当发出信号时触发的事件。

触发事件的内容是获取播放文件目录, 使用的接口是 `QFileDialog::getExistingDirectory`, 得到是选着的目录路径, 可以用 `QString` 将目录存放起来。接着就是选着该目录下我们需要的文件, 使用 `dir.entryList` 接口返回的是 `QStringList` 文件列表, 其中 `QStringList` 对象 `.size()` 可以获得列表大小, 并通过 `for` 循环传递给 `QStandardItemModel` 数据项。使用 `listview->setModel()` 将视频文件显示在 `listview` 中。这样便从电脑中选着视频文件名称显示在播放列表中。本文中将该功能代码封装为 `getFileNames` 函数。

3.2 播放功能

在实现播放功能之前需要在项目 `.h` 文件中添加头文件 `QMediaPlayer` 使用 `Qt` 自带的播放器、`QVideoWidget` 添加播放器控件、`QVBoxLayout` 播放器的布局。添加完成之后需要在类中定义三个对象一一对应, 并在全局中分别 `new` 一个空间。在播放控件中需要设置大小 `resize()`, 根据 `Qt` 设计师中 `lable` 的大小定义, 布局需要设置和 `lable` 的边距 `setMargin()`, 添加设置好的播放控件 `addWidget()`, 并在 `lable` 中设置当前布局 `setLayout()`。在使用播放器之前需要先设置播放器的输出位置 `setVideoOutput()` 填入当前控件, 再设置 `setMedia()` 参数

为播放视频的完整路径。如果调换位置会导致第一次播放成功。而第二次会给出提示媒体无法载入。接着就是使用提供的 `play()` 接口播放视频。

在完成这一步操作时会有一些环境问题需要解决。`Qt` 中自带的播放器使用的是 `Windows` 默认的解码器, 没有安装相关的 `DirectShowService` 解码器无法播放视频, 例如可以使用 `LAV Filters`。安装过后运行项目提示没有 `msvcp120.dll`、`msvcr120.dll` 动态库则需要在项目运行目录中加入。其次若是播放目录中有中文路径名称, 有可能会遇上中文乱码问题, 需要用 `string` 对 `QString` 进行转化一下, 就可以解决中文乱码, 可以单独封装一个函数, 避免代码的冗余以及影响函数的封装性。

3.3 播放进度、音量控制

播放进度的实现首先需要将我们设置的进度条与视频进度相互连接起来, 实现同步。可以定义一个计时器实现视频播放过程中滑块的跳动。在运用计时器时也需要在 `.h` 文件中加入头文件 `QTimer` 并定义一个全局对象。使用计时器时需要定义计时器跳动的频率 `setInterval()`, 并建立槽函数, 当计时器跳动时发出信号, 实现播放进度, 与进度条的同步。计时器跳动信号为 `timeout()`, 槽函数的使用在上文中已有示例。`Qt` 设计师进度条的大小通过 `setValue()` 接口定义, 在 `QMediaPlayer` 中视频播放的进度是 `position()` 接口。视频总时长用 `duration()` 接口表示, 通过运算表达式就可以得出当前播放进度比例并表示在界面进度条中。

除此我们还需要实现快进回放的功能。在拖动进度条时可以分为三个动作, 点击滑块、拉动滑块、释放滑块。因此根据三个动作可以分步实现快进, 回放功能。当点击滑块时, 停止计时器的跳动 `stop()`, 让滑块不移动。拉动滑块时同步视频播放进度 `setPosition()`。释放滑块时开始计时器的跳动 `start()`。其中关于三种不同操作的信号为, 点击滑块 `sliderPressed()`、拖动滑块 `sliderMoved()`、释放滑块 `sliderReleased()`。

音量的控制, 在 `QMediaPlayer` 中也有接口 `setVolume()` 是当前播放视频的音量大小。根据上文播放进度的调节示例便可以完成视频音量与滑块数值的同步, 具体也是三步骤, 点击滑块、拉动滑块、释放滑块。到此也就完成了播放进度与音量大小的控制。

3.4 属于按键功能的具体实现

信号和槽机制是 `QT` 的核心机制, 信号和槽是一种高级接口, 应用于对象之间的通信, 它是 `QT` 的核心特性, 也是 `QT` 区别于其它工具包的重要地方^[3]。所有按键点击事件都需要通过槽函数进行触发 `connect()`, 信号都为 `clicked()`。

实现点击播放列表播放视频。由于也是点击列表中某一项触发事件，所以需要用到槽函数。信号是点击 clicked()。使用前需要添加头文件 QModelIndex 用于定位 viewlist 中的数据项。在使用 QModelIndex 定位到我们选着的数据项后需要用 QString 进行保存。再通过字符串的拼接获得完整的播放路径。在传递给封装好的播放函数就可以实现点击列表视频进行播放。

播放下一个视频功能，在进行获取播放列表时，对该目录下所有视频文件用 QStringList 进行存储，通过 for 循环进行字符串匹配找到当前播放视频，再将播放路径通过 QStringList.at(i+1) 调置下一个播放视频，传递给播放功能遍实现播放下个视频功能。

实现暂停、继续播放，需要通过 Qmediaplayer 提供的接口。首选判断当前播放器的状态 state()，若是处于播放状态则按钮播放不起作用，反正暂停按钮不起作用。其中播放状态中播放 QMediaPlayer::PlayingState、暂停为 QMediaPlayer::PausedState，让视频继续播放使用 play()、暂停 pause()。便完成了播放器的暂停，继续功能。

4 结束语

主要就如何使用 Qt 开发视频播放器的基本功能做了讲解，主要运用到的是 Qt 自带的 QMdeiaolayer 模块。在开发过程中遇见环境问题，需要结合实际情况寻求解决办法。在播放器功能上，还有许多可以完善的地方，例如实现全屏播放、如何连接网络，进行视频搜索并播放视频。若想更深层次的开发播放器，可以往嵌入式系统中开发播放器方向进行研究。所以本文适用于视频播放器开发的初学者。

【参考文献】

- [1] 王建民, 张宏壮. 基于 Qt 的嵌入式媒体播放器系统的设计 [J]. 微计算机信息, 2008(20):64-66.
- [2] 陈云鹤. 基于 Qt 的嵌入式媒体播放器的设计与实现 [D]. 武汉: 华中科技大学, 2006.
- [3] 朱浩悦, 张红祥. 基于 QT 的视频媒体播放器的设计与实现 [J]. 科技信息, 2011(30):274-275.