

基于QT的视频媒体播放器设计与实现

西安文理学院计算机科学与技术系 朱浩悦 张红祥

[摘要]目前常见的视频多媒体软件各有特色,但是都存在着不如人意的地方,如占用系统资源多、支持格式少和界面臃肿等等。本文设计并开发了一个视频媒体播放器,该播放器基于QT集成开发环境,利用系统的Mplayer命令接口来编程,具有体积小、支持的播放格式多、系统资源占用少、支持播放列表编辑、支持各种视频格式等功能。

[关键词]视频 播放器 QT Mplayer

0.引言

视频播放器是指能播放以数字信号形式存储的视频的软件,也指具有播放视频功能的电子器件产品。除了少数波形文件外,大多数视频播放器携带解码器以还原经过压缩的媒体文件,视频播放器还要内置一整套转换频率以及缓冲的算法。如今,PC机所支持的音频、视频播放软件越来越多,想在PC机上听CD、MP3或者欣赏VCD、DVD大片,都已易如反掌。一款款五花八门的播放器软件任你挑选,有的作风精简、有的功能强大,多数都支持所有的常见的媒体格式(.rm、.rmvb、.mpg、.mp4、.avi)。虽然各大厂商所设计制作的播放器都有自己独特的功能和设计理念,但是也都存在着自己的弊端,如暴风影音占用系统资源大,射手播放器的播放列表并不显示到主界面上等。本文将设计并实现一款界面清楚了,操作简单实效,支持多种格式媒体,在播放列表中加载一个或者多个多媒体文件;显示播放进度条;用户能随意改变屏幕尺寸;在多媒体文件播放完后可以选择自动关闭播放器和循环播放功能的视频媒体播放器。

1.功能需求

对于一个播放器来说,用户最基本的要求就是播放视频文件。电脑用户在观看视频时会根据自身需求来调整音量、播放速度、在播放列表中添加文件等,所以播放器必须要满足用户的各种需求。视频有多种格式,所以这款播放器最好能够支持所有的视频格式。

从用户的角度可以得出如图1的系统功能结构图:

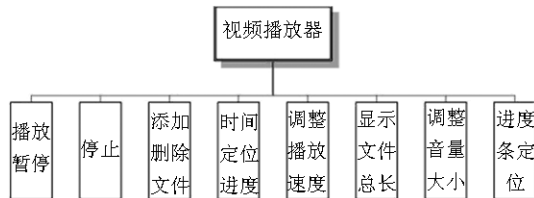


图1 软件功能结构图

2.开发环境介绍

QT是诺基亚开发的一个跨平台的C++图形用户界面应用程序框架。它提供给应用程序开发者建立艺术级的图形用户界面所需的所用功能。QT是完全面向对象的,很容易扩展,并且允许真正地组件编程。自从1996年,QT进入商业领域,它已经成为全世界范围内数千种成功的应用程序的基础。QT也是流行的Linux桌面环境KDE的基础。基本上,QT同X Window上的Motif, Openwin, GTK等图形界面库和Windows平台上的MFC, OWL, VCL, ATL是同类型的东西,但QT具有优良的跨平台特性、面向对象、丰富的API、大量的开发文档等优点。

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。信号和槽是QT自行定义的一种通信机制,它独立于标准的C/C++语言,因此要正确地处理信号和槽,必须借助一个称为MOC(Meta Object Compiler)的QT工具,该工具是一个C++预处理程序,它为高层次的事件处理自动生成所需要的附加代码。

3.功能实现

本播放器界面分为四部分:

- 1)定义QLabel的对象制作播放窗口。
- 2)定义QListWidget的对象控件制作播放列表。
- 3)定义QPushButton的对象制作按钮。
- 4)定义QSlider的对象制作声音控制条和播放进度控制条。

本播放器定义的功能函数如下:

void play(QString fileName)——播放功能的函数。

void next()——播放播放列表中下一个文件功能的函数。

void pre()——播放播放列表中上一个文件功能的函数。

void pause()——播放暂停功能的函数。

void changeVolume(int)——调节音量大小功能的函数。

void setSpeed()——设置播放速度功能的函数。

void dataRecieve()——获取当前播放文件的总时间和当前播放进度功能的函数。

void jumpByTime()——通过输入时间来设置播放进度功能的函数。

void playListItem(QListWidgetItem *)——程序按照播放列表来播放的功能函数。

void closeEvent(QCloseEvent *)——关闭程序功能的函数。

void add()——向播放列表中添加文件功能的函数。

void del()——从播放列表中删除文件功能的函数。

视频播放最为主要的功能是可以正常播放视频格式文件,以下代码是播放功能的函数代码和详细解释。

void MovieForm::play(QString fileName)

```
{
    QStringList args;
    p->kill(); //先关闭原来的进程
    p=new QProcess(this);// 创建一个新的进程
    connect(p,SIGNAL(readyReadStandardOutput()),this,SLOT(dataRecieve())); // readyReadStandardOutput()信号
    //是用来获取新播放文件的信息。将信号与槽链接,有新进程创建就会触发
    args<<"-slave";
    args<<"-quiet";
    args<<"-wid"<<QString::number(view->winId(),10);
    args<<"-vo";
    args<<fileName; //用来获取文件名
    p->start("mplayer",args); //播放函数
    playBtn->setEnabled(true);
    stopBtn->setEnabled(true);
}
```

获取播放文件的总时间和当前播放进度的方法:用p->write("get_time_length\n"); p->write("get_time_pos\n");和p->write("get_percent_pos\n");将时间标准输出显示。通过connect(process,SIGNAL(readyReadStandardOutput()),this,SLOT(dataRecieve()));读出我们需要的信息。process有可读取的信息时,发出信号,在槽函数dataRecieve()中读取信息。dataRecieve()函数是获取播放文件的总时间和当前播放进度的函数。

```
void MPlayer::dataRecieve()
{
    while(process->canReadLine())
    {
        QString message(process->readLine());
    }
}
```

其中message即为读取的信息,我们可以根据需要取我们要的信息,例如:文件总时间为:ANS_LENGTH=23.00,当前时间为:ANS_TIME_POSITION=23.00。

我们用下面函数实现快进功能:

void MovieForm::setSpeed()

```
{
    double speed=QInputDialog::getDouble(this,"设置播放速度");
}
```

```

if(speed>0)
p->write(QString("speed_set "+QString::number(speed)+" 2\n").toUtf8
());
}

```

当speed的数值为1的时候,即为正常速度播放,当value的数值为X时,则为X倍的速度进行视频播放。

我们可以通过输入时间来调整播放进度:

```

QDialog *timeDialog=new QDialog(this);
minBox=new QSpinBox(timeDialog);
QLabel *minLabel=new QLabel("分",timeDialog);
secBox=new QSpinBox(timeDialog);
QLabel *secLabel=new QLabel("秒",timeDialog);
QPushButton *sureBtn=new QPushButton("确定",timeDialog);
connect(sureBtn,SIGNAL(clicked()),this,SLOT(sureBtnClicked()));
timeDialog->show();

```

前6行代码设置了输入时间的对话框,第7行将clicked信号和槽函数sureBtnClicked()连接起来,当点击确定Button时,将命令传递给Mplayer控件实现跳转。

暂停、音量调节、静音功能的实现分别可以通过process->write()来实现。

4. 播放器实现结果

运行视频媒体播放器,会打开初始界面,在没有选定播放文件时,播放、停止按钮以及播放进度条为不可点击状态,上一个和下一个可以选定播放列表中上一个和下一个文件,add按钮选择视频媒体文件添加到播放列表中,del按钮从播放列表中删除当前的视频文件。单击add按钮会弹出选择视频文件的对话框,如图2所示。



图2 添加视频文件图

将视频文件添加到播放列表后,在播放列表中选中文件名称,点击播放按钮或者双击文件名称进行视频播放。此时播放、停止、上一个、下一个按钮都变为可以选择的状态。拖动声音控制条和播放进度控制条可以随意改变音量高低和定位播放时间。点击速度设置按钮播放速度设置对话框,用户在对话框中输入想要播放的速度可以设置视频的播放速度。如图3所示。

视频播放时,点击时间跳转按钮会弹出时间跳转对话框,用户在此对话框中输入想要跳转的时间,播放器在判断输入时间没有超过了视频时间总长后进行跳转,播放进度条也会同步改变。如图4所示。



图3 播放速度设置图



图4 时间跳转图

5. 测试及结论

本播放器在开发完成后,进行了黑盒测试,具体测试内容如下:

- 1)重复打开关闭程序,看系统会不会产生错误。
- 2)在播放过程中多次点击播放、暂停和停止按钮,看每次点击是否都可以产生相应效果。
- 3)在播放列表中添加各种类型的视频媒体文件,看是否能够播放各类文件。
- 4)随机通过拖动播放进度条来控制播放进度。
- 5)通过随机输入时间来进行播放定位。
- 6)随机调整音量的大小。

以上6点测试的通过,基本保证了此播放器可以进行播放、暂停、添加文件、控制播放进度和音量的功能。由于时间的仓促,本次设计基本上完成了预期的目标,实现了播放、暂停、停止、快进、快退、播放列表以及音量调节和时间显示,实现了一个小巧并支持多种格式的基于QT的视频媒体播放器。

参考文献

- [1] W.Richard Stevenson.Stephen A.Rago.UNIX 环境 高级编程[M].人民邮电出版社,2006.
- [2] Stanley B.Lippman And JOSEE Lajoie And Barbara E.Moo.C++ Primer Plus[M].第五版.人民邮电出版社,2006.
- [3] Jasmin Blanchette And Mark Summerfield.C++ GUI Programming with QT4[M].第二版.电子工业出版社,2008.
- [4] 蔡志明,卢传富等.精通Qt4编程[M].电子工业出版社,2008.

(上接第273页) 学生树立克服困难、解决问题的信心^[5]。

3.3 师资队伍的建设与培养

计算机学科是一门不断发展与更新的科学,这就要求任课老师需要不断提高自身的文化修养,时刻获取相关的最新知识,完善充实知识体系。NIIT的榜样学习法很大程度上依赖教师,学院为了培养教师,要求担任每一模块教学任务的教师都要参加NIIT的培训,并取得相关模块的证书。如笔者曾经参加南京邮电大学的课程培训,系统学习了第三模块的课程内容并取得了相关教学证书。另外,备课工作也是任课教师的一项必须认真对待的工作。任课教师在准备教案时,可以通过Internet和多媒体技术来获得大量较新的有关计算机学科的技术资料、信息等。结合企业具体的实际案例,从而使得教学内容更加的丰富,更具有针对性,突出实用性。

4. 总结

NIIT在偏重于应用的计算机教育中具有突出的优势,与NIIT合作的高校本科教育是一大进步,但合作也给课程设置、学生培养方式和教师能力等方面提出了新的要求,本科计算机教育只有合理地对待这些问题,在合作过程中不丢失传统教学优势又引入先进教学模式,才能成

功进行教学改革,才能真正提高学生的素质和就业率。优集学院与NIIT合作的方式是嵌入式的四学期软件教育,笔者自担任NIIT教学工作以来,在教学过程中全程使用了项目教学法,给学生树立了成功的典范。从最后的考核成绩和项目完成情况来看,NIIT的教学取得了良好的效果。

参考文献

- [1] 中国高等院校计算机基础教育课程改革课题调研组.中国高等院校计算机基础教育课程体系(CFC2006)[M].北京:清华大学出版社,2006
- [2] 霍立红.新时代计算机教改心得[J].中国科教创新导刊,2009,25: 177
- [3] 章华,苏一丹.印度NIIT教学法及其在高校教学中应用的研究[J].广西大学学报,2004,9:7-9
- [4] 刘志成,李忠华,熊芊.全面剖析NIIT教学体系[J].计算机教育,2007,9:47-50
- [5] 任周.基于计算机网络课程教改的研究[J].软件导刊,2009,8: 187-188