Tedu.cn
达内教育

欢迎大家来到第五阶段课程

《分布式流媒体》实训项目

预习课

TNV DAY05

预习课

预习内容

媒体播放器（6）

预习课

Tedu.cn
达内教育

媒体播放器（6）

主窗口类

知识讲解

- 响应进度滑块值改变信号的槽函数(拖拽滑块或点击滑轨)
  - 若当前媒体时间不同于设置媒体时间
    - 定位VLC媒体
    - 在响应时间改变事件的处理函数中，根据媒体时间的毫秒值调整进度滑块的位置，也会引发值改变信号，此处若不加判断势必构成死循环
- 响应播放按钮点击信号的槽函数
  - 若网络媒体，则创建针对URL的VLC媒体，否则创建针对本地路径的VLC媒体
  - 将VLC媒体设置到VLC媒体播放器中
  - 播放VLC媒体
  - 若内嵌视频，则禁止视频框控件更新显示，避免闪烁
  - 调整界面

主窗口类

知识讲解

- 响应暂停按钮点击信号的槽函数
  - 若正在播放，则暂停VLC媒体，否则播放VLC媒体
- 响应停止按钮点击信号的槽函数
  - 停止VLC媒体
  - 若内嵌视频，则激活视频框控件更新显示
  - 调整界面
- 响应快进按钮点击信号的槽函数
  - 向前步进媒体长度的百分之一
  - 定位VLC媒体

主窗口类

- 响应快退按钮点击信号的槽函数
  - 向后步进媒体长度的百分之一
  - 定位VLC媒体
- 响应音量滑块移动信号的槽函数(拖拽滑块)
  - 调整音量
  - 设置静音
- 响应音量滑块值改变信号的槽函数(点击滑轨)
  - 若当前音量不同于设置音量
    - 调整音量
    - 设置静音

知识讲解

主窗口类

- 响应关于按钮点击信号的槽函数
  - 显示版本消息框
- 响应退出按钮点击信号的槽函数
  - 关闭窗口

知识讲解

Tedu.cn
达内教育

附录：程序清单

QtPlayer/Src/MainWindow.cpp

知识讲解

```
//响应进度滑块值改变信号的槽函数(拖拽滑块或点击滑轨)
void MainWindow::on_sliderProgress_valueChanged(int value)
{
    //在响应时间改变事件的处理函数中，根据媒体时间的毫秒值调整进度
    //滑块的位置，也会引发值改变信号，此处若不加判断势必构成死循环
    if (libvlc_media_player_get_time(vlcMediaPlayer) != value)
        //定位VLC媒体
        libvlc_media_player_set_time(vlcMediaPlayer, value);
}

//响应播放按钮点击信号的槽函数
void MainWindow::on_btnPlay_clicked()
{
    string url = ui->editURL->text().toStdString();
```

QtPlayer/Src/MainWindow.cpp

知识讲解

```
vector<string> protocols;
protocols.push_back("http");
protocols.push_back("https");
protocols.push_back("ftp");
protocols.push_back("rstp");

vector<string>::const_iterator it;
for (it = protocols.begin(); it != protocols.end(); ++it)
    if (!url.compare(0, it->size(), *it))
        break;

//若网络媒体...
if (it != protocols.end())
{
    //创建VLC媒体
```

QtPlayer/Src/MainWindow.cpp

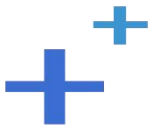

```
        if (!(vlcMedia = libvlc_media_new_location(vlcInstance, url.c_str())))
        {
            QMessageBox(QMessageBox::Critical, windowTitle(),
                "无法创建VLC媒体！", QMessageBox::Ok, this).exec();
            return;
        }
    }
    else //否则，即本地媒体...
    {
        //创建VLC媒体
        if (!(vlcMedia = libvlc_media_new_path(vlcInstance, url.c_str())))
        {
          QMessageBox(QMessageBox::Critical, windowTitle(),
              "无法创建VLC媒体！", QMessageBox::Ok, this).exec();
          return;
```

QtPlayer/Src/MainWindow.cpp


```
    }
}

// 将VLC媒体设置到VLC媒体播放器中
libvlc_media_player_set_media(vlcMediaPlayer, vlcMedia);

// 播放VLC媒体
if (libvlc_media_player_play(vlcMediaPlayer) == -1)
{
    QMessageBox(QMessageBox::Critical, windowTitle(),
        "无法播放VLC媒体！", QMessageBox::Ok, this).exec();
    return;
}

// 若内嵌视频...
```

QtPlayer/Src/MainWindow.cpp

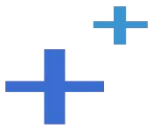

```
    if (embed)
        //禁止视频框控件更新显示，避免闪烁
        ui->frmVideo->setUpdatesEnabled(false);

    //调整界面
    ui->sliderProgress->setEnabled(true);
    ui->btnPlay->setEnabled(false);
    ui->btnPause->setEnabled(true);
    ui->btnStop->setEnabled(true);
    ui->btnFastForward->setEnabled(true);
    ui->btnFastBackward->setEnabled(true);
}

//响应暂停按钮点击信号的槽函数
void MainWindow::on_btnPause_clicked()
```



QtPlayer/Src/MainWindow.cpp

知识讲解

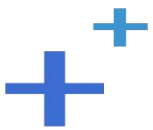
```
{
    if (libvlc_media_player_is_playing(vlcMediaPlayer))
    {
        // 暂停VLC媒体
        libvlc_media_player_pause(vlcMediaPlayer);
        ui->btnPause->setText("继续");
    }
    else
    {
        // 播放VLC媒体
        libvlc_media_player_play(vlcMediaPlayer);
        ui->btnPause->setText("暂停");
    }
}
```



QtPlayer/Src/MainWindow.cpp

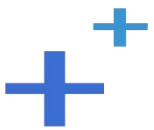
知识讲解

```
//响应停止按钮点击信号的槽函数
void MainWindow::on_btnStop_clicked()
{
    //停止VLC媒体
    libvlc_media_player_stop(vlcMediaPlayer);

    //若内嵌视频...
    if (embed)
        //激活视频框控件更新显示
        ui->frmVideo->setUpdatesEnabled(true);

    //调整界面
    ui->sliderProgress->setValue(0);
    ui->sliderProgress->setEnabled(false);
    ui->labProgress->setText("00:00:00.000/00:00:00.000");
```



QtPlayer/Src/MainWindow.cpp


```
    ui->btnPlay->setEnabled(true);
    ui->btnPause->setEnabled(false);
    ui->btnPause->setText("暂停");
    ui->btnStop->setEnabled(false);
    ui->btnFastForward->setEnabled(false);
    ui->btnFastBackward->setEnabled(false);
}

//响应快进按钮点击信号的槽函数
void MainWindow::on_btnFastForward_clicked()
{
    //向前步进媒体长度的百分之一
    libvlc_time_t time = libvlc_media_player_get_time(vlcMediaPlayer) +
        vlcMediaLength / 100;
```



QtPlayer/Src/MainWindow.cpp


```
    //定位VLC媒体
    libvlc_media_player_set_time(vlcMediaPlayer,
        time < vlcMediaLength ? time : vlcMediaLength);
}

//响应快退按钮点击信号的槽函数
void MainWindow::on_btnFastBackward_clicked()
{
    //向后步进媒体长度的百分之一
    libvlc_time_t time = libvlc_media_player_get_time(vlcMediaPlayer) -
        vlcMediaLength / 100;

    //定位VLC媒体
    libvlc_media_player_set_time(vlcMediaPlayer,
        time > 0 ? time : 0);
```

QtPlayer/Src/MainWindow.cpp


```
}

//响应音量滑块移动信号的槽函数(拖拽滑块)
void MainWindow::on_sliderVolume_sliderMoved(int position)
{
    //调整音量
    libvlc_audio_set_volume(vlcMediaPlayer, position);
    //设置静音
    libvlc_audio_set_mute(vlcMediaPlayer, !position);
}

//响应音量滑块值改变信号的槽函数(点击滑轨)
void MainWindow::on_sliderVolume_valueChanged(int value)
{
    if (libvlc_audio_get_volume(vlcMediaPlayer) != value)
```

QtPlayer/Src/MainWindow.cpp


```
    {
        //调整音量
        libvlc_audio_set_volume(vlcMediaPlayer, value);
        //设置静音
        libvlc_audio_set_mute(vlcMediaPlayer, !value);
    }
}

//响应关于按钮点击信号的槽函数
void MainWindow::on_btnAbout_clicked()
{
    //显示版本
```

QtPlayer/Src/MainWindow.cpp

知识讲解

```
    QMessageBox(QMessageBox::Information, windowTitle(),
        "基于Qt和libVLC的流媒体播放器\n\n"
        "版本：1.0\n\n"
        "版权所有(C)达内科技,2020",
        QMessageBox::Ok, this).exec();
}

//响应退出按钮点击信号的槽函数
void MainWindow::on_btnQuit_clicked()
{
    //关闭窗口
    close();
}
```

直播课见