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 DAY04

预习课

预习内容

媒体播放器（4）

预习课

媒体播放器（4）

主窗口类

- 成员变量
  - 用户界面：ui
  - 内嵌视频：embed
  - VLC实例：vlcInstance
  - VLC媒体播放器：vlcMediaPlayer
  - VLC事件管理器：vlcEventManager
  - VLC媒体：vlcMedia
  - VLC媒体长度：vlcMediaLength

知识讲解

主窗口类

知识讲解

- 构造函数
  - 实例化用户界面
  - 构造VLC参数列表
  - 创建VLC实例
  - 创建VLC媒体播放器
  - 视频窗口模式
    - 内嵌视频窗口
      - 将视频框控件设置为VLC播放器的输出窗口
      - 为视频框控件安装事件过滤器，以截获鼠标双击事件，切换全屏和窗口模式
    - 独立视频窗口
      - 销毁屏幕框控件(及其子控件——视频框控件)
  - 创建VLC事件管理器

主窗口类

知识讲解

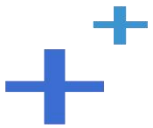
- 构造函数
  - 设置VLC事件回调
    - 响应媒体长度改变事件——设置进度滑块控件
    - 若内嵌视频
      - 响应视频输出启动事件——枚举视频框控件的所有子窗口并禁用之，在Windows上若不禁用视频框控件的子窗口，即VLC视频输出窗口，鼠标停留其上将导致界面崩溃
    - 响应时间改变事件——同步进度滑块控件和进度标签控件
    - 响应终点到达事件——停止播放
    - 初始化界面
    - 初始化音量
- 析构函数
  - 销毁VLC媒体播放器、VLC实例和用户界面

附录：程序清单

QtPlayer/Src/MainWindow.h

知识讲解

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <vlc/vlc.h>
#ifdef Q_OS_WINDOWS
#include <windows.h>
#endif // Q_OS_WINDOWS


QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
```



QtPlayer/Src/MainWindow.h

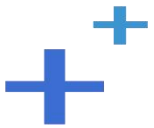
知识讲解

```
{
    Q_OBJECT

public:
    //构造函数
    MainWindow(QWidget *parent = nullptr, bool embed = false);
    //析构函数
    ~MainWindow();

private:
    //响应媒体长度改变事件的静态处理函数
    static void onMediaPlayerLengthChanged(
        libvlc_event_t const* event, void* data);
    //响应媒体长度改变事件的普通处理函数
    void onMediaPlayerLengthChanged(libvlc_time_t length);
```

QtPlayer/Src/MainWindow.h

知识讲解

```
#ifdef Q_OS_WINDOWS
    //子窗口枚举回调函数
    static BOOL CALLBACK onEnumVLCWindow(HWND hwnd, LPARAM);
    //响应视频输出启动事件的静态处理函数
    static void onMediaPlayerVout(
        libvlc_event_t const*, void* data);
    //响应视频输出启动事件的普通处理函数
    void onMediaPlayerVout(void);
#endif // Q_OS_WINDOWS

    //响应时间改变事件的静态处理函数
    static void onMediaPlayerTimeChanged(
        libvlc_event_t const* event, void* data);
    //响应时间改变事件的普通处理函数
    void onMediaPlayerTimeChanged(libvlc_time_t time);
```



QtPlayer/Src/MainWindow.h

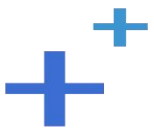
知识讲解

```
    //响应终点到达事件的静态处理函数
    static void onMediaPlayerEndReached(
        libvlc_event_t const*, void* data);
    //响应终点到达事件的普通处理函数
    void onMediaPlayerEndReached(void);

    //显示窗口时被调用的虚函数
    void showEvent(QShowEvent*);
    //视频框控件有事件时被调用的虚函数
    bool eventFilter(QObject* obj, QEvent* event);

private slots:
    //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 on_sliderProgress_valueChanged(int value);
    //响应播放按钮点击信号的槽函数
```

QtPlayer/Src/MainWindow.h

知识讲解

```
void on_btnPlay_clicked();
//响应暂停按钮点击信号的槽函数
void on_btnPause_clicked();
//响应停止按钮点击信号的槽函数
void on_btnStop_clicked();
//响应快进按钮点击信号的槽函数
void on_btnFastForward_clicked();
//响应快退按钮点击信号的槽函数
void on_btnFastBackward_clicked();
//响应音量滑块移动信号的槽函数(拖拽滑块)
void on_sliderVolume_sliderMoved(int position);
//响应音量滑块值改变信号的槽函数(点击滑轨)
void on_sliderVolume_valueChanged(int value);
//响应关于按钮点击信号的槽函数
void on_btnAbout_clicked();
```



QtPlayer/Src/MainWindow.h

知识讲解

```
    //响应退出按钮点击信号的槽函数
    void on_btnQuit_clicked();

private:
    Ui::MainWindow *ui;

    bool embed; //内嵌视频

    libvlc_instance_t*      vlcInstance;     //VLC实例
    libvlc_media_player_t*  vlcMediaPlayer;  //VLC媒体播放器
    libvlc_event_manager_t* vlcEventManager; //VLC事件管理器
    libvlc_media_t*         vlcMedia;        //VLC媒体
    libvlc_time_t           vlcMediaLength;  //VLC媒体长度
};
#endif // MAINWINDOW_H
```

QtPlayer/Src/MainWindow.cpp


```
#include <QMessageBox>
#include <QTime>
#include <QKeyEvent>
#include <string>
#include <vector>
using namespace std;
#include "MainWindow.h"
#include "ui_MainWindow.h"

//构造函数
MainWindow::MainWindow(QWidget *parent, bool embed)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
    , embed(embed)
    , vlcInstance(NULL)
```

QtPlayer/Src/MainWindow.cpp

知识讲解

```
    , vlcMediaPlayer(NULL)
    , vlcEventManager(NULL)
{
    ui->setupUi(this);

    // 设置VLC插件路径环境变量
#ifndef Q_OS_WINDOWS
    if (setenv("VLC_PLUGIN_PATH", "/usr/lib/vlc/plugins", 0) == -1)
    {
        QMessageBox(QMessageBox::Critical, windowTitle(),
            "无法设置VLC插件路径环境变量！", QMessageBox::Ok, this).exec();
        return;
    }
#endif // Q_OS_WINDOWS
```

QtPlayer/Src/MainWindow.cpp

知识讲解

```
//创建VLC实例
char const* const args[] = {"-I", "dumy", "--ignore-config"};
if (!(vlcInstance = libvlc_new(sizeof(args) / sizeof(args[0]), args)))
{
    QMessageBox(QMessageBox::Critical, windowTitle(),
        "无法创建VLC实例！", QMessageBox::Ok, this).exec();
    return;
}

//创建VLC媒体播放器
if (!(vlcMediaPlayer = libvlc_media_player_new(vlcInstance)))
{
    QMessageBox(QMessageBox::Critical, windowTitle(),
        "无法创建VLC媒体播放器！", QMessageBox::Ok, this).exec();
    return;
```

QtPlayer/Src/MainWindow.cpp

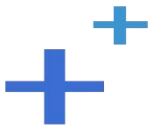
知识讲解

```
    }

    //若内嵌视频...
    if (embed)
    {
        //将视频框控件设置为VLC播放器的输出窗口
#ifdef Q_OS_WINDOWS
        libvlc_media_player_set_hwnd(
            vlcMediaPlayer, reinterpret_cast<void*>(ui->frmVideo->winId()));
#else
        libvlc_media_player_set_xwindow(vlcMediaPlayer, ui->frmVideo->winId());
#endif // Q_OS_WINDOWS

        //为视频框控件安装事件过滤器，以截获鼠标双击事件，切换全屏和窗口模式
        ui->frmVideo->installEventFilter(this);
```

QtPlayer/Src/MainWindow.cpp

知识讲解

```
}
else //否则...
    //销毁屏幕框控件(及其子控件————视频框控件)
    delete ui->frmScreen;

//创建VLC事件管理器
if (!(vlcEventManager = libvlc_media_player_event_manager(vlcMediaPlayer)))
{
    QMessageBox(QMessageBox::Critical, windowTitle(),
        "无法创建VLC事件管理器！", QMessageBox::Ok, this).exec();
    return;
}

//设置VLC事件回调
```



QtPlayer/Src/MainWindow.cpp

知识讲解

```
    //响应媒体长度改变事件————设置进度滑块控件
    libvlc_event_attach(
        vlcEventManager, libvlc_MediaPlayerLengthChanged,
        onMediaPlayerLengthChanged, this);
#ifdef Q_OS_WINDOWS
    //若内嵌视频...
    if (embed)
        //响应视频输出启动事件————枚举视频框控件的所有子窗口并禁用之
        //在Windows上若不禁用视频框控件的子窗口，即VLC视频输出窗口，
        //鼠标停留其上将导致界面崩溃
        libvlc_event_attach(
            vlcEventManager, libvlc_MediaPlayerVout,
            onMediaPlayerVout, this);
#endif // Q_OS_WINDOWS
    //响应时间改变事件————同步进度滑块控件和进度标签控件
```

QtPlayer/Src/MainWindow.cpp


```
    libvlc_event_attach(
        vlcEventManager, libvlc_MediaPlayerTimeChanged,
        onMediaPlayerTimeChanged, this);
    //响应终点到达事件————停止播放
    libvlc_event_attach(
        vlcEventManager, libvlc_MediaPlayerEndReached,
        onMediaPlayerEndReached, this);

    //初始化界面
    ui->editURL->setEnabled(true);
    ui->btnPlay->setEnabled(true);

    //初始化音量
    libvlc_audio_set_volume(vlcMediaPlayer, ui->sliderVolume->value());
}
```

QtPlayer/Src/MainWindow.cpp


```
//析构函数
MainWindow::~MainWindow()
{
    //销毁VLC媒体播放器
    if (vlcMediaPlayer)
    {
        libvlc_media_player_release(vlcMediaPlayer);
        vlcMediaPlayer = NULL;
    }

    //销毁VLC实例
    if (vlcInstance)
    {
        libvlc_release(vlcInstance);
        vlcInstance = NULL;
    }

    delete ui;
}
```

直播课见