

### 三设计过程和算法描述（宋体，四号）

#### 6. 2. 1 利用 VC6.0 的 MFC AppWizard 创建工程

首先：注册该控件在 windows “运行” 里写入：regsvr32 msdrm.ocx 单击确定

1. 在 VC++集成开发环境中，选择菜单 File|New，弹出 New 对话框。
2. 在 Projects 选项卡中选择 MFC AppWizard (exe)，并在 Location 项中输入“D:\Samples”。
3. 在 Project name 中输入“ CTLY2412”，项中的内容自动设置为“D:\Samples\Player”。
4. 按下 OK 按钮，在弹出的 MFC AppWizard-Step-1 对话框中选择 Dialog based 项，生成基于对话框的应用程序。
5. 按下 Finish 按钮，在弹出的对话框中按下 OK 按钮，等待创建完相应的工程。

#### 6. 2. 2 为对话框添加菜单

1. 在工作区中单击 Resource View 选项卡，展开 Dialog 资源组，双击 IDD\_PLAYER\_DIALOG。在右边的工作台中显示可以进行可视化编辑的 IDD\_PLAYER\_DIALOG 对话框，并打开一个控件工具窗口。删除 AppWizard 在对话框中生成的控件,即一个 OK 按钮、一个 Cancel 按钮和一个文本框。
2. 鼠标右击 Dialog 资源组中的 IDD\_PLAYER\_DIALOG 条目，在弹出的菜单中选择 Insert...选项。这时，显示 Insert Resource（增加资源）窗口，选择其中的 Menu 项，单击右边的 New 按钮。这样，VC++6.0 在资源文件 Player.rc 中增加了一项资源，即一个菜单。
3. 双击 ID 号：IDR\_MENU1，则在右边显示一个可编辑的菜单条，该菜单条的最右边是一个虚方框，表示最右边的一个菜单项。

4. 双击虚方框，弹出 Menu Item Properties（菜单项属性）对话框，在“C 编辑”框中输入菜单标题：文件（&F），其中“&”的作用是使其后面的字符加上下划线，让用户可用键盘来选择菜单。另外，“O 弹出”检查框默认选中，表明新加的菜单可以有其弹出的子菜单。
5. 在“文件”菜单下有一个虚方框，表示要新建的菜单项。在“C 标题”编辑框中输入：打开（&O），在 ID 编辑框中输入：ID\_FILEOPEN（当“O 弹出”检查框未选中时，必须输入 ID）。另外，可根据需要在“M 提示”编辑框中输入一串字符，提示该菜单选项功能。当光标停在该菜单选项上时，提示字符串显示在应用程序主窗口的状态条中。
6. 类似步骤 5）、6），按下表 6-1 内容，建立完整的菜单。

表 6-1 媒体播放器的各菜单项属性

顶层菜单项	子菜单项	ID 号
文件(&F)	打开(&O)	ID_FILEOPEN
	关闭(&C)	ID_FILECLOSE
	具有 Separator 样式的菜单分隔条	
	退出(&X)	ID_FILEEXIT
播放(&P)	开始(&S)	ID_PLAYSTART
	暂停(&P)	ID_PLAYPAUSE
	停止(&T)	ID_PLAYSTOP
视频(&V)	原始大小(&O)	ID_VIDEO1X
	原始大小的 2 倍(&D)	ID_VIDEO2X
音频(&A)	调节音量(&V)	ID_AUDEVOLUME
	调节左右声道平衡(&B)	ID_AUDEOBALANCE

帮助(&H)	关于 播放器(&A)	ID_HELPABOUT
--------	------------	--------------

## 7. 给新建的菜单连接一个类

① 将鼠标移至编辑状态的菜单上，单击右键，弹出快捷菜单，选择 ClassWizard 命令。在显示 ClassWizard 窗口的同时，弹出一个 Adding a Class 对话框，该对话框询问是增加一个新类，还是选择一个已经存在的类进行连接。因为我们要连接的类是与 Player 应用程序相关联的对话框类 CPlayerDlg，此类已经存在，所以选择第二项。

② 单击 Adding a Class 对话框中的 OK 按钮，显示一个 Select Class（选择类）对话框窗口，在 Class list 列表中选择 CPlayerDlg，单击 Select 按钮，确定所选项。

③ 单击“确定”按钮，关闭 ClassWizard 窗口。现在，菜单 ID\_PLAYER\_MENU 已经和 CPlayerDlg 类连接起来了。

## 8. 连接菜单和应用程序的主窗口

① 在工作区中单击 Resource View 选项卡，展开 Dialog 资源组，双击 IDD\_PLAYER\_DIALOG。在右边工作台中显示的 IDD\_PLAYER\_DIALOG 对话框的任何一个空白区域，单击鼠标右键，在弹出的菜单中选择 Properties 项，随后弹出 Dialog Properties 对话框。

② 在 Dialog Properties 对话框中有一个“M 菜单”选项。单击“M 菜单”组合框的箭头，在弹出的列表中选择 IDR\_MENU

## 9. 选择 File|Save 命令，保存所有工作。

现在，对已存在的程序进行编译、运行

## 6. 2. 3 在对话框中添加 ActiveMovie 控件

在对话框中单击鼠标右键，选择 Insert ActiveX Control 命令。在弹出的对话框中选择 ActiveMovieControl Object，并按“确定”按钮。

10. 为新添加的 ActiveMovie 控件确定 ID 号，为 IDC\_AMOVIE。

11. 为 IDC\_AMOVIE 映射一个成员变量

在 ActiveMovie 控件上单击右键，在弹出菜单中选择 ClassWizard，在弹出的 MFC ClassWizard 对话框中选择 Member Variables 选项卡，按下 Add Variable 按钮，弹出对话框。在随后弹出对话框中，按下“OK”按钮，以确定新建类：CActiveMovie。在弹出的 Add Member Variable 对话框中确定与 IDC\_AMOVIE 相映射的成员变量：m\_AMovie。

#### 6. 2. 4 完成“文件”菜单的各子菜单功能

以下的函数基本上是通过 MFC 的消息映射机制实现

12. 实现“打开”菜单功能

① 为“打开”菜单添加消息响应函数：OnFileopen()。

② OnFileopen() 函数的代码如下：

```
void CTLY2412Dlg::OnFileopen()
{
    // TODO: Add your command handler code here

    //文件类型过滤

    char szFileFilter[]=

    "Mp3 File(*.mp3)|*.mp3|"

    "Wma File(*.wma)|*.wma|"

    "Video File(*.dat)|*.dat|"

    "Wave File(*.wav)|*.wav|"

    "AVI File(*.avi)|*.avi|"
```

```

"Movie File(*.mov)|*.mov|"
"Media File(*.mmm)|*.mmm|"
"Mid File(*.mid;*.rmi)|*.mid;*.rmi|"
"MPEG File(*.mpeg)|*.mpeg|"
"All File(*.*)|*.*||";

CTLY2412Dlg dlg(TRUE, NULL, NULL, OFN_HIDEREADONLY, szFileFilter);

if(dlg.DoModal()==IDOK)

    {

CString PathName=dlg.GetPathName();

PathName.MakeUpper();

m_AMovie.SetFileName(PathName);

}

}

```

### 13. 实现“关闭”菜单功能

- ① 为“关闭”菜单添加消息响应函数：OnFileclose()。
- ② OnFileclose()函数的代码如下：

```

void CTLY2412Dlg::OnFileclose()

{

// TODO: Add your command handler code here

m_AMovie.SetFileName("");

}

```

### 14. 实现“退出”菜单功能

- ① 为“退出”菜单添加消息响应函数：OnFileexit()。
- ② OnFileexit()函数的代码如下：

```

void CTLY2412Dlg::OnFileexit()
{
// TODO: Add your command handler code here

OnCancel();
}

```

## 6. 2. 5 完成“播放”菜单的各子菜单功能

### 1. 实现“开始”菜单功能

① 为“开始”菜单添加消息响应函数：OnPlaystart()。

② OnPlaystart() 函数的代码如下：

```

void CTLY2412Dlg::OnPlaystart()
{
// TODO: Add your command handler code here

m_AMovie.Run();

SetTimer(0, 20, NULL); //设置定时器
}

```

### 2. 实现“暂停”菜单功能

① 为“暂停”菜单添加消息响应函数：OnPlaypause()。

② OnPlaypause() 函数的代码如下：

```

void CTLY2412Dlg::OnPlaypause()
{
// TODO: Add your command handler code here

m_AMovie.Pause();
}

```

### 3. 实现“停止”菜单功能

① 为“停止”菜单添加消息响应函数：OnPlaystop()。

② OnPlaystop() 函数的代码如下：

```
void CTLY2412Dlg::OnPlaystop()
{
    // TODO: Add your command handler code here

    m_AMovie.Stop();

    KillTimer(0); //关掉定时器
}
```

4. 当 ActiveMovie 控件完成媒体文件的加载时自动播放该文件，代码如下：

```
void CTLY2412Dlg::OnOpenCompleteAmovie()
{
    m_AMovie.Run();
}
```

## 6. 2. 6 完成“视频”菜单的各子菜单功能

### 1. 实现“原始大小”菜单功能

15. 编写代码完善 OnInitDialog(), 完整代码如下：

```
BOOL CTLY2412Dlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    SetIcon(m_hIcon, TRUE);

    SetIcon(m_hIcon, FALSE);

    // 设置 ActiveMovie 控件的初始文件为空
```

```

m_AMovie.SetFileName("");

// 改变窗口的大小以适应 ActiveMovie 控件的大小

MoveMovieWindow();

return TRUE;

}

```

16. 添加 **CTLY2412Dlg** 类的成员函数 **MoveMovieWindow()**, 实现改变对话框大小以适应 **ActiveMovie** 控件的大小, 其代码如下:

```

void CTLY2412Dlg::MoveMovieWindow()

{

CRect rc1, rc2, rc3;

// 得到 ActiveMovie 控件的大小

m_AMovie.GetWindowRect(rc1);

// 保证对话框客户区的宽不小于 300 像素, 高不小于 225 像素

if (rc1.Width()<300 || rc1.Height()<225)

{

rc1.right=rc1.left+300;

rc1.bottom=rc1.top+225;

}

// 获得对话框的大小

GetWindowRect(rc2);

// 获得对话框客户区的大小

GetClientRect(rc3);

// 改变对话框的大小以适应 ActiveMovie 控件的大小

MoveWindow(rc2.left, rc2.top,

```



```

rc2.Width()-rc3.Width()+rc1.Width(),

rc2.Height()-rc3.Height()+rc1.Height());

// 获得 ActiveMovie 控件的大小

m_AMovie.GetWindowRect(rc1);

GetClientRect(rc3);

// 使 ActiveMovie 控件在对话框的客户区居中

m_AMovie.MoveWindow((rc3.Width()-rc1.Width())/2,

(rc3.Height()-rc1.Height())/2, rc1.Width(), rc1.Height());

}

```

③ 为“原始大小”菜单添加消息响应函数：OnVideo1x()。

17. OnVideo1x() 函数的代码如下：

```

void CTLY2412Dlg::OnVideo1x()

{

// TODO: Add your command handler code here

m_AMovie.Pause();

m_AMovie.SetMovieWindowSize(1);

m_AMovie.Run();

}

```

⑤ 另外，要完成“视频”菜单的功能，还应完成下列函数的编写

```

// 当 ActiveMovie 控件的准备状态发生改变时，在需要的情况下改变对话框

的大小以// 适应 ActiveMovie 控件的大小

void CTLY2412Dlg::OnReadyStateChangeAmovie(long ReadyState)

{

MoveMovieWindow();

```

```

}

// 当 ActiveMovie 控件的状态发生改变时, 在需要的情况下改变对话框的大
小以适应

// ActiveMovie 控件的大小

void CTLY2412Dlg::OnStateChangeAmovie(long oldState, long newState)
{
    MoveMovieWindow();
}

```

这两个函数都是利用 MFC ClassWizard 添加的, 方法为: 在 Message Maps 标签中的 Class Name 项中选择 CVideoPlayerDlg, 在 Object IDs 中选择 IDC\_AMOVIE, 在 Messages 中先后选择 StateChange 和 ReadyStateChange, 按下“确定”按钮, 两个函数添加成功。

## 2. 实现“原始大小的两倍”菜单功能

① 为“原始大小的两倍”菜单添加消息响应函数: OnVideo2x()。

② OnVideo2x() 函数的代码如下:

```

void CTLY2412Dlg::OnVideo2x()
{
    // TODO: Add your command handler code here

    m_AMovie.Pause();

    m_AMovie.SetMovieWindowSize(2);

    m_AMovie.Run();
}

```

## 6. 2. 7 完成“音频”菜单的各子菜单功能

## 1. 实现“调节音量”菜单功能

### 18. 设计“音量调节”对话框

**1** 在工程中，插入对话框资源（方法如前所示）。

**2** 按图 6-15 所示设计好“音量调节”对话框。

**3** 使用 ClassWizard 为该对话框创建新的类 CVolumeDlg，并为滑块控件映射类型为 CSliderCtrl 的成员变量 m\_sld。

**4** 编写类 CVolumeDlg 的代码

为该类添加 protect 型变量，声明如：CActiveMovie \*m\_pAMovie;

用 MFC 为该类添加成员函数 void SetVolume(CActiveMovie\* pAMovie);并添加下列代码

```
void CVolumeDlg::SetVolume(CActiveMovie * pAMovie)
{
    m_pAMovie=pAMovie;

    // 如果未打开任何媒体文件，则音量调节不可用。ActiveMovie 控件的 //
    CurrentState 属性返回控件的当前状态，-1 表示未打开任何文件，此时弹出
    // 出错提示信息。

    if (m_pAMovie->GetCurrentState()!==-1)
    { DoModal(); }

    else

    { MessageBox("音频设备尚未加载，请先打开一个媒体文件。"); }

}
```

## 5 通过 MFC 建立 ON\_WM\_HSCROLL() 的消息映射并添加下列代码

// 用户拖动或点击滑块控件时，滑块控件的变化立即作用于 ActiveMovie 控件

```
void CVolumeDlg::OnHScroll(UINT nSBCode, UINT nPos,
```

```
CScrollBar* pScrollBar)
```

```
{
```

```
CSliderCtrl *pSlider=(CSliderCtrl *)pScrollBar;
```

```
int nVolume=pSlider->GetPos();
```

```
m_pAMovie->SetVolume(nVolume);
```

```
CDialog::OnHScroll(nSBCode, nPos, pScrollBar);
```

```
}
```

## 6 用 MFC 为该类添加成员函数 virtual BOOL OnInitDialog();并添加下列代码

```
BOOL CVolumeDlg::OnInitDialog()
```

```
{
```

```
CDialog::OnInitDialog();
```

// 设置音量滑块的最小值和最大值、标度和当前位置等

```
m_sld.SetRange(-10000, 0);
```

```
m_sld.SetTicFreq(1000);
```

```
m_sld.SetLineSize(200);
```

```
m_sld.SetPageSize(1000);
```

// 以 ActiveMovie 控件的当前音量作为音量滑块的当前位置

```

m_sld.SetPos(m_pAMovie->GetVolume());

return TRUE;

}

```

**7** 为“调节音量”菜单添加消息响应函数：OnAudeovolume()。

OnAudeovolume() 函数的代码如下：

```

void CTLY2412Dlg::OnAudeovolume()
{
    // TODO: Add your command handler code here

    CVolumeDlg dlgVolum;

    dlgVolum.SetVolum(&m_AMovie);
}

```

在该类的头文件里加上#include "activemovie3.h"和#include "VolumeDlg.h"

## 2. 实现“调节左右声道平衡”菜单功能

(1) 设计“调节左右声道平衡”对话框

**1** 如图 6-16 所示设计“调节左右声道平衡”对话框。

**2** 由 ClassWizard 创建的新类 CBalanceDlg 封装了该对话框。同前面的对话框

IDD\_VOLUME 相类似，CBalanceDlg 定义