

Multimedia Programming 2004

Assignments No. 3 December 22nd 2004

Due: 11.00h, January 12th 2005

Goals of the assignments:

- Learn how to load, and play sound files
- Learn how to combine existing routines with a new windows program.
- Learn the basic dialog programming and use windows messages

Preparations:

1. Download the code from the MMP2004 web-site and unzip it to a local directory
2. All further directories mentioned in the assignments can be found in this local directory

Posting Your Work: See the last page of Assignment Set 1 for the procedure for posting your work.

Assignment 1: Playing Audio Files

1. Browse to the *Tutorial1* directory and double-click the Visual C++ workspace file *tutorial1.dsw*
2. Browse through the code and read the comments to understand what is happening
3. For building the executable select <Build><Build Tutorial1.exe>
4. Execute by selecting <Build><Execute Tutorial1.exe>
5. Close your workspace by selecting <File><Close Workspace>
6. Select <All Programs><Microsoft DirectX 9.0 ...><DirectX Documentation for C++> to start the *DirectX Help dialog*.
7. Select in the *Help dialog* <Contents><DirectMusic><Direct Music C++ Tutorials><Tutorial 1: Playing Audio Files> and <Tutorials> and follow the four steps.

Assignment 2: Playing Audio Files Using a Dialog

1. Browse to the *PlayAudio* directory and double-click the workspace *PlayAudio.dsw*
2. Select the *ResourceView* tab and double-click <PlayAudio resources> <Dialog><IDD_MAIN>. This opens the *design form* for the dialog that is used in the application.
3. Right click the <Open file> button and select *Properties*. This opens the *property dialog* for this button. You will see the ID: IDC_SOUNDFILE, and the caption: *Open & file* of the button.
4. Search in the source code *PlayAudio.cpp* for the constant IDC_SOUNDFILE. You will find it in the *MainDlgProc*, the callback function for the main dialog. This function handles the messages for the dialog. If a button is pressed on the dialog a WM_COMMAND message will be sent with as parameter the ID of the button.
5. Compile and execute the program. Select some other files to play.
6. Press the Help button and read the comments.
7. Change the program such that on pressing *Open file* your own local media directory is selected as the default directory to start browsing for sound files.

Assignment 3: Compiling Your Local DirectX Workspace Copies

1. Locate the *PlayMulti* example in the DirectX sample directory and copy the project to your local *Code03* directory.
2. Open the workspace and try to compile it.
3. Now be sure to select <Build><Set Active Configuration>. In the dialog select *the Win32 Release* configuration and left-click on the <ok> button.
4. Double click on the error and change "*..\..\Common\directx.ico*" to "*..\Common\directx.ico*" Compile again. Why is this error solved? Browse to the local *Common* subdirectory and search for **.ico* files.
5. There are some new errors though. They are all related to incorrect path settings for this project. Select <Project><Settings> and select in the dialog <C++> Category: <Preprocessor>. Again change the *Additional include directories* into "*..\Common*" and recompile.
6. Select <FileView> browse to *Common* and right-click the *.cpp* files. Again change the paths and recompile. Now you should be successful in compiling without any errors and be able to execute the program.
7. Study the code. Note: do not lose yourself in the details, be sure to grasp the global picture: *Windows Programming*: dialog initialization, message passing, message handling in the callback functions; *DirectMusic*: initialization, loading, playing, and closing of the sound files.

Assignment 4: Browse the DirectMusic and DirectSound C++ Samples

1. Select <All Programs><DirectX 9.0 SDK ...><DirectX Sample Browser>.
2. Select <DirectMusic> on the top bar of the *DirectX Sample Browser* window.
3. Select *C++*, *Tutorials*, and *Samples* and try some of the examples. Look at the documentation and the source code of all the beginner projects.

Assignment 5: Image and Sound

1. Browse to your local *ImageAndSound* subdirectory and open the *BitBltDemo.dsw* workspace. This is a slightly adapted example from the previous workshop. It is a partly implemented game. Compile and execute the program. You should start the game by touching the red arrow in the lower right corner with your mouse pointer. Then you guide your mouse pointer along the black path from one colored dot till the other, until you reach the upper left corner. If you touch something outside the black route a sad face appears and a disturbing sound is played. (Do not touch the red dot in the middle; it will start the game also.)
2. The game is only implemented until the yellow dot. Finish the game such that after each dot also a nice different winning sound is played, and if the upper left corner is reached, a small smiling face appears at the end point. If you touch something outside the black route, the game should stop, a disturbing sound should be played, and a small sad face should appear at that spot.
3. Note that the *WM_LBUTTONDOWN* message is sent if you press the left mouse button. Change your program such that it starts a new game, if you press the left button of your mouse within the application window.
4. At this moment in the upper left corner the mouse position and the color information is shown. Change this information into: *'Time: xxx Record: xxx'*, where *xxx* denotes the number of seconds that has passed for the current player, and the record time in seconds until now, respectively.
5. Use your own graphics for the routes, sad face, and smile face.