Multimedia Programming 2004

Assignments No. 2

December 15th 2004

Assignments 3, 4, and 5 due: 11.00h December 22nd 2004

Goals of the assignments:

- Learn how to create, build an execute a simple window program using Visual C++
- Learn how to program using the basic constructs and simple functions of C/C++
- Learn how the basic windows message loop works for the base class CWnd
- Learn how to load, and display 24-bit bitmap images (*.bmp)
- Learn how to use transparency and play a sequence of images

Preparations:

1. Download the code from the MMP2004 web-site (www.liacs.nl/~erwin/MMP2004) 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. Only assignments 3, 4, and 5 of this set have to be posted.

Assignment 1: "Hello There" Revisited

1. Browse to the *Hello There* directory and double-click the Visual C++ workspace file *HelloThere.dsw*
2. Browse through the code and read the comments to understand what is happening
3. For building the executable select <Build><Build HelloThere.exe>
4. Execute by selecting <Build><Execute HelloThere.exe>
5. Change your code such that another text is printed.
6. Change the background color of the window.
7. Change the cursor of your window.
8. Close your workspace by selecting <File><Close Workspace>

Assignment 2: Create a HelloWorld Visual C++ Project

1. Start Microsoft Visual C++, if not already started, and Select <File><New>
2. In the dialog select <Projects><Win32 Application>
3. Type in the edit box <Project Name>: "HelloWorld" (this is the name of your project)
4. Browse to your local code directory by selecting the […] button besides the edit box <Location>
5. Select <ok>
6. In the new dialog select 'A typical "Hello World" Application'
7. Select <Finish> and <ok>. A new workspace is created.
8. Now compile, build, and execute this new code.
9. Browse your newly created project by selecting the <FileView> sheet and clicking on the <Source Files> folder, and double clicking the *HelloWorld.cpp* file.
10. Change the text, background color and the cursor. **Trick Note:** To find the different options for a cursor, find IDC_ARROW in the program; right click it and select <Go To Definition Of IDC_ARROW>. (If a rebuild is necessary, do a rebuild.) It should then open the file WINUSER.H where you find a list of standard cursor IDs of which you can select one for use in your program.
11. Also browse through the other folders and files in your project.

Assignment 3: Drawing an Image

1. Browse to the *Images BitBlt* directory and double-click the Visual C++ workspace file *BitBltDemo.dsw*
2. Browse through the code and read the comments to understand the macro structure of the program.
3. For building the executable select <Build><Build BitBltDemo.exe>
4. Execute by selecting <Build><Execute BitBltDemo.exe> A smiley should be visible.
5. Change the code such that if you press the <F1> button, the image that will be drawn changes from the *smile.bmp* into the *sad.bmp* image, and vice versa. (Note that you can find these images in the *Images* subdirectory.) (VK_F1 is the virtual key code sent, if <F1> is pressed.)
6. If you press the left button of your mouse, a message WM_LBUTTONDOWN is sent. This message will also invoke the message handling callback function *WndProc* that subsequently will process/handle it. Change the code such that if you press your left mouse button, the images changes in the same way as when you press <F1>.
7. **Trick Note:** Use <help><index> with WM_LBUTTONDOWN to find the predefined list of possible messages that you can use in future programs,

Assignment 4: Drawing and Transparency

1. Browse to the *Transparency BitBlt* directory and double-click the Visual C++ workspace file *BitBltDemo.dsw*
2. Browse through the code and read the comments to understand the macro structure of the program.
3. For building the executable select <Build><Build BitBltDemo.exe>
4. Execute by selecting <Build><Execute BitBltDemo.exe> One image should be visible.
5. Browse to the Image directory and double click the *mars.bmp*, *earth.bmp,* and *person.bmp* images to view them. These are the images that have been shown on top of each other. The black area in the *earth.bmp,* and *person.bmp* image has been made transparent by the program.
6. Change the code such that the cloths of the person become transparent (RGB color of the cloths is equal to (2, 5, 40)).
7. Reset the transparent color to black. Now change the code such that a fourth image *mars2.bmp* will be drawn on top of the existing images. (Of course with transparency.)
8. Further change the code such that:
   - If you press the <F1> button, the transparent color can be changed from black into the color of the cloths.
   - If you press the <F2> button, the *person.bmp* image can be toggled on or off.
   - If you press the <F3> button, the *mars2.bmp* image can be togged on and off.
   - If you left click your mouse somewhere in the application window, draw the earth.bmp image on that position. (In the program some remarks are made on how to get the mouse position.)

Assignment 5: Drawing a Sequence of Images

1. Browse to the *Movies BitBlt* directory and double-click the Visual C++ workspace file *BitBltDemo.dsw*
2. Browse through the code and read the comments to understand the macro structure of the program
3. For building the executable select <Build><Build BitBltDemo.exe>
4. Execute by selecting <Build><Execute BitBltDemo.exe> A small video in which a boat dissolves and a beach appears should be visible.
5. Change the code such that the movie plays faster/slower. (You do not have to submit the solution.)
6. Further change the code such that:
   - A counter will be visible that indicates which image of the sequence is visible.
   - If you press the left mouse button the sequence plays backward.
   - If you press the middle mouse button the playing of the sequence stops.
   - If you press the right mouse button the sequence plays forward.

   Note: maybe you can experiment with another sequence of background and foreground images.