**Tên đề tài: Chương 8: Vẽ pixel, Bitmaps, Fonts và Hình ảnh.**

**Danh sách các thành viên trong nhóm:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stt** | **Họ tên** | **MSSV** | **Email** | **Sdt** |
| 1 | Đỗ Đức Huy | 1511061119 | doduchuy.it.br@gmail.com | 01696260804 |
| 2 | Lê Quang Tiến | 1511061395 | Lequangtien0508@gmail.com |  |
| 3 | Võ Minh Đạt | 1511061040 | vominhdatqn@gmail.com |  |
| 4 | Võ Quốc Đạt | 1511061041 | datvo260197@gmail.com |  |

**Hướng dẫn sử dụng thư viện OpenGL với Visual Studio**

Mục đích sử dụng thư viện OpenGL với các hàm có sẵn để có thể dễ dàng vẽ các hình ảnh đồ họa trong Win32 Console application. Trước đây, trong khác phiên bản từ Visual C++ 6.0 trở về trước, cũng như các phiên bản Turbo C, Borland C để vẽ hình có thể sử dụng thư viện graphics.h tuy nhiên ở những phiên bản Visual Studio sau này, thư viện graphics.h đã bị loại bỏ và cũng không còn được hỗ trợ. Bởi vì bên cạnh thư viện graphics.h đã lỗi thời, nổi lên các thư viện đồ hoạ tốt hơn và cũng dễ dàng sử dụng hơn, điển hình trong đó là thư viện OpenGL.

**Bài hướng dẫn này sẽ hướng dẫn cách cài đặt thư viện OpenGL, tạo project trong Visual Studio C++ sử dụng thư viện OpenGL và một ví dụ cơ bản về vẽ hình bằng các hàm trong thư viện OpenGL.**

**I. Cài đặt Visual Studio Nếu chưa cài đặt Visual Studio trên máy tính chúng ta thực hiện 2 bước sau**

1. Tải tập tin cài đặt Visual Studio C++ từ <http://www.microsoft.com/express/Downloads/>
2. Thực thi tập tin cài đặt và làm theo hướng dẫn để cài đặt Visual Studio C++.

**II. Cài đặt thư viện GLUT**

1. Tải tập tin "glut-3.7.6-bin-32and64.zip" từ

<http://www.mediafire.com/?12ncmty5g6mgr7t>

1. Giải nén tập tin glut-3.7.6-bin-32and64.zip ra thư mục glut-3.7.6-bin
2. Trong thư mục **glut-3.7.6-bin:**

* Với phiên bản Windows 32bit (viết ứng dụng 32 bit)
* Sao chép tập tin **glut.h** và **glut.def** vào thư mục **C:\Program Files\Microsoft SDKs\Windows\v7.0A\Include\gl**
* Sao chép tập tin **glut32.lib** vào thư mục **C:\Program Files\Microsoft SDKs\Windows\v7.0A\Lib**
* Sao chép tập tin **glut32.dll** vào thư mục **C:\windows\system32\**
* Với phiên bản **Windows 64bit** (viết ứng dụng 32 bit và 64 bit)
* Sao chép tập tin **glut.h** và **glut.def** vào thư mục **C:\Program Files(x86)\Microsoft SDKs\Windows\v7.0A \Include\gl**
* Sao chép **glut32.lib** và **glut64.lib** vào thư mục **C:\Program Files(x86)\Microsoft SDKs\Windows\v7.0A\Lib**
* Sao chép **glut32.lib** và **glut64.lib** vào thư mục **C:\Program Files(x86)\Microsoft SDKs\Windows\v7.0A\Lib\x64**
* Sao chép tập tin **glut32.dll** và **glut64.dll** vào thư mục C:\windows\SysWOW64
* Sao chép tập tin **glut32.dll** và **glut64.dll** vào thư mục **C:\windows\system32**

**III. Cài đặt GLEW (Nên cài đặt)**

1. Tải tập tin "Windows 32-bit and 64-bit Binaries" từ <http://glew.sourceforge.net/>
2. Giải nén tập tin glew-1.10.0-win32.zip vào thư mục có tên glew

* Với phiên bản Windows 32bit
* Sao chép tất cả tập tin trong thư mục **include/GL** vào thư mục **C:\Program Files\Microsoft SDKs\Windows\v7.0A\Include\gl**
* Sao chép tập tin **glew32.lib** trong thư mục **lib/Release/Win32/** vào thư mục **C:\Program Files\Microsoft SDKs\Windows\v7.0A\Lib**
* Sao chép tập tin **glew32.dll** trong thư mục **bin/Release/Win32/** vào thư mục **C:\windows\system32**
* Với phiên bản Windows 64bit
* Sao chép tất cả tập tin trong thư mục **include/GL** vào thư mục **C:\Program Files(x86)\Microsoft SDKs\Windows\v7.0A \Include\gl**
* Sao chép tập tin **glew32.lib** trong thư mục **lib/Release/x64/** vào thư mục **C:\Program Files(x86)\Microsoft SDKs\Windows \v7.0A\Lib**
* Sao chép tập tin **glew32.dll** trong thư **mục bin/Release/x64/** vào thư mục **C:\windows\SysWOW64**

**IV. Tạo Visual Studio C++ project với OpenGL**

1. Mở chương trình **Visual Studio C++.**
2. Nhấn vào mục **Create new project** trong Visual Studio
3. Lựa chọn **Win32 Console Application**, nhập tên của project và nhấn nút OK
4. Trong cửa sổ wizard hiện ra sau đó, nhấn vào mục Application Settings và chọn mục **Empty Project**, sau đó nhấn nút Finish.
5. Lựa chọn **Add New Item** từ Project menu, sau đó chọn loại **tập tin cpp**, nhập tên tập tin và nhấn OK.
6. Nội dung trong tập tin mã nguồn vừa thêm như **phần V dưới đây**
7. Nhấn chuột phải vào Project và chọn **Properties**
8. Ở cửa sổ vừa hiện ra, ở trên cùng, đổi Configuration thành **All Configuraitons.**
9. Ở phía bên cột bên trái, chọn **Linker -> Input**
10. Khi đó ở phía bên phải sẽ hiện ra danh mục, trong đó có mục **Addititonal Dependencies**, thêm ";glut32.lib" vào sau mục này, nếu có sử dụng glew, cũng thêm cả ";glew32.lib" vào sau đó.
11. Nhấn nút OK và chọn **Build Solution** từ Build menu.
12. Thực thi chương trình bằng cách nhấn tổ hợp phím **Ctrl+F5.**
13. Nếu cài đặt đúng chính xác tất cả các bước, chương trình khi thực thi sẽ có kết quả như sau

**V. Nội dung mã nguồn ví dụ**

1. **Example 8-1    Drawing a Bitmapped Character: drawf.c**

#include <GL/glew.h>

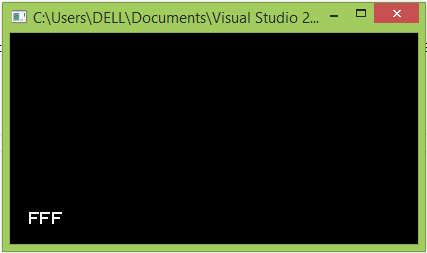
#include <GL/glut.h>

#include <stdlib.h>

GLubyte rasters[24] = {

0xc0, 0x00, 0xc0, 0x00, 0xc0, 0x00, 0xc0, 0x00, 0xc0, 0x00,0xff, 0x00, 0xff, 0x00, 0xc0, 0x00, 0xc0, 0x00, 0xc0, 0x00,

0xff, 0xc0, 0xff, 0xc0};



void init(void)

{

//giá trị tỷ lệ, độ lệch, thay đổi trong vẽ pixel

glPixelStorei(GL\_UNPACK\_ALIGNMENT, 1);

//xóa màu trắng

glClearColor(0.0, 0.0, 0.0, 0.0);

}

void display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0, 1.0, 1.0);

//Chỉ định tọa độ x,y cho vị trí raster hiện tại.

glRasterPos2i(20, 20);

//xác định chiều rộng và chiều cao của hình ảnh bitmap chỉ định vị trí các giá trị x và y được thêm vào vị trí raster hiện tại sau khi bitmap được vẽ.

glBitmap(10, 12, 0.0, 0.0, 11.0, 0.0, rasters);

glBitmap(10, 12, 0.0, 0.0, 11.0, 0.0, rasters);

glBitmap(10, 12, 0.0, 0.0, 11.0, 0.0, rasters);

glFlush();

}

void reshape(int w, int h)

{

glViewport(0, 0, (GLsizei)w, (GLsizei)h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0, w, 0, h, -1.0, 1.0);

glMatrixMode(GL\_MODELVIEW);

}

void keyboard(unsigned char key, int x, int y)

{

switch (key) {

case 27:

exit(0);

}

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(100, 100);

glutInitWindowPosition(100, 100);

glutCreateWindow(argv[0]);

init();

glutReshapeFunc(reshape);

glutKeyboardFunc(keyboard);

glutDisplayFunc(display);

glutMainLoop();

return 0;

}

1. **Example 8-2 Drawing a Complete Font: font.c**

#include<GL\glew.h>

#include <GL\GL.h>

#include <GL\GLU.h>

#include <GL\glut.h>

#include <stdlib.h>

#include <string.h>

//Khai báo mảng khoảng trắng

GLubyte space[] =

{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };

//Khai báo mảng các chữ cái

GLubyte letters[][13] = {

{ 0x00, 0x00, 0xc3, 0xc3, 0xc3, 0xc3, 0xff, 0xc3, 0xc3, 0xc3, 0x66, 0x3c, 0x18 },

{ 0x00, 0x00, 0xfe, 0xc7, 0xc3, 0xc3, 0xc7, 0xfe, 0xc7, 0xc3, 0xc3, 0xc7, 0xfe },

{ 0x00, 0x00, 0x7e, 0xe7, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xe7, 0x7e },

{ 0x00, 0x00, 0xfc, 0xce, 0xc7, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3, 0xc7, 0xce, 0xfc },

{ 0x00, 0x00, 0xff, 0xc0, 0xc0, 0xc0, 0xc0, 0xfc, 0xc0, 0xc0, 0xc0, 0xc0, 0xff },

{ 0x00, 0x00, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xfc, 0xc0, 0xc0, 0xc0, 0xff },

{ 0x00, 0x00, 0x7e, 0xe7, 0xc3, 0xc3, 0xcf, 0xc0, 0xc0, 0xc0, 0xc0, 0xe7, 0x7e },

{ 0x00, 0x00, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3, 0xff, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3 },

{ 0x00, 0x00, 0x7e, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18, 0x7e },

{ 0x00, 0x00, 0x7c, 0xee, 0xc6, 0x06, 0x06, 0x06, 0x06, 0x06, 0x06, 0x06, 0x06 },

{ 0x00, 0x00, 0xc3, 0xc6, 0xcc, 0xd8, 0xf0, 0xe0, 0xf0, 0xd8, 0xcc, 0xc6, 0xc3 },

{ 0x00, 0x00, 0xff, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0 },

{ 0x00, 0x00, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3, 0xdb, 0xff, 0xff, 0xe7, 0xc3 },

{ 0x00, 0x00, 0xc7, 0xc7, 0xcf, 0xcf, 0xdf, 0xdb, 0xfb, 0xf3, 0xf3, 0xe3, 0xe3 },

{ 0x00, 0x00, 0x7e, 0xe7, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3, 0xe7, 0x7e },

{ 0x00, 0x00, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xfe, 0xc7, 0xc3, 0xc3, 0xc7, 0xfe },

{ 0x00, 0x00, 0x3f, 0x6e, 0xdf, 0xdb, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3, 0x66, 0x3c },

{ 0x00, 0x00, 0xc3, 0xc6, 0xcc, 0xd8, 0xf0, 0xfe, 0xc7, 0xc3, 0xc3, 0xc7, 0xfe },

{ 0x00, 0x00, 0x7e, 0xe7, 0x03, 0x03, 0x07, 0x7e, 0xe0, 0xc0, 0xc0, 0xe7, 0x7e },

{ 0x00, 0x00, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18, 0xff },

{ 0x00, 0x00, 0x7e, 0xe7, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3 },

{ 0x00, 0x00, 0x18, 0x3c, 0x3c, 0x66, 0x66, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3 },

{ 0x00, 0x00, 0xc3, 0xe7, 0xff, 0xff, 0xdb, 0xdb, 0xc3, 0xc3, 0xc3, 0xc3, 0xc3 },

{ 0x00, 0x00, 0xc3, 0x66, 0x66, 0x3c, 0x3c, 0x18, 0x3c, 0x3c, 0x66, 0x66, 0xc3 },

{ 0x00, 0x00, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18, 0x3c, 0x3c, 0x66, 0x66, 0xc3 },

{ 0x00, 0x00, 0xff, 0xc0, 0xc0, 0x60, 0x30, 0x7e, 0x0c, 0x06, 0x03, 0x03, 0xff }

};

GLuint fontOffset;

//Tạo font raster

void makeRasterFont(void)

{

GLuint i, j;

glPixelStorei(GL\_UNPACK\_ALIGNMENT, 1);

fontOffset = glGenLists(128);

for (i = 0, j = 'A'; i < 26; i++, j++) {

// Danh sách hiển thị được tạo bằng glNewList gồm tên hiển thị và Chỉ định chế độ biên dịch, có thể là GL\_COMPILE dùng để lưu trữ các nhóm lệnh GL

glNewList(fontOffset + j, GL\_COMPILE);

//xác định chiều rộng và chiều cao của hình ảnh bitmap chỉ định vị trí các giá trị x và y được thêm vào vị trí raster hiện tại sau khi bitmap được vẽ.

glBitmap(8, 13, 0.0, 2.0, 10.0, 0.0, letters[i]);

glEndList();

}

glNewList(fontOffset + ' ', GL\_COMPILE);

glBitmap(8, 13, 0.0, 2.0, 10.0, 0.0, space);

glEndList();

}

void init(void)

{

glShadeModel(GL\_FLAT);

makeRasterFont();

}

//In chuỗi

void printString(char \*s)

{

glPushAttrib(GL\_LIST\_BIT);

glListBase(fontOffset);

glCallLists(strlen(s), GL\_UNSIGNED\_BYTE, (GLubyte \*)s);

glPopAttrib();

}

//\*Mọi thứ ở trên dòng này có thể nằm trong thư viện

//\* xác định phông chữ.Để làm cho nó hoạt động, bạn đã có

//\* để gọi makeRasterFont() trước khi bạn bắt đầu thực hiện

//\* các cuộc gọi tới printString().

void display(void)

{

GLfloat white[3] = { 1.0, 1.0, 1.0 };

GLfloat blue[3] = { 0.0, 0.0, 1.0 };

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3fv(blue);

//Chỉ định tọa độ x,y cho vị trí raster hiện tại.

glRasterPos2i(20, 60);

printString("XIN CHAO CAC BAN");

glRasterPos2i(20, 40);

printString("MINH LA HUY");

glFlush();

}

void reshape(int w, int h)

{

glViewport(0, 0, (GLsizei)w, (GLsizei)h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0.0, w, 0.0, h, -1.0, 1.0);

glMatrixMode(GL\_MODELVIEW);

}

void keyboard(unsigned char key, int x, int y)

{

switch (key) {

case 27:

exit(0);

}

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutInitWindowPosition(100, 100);

glutCreateWindow(argv[0]);

init();

glutReshapeFunc(reshape);

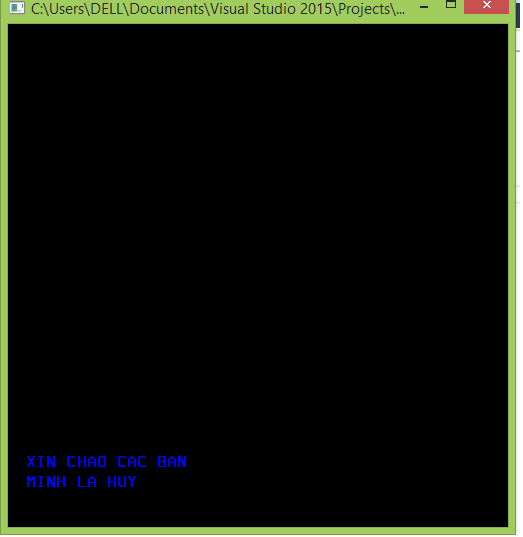
glutKeyboardFunc(keyboard);

glutDisplayFunc(display);

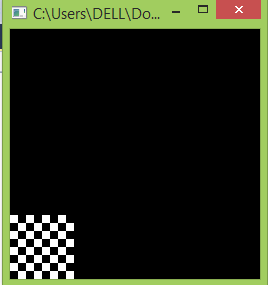
glutMainLoop();

return 0;

}



1. **Example 8-3 Drawing a Complete Images: images.c**



#include<GL\glew.h>

#include <GL/gl.h>

#include <GL/glu.h>

#include <GL/glut.h>

#include <stdlib.h>

#include <stdio.h>

#define checkImageWidth 64

#define checkImageHeight 64

GLubyte checkImage[checkImageHeight][checkImageWidth][3];

static GLdouble zoomFactor = 1.0;

static GLint height;

void makeCheckImage(void)

{

int i, j, c;

for (i = 0; i < checkImageHeight; i++) {

for (j = 0; j < checkImageWidth; j++) {

c = ((((i & 0x8) == 0) ^ ((j & 0x8)) == 0)) \* 255;

checkImage[i][j][0] = (GLubyte)c;

checkImage[i][j][1] = (GLubyte)c;

checkImage[i][j][2] = (GLubyte)c;

}

}

}

void init(void)

{

glClearColor(0.0, 0.0, 0.0, 0.0);

glShadeModel(GL\_FLAT);

makeCheckImage();

glPixelStorei(GL\_UNPACK\_ALIGNMENT, 1);

}

void display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

//Chỉ định tọa độ x,y cho vị trí raster hiện tại.

glRasterPos2i(0, 0);

glDrawPixels(checkImageWidth, checkImageHeight, GL\_RGB,

GL\_UNSIGNED\_BYTE, checkImage);

glFlush();

}

void reshape(int w, int h)

{

glViewport(0, 0, (GLsizei)w, (GLsizei)h);

height = (GLint)h;

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0, (GLdouble)w, 0.0, (GLdouble)h);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

void motion(int x, int y)

{

static GLint screeny;

screeny = height - (GLint)y;

glRasterPos2i(x, screeny);

//phóng to hoặc thu nhỏ hình chữ nhật pixel trước khi được vẽ

glPixelZoom(zoomFactor, zoomFactor);

//Sao chép dữ liệu pixel trong bộ đệm khung

glCopyPixels(0, 0, checkImageWidth, checkImageHeight,

GL\_COLOR);

glPixelZoom(1.0, 1.0);

glFlush();

}

void keyboard(unsigned char key, int x, int y)

{

switch (key) {

case 'r':

case 'R':

zoomFactor = 1.0;

glutPostRedisplay();

printf("zoomFactor reset to 1.0\n");

break;

case 'z':

zoomFactor += 0.5;

if (zoomFactor >= 3.0)

zoomFactor = 3.0;

printf("zoomFactor is now %4.1f\n", zoomFactor);

break;

case 'Z':

zoomFactor -= 0.5;

if (zoomFactor <= 0.5)

zoomFactor = 0.5;

printf("zoomFactor is now %4.1f\n", zoomFactor);

break;

case 27:

exit(0);

break;

default:

break;

}

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(250, 250);

glutInitWindowPosition(100, 100);

glutCreateWindow(argv[0]);

init();

glutDisplayFunc(display);

glutReshapeFunc(reshape);

glutKeyboardFunc(keyboard);

glutMotionFunc(motion);

glutMainLoop();

return 0;

}

1. **Example 8-4 Drawing a Complete Drawf: drawf.c**



#include <GL/glew.h>

#include <GL/glut.h>

#include <stdlib.h>

GLubyte rasters[24] = {

0xc0, 0x00, 0xc0, 0x00, 0xc0, 0x00, 0xc0, 0x00, 0xc0, 0x00,0xff, 0x00, 0xff, 0x00, 0xc0, 0x00, 0xc0, 0x00, 0xc0, 0x00,

0xff, 0xc0, 0xff, 0xc0 };

void init(void)

{

glPixelStorei(GL\_UNPACK\_ALIGNMENT, 1);

glClearColor(0.0, 0.0, 0.0, 0.0);

}

void display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0, 0.0, 1.0);

//Chỉ định tọa độ x,y cho vị trí raster hiện tại.

glRasterPos2i(100, 100);  
//xác định chiều rộng và chiều cao của hình ảnh bitmap chỉ định vị trí các giá trị x và y được thêm vào vị trí raster hiện tại sau khi bitmap được vẽ.

glBitmap(10, 12, 0.0, 0.0, 30.0, 0.0, rasters);

glBitmap(10, 12, 10.0, 0.0, 11.0, 0.0, rasters);

glBitmap(10, 12, 0.0, 0.0, 11.0, 0.0, rasters);

glBitmap(10, 12, 0.0, 0.0, 11.0, 0.0, rasters);

glFlush();

}

void reshape(int w, int h)

{

glViewport(0, 0, (GLsizei)w, (GLsizei)h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0, w, 0, h, -1.0, 1.0);

glMatrixMode(GL\_MODELVIEW);

}

void keyboard(unsigned char key, int x, int y)

{

switch (key) {

case 27:

exit(0);

}

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutInitWindowPosition(100, 100);

glutCreateWindow(argv[0]);

init();

glutReshapeFunc(reshape);

glutKeyboardFunc(keyboard);

glutDisplayFunc(display);glutMainLoop();return 0;}