JMU - Department of Computer Science

Installing GLUT for MinGW

1 Introduction

OpenGL is independent of any windowing system. As a result, it contains no functions for opening windows or interacting with the user. Each windowing systems that supports OpenGL has its own library that procides the support.

GLUT, the OpenGL Utility Toolkit, is a simple windowing system that has been ported to several different operating systems. It is commonly used when teaching OpenGL.

This document describes the installation of GLUT for developing on MS-Windows using MinGW. It assumes that you have already installed MinGW.

2 Getting Started

The first thing you need to do is download the binaries for the Win32 port of GLUT.

GLUT 3.7.6

3 Putting Files in the Right Place

You need to copy three files from the .ZIP file.

1. Copy `glut.h` to the MinGW\include\GL directory.
2. Copy `glut32.lib` to your build directory (i.e., the directory that you compile into and link from).
3. Copy `glut32.dll` to the same directory where your executable will be created.

   (You can actually put `glut32.dll` in any directory in your path.)

4 Building an Executable

You need to be aware of the following:

- You must `#include <windows.h>` before you `#include <"GL/glut.h">`
- When you link, you must explicitly link-in `glut32.lib` (and not use the `-lglut32` option).
- You may get some warnings like the following:

  ```
  ignoring #pragma comment
  ```

  ```
  warning: 'int glutCreateMenu_ATEXIT_HACK(void (*)(int))' defined but not used
  ```

  You can ignore them.

Help resolving other problems is available at the MinGWiki.

5 Testing Your Installation

You can use the following small program (named `test.c`) to test your installation:

```
#include <windows.h>
#include "GL/glut.h"



void display()
{
```

```
  glClear(GL_COLOR_BUFFER_BIT);

  glBegin(GL_POLYGON);
  glVertex2f(-0.5, -0.5);
  glVertex2f(-0.5,  0.5);
  glVertex2f(0.5,  0.5);
  glVertex2f(0.5, -0.5);
  glEnd();

  glFlush();
}

void init()
{
  glClearColor(0.000, 0.110, 0.392, 0.0); // JMU Gold

  glColor3f(0.314, 0.314, 0.000); // JMU Purple

  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  gluOrtho2D(-1.0, 1.0, -1.0, 1.0);
}

int main(int argc, char** argv)
{
  glutInit(&argc, argv);
  glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
  glutInitWindowSize(640, 480);
  glutInitWindowPosition(0, 0);
  glutCreateWindow("Test");
  glutDisplayFunc(display);
  init();
  glutMainLoop();
}
```

You should be able to build this program from the command line as follows:

```
g++ -o test -Wall test.c -mwindows glut32.lib -lopengl32 -lglu32
```

6 Using jGrasp

Some of you may have used MinGW under jGrasp in the past. You can do so with GLUT but you will have to change some settings in jGrasp.

In jGrasp, choose Settings on the main menu, pull down to Compiler Settings, and then pull down to either Workspace or Project. Next, click on the Compiler tab and change the "Language" to C.

From here, you can set the "FLAGS or ARGS" for "Make", the "Compiler", etc... See, for example, the flags and arguments used in the example above.

Department of Computer Science