

OpenGL on Linux

Introduction

OpenGL on Linux is a bit of an adventure. In the past it has not been well supported and support depends upon the GPU that you have. The situation is getting much better and on standard distributions, such as Ubuntu, it shouldn't be much of a problem. I've never been able to completely test these instructions since most of my Linux computers are servers and have very old graphics hardware.

Installing Libraries

Most Linux distributions have pre-built libraries, but they may not be installed by default. You will need at least the following libraries:

```
libgl1-mesa-dev  
libglu1-mesa-dev  
libglew-dev  
libglfw3-dev  
libglm-dev
```

On Ubuntu they can all be installed with apt-get. You should also install mesa-utils. After this you can run the glxinfo program to see if everything has been installed correctly.

Building Programs

As far as I can tell all the libraries are installed in the standard locations. There may be some additional locations depending upon your GPU. You can use pkg-config to check. It doesn't appear that pkg-config knows about glfw, so you may not be able to use it in build. Something like the following should build an OpenGL program:

```
cc program.c -lglfw -lGLEW -lGLU -lGL
```

Issues

OpenGL on Linux doesn't always do what you expect it to. In particular it might not find the appropriate driver and version. There are two things that you can do to help diagnose this problem. The first is to call `glGetString(GL_VERSION)`, which returns a string which is the highest version of OpenGL supported by your current software configuration. At the very minimum it should be 3.2, but 4.0 or higher is better. The second call is `glGetString(GL_VENDOR)` this will tell you if you are actually using your GPU. If it says you are using a software renderer you are not using your GPU. In this case you should look at the GPU vendor website to see if they provide a Linux driver for your GPU. Some versions of Linux provide drivers for some GPUs, you may need to investigate this.

When I tried this on my Nvidia TX1 I got the following output:

```
Version: 4.5.0 NVIDIA 24.2.1  
Vendor: NVIDIA Corporation
```

In this case the vendor is clearly Nvidia, so the appropriate driver is being used. Also note that the OpenGL version is greater than 4.0, so everything should work okay.

One more thing, remove the include of Windows.h and replace it with includes for unistd.h and stdlib.h. There are also a few places where capitalization will need to be changed.