

Design of a 3D Graphics Engine

TJHSST Senior Research Project Proposal

Computer Systems Lab 2009-2010

Joseph Hallahan, period 5

October 28, 2009

Abstract

As computer power steadily increases, the search for newer and better programming techniques increases as well. One technique is to use a graphics engine in a program in order to make certain processes, such as, in this case, the drawing or rendering of graphics, easier to do. This project involves the creation of a graphics engine for use in rendering both 2D and 3D graphics as well as incorporating some minor physics and keyboard and mouse input. C++ and OpenGL will be used to create this program, which should make the creation of other 3D programs more easy and efficient.

keywords: graphics engine, OpenGL

1 Introduction

1.1 Situation/Statement of Problem

There are a lot of programming engines. They make programmers' lives easier because of their ability to do many complicated processes automatically. Graphics engines focus more on the rendering of 2D and especially 3D graphics, since those processes can be incredibly difficult. Basically what happens is the engine does this work automatically so the programmer doesn't have to.

1.2 Purpose

The goal of this project will be to create a graphics engine for use in the use of computer games and other 3D applications. This engine will need to be able to display 2D and 3D graphics and get input from the keyboard and mouse, at minimum. However, it will also need to be able to be used in a variety of

applications, unlike most other engines that are only suited for one type of project.

2 Background

A lot of work has been done in this area in recent years. One obvious use for graphics engines is the creation of video games, and the Unreal engine and the Torque Game engine are good examples of these. One common idea to take into account when using a graphics engine is its efficiency, and engine creators are continuously trying to find ways to make them do more work in less time. This idea that faster is better is prevalent in many aspects of the computing industry, but graphics, 3D rendering in particular, is both hardware and software-intensive. There are several different algorithms for displaying 3D graphics, and the speed at which they are drawn can be a big issue. One other idea to know with a graphics engine is its versatility. Some engines can only work on graphics, while the two I mentioned above, Unreal and Torque, can also accomplish other tasks such as playing sounds, connecting to and hosting servers over the Internet or a local area network, and many others. Although the emphasis on topics cross-platform and parallel programming have enabled engines to take on still more tasks, most still focus on one in particular, with both Unreal and Torque being primarily focused on a certain genre of video game. While they can, with a number of modifications, be used in other types of applications, there have been few engines that are designed to work with several dif-

ferent programs without needing to change anything.

3 Development

3.1 Introduction to OpenGL and 2D Graphics

When I began my project I had no previous experience with OpenGL. I had programmed in C++ before several times and had used a few external graphics libraries, including Irrlicht and Allegro. However, OpenGL is much more powerful than these and, as a result, is much more complicated. My initial problems with compiling my code were solved through the use of a program which compiled the code and found OpenGL for it, since otherwise it would have generated errors. Because of my lack of experience, my first program was simply copied from a test program from the TJ web site. I began to make small modifications to the code to get a better feel for it, such as changing colors and rendering text.

3.2 3D Graphics

In order to be able to show any necessary graphics, my engine will need to encompass both 2D and 3D support. OpenGL made working in 3D significantly more easy than I thought, but it still did take several example problems and a lot of debugging to end up with a program that could render a sphere. The sphere-inside-a-wireframe-box was taken directly from the same web page my 2D skills

were tried out on, but, as before, I continued to make adjustments to get to know all of the functions better. Also, I implemented a function that reads keyboard input and moves the object around accordingly. This code was completely original. I also created a mouse function that is not particularly useful at this point, but should mouse support ever be needed, it will be easy to implement. My programs at this point are only to be used as examples that incorporate all of the code I will need to have in my engine, since its point will be to prevent having to code too much in its derivative programs.

4 Tests and Analysis

My original 2D program could only be tested by running it and seeing if what was drawn, i.e. a line, was what was intended. As I began adding features such as 3D rendering and text support, my tests grew to include seeing if my object was drawn correctly and ensuring the key inputs did only what they needed to do. Because I had the program move the sphere around based on this keyboard input, I had to make sure it was moved in the right directions and didn't continue moving after the keys were released. All of this testing so far is to make sure that my code is working, so that my engine will be able to do this automatically. Once it is complete, my tests will consist of several different programs that will be created using the engine.

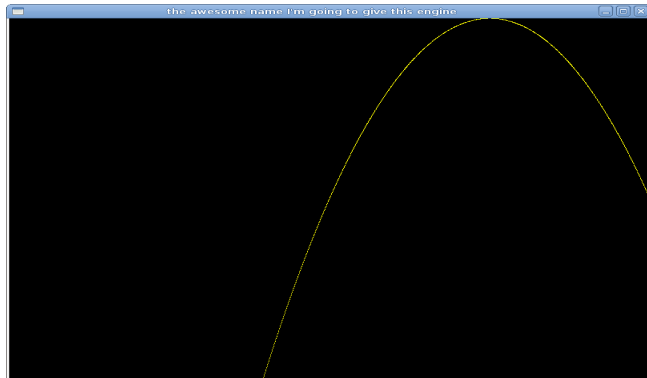


Figure 1: This is an introductory OpenGL program.

5 Results

As of right now, the program works correctly. It successfully renders a sphere and a wire-frame cube, and they can be moved in the right directions using the keyboard. Probably the only significant problem with it right now is a slight flickering when moved, most likely just a result of my rendering method. I will need to work to improve my code in order to get rid of the flickering.

6 Conclusion

I have been programming for many years, and one of the most difficult parts of working on a project is setting the program up. Actually coding logic and instructions for the computer is fairly easy, but explaining those instructions can be a lot of work. If my engine works correctly, the creation of 2D and 3D applications should become much easier. Creating 3D programs from scratch

especially is a difficult endeavor, and hopefully this engine will take at least some of the workload off of the creator. I have said "application" throughout this paper instead of "game," although many engines are used to make games. However, programs such as Paint or Blender also require the use of graphics programming, and my goal is to make my engine as versatile as possible, so that these types of programs will also be able to be created with as much ease as just a game.

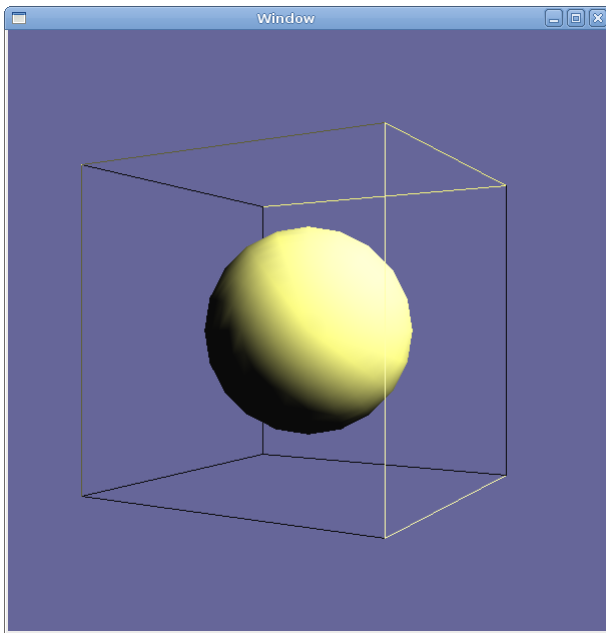


Figure 2: This is a sphere inside a wireframe cube. It can be moved with the keyboard.

7 Bibliography

- Development of a 3D Graphics Engine (Kassing)
- Modular Architecture for Computer Game Design (McNeill)
- Multi-threaded Game Engine (Tulip, Bekkema, Nesbitt)
- Interactive 3D Geometry in OpenGL (Welsh)
- FROG: The Fast & Realistic OPENGL Displayer
- Some code used from "<http://www.tjhsst.edu/dhyatt/superap/opengl.html>"