

Design of a 3D Graphics Engine for use in Various Applications

Joseph Hallahan

Computer Systems Lab 2009-2010

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.

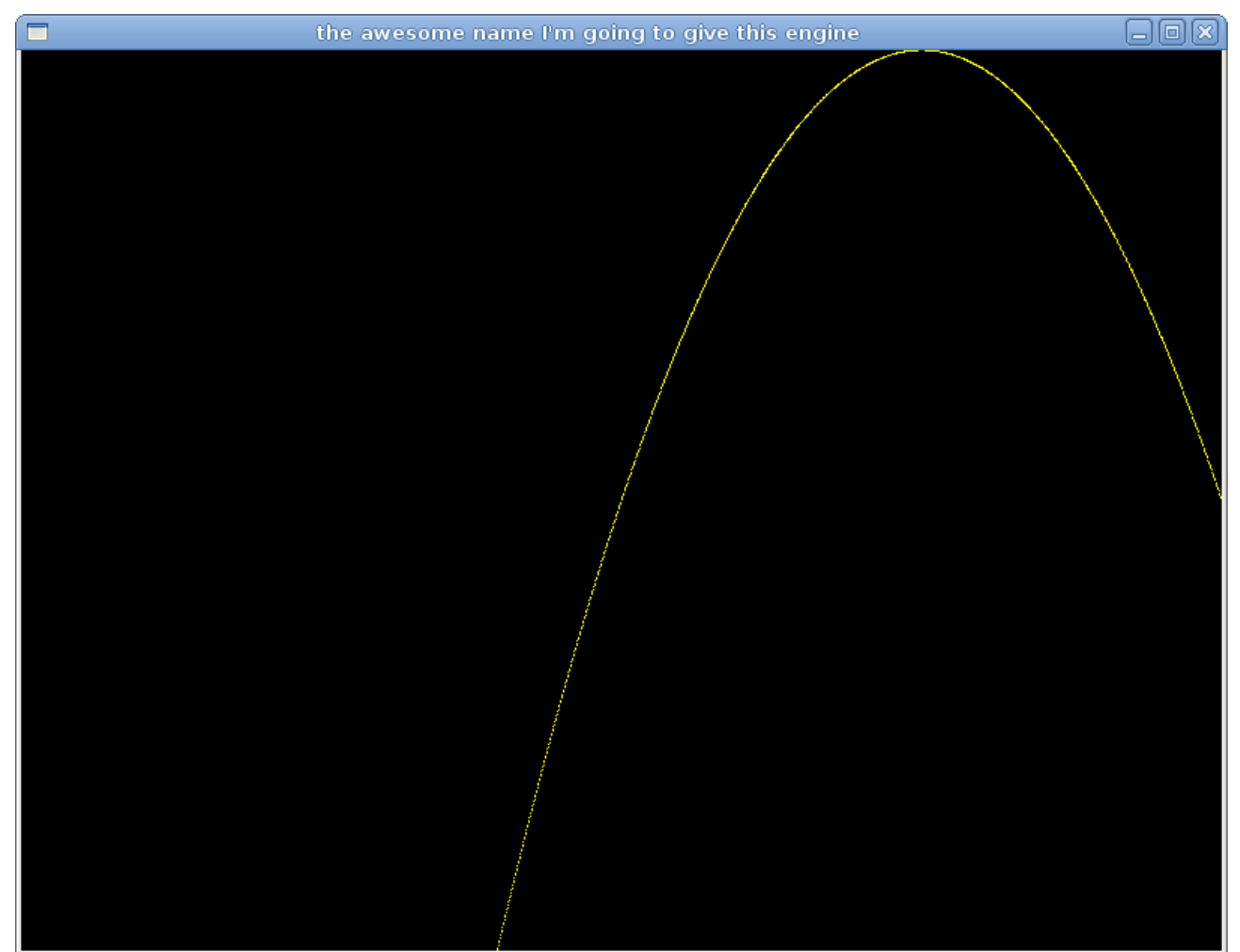


Fig 1: An introductory OpenGL program

Background and Introduction

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 different programs without needing to change anything.

Discussion

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 was 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.

Results and Conclusions

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..

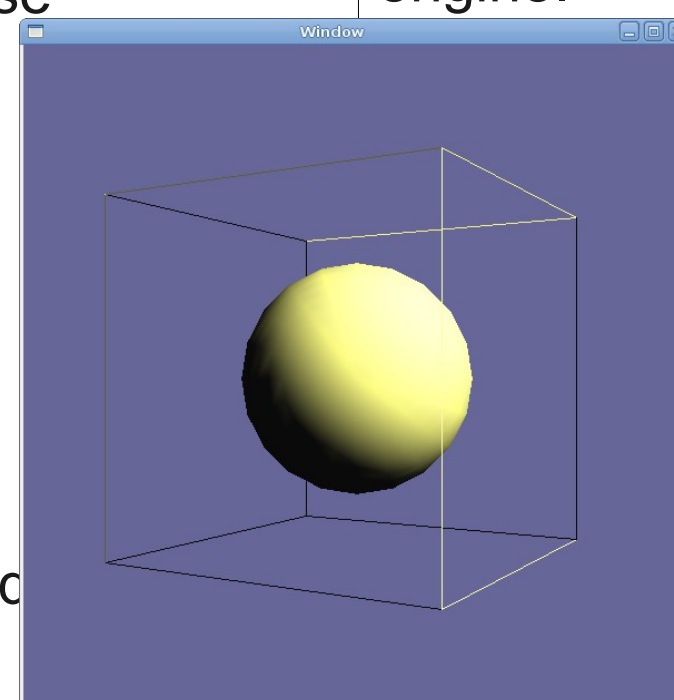


Fig 2: This is a sphere inside of a cube. They can be moved using the keyboard.