

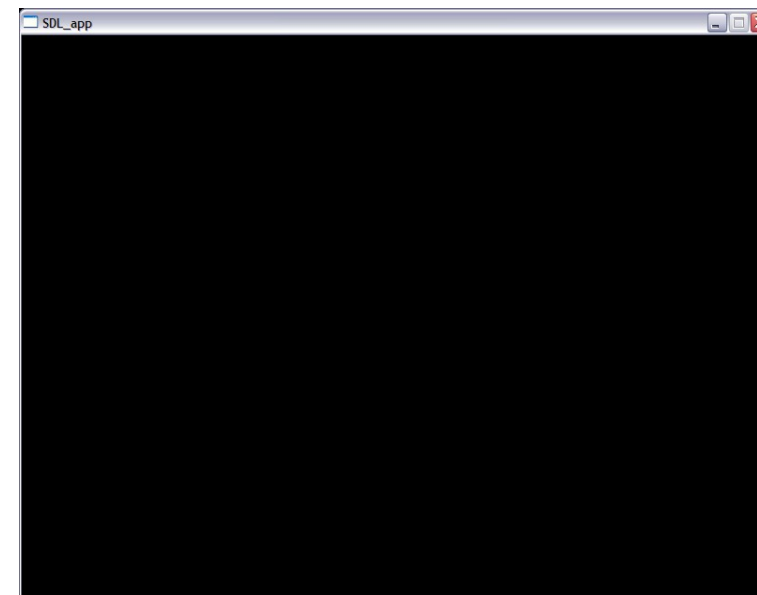
# Design of a 3D Graphics Engine for use in Various Applications

Joseph Hallahan

Computer Systems Lab 2009-2010

## Abstract

Physics play a large role in many modern programs. Simulations of every kind and numerous video games use physics equations and functions in order to generate as realistic scenario as possible. The uses of a physics engine, a framework of various pieces of code containing these physics functions, are many. A physics engine that works along with a graphics engine - one that could take on both duties simultaneously, would be very useful indeed. This project involves creating a fully runnable graphics engine that also supports physics relating to collision detection. In addition to displaying and performing physics on 2D images, it will also keep track of keyboard input.



A window created with SDL

## Background and Introduction

The Engine is often the most important part of a program. It contains all the functionality needed for whatever purposes the program may need. Some are used for graphics, some for physics, and some for both. One feature common to all of them is that they can be used for many different projects, rather than just the one they are originally made for. As a result, they must either be able to have all this functionality without needing to rely on any external information, or use such generic external information that any similar program would be able to provide it as well.

The goal of this project is to create a working engine that focuses both on two-dimensional graphics and physics. It should be able to be useful for various projects, including possibly some agent based modeling, gaming, and simulations. It will need to be able to be used as a standalone program to ensure that it is not tied specifically to the programs I test it with. It will need to do the math related to the drawing and movement of shapes, and also relating to collision handling. The engine will also need to be able to get input from the user, particularly from the keyboard.

## Discussion

This project is actually an engine, as opposed to my earlier efforts: a single one-time-use file and a header file used as a library. My code not only creates all the resources needed for a program to run, but also runs it and stops when it is finished. My biggest changes to this iteration of my project were adding this extra functionality, adding SDL, and beginning to add some physics methods. Next quarter I hope to be able to improve upon my collision and physics code, and come up with a program that is able to test all of that. My "Pong" program will go by the wayside in favor of a new test program that can illustrate the full functionality of the engine.

## Results and Conclusions

I would have preferred to put more functionality in the engine this quarter, but I am at least pleased that I got my structure issues from last quarter resolved. Apparently my second quarter OpenGL code was flawed in some way, because my drawing code didn't work at first. This may have just been due to a missing line or two, since it didn't generate any errors or anything. The switch from separate files to a Code::Blocks (my IDE) project permitted the files all to be compiled and linked correctly on my Windows system, but unfortunately I do not have the Linux skills necessary to do the same thing from a command line. Luckily Code::Blocks is apparently open-source and therefore compatible with Linux, so by next quarter I hope to be able to install it as part of my personal quota on the school computers.