Open GL Framework For A Computer Graphics Course

Programmer: Daniel Odle
Sponsor / Advisor: Dr. Morse
University of Evansville
4-26-03
# Table of Contents

Introduction ............................................. 3  
Statement of Problem ...................................... 3  
Design Approach .......................................... 3  
  Window Classes ....................................... 4  
  Main Helper Classes ................................ 4  
  Sample Projects .................................... 5  
    Etch-a-Sketch Clone ............................... 5  
    Hangman ........................................... 6  
    Puzzle Program .................................. 7  
    Board Game ....................................... 8  
Results .................................................. 9  
Possible Future Additions ............................. 10  
Conclusion ............................................ 10  

# List of Figures

Figure 1 ................................................ 6  
Figure 2 ................................................ 7  
Figure 3 ................................................ 8  
Figure 4 ................................................ 9  
Introduction
For my senior project I have written an OpenGL framework that can be used by future computer graphics courses at the University of Evansville. A framework is a collection of code that sets up the foundation of a project for the programmer to build off of; much like a framework of a house sets up the foundation of the house for a carpenter to build off of. It takes care of all the initialization of setting up a main window for a project and allows the programmer to just add the extra functionality to it. The main purpose of the project was to fix and build from the code and ideas of the textbook used in the computer graphics class as well as give the future students a more object-oriented framework to work with.

Statement of Problem
There were a few goals for this project. One was to get a working framework for future students, and anyone else who wishes to use it. Another was to encapsulate the ideas of the book into a more object-oriented structure. The textbook used by the computer graphics course is Computer Graphics using OpenGL 2nd edition by F.S. Hill Jr. There are a number of problems with this book. One problem is that not all the code that is in the book can be used and compiled in a real program; either due to errors in typing, or the code not being used consistently throughout the text. In some causes a function will be written in one way in one part of the book, and then used differently in another place in the book. The goal was to get the ideas and code from the first 7 chapters and appendix 3 of the book into a form that can be easily used by students. This will allow them to focus less on trying to get everything from the book working and more on learning the ideas and principles in the text, while making use of object-oriented skills learned in courses I have taken. In addition to just writing the framework, I also created a few small sample projects. These are to show how the framework being used in different projects. The final results of my project are to be posted on my site, located at http://csserver.evansville.edu/~do7, for other to download and use.

Design Approach
I wrote the framework using the OpenGL and GLUT libraries in C++ language. The framework is comprised of multiple classes each with their own purpose. It is also coded to work both under the Windows and Linux platform. This allows the exact same code to be compiled under both platforms as long as the programmer does not add any platform specific code to their projects. At the very base of the framework is a class that controls setting up the main window for the project and can manipulate different aspects of the window. This class has two children class that are derived from it. One class for 2D projects and the other is for 3D projects. They both set up the window specifically for its needs. These two classes along with a collection of other classes make up the base of the framework.
Window classes

At the very base of the framework is the glWindow class. This class sets up all aspects of the project window such as the size of the window, its position on the screen and the type of frame buffering used. This class has two derived classes, the Canvas class and the Scene class. The Canvas class sets up the window for 2D projects and contains the functionality for drawing different 2D graphics. The Scene class sets up the window for use in a 3D project, and contains the functionality to read in and parse a file describing a scene and draw the resulting scene.

The Canvas class contains simple instructions for drawing different 2D graphics. It can draw a line between two given points. It can also move a cursor point around the window and draw lines using it. This is similar to the functionality found in turtle graphics, in which the turtle is the cursor. In addition to drawing simple lines, there are also functions that will draw regular polygons with a given number of sides. There are also functions for drawing other types of 2D graphics.

The Scene class itself doesn’t contain as much functionality as the Canvas class and can not really do much on its own. It relies mostly on other classes to be able to do anything. It is mostly just used to set up the window for a 3D project. It has the ability to read in a file that describes a scene using the scene description language, or SDL for short, and can create the objects that the files describes and render them to the screen.

Main Helper Classes

There are a few main helper classes used in the base part of the framework. The most used out of these is the Point class. This class is used in both 2D and 3D projects. It is used for setting the vertices of objects and for setting the position of objects. This class has a few forms to allow for storing both 2D and 3D points and also for storing coordinates as integers, floats, or doubles. There are a few other classes that build off of the Point class. These are classes for drawing lines with contain a set of points and can draw the lines between them. The PolyLine class can draw a single line connecting a set of points. A step up from this is the Picture class which is a collection of PolyLines that can be drawn to form a picture. In addition to the Point class, another widely used class is the GLVector class. This class represents a mathematical vector in either 2D or 3D space. It contains different functionality that is usually associated with vectors, such as finding the dot product or cross product of two vectors. This class is used for determining which way a light is pointing, and is used with drawing faces of a 3D object. Another use of the vector class is to determine which way the camera is facing.

The Camera class is used by the 3D projects to provide the view for the project. It represents the user’s eyes in the project and determines what the user will see. It contains multiple vectors that are used to move the camera around and allow it to look around. The position of the camera is stored as a point in 3D space. The camera contains the ability to move in any direction, and can pitch, yaw and rotate. The vector values and the position of the camera are used to produce a matrix that is used to help transform the objects in the project to determine which objects are being looked at by the camera.

As stated earlier the scene class is not very usefully by itself. In addition to using the Camera class to be able to see the objects in the project, the scene class also makes use of a lot of classes for drawing various 3D objects such as cubes, spheres, cones, and
cylinders. Each of these classes can store information about their color and transformations that are to be applied to them before they are rendered. By using a SDL file the scene class can create objects of these types and render them.

A special 3D shape class is the Mesh class which can create an object by using a list of vertices of the object and then using vectors that are normal to the faces of the object to calculate how light should reflect off of it. This class is used to create more complex objects than those that are created by using just cubes, spheres and other basic shapes. It can also read the object in form a file to get all the needed information for creating the object.

While it is not used as extensively as the other classes another class that has an important role in the framework is the pixmap class. This class creates a picture by storing an array of pixels and drawing them to the screen, usually referred to as a bitmap. There is a child class of this class called RGBPixmap that can store a pixmap where each pixel is specified by a RGB values. A RGB value represents a color as three components, one for the red aspect of the color, one for the green and one for the blue. The RGBPixmap class can read and write Windows bitmap file (.bmp) and manipulate them for flipping them and changing the color at different positions in the picture. This class can be used both by 2D and 3D projects. It can either by used to save the current screen image or to display a picture to screen. Another use of the class is to load an image to use as a texture on either a 2D or 3D object.

Sample Projects

To go along with my framework, I have also created a few small sample projects to show how the framework can be used. These include three 2D projects and one 3D project. The 2D projects are an etch-a-sketch clone, a hangman game, and a puzzle program. The 3D project is a simple 3D board game. None of the projects are complex but are just simple projects that show the framework in use, each project was completed in a few hours time.

Etch-a-sketch clone

This program replicates the etch-a-sketch toy. It allows the user to move a dot around the window and draw a line where it has been. The program works by keeping track of the current location of the cursor (dot) and moves its location when ever the user presses a key to move it. As long as the user keeps pressing the same direction the program will just update the location of the cursor. However, when a new direction is chosen the program adds the point to a list of points that it stores. The line is drawn by starting at the beginning of the list and drawing a line between each point until the last point is reached, then from that location a line is drawn to the current location of the cursor.
Hangman

This program replicates the game of hangman. It can be played by either one or two people. In single player mode the program reads in a file that contains a list of words and categories and displays them to the screen with the characters replaced with underscores. In a two player game, one of the players is prompted to enter a category and a phrase for the other person to guess, with the phrase being displayed in the same method as that of a single player game. During the game the player pressing the letter corresponding to their guess and if it is in the phrase then it appears in place of the underscore. If the letter is not in the phrase then they get a body part. The player loses after getting an entire person, which is after six incorrect guesses. This project makes uses of the print function which was not actually present in the book, but that I thought would be a good edition to the framework.
Figure 2: Screen shot of hangman program.

**Puzzle Program**

This program quasi-simulates a jigsaw puzzle. It reads in an 800x600 24-bit color bitmap and breaks it up into 300 40x40 bitmaps. The program then randomized the position of the pieces and the user then puts the pieces back in order. Unlike a real jigsaw puzzle the pieces are square and can not be rotated. They can only have their location swapped with another piece. The user clicks on two pieces then presses the space bar to swap two pieces. Due to it being harder to assemble a picture using this method, as opposed to a real puzzle, I have added a cheat key that will solve a piece of the puzzle for you. It will start at the lower left corner of the puzzle and find the first spot with an incorrect piece and put the correct piece into that spot. When the puzzle if fully complete the lines of the puzzle disappear and the picture is displayed. This program relies heavily on the use of the RGB pixmap class from the framework. This class gave the ability to read in the bitmap file and to display it on the screen.
Board Game

This is a simple 3D board game for 3 players. The goal of the game is to move around the board and return to the starting location by landing on it. During the game the title bar displays whose turn it currently is. The camera will move to look at the current player. For a player’s turn, they press the p key once to cause the triangular prism to start spinning. They then press the p key again to stop it. This will result in them getting a number between one and three, which is the number of spaces they get to move around the board. The player’s piece will move to the appropriate spot and the camera will stay focused on them for a few seconds before moving to the next player. When a player reaches the start state again, the game ends, and a message is displayed in the title bar to tell who won. This project makes use of the Camera, RGBixmap, and Mesh classes. The player pieces are mesh objects that are read in from a file. The texture on the board and the prism are textures that are read in upon start up of the program.
Results

The results of working on this project for an entire school year is a framework that will run on both a Windows and Linux platform without any changes to existing code that can be used by other programmers to speed up the coding process of their projects. The framework deals with setting up the main window and allows the programmer to just add stuff to the project that they want. It contains most of the major points from the chapters 1 – 7 and Appendix 3; however, it does not include all the code and ideas from these sections. Also for the functionality I have included in the framework, while it does what it is suppose to do, it does not do so in the most efficient way. Everything is currently coded to work without any real thought effort being put into making the code work efficiently. However, the current framework meets the goal of being able to be used by future students. The code for the framework, along with the sample projects will be posted on my site for other to download and use.

Along with the code for the framework itself, a help file in the form of an html document, is also being released to explain to users how to use the framework. The framework help files will explain the basics of using the framework and include templates to be used for both 2D and 3D projects, and will also include makefiles that can be used to compile a 2D and 3D project under a Linux environment. The help files will also contain a description of what is in each header file with an explanation of the classes and member functions found in the files. These descriptions were taken from the
comments in the files themselves, and just put into a form that is more accessible and usable for the user of the framework.

Possible Future Additions

There are a multiple things that can be done to this framework in the future. The main thing that can be done is to optimize the code for it to perform its operations using the fastest methods possible. This will improve the performance of projects written using this framework. Another thing that needs to be fixed is the fact that some classes still have all their information in the public section of the class and do not make use private data or methods to access and modify these values. For the 3D shape classes they all could be set up to give the user more control over the object they are creating. At the current moment all the shapes have a default size and must be scaled to change this size.

In addition to just fixing up what has been done, there is also stuff from the sections that I was not able to add, along with stuff from the sections that I did not cover, that could still be added. There is also functionality that is not included in the book that can be added to the framework. This includes stuff like anti-aliasing, mipmapping, and other OpenGL features that are used to improve quality of projects. With enough work, this framework can be made to be useable to a larger range of people other than just those who are taking the graphics courses taught at the University of Evansville.

Conclusion

The goal of this senior project was to create an OpenGL framework that can be used by future students and other people to create OpenGL projects. This was needed as the code in the current textbox being used in the computer graphics course does not contain code that is useable by students and also requires them to do a lot of the work on their own. By creating a framework, the students will have a useable collection of code to use in their projects and can also spent more time learning more advanced features of OpenGL and not have to spent time coding all the basics in order to be able to create a project. This framework will be posted on my site for other to download and use in their projects. In addition the framework, small sample programs have been written to show how the framework can be used.

I have created a framework in C++ using the OpenGL and GLUT libraries. The framework is made up of multiple classes, each with their own function. At the very core of the framework is the glWindow class which controls the aspects of the main window of the project. The two children of this class are designed to configure the main window to fits its specific needs, one for 2D projects and one for 3D projects. In addition to the window classes are a collection of other classes that are used, resulting in a framework that can control most base functionality of the project.

The results of this project are a framework that can be used under both the Windows and Linux platforms without any changes needing to be made to the framework itself. While it does not contain all the code and ideas from the section of the book I was working on, it does contain the main points and is useable by future students for their projects. I have also created four small sample projects to show off the use of my framework. These include an etch-a-sketch clone, a hangman game, a puzzle program, and a 3D board game. The framework also includes a help file that explains to the user
how to use it in their project. The framework and sample programs will be available on my website, located at http://csserver.evansville.eud/~do7, for other to download and use.