CALIFORNIA STATE SCIENCE FAIR
2011 PROJECT SUMMARY

Name(s)

Yousuf Soliman

Project Number

S1425

Project Title

Cube It: Creating a 3D Rubik's Cube Simulator in C++ and OpenGL

Abstract

Objectives/Goals

The objective of this project was to program a 3D Rubik's Cube simulator in C++ and OpenGL. The application should be able to scramble and solve the Rubik's Cube. Along with that, the application should allow the user to put in their own position and have the application generate a moves list needed to solve it.

Methods/Materials

The program I wrote has a graphical user interface in which the user can view the Rubik's Cube in 3D or as a 2D net. The application can scramble and solve the Rubik's Cube. The computer scrambles the cube by choosing a random face, executing a move on that face, then moving on to another face; the application does this 30 times per scramble. The application also has the ability to have the user input their own position. For every solve the application will generate a moves list needed to solve the cube.

Results

After the project was completed, many runs were taken. These runs helped show truly how the application should be rated. One hundred thousand runs were executed to see how many moves on average it would take to solve the cube. On average, it was able to solve it in about eighty-one moves. The minimum number of moves was thirty-nine, and the maximum was one hundred and six moves. The application was very successful and was able to accomplish its task.

Conclusions/Discussion

This application has been working flawlessly 100% of the time. The application was able to have the user input their position and the program will solve it. In addition the application can scramble and solve it on its own. The user can also view the cube in 3D or as a 2D net. If the user is viewing it in a 2D map they will have the option to edit the position of the cube. After the application is told to solve the cube it will generate a moves list needed to solve it. Although the application solves the Rubik's Cube fairly quickly, more complex algorithms can result in faster speeds and less moves. The optimal solution would be God's Algorithm, this solves the cube in the minimum amount of moves needed.

Summary Statement

For this project I programmed a 3D Rubik's Cube simulator in C++ and OpenGL.

Help Received

My computer science teacher, Mrs. Najwan, for honing my programming skills and teaching me object oriented programming; My mother for helping me with my board.