CHAPTER 1

INTRODUCTION

1.1 Overview of Computer Graphics

Computer Graphics become a powerful tool for the rapid and economical production of pictures. There is virtually no area in which Graphical displays cannot be used to some advantage so it is not surprising to find the use of CG so widespread.

Although early application in engineering & science had to rely on expensive & cumbersome equipment, advances in computer technology have made interactive computer graphics a practical tool. Today Computer Graphics is found in diverse area such as science, engineering, medicine, business, industry, government, art, entertainment, education and training.

1.2 History

William fetter was credited with coining the term Computer Graphics in 1960, to describe his work at Boeng. One of the first displays of computer animation was future world (1976), which included an animation of a human face and hand-produced by Carmull and Fred Parkle at the University of Utah.

There are several international conferences and journals where the most significant results in computer-graphics are published. Among them are the SIGGRAPH and Euro graphics conferences and the association for computing machinery (ACM) transaction on Graphics journals.

1.3 Problem Statement

"Tower of Hanoi using Computer Graphics" The above is achieved by using a keyboard which is an interface to the computer by making use of standard functions defined in OpenGL package using IDE Code::blocks.

1.4 Objectives of the project

The main objective of this project is to graphically illustrate the Tower of Hanoi algorithm by making the disks stack on top of each other.

The further objectives are:

- To be able to implement OpenGL functions
- To be able to move disks
- To utilize coordinates to detect motion

CHAPTER 2

LITERATURE SURVEY

Computer graphics is graphics created using computers and, more generally, the representation and manipulation of pictorial data by a computer. The development of computer graphics has made computers easier to interact with and better for understanding and interpreting many types of data. Developments in computer graphics have had a profound impact on many types and have revolutionized animation and video game industry.

The term computer graphics includes almost everything on computers that is not text or sound. Today nearly all computers use some graphics and users expect to control their computer through icons and pictures rather than just by typing. Today computers and computer-generated images touch many aspects of our daily life. Computer imagery is found on television, in newspapers, in weather reports, and during surgical procedures.

The project has been developed using Code::Blocks which is a commercial integrated development environment (IDE) with OpenGL (Open Graphics Library). OpenGL is a standard specification to produce 2D and 3D computer graphics. We use, the OpenGL Utility Toolkit called GLUT which is a library of utilities for OpenGL programs.

2.1 Code::Blocks

Code::Blocks is a free, open-source cross-platform IDE that supports multiple compilers including GCC, Clang and Visual C++. It is developed in C++ using wxWidgets as the GUI toolkit. Using a plugin architecture, its capabilities and features are defined by the provided plugins. Currently, Code::Blocks is oriented towards C, C++, and Fortran. It has a custom build system and optional Make support.

Features of Code::Blocks are as follows:

- Compilers

  Code::Blocks supports multiple compilers, including GCC, MinGW, Digital Mars, Microsoft Visual C++, Borland C++, LLVM Clang, Watcom, LCC and the Intel

C++ compiler. Although the IDE was designed for the C++ language, there is some support for other languages, including Fortran and D. A plug-in system is included to support other programming languages.

- Code editor

  The IDE features syntax highlighting and code folding (through its Scintilla editor component), C++ code completion, class browser, a hex editor and many other utilities. Opened files are organized into tabs. The code editor supports font and font size selection and personalized syntax highlighting colours.

- Debugger

  The Code::Blocks debugger has full breakpoint support. It also allows the user to debug their program by having access to the local function symbol and argument display, user-defined watches, call stack, disassembly, custom memory dump, thread switching, CPU registers and GNU Debugger Interface.

- GUI designer

  As of version 13.12 Code::Blocks comes with a GUI designer called wxSmith. It is a derivative port of wxWidgets version 2.9.4.[6] To make a complete wxWidgets application, the appropriate wxWidgets SDK must be installed.

- User migration

  Some of Code::Blocks features are targeted at users migrating from other IDE's - these include Dev-C++, Microsoft Visual C++ project import (MSVC 7 & 10), and Dev-C++ Devpak support.

- Project files and build system

  Code::Blocks uses a custom build system, which stores its information in XML-based project files. It can optionally use external makefiles, which simplifies interfacing with projects using the GNU or qmake build systems.

2.2 OpenGL and Glut

OpenGL (Open Graphics Library) is a standard specification defining a cross-language, cross-platform API for writing applications that produce 2D and 3D computer graphics, describing a set of functions and the precise behaviors that they must perform. From this specification, hardware vendors create implementations - libraries of functions created to match the functions stated in the OpenGL specification, making use of hardware acceleration where possible. Hardware vendors have to meet specific tests to be able to qualify their implementation as an OpenGL implementation.

GLUT is the OpenGL Utility Toolkit, a window system independent toolkit for writing OpenGL programs. It implements a simple windowing application programming interface (API) for OpenGL. GLUT makes it considerably easier to learn about and explore OpenGL programming. GLUT provides a portable API so you can write a single OpenGL program that works across all PC and workstation OS platforms.

2.3 Applications of computer graphics

Nowadays Computer Graphics used in almost all the areas ranges from science, engineering, medicine, business, industry, government, art, entertainment, education and training.

- CG in the field of CAD
  Computer Aided Design methods are routinely used in the design of buildings, automobiles, aircraft, watercraft, spacecraft computers, textiles and many other applications.
- CG in presentation Graphics
  Another major application area presentation graphics used to produce illustrations for reports or generate slides. Presentation graphics is commonly used to summarize financial, statistical, mathematical, scientific data for research reports and other types of reports.2D and 3D bar chart to illustrate some mathematical or statistical report.

- CG in computer Art
  CG methods are widely used in both fine art and commercial art applications. Artists use a variety of computer methods including special purpose hardware, artist's

paintbrush program (lumena), other pain packages, desktop packages, maths packages, animation packages that provide facility for designing object motion. Ex: cartoons decision is an example of computer art which uses CG.

- Entertainment

  Computer graphics methods are now commonly used in making motion pictures, music, videos, games and sounds. Sometimes graphics objects are combined with the actors and live scenes.

- Education and Training

  Computer generated models of physical financial, economic system is often used as education aids. For some training application special systems are designed. Ex: specialized system is simulator for practice sessions or training of ship captain, aircraft pilots and traffic control.

- Image Processing

  Although the methods used in CG image processing overlap, the 2 areas are concerned with fundamentally different operations. In CG a computer is used to create picture. Image processing on the other hand applies techniques to modify existing pictures such as photo scans, TV scans.

The various methods used in this project are as follows:

- display()

  The entire working of the program is graphically displayed on the screen by the contents defined in the function.

- main()

  The execution first start with main() function.

- mouse()

  The button click on the mouse will display various scenes of celebration.

- keyboard()

  Pressing the enter key displays scene1 after displaying the title page.

CHAPTER 3

REQUIREMENT SPECIFICATION

A software requirement definition is an abstract description of the services which the system should provide, and the constraints under which the system must operate. It should only specify the external behavior of the system. The requirements are specified as below:

3.1 Functional requirements

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define *what* a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases.

The applications of computer graphics in some of the major areas are as follows:

- Display of Information
- Design
- Simulation and Animation
- User Interfaces

3.2 Non-functional requirements

These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process and standards. Non-Functional requirements often apply to the system as a whole.

Non-Functional requirements are as follows:

- Dependability:

  The dependability of a computer system is a property of the system that equates to its trustworthiness. Trustworthiness essentially means the degree of user confidence that the system will operate as they expect and that the system will not fail in normal use

- Availability:

  The ability of the system to deliver services when requested. There is no error in the program while executing.

- Reliability:

  The ability of the system to deliver services as specified. The program is compatible with all types of operating system without any failure.

- Safety:

  The ability of the system to operate without catastrophic failure. This program is user friendly and it will never effects on the system.

- Security:

  The ability of the system to protect itself against accidental or deliberate intrusion

3.3 Software requirements

OPERATING SYSTEM	Windows 98, Windows XP, Windows Vista, Windows 7, Windows 8, Windows 10, Ubuntu 14.04LTS.
INTERACTIVE DEVELOPMENT ENVIRONMENT	CodeBlocks 16.01
CODING LANGUAGE	C++

3.4 Hardware requirements

SYSTEM	Pentium IV 2.4 GHz or above
HARD DISK	40 GB, 80 GB, 160 GB or above
MONITOR	15 VGA colour
RAM	256 MB, 512 MB, 1 GB or above

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 Design Description

This project is object oriented that allows the user to view the Tower of Hanoi algorithm visually and to understand how it works. It involves the use of three disks to demonstrate this process. On successful completion of the algorithm the visualization stops. The number of steps required is also shown on the top right corner and the current step that we are in.

The application's background will implement the design aspect of computer graphics and will introduce the animation and simulation aspect of it. The respective keyboard interfacing will lead the user through different stages of the application.

The main objective of this project is that it demonstrates the capability of OpenGL commands in the field of Graphics and Animation.

In proposed system, consists of the addition of the following features:

- Introduction of transformation effects in the project
- Introduction of moving objects
- Introduction of Keyboard Keys for movement of the camera

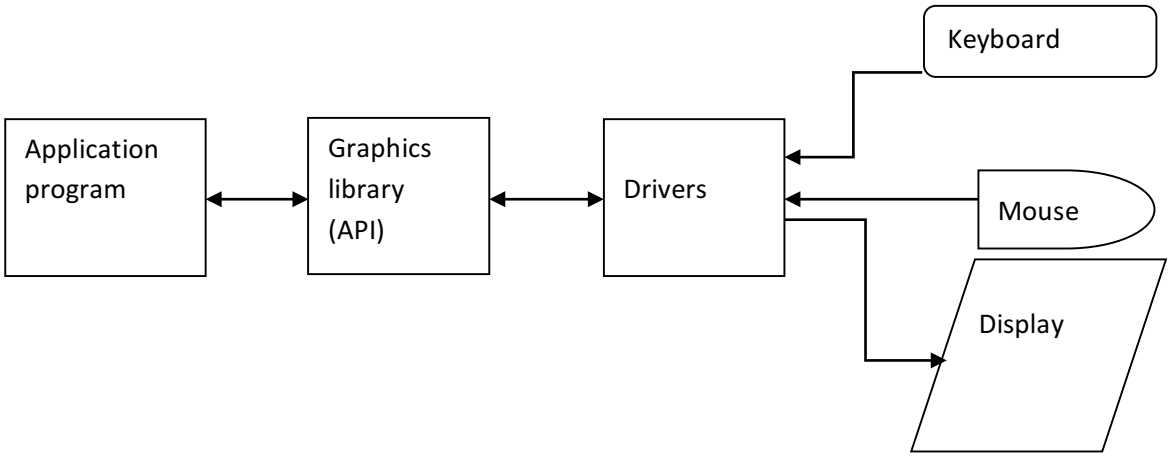

Fig 3.1 Application Programmers model of Graphics System

4.2 Flow Diagram

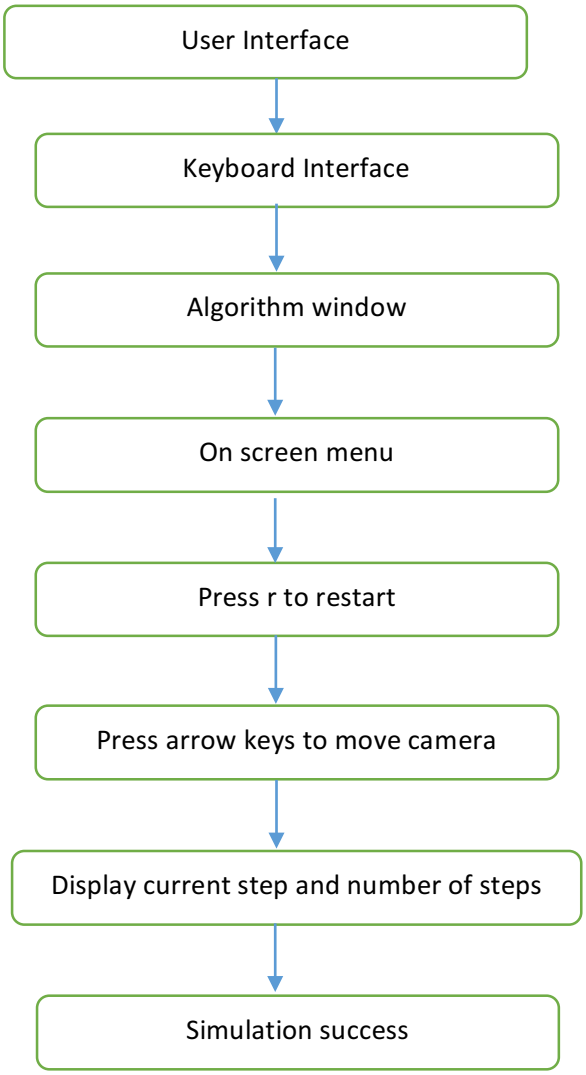

Fig 3.2 Flow of Control Diagram

CHAPTER 5

IMPLEMENTATION

5.1 Header Files

- include <GL/glut.h>

  The GLU (GL Utilities) in glut.h refers to a set of utility functions that make some OpenGL operations easier to program. It is standard on all OpenGL implementations and is generally thought of as part of standard OpenGL.

- include <stdio.h>

  Should be at the beginning of the source file, because the definition for printf () is found in the file stdio.h All header files have the extension .h and generally reside in the /include subdirectory.

- #include <stdlib.h>

  It has various library functions to handle the system. Some of them are exit ().

5.2 Graphics Functions Implementation

Design a project with specific goals, tasks and outcomes. The more specific, the better; the more closely aligned with traditional instructional objectives, the better.

Graphics systems used general – purpose computers with the standard von Neumann architecture. Such computers are characterized by a single processing unit that processes a single instruction at a time. Information had to be sent to the display at a rate high enough to avoid flicker on the display. In the early days of computer graphics, computers were so slow that refreshing even simple images, containing a few hundred-line segments, would burden an expensive computer.

The various functions used in the program are:

- hanoi():

  Implements the main tower of Hanoi algorithm.

- push():

The pin that is visible and that is used to hold the disks is a stack. The push function pushes the disks onto the pin just as we push elements onto the stack. The stack is a FIFO(First In First Out) Data structure.

- pop():

  The pin that is visible and that is used to hold the disks is also stack. The pop function removes the disks from the pin just as we pop elements from the stack. The stack is a FIFO(First In First Out) Data structure.

- drawDisk():

  This function draws the disks that are visible on the pins.

- drawPin():

  This function draws the pins that are visible during the simulation.

- drawAllPins ()

  Draws all the three pins that are visible on the screen an contain the disks that are simulated to move among the pins.

- void glEnd()

  Terminates a list of vertices

- void glColor3f [i f d] (TYPE r, TYPE g, TYPE b)

  Sets the present RGB colors. Valid types are int (I), float (f) and double (d). The maximum and minimum values of the floating-point types are 1.0 and 0.0, respectively.

- void glClearColor()

  Sets the present RGBA clear color used when clearing the color buffer. Variables of GLclampf are floating point numbers between 0.0 and 1.0.

- int glutCreateWindow()

  Creates a window on the display. The string title can be used to label the window. The return value provides a reference to the window that can be used where there are multiple windows.

- void glutInitWindowSize()

  Specifies the initial height and width of the window in pixels.

- void glutInitWindowPosition()

  Specifies the initial position of the top-left corner of the window in pixels.

CHAPTER 6

INTREPRETATION AND RESULTS

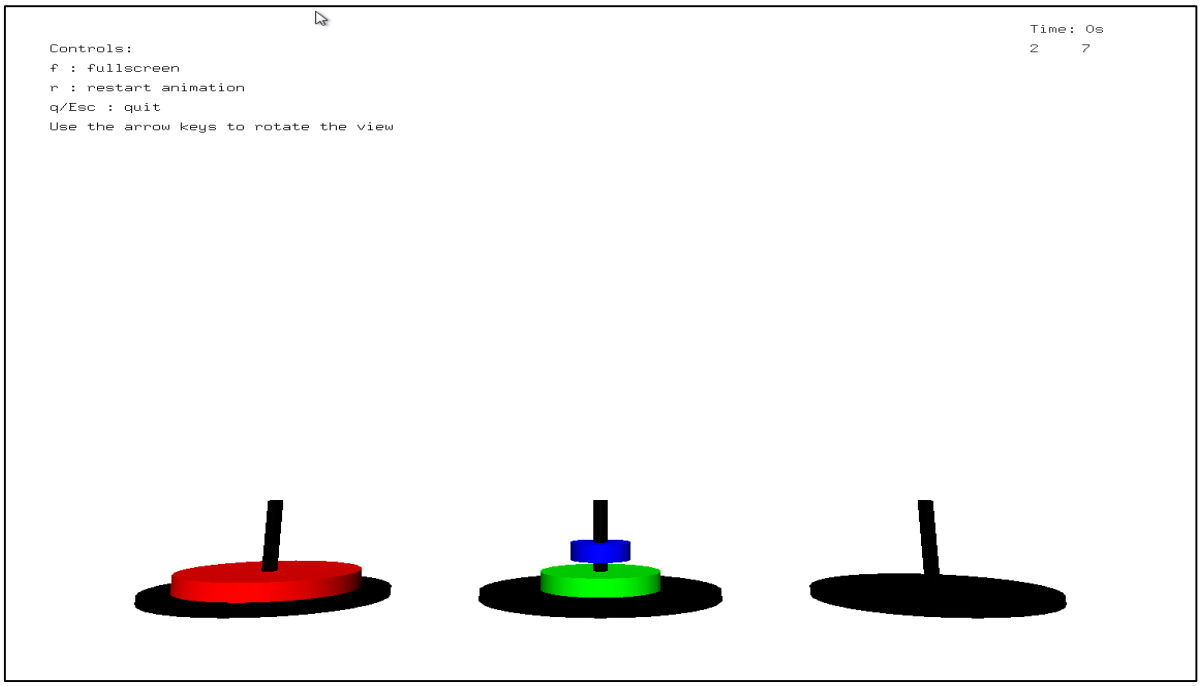

Fig 6.1: Display of main window

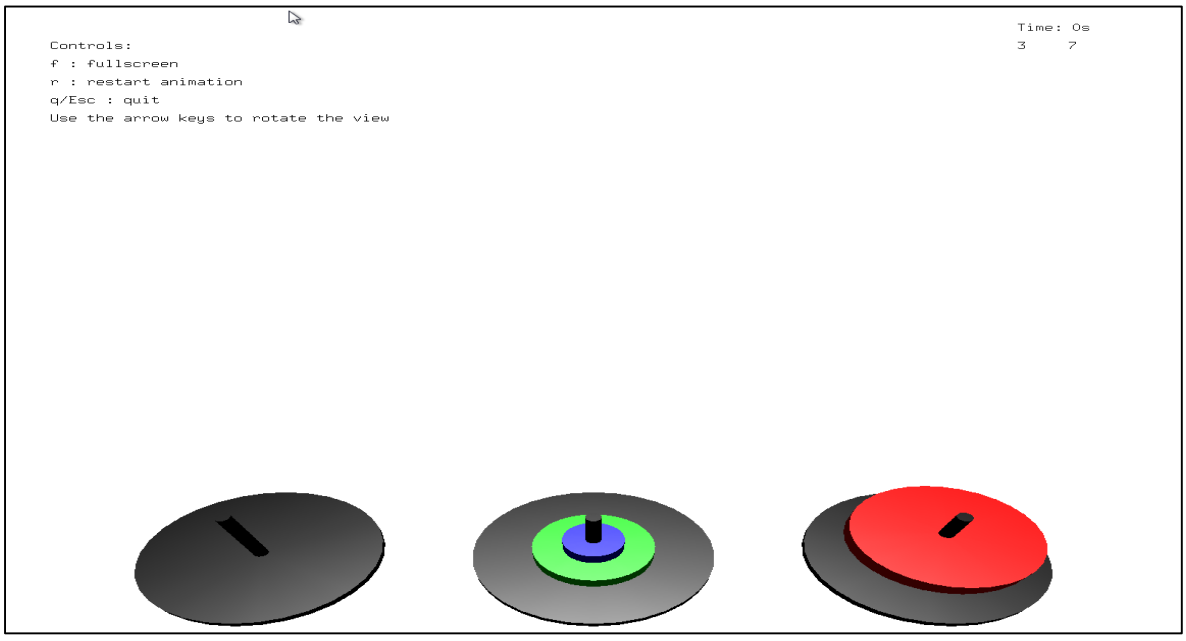

Fig 6.2: Completion of algorithm

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

An attempt has been made to develop an OpenGL package, which meets necessary requirements to create a 3D simulation. Since it is user friendly, it enables the user to interact efficiently and easily.

The development of the mini project has given us a good exposure to OpenGL by which we have learnt some of the technique, which help in development of animated pictures, gaming.

Hence, it is helpful for us even to take up this field as our career too, develop some other features in OpenGL, and provide as a token of contribution to the graphics world.

Further development in the project can be done by adding the ability of the user to choose the number of disks as well as specifying the speed of the disks.

BIBLIOGRAPHY

Reference Books:

- Edward Angle, “Computer Graphics and Visualization Using OpenGL”, 2006.
- Neider, Jackie, Davis, Tom “OpenGL Programming Guide”, 1996.
- Yeshwant Kanetkar,”Computer Graphics under C”, 2004
- OpenGL SuperBible 7th Edition
- OpenGL ES 3.0 Programming Guide (2nd Edition)

Online Resources:

- http://www.opengl.org
- http://learnopengl.com
- http://www.nehe.gamedev.net
- http://www.opengl-tutorial.org/
- http://www.glprogramming.com

Project Name: Tower of Hanoi

PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
✓	✓	✓		✓			✓	✓	✓	✓	

Project Name: Tower of Hanoi

PSO1	PSO2
✓	✓