VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI

Mini Project Report on

"FISHING"

Submitted in the partial fulfillment for the requirements of Computer Graphics & Visualization Laboratory of 6th semester CSE requirement in the form of the Mini Project work

Submitted By

NEHA R	USN: 1BY18CS100
RAMYASHREE N R	USN: 1BY18CS134

Under the guidance of

Mr. SHANKAR R
Assistant Professor, CSE, BMSIT&M

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT
YELAHANKA, BENGALURU - 560064.

2020-2021

BMS INSTITUTE OF TECHNOLOGY &MANAGEMENT

YELAHANKA, BENGALURU – 560064

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project work entitled **"FISHING"** is a bonafide work carried out by **NEHA R (1BY18CS100) and RAMYASHREE N R (1BY18CS134)** in partial fulfillment for *Mini Project* during the year 2020-2021. It is hereby certified that this project covers the concepts of *Computer Graphics & Visualization*. It is also certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in this report.

Signature of the Guide with data
Mr. Shankar R
Assistant Professor
CSE, BMSIT&M

Signature of HOD with date
Dr. Anil G N
Prof & Head
CSE, BMSIT&M

INSTITUTE VISION

To emerge as one of the finest technical institutions of higher learning, to develop engineering professionals who are technically competent, ethical and environment friendly for betterment of the society.

INSTITUTE MISSION

Accomplish stimulating learning environment through high quality academic instruction, innovation and industry-institute interface.

DEPARTMENT VISION

To develop technical professionals acquainted with recent trends and technologies of computer science to serve as valuable resource for the nation/society.

DEPARTMENT MISSION

Facilitating and exposing the students to various learning opportunities through dedicated academic teaching, guidance and monitoring.

PROGRAM EDUCATIONAL OBJECTIVES

1. Lead a successful career by designing, analyzing and solving various problems in the field of Computer Science & Engineering.
2. Pursue higher studies for enduring edification.
3. Exhibit professional and team building attitude along with effective communication.
4. Identify and provide solutions for sustainable environmental development.

ACKNOWLEDGEMENT

We are happy to present this project after completing it successfully. This project would not have been possible without the guidance, assistance and suggestions of many individuals. We would like to express our deep sense of gratitude and indebtedness to each and every one who has helped us make this project a success.

We heartily thank our Principal, **Dr. MOHAN BABU G N**, BMS Institute of Technology &Management, for his constant encouragement and inspiration in taking up this project.

We heartily thank our Professor and Head of the Department, **Dr. ANIL G N**, Department of Computer Science and Engineering, BMS Institute of Technology &Management, for his constant encouragement and inspiration in taking up this project.

We gracefully thank our Project Guide, **Mr. Shankar R**, Assistant Professor, Department of Computer Science and Engineering for his intangible support and for being constant backbone for our project.

Special thanks to all the staff members of Computer Science Department for their help and kind co-operation.

Lastly, we thank our parents and friends for the support and encouragement given throughout in completing this precious work successfully.

NEHA R (1BY18CS100)

RAMYASHREE N R (1BY18CS134)

ABSTRACT

‘A picture is worth a thousand words’ goes the ancient Chinese proverb. This has become a cliché in our society after the advent of inexpensive and simple techniques for producing pictures.

Computers have become a powerful medium for the rapid and economical production of pictures. There is virtually no area in which graphical displays cannot be used to some advantage. Graphics provide a so natural means of communicating with a computer that they have become widespread. The fields in which Computer Graphics find their uses are many. Some of them being User Interfaces, Computer Aided Design, Office automation, Desktop Publishing, plotting of mathematical, scientific or industrial data, Simulation, Art, Presentations, Cartography, to name a few…Here, we have tried to incorporate and present the working environment of a fishing.

Our project fishing is based on human and nature interaction theme. Which is 2D interactive animation. the fishes will be in the sea the scaleman comes near the sea and drops the wire into the sea to hunt the fishes. Then finally the scaleman catches the fishes from the sea.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 COMPUTER GRAPHICS

Computer graphics are graphics created using computers and, more generally, the representation and manipulation of image data by a computer hardware and software. The development of computer graphics, or simply referred to as CG, 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 of media and have revolutionized the animation and video game industry. 2D computer graphics are digital images—mostly from two-dimensional models, such as 2D geometric models, text (vector array), and 2D data. 3D computer graphics in contrast to 2D computer graphics are graphics that use a three-dimensional representation of geometric data that is stored in the computer for the purposes of performing calculations and rendering images.

The user controls the contents, structure, and appearance of the objects and of their displayed images by using input devices, such as keyboard, mouse, or touchscreen. Due to close relationships between the input devices and the display, the handling of such devices is included in the study of computer graphics. The advantages of the interactive graphics are many in number. Graphics provides one of the most natural means of communicating with a computer, since our highly developed 2D and 3D patter-recognition abilities allow us to perceive and process data rapidly and efficiently. In many design, implementation, and construction processes today, the information pictures can give is virtually indispensable. Scientific visualization became an important field in the 1980s when the scientists and engineers realized that they could not interpret the prodigious quantities of data produced in supercomputer runs without summarizing the data and highlighting trends and phenomena in various kinds of graphical representations.

1.2 OpenGL

OpenGL is the most extensively documented 3D graphics API (Application Program Interface) to date. Information regarding OpenGL is all over the Web and in print. It is impossible to exhaustively list all sources of OpenGL information. OpenGL programs are typically written in C and C++. One can also program OpenGL from Delphi (a Pascal-like language), Basic, Fortran, Ada, and other languages. To compile and link OpenGL programs, one will need OpenGL header files. To run OpenGL programs, one may need shared or dynamically loaded OpenGL libraries, or a vendor-specific OpenGL Installable Client Driver (ICD).

OpenGL is a low-level graphics library specification. It makes available to the programmer a small set of geometric primitives - points, lines, polygons, images, and bitmaps. OpenGL provides a set of commands that allow the specification of geometric objects in two or three dimensions, using the provided primitives, together with commands that control how these objects are rendered (drawn).

1.3 GLUT

The OpenGL Utility Toolkit (GLUT) is a library of utilities for OpenGL programs, which primarily perform system-level I/O with the host operating system. Functions performed include window definition, window control, and monitoring of keyboard and mouse input. Routines for drawing a number of geometric primitives (both in solid and wireframe mode) are also provided, including cubes, spheres, and cylinders. GLUT even has some limited support for creating pop-up menus. The two aims of GLUT are to allow the creation of rather portable code between operating systems (GLUT is cross platform) and to make learning OpenGL easier. All GLUT functions start with the glut prefix (for example, glutPostRedisplay marks the current window as needing to be redrawn).

CHAPTER 2

LITERATURE SURVEY

CG (Computer graphics) started with the display of data on hardcopy plotters and cathode ray tube screens soon after the introduction of computer themselves. It includes the creation, storage, and manipulation of models and images of objects. These models include physical, mathematical, engineering, architectural, and even conceptual or abstract structures, natural phenomena, and so on. Computer Graphics today is largely interactive- the user controls the contents, structure, and appearance of objects and their displayed images by using input devices, such as keyboard, mouse or touch sensitive panel on the screen. Bitmap graphics is used for user-computer interaction. A Bitmap is an ones and zeros representation of points (pixels, short for 'picture elements') on the screen. Bitmap graphics provide easy-to-use and inexpensive graphics-based applications.

The concept of 'desktop' is a popular metaphor for organizing screen space. By means of a window manager, the user can create, position, and resize rectangular screen areas, called windows, that acted as virtual graphics terminals, each running an application. This allowed users to switch among multiple activities just by pointing at the desired window, typically with the mouse. Graphics provides one of the most natural means of communicating with the computer, since our highly developed 2D and 3D pattern – recognition abilities allow us to perceive and process pictorial data rapidly and efficiently. In many design, implementation, and construction processes, the information pictures can give is virtually indispensable.

Computer graphics is the creation and manipulation of pictures with the aid of computers. It is divided into two broad classes:

- Non-Interactive Graphics.
- Interactive Graphics.

2.1 NON-INTERACTIVE GRAPHICS

This is a type of graphics where observer has no control over the pictures produced on the screen. It is also called as Passive graphics.

2.2 INTERACTIVE GRAPHICS

This is the type of computer graphics in which the user can control the pictures produced. It involves two-way communication between user and computer. The computer upon receiving signal from the input device can modify the displayed picture appropriately. To the user it appears that the picture changes instantaneously in response to his commands. The following fig. shows the basic graphics system:

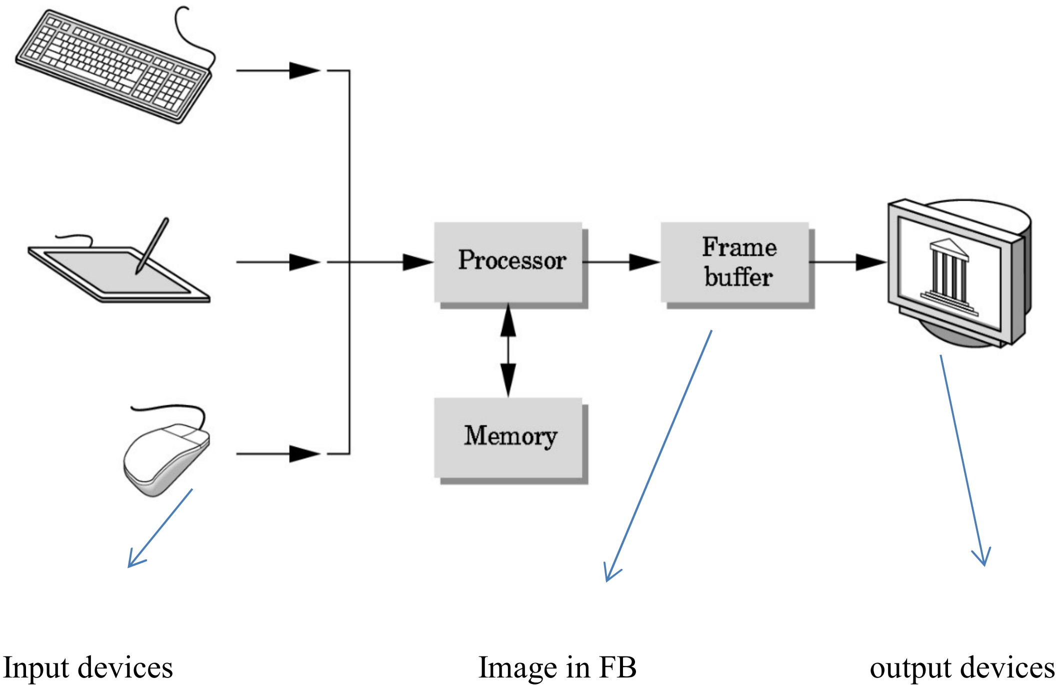

Fig 2.2: Basic Graphics System.

CHAPTER 3

PROJECT IN DETAIL

3.1 PROBLEM STATEMENT

Computer graphics is no longer a rarity. It is an integral part of all computer user interfaces, and is indispensable for visualizing 2D; 3D and higher dimensional objects. Creating 3D objects, rotations and any other manipulations are laborious process with graphics implementation using text editor. OpenGL provides more features for developing 3D objects with few lines by built in functions. The geometric objects are the building blocks of any individual. Thereby developing, manipulating, applying any transformation, rotation, scaling on them is the major task of any image development.

Our project "Fishing" is based on Human and Nature interaction theme, which is 2D interactive animation. The purpose of this project is to implement and understanding the basic openGL functions and accurately know the working of it, which is the important aspects of computer graphics.

3.2 MOTIVATION

The ultimate motivation for this project is to provide graphical interfaces between user and the system by using the OpenGL interactive application provided by Computer graphics. And also, our main goal is to implement(apply) the knowledge whatever we have learnt about the Open Graphical Library for designing mini graphical applications.

3.3 PROPOSED SYSTEM

Here in this project, Fishing is based on Human and Nature interaction theme, which is 2D interactive animation.

We have two menus like select snapshot and exit

In select snapshot we have sub menus FISHES INSIDES THE WATER, to see the SCALMAN, FISHING, CATCHED FISHES and SCALEMAN with FISH.

- In first sub menu we can see the fishes translating from one co-ordinate to another
- In second sub menu we can see the scaleman with the fishing wire to catch the fishes
- In the third sub menu the scaleman puts on the wire to the see to catch fish.
- In the fourth sub menu the fish get hooked with the wire
- In the fifth sub menu we can see the scaleman with the catched fish.

CHAPTER 4

REQUIREMENT SPECIFICATIONS

4.1 HARDWARE SPECIFICATION

- Processor: INTEL / AMD
- Main Memory: 2 GB RAM
- Hard Disk: Built-In Enough
- Mouse: Default Mouse
- Monitor: 1024 x 768 Display Resolution

4.2 SOFTWARE REQUIREMENTS

- Programming Language: C / C++ Using OpenGL
- Operating System: Windows / Linux
- Compiler: C / C++ Compiler (GCC)
- IDE: Code Blocks
- Functional Requirements: GLUT

CHAPTER 5

IMPLEMENTATION

The fishing can be implemented using some of the OpenGL inbuilt functions along with some user defined functions. The inbuilt OpenGL functions that are used mentioned under the FUNCTIONS USED category. The user defined functions are mentioned under USER DEFINED FUNCTIONS category.

5.1 FUNCTIONS USED

- **Void glColor3f (float red, float green, float blue):**

  This function is used to mention the color in which the pixel should appear. The number 3 specifies the number of arguments that the function would take. The 'f' gives the data type float. The arguments are in the order RGB (Red, Green and Blue). The color of the pixel can be specified as the combination of these 3 primary colors.
- **Void glClearColor(int red, int green, int blue, int alpha):**

  This function is used to clear the color of the screen. The 4 values that are passed as arguments for this function are (RED, GREEN, BLUE, ALPHA) where the red green and blue components are taken to set the background color and alpha is a value that specifies depth of the window. It is used for 3D images.
- **Void glutKeyboardFunc():**

  Where func () is the new keyboard callback function. glutKeyboardFunc sets the keyboard callback for the current window. When a user types into the window, each key press generating an ASCII character will generate a keyboard callback. The key callback parameter is the generated ASCII character.
- **Void GLflush():**

  Different GL implementations buffer commands in several different locations, including network buffers and the graphics accelerator itself. GLflush () empties all of these buffers, causing all issued commands to be executed as quickly as they are accepted by the actual rendering engine. Though this execution may not be completed in any particular time period, it does complete in finite time.
- **Void glMatrixMode(GLenum mode):**

  Where "mode" specifies which matrix, stack is the target for subsequent matrix operations. Three values are accepted are:

GL_MODELVIEW, GL_PROJECTION and GL_TEXTURE

The initial value is GL_MODELVIEW.

The function GlMatrixMode sets the current matrix mode. *Mode* can assume one of these values:

GL_MODELVIEW : Applies matrix operations to the model view matrix stack.

GL_PROJECTION**:** Applies matrix operations to the projection matrix stack.

- **void viewport(GLint x, GLint y, GLsizei width, GLsizei height):**

  Here, (x, y) specifies the lower left corner of the viewport rectangle, in pixels. The initial value is (0, 0).

  Width, height: Specifies the width and height of the viewport. When a GL context is first attached to a surface (e.g. window), width and height are set to the dimensions of that surface.

- **void glutInit (int *argc, char **argv):**

  GlutInit will initialize the GLUT library and negotiate a session with the window system. During this process, glutInit may cause the termination of the GLUT program with an error message to the user if GLUT cannot be properly initialized. Examples of this situation include the failure to connect to the window system, the lack of window system support for OpenGL, and invalid command line options. GlutInit also processes command line options, but the specific options parse are window system dependent.

- **glOrtho ( ):**

  Syntax: void glOrtho (GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far);

  The function defines an orthographic viewing volume with all parameters measured from the center of the projection plane.

- **void glutMainLoop(void):**

  GlutMainLoop enters the GLUT event processing loop. This routine should be called at most once in a GLUT program. Once called, this routine will never stop.

- **glutPostRedisplay():**

  GlutPostRedisplay, glutPostWindowRedisplay - marks the current or specified window as needing to be redisplayed.

4.2 USER DEFINED FUNCTIONS

- **Void fish():** This function used to display the fish on the scene
- **Void sea():** This function depicts the sea on the scene.
- **Void land():** This function used to display the land on the scene.
- **Void sky():** This function used to display the sky on the scene.
- **Void water():** This function used to display the water on the scene.
- **Void man():** This function used to display the fisherman(features include eyes ,nose, eyebrows, mouth, shirt, hands, pant) who comes with the wire to hunt the fishes.
- **Void GoMenu():**

This function would provide the menu that consists following options:

1.Fishes inside the water

2.To see the Scaleman

3.Fishing

4.Catches fish

5.scaleman with fishes.

CHAPTER 6

SNAPSHOTS

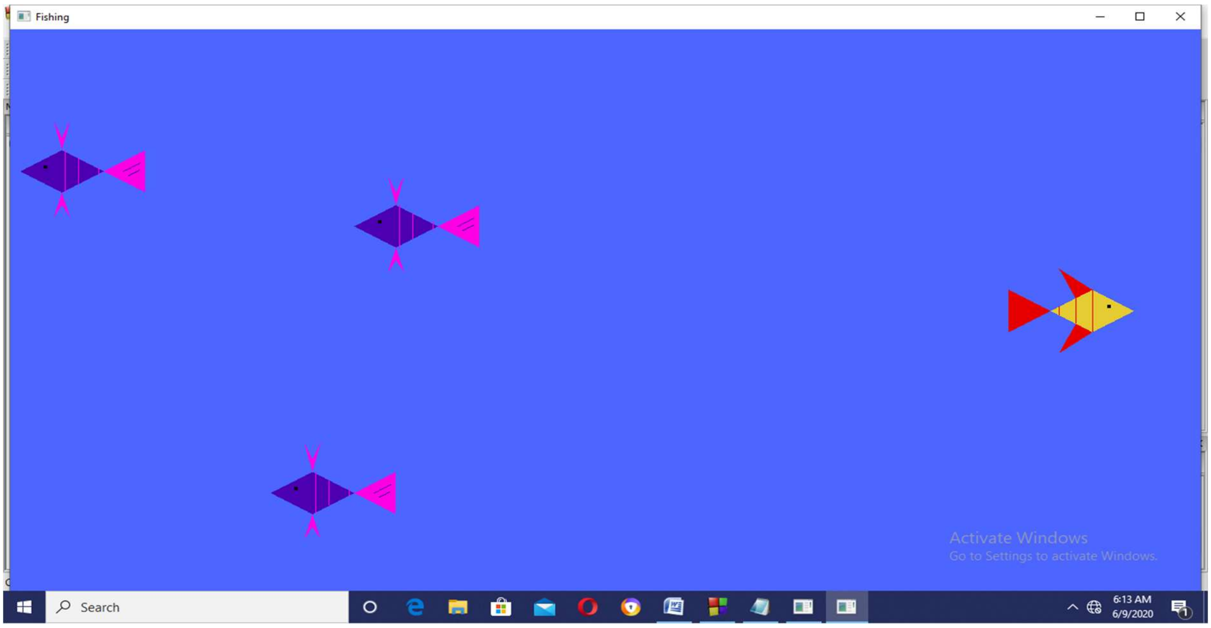

Fig: 6.1 Snap shot shows the fishes inside the water

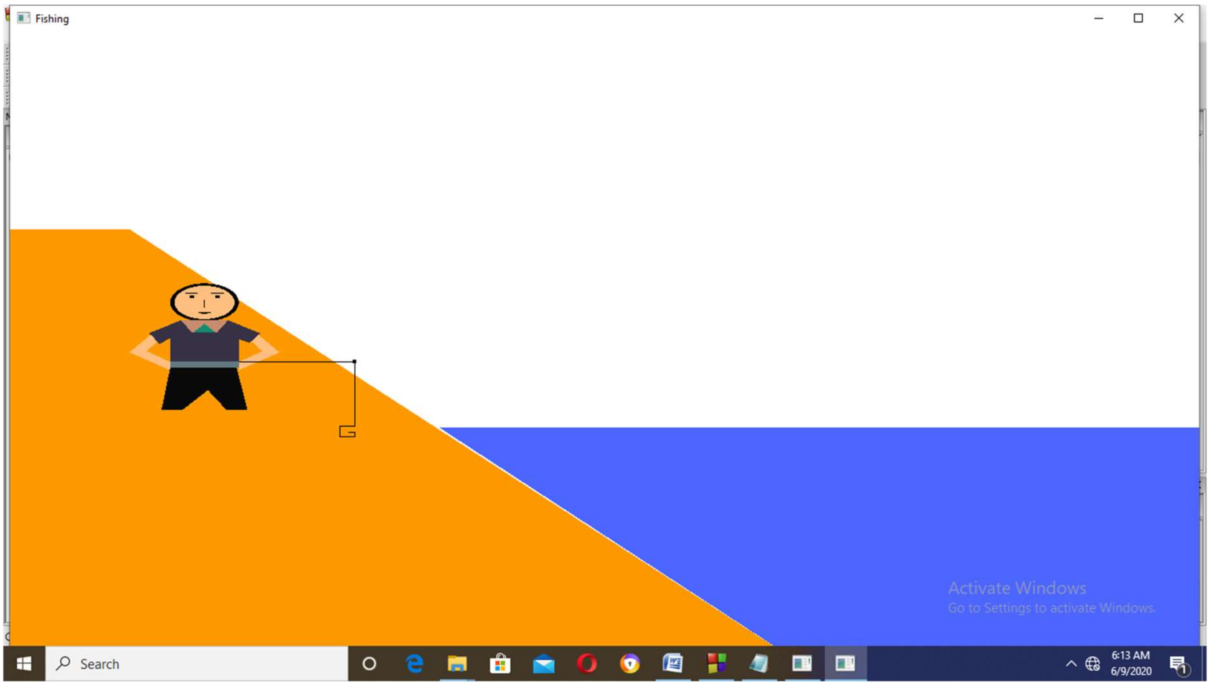

Fig: 6.2 Snap Shot to see the scaleman.

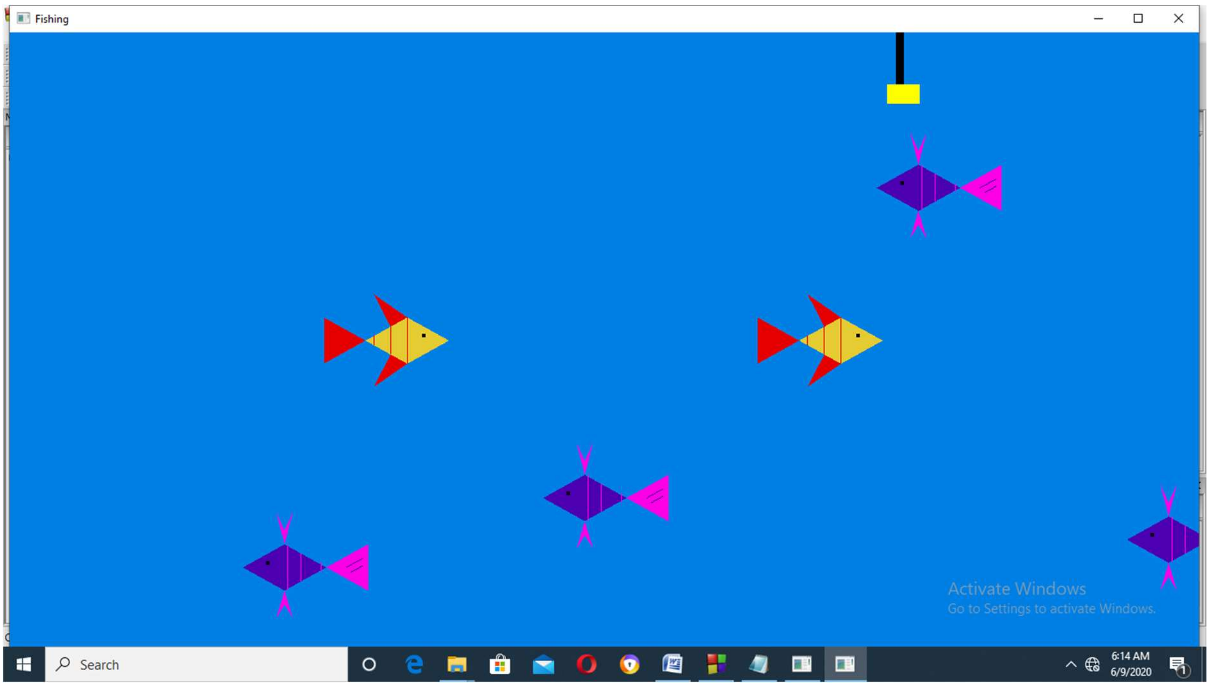

Fig: 6.3 Snapshot shows the fishing.

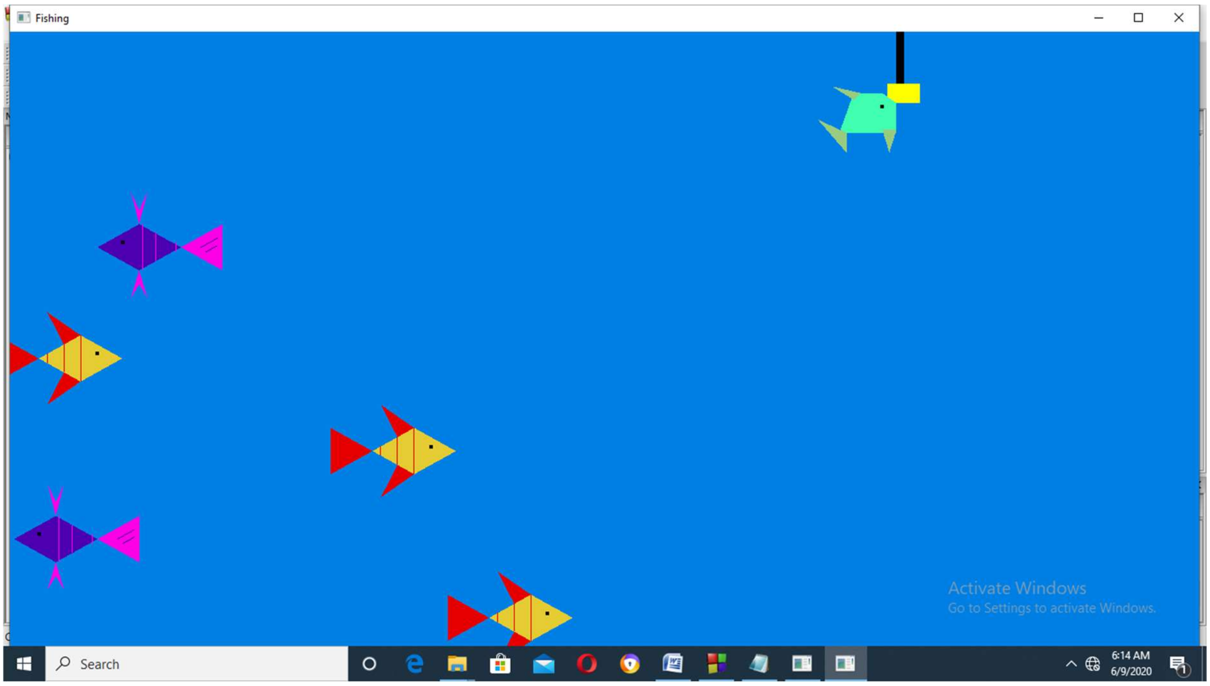

Fig: 6.4 Snap Shot Shows the catched fish.

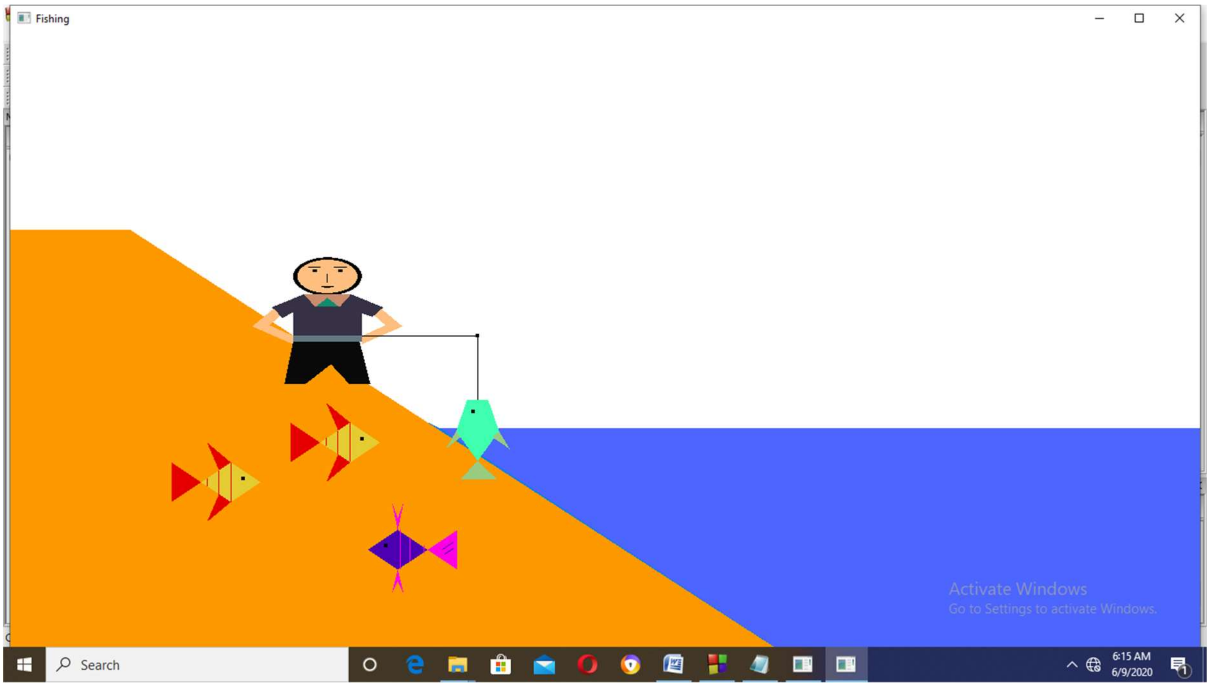

Fig: 6.5 Snap Shot Shows the scalesman with the fishes

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1.1 GENERAL CONSTRAINTS

- As the software is being built to run on Ubuntu platform, which gives access to limited conventional memory, the efficient use of the memory is very important.
- As the program needs to be run even on low-end machines the code should be efficient and optimal with the minimal redundancies.
- Needless to say, the computation of algorithms should also be robust and fast.
- It is built assuming that the standard output device (monitor) supports colors.

7.1.2 ASSUMPTIONS AND DEPENDENCIES

- One of the assumptions made in the program is that the required libraries like GL, GLU and glut have been included.
- The user's system is required to have the C compiler of the appropriate version.
- The system is also expected to have a keyboard and mouse connected since we provide the inputs via these devices.

7.1.3 FURTHER ENHANCEMENTS

The following are some of the features that can be included in the revised versions of this code are:

- Sounds of sea, boat, bus and bridge movement can be incorporated.
- Support for different types of vehicles all moving simultaneously on bridge.
- Support for advanced 3D representation of the entire scenario.
- Support for transparency of layers and originality.

CHAPTER 8

REFRENCES

8.1 BOOK REFERENCES:

- Ariponnammal, S. and Natarajan, S. (1994) ‘Transport Phenomena of Sm Sel-X Asx’, Pramana – Journal of Physics Vol.42, No.1, pp.421-425.
- Barnard, R.W. and Kellogg, C. (1980) ‘Applications of Convolution Operators to Problems in Univalent Function Theory’, Michigan Mach, J., Vol.27, pp.81–94.
- Shin, K.G. and McKay, N.D. (1984) ‘Open Loop Minimum Time Control of Mechanical Manipulations and its Applications’, Proc.Amer.Contr.Conf., San Diego, CA, pp. 1231-1236.

8.2 WEB REFERENCES:

- www.opengl.org
- www.google.com
- www.sourcecode.com
- www.pearsoned.co.in
- www.wikipedia.org