VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI

Mini Project Report on

"CATCH ME GAME"

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

MEGHANA PUVVADA USN: 1BY17CS095

SANKARA TEJA USN: 1BY17CS148

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.

2019-2020

BMS INSTITUTE OF TECHNOLOGY &MANAGEMENT

YELAHANKA, BENGALURU – 560064

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project work entitled **"CATCH ME GAME"** is a bonafide work carried out by **Meghana Puvvada (1BY17CS095) and Sankara Teja (1BY17CS148)** in partial fulfillment for *Mini Project* during the year 2019-2020. 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 date
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, analysing 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 theirhelp and kind co-operation.

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

MEGHANA PUVVADA (1BY17CS095)

SANKARA TEJA (1BY17CS148)

ABSTRACT

The project CATCH ME!!! is created to demonstrate OpenGL's concepts. It encompasses some of the skills learnt in our OpenGL classes such as pushmatrix(), translate(), popmatrix(), timer function.

The main aim of the project is to play the game. To start with the game an object will be moving on the screen. You have to catch the object by clicking the left mouse button on the object, there will be ten chances and each carries ten points. You can choose different level in the game.

Controls:-

Press:- 1: START GAME

2: HOW TO PLAY.

3: to go 'ABOUT GAME' or LEFT CLICK to go back.

4: ABOUT GAME.

aboutf() shows about the game.

Lev() shows different levels of the game.

Keyboard() shows control from the keyboard.

Mypos() shows current position.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1Brief Introduction:

Computer graphics is responsible for displaying art and image data effectively and meaningfully to the consumer. It is also used for processing image data received from the physical world. Computer graphics development has a significant impact on many types of media and has revolutionized animation, movies, advertising, video games, and graphic design in general.

- Graphics provides one of the most natural means of communicating with a computer, since our highly developed 2D or 3D pattern recognition abilities allow us to perceive and process pictorial data rapidly.
- Computers have become a powerful medium for the rapid and economical production of pictures.
- Graphics provide a so natural means of communicating with the computer that they have become widespread.
- Interactive graphics is the most important means of producing pictures since the invention of photography and television.
- We can make pictures of not only the real world objects but also of abstract objects such as mathematical surfaces on 4D and of data that have no inherent geometry.
- A computer graphics system is a computer system with all the components of the general purpose computer system. There are five major elements in system: input devices, processor, memory, frame buffer, output devices.

1.2 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 many graphical applications.

1.3 Scope:

The scope is to use the basic primitives defined in openGL library creating complex objects. We make use of different concepts such as pushmatrix(),translate() ,popmatrix(),timer function.

1.4 Problem Statement:

The project CATCH ME IF YOU CAN!!! is created to demonstrate OpenGL's concepts. It encompasses some of the skills learnt in our OpenGL classes such as pushmatrix(), translate() , popmatrix(), timer function.

1.5 Proposed System:

In this project, we have proposed graphical interfaces between user and the system by using the OpenGL concepts. This project deals with the visualization of 2D objects which are specified in the program. In this when you start the game you need to chase the object of the type rectangular in shape. And there even exists a timer in the game, where you need to chase and catch the rectangular object. The rectangular object holds for 10 points. And the more times you chase and catch the object (i.e. the rectangular object), the more score you gain at the end of the timer.

CHAPTER 2: LITERATURE SURVEY

The basic functions like glcolor3f(…..); glrotatef(…..); gltranslate(…..) etc. that are most commonly Used in the code are taken form the prescribed VTU text book "INTERACTIVE COMPUTER GRAPHICS" 5th edition by Edward Angel.[1].

The lab programs in the syllabus also serve as a basic template for creating a project. The usage of colors and specifications are taken from the various programs that were taught in the lab. [1]. The VTU prescribed text book serves as a huge database of functions and they are used in the project.

The C concepts which are used, are being taken from the book named Yeshwant Kanitkar.

CHAPTER 3: SYSTEM REQUIREMENT

The software requirement specification is the official statement of what is required for development of particular project. It includes both user requirements and system requirements. This requirement document is utilized by variety of users starting from project manager who gives project to the engineer responsible for development of project. It should give details of how to maintain, test, verify and what all the actions to be carried out through life cycle of project.

1. User requirement:

- Easy to understand and should be simple.
- The built-in functions should be utilized to maximum extent.
- OpenGL library facilities should be used.

2. Software requirements:

- Operating system: Windows XP
- Language: Visual studio
- API: OpenGL

3. Hardware requirements:

- Processor: Pentium IV 2.0GHz and above
- RAM: 1GB DDR with 256 MHZ
- Hard disk: 40GB

CHAPTER 4: SYSTEM ANALYSIS

OpenGL is the industry-leading, cross-platform graphics application programming interface (API), and the only major API with support for virtually all operating systems. Many languages, such as Fortran, Java, Tcl/Tk, and Python, have OpenGL bindings to take advantage of OpenGL visualization power. In this article, we present OpenGL Toolkit, a truly platform-independent binding to OpenGL for computer graphics. Ch is an embeddable C/C++ interpreter for cross platform scripting, shell programming, numerical computing, and embedded scripting. Ch extends C with salient numerical and plotting features. Like some mathematical software packages, such as MATLAB, has built-in support for two and three-dimensional graphical plotting, computational arrays for vector and matrix computation, and linear system analysis with advanced numerical analysis functions based on LAPACK. OpenGL Toolkit allows OpenGL application developers to write applications in a cross-platform environment, and all of the OpenGL application source code can readily run on different platforms without compilation and linking processes. In addition, the syntax of OpenGL Toolkit is identical to C interface to OpenGL. OpenGL Toolkit saves OpenGL programmers' energies for solving problems without struggling with mastering new language syntax. OpenGL Toolkit is embeddable. Embedded OpenGL graphics engine enables graphical application developers or users to dynamically generate and manipulate graphics at run-time. The truly platform independent, scriptable, and embeddable features of OpenGL Toolkit make it a good candidate for rapid prototyping, mobile graphics applications, Web-based applications, and classroom interactive presentation. The design issues of OpenGL Toolkit and its potential applications are presented in the article. A methodology that can be used to implement a Web-based visualization system based on OpenGL and CGI is also introduced. The method described in the article can be easily followed to create a Web-based visualization system at low cost and with minimal effort. The software packages and CGI Toolkit are freely available and can be downloaded from the Internet.

CHAPTER 5: IMPLEMENTATION

FUNCTIONS GL_LINES :

Treats each pair of vertices as an independent line segment. Vertices 2n - 1 and 2n define line n. N/2 lines are drawn.

GL_LINE_LOOP :

Draws a connected group of line segments from the first vertex to the last, then back to the first. Vertices n and n + 1 define line n. The last line, however, is defined by vertices N and N lines are drawn.

Basic Functions :

GlPushMatrix, glPopMatrix Function.

The glPushMatrix and glPopMatrix functions push and pop the current matrix stack.

SYNTAX: void glPushMatrix();

void glPopMatrix(void);

glBegin, glEnd Function :

The glBegin and glEnd functions delimit the vertices of a primitive or a group of like primitives.

SYNTAX: void glBegin;

glEnd(GLenum mode);

PARAMETERS:

mode -

The primitive or primitives that will be created from vertices presented between glBegin and the subsequent glEnd.

Transformation Functions :

glTranslate Function :

The glTranslated and glTranslatef functions multiply the current matrix by a translation matrix.

SYNTAX: void glTranslate(x, y, z);

PARAMETERS: x, y, z - The x, y, and z coordinates of a translation vector.

glMatrixMode Function :

The glMatrixMode function specifies which matrix is the current matrix.

SYNTAX: void glMatrixMode(GLenum mode);

PARAMETERS: mode - The matrix stack that is the target for subsequent matrix operations. The mode parameter can assume one of three values:

Value	Meaning
GL_MODELVIEW	Applies subsequent matrix operations to the modelview .

glLoadIdentity Function :

The glLoadIdentity function replaces the current matrix with the identity matrix.

SYNTAX: void glLoadIdentity(void);

FUNCTIONS USED TO SET THE VIEWING VOLUME

glOrtho : This function defines orthographic viewing volume with all parameters measured from the centre of projection.

Multiply the current matrix by a perspective matrix.

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

PARAMETERES:

 left, right - Specify the coordinates for the left and right vertical clipping planes.

 bottom, top - Specify the coordinates for the bottom and top horizontal clipping planes.

 nearVal, farVal - Specify the distances to the nearer and farther depth clipping planes. These values are negative if the plane is to be behind the viewer.

CALL BACK FUNCTIONS :

glutDisplayFunc Function :

glutDisplayFunc sets the display callback for the current window.

SYNTAX: void glutDisplayFunc(void (*func)(void));

glutReshapeFunc Function :

glutReshapeFunc sets the reshape callback for the current window.

SYNTAX: void glutReshapeFunc(void (*func)(int width, int height));

MAIN FUNCTION :

glutInit Function :

glutInit is used to initialize the GLUT library.

SYNTAX: glutInit(int *argcp, char **argv);

PARAMETERS:

 argcp - A pointer to the program's unmodified argc variable from main. Upon return, the value pointed to by argcp will be updated, because glutInit extracts any command line options intended for the GLUT library.

 Argv - The program's unmodified argv variable from main. Like argcp, the data for argv will be updated because glutInit extracts any command line options understood by the GLUT library.

- glutInit(&argc,argv);

glutInitDisplayMode Function :

glutInitDisplayMode sets the initial display mode.

SYNTAX: void glutInitDisplayMode(unsigned int mode);

PARAMETERS:

 mode - Display mode, normally the bitwise OR-ing of GLUT display mode bit masks. See values below:

GLUT_RGB: An alias for GLUT_RGBA.

GLUT_DOUBLE: Bit mask to select a double buffered window. This overrides GLUT_SINGLE.
GLUT_DEPTH: Bit mask to select a window with a depth buffer.

glutMainLoop Function

glutMainLoop enters the GLUT event processing loop.

SYNTAX: void glutMainLoop(void)

CHAPTER 6: IMPLEMENTATION OF RESULTS

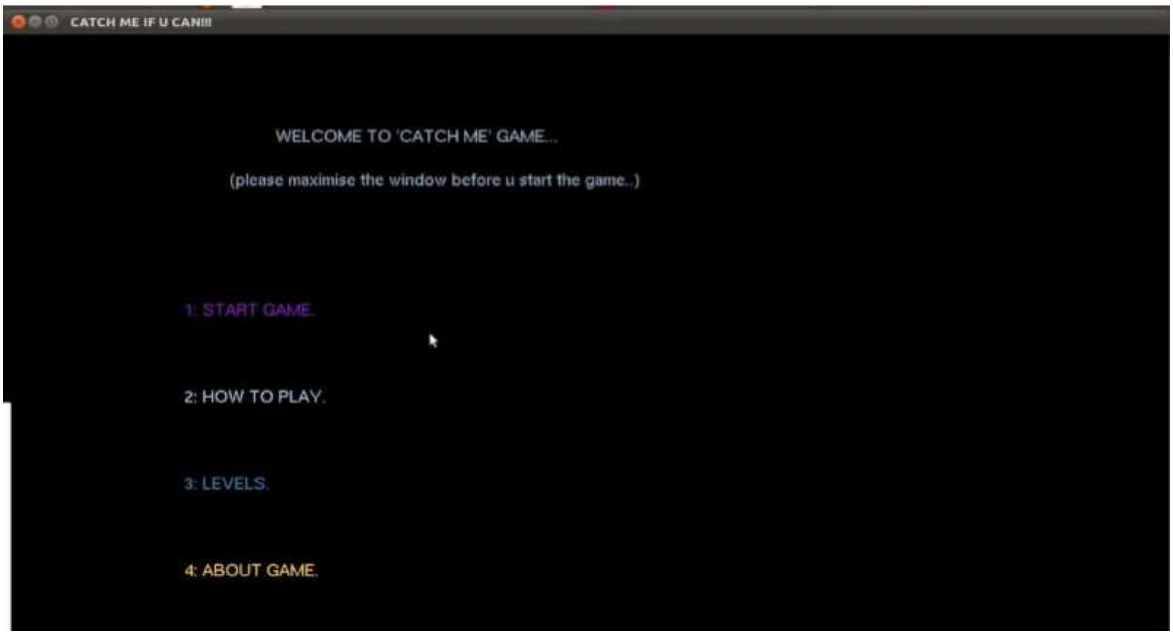

Fig 1: Menu

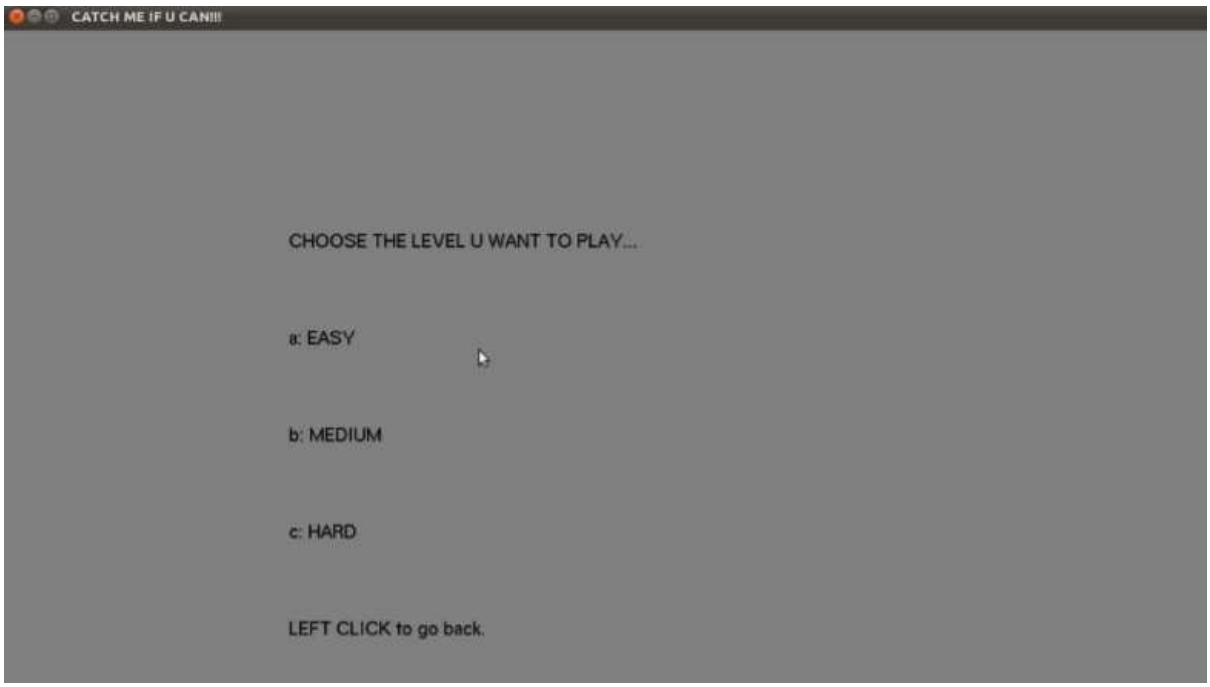

Fig 2: Select level

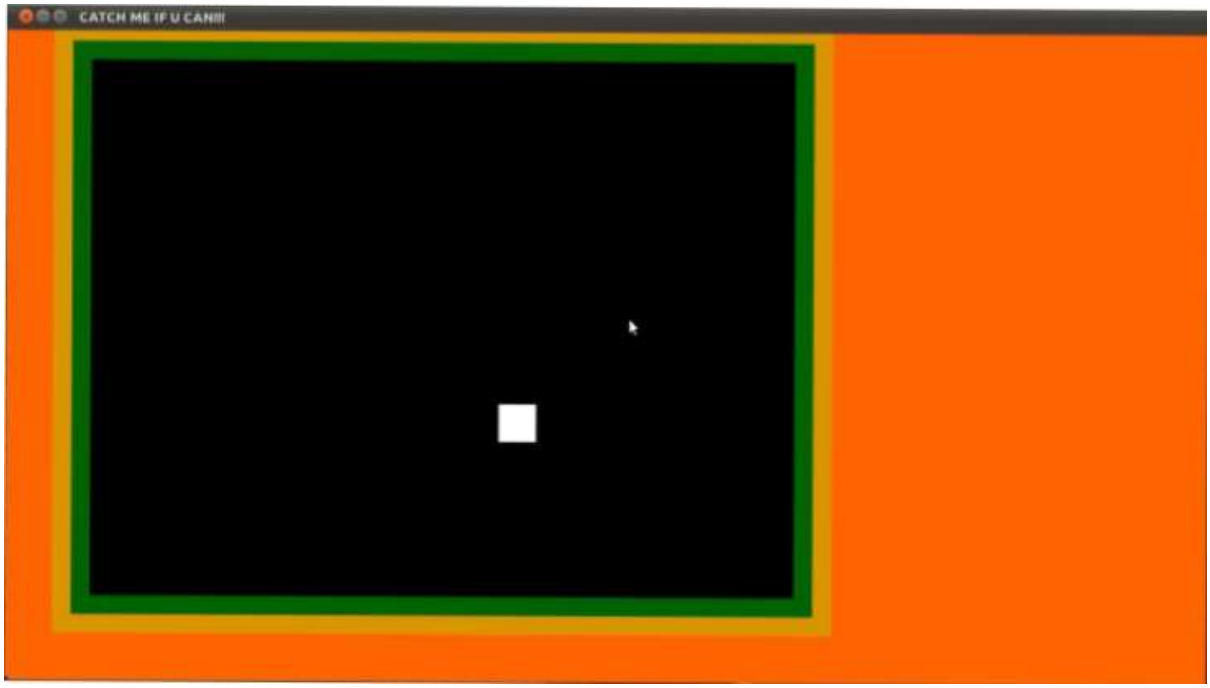

Fig 3: Mouse pointing

Fig 4: Mouse hit

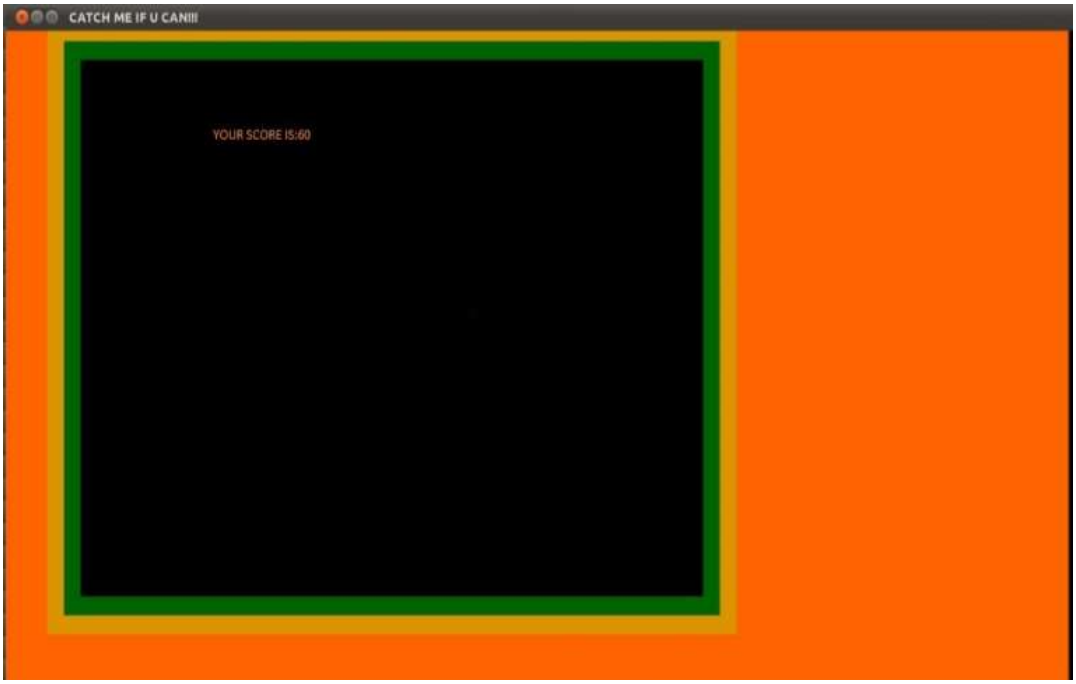

Fig 5: Score interface

CONCLUSION

The project was started with the designing phase in which we figured the requirements needed, the layout design, then comes the detail designing of each function after which, was the testing and debugging stage.

We have tried to implement the project making it as user-friendly and error free as possible. We regret any errors that may have inadvertently crept in.

BIBLIOGRAPHY

[1] OpenGL Programming Guide (Addison-Wesley Publishing Company)

[2] The OpenGL Utility Toolkit (GLUT)Programming Interface -API Version 3 BY MARK J. KILGARD

[3] Interactive computer graphics -A top down approach by using Open GL by EDWARD ANGEL