Visvesvaraya Technological University

Belagavi , 590-014, Karnataka

A Mini Project Report on

"THE ARCHERY GAME"

Submitted in partial fulfilment of the requirements for the award of

Bachelor of Engineering

in

Computer Science and Engineering

Submitted By

AKSHATA UBRANI **2JI18CS002**

ANUSHKA KULKARNI **2JI18CS007**

Under the Guidance of

PROF. PRATIK SAYANAK

Department of Computer Science and Engineering

Sri Bhagawan Mahaveer Jain Educational & Cultural Trust's

Jain College of Engineering

Belagavi-590 014

Academic Year 2020-21

Sri Bhagawan Mahaveer Jain Educational & Cultural Trust's

Jain College of Engineering

Belagavi-590 014

Department of Computer Science and Engineering

Certificate

This is to certify that the mini-project entitled **"THE ARCHERY GAME"** is carried out by **Miss. AKSHATA,** bearing **USN-2JI18CS002,** a bonafide student of **Jain College of Engineering, Belagavi**, in partial fulfilment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** from **Visvesvaraya Technological University, Belagavi**, during the academic year **2020-21.** It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The mini project report has been approved as it satisfies the academic requirements in respect of **Computer Graphics Laboratory with Mini Project** prescribed for the said degree.

PROF. PRATIK SAYANAK
Guide

PROF. PAVAN UGHADE
HOD, CSE

Sri Bhagawan Mahaveer Jain Educational & Cultural Trust's

Jain College of Engineering

Belagavi-590 014

Department of Computer Science and Engineering

Certificate

This is to certify that the mini-project entitled **"THE ARCHERY GAME"** is carried out by **AKSHATA UBRANI(2JI18CS002) and ANUSHKA KULKARNI(2JI18CS007),** bonafide students of **Jain College of Engineering, Belagavi**, in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** from **Visvesvaraya Technological University, Belagavi**, during the academic year **2020-21.** It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The mini project report has been approved as it satisfies the academic requirements in respect of **Computer Graphics Laboratory with Mini Project** prescribed for the said degree.

---	---
Prof. Pratik K Sayanak	**Prof. Pavan Ughade**
Guide	**HOD, CSE**

ABSTRACT

Archery game is a precision sport where the competitors aim and shoot at the target using the arrow.

Every arrow is made of 3 parts: Tip Shaft Nock Tip of the arrow is used to hit the target. Tip is drawn using a triangle. Shaft of arrow is drawn using lines. Nock is drawn using quads.

The target used is a block which is drawn using point and enclosing the point within lone loop to create boundary for the target such that only the point hit by the player disappears the target. Disappearing is highlighted by drawing the target hit by the player using white color which appears as a hole. This game needs good vision and good concentration.

TABLE OF CONTENTS

INTRODUCTION

1.1 Introduction to Computer Graphics

COMPUTER GRAPHICS is concerned with all aspects of producing pictures or images using a computer. The field began humbly almost 50 years ago, with the display of a few lines on a cathode-ray tube (CRT); now, we can create images by computer that are indistinguishable from photographs of real objects. We routinely train pilots with simulated airplanes, generating graphical displays of a virtual environment in real time. Feature-length movies made entirely by computer have been successful, both critically and financially. Massive multiplayer games can involve tens of thousands of concurrent participants.

VISUALIZATION is any technique for creating images, diagrams or animations to communicate a message.

1.2 Image Types

➢ 2D computer graphics:

2D computer graphics are the computer-based generation of digital images mostly from two-dimensional models, such as 2D geometric models, text, and digital images, and by techniques specific to them.2D computer graphics are mainly used in applications that were originally developed upon traditional printing and drawing technologies, such as typography, cartography, technical drawing, and advertising. Two-dimensional models are preferred, because they give more direct control of the image than 3D computer graphics, whose approach is more akin to photography than to typography.

There are two approaches to 2D graphics: vector and raster graphics.

- Pixel art: Pixel art is a form of digital art, created through the use of raster graphics software, where images are edited on the pixel level.
- Vector graphics: Vector graphics formats are complementary to raster graphics, which is the representation of images as an array of pixels, as it is typically used for the representation of photographic images.

➢ 3D computer graphics:

With the birth of the workstation computers(like LISP machines, paint box computers and Silicon Graphics workstations)came the 3D computer graphics.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 purposes0 of performing calculations and rendering 2D images.

1.3 Applications of Computer Graphics

Some of the applications of computer graphics are listed below:

- Computational biology
- Computational physics
- Computer-aided design
- Computer simulation
- Digital art
- Education
- Graphic design
- Video Games
- Virtual reality
- Web design

1.4 Introduction to OpenGL

As a software interface for graphics hardware, OpenGL's main purpose is to render two- and three-dimensional objects into a frame buffer. These objects are described as sequences of vertices (which define geometric objects) or pixels (which define images). OpenGL performs several processing steps on this data to convert it to pixels to form the final desired image in the frame buffer.

1.5 Introduction to GLUT

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 on both Win32 PCs and X11 workstations.

GLUT is designed for constructing small to medium sized OpenGL programs. While GLUT is well-suited to learning OpenGL and developing simple OpenGL applications, GLUT is not a full-featured toolkit so large applications requiring sophisticated user interfaces are better off using native window system toolkits like Motif. GLUT is simple, easy, and small. My intent is to keep GLUT that way.

The GLUT library supports the following functionality:

- Multiple windows for OpenGL rendering.
- Call back driven event processing.
- An `idle' routine and timers.
- Utility routines to generate various solid and wire frame objects.
- Support for bitmap and stroke fonts.
- Miscellaneous window management functions.

1.6 Overview of the project

ARCHERY game is single player game where the player hits the target. The target is a block with hole in center & the player should hit the target. There are 15 arrows, 10 blocks.

The arrow is made of three parts tip, shaft and nock. The block is covered by an elastic material which breaks when the tip of the arrow hits it exactly at the center. We have arrow count shown on the screen.

The player should aim the first arrow with lot of concentration as the speed is max initially and decreases as each arrow vanishes. We have used right button of the mouse to help the user know about the instructions. The target can be hit by arrow by pressing ‘r’ key on the keyboard. If the user wishes to quit or exit from the game he can use the key ‘q’ on the keyboard.

Once the player begins the game the arrow starts moving to hit the target by following the given instruction. As the arrow is heading to the target and finally reaches the end of the screen the arrow count increases indication the no of arrows already used. Once the arrow count becomes 15 the game ends. If the player has hit all the targets then he wins the game, otherwise loses it.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Software Requirements

- Operating System : Windows 98/XP or Higher
- Programming Language : C,C++
- Microsoft Visual Studio 2005 or higher: This Software package containing visual basics in C++ language is required.
- Toolkit : GLUT Toolkit, VC++

2.2 Hardware Requirements

This package has been developed on:

- Processor : Pentium Processor
- Processor Speed : 333 MHz
- RAM : 32 MB or Higher
- Graphics Card : 512MB
- Monitor : Color
- Keyboard : Low Profile, Dispatchable Type
- I/O Parts : Mouse, Monitor

CHAPTER 3

DESIGN

3.1 Initialization

Initialize the interaction with the windows. Initialize the display mode, double buffer and depth buffer. Initialize the various callback functions for drawing and redrawing arrows and target blocks, for mouse interface. Initialize the window position and size and create the window to display the output.

3.2Flowchart

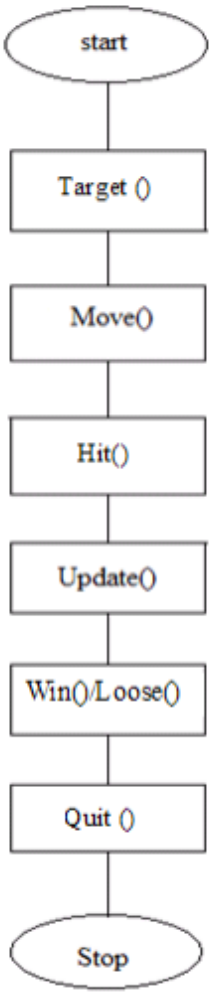

Fig 3.1: Flowchart for archery program.

3.3 Flow of control

The flow of control in the flowchart is with respect to the Archery game.

For any program flowchart is compulsory to understand the program.

We consider the flowchart for the game in which the flow starts from the start and proceeds to the main function after which it comes to the initialization of callback functions and further it proceeds to keyboard function, the flow comes to quit which is the end of the flowchart.

CHAPTER 4

IMPLEMENTATION

4.1 OpenGL Functions

The various functions used in implementing this project are:

- **glutInit(&argc,argv)**
  - This function is used to initialize glut.

- **glutInitDisplayMode()**
  - This function is used to initialize a display mode for the screen. It can choose one of the following constant values as it's parameters:
    - GLUT_SINGLE
    - GLUT_RGB
    - GLUT_DOUBLE

- **glutInitWindowPosition()**
  - This function is used to set the position of the top left corner of the window. It takes in 2 integer values as input parameters which are the coordinates specified for the position for the top left corner of the window.

- **glutInitWindowSize()**
  - This function is used to specify the size of the created window, inside which the scene will be generated. It takes in two integer parameters, the width and height of the window.

- **glutCreateWindow()**
  - This function creates the window with the specified dimensions and position and assigns the string parameter passed to it as the name of the window.

- **glutMainLoop()**
  - This function is called at most once in a GLUT program. Once called, this routine will never return. It will call as necessary, any callbacks required for the program.
- **glBegin()**
  - This function is used in drawing the basic primitives like lines, points, polygons and quads. It takes in a constant value which specifies the primitive to be drawn. Some of the parameters it can have are:
    - GL_POLYGON
    - GL_LINES

- GL_LINE_LOOP
- GL_QUADS
- GL_QUAD_STRIP

- **glClear()**
  - This function is used to clear the contents of the buffer passed as the parameter to this function, onto the screen.
  - It takes a constant value as a parameter like:
    - GL_CLEAR_BUFFER_BIT
    - GL_DEPTH_BUFFER_BIT

- **glClearColor()**
  - This function takes in four floating values:
    - Red
    - Green
    - Blue
    - Alpha
  - Based on RGB values, the color is decided. The Alpha value gives the degree of transparency of the color.
  - The values for RGBA will range from 0.0 to 1.0

- **glEnd()**
  - This function works with the glBegin() function. It marks the end of all the vertices to be used for drawing the primitive.
- **glVertex2x()**
  - This function is used to take in the coordinates of a vertex in a geometric primitive. For integer values, the placeholder ‘x’ is ‘i’ while for floating values, ‘x’ is’d’.

CHAPTER 5

TESTING

Test Cases

Insert for Test Case	Expected Output	Observed Output	Remarks
'R' KEY	Should move the arrow towards the target.	Should move the arrow towards the target.	Pass.
'Q' KEY	Should exits the game.	Should exits the game..	Pass.

Fig 5.1: Testing

CHAPTER 6

RESULTS

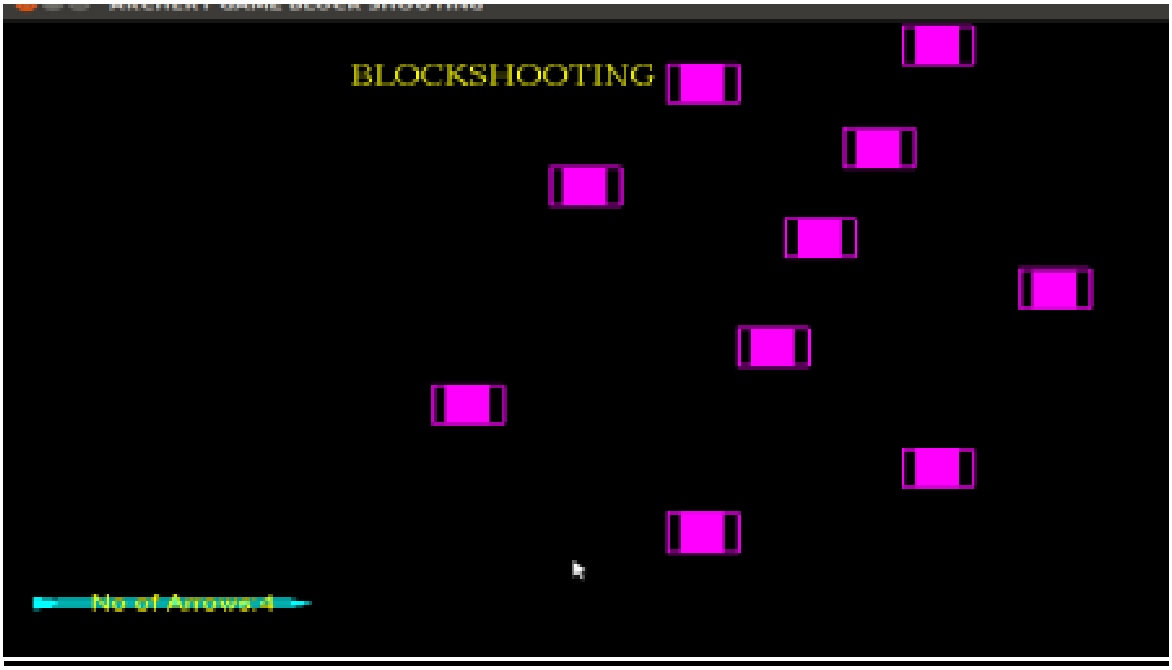

Fig 6.1: Initial position of Archery Game

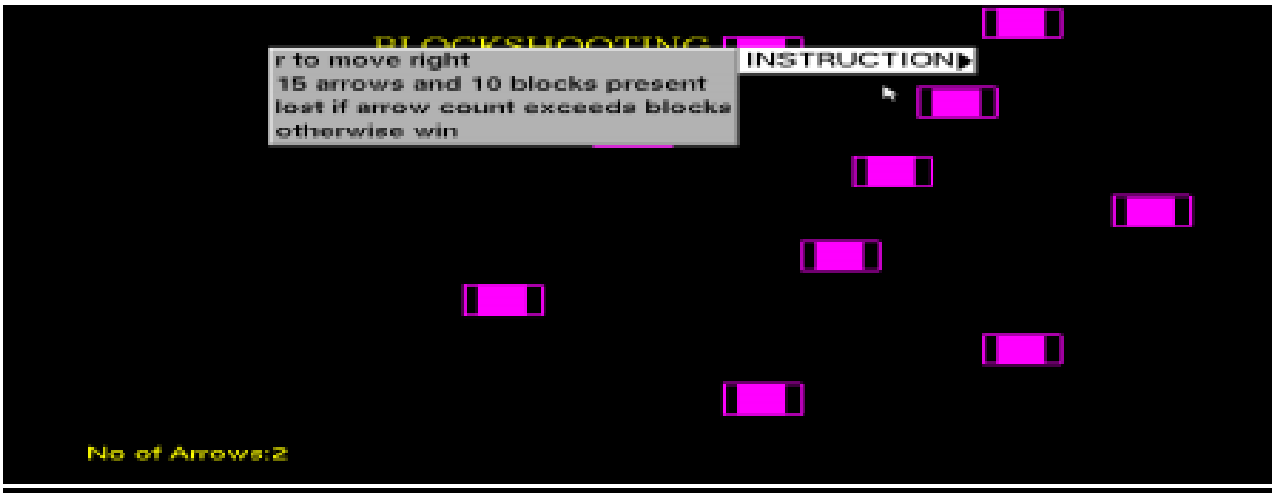

Fig 6.2: Menu of Instructions

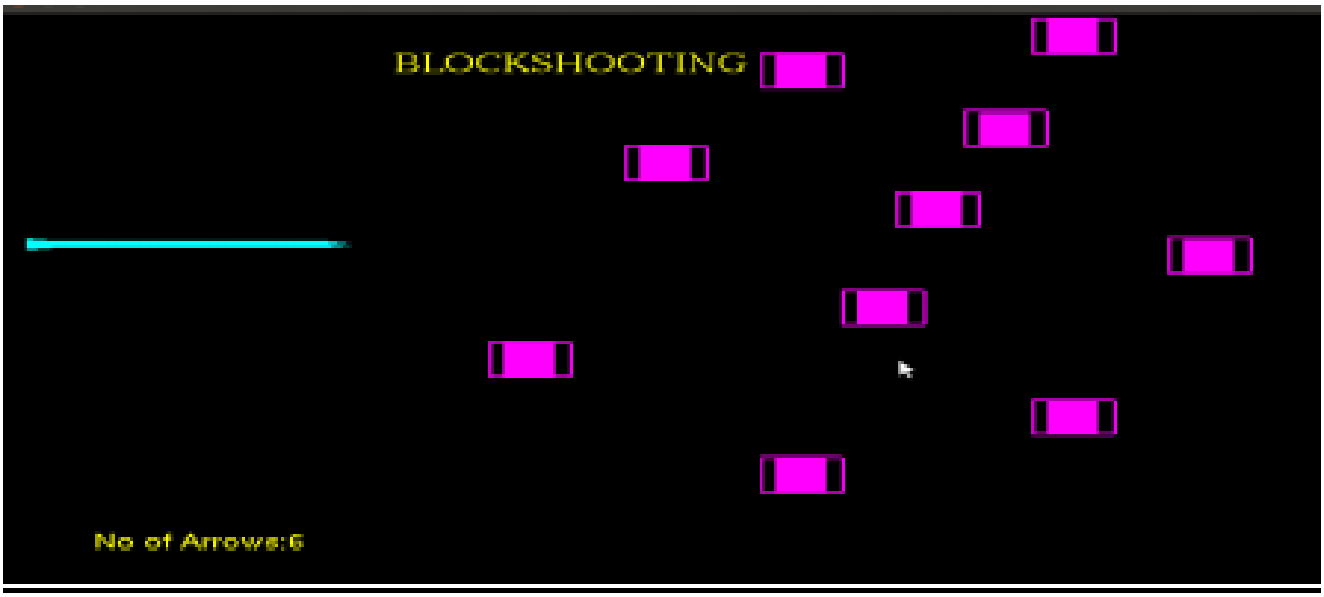

Fig 6.3:Movement of Arrow Upwards

Fig 6.4: Game Over

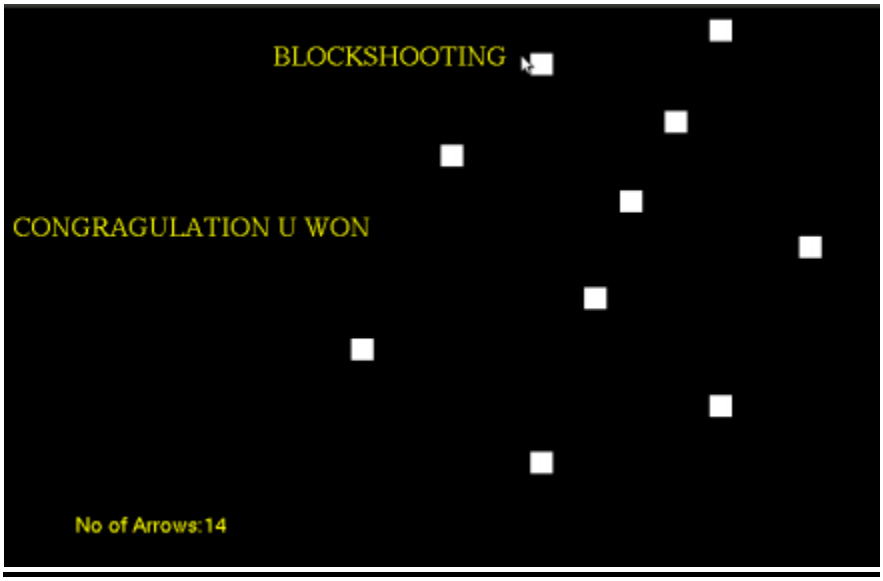

Fig 6.5: Game Won

.

Conclusion

We have tried out level best to build the project efficiently and correctly and have succeeded in building a better project, but may not be a best project. We have implemented the required functions which we had stated earlier. After all testing process, the game is now ready to be played.

References

[1]. Edward Angel *"Interactive Computer Graphics"* Pearson Education, 5th Edition.2008
[2]. Donald Hearn & Pauline baker *"Computer Graphics with OpenGL"* 3rd Edition. 2011

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the progress and completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement ground my efforts with success.

I consider it is a privilege to express my sincere gratitude and respect to all those who guided and inspired me.

I express my sincere thanks and gratitude to our guide ***Prof. Pratik Sayanak ,*** Department of Computer Science & Engineering, JCE, Belagavi, for his constant guidance and suggestions. His incessant encouragement and invaluable support has been of immense help.

It's a great privilege to express my respect to ***Prof. Pavan Ughade,*** HOD, Department of Computer Science & Engineering, JCE, Belagavi, who had been a great source of inspiration towards taking up this mini-project and its successful completion.

I am thankful to ***Dr. K. G. Vishwanath***, Principal, JCE, Belagavi for providing us with the necessary facilities for carrying out this project work successfully.