VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI

Mini Project Report on

"AEROCRASH"

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

Submitted By

VISHAL JAIN HK	**USN: 1BY19CS183**
SHRIKANT K	**USN: 1BY20CS422**
SANDESH D GOWDA	**USN: 1BY18CS137**

Under the guidance of

Prof. Shankar R
Assistant Professor
Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT

YELAHANKA, BENGALURU - 560064.

2021-2022

BMS INSTITUTE OF TECHNOLOGY &MANAGEMENT

YELAHANKA, BENGALURU – 560064

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project work entitled **"AEROCRASH"** is a bonafide work carried out by **VISHAL JAIN HK (1BY19CS183), SANDESH D GOWDA (1BY18CS137) and SHRIKANT K (1BY20CS422)** in partial fulfilment for *Mini Project* during the year 2021-2022. 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
Prof. Shankar R
Assistant Professor
CSE, BMSIT&M

Signature of the HOD
Dr. Thippeswamy G
Professor & HOD
CSE, BMSIT&M

Name and Signature of the Examiners

Internal Examiner

External Examiner

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.

PROGRAM SPECIFIC OUTCOMES

1. Analyse the problem and identify computing requirements appropriate to its solution.
2. Apply design and development principles in the construction of software systems of varying complexity.

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. Thippeswamy G**, Department of Computer Science and Engineering, BMS Institute of Technology and Management, for his constant encouragement and inspiration in taking up this project.

We gracefully thank our Project Guide, Prof. **Shankar R** Assistant Professors, Department of Computer Science and Engineering for their 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.

VISHAL JAIN HK (1BY19CS183)
SHRIKANT K (1BY15CS422)
SANDESH D GOWDA (1BY18CS137)

ABSTRACT

The main aim of the Aerocrash Computer Graphics Mini Project is to illustrate the concepts and usage of pre- built functions in OpenGL. Simulation of “Aeroplane crash” is being done using computer graphics. The development of this animation has large scope to learn computer graphics and visualization from scratch. We will be using OpenGL utility toolkit to implement the algorithm, written in C++ language.

Aeroclash is a simple computer graphic project based on C++ and OpenGL programming. The project displays the 2D transformation animation of an aeroplane which takes off from a runway and crashes to a building. This project is based on the concepts of C++ and OpenGL. OpenGL is the premier environment for developing portable, interactive 2D and 3D graphics applications. 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. OpenGL is most widely used and supported 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide variety of computer platforms. 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.

TABLE OF CONTENTS

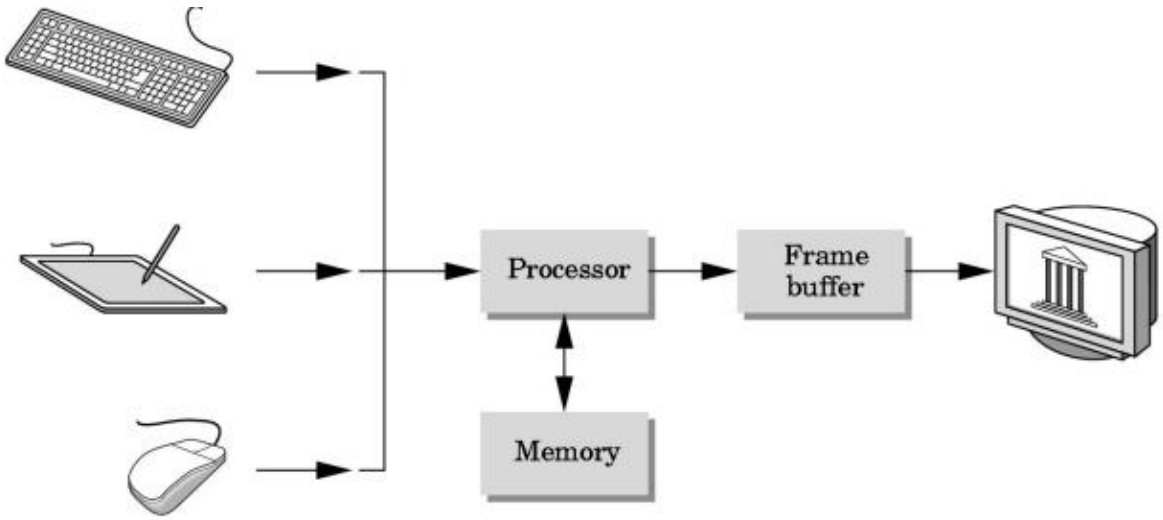
CHAPTER NO	TITLE	PAGE NO
CHAPTER 1	**INTRODUCTION**	**1-2**
	1.1 Computer Graphics	**1**
	1.2 OpenGL Technology	**2**
CHAPTER 2	**SCOPE AND MOTIVATION**	**3**
CHAPTER 3	**PROPOSED SYSTEM**	**3**
CHAPTER 4	**LITERATURE SURVEY**	**4**
CHAPTER 5	**REQUIREMENT AND SPECIFICATIONS**	**5-6**
CHAPTER 6	**DESIGN**	**7-8**
CHAPTER 7	**SYSTEM IMPLEMENTATION**	**9-12**
CHAPTER 8	**INTERPRETATION OF RESULTS**	**13-14**
CHAPTER 9	**CONCLUSION**	**15**
CHAPTER 10	**FUTURE ENHANCEMENTS**	**15**

CHAPTER 1

INTRODUCTION

1.1 Computer Graphics

- 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 OpenGL Technology

OpenGL is the premier environment for developing portable, interactive 2D and 3D graphics applications. Since its introduction in 1992, OpenGL has become the industry's most widely used and supported 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide variety of computer platforms.

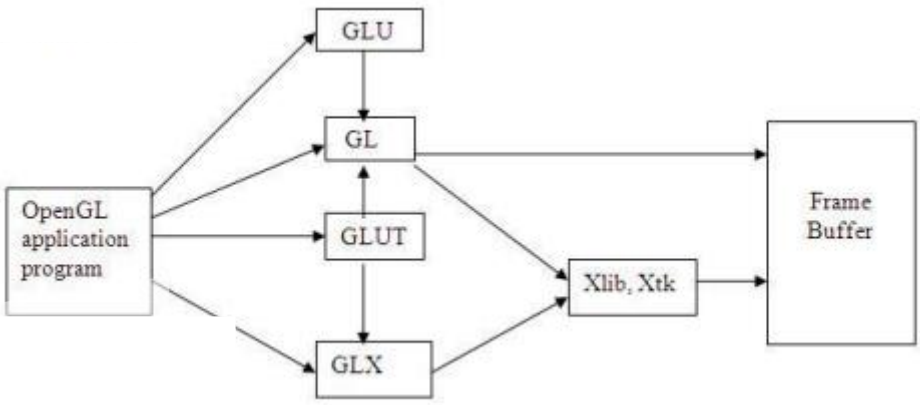
OpenGL fosters innovation and speeds application development by incorporating a broad set of rendering, texture mapping, special effects, and other powerful visualization functions. Developers can leverage the power of OpenGL across all popular desktop and workstation platforms, ensuring wide application deployment.

OpenGL Available Everywhere: Supported on all UNIX® workstations and shipped standard with every Windows 95/98/2000/NT and MacOS PC, no other graphics API operates on a wider range of hardware platforms and software environments.

OpenGL runs on every major operating system including Mac OS, OS/2, UNIX, Windows 95/98, Windows 2000, Windows NT, Linux, Open Step, and BeOS; it also works with every major windowing system, including Win32, MacOS, Presentation Manager, and X-Window System. OpenGL is callable from Ada, C, C++, Fortran, Python, Perl, and Java and offers complete independence from network protocols and topologies.

The OpenGL interface

Our application will be designed to access OpenGL directly through functions in three libraries namely: gl,glu,glut.



CHAPTER 2

SCOPE AND MOTIVATION

Computer graphics deals with generating images with the aid of computers. Today, computer graphics is a core technology in digital photography, film, video games, cell phone and computer display, and many specialized applications. A great deal of specialized hardware and software has been developed, with the displays of most devices being driven by computer graphics hardware. It is a vast and recently developed area of computer science. The phrase was coined in 1960 by computer graphics researchers Verne Hudson and William Fetter of Boeing. It is often abbreviated as CG, or typically in the context of film as computer generated imagery (CGI). The non-artistic aspects of computer graphics are the subject of computer science research.

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, such as photo and video content. Computer graphics development has had a significant impact on many types of media and has revolutionized animation, movies, advertising, video games, in general.

In this project we are trying to develop an easy, efficient, and cost-free method of aeroplane crash simulation and animation.

CHAPTER 3

PROPOSED SYSTEM

With the use of different functionality like of making polygons, using various functions of mouse and creation of menu, the whole story is plotted in this project. This project gives a user-friendly interface for the end user to interact with the frames. This project contains several different frames that are used in a particular sequence to help the user see the animation.

So, we are adding a good visual and making it a video simulation. Because visual is always more interesting and easier to remember.

CHAPTER 4

LITERATURE SURVEY

The basic functions like glcolor3f(…); gltotatef(..),gltranslate(..) etc that are most commonly used in the code are taken from 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 colours and specifications are taken from the various programs that were taught in the lab.[1].

The VTU prescribed textbook serves as a huge database of functions, and they are used in the project.

The C++ concepts which are used are being taken from "object-oriented programming" by Sourav Sahay.[2].

Some concepts like constructing bowl and fountain are taken from the search results in codecolony.com.

CHAPTER 5

REQUIREMENTS AND SPECIFICATIONS

3.1 Purpose of the requirements document

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.

Scope of the project

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.

Definition

The project **DEMOLITION OF A BUILDING BY AEROPLANE CRASH** 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

Acronyms & Abbreviations

OpenGL provides a powerful but primitive set of rendering command, and all higher level design must be done in terms of these commands.

OpenGL Utility Toolkit(GLUT):- windows-system-independent toolkit.

3.2 Specific requirements

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.

Software Requirements:

- Windows OS.
- CodeBlocks.
- Programming Language : C++ & OpenGL

Hardware Requirements:

- Processor- Intel or AMD(Advanced Micro Devices).
- RAM – 2GB
- Hard Disk – 50GB

CHAPTER 6

DESIGN

4.1 User Defined Functions

- **myinit():**
  This function initializes light source for ambient, diffuse, and specular types.

- **display():**
  This function creates and translates all the objects in a specified location in a particular order and rotates the objects in different axes.

  glClear(GL_COLOR_BUFFER_BIT);
  glFlush();
- **timerfunc():**
  This function starts a timer in the event loop that delays the event loop for delay milliseconds.

- **MainLoop():**
  This function whose execution will cause the program to begin an event processing loop.

- **PushMatrix():**
  Save the present values of attributes and matrices placing or pushing on the top of the stack.

- **PopMatrix():**
  We can recover them by removing them from stack.

- **Translated();**

  In translate function the variables are components of the displacement vector.

- **main():**

  The execution of the program starts from this function. It initializes the graphics system and includes many call-back functions.

- **PostRedisplay():**

  It ensures that the display will be drawn only once each time the program goes through the event loop.

CHAPTER 7

SYSTEM IMPLEMENTATION

5.1 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. The following are accepted symbolic constants and their meanings:

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.

Functions used to display

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	Meaning
GL_MODELVIEW	Applies subsequent matrix operations to the modelview matrix stack.

glLoadIdentity Function

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

SYNTAX:

void glLoadIdentity(void);

5.2 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.

5.3 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));

5.4MAIN 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.

CHAPTER 8

INTERPRETATION OF RESULTS

Fig 1: Plane ready to take-off

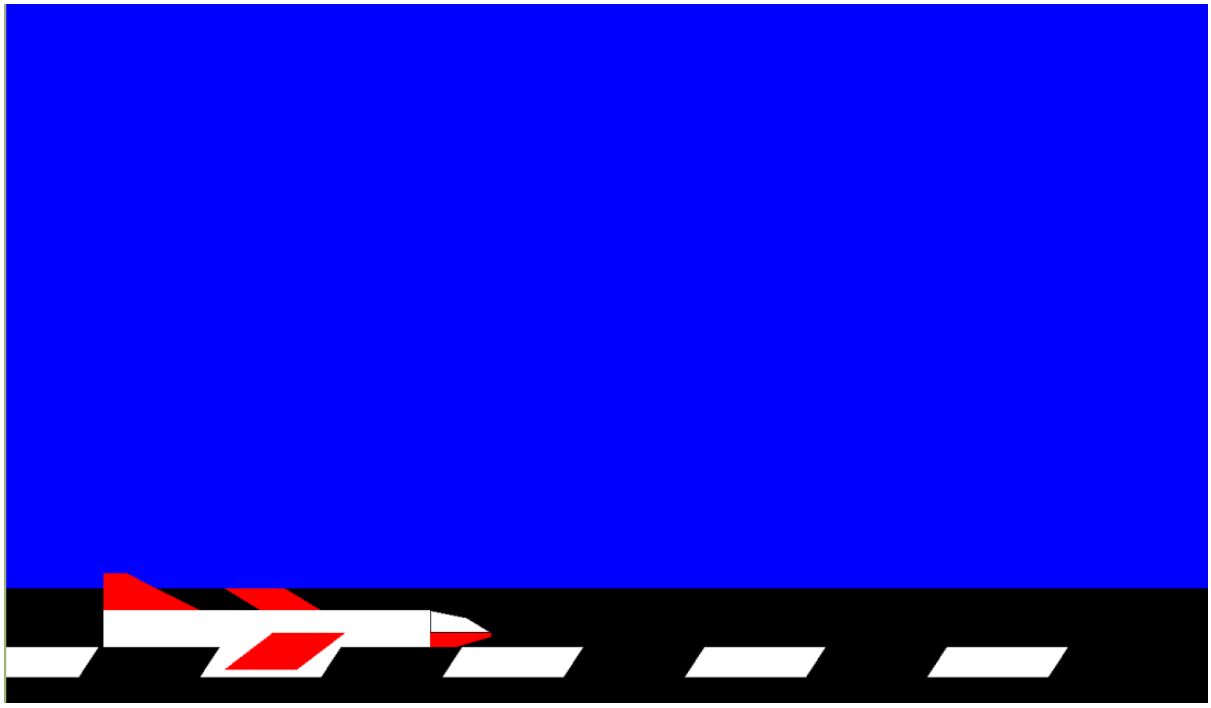


Fig 2: Plane in the sky

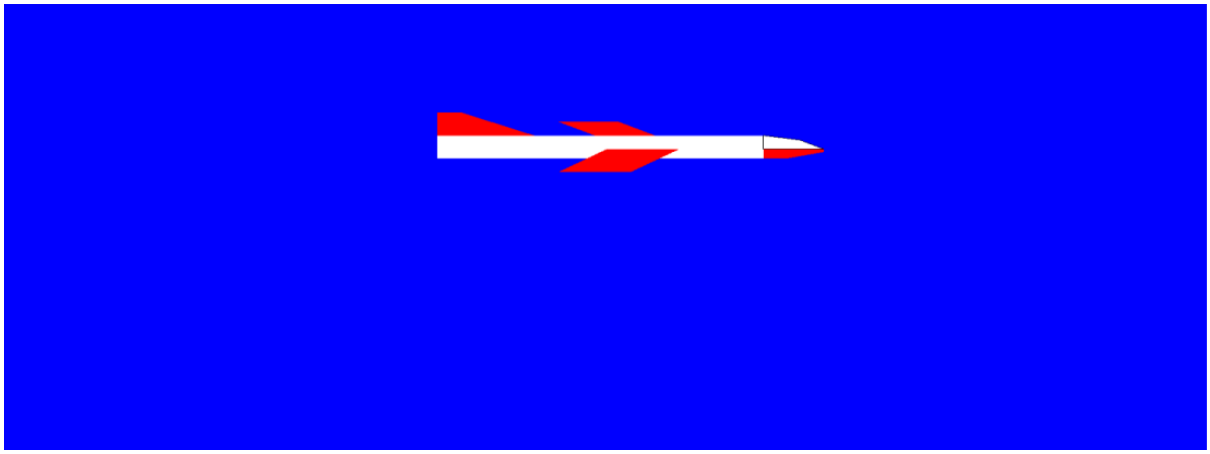


Fig 3: Plane moving towards the building

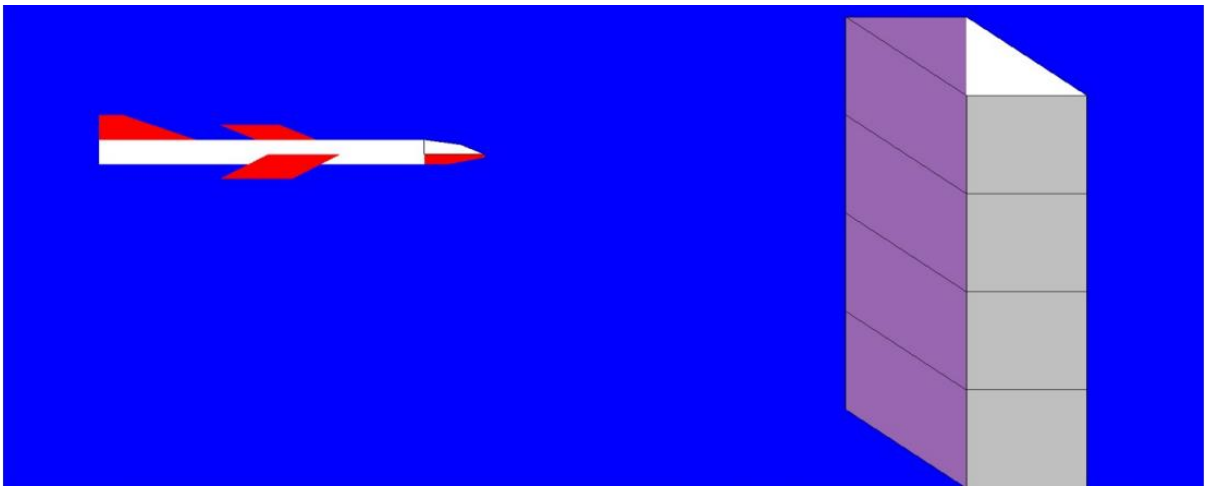
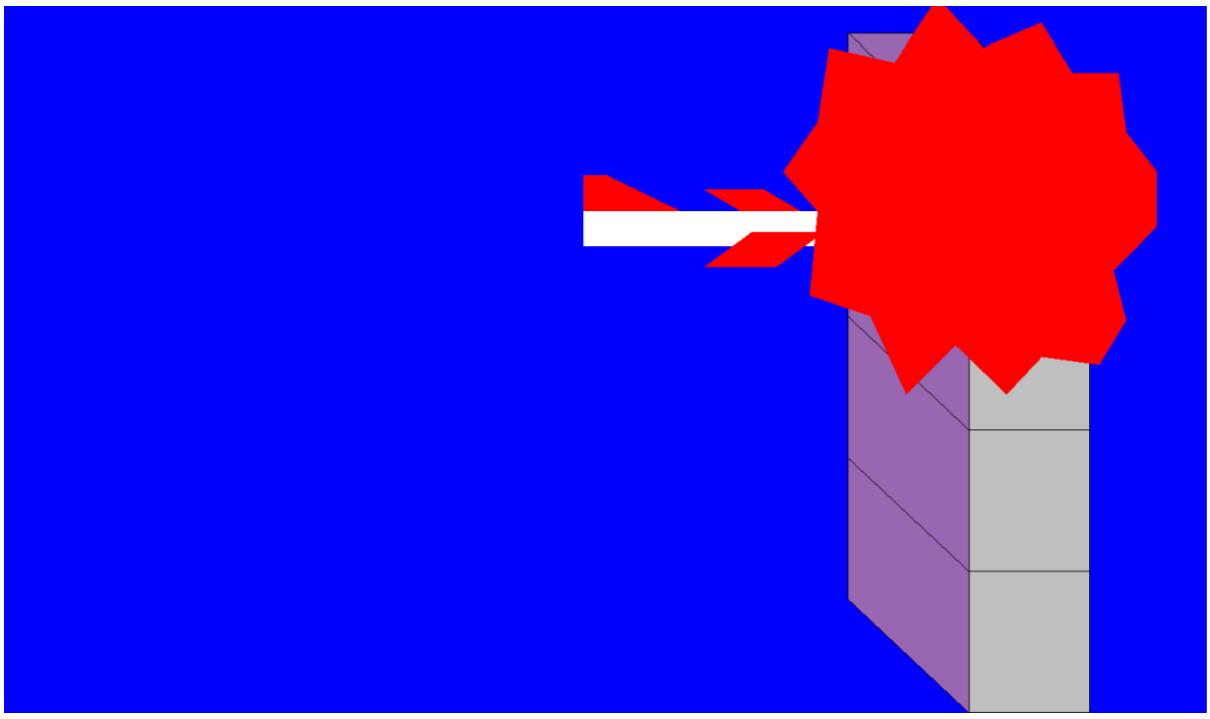


Fig 4: Plane crashing



CHAPTER 9

CONCLUSION

This project is one of the sample projects on Computer Graphics. Though many difficulties were faced during the project as well as many errors occurred, we became succeed to compile and run the program. There may be some limitations on this project as well, so, soon we would like to be hopeful in further improvements. We are highly obliged to all helping hands and to all inspirations to make this project successful. Well, this project will be applicable to most of all. As well as we are more hopeful for more advice, new ideas and inspiration to make more other projects. To had tried our best to include each and every basic features of graphics in our projects.

We aimed it to be an interfacing application to the real world that means our project must not be a project for any examination but also applicable for real world use. We have able to give some benefits to the disability. It somehow makes our life easier in this or that way. From this very project we were able to achieve various knowledge in computer graphics and in logical coding. We refresh our knowledge in C Programming. Moreover, we also gained an experience of group work, team coordination. We learned how teamwork is very much important in engineering field.

CHAPTER 10

FUTURE ENHANCEMENTS

Some of the future enhancements would be:

- Support for advanced 3D representation of the entire scenario.
- Support for transparency of layers and originality that is, simulating the story in a more realistic way.
- Making the user interface of this project more user friendly which will certainly be more effectively and efficiently narrated.

REFERENCES

1) Donald Hearn & Pauline Baker: Computer Graphics with OpenGL Version, 4th Edition, Pearson Education, 2011.
2) Edward Angel: Interactive Computer Graphics- A Top-Down approach with OpenGL, 5 th edition. Pearson Education, 2008.
3) Jackie.L.Neider,Mark Warhol,Tom.R.Davis,"OpenGL Red Book",Second Revised Edition,2005.
4) www.opengl.org
5) https://learnopengl.com/