A Project Report On

2D Aeroplane

Submitted by

Gaurav U D 16IT113

Abhilash V 16IT201

Abhishek Kamal 16IT202

Shrinivas V Shanbhag 16IT244

IV Sem B.Tech (IT)

in partial fulfillment for the award of the degree

of

Bachelor of Technology

In

Information Technology

At

Department of Information Technology

National Institute of Technology Karnataka, Surathkal.

April 2018

CERTIFICATE

This is to certify that the project entitled “2D Airplane Animation” has been presented by Gaurav U D, Abhishek Kamal, Abhilash V, Shrinivas V Shanbhag students of second year, B.Tech (IT), Department of Information Technology, National Institute of Technology Karnataka , Surathkal, on April 19, 2018, during the even semester of the academic year 2017 - 2018, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information Technology at NITK, Surathkal.

Place: NITK, Surathkal

Date: 19-04-2018 (Signature of Course instructor)

ACKNOWLEDGEMENT

This project would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them. We are highly indebted to the Department of Information Technology, NITK for their guidance and constant supervision as well as for providing necessary information and resources for the project and also for their support in completing the project. We would like to express our gratitude towards our teacher Ms Sangeetha Harikantra for her kind co-operation and encouragement which helped us in completion of this project. Our thanks and appreciations also go to our group members in developing the project and people who have willingly helped us out with their abilities.

CONTENTS

ABSTRACT

A simple 2D animation of an airplane's takeoff from the runway , it's travel over a city and finally landing on runway.The airplane’s travel over city has background of clouds , buildings and sky.

User can control the movements of airplane while flying through keyboard interrupt. This project uses drawing of line ,circle, color filling algorithms for drawing airplane, background objects.It uses rotation, translation, transformation for take off and landing.

INTRODUCTION

This project has been developed in C++. OpenGL and GLUT frameworks are used.
In this project a two dimensional airplane is designed which is first at an airport.
This airplane is designed using separate polygon’s merging together. We have used our own color filling algorithms to do this. The airplane take off is displayed using various transformation functions like translation and rotation. Separate backgrounds for airport ,city landscape has been designed which has moving objects like clouds.
The airplane finally reaches the destination airport and lands.

OBJECTIVES

- The aim of this mini project was to make a simple 2D animation of an airplane taking off and landing
- Develop an idea to design some graphical applications.
- This project can be used to study the various functionalities of computer graphics.
- Develop our own algorithms to implement functionalities like translation, rotation, and others.
- Allow the user to control the airplane by implementing Keyboard Interrupt functions in the code

SYSTEM REQUIREMENTS

Operating System	: Windows 7 or above , Linux , Mac OSX.
Processor	: Intel i5 or better.
Interface	: OpenGL

METHODS

The different methods used in this project are :

- polygonScanFill() :

This method is used for coloring all the objects which includes background, objects in the background and also the airplane.

Scanline filling is basically filling up of polygons using horizontal lines or scanlines. The purpose of the SLPF algorithm is to fill (color) the interior pixels of a polygon given only the vertices of the figure. To understand Scanline, think of the image being drawn by a single pen starting from bottom left, continuing to the right, plotting only points where there is a point present in the image, and when the line is complete, start from the next line and continue.

This algorithm works by intersecting scanline with polygon edges and fills the polygon between pairs of intersections.

- circleScanFill():

This method is used to fill color inside circular objects. The basic working of this algorithm is almost same as above ,but since we don’t have straight edges in these objects we use circle drawing algorithms from which we get the left and right points to apply the scan fill algorithm.

TRANSFORMATION METHODS

- translation():

Translation refers to moving an object to a different position on screen. It uses Matrix Multiplication to calculate the new position of the points.

- rotation():

Rotation refers to rotating a point. This also uses Matrix multiplication to rotate the point about a fix point.

WORKING

In this project different backgrounds are used . At the beginning the airplane will be at an airport. This background has runway and airport buildings .The next background shows a city landscape with a number of buildings. The next one shows a hilly country side. Then the airplane reaches its destination airport and proceeds to landing. All these backgrounds have moving clouds and sun.

The polygonScanFill() method is used to create a background. We draw different objects like the airport ,hills , buildings , airplane.

The circleScanFill() method is used to create circular objects like sun and clouds.

translate() method is used to make the airplane move forward.

rotation() method is used to make the airplane rotate above and below while taking off and landing.

Keyboard Interrupt is used to move the airplane up ,down and right using glutSpecialFunc() - a callback function which is called when up, down, right, or left key is pressed.

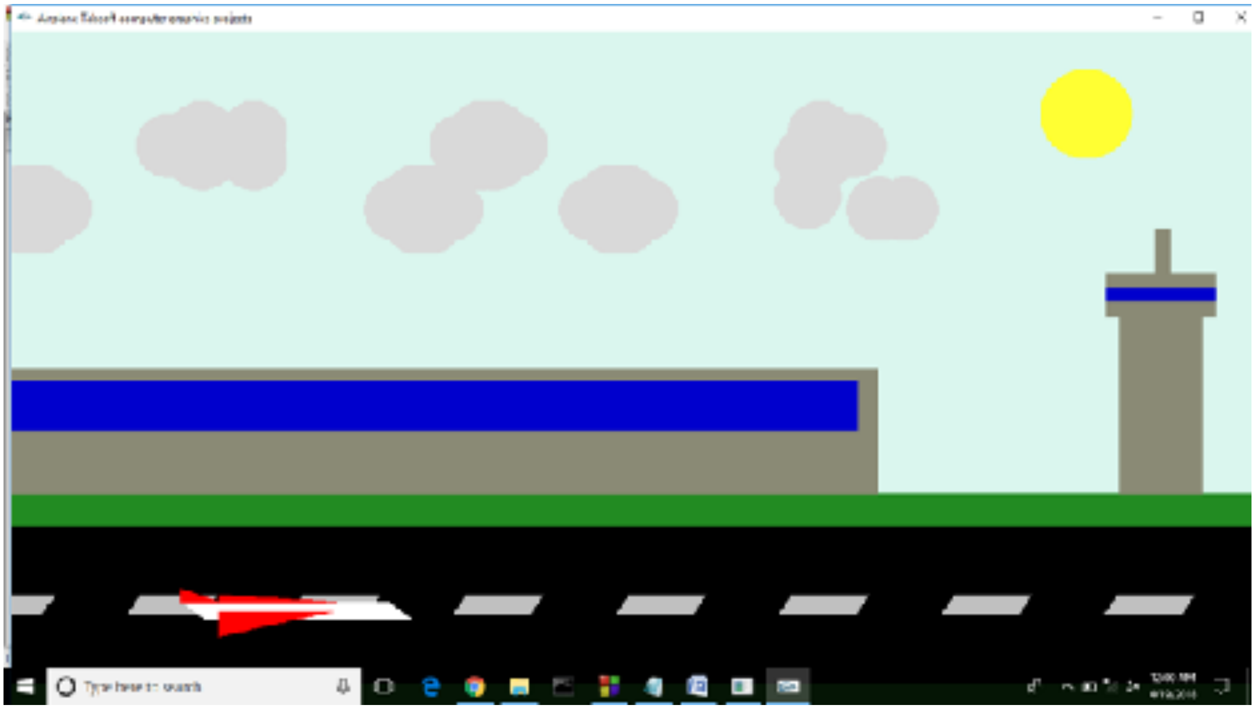
Animation is done using glutTimerfunc() and glutPostRedisplay() . glutTimerfunc() is callback function which is called forever after specific time(16 ms to get 60 fps) which updates motion parameters and is redisplayed using glutPostRedisplay() to get desired frame rate for animation.Double buffers are used to get non-flickery animation as it only draws after whole frame is computed until then it displays the old frame.

FUTURE WORKS

This project can be further developed as an interactive game in which the player has to fly his plane avoiding obstacles like building. We can also make this as a game in which there are enemy aircrafts firing at the player and the player has to dodge and fire back. We can also improve the graphics by making the objects 3D. We can add more background pictures to make the game more attractive.

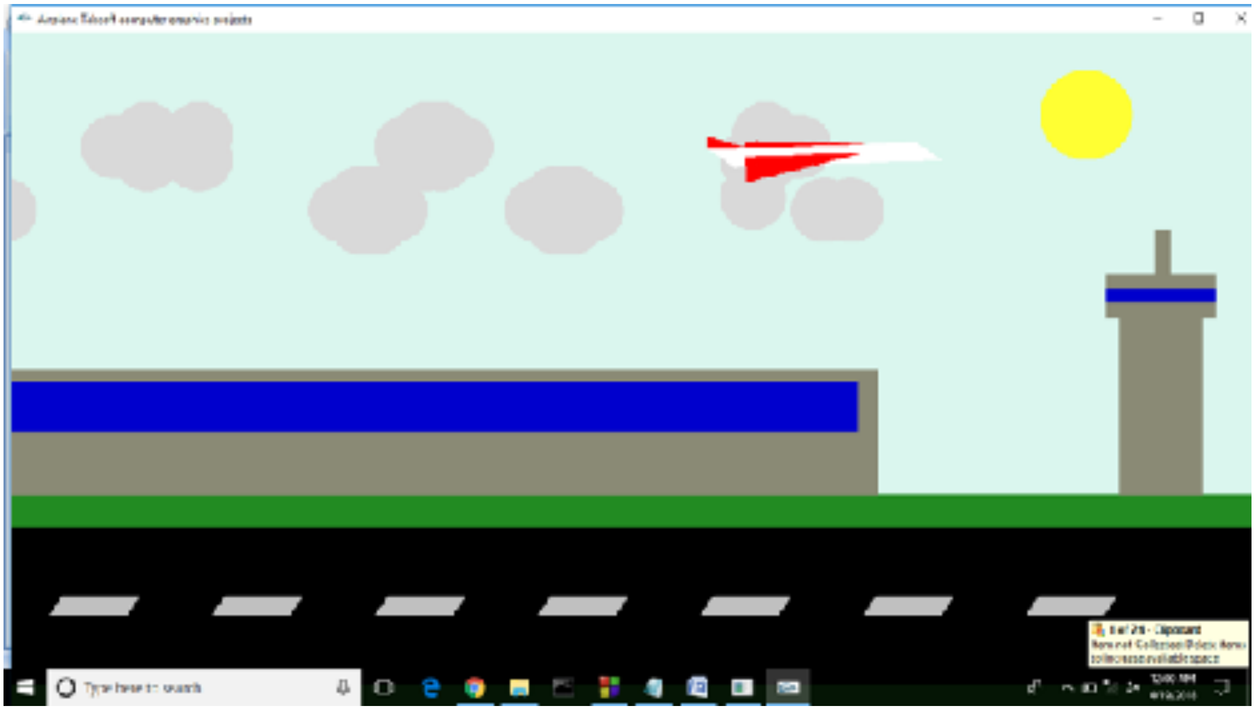
RESULTS AND DISCUSSION

The result of this project, is a visually appealing airplane graphic .There are many features to this result; all contributing to the final look and feel of the project. The most impactful features is animation. These features provide a sense of speed and logical movement . The other implementations are as important, but are hard to distinguish from each other in terms of effect, as they together contribute to the visual quality of the rendered image.

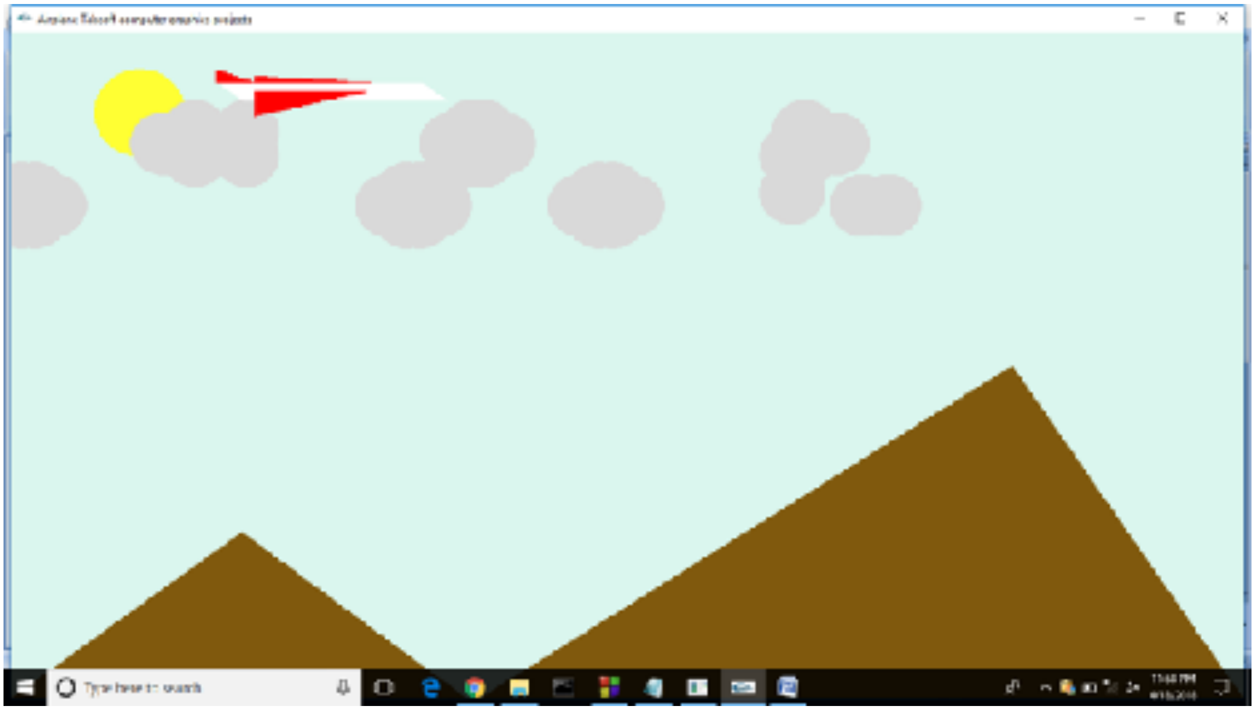


a) Airplane at the airport before take off

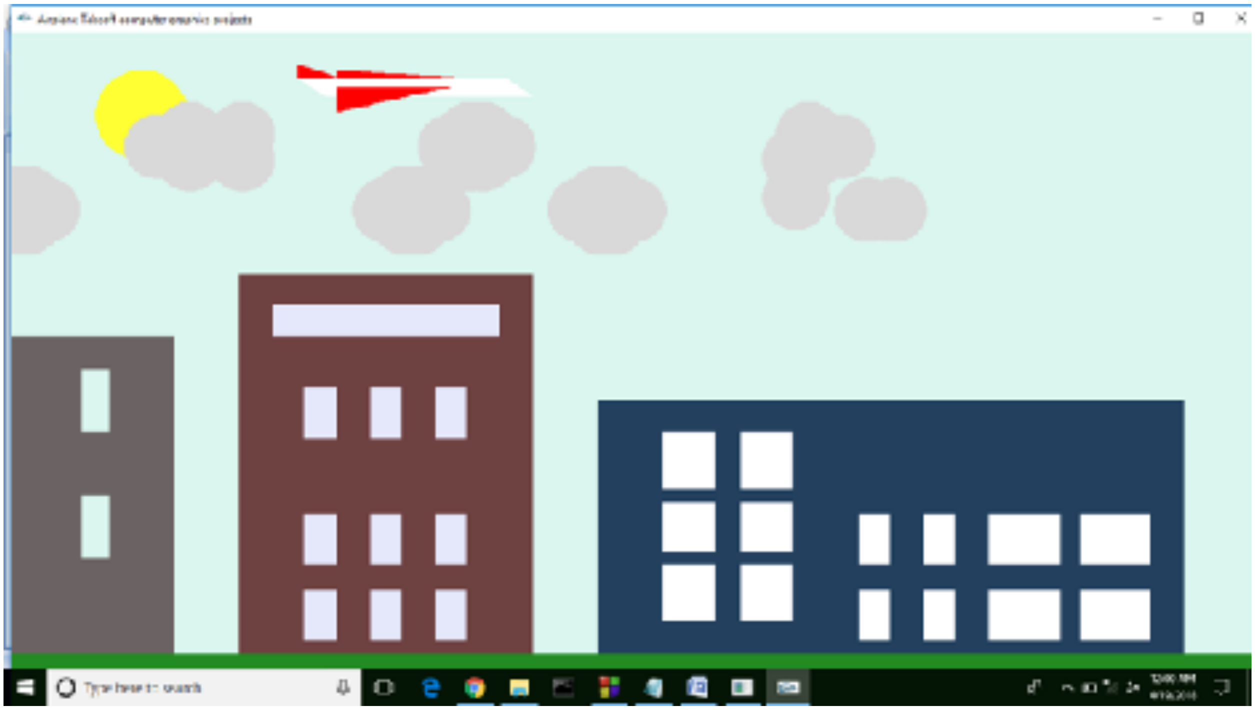
Type to enter text



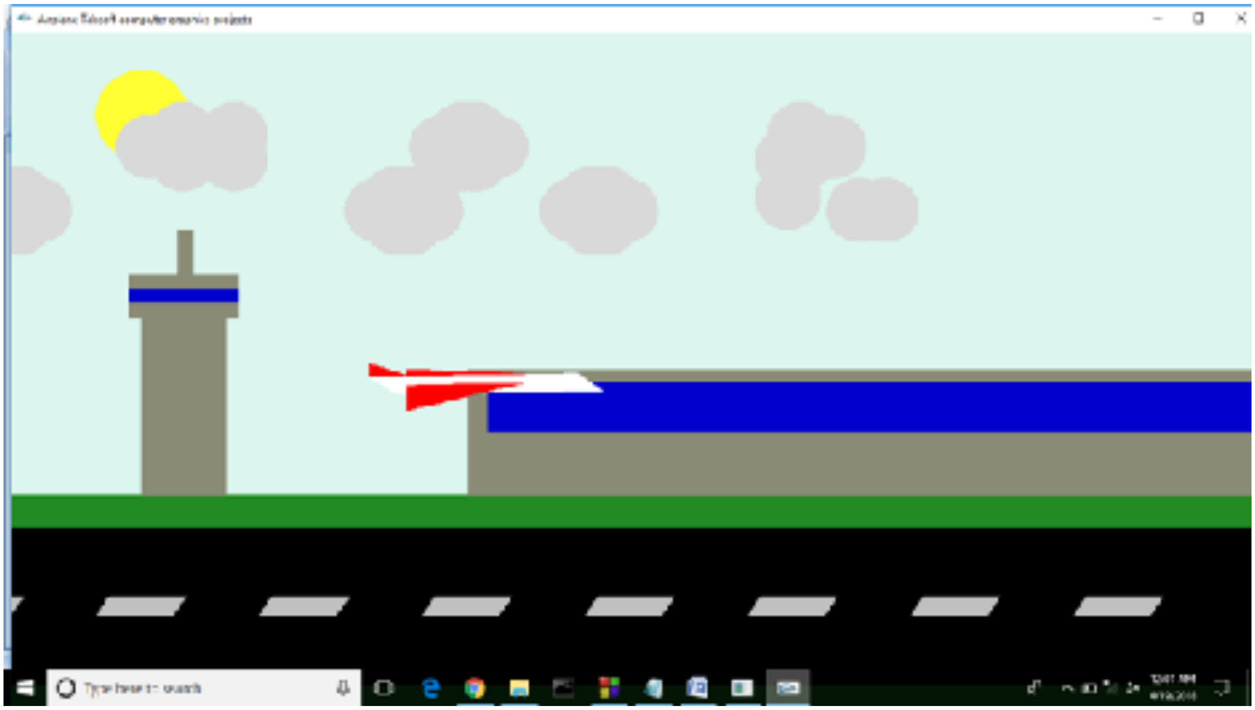
b) Airplane taking off from the airport



c) Airplane flying in hilly country side.



d) Airplane flying over a cityscape.



e)Airplane landing at its destination airport.

CONCLUSION

The resulting project is in our opinion well realized, in perspective to our set goals. Of course we would like to have added more content and features, but due to time constraints and team size we had to prioritize essential components that we thought would contribute the most to the project. Therefore, project development in a small and rather inexperienced team, over a short period of time is nothing we would recommend. However, if the project idea is clear, and if the development is done with respect to the physics library, rather than the opposite; there is still a chance of creating a visually pleasing, as well as an entertaining project. Probably the greatest insight of the development team, is that proper research is crucial, with respect to time, as well as the final result. For instance, by taking the time to analyze the various rendering algorithms, we might have ended up with a deferred renderer in an earlier stage of the project. Thus, adding more visual effects might have been possible. However, doing proper time estimation for the implementation is still considered difficult, due to lack of experience.

REFERENCES

- https://en.wikipedia.org/wiki/OpenGL
- Computer Graphics C Version by Donald Hearn & M Pauline Baker II Edition.
- https://www.khronos.org/opengl/wiki/OpenGL_Reference

CODE

https://github.com/abhilashvenkatesh/Opengl-2D-Airplane-animation