A REPORT ON PRACTICAL ASSIGNMENT OF COMPUTER GRAPHICS

Shape, arrow

Description automatically generated

Submitted by Sushil Gyawali

December 16, 2021

**Preface**

This report has been prepared as a part of my project APOLLO 11. The report is prepared with the view to include all the details regarding the project that I carried out.

Computer graphics remains one of the most exciting and rapidly growing computer fields. Computer graphics has now become a common element in user interfaces, data visualization, television commercial, motion pictures, and many-many other applications. With the advantages of graphics, modern subject, we can be able to design different application and tools which can be beneficial in our daily life. Besides, some entertaining applications can also be designed using Graphics such as Games, Photo Effect etc. I am trying to apply the Graphics on designing graphics application to launch rocket.

**Acknowledgements**

I would like to express my gratitude toward our respected **Computer Graphics** lecturer **Mr. Ramesh Kharbhuja** for assigning and encouraging us to do this project. He provided us an opportunity to grow our interest in the field of Computer Graphics and animation and by giving us a chance to learn more about this broad topic and make an eye-catching graphics application via this project.

While making this project I came to know many things about graphics which were not taught in syllabus due to time barrier. Similarly, I would like to acknowledge Bhaktapur Multiple Campus, CSIT department for providing and innovative way to learn about graphics and programming language to make our project look attractive and detailed.

**Contents**

**1 INTRODUCTION**

1.1 BACKGROUND

1.1.1 BACKGROUND OF THE PROJECT

1.1.2 BACKGROUND OF THE APOLLO11

1.2 OBJECTIVE OF THE STUDY

**2 IDEA OF PROJECT**

**3 PROGRAMMING LANGUAGE AND LIBRARY**

3.1 ABOUT C

3.2 ABOUT LIBRARY

**4 SPECIFICATIONS**

4.1 SOFTWARE REQUIREMENT

4.2 HARDWARE REQUIREMENT

**5 BUILDING BLOCKS**

**6 CONCLUSION AND RECOMMENDATION**

6.1 CONCLUSION

6.2 LIMITATION

6.3 FUTURE ENHANCEMENTS

**7 REFRENCE**

**8 Appendix**

**Chapter 1**

**INTRODUCTION**

**1.1 BACKGROUND**

**1.1.1 BACKGROUND OF THE PROJECT**

Visualization and animation help us to learn and understand clearly about any things or any process. In this project I had visualize the rocket launching steps via animation using graphics (Computer Graphics). The main motto of the project is to simulate the real-life launch and make the teaching method easier. It’s very difficult to learn theory-based launch and to solve this problem I had purposed this project. Background sound in the project has made the project more realistic. Through such visual and graphics, it would be fun and easier while teaching.

**1.1.2 BACKGROUND OF THE APOLLO 11**

Apollo 11 was the first manned mission to land on the Moon. The first steps by humans on another planetary body were taken by Neil Armstrong and Buzz Aldrin on July 20, 1969. The astronauts also returned to Earth the first samples from another planetary body. Apollo 11 achieved its primary mission - to perform a manned lunar landing and return the mission safely to Earth - and paved the way for the Apollo lunar landing missions to follow.

**1.2 OBJECTIVE OF THE PROJECT**

The main objectives of the study are as follows:

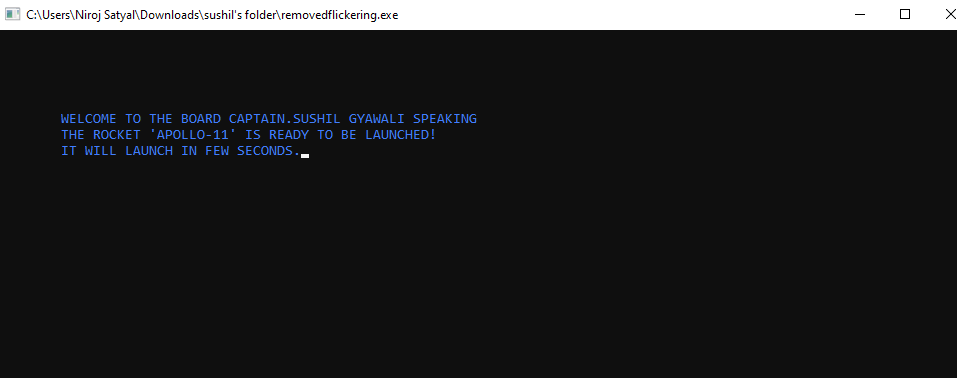
1. To understand and learn how to use tools related to computer graphics and to implement them to create projects.
2. To learn the graphics tools provided by c graphics library and understanding how to use them and process them in DevC++ IDE.
3. To visualize the rocket launching process.
4. To make teaching method easier through visual and animations.

**2 IDEA OF PROJECT**

In this project I have simulated the launching of rocket via visual and animation using library and built-ins. The idea is to simulate rocket launch, the countdowns, the rocket, the background sounds are part of it. when the code runs the terminal will open and show the specification of rocket. Another window will pop up and shows the countdown with background sound and after that the rocket will launch, after reaching certain height it will detach its payload.

Here many builtins function are used such as settextstyle(), setcolor(), cleardevice(), delay(), getmaxx(), getmaxy(), closegraph(), initwindow(), setcurrentwindow() and so on.

Below is the terminal with rocket specification.

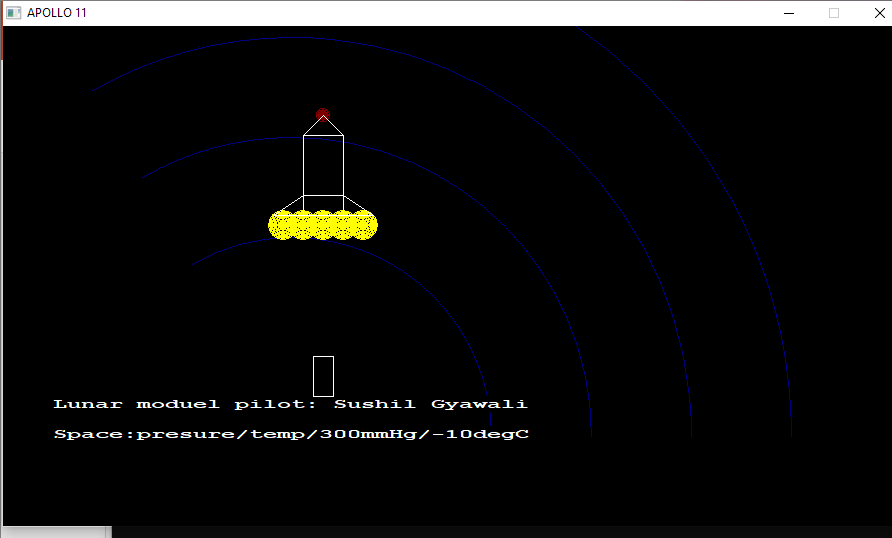


Here I have printed each word and letter in console by keeping delay of 50milisecond. The text is printed black using blue color value in set color function.

After that another windowBGI terminal will pop-up and start count down from 10 keeping the delay of thousand millisecond between that the sound will play using Playsound() function and clear device will clear current screen state and ready for next step. Below is the windowBGI terminal with countdown which current state is 6.



After countdown end the rocket will load and launch in the same windowBGI terminal.



The rocket boosters are kept yellow using setcolor. The arc function is used to create rings. Rocket is made using rectangle, circle, and line function. The texts are presented in the screen using outtextxy function. When the rocket reaches certain height (200) then it detaches its payload as shown in screen.

**Chapter 3**

**PROGRAMMING LANGUAGE AND LIBRARY**

**3.1 ABOUT C**

C is a procedural programming language. It was initially developed by Dennis Ritchie in the year 1972. It was mainly developed as a system programming language to write an operating system. The main features of the C language include low-level memory access, a simple set of keywords, and a clean style, these features make C language suitable for system programming like an operating system or compiler development.   
Many later languages have borrowed syntax/features directly or indirectly from the C language. Like syntax of Java, PHP, JavaScript, and many other languages are mainly based on the C language. C++ is nearly a superset of C language (Few programs may compile in C, but not in C++).

**3.2**  **ABOUT LIBRARY**

**Graphics.h**

C graphics using graphics.h functions or WinBGIM (Windows 7) can be used to draw different shapes, display text in different fonts, change colors and many more. Using functions of graphics.h in Turbo C compiler you can make graphics programs, animations, projects, and games. You can draw circles, lines, rectangles, bars and many other geometrical figures. You can change their colors using the available functions and fill them. Following is a list of functions of graphics.h header file. Every function is discussed with the arguments it needs, its description, possible errors while using that function and a sample C graphics program with its output.

**MMsystem.h**

MMSystem.h is a header file for Windows multimedia functions, introduced in Multimedia Windows 1.0 or Windows 3.1. The name states for 'multimedia system.' A related file is WinMM.h. It looks like you can find some information on the relevant API's and services in Windows Multimedia.

**Stdio.h**

Input and Output operations can also be performed in C++ using the **C** **St**andar**d** **I**nput and **O**utput Library (**cstdio**, known as stdio.h in the C language). This library uses what are called *streams* to operate with physical devices such as keyboards, printers, terminals or with any other type of files supported by the system. Streams are an abstraction to interact with these in an uniform way; All streams have similar properties independently of the individual characteristics of the physical media they are associated with. Streams are handled in the cstdio library as pointers to FILE objects. A pointer to a FILE object uniquely identifies a stream, and is used as a parameter in the operations involving that stream.  
  
There also exist three standard streams: stdin, stdout and stderr, which are automatically created and opened for all programs using the library.

**Stdlib.h**

**<stdlib.h>** is the header for the **General-Purpose Standard Library** of C programming language which declares a variety of utility functions for type conversions, memory allocation, process control including dynamic memory management, random number generation, communication with the environment, integer arithmetics, searching, sorting and converting. It also has multiple data types and macros defined in the header.

**CHAPTER 4**

**SPECIFICATIONS**

**4.1 SOFTWARE REQUIREMENT**

So as to run the project the software specification in not high. I used DevC++ with Mingw compiler for window 10 and some external graphics and multimedia system(MMsystem.h) library to link the header file. These software setups can be used in window 7 as well. Mine software setup is

Operating System (OS): Window 10(64 bit)

Text Editor: DevC++

External Library: Graphics.h library, MMsystem.h library

**5.2 HARDWARE REQUIREMENT**

Likewise, software there is no need of higher end hardware requirement or setup. The computer with minimum of 2gb ram, built in 2 Gb graphics card provided by intel or AMD Ryzen with following software setup can run this project efficiently without any issue. Mine Hardware setup is

Brand: Dell

Ram: 12Gb

Secondary storage: SSD (128GB), HDD(1Tb)

Graphics: 4Gb provided by Intel

No of core: 8

Intel: I5,8th generation 3000series

**Chapter 5**

**BUILDING BLOCKS**

**FUNCTION DESCRIPTIONS**

**initgraph()** :- This function initializes the graphics system. Prototype is defined in ‘graphics.h’. It is declared as per given below.

***“void far initgraph (int far \*gd , int far \*gm, char far \*path to driver);***

***gd (Graphics Driver)***

***gm (Graph Mode)”***

**Circle() *:-*** *This function Draw a circle at (x,y) of the given radius. This function is declared as under.void far circle****(int x, int y, int radius);*** *‘or’circle****(getmaxx, getmaxy, radius); .***

**getmaxx() :-** This function return maximum X coordinate on the screen. Its Prototype is defined in **‘graphics.h’** . And declared as under.***int far getmaxx (void);***

**getmaxy() :-** This function return maximum Y coordinate on the screen. Its Prototype is defined in **‘graphics.h’** . And declared as under.***int far getmaxy (void);***

**setcolor() :-** This function sets the current drawing color . This is declared as under.***void for set color (int color);***

**closegraph()** :-This function shuts down the graphics system. The declaration is as under. Prototype is defined in **‘graphics.h’.*void far closegraph(void); ‘or’ closegraph();***

**Arc() :-** “arc” function is used to draw an arc with center (x, y) and 1st angle specifies the starting angle, end angle specifies the end angle and the last parameter specifies the radius of the arc. **arc** function can also be used to draw a circle but for that starting angle and end angle should be 0 and 360 respectively.

Declaration :- void arc(int x, int y, int 1stangle, int endangle, int radius);

**setcolor( ):-**In Turbo C Graphics each color is assigned a number. A total of 16 **colors** are available. The number of available colors depend on the current graphics mode and driver.

For Example- BLACK is assigned **0**, and BLUE is assigned **9** ( Instead of color value, you can put their name in capital Latter) etc. setcolor function is used to change the current drawing color.

For Example- setcolor(BLUE) or setcolor(9) changes the current drawing color to RED. Remember that the default drawing color is WHITE.

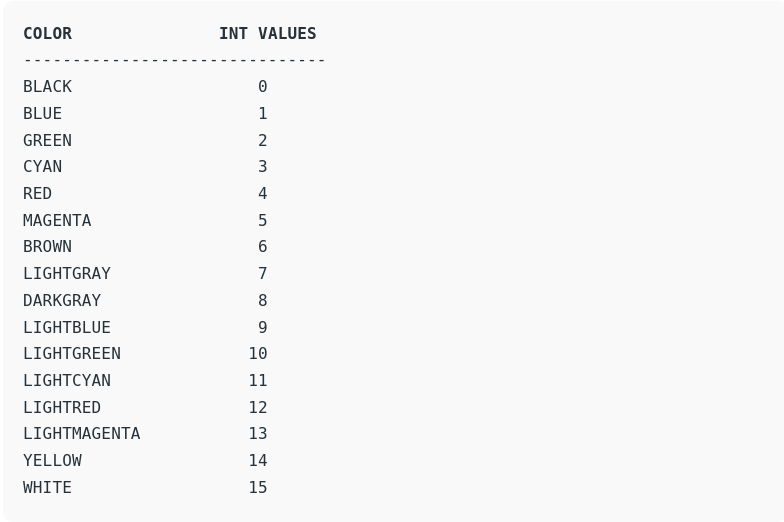
Declaration :- void setcolor(int color);

**Ellipse() :-** Ellipse is used to draw an ellipse (x,y) are coordinates of the center of the ellipse, stangle is the starting angle, end angle is the ending angle, and the fifth and sixth parameters specify the X and Y radius of the ellipse. To draw a complete ellipse strangles and end angle should be 0 and 360 respectively.

**Declaration:-** void ellipse(int x, int y, int stangle, int endangle, int xradius, int yradius);

**Setbkcolor() :-** setbkcolor function changes current background color e.g. setbkcolor(YELLLOW) changes the current background color to YELLOW.  
Remember that default drawing color is WHITE and background color is BLACK.  
Declaration:- setbkcolor(int color);

Below is the table showing INT VALUES corresponding to Colors :



**Delay():-** delay function is used to suspend execution of a program for a particular time.Here unsigned int is the number of milliseconds (remember 1 second = 1000 milliseconds). To use delay function in your program you should include the "dos.h" header file which is **not** a part of standard C library.

Declaration:- void delay(unsigned int);

**Settextstyle():-** The header file graphics.h contains **settextstyle()** function which is used to change the way in which text appears. Using it we can modify the size of text, change direction of text and change the font of text.

Declaration:- settextstyle(int font, int direction, int font\_size);

The table below shows the fonts with their INT values and appearance:



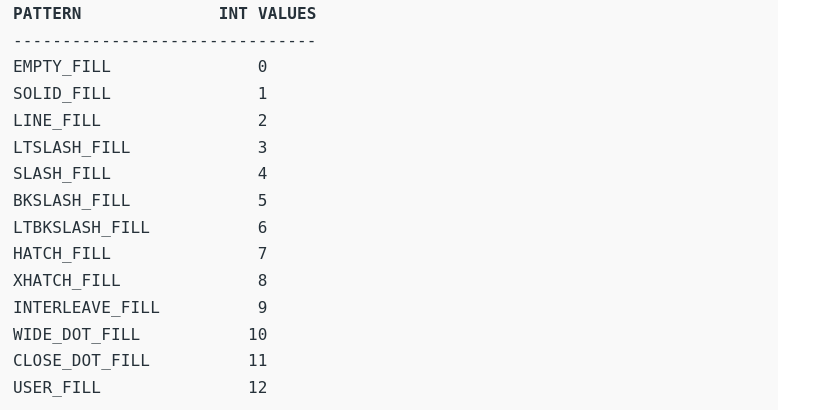
# **Setfillstyle() and floodfill():-** The header file graphics.h contains **setfillstyle()** function which sets the current fill pattern and fill color. **floodfill()** function is used to fill an enclosed area. Current fill pattern and fill color is used to fill the area.

Declaration:-

setfillstyle(int pattern, int color);

floodfill(int x, int y, int border\_color);

Below is the table showing INT VALUES corresponding to Patterns:



**Playsound():-** The PlaySound function plays a sound specified by the given file name, resource, or system event. (A system event may be associated with a sound in the registry or in the WIN.INI file.)

Declaration:- Playsound(“filename.wav”,NULL,SND\_ASYNC);

The sound is played asynchronously and **PlaySound** returns immediately after beginning the sound. To terminate an asynchronously played waveform sound, call **PlaySound** with *pszSound* set to **NULL**.

**Chapter 6**

**CONCLUSION AND RECOMMENDATION**

**6.1 CONCLUSION**

This project is one of the sample project on computer graphics. Though many difficulties were faced during the project as well as many error encountered, but I successfully compiled and ran the program. Well this project will be applicable to most of all. As well as I am very hopeful or more advice, new ideas and inspiration to make more other projects.

**6.2 LIMITATION**

Well, there are some limitations on this project that if it’s solved it would run very efficiently and the animation will be flicker less.

1. The flickering of window BGI screen is average.
2. The sound is not synchronizing properly with the rocket launching.
3. The detaching of payload from the rocket is not smooth.
4. There is no background (stars, cloud, launching pad) in the rocket animation.
5. The dimension of rocket is average and not much smooth.

**6.3 FUTURE ENHANCEMENTS**

As mentioned, the rocket dimension is average. In future, we can use OpenGL and GLUT and use sphere, cylinder and polygons to make rocket look more realistic. Using external library, the rocket launching sound could be synchronized with the rocket flying. The detachment of pay load can be make smoother giving more time in graphics.h library. The first improvement that I could do in future is to modify the rocket shape to make its more realistic.

**Chapter 7**

**Reference**

**[1] https://www.programmingsimplified.com/c-graphics-programming-tutorial**

**[2] https://www.geeksforgeeks.org/computer-graphics-2/**

**[3] https://developerinsider.co/graphics-graphics-h-c-programming/**

**[4] https://en.wikipedia.org/wiki/Computer\_graphics\_**

**[5] https://en.wikipedia.org/wiki/Apollo\_11**

**[6] some from project done in 1st sem using graphics.h library**