SAGARMATHA ENGINEERING COLLEGE

SANEPA, LALITPUR

A REPORT ON

COMPUTER GRAPHICS PROJECT: ROCKET LAUNCH

SUBMITTED BY:

BINAYAK PRADHAN (077BCT016)

SWORNIMA SHRESTHA (077BCT046)

YUDHIR ACHARYA (077BCT047)

SAUHARDA KHADKA (077BCT036)

SUBMITTED TO:

Department of Electronics

and Computer Engineering

ACKNOWLEDGEMENT

We would like to extend our sincere gratitude to **Mr. Shankar Bhandari** sir and **Mr. Sujan Pokhrel** sir for their invaluable guidance and support throughout the duration of this computer graphics project. Their mentorship and expertise have been instrumental in shaping our project and making it a success.

Their passion for computer graphics and their dedication to teaching have inspired us to delve deeper into this fascinating subject. Their patient explanations, insightful feedback, and willingness to address our doubts have been crucial in our learning process. We are grateful for the opportunity to work on this project as a group under their supervision. Their encouragement and belief in our abilities have motivated us to put in our best efforts. Furthermore, we would like to express our thanks to the entire faculty for providing a conducive learning environment and valuable resources that aided our project development.

Finally, we would like to acknowledge the collaboration and support of our group members. Each team member's contributions and teamwork have been vital in successfully completing this project.

TABLE OF CONTENTS

INTRODUCTION

The rocket launching graphics project is an exciting endeavor aimed at creating a visually captivating simulation of a rocket launch. Inspired by our interest in computer graphics and animation, we set out to develop an interactive and educational animation that brings the thrill of space exploration to life. Rocket launches are of paramount importance, encompassing a wide range of critical applications and contributions to humanity. At the forefront is space exploration, as rockets enable the deployment of spacecraft, satellites, and telescopes, empowering us to study celestial bodies and unravel the mysteries of the cosmos. These missions offer profound insights into our place in the universe and the potential for life beyond Earth.

The significance of rocket launches extends to practical applications as well. The robust network of communication satellites orbiting Earth, made possible by rockets, underpins modern telecommunications, enabling global connectivity and real-time information exchange. Additionally, military and reconnaissance satellites deployed by rockets bolster national security, providing vital intelligence and surveillance capabilities.

OBJECTIVE AND SCOPE

The main objective of this project is to design and implement a rocket launching graphics animation that showcases the key stages of a rocket launch, from liftoff to reaching the upper atmosphere. Through this project, we aim to gain practical experience in graphics programming and animation techniques while creating an engaging and informative simulation.

In this project, we will focus on developing a visually appealing rocket launching animation using the Turbo C/C++ graphics library. We will explore various graphics functions and animation concepts to illustrate the rocket's ascent, the ignition of booster stages, and the subsequent separation of components.

IMPORTANCE AND APPLICATIONS

The rocket launching graphics animation has significant educational value, as it provides a simplified representation of a complex engineering process. It can be used as a tool for educational institutions to demonstrate the principles of rocket science and space exploration. Additionally, the animation has entertainment value and can captivate audiences interested in space missions and scientific simulations. In this report, we will detail the design and implementation of the rocket launching animation. We will explain the techniques used to create the captivating fireworks animation during liftoff and integrate sound effects to enhance the user experience. Furthermore, we will present the results of our project, along with reflections on the development process and potential areas for future enhancement.

METHODOLOGY

CONCEPTUALIZATION:

Program requirements were conceptualized based on typical rocket launch visuals in media and simulations. The visual design incorporated basic shapes like lines, rectangles and circles to construct the rocket graphics. Color gradients were planned to illustrate sky, fire, and smoke effects. Timing loops would animate ascent and refresh the scene.

DESIGN:

The rocket launch animation program was envisioned to demonstrate fundamental 2D graphics concepts and skills in C using the graphics.h library.

Key design goals included:

- Showcasing coordinate drawing of shapes

- Implementing screen refresh and timing

- Animating basic motion and visual effects

- Layering visual elements into a scene

Based on common rocket launch media, requirements were defined for elements like countdown, liftoff, smoke, fire, and background. Modular code organization was planned to separate rocket drawing, motion loops, and effects. Storyboards of the sequence guided development.

COMPILER:

We used TurboC++ for building this project. During the 1990s, turbo C++ gained immense popularity and was widely used for programming on the dos platform. Many graphics programming tutorials and resources were available for turbo C++. TurboC++ comes with built in graphics mechanisms, so it fast and easy to implement the program in this compiler.

HEADER FUNCTIONS

- graphics.h

  This header provided a simple way to draw basic graphics and create graphical applications on the MS-DOS platform. The **graphics.h** header file was part of the BGI (Borland Graphics Interface) library, which was developed by Borland for use with their TurboC++ compiler. **line**, **rectangle()**, **circle()**, **ellipse()**, **arc()**, **polygon()** etc., were functions used to draw basic shapes on the screen.

- dos.h

  It provided access to various dos-specific functions and system calls, allowing programmers to interact with the operating system, perform low-level tasks, and access hardware resources. Some functions like **delay**() function to introduce a delay in the program execution and **exit**() function to terminate the program and return control to the operating system are used widely.

- conio.h

  "conio" stands for "console input/output." like **dos.h**, **conio.h** was specific to MS-DOS and was widely used during the 1980s and early 1990s for text-based user interfaces and simple console-based applications. Functions such as **clrscr**() function to clear the console screen, **getch**() function to read a single character from the keyboard without displaying it on the screen is used in compiler.

- stdio.h

  stdio.h is a standard header file in C programming languages that stands for "Standard input/output." it provides functions and macros for performing input and output operations on streams, including console input/output and file input/output. Functions like **printf()** function for formatted output to the console or files and **scanf**() functions for formatted input from the console or files are used for input and output functions.

- math.h

  This header file is part of the c standard library (for C) and the C++ standard library (for C++), and it provides various mathematical functions and constants that you can use in your programs.

FUNCTIONS USED

“graphics.h” library is used to include and facilitate graphical operations in program. “graphics.h” functions can be used to draw different shapes, display text in different fonts, change colors and many more. Using functions of “graphics.h” 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.

Some of the function we used in this project are:

1. Initialization modes:
    - **initgraph()**: function to initialize the graphics system and set the graphics mode (e.g., cga, ega, vga).
2. Drawing functions:
    - **line()**, **rectangle()**, **circle()**, **ellipse()**: functions to draw basic shapes on the screen.
    - **arc()**, **floodfill()**: functions for drawing arcs and filling areas with colors.
3. Text output on screen:
    - **outtext()**, **outtextxy()**: functions to display text on the screen at specific positions.
4. Color and fill:
    - **setcolor()**, **setbkcolor()**: functions to set the drawing and background colors.
    - **setfillstyle()**: function for filling shapes with patterns and solid colors.
5. User input:
    - **getch()**: function to handle keyboard input.
6. Other functions:
    - **delay()**: function to introduce a delay in the program execution.

CODE and OUTPUT

i) Code

```
#include <graphics.h>
#include <dos.h>
#include <conio.h>
#include <string.h>
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

int main()
 {
      int gdriver = DETECT, gmode;
      initgraph(&gdriver, &gmode, "C:\\TurboC3\\BGI");
      int choice;
int style,midx,midy;
midx = getmaxx() / 2;
midy = getmaxy() / 2;
sound(400);
delay(50);
nosound();
     for(int i=0;i<50;i++){
        settextstyle(3,HORIZ_DIR,4);
        outtextxy(80,150,"A rocket is going to launch in...");
        settextstyle(3,HORIZ_DIR,10);
        outtextxy(250,200,"3");
        setcolor(RED);
        delay(50);
        cleardevice();
    }
    sound(500);
    delay(50);
    nosound();

 for(int j=0;j<30;j++){
        settextstyle(3,HORIZ_DIR,4);
        outtextxy(80,150,"A rocket is going to launch in...");
        settextstyle(3,HORIZ_DIR,10);
        outtextxy(250,200,"2");
        setcolor(YELLOW);
        delay(70);
        cleardevice();
    }
    sound(600);
    delay(50);
    nosound();
for(int k=0;k<20;k++){
        settextstyle(3,HORIZ_DIR,4);
        outtextxy(80,150,"A rocket is going to launch in...");
        settextstyle(3,HORIZ_DIR,10);
        outtextxy(250,200,"1");
        setcolor(GREEN);
        delay(70);
        cleardevice();
    }
do{
     sound(1000);
    delay(80);
    sound(2000);
    delay(120);
    sound(3000);
    delay(150);
    nosound();

      cleardevice();
      setcolor(LIGHTGREEN);
      settextjustify(CENTER_TEXT, CENTER_TEXT);
      style = DEFAULT_FONT;
      settextstyle(style, HORIZ_DIR, 8);
      outtextxy(midx, midy,"GO!!!");
      delay(1800);
      cleardevice();
      for (i=0; i<400; i++) {
            setcolor(WHITE);
            line(300,300-i,350,300-i);  // upper horizon
            line(300,300-i,300,400-i);  // left vertical
            line(270,400-i,380,400-i);  // bottom horizon
            line(350,300-i,350,400-i);  // right vertical
            line(300,350-i,270,400-i);  // left slant
            line(350,350-i,380,400-i);  // right slant
            line(300,300-i,325,260-i);  // top left slant
            line(325,260-i,350,300-i);  // top right slant
      // LAUNCHING SMOKE
            if (i==0) {
                  delay(1000);
                  for ( int k=0; k<70; k++) {
                        delay(100);
                        setcolor(DARKGRAY);
                        circle(250,400,k);
                        circle(270,400,k);
                        setcolor(LIGHTGRAY);
                        circle(285,400,k+10);
                        circle(318,400,k+30);
                        circle(335,400,k+30);
                        circle(370,400,k+10);
                        setcolor(DARKGRAY);
                        circle(390,400,k);
                        circle(410,400,k);
                  }
            }
```

```
//stars
    circle(123, -456+i, 1);
    circle(256, -278+i, 1);
    circle(640, -390+i, 1);
    circle(512, -100+i, 1);
    circle(380, -290+i, 1);
    circle(312, -239+i, 1);
    circle(200, -160+i, 1);
    circle(480, -321+i, 1);
    circle(410, -70+i, 1);
    circle(50, -140+i, 1);
    circle(123, 344+i, 1);
    circle(256, 122+i, 1);
    circle(640, 10+i, 1);
    circle(512, 300+i, 1);
    circle(380, 110+i, 1);
    circle(312, 161+i, 1);
    circle(200, 240+i, 1);
    circle(480, 79+i, 1);
    circle(410, 300+i, 1);
    circle(50, 270+i, 1);
    circle(345, 200+i, 1);
    circle(520, 230+i, 1);
    circle(176, 301+i, 1);
    circle(290, 190+i, 1);
    circle(640, 10+i, 1);
    circle(30, 390+i, 1);
    circle(256, 122+i, 1);
    circle(500, 220+i, 1);
    circle(400, 250+i, 1);
    circle(112, 201+i, 1);
    circle(640, 10+i, 1);
    circle(510, 90+i, 1);
    circle(590, 210+i, 1);
    circle(320, 270+i, 1);
    circle(260, 50+i, 1);
    circle(390, 90+i, 1);
    circle(500, 50+i, 1);
    circle(150, 320+i, 1);
    circle(450, 50+i, 1);
    circle(640, 10+i, 1);
    circle(470, 100+i, 1);
    circle(380, 250+i, 1);
    circle(190, 110+i, 1);
    circle(296, 251+i, 1);
    circle(390, 290+i, 1);
    circle(120, 390+i, 1);
    circle(310, 340+i, 1);
    circle(250, 270+i, 1);
    circle(90, 260+i, 1);
    circle(410, 90+i, 1);
    circle(640, 10+i, 1);
    circle(160, 210+i, 1);
    circle(420, 50+i, 1);
    circle(350, 260+i, 1);
    circle(480, 210+i, 1);
    circle(256, 122+i, 1);
    circle(200, 40+i, 1);
    circle(312, 250+i, 1);
    circle(176, 350+i, 1);
    circle(290, 50+i, 1);
    circle(150, -80+i, 1);
    circle(450, -160+i, 1);
    circle(640, -390+i, 1);
    circle(470, -290+i, 1);
    circle(380, -90+i, 1);
    circle(190, -240+i, 1);
    circle(296, -296+i, 1);
    circle(390, -320+i, 1);
    circle(120, -100+i, 1);
    circle(310, -280+i, 1);
    circle(250, -390+i, 1);
    circle(90, -130+i, 1);
    circle(410, -10+i, 1);
    circle(640, -390+i, 1);
    circle(160, -320+i, 1);
    circle(420, -190+i, 1);
    circle(350, -260+i, 1);
    circle(480, -310+i, 1);
    circle(256, -278+i, 1);
    circle(200, -100+i, 1);
    circle(312, -250+i, 1);
    circle(176, -150+i, 1);
    circle(290, -350+i, 1);
    circle(640, -390+i, 1);
    circle(30, -210+i, 1);
    circle(250, -270+i, 1);
    circle(390, -390+i, 1);
    circle(112, -130+i, 1);
    circle(640, -390+i, 1);
    circle(510, -250+i, 1);
    circle(590, -310+i, 1);
    circle(320, -90+i, 1);
    circle(260, -140+i, 1);
    circle(390, -120+i, 1);
    circle(500, -100+i, 1);
    circle(150, -200+i, 1);
    circle(450, -90+i, 1);
    circle(640, -390+i, 1);
    circle(470, -120+i, 1);
    circle(380, -180+i, 1);
    circle(190, -90+i, 1);
    circle(296, -250+i, 1);
    circle(390, -210+i, 1);
    circle(120, -110+i, 1);
    circle(310, -160+i, 1);
    circle(250, -230+i, 1);
    circle(90, -240+i, 1);
    circle(410, -310+i, 1);
    circle(640, -390+i, 1);
    circle(160, -180+i, 1);
    circle(420, -70+i, 1);
    circle(350, -130+i, 1);
    circle(480, -160+i, 1);
    circle(256, -78+i,1);
    circle(320, 20+i, 1);
    circle(280, 70+i, 1);
    circle(550, -10+i, 1);
    circle(110, 60+i, 1);
    circle(490, -20+i, 1);
    circle(430, -30+i, 1);
    circle(240, 50+i, 1);
    circle(600, -30+i, 1);
    circle(370, 40+i, 1);
    circle(180, 30+i, 1);
    circle(520, -40+i, 1);
    circle(330, -20+i, 1);
    circle(280, -50+i, 1);
    circle(360, -40+i, 1);
    circle(400, -30+i, 1);
    circle(560, -50+i, 1);
    circle(60, 60+i, 1);
    circle(470, -10+i, 1);
    circle(110, 80+i, 1);
    circle(390, -30+i, 1);
    circle(130, 60+i, 1);
    circle(200, 80+i, 1);
    circle(340, 20+i, 1);
    circle(70, 70+i, 1);
    circle(330, -20+i, 1);
    circle(260, 60+i, 1);
    circle(130, -10+i, 1);
    circle(390, -20+i, 1);
    circle(540, -30+i, 1);
    circle(200, 20+i, 1);
    circle(300, 10+i, 1);
    circle(410, -20+i, 1);
    circle(250, 80+i, 1);
    circle(160, 40+i, 1);
    circle(360, 60+i, 1);
    circle(290, 0+i, 1);
    circle(460, 20+i, 1);
    circle(190, 60+i, 1);
    circle(250, -10+i, 1);
    circle(520, -40+i, 1);
    circle(420, 30+i, 1);
    circle(350, 0+i, 1);
    circle(450, -50+i, 1);
    circle(120, 30+i, 1);
```

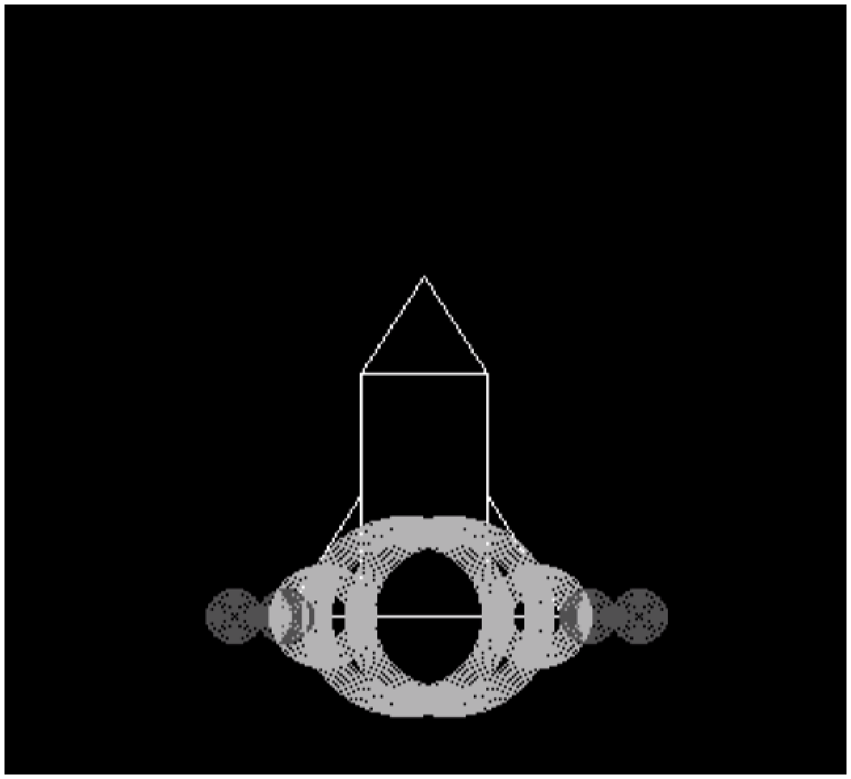
```
    circle(550, -30+i, 1);
    circle(90, 0+i, 1);
    circle(330, 30+i, 1);
    circle(400, 60+i, 1);
    circle(550, 40+i, 1);
    circle(150, 50+i, 1);

            //earth
            setcolor(BLUE);
            circle(250,300+i,400-i);
            circle(250,300+i,399-i);
            circle(250,300+i,398-i);
            circle(250,300+i,397-i);
            circle(250,300+i,396-i);
            circle(250,300+i,395-i);
            circle(250,300+i,394-i);
            circle(250,300+i,393-i);
            circle(250,300+i,392-i);
            circle(250,300+i,391-i);
            circle(250,300+i,390-i);
            circle(250,300+i,389-i);
            circle(250,300+i,388-i);

            setcolor(CYAN);
            circle(249,300+i,400-i);
            circle(249,300+i,399-i);
            circle(249,300+i,398-i);
            circle(249,300+i,397-i);
            circle(249,300+i,396-i);
            circle(249,300+i,395-i);
            circle(249,300+i,394-i);
            circle(249,300+i,393-i);
            circle(249,300+i,392-i);
            circle(249,300+i,391-i);
            circle(249,300+i,390-i);
            circle(249,300+i,389-i);
            circle(249,300+i,388-i);

//moon
setcolor(WHITE);
circle(0, -200+i, 120);
circle(0, -200+i, 120);
circle(0, -200+i, 120);
circle(0, -200+i, 120);
circle(0, -200+i, 120);
circle(0, -200+i, 120);
circle(0, -200+i, 120);
circle(0, -200+i, 120);
circle(0, -200+i, 120);
circle(0, -200+i, 120);
circle(0, -200+i, 120);
circle(0, -200+i, 120);
circle(0, -200+i, 120);
```

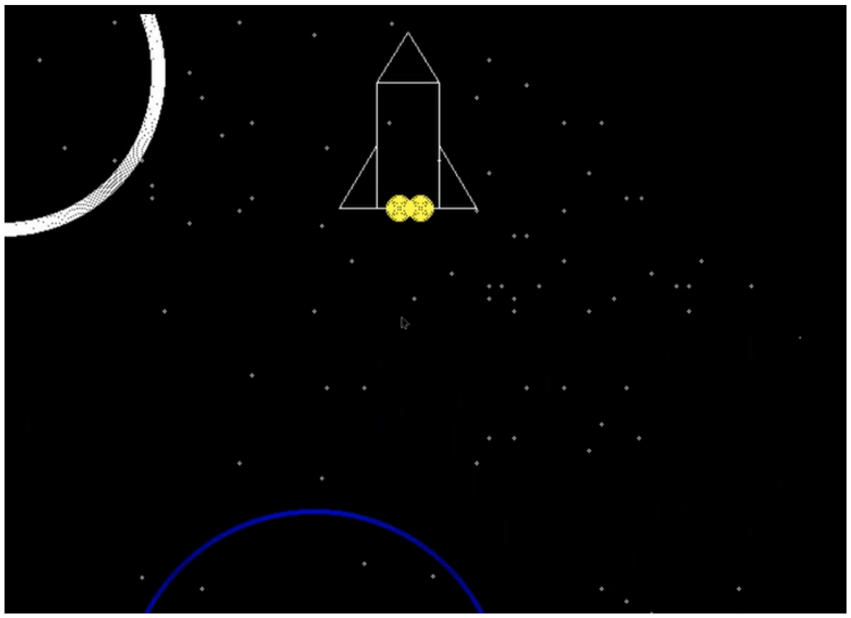
```
    circle(0, -200+i, 121);
    circle(0, -200+i, 121);
    circle(0, -200+i, 121);
    circle(0, -200+i, 121);
    circle(0, -200+i, 121);
    circle(0, -200+i, 121);
    circle(0, -200+i, 121);
    circle(0, -200+i, 121);
    circle(0, -200+i, 121);
    circle(0, -200+i, 121);
    circle(0, -200+i, 121);
    circle(0, -200+i, 121);
    circle(0, -200+i, 121);

    circle(0, -200+i, 122);
    circle(0, -200+i, 122);
    circle(0, -200+i, 122);
    circle(0, -200+i, 122);
    circle(0, -200+i, 122);
    circle(0, -200+i, 122);
    circle(0, -200+i, 122);
    circle(0, -200+i, 122);
    circle(0, -200+i, 122);
    circle(0, -200+i, 122);
    circle(0, -200+i, 122);
    circle(0, -200+i, 122);
    circle(0, -200+i, 122);

    circle(0, -200+i, 123);
    circle(0, -200+i, 123);
    circle(0, -200+i, 123);
    circle(0, -200+i, 123);
    circle(0, -200+i, 123);
    circle(0, -200+i, 123);
    circle(0, -200+i, 123);
    circle(0, -200+i, 123);
    circle(0, -200+i, 123);
    circle(0, -200+i, 123);
    circle(0, -200+i, 123);
    circle(0, -200+i, 123);
    circle(0, -200+i, 123);

    circle(0, -200+i, 124);
    circle(0, -200+i, 124);
    circle(0, -200+i, 124);
    circle(0, -200+i, 124);
    circle(0, -200+i, 124);
    circle(0, -200+i, 124);
    circle(0, -200+i, 124);
    circle(0, -200+i, 124);
    circle(0, -200+i, 124);
    circle(0, -200+i, 124);
    circle(0, -200+i, 124);
    circle(0, -200+i, 124);
    circle(0, -200+i, 124);

    circle(0, -200+i, 125);
    circle(0, -200+i, 125);
    circle(0, -200+i, 125);
    circle(0, -200+i, 125);
    circle(0, -200+i, 125);
    circle(0, -200+i, 125);
    circle(0, -200+i, 125);
    circle(0, -200+i, 125);
    circle(0, -200+i, 125);
    circle(0, -200+i, 125);
    circle(0, -200+i, 125);
    circle(0, -200+i, 125);
    circle(0, -200+i, 125);

    circle(0, -200+i, 126);
    circle(0, -200+i, 126);
    circle(0, -200+i, 126);
    circle(0, -200+i, 126);
    circle(0, -200+i, 126);
    circle(0, -200+i, 126);
    circle(0, -200+i, 126);
    circle(0, -200+i, 126);
    circle(0, -200+i, 126);
    circle(0, -200+i, 126);
    circle(0, -200+i, 126);
    circle(0, -200+i, 126);
    circle(0, -200+i, 126);

    circle(0, -200+i, 127);
    circle(0, -200+i, 127);
    circle(0, -200+i, 127);
    circle(0, -200+i, 127);
    circle(0, -200+i, 127);
    circle(0, -200+i, 127);
    circle(0, -200+i, 127);
    circle(0, -200+i, 127);
    circle(0, -200+i, 127);
    circle(0, -200+i, 127);
    circle(0, -200+i, 127);
    circle(0, -200+i, 127);
    circle(0, -200+i, 127);

    circle(0, -200+i, 128);
    circle(0, -200+i, 128);
    circle(0, -200+i, 128);
    circle(0, -200+i, 128);
    circle(0, -200+i, 128);
    circle(0, -200+i, 128);
    circle(0, -200+i, 128);
```

```
circle(0, -200+i, 128);
circle(0, -200+i, 128);
circle(0, -200+i, 128);
circle(0, -200+i, 128);
circle(0, -200+i, 128);
circle(0, -200+i, 128);

circle(0, -200+i, 129);
circle(0, -200+i, 129);
circle(0, -200+i, 129);
circle(0, -200+i, 129);
circle(0, -200+i, 129);
circle(0, -200+i, 129);
circle(0, -200+i, 129);
circle(0, -200+i, 129);
circle(0, -200+i, 129);
circle(0, -200+i, 129);
circle(0, -200+i, 129);
circle(0, -200+i, 129);
circle(0, -200+i, 129);

circle(0, -200+i, 130);
circle(0, -200+i, 130);
circle(0, -200+i, 130);
circle(0, -200+i, 130);
circle(0, -200+i, 130);
circle(0, -200+i, 130);
circle(0, -200+i, 130);
circle(0, -200+i, 130);
circle(0, -200+i, 130);
circle(0, -200+i, 130);
circle(0, -200+i, 130);
circle(0, -200+i, 130);
circle(0, -200+i, 130);

      if (i%2==0) {
         for ( int v=10; v>=0; v--) {
            setcolor(YELLOW);
            circle(318,400-i,v);
            circle(335,400-i,v);
         }
         setcolor(LIGHTRED);
         circle(325,260-i,5);
         circle(325,260-i,4);
         circle(325,260-i,3);
         circle(325,260-i,2);
         circle(325,260-i,1);
      }
      else {
         setcolor(YELLOW);
         for ( int w=10; w>=0; w--) {
            circle(300,400-i,w);
            circle(325,400-i,w);
```

```
                        circle(350,400-i,w);
                  }
            }
            delay(5);
            cleardevice();
      }
        for(i=0;i<30;i++){

      settextstyle(8,HORIZ_DIR,9);
       outtextxy(350,150,"Thank you!");
      delay(50);
      cleardevice();
    }

        // Set word durations
  int thankDuration = 200;
  int youDuration = 500;

  // Set tone frequencies
  int thankFreq = 900;
  int youFreq = 700;

  // Play "Thank"
  sound(thankFreq);
  delay(thankDuration);

  // Brief pause between words
  delay(100);

  // Play "You"
  sound(youFreq);
  delay(youDuration);

  // End sound
  nosound();
cleardevice();
printf("Enter 1 to restart simulation and 0 to exit:");
scanf("%d",&choice);
if(choice==0){
cleardevice();
exit(0);
}
}while(choice==1);
      getch();
      closegraph();
      return 0;
}
```

ii) Output





REFERENCES

https://www.techcrashcourse.com/2015/08/c-program-draw-circle-graphics.html

https://home.cs.colorado.edu/~main/bgi/doc/

https://stackoverflow.com/questions/64537330/how-to-play-sound-effect-or-music-in-c

INDEX