Report of Mini-project On

Rising Sun

Submitted in partial fulfilment of the requirements of the Mini project in the Computer Graphics Lab of
Semester III, Second Year Computer Science and Engineering [Data Science]

By

Ashwin Pawar
Arpan Mahadik
Mahek Shah
Mukta Zore

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Computer Science and Engineering [Data Science]

(A.Y. 2023-24)

CERTIFICATE

This is to certify that the Mini Project entitled **“Rising Sun”** is submitted by **Ashwin Pawar, Arpan Mahadik, Mahek Shah, Mukta Zore** for the subject of Computer Graphics Lab in the Department of Computer Science and Engineering (Data Science) as a record of work done by him/her under our supervision and guidance.

Internal Examiner

Guide
Asst. Prof. Odilia Gonsalves

Contents

1. Introduction

1.1 Basic idea of the project

The animation consists of a stick figure representing a man, a cloud and simulated raindrops. Overall, the project demonstrates a simple graphical animation in which a stick figured man walks under a cloud with simulated rain. This C program utilizes a graphics library,(likely Turbo C,s BGI), to create a basic animated scene. The animation's execution involves repeatedly clearing the screen, redrawing the stick figure and cloud, and generating random raindrop-like lines. The stick figure's walking motion is simulated by adjusting its position and the position of its legs within the loop. While relatively basic, this program offers a hands-on example of how to create graphical animations in C programming.

2. Computer Graphics Concepts

2.1 Graphic Functions used

Header and Graphics Initialization: The code begins by including necessary header files like <conio.h>, <stdio.h>, and <graphics.h>. It then initializes the graphics mode using the initgraph function.

Drawing Functions:

Header Files:

The program starts by including necessary header files: conio.h for console I/O functions like getch(). stdio.h for standard input/output functions. graphics.h for graphics functions used to create animations.

displayMan Function:
This function is responsible for drawing a stick figure representing a person. It includes components such as a face, neck, arms, body, and an umbrella.

drawCloud Function:

This function draws a cloud-like shape on the screen using arcs. It's used to create a cloud in the animation.

main Function:

The main function is the entry point of the program.
It initializes the graphics mode using initgraph with a specified graphics driver and mode.

cleardevice: Clears the screen in each iteration to prepare for redrawing.

displayMan(x, 340): Draws the man on the screen at coordinates (x, 340). **drawCloud(z, 60)**: Draws clouds on the screen at coordinates (z, 60).

line(0, 430, 639, 430): Draws a straight line near the bottom of the screen, representing the ground.

Random Rain: It generates random raindrops by repeatedly drawing lines at random positions within certain constraints. These lines simulate rainfall.

Leg Animation: The program animates the man's legs to make it appear as if he is walking.

It uses a variable d to control the position of the legs.

shouldMove is used to determine whether the legs should move forward or backward.

The legs are drawn as lines, and their position changes as d increases or decreases. **delay(200)**: Introduces a delay of 200 milliseconds to control the animation speed. **getch**: Waits for a keypress before exiting the program.

initgraph :(&gdriver, &gmode, "C:\\tc\\BGI"): This function initializes the graphics system with the specified driver, mode, and path to the BGI driver file. It sets up the graphics environment for drawing.

putpixel :(x, y, color): This function is used to set a single pixel at the specified coordinates (x, y) with the specified color. In this code, it's used to draw individual pixels in yellow to create the circle.

closegraph(): This function closes the graphics system and deallocates any resources used by the graphics functions

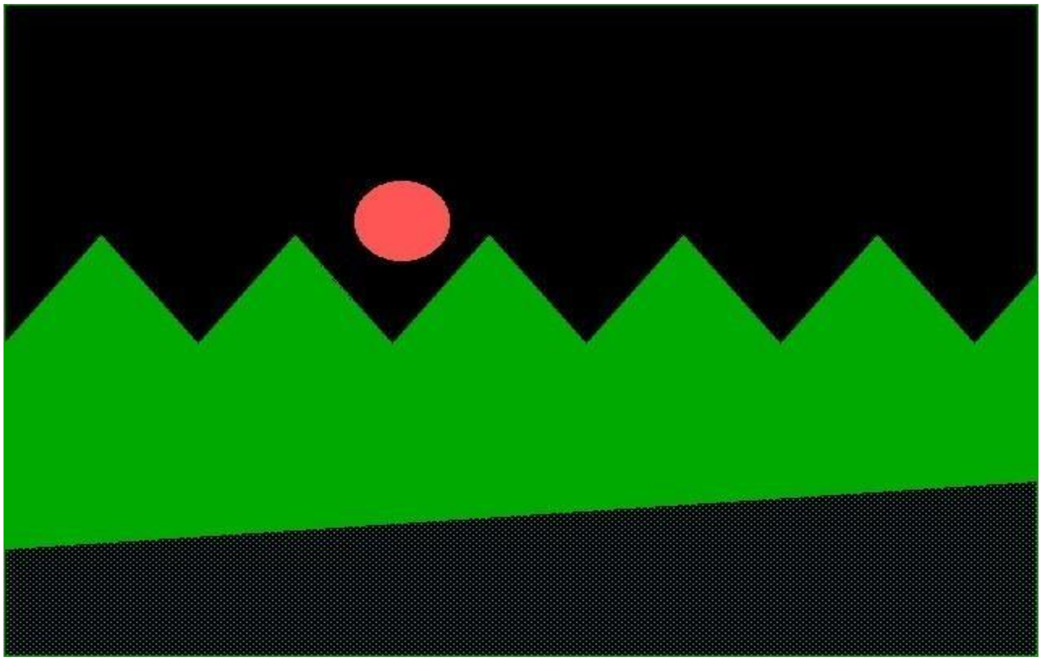
3. Code and Result

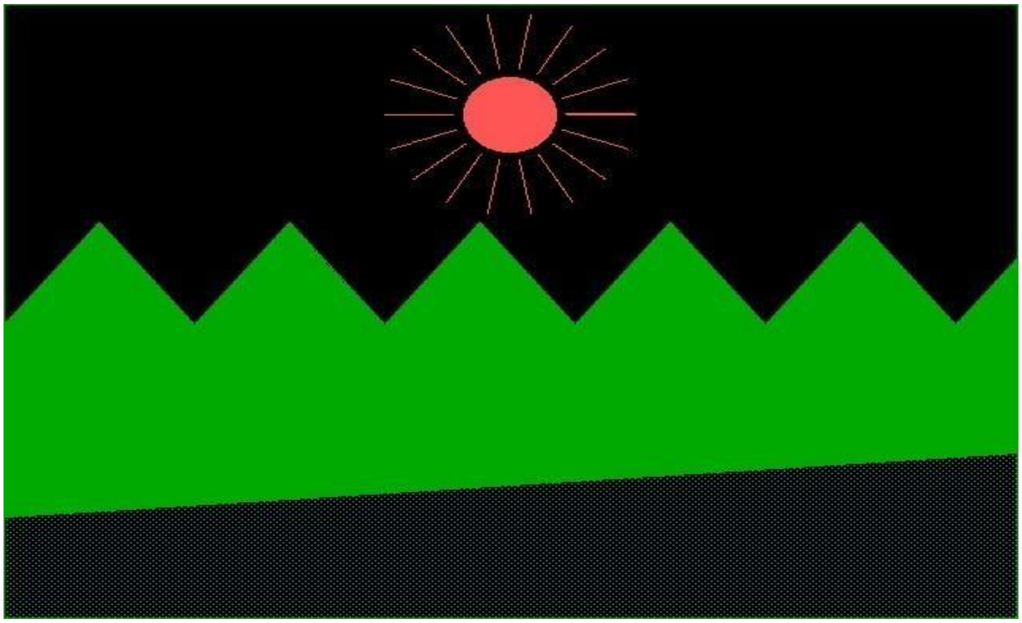
3.1 Source Code

```
#include<graphics.h>
#include<math.h>
void   main()   {   int
gd=DETECT,gm;
int       i,j,k,t,q;       float       x,y;
initgraph(&gd,&gm,"c:\\tc\\bgi");

setcolor(2);


rectangle(0,0,getmaxx(),getmaxy());
setcolor(2); i=0;
for(t=0;t<getmaxx();t+=120)
{
line(t,250,t+60,170);
line(t+60,170,t+120,250);
}
line(0,400,getmaxx(),350);
setfillstyle(11,CYAN);

floodfill(2,420,2);
     setfillstyle(4,LIGHTGREEN);
floodfill(1,300,2);

i=0;
while(i!=150)
4
{
```

```
setcolor(BLACK);
setfillstyle(SOLID_FILL,BLACK);
fillellipse(k,j,30,30);
setfillstyle(SOLID_FILL,LIGHTRED);
fillellipse(170+i,235-i,30,30); j=235-i;
k=170+i;          i++;          setcolor(2);
for(t=0;t<getmaxx();t+=120)
{
line(t,250,t+60,170);
line(t+60,170,t+120,250); }
setfillstyle(1,GREEN);
floodfill(202,200,GREEN);
delay(25); }

for(i=36;i<80;i++)
for(j=0;j<=360;j+=20)
{
x=319+i*cos(((float)j*3.14)/180);
y=86+i*sin(((float)j*3.14)/180);
putpixel(x,y,LIGHTRED);
delay(1); }
getch();
```

3.2 Snapshots of Result





4. Conclusion

This code is a simple graphics program that continuously displays a Rising Sun between the mountains in a graphics window. The Sun is of orange color and mountains is of green color .The program runs until a key is pressed and uses the Turbo C graphics library. While this program serves as a simple demonstration of graphics capabilities, it provides an example of how to use fundamental drawing functions, colors, and animation loops in the Turbo C/C++ environment to create visual effects.

Video of the Output:

https://drive.google.com/file/d/1HkmXvi0bcPQMVx6wuHd6J5a9mcALWqk/view?usp=drivesdk