Report On

Smiley Face using C graphics

Submitted in partial fulfillment of the requirements of the Course project in
Semester III of Second Year Artificial Intelligence and Data Science

by
Pranav Shetty (Roll No. 53)
Rithesh Shetty (Roll No. 54)
Vinith Shetty (Roll No. 55)

Supervisor
Prof. Sejal D'Mello

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science

(2023-24)

Vidyavardhini's College of Engineering & Technology
Department of Artificial Intelligence and Data Science

CERTIFICATE

This is to certify that the project entitled “Smiley Face using C graphics ” is a bonafide work of "Pranav Shetty (Roll No. 53), Rithesh Shetty (Roll No. 54), Vinith Shetty (Roll No. 55)" submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester III of Second Year Artificial Intelligence and Data Science engineering.

Supervisor

Prof. Name Surname

Dr. Tatwadarshi P. N.
Head of Department

Table of Contents

Pg. No

Abstract

The project aims to create a smiley face using C graphics, showcasing a basic yet universally recognized symbol of joy and positivity. The program utilizes fundamental shapes and graphical elements to generate a smiling face on the screen. Circles are employed to form the face, eyes, and mouth, while lines contribute to the facial features. The code utilizes C graphics libraries to draw and position these elements, allowing the smiley face to be displayed on the screen.
The primary focus is on the simplicity and visual representation of a smiling face, intending to provide an introductory illustration of basic graphics programming in C. The implementation provides a foundation for understanding graphical output and basic geometric shapes using C, catering to beginners exploring graphics programming concepts.

Problem Statement

Design a program in C that utilizes graphical capabilities to create a simple smiley face on the screen. The objective is to develop an application that generates a graphical representation of a smiley face, composed of basic shapes and elements, such as circles and lines, to depict the facial features. The program should allow the user to view the smiley face within a graphical window or output display. The primary challenge is to implement the graphical elements using C graphics libraries effectively, ensuring the creation and positioning of circles, lines, and other elements to form the classic representation of a smiley face.
This problem statement outlines the goal of the project, emphasizing the creation of a smiley face through graphical elements using C programming. It sets the objective of generating a visual representation of the face and hints at the utilization of C graphics libraries to achieve this goal.

Module Description:

Smiley Face in C

Initialization Module:

Description: Initializes the graphical environment for the C program, setting up the window and essential configurations for drawing.

Functions:

initializeGraphics(): Initializes the graphic environment, sets up the window, and prepares the screen for drawing.

Circle Drawing Module:

Description: Contains functions to draw circles, forming the base for the face and eyes of the smiley.

Functions:

drawCircle(int centerX, int centerY, int radius): Draws a circle with a specified center and radius.

Line Drawing Module:

Description: Consists of functions to draw lines, essential for forming the mouth and other facial features of the smiley.
Functions:
drawLine(int x1, int y1, int x2, int y2): Draws a line between two given points.
Facial Features Module:

Description: Contains functions to position and draw the eyes and mouth of the smiley face.
Functions:
drawEyes(): Positions and draws the eyes of the smiley face.
drawSmile(): Positions and draws the smile (mouth) of the smiley.
Main Module:

Description: Orchestrates the program flow, calling functions from other modules to draw and display the smiley face.
Functions:
main(): Controls the program execution, calling initialization functions and invoking other modules to draw and display the smiley face.

These module descriptions outline the functionalities and key functions involved in the C program to create a smiley face using basic graphical elements. Each module has a specific role, focusing on tasks such as initialization, drawing shapes, positioning facial features, and controlling the overall program flow.

Brief Description of Software and Hardware Used and Its Programming for the Smiley Face in C:

Software Used:

1. **Turbo C Compiler**: This is an integrated development environment (IDE) and compiler for the C programming language. It's often used for developing DOS applications. The Turbo C compiler provides the graphics.h and dos.h header files, essential for our graphical project.

2. **BGI (Borland Graphics Interface)**: An integral part of Turbo C, BGI provides the necessary functions and routines to draw various shapes, control colors, and manage graphical content. Our project's graphical representation heavily relies on BGI functions such as `initgraph()`, `line()`, `arc()`, etc.
3. **DOSBox**: An x86 emulator with built-in DOS, useful for running Turbo C on modern operating systems, especially if native support is missing.

Hardware Used:

1. **Computer**: Any basic computer system that can run the DOS or DOSBox emulator is suitable. The graphical requirements of the project are minimal, so advanced graphics cards or high RAM is not necessary.

2. **Monitor**: Required for visualizing the animation. Given the simplicity of the graphics, there's no need for high-resolution or specialty screens.

Programming:

Initializing Graphics:

Begin by initializing the graphical environment using C graphics libraries or platforms such as SDL, OpenGL, or graphics.h.
Set up the window, screen, and necessary configurations for drawing.
Drawing Circles:

Use circle-drawing functions or algorithms to draw circles on the screen.
Circles form the basis for the face and eyes of the smiley.
Drawing Lines:

Employ line-drawing functions to draw lines for forming the mouth and other facial features of the smiley face.

Positioning Facial Features:

Determine the position of the eyes and mouth within the circle representing the face.

Use the drawn circles and lines to position and create the eyes and smiling mouth.

Main Program Flow:

Control the program flow in the main function.

Call functions to initialize the graphics, draw circles, lines, and position facial features.

Displaying the Smile:

Once all elements are drawn and positioned, display the smiley face on the screen or graphical window.

Code:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include<malloc.h>
#include<dos.h>
#include <conio.h>
int xasp,yasp,gdriver = VGA, gmode=VGAMED, errorcode;
struct pos
{
 int x;
 int y;
};
struct face
{
 int radius;
struct pos posiϴon;
 int mood;
};
```

```
typedef struct face face;
face *face1;
void getposiϴon()
{
prinɳ("Enter X Co-ordinate:");
 scanf("%d",&face1->posiϴon.x);
prinɳ("Enter X Co-ordinate:");
 scanf("%d",&face1->posiϴon.y);
}
void drawface()
{
 char ch='x';
 int i=0,x,y,color,r,imgsize,dif;
 x=face1->posiϴon.x=320;
 y=face1->posiϴon.y=180;
 face1->radius=150;
 color=15;
 r=face1->radius;
 setbkcolor(0);
getaspectraϴo(&xasp,&yasp);
 setcolor(8);
 circle(x,y,face1->radius);
seɳillstyle(1,color);
 floodfill(x,y,getcolor());
 draweyes(face1);
 drawhair(face1);
 drawmouth(face1);
 drawnose(face1);
}
drawnose()
{
 int i,x,y,r;
 x=face1->posiϴon.x;
 y=face1->posiϴon.y;
```

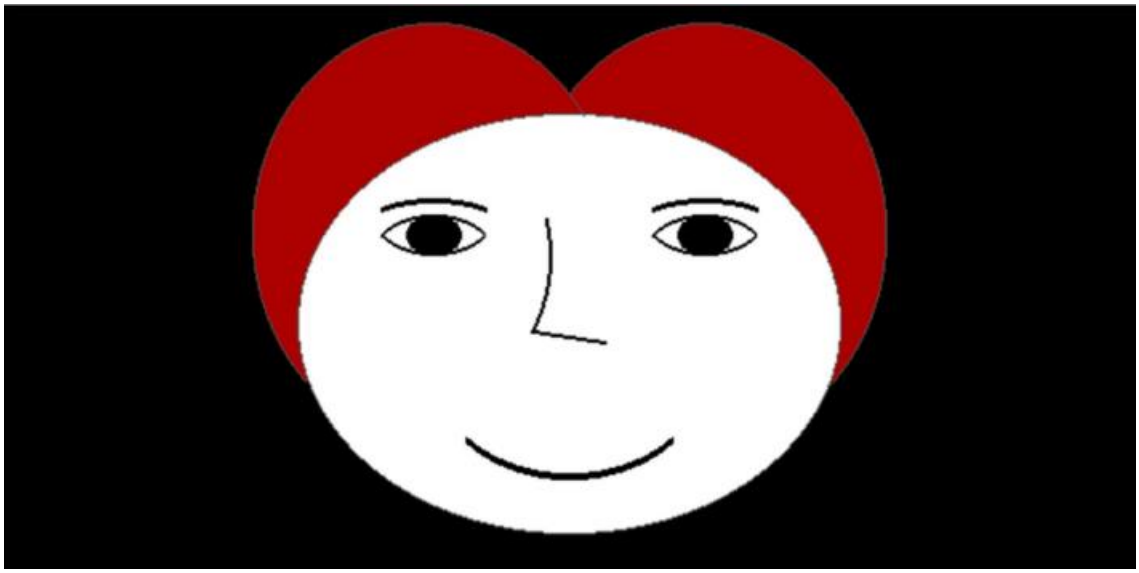
```
 r=face1->radius;
 setcolor(0);
 for(i=0;i<2;i++)
 {
 arc(x-160-i,y-r/4,340-i,10,r);
 line(x-20,y+4+i,x+20,y+10+i);
 }
}
draweyes()
{
 int i,x1,x2,y1,y2,r;
 setcolor(0);
 r=face1->radius;
 x1=face1->posiΘon.x-r/2;
 y1=face1->posiΘon.y-r/4;
 x2=face1->posiΘon.x+r/2;
 y2=face1->posiΘon.y-r/4;
setaspectraΘo(xasp/2,yasp);
 arc(x1,y1-r/8,40,140,r/4);//leŌ eyebrow
 arc(x1,y1-r/8+1,40,140,r/4);//leŌ eyebrow
 arc(x1,y1-r/8+2,40,140,r/4);//leŌ eyebrow
setaspectraΘo(xasp,yasp);
 for(i=0;i<2;i++)
 {
 arc(x1,y1+i+5,40,140,r/4); //upper leŌ eye
 arc(x1,y1-r/5+i,220,320,r/4); //lower leŌ eye
 }
 circle(x1,y1-r/12,r/10);//leŌ pupul
seɳillstyle(1,0);
 floodfill(x1,y1-r/10,getcolor());
seɳillstyle(1,WHITE);
 floodfill(x1-15,y1-r/6,getcolor());
setaspectraΘo(xasp/2,yasp);
 arc(x2,y2-r/8,40,140,r/4);//right eyebrow
```

```
 arc(x2,y2-r/8+1,40,140,r/4);//right eyebrow
 arc(x2,y2-r/8+2,40,140,r/4);//right eyebrow
setaspectraϴo(xasp,yasp);
 for(i=0;i<2;i++)
 {
 arc(x2,y2+i+5,40,140,r/4);//upper right eye
 arc(x2,y2-r/5+i,220,320,r/4);//lower right eye
 }
 circle(x2,y2-r/12,r/10);//right pupil
seɳillstyle(1,0);
 floodfill(x2,y2-r/12,getcolor());
seɳillstyle(1,WHITE);
 floodfill(x2-15,y2-r/6,getcolor());
}
drawmouth()
{
 int x,y,r,i;
 x=face1->posiϴon.x;
 y=face1->posiϴon.y+(face1->radius/1.5);
 r=face1->radius;
 setcolor(BLACK);
 if((face1->mood)==1)
 for(i=0;i<4;i++)
 arc(x,y-r/2+i,220,320,r/2);//make happy
 if((face1->mood)==0)
 for(i=0;i<4;i++)
 arc(x,y-i,40,140,r/2);//make sad
}
drawhair()
{ int x,y,r;
 setcolor(8);
setaspectraϴo(xasp,yasp/1.5);
 r=face1->radius;
 x=face1->posiϴon.x-r/2;
```

```
 y=face1->posiϴon.y-r/3;
 arc(x,y,34,225,100);
 arc(x+r,y,314,138,100);
seɳillstyle(1,RED);
 floodfill(x,y-70,getcolor());
 floodfill(x+r,y-70,getcolor());
setaspectraϴo(xasp,yasp);
}
void main(void)
{
 int i=0;
 initgraph(&gdriver, &gmode,"C:\\TC\\BGI");
 while(!kbhit())
 {
 if((i%2)==1)
 {
 setvisualpage(1);
 setacϴvepage(0);
 clearviewport();
 face1->mood=0;
 drawface();
 delay(1000);
 }
 else
 {
 setvisualpage(0);
 setacϴvepage(1);
 clearviewport();
 face1->mood=1;
 drawface();
 delay(300);
 }
 i++;
 }
```

```
getch();
closegraph();
}
```

Results and Conclusion:



Conclusion:

Creating a smiley face in C using basic graphics functions is a simple yet illustrative exercise in programming graphical elements. By utilizing circles for the face and eyes, along with lines for the mouth, a visual representation of a smiling face is achieved.

The process involves initializing the graphical environment, drawing circles for the face and eyes, positioning the facial features, and displaying the smiley face on the screen. The fundamental aspects of graphics programming, such as handling shapes and positions, are exercised during this process.

Overall, creating a smiley face in C using graphics functions is a foundational exercise that introduces the basic principles of graphical representation and serves as an initial step in understanding and implementing graphical concepts within a programming environment.