



Sonargaon University

Department: Computer Science and Engineering

Course Title: Computer Graphics

Course code: CSE413

Final Project

Topics: Design some objects

Submitted to:

Nabila Anwar, Lecturer of SU

Submitted by :

Name: Suma Akter

Student ID: CSE_1903018050

Code:

```
#include<iostream>

#include<bits/stdc++.h>

#include<graphics.h>

using namespace std;

void name_print();

void emoji();

void DDA_Algorithm();

void bresenhum_algorithm();

void mid_circle();

void Moving_Cycle();

int main()

{

    int ch;

    cout<< "\tWelcome to my
project\n=====";

    cout<< "\nMenu\n1.Name print\n2.Emoji print\n3.DDA
Algorithm\n4.Bresenham line drawing algorithm\n";

    cout<< "5.Mld point circle algorithm\n6.Moving Cycle\n0.Exit\n";
```

```
while(true)
{
    cout << "\nEnter your choice: ";
    cin>> ch;
```

```
switch(ch)
```

```
{
```

```
case 1:
```

```
{
```

```
    name_print();
```

```
}
```

```
case 2:
```

```
{
```

```
    emoji();
```

```
}
```

```
case 3:
```

```
{
```

```
    DDA_Algorithm();
```

```
}
```

```
case 4:
```

```
{
```

```
    bresenhum_algorithm();
```

```
}
```

```
case 5:
```

```
{
```

```
    mid_circle();
```

```
}
```

```
case 6:
```

```
{
```

```
    Moving_Cycle();
```

```
}
```

```
case 0:
```

```
{
```

```
    exit(0);
```

```
}
```

```
default:
```

```
    cout<< "Invalid choice. Please try again.\n";
```

```
}
```

```
}
```

```
return 0;
```

```
}
```

```
void name_print()
```

```
{
```

```
int gd = DETECT, gm;  
initgraph(&gd, &gm, "");  
initwindow(800,800);
```

```
line(100,100,200,100);  
line(100, 100,100,200);  
line(100,200,200,200);  
line(200,200,200,300);  
line(200,300,100,300);  
line(300,100,300,200);  
line(400,100,400,200);  
line(300,200,400,200);  
line(500,100,500,200);  
line(600,100,600,200);  
line(500,100,550,200);  
line(600,100,550,200);  
line(700,100,650,200);  
line(700,100,750,200);  
line(675,150,725,150);
```

```
getch();
```

```
closegraph();
```

```
}
```

```
void emoji()
```

```
{
```

```
int gd= DETECT, gm;
```

```
initgraph(&gd, &gm, "");
```

```
initwindow(1200, 1200, "Smile Emoji");
```

```
setcolor(RED);
```

```
line(345, 350, 455, 350);
```

```
ellipse(400, 350, 180, 360, 55, 50);
```

```
setfillstyle(2,RED);
```

```
floodfill(346, 351, RED);
```

```
setcolor(RED);
```

```
circle(400, 310, 150);
```

```
setfillstyle(1,YELLOW);
```

```
floodfill(430, 315, RED);
```

```
setcolor(WHITE);  
circle(350, 260, 30);  
setfillstyle(1,WHITE);  
floodfill(351, 261, WHITE);  
setcolor(BLACK);  
circle(350, 260, 10);  
setfillstyle(1,BLACK);  
floodfill(351, 261, BLACK);  
setcolor(WHITE);  
circle(450, 260, 30);  
setfillstyle(1,WHITE);  
floodfill(450, 260, WHITE);  
setcolor(BLACK);  
circle(450, 260, 10);  
setfillstyle(1,BLACK);  
floodfill(451, 261, BLACK);  
setcolor(WHITE);  
line(400, 310, 420, 330);  
line(400, 310, 380, 330);  
line(380, 330, 420, 330);  
setfillstyle(1,BLACK);
```



```
floodfill(400, 315, WHITE);
```

```
getch();
```

```
closegraph();
```

```
}
```

```
void DDA_Algorithm()
```

```
{
```

```
    float x1,x2,y1,y2,step;
```

```
    int gd=DETECT,gm;
```

```
    initgraph(&gd,&gm," ");
```

```
    cout<<("Enter the value x1 & y1:");
```

```
    cin >>x1>>y1;
```

```
    cout<<("Enter the value x2 & y2:");
```

```
    cin >>x2>>y2;
```

```
    int dx=abs(x2-x1);
```

```
    int dy=abs(y2-y1);
```

```
    cout << dx <<" " << dy << endl;
```

```
    if(dx>dy)
```

```
    {
```

```
        step=dx;
```

```
    }
```

```
else
{
    step = dy;
}

cout << "Step :" << step << endl;

float xin,yin;

xin=dx/step;

yin=dy/step;

cout << xin <<" " << yin << endl;

int x=x1;

int y=y1;

for(int i=0; i<step; i++)
{
    putpixel(x,y,YELLOW);

    x=x+xin;

    y=y+yin;

    cout << x <<" " << y << endl;

    delay(150);
}

getch();

closegraph();
```

```
}
```

```
void bresenhum_algorithm()
```

```
{
```

```
int gd=DETECT, gm, error, x0, y0, x1, y1,dx, dy, p, x, y;
```

```
initgraph(&gd, &gm, "");
```

```
initwindow(700,700, "Bresenhum");
```

```
cout<<"Enter co-ordinates of first point: ";
```

```
cin>>x0>>y0;
```

```
cout<<"Enter co-ordinates of second point: ";
```

```
cin>>x1>>y1;
```

```
dx=x1-x0;
```

```
dy=y1-y0;
```

```
x=x0;
```

```
y=y0;
```

```
p=2*dy-dx;
```

```
while(x<x1)
```

```
{
```

```
    if(p>=0)
```

```
    {
```

```
        putpixel(x,y,7);
```

```
        y=y+1;
```

```
        p=p+2*dy-2*dx;
```

```
    }
```

```
    else
```

```
    {
```

```
        putpixel(x,y,7);
```

```
        p=p+2*dy;
```

```
    }
```

```
    x=x+1;
```

```
}
```

```
getch();
```

```
closegraph();
```

```
}
```

```
void mid_circle()
{
    int gd=DETECT, gm;

    initwindow(800,700,"Circle Algorithm");

    outtextxy(150,50,"Name: Suma Akter");

    int x,y,r;
    cout << "Enter the value of x, y & r: ";
    cin >> x >> y >> r;

    int x1=0;
    int y1=r;
    int p0=1-r;
    putpixel(x,y,7);

    while(x1<y1)
    {
```

```
if(p0<0)
{
    x1=x1+1;
    p0=p0+2*x1+1;
}
else
{
    x1=x1+1;
    y1=y1-1;
    p0=p0+2*x1+1-2*y1;
}
cout << "(" << x1 << ", " << y1 << ")"<<endl;
putpixel(x+x1,y+y1,7);
putpixel(x+x1,y-y1,7);
putpixel(x-x1,y+y1,7);
putpixel(x-x1,y-y1,7);
putpixel(x+y1,y+x1,7);
putpixel(x+y1,y-x1,7);
putpixel(x-y1,y+x1,7);
putpixel(x-y1,y-x1,7);
delay(70);
```

```
}
```

```
getch();
```

```
closegraph();
```

```
}
```

```
void Moving_Cycle()
```

```
{
```

```
// Declaration
```

```
int h=0,page=0;
```

```
int circle1x = 100,circle2x =250;
```

```
int
```

```
poly_upline=150,poly_lowerline=100,poly_leftline=100,poly_rightline  
=225;
```

```
int handle_1=250,handle_2=225,handle_3 =200 ;
```

```
int sheet_1=150,sheet_2=145;
```

```
int speed =5;
```

```
int n=0;
```

```
int Cycle_Color=15;
```

```
// Creating Graphics Window
```

```
DWORD screenWidth=GetSystemMetrics(SM_CXSCREEN);
```

```
DWORD screenHeight=GetSystemMetrics(SM_CYSCREEN);
```

```
initwindow(screenWidth,screenHeight, "Cool Programming  
Projects",-3);
```

```
char s[30]="Loading...";
```

```
while(1)
```

```
{
```

```
if(n==0)
```

```
{
```

```
setcolor(WHITE);
```

```
rectangle(550+25,320,700+25,340);
```



```
for(int i=0; i<148; i++)  
{  
    setcolor(GREEN);  
    line(551+i+25,321,551+i+25,339);  
    setcolor(WHITE);  
    outtextxy(615,350,s);  
    delay(10);  
  
}  
}
```

```
if(n!=0)  
{
```

```
    setactivepage(page);  
    setvisualpage(1-page);
```

```
}  
  
cleardevice();  
  
setcolor(WHITE);  
  
  
line(0,500,1400,500);  
  
  
//cycle tires  
  
  
setcolor(Cycle_Color);  
  
circle(circle1x,450,50); //first circle left side  
circle(circle2x,450,50); //second circle risht side  
  
  
  
  
circle(circle1x,450,49); //first circle left side  
circle(circle2x,450,49); //second circle risht side  
  
  
  
  
circle(circle1x,450,48); //first circle left side  
circle(circle2x,450,48); //second circle risht side
```

```
//cycle polygon
```

```
line(poly_lowerline,450,poly_lowerline+75,450); //lower line of  
cycle
```

```
line(poly_leftline,450,poly_leftline+50,390); //left line of cycle
```

```
line(poly_upline,390,poly_upline+75,390); //upper line of cycle
```

```
line(poly_rightline,390,poly_rightline-50,450); //right line of  
cycle
```

```
line(poly_lowerline,449,poly_lowerline+75,449); //lower line of  
cycle
```

```
line(poly_leftline+1,450,poly_leftline+50+1,390); //left line of  
cycle
```

```
line(poly_upline,391,poly_upline+75,391); //upper line of cycle
```

```
line(poly_rightline-1,390,poly_rightline-50-1,450); //right line of  
cycle
```

```
line(poly_lowerline,448,poly_lowerline+75,448); //lower line of  
cycle
```

```
line(poly_leftline+2,450,poly_leftline+50+2,390); //left line of  
cycle
```

```
line(poly_upline,392,poly_upline+75,392); //upper line of cycle
```

```
line(poly_rightline-2,390,poly_rightline-50-2,450); //right line of  
cycle
```

//cycle handle

line(handle_1,450,handle_1-25,390);

line(handle_2,390,handle_2-25,340);

line(handle_3,340,handle_3-20,340);

line(handle_1-1,450,handle_1-26,390);

line(handle_2-1,390,handle_2-26,340);

line(handle_3,341,handle_3-20,341);

line(handle_1-2,450,handle_1-27,390);

line(handle_2-2,390,handle_2-27,340);

line(handle_3,342,handle_3-20,342);

//cycle sheet

line(sheet_1,390,sheet_1,380);

line(sheet_2,380,sheet_2+10,380);

line(sheet_1+1,390,sheet_1+1,380);

```
line(sheet_2,381,sheet_2+10,381);
```

```
line(sheet_1-1,390,sheet_1-1,380);
```

```
line(sheet_2,379,sheet_2+10,379);
```

```
if(n==0)
```

```
{
```

```
    setcolor(GREEN);
```

```
    settextstyle(0,HORIZ_DIR,2);
```

```
    outtextxy(440,570,"Press Any Key To Continue... ");
```

```
    getch();
```

```
    n++;
```

```
}
```

```
if(n==1)
```

```
{
```

```
    setcolor(RED);
```

```
    rectangle(550,570,600,620);
```

```
    rectangle(549,571,599,619);
```

```
    rectangle(548,572,598,618);
```

```
    setcolor(RED);
```

```
    line(560,595,590,595);
```

```
    line(560,595-1,590,595-1);
```

```
    line(560,595+1,590,595+1);
```

```
    int points[8]={560,595,570,590,570,600,560,595};
```

```
    setfillstyle(SOLID_FILL,RED);
```

```
    fillpoly(4,points);
```

```
    rectangle(720,570,770,620);
```

```
    rectangle(719,571,769,619);
```

```
    rectangle(718,572,768,621);
```

```
    line(730,595,760,595);
```

```
line(730,595+1,760,595+1);  
line(730,595-1,760,595-1);  
int points2[8]={760,595,750,590,750,600,760,595};;  
fillpoly(4,points2);
```

```
}
```

```
if(GetAsyncKeyState(VK_LEFT))
```

```
{
```

```
    circle1x    -= speed ;  
    circle2x    -= speed ;  
    poly_upline -= speed ;  
    poly_lowerline -= speed ;  
    poly_leftline -= speed ;  
    poly_rightline -= speed ;  
    handle_1     -= speed ;  
    handle_2     -= speed ;
```

```
    handle_3    -= speed ;  
    sheet_1     -= speed ;  
    sheet_2     -= speed ;  
}
```

```
if(GetAsyncKeyState(VK_RIGHT))
```

```
{  
  
    circle1x    += speed ;  
    circle2x    += speed ;  
    poly_upline += speed ;  
    poly_lowerline += speed ;  
    poly_leftline += speed ;  
    poly_rightline += speed ;  
    handle_1    += speed ;  
    handle_2    += speed ;  
    handle_3    += speed ;  
    sheet_1     += speed ;  
    sheet_2     += speed ;  
}
```



```
if(GetAsyncKeyState(VK_F1))
{
    Cycle_Color=15;

}

if(GetAsyncKeyState(VK_F2))
{
    Cycle_Color=14;
}

if(GetAsyncKeyState(VK_F3))
{
    Cycle_Color=13;
}

if(GetAsyncKeyState(VK_F4))
{
    Cycle_Color=12;
}

if(GetAsyncKeyState(VK_F5))
{
    Cycle_Color=11;
}
```

```
if(GetAsyncKeyState(VK_F6))
```

```
Cycle_Color=10;
```

```
if(GetAsyncKeyState(VK_F7))
```

```
Cycle_Color=9;
```

```
if(GetAsyncKeyState(VK_F8))
```

```
Cycle_Color=3;
```

```
if(GetAsyncKeyState(VK_F9))
```

```
Cycle_Color=5;
```

```
delay(10);
```

```
page=1-page;
```

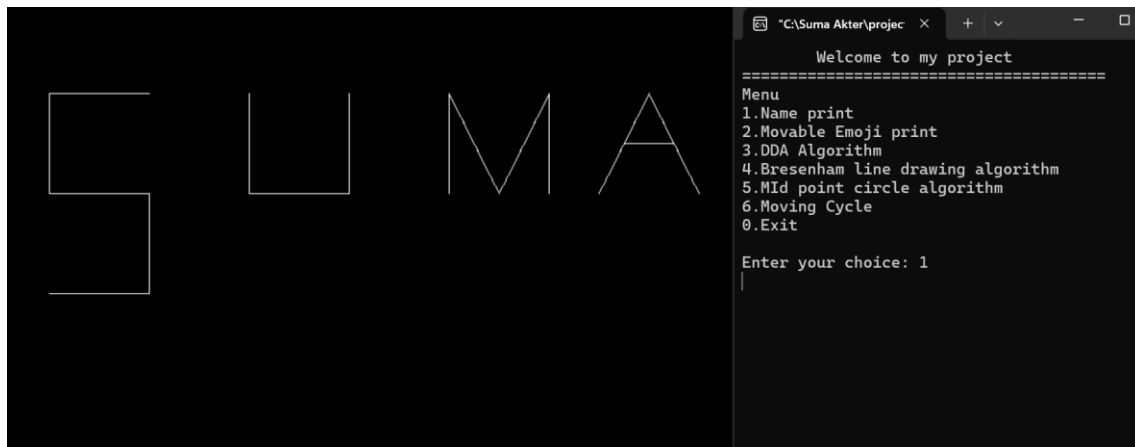
```
}
```

```
getch();
```

```
closegraph();
```

```
}
```

Output:



The screenshot displays a C++ program's output in a dark-themed terminal window. On the left, the name 'SUMA' is rendered in a large, hollow, white-outlined font. On the right, a console window titled '*C:\Suma Akter\projec' shows the program's execution. It begins with a 'Welcome to my project' message, followed by a menu of six options: 1.Name print, 2.Movable Emoji print, 3.DDA Algorithm, 4.Bresenham line drawing algorithm, 5.Mid point circle algorithm, 6.Moving Cycle, and 0.Exit. The user has entered '1' as their choice.

```
*C:\Suma Akter\projec x + - □  
  
Welcome to my project  
=====
```

```
Menu  
1.Name print  
2.Movable Emoji print  
3.DDA Algorithm  
4.Bresenham line drawing algorithm  
5.Mid point circle algorithm  
6.Moving Cycle  
0.Exit  
  
Enter your choice: 1  
|
```

