****

**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: Alamin**

**Student ID: CSE\_2102023107**

**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 analog\_clock();

void home();

int main()

{

int ch;

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

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

cout<< "5.MId point circle algorithm\n6.Clock\n7.Home\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:

{

analog\_clock();

}

case 7:

{

home();

}

case 0:

{

exit(0);

}

default:

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

}

}

return 0;

}

void name\_print()

{

int gd = DETECT, gm;

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

line(50, 10, 10, 100);

line(50, 10, 100, 100);

line(25, 65, 80, 65);

line(110, 10, 110, 100);

line(110, 100,170, 100);

line(230, 10, 180, 100);

line(230,10, 280, 100);

line(200, 65,260, 65);

line(290, 10, 290, 100);

line(290, 10, 340, 50);

line(390, 10, 340, 50);

line(390, 10,390,100);

line(400, 10,450, 10);

line(425,10, 425, 100);

line(400, 100, 450, 100);

line(460, 10, 460, 100);

line(460, 10, 560,100);

line(560, 10, 560, 100);

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, WHITE);

floodfill(346, 351, RED);

setcolor(RED);

circle(400, 310, 150);

setfillstyle(1,GREEN);

floodfill(430, 315, RED);

setcolor(WHITE);

circle(350, 260, 30);

setfillstyle(1,BLUE);

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,BLUE);

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,BLUE);

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,GREEN);

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: Alamin");

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 analog\_clock()

{

int gd = DETECT, gm;

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

initwindow(1200, 1200, "Analog Clock");

setfillstyle(SOLID\_FILL, BLACK);

floodfill(1, 1, BLACK);

setcolor(WHITE);

circle(300, 300, 200);

circle(300, 300, 202);

setfillstyle(SOLID\_FILL, LIGHTBLUE);

floodfill(300, 300, WHITE);

setcolor(WHITE);

for (int i = 0; i < 4; i++)

{

circle(300, 300, 200 + i);

}

outtextxy(295, 100, "12");

outtextxy(390, 130, "1");

outtextxy(440, 180, "2");

outtextxy(470, 295, "3");

outtextxy(440, 410, "4");

outtextxy(390, 460, "5");

outtextxy(295, 490, "6");

outtextxy(200, 460, "7");

outtextxy(150, 410, "8");

outtextxy(120, 295, "9");

outtextxy(150, 180, "10");

outtextxy(200, 130, "11");

setcolor(WHITE);

line(300, 300, 400, 200);

setcolor(WHITE);

line(300, 300, 450, 300);

setcolor(RED);

line(300, 300, 300, 470);

getch();

closegraph();

}

void home()

{

int gd = DETECT, gm;

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

initwindow(800, 800, "Home");

setcolor(WHITE);

rectangle(100, 300, 500, 400);

setfillstyle(SOLID\_FILL, LIGHTBLUE);

floodfill(101, 301, WHITE);

rectangle(120, 200, 480, 300);

setfillstyle(SOLID\_FILL, LIGHTGREEN);

floodfill(121, 201, WHITE);

rectangle(140, 100, 460, 200);

setfillstyle(SOLID\_FILL, LIGHTCYAN);

floodfill(141, 101, WHITE);

rectangle(80, 50, 120, 300);

setfillstyle(SOLID\_FILL, LIGHTGRAY);

floodfill(81, 51, WHITE);

setfillstyle(SOLID\_FILL, LIGHTMAGENTA);

floodfill(100, 30, WHITE);

rectangle(480, 50, 520, 300);

setfillstyle(SOLID\_FILL, LIGHTGRAY);

floodfill(481, 51, WHITE);

rectangle(270, 340, 330, 400);

setfillstyle(SOLID\_FILL, DARKGRAY);

floodfill(271, 341, WHITE);

rectangle(270, 340, 330, 400);

setfillstyle(SOLID\_FILL, BROWN);

floodfill(271, 341, WHITE);

rectangle(290, 240, 310, 300);

setfillstyle(SOLID\_FILL, BROWN);

floodfill(291, 241, WHITE);

rectangle(310, 140, 330, 200);

setfillstyle(SOLID\_FILL, BROWN);

floodfill(311, 141, WHITE);

rectangle(150, 330, 190, 370);

setfillstyle(SOLID\_FILL, DARKGRAY);

floodfill(151, 331, WHITE);

rectangle(410, 330, 450, 370);

setfillstyle(SOLID\_FILL, DARKGRAY);

floodfill(411, 331, WHITE);

rectangle(170, 230, 210, 270);

setfillstyle(SOLID\_FILL, DARKGRAY);

floodfill(171, 231, WHITE);

rectangle(390, 230, 430, 270);

setfillstyle(SOLID\_FILL, DARKGRAY);

floodfill(391, 231, WHITE);

rectangle(190, 130, 230, 170);

setfillstyle(SOLID\_FILL, DARKGRAY);

floodfill(191, 131, WHITE);

rectangle(370, 130, 410, 170);

setfillstyle(SOLID\_FILL, DARKGRAY);

floodfill(371, 131, WHITE);

setcolor(WHITE);

setfillstyle(SOLID\_FILL, LIGHTGRAY);

rectangle(280, 400, 320, 415);

floodfill(286, 401, WHITE);

rectangle(290, 415, 330, 430);

floodfill(301, 416, WHITE);

rectangle(300, 430, 340, 445);

floodfill(301, 431, WHITE);

rectangle(310, 445, 350, 460);

floodfill(311, 446, WHITE);

getch();

closegraph();

}

Output:

