GM-Labexam - 01 - Drawing a Clown in C++ using OpenGL

11/10/2021

Venkataraman Nagarajan, CSE - C
18500192

AIM

To draw a clown using lines(Bresenham line drawing algorithm) and circles(Midpoint circle drawing algorithm)

SPECIFICATION

Using Bresenham's Line drawing and Circle drawing algorithms draw the face of a clown.

PROGRAM - 01

3D ransformations

```
// Q: To print a clown using lines and circles
#include<GL/glew.h>
#include<GL/freeglut.h>
#include<GL/glut.h>

#include<iostream>
#include<vector>
#include<utility>
#include<algorithm>

using namespace std;

typedef long double ld;
typedef long long ll;

//WINDOW DIMENTIONS
const int WINDOW_WIDTH = 700;
const int WINDOW_HEIGHT = 700;

//ORTHO LIMITS
const ll X_MIN = -400;
const ll X_MAX = 400;
const ll Y_MIN = -400;
const ll Y_MAX = 400;

//REFRESH RATE
const ll SCREEN_FPS = 5;

//GLOBAL COUNTER
ll val = 0;

//Base functions
void myInit();
void myDisplay();
void runMainLoop(int val);

//Helper Functions
void printBresenhamLine(ld x1, ld y1, ld x2, ld y2);
void midPointCircleAlgorithm(ld x0, ld y0, ld r);

//Logic Functions
void drawClown();

```

```
int main(int argc,char* argv[]) {
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowSize(WINDOW_WIDTH,WINDOW_HEIGHT);
    glutCreateWindow("Clowns");
    glutDisplayFunc(myDisplay);
    glutTimerFunc(1000/ SCREEN_FPS, runMainLoop, 0);
    myInit();
    glutMainLoop();
    return 1;
}

void runMainLoop(int val) {
    myDisplay();

    glutTimerFunc(1000/ SCREEN_FPS, runMainLoop, 0);

}

void myInit() {
    glClearColor(1.0,1.0,1.0,0.0);
    glColor3f(0.0f,0.0f,0.0f);
    glPointSize(2);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-400,400,-400,400);
}

void myDisplay() {
    glClear(GL_COLOR_BUFFER_BIT);
    drawClown();
    val = (val+1)%2;
    glFlush();
}

void drawClown() {

    //Face outline
    glColor3f(1.0,0,0);         //red
    midPointCircleAlgorithm(0,0,200);

    //Eyes-Outer
    glColor3f(1.0,0,0);         //red
    midPointCircleAlgorithm(60,45,35);
    midPointCircleAlgorithm(-60,45,35);

    //Eyes-Inner
    glColor3f(0,0,1);           //blue
    midPointCircleAlgorithm(60,45,25);
    midPointCircleAlgorithm(-60,45,25);

    //Eye-balls
    glColor3f(0,0,1);           //blue
    for(int r=0;r<=15;r++){
        midPointCircleAlgorithm(60,45,r);
        midPointCircleAlgorithm(-60,45,r);
    }

    //Eye-Brows
    glColor3f(0,0,1.0);         //blue
    if(val) {
        printBresenhamLine(25,70,65,90);
        printBresenhamLine(65,90,100,75);
        printBresenhamLine(-25,80,-100,100);
    }
    else {
        printBresenhamLine(-25,70,-65,90);
        printBresenhamLine(-65,90,-100,75);
        printBresenhamLine(25,80,100,100);
    }
    //Nose
    glColor3f(0,1.0,0.0);       //green
    printBresenhamLine(-30,-60,0,30);
    printBresenhamLine(-25,-60,25,-60);

    //Ears
    glColor3f(1.0,0,0);         //red
    midPointCircleAlgorithm(175,175,50);
    midPointCircleAlgorithm(-175,175,50);
    glColor3f(0,0,1.0);         //blue
    midPointCircleAlgorithm(160,160,25);
    midPointCircleAlgorithm(-160,160,25);

    //cheeks
    glColor3f(0.7,0.1,0.9);
    for(int r=1;r<=20;r++) {
        midPointCircleAlgorithm(-150,-60,r);
        midPointCircleAlgorithm(150,-60,r);
    }
}

void printBresenhamLine(ld x1, ld y1, ld x2, ld y2) {
    // m : slope;

    ld dx, dy;
    ld x, y, xEnd, p, mirrorLine;
    bool printMirror = false;

    dx = abs(x2-x1);
    dy = abs(y2-y1);

    p = 2*dy - dx;

    if(x1 > x2) swap(x1,x2), swap(y1, y2);

    x = x1;
    y = y1;
    xEnd = x2;

    glBegin(GL_POINTS);
    glVertex2d(x,y);

    if(y1 > y2) {
        mirrorLine = y;
        printMirror = true;
        y2 = y1 + (y1 - y2);
    }

    while(x < xEnd) {
        x ++;

        if(p < 0) {
            p += 2*dy;
        } else {
            y ++;
            p = 2*(dy-dx);
        }

        if(printMirror) glVertex2d(x,mirrorLine - (y-mirrorLine));
        else            glVertex2d(x,y);

    }

    glEnd();
}

void midPointCircleAlgorithm(ld x0, ld y0, ld r) {
    ld p = 1-r, x = 0 , y = r;

    glBegin(GL_POINTS);

    while(x < y ) {

        glVertex2d(x0+x,y0+y);
        glVertex2d(x0+y,y0+x);
        glVertex2d(x0+x,y0-y);
        glVertex2d(x0-y,y0+x);
        glVertex2d(x0-x,y0+y);
        glVertex2d(x0+y,y0-x);
        glVertex2d(x0-x,y0-y);
        glVertex2d(x0-y,y0-x);

        if(p < 0) {
            x++;
            p += 2*x+1;
        } else {
            x++;
            y--;
            p += 1+2*(x-y);
        }
    }
    glEnd();
}
```

SAMPLE I/0

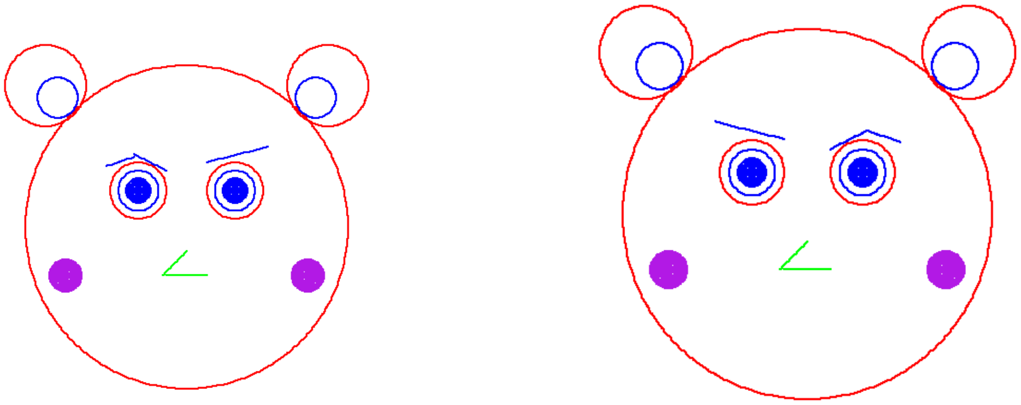

Figure 1: Clown switching eyebrows by a time delay

RESULT

The code to draw a clown is written and output is verified.