

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

```
1 // Q: To print a clown using lines and circles
2 #include<GL/glew.h>
3 #include<GL/freeglut.h>
4 #include<GL/glut.h>
5
6 #include<iostream>
7 #include<vector>
8 #include<utility>
9 #include<algorithm>
10
11 using namespace std;
12
13 typedef long double ld;
14 typedef long long ll;
15
16 //WINDOW DIMENTIONS
17 const int WINDOW_WIDTH = 700;
18 const int WINDOW_HEIGHT = 700;
19
20 //ORTHO LIMITS
21 const ll X_MIN = -400;
22 const ll X_MAX = 400;
23 const ll Y_MIN = -400;
24 const ll Y_MAX = 400;
25
26 //REFRESH RATE
27 const ll SCREEN_FPS = 5;
28
29 //GLOBAL COUNTER
30 ll val = 0;
31
32 //Base functions
33 void myInit();
34 void myDisplay();
35 void runMainLoop(int val);
36
37 //Helper Functions
38 void printBresenhamLine(ld x1, ld y1, ld x2, ld y2);
39 void midPointCircleAlgorithm(ld x0, ld y0, ld r);
40
41 //Logic Functions
42 void drawClown();
43
```

```

44 int main(int argc, char* argv[]) {
45     glutInit(&argc, argv);
46     glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
47     glutInitWindowSize(WINDOW_WIDTH, WINDOW_HEIGHT);
48     glutCreateWindow("Clowns");
49     glutDisplayFunc(myDisplay);
50     glutTimerFunc(1000/ SCREEN_FPS, runMainLoop, 0);
51     myInit();
52     glutMainLoop();
53     return 1;
54 }
55
56 void runMainLoop(int val) {
57     myDisplay();
58
59     glutTimerFunc(1000/ SCREEN_FPS, runMainLoop, 0);
60
61 }
62
63 void myInit() {
64     glClearColor(1.0, 1.0, 1.0, 0.0);
65     glColor3f(0.0f, 0.0f, 0.0f);
66     glPointSize(2);
67     glMatrixMode(GL_PROJECTION);
68     glLoadIdentity();
69     gluOrtho2D(-400, 400, -400, 400);
70 }
71
72 void myDisplay() {
73     glClear(GL_COLOR_BUFFER_BIT);
74     drawClown();
75     val = (val+1)%2;
76     glFlush();
77 }
78
79 void drawClown() {
80
81     //Face outline
82     glColor3f(1.0, 0, 0);          //red
83     midPointCircleAlgorithm(0, 0, 200);
84
85     //Eyes-Outer
86     glColor3f(1.0, 0, 0);          //red
87     midPointCircleAlgorithm(60, 45, 35);
88     midPointCircleAlgorithm(-60, 45, 35);
89
90     //Eyes-Inner

```

```

91     glColor3f(0,0,1);           //blue
92     midPointCircleAlgorithm(60,45,25);
93     midPointCircleAlgorithm(-60,45,25);
94
95     //Eye-balls
96     glColor3f(0,0,1);           //blue
97     for(int r=0;r<=15;r++){
98         midPointCircleAlgorithm(60,45,r);
99         midPointCircleAlgorithm(-60,45,r);
100    }
101
102    //Eye-Brows
103    glColor3f(0,0,1.0);           //blue
104    if(val) {
105        printBresenhamLine(25,70,65,90);
106        printBresenhamLine(65,90,100,75);
107        printBresenhamLine(-25,80,-100,100);
108    }
109    else {
110        printBresenhamLine(-25,70,-65,90);
111        printBresenhamLine(-65,90,-100,75);
112        printBresenhamLine(25,80,100,100);
113    }
114    //Nose
115    glColor3f(0,1.0,0.0);         //green
116    printBresenhamLine(-30,-60,0,30);
117    printBresenhamLine(-25,-60,25,-60);
118
119    //Ears
120    glColor3f(1.0,0,0);           //red
121    midPointCircleAlgorithm(175,175,50);
122    midPointCircleAlgorithm(-175,175,50);
123    glColor3f(0,0,1.0);           //blue
124    midPointCircleAlgorithm(160,160,25);
125    midPointCircleAlgorithm(-160,160,25);
126
127    //cheeks
128    glColor3f(0.7,0.1,0.9);
129    for(int r=1;r<=20;r++) {
130        midPointCircleAlgorithm(-150,-60,r);
131        midPointCircleAlgorithm(150,-60,r);
132    }
133 }
134
135 void printBresenhamLine(ld x1, ld y1, ld x2, ld y2) {
136     // m : slope;
137

```

```

138     ld dx, dy;
139     ld x, y, xEnd, p, mirrorLine;
140     bool printMirror = false;
141
142     dx = abs(x2-x1);
143     dy = abs(y2-y1);
144
145     p = 2*dy - dx;
146
147     if(x1 > x2) swap(x1,x2), swap(y1, y2);
148
149     x = x1;
150     y = y1;
151     xEnd = x2;
152
153     glBegin(GL_POINTS);
154     glVertex2d(x,y);
155
156     if(y1 > y2) {
157         mirrorLine = y;
158         printMirror = true;
159         y2 = y1 + (y1 - y2);
160     }
161
162     while(x < xEnd) {
163         x ++;
164
165         if(p < 0) {
166             p += 2*dy;
167         } else {
168             y ++;
169             p = 2*(dy-dx);
170         }
171
172         if(printMirror) glVertex2d(x,mirrorLine - (y-mirrorLine));
173         else             glVertex2d(x,y);
174
175     }
176
177     glEnd();
178 }
179
180 void midPointCircleAlgorithm(ld x0, ld y0, ld r) {
181     ld p = 1-r, x = 0 , y = r;
182
183     glBegin(GL_POINTS);
184

```

```
185     while(x < y ) {
186
187         glVertex2d(x0+x,y0+y);
188         glVertex2d(x0+y,y0+x);
189         glVertex2d(x0+x,y0-y);
190         glVertex2d(x0-y,y0+x);
191         glVertex2d(x0-x,y0+y);
192         glVertex2d(x0+y,y0-x);
193         glVertex2d(x0-x,y0-y);
194         glVertex2d(x0-y,y0-x);
195
196         if(p < 0) {
197             x++;
198             p += 2*x+1;
199         } else {
200             x++;
201             y--;
202             p += 1+2*(x-y);
203         }
204     }
205     glEnd();
206 }
```

SAMPLE I/O

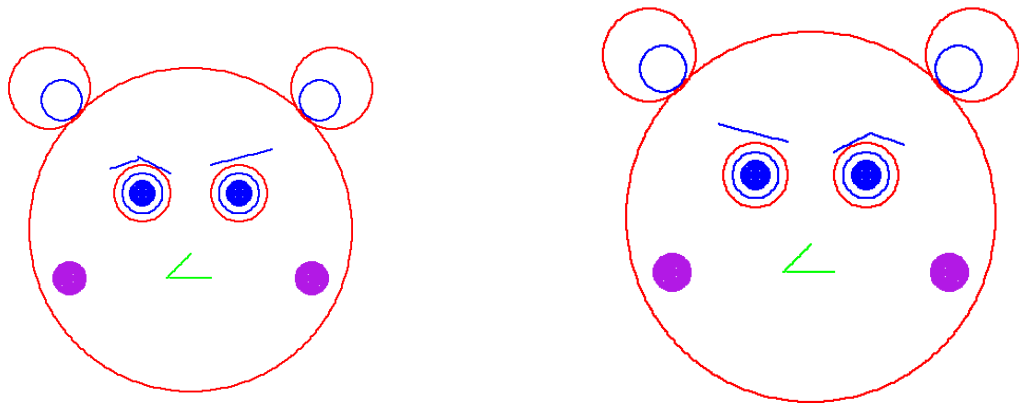


Figure 1: Clown switching eyebrows by a time delay

RESULT

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