

Bresenham's Circle Drawing Algorithm

```
#include<GL/glut.h>
#include<iostream>
using namespace std;

int r;

void E_way(int x, int y){

    glBegin(GL_POINTS);
    glVertex2i(x+320,y+240);
    glVertex2i(y+320,x+240);
    glVertex2i(y+320, -x+240);
    glVertex2i(x+320, -y+240);
    glVertex2i(-x+320,-y+240);
    glVertex2i(-y+320,-x+240);
    glVertex2i(-y+320,x+240);
    glVertex2i(-x+320,y+240);
    glEnd();
    glFlush();

}

void B_circle(){

    float d;
    d = 3 - 2*r;

    int x,y;
    x = 0 ;
    y = r ;

    do{
        E_way(x,y);

        if(d<0){

            d=d+4*x+6;
        }
        else{
            d= d+4*(x-y)+10;
            y=y-1;
        }
        x=x+1;
    }
```

```

        }while(x<y);

    }

void init(){

    glClearColor(1,1,1,0);
    glColor3f(1,0,0);
    gluOrtho2D(0,640,0,480);
    glClear(GL_COLOR_BUFFER_BIT);
}

int main(int argc, char **argv){

    cout<<"\n Enter Radius \t ";
    cin>>r;

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);

    glutInitWindowPosition(100,100);
    glutInitWindowSize(640,480);
    glutCreateWindow("Circle");
    init();
    glutDisplayFunc(B_circle);

    glutMainLoop();

    return 0;
}

```

Output:

