

Algoritmo para hacer Líneas

Christofer Fabián Chávez Carazas

Universidad Nacional de San Agustín de Arequipa

Escuela Profesional de Ciencia de la Computación

Computación Gráfica

19 de octubre de 2017

Problema

Programar un algoritmo propio para dibujar líneas

En el principio del algoritmo se verifica si la línea es recta. Si no lo es se pasa a la siguiente parte. Ahí se verifica a qué dirección va a ir la recta y en base a qué eje se van a pintar los tramos. Cada tramo tiene un tamaño del eje más largo entre el eje mas corto. Cada vez que se pinta un tramo se acumula en una variable los decimales de la anterior división. Si esta variable es mayor de uno, entonces el tramo en el que se encuentra aumenta su tamaño en uno.

```
#include <GL/glut.h>
#include <iostream>
#include <cmath>
#include <algorithm>
#include <tuple>

using namespace std;

enum Direcciones {ARRIBA, ABAJO, DERECHA, IZQUIERDA};

void init(void){
    glClearColor(1.0,1.0,1.0,0.0);
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(0.0,400.0,0.0,300.0);
}

void drawPoint(int x, int y){
    glBegin(GL_POINTS);
    glVertex2i(x,y);
    glEnd();
}

tuple<int,int> getDir(int x1, int y1, int x2, int y2){
    int dirX = -1;
    int dirY = -1;
    if(x1 < x2) dirX = DERECHA;
    else dirX = IZQUIERDA;
    if(y1 < y2) dirY = ARRIBA;
    else dirY = ABAJO;
    return make_tuple(dirX, dirY);
}

void drawLine(int x1, int y1, int x2, int y2){
    if(x1 == x2){
        int yMin = min(y1, y2);
        int yMax = max(y1, y2);

        for(int i = yMin; i <= yMax; i++){
            drawPoint(x1, i);
        }
    }
}
```

```

else if(y1 == y2){
    int xMin = min(x1,x2);
    int xMax = max(x1,x2);
    for(int i = xMin; i <= xMax; i++){
        drawPoint(i,y1);
    }
}
else{
    int dirX = -1;
    int dirY = -1;
    tie(dirX,dirY) = getDir(x1,y1,x2,y2);
    int ejeMax = max(abs(x1-x2) + 1,abs(y1-y2) + 1);
    int ejeMin = min(abs(x1-x2) + 1,abs(y1-y2) + 1);
    float saltosF = (float)ejeMax / (float)ejeMin;
    int saltos = saltosF;
    float decimalesSaltos = saltosF - saltos;
    float decimalActual = 0;
    int direcX = 1;
    int direcY = 1;
    int actualX = x1;
    int actualY = y1;
    if(dirX == IZQUIERDA) direcX = -1;
    if(dirY == ABAJO) direcY = -1;
    if(ejeMax == abs(x1-x2) + 1){
        for(actualY = y1; actualY != y2 + direcY; actualY = actualY + direcY){
            for(int i = 0; i < saltos; i++){
                drawPoint(actualX,actualY);
                if(actualX == x2) break;
                actualX = actualX + direcX;
            }
            decimalActual+= decimalesSaltos;
            if(actualX == x2) break;
            if(decimalActual >= 1){
                drawPoint(actualX,actualY);
                if(actualX == x2) break;
                actualX = actualX + direcX;
                decimalActual -= 1;
            }
        }
        actualY = y2;
        while(actualX != x2){
            actualX = actualX + direcX;
            drawPoint(actualX,actualY);
        }
    }
    else{
        for(actualX = x1; actualX != x2 + direcX; actualX = actualX + direcX){
            for(int i = 0; i < saltos; i++){
                drawPoint(actualX,actualY);
                if(actualY == y2) break;
                actualY = actualY + direcY;
            }
            decimalActual+= decimalesSaltos;
            if(actualY == y2) break;
            if(decimalActual >= 1){
                drawPoint(actualX,actualY);
                if(actualY == y2) break;
                actualY = actualY + direcY;
                decimalActual -= 1;
            }
        }
        actualX = x2;
        while(actualY != y2){
            actualY = actualY + direcY;
            drawPoint(actualX,actualY);
        }
    }
}
}

void display(){
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0,0.0,0.0);
    drawLine(20,20,200,25);
    glColor3f(0.0,0.0,0.0);
    glBegin(GL_LINES);
        glVertex2i(20,20);
        glVertex2i(200,25);
    glEnd();
    glFlush();
}

int main(int argc, char **argv)
{
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition(50,100);
    glutInitWindowSize(400,300);
    glutCreateWindow("Ejemplo");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
}

```

}

Resultados



