

# Colorear Polígonos

Christofer Fabián Chávez Carazas

Universidad Nacional de San Agustín de Arequipa

Escuela Profesional de Ciencia de la Computación

Computación Gráfica

9 de noviembre de 2017

Todas las funciones para dibujar son las hechas por mi presentadas en trabajos anteriores. A las dos funciones de pintado se le pasan la matrices con los bordes del polígono, generadas por las funciones antes mencionadas.

## 1. Qué idea propone para evitar el desbordamiento de pila en los algoritmos de inundación

Una respuesta sería implementarlo de forma iterativa con una pila.

```
void fillFigureInundacion(Matrix matrix, int color){
    changeColor(color);
    Matrix flags(matrix.size);
    list<Point> puntos;
    Point inicial;
    inicial.x = matrix.xMax / 2; inicial.y = matrix.yMax / 2;
    puntos.push_front(inicial);
    Point actual;
    Point siguiente;

    while(!puntos.empty()){
        actual = puntos.front();
        puntos.pop_front();

        if(matrix.matrix[abs(actual.y - matrix.size.height - 1)][actual.x] == false and
           flags.matrix[abs(actual.y - matrix.size.height - 1)][actual.x] == false){

            drawPoint(actual);
            flags.matrix[abs(actual.y - matrix.size.height - 1)][actual.x] = true;
            siguiente.x = actual.x;
            siguiente.y = actual.y - 1;
            if(siguiente.y != -1) puntos.push_front(siguiente);
            siguiente.x = actual.x;
            siguiente.y = actual.y + 1;
            if(siguiente.y != matrix.size.height + 1) puntos.push_front(siguiente);
            siguiente.x = actual.x - 1;
            siguiente.y = actual.y;
            if(siguiente.x != -1) puntos.push_front(siguiente);
            siguiente.x = actual.x + 1;
            siguiente.y = actual.y;
            if(siguiente.x != matrix.size.width + 1) puntos.push_front(siguiente);

        }
    }
}
```

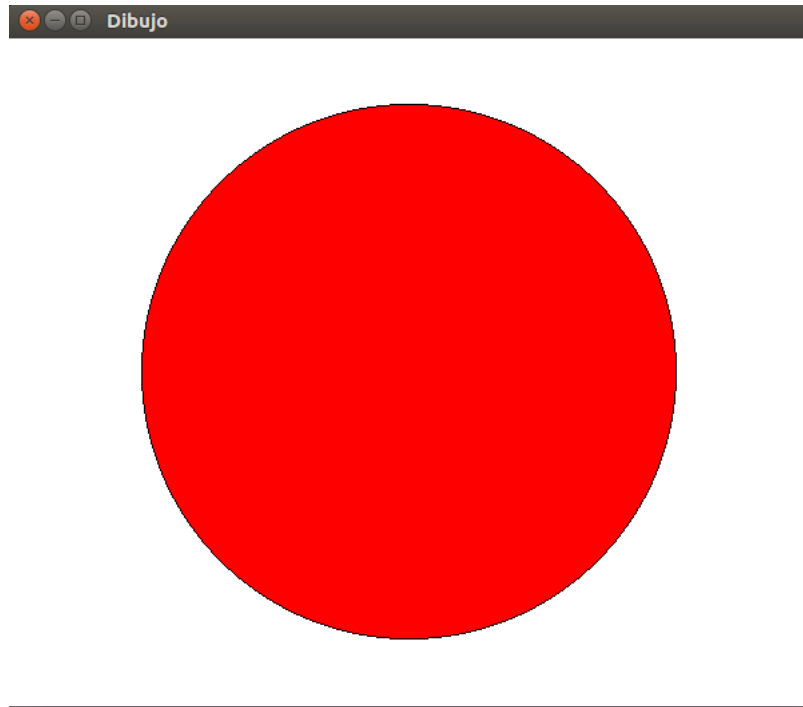


Figura 1: Resultados

## 2. Implementar el método SCANLINE para el relleno de polígonos

```
void fillFigureScanLine(Matrix matrix, int color){
    changeColor(color);
    bool flag = false;
    bool flag2 = false; // flag para pixeles seguidos
    for(int i = matrix.yMax - 1; i > matrix.yMin; i--){
        flag = false;
        for(int j = 0; j < matrix.size.width; j++){
            if(matrix.matrix[abs(i - matrix.size.height - 1)][j] == true){
                if(flag2 == false){
                    flag2 = true;
                    flag = !flag;
                }
            }
            else{
                if(flag == true){
                    setPixel(j,i);
                }
                if(flag2 == true) flag2 = false;
            }
        }
    }
}
```

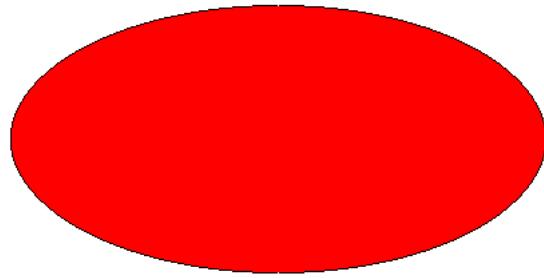


Figura 2: Resultados