

# Ejercicio del Examen

Christofer Fabián Chávez Carazas

Universidad Nacional de San Agustín de Arequipa

Escuela Profesional de Ciencia de la Computación

Computación Gráfica

7 de diciembre de 2017

## Problema

Realizar un programa el cual reciba un patrón y rellene un polígono con dicho patrón.

## Resultado

El algoritmo usa la estructura *Matrix* que contiene el borde de un polígono. Esta estructura ya ha sido utilizada en anteriores laboratorios. La estructura *Patron* es un vector de vectores de booleanos, si es 0 no se pinta, si es 1 sí se pinta.

### ■ Algoritmo

```
void fillFigureWithPatron(Matrix matrix, Patron patron, int color){
    changeColor(color);
    Point actual;
    actual.x = matrix.xMin + 1;
    actual.y = matrix.yMax - 1;
    int actualFil = 0;
    int actualCol = 0;
    while(actual.y > matrix.yMin){
        while(actual.x < matrix.xMax){
            if(matrix.matrix[abs(actual.y - matrix.size.height - 1)][actual.x] == 1) break;
            else if(patron[actualFil][actualCol] == true) drawPoint(actual);
            actual.x++;
            actualCol++;
            if(actualCol == patron[0].size()) actualCol = 0;
        }
        actual.x = matrix.xMin + 1;
        actual.y--;
        actualFil++;
        if(actualFil == patron.size()) actualFil = 0;
    }
}
```

### ■ main.cpp

```
#include <GL/glut.h>
#include <iostream>
#include <tuple>
#include <cstdlib>
#include <ctime>
#include "primitivas.h"

using namespace std;

GLsizei winWidth = 1200, winHeight = 800;
Window window;

Patron patron = {{0,0,0,0,0,0,0,0,0,0},
```

```

        {0,1,0,0,0,0,0,1,0},
        {0,1,1,0,0,0,1,1,0},
        {0,1,0,1,0,1,0,1,0},
        {0,1,0,0,1,0,0,1,0},
        {0,1,0,1,0,1,0,1,0},
        {0,1,1,0,0,0,1,1,0},
        {0,1,0,0,0,0,0,1,0},
        {0,0,0,0,0,0,0,0,0}};

void init(void){
    glClearColor(1.0,1.0,1.0,1.0);
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(0.0, 1200.0, 0.0, 800.0);
}

void display(void){
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0,0.0,0.0);
    Point iniRec;
    iniRec.x = 150; iniRec.y = 150;
    Point finRec;
    finRec.x = 300; finRec.y = 300;
    Matrix rectangle = drawRectangle(iniRec, finRec, window);
    fillFigureWithPatron(rectangle, patron, AZUL);
    glFlush();
}

int main(int argc, char **argv){
    srand (time(NULL));
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(winWidth, winHeight);
    window.height = winHeight;
    window.width = winWidth;
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Programa Primitivas");
    init();
    glutDisplayFunc(display);

    glutMainLoop();
    return 0;
}

```

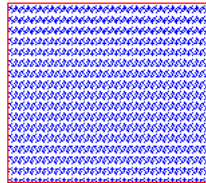


Figura 1: Resultados del algoritmo