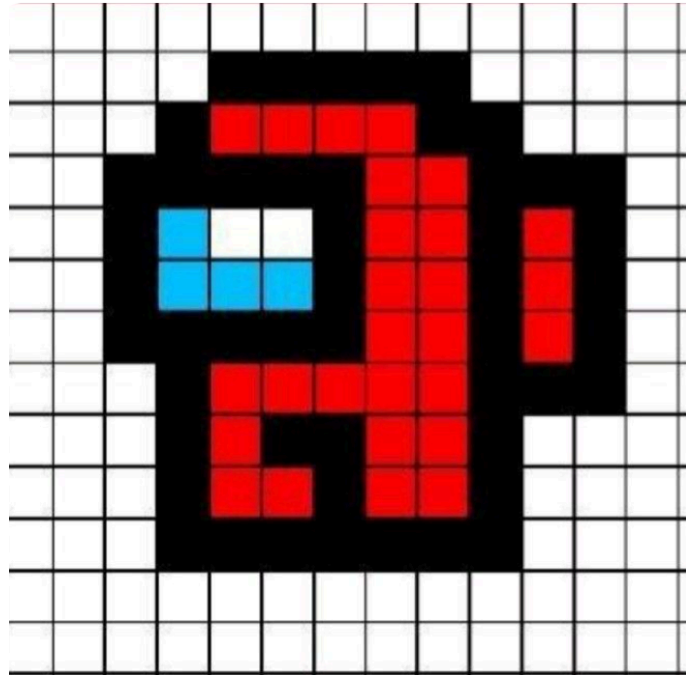


## Práctica 2. Dibujo de Primitivas 2D.

### Procedimiento

El objetivo principal de esta práctica es realizar el dibujo de una figura de estilo pixelart utilizando primitivas 2D en OpenGL. La figura que se optó por dibujar es la siguiente:



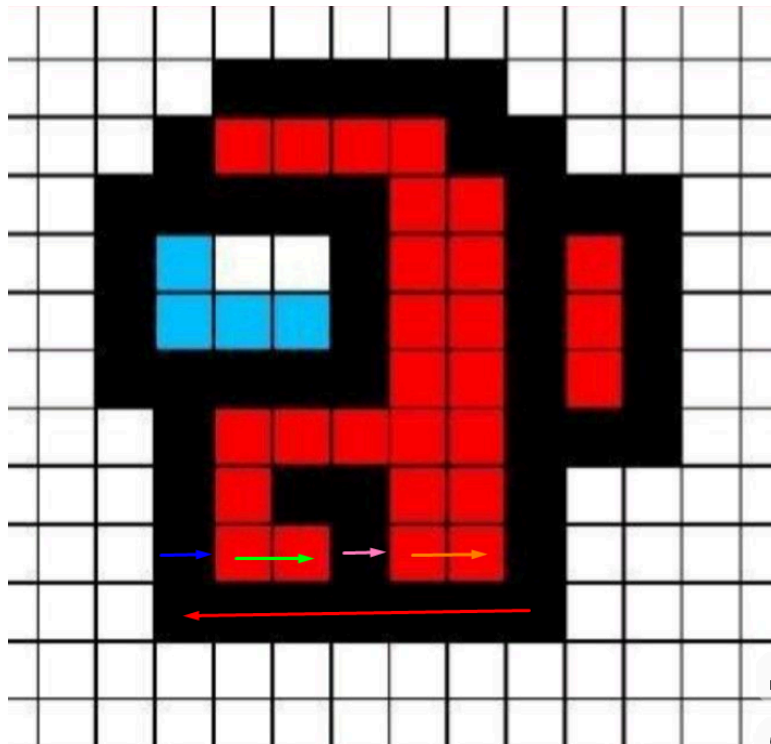
Para poder construir esta figura se usó una rejilla de celdas. Cada celda se dibuja como un rectángulo en coordenadas donde el eje X e Y van de -1 a 1. La idea principal de esto es especificar centros de 10 columnas y 10 filas (es lo que la figura utiliza para poder visualizarse correctamente en la imagen), tamaño de cada celda en alto y ancho. Esto se puede ver definido a continuación en el siguiente segmento de código:

```
const float Xc[10] = { -0.4875f, -0.4125f, -0.3375f, -0.2625f, -0.1875f, -0.1125f, -0.0375f, 0.0375f, 0.1125f, 0.1875f };  
const float Yc[10] = { -0.8f, -0.7f, -0.6f, -0.5f, -0.4f, -0.3f, -0.2f, -0.1f, 0.0f, 0.1f };  
const float DX = 0.075f;  
const float DY = 0.1f;
```

De forma implícita se fija el tamaño de la celda al definir `const float DX=0.075f` y `const float DY=0.1f`, para poder determinar esto antes había utilizado a manera de prueba una celda de 30px `glPointSize(30)` y observe que había una contigüidad con un espacio de 0.075 para el eje x y de 0.10 para el eje y.

Es por lo anterior que en Xc el paso entre cada uno de los vértices es de 0.075 y en Yc de 0.1

Para poder hacer el dibujo se pensó en dibujarlo dada una cierta cantidad de franjas de color en total son 4 colores en este caso, de modo que en cada pasada se va dibujando una parte de la figura. A continuación se muestra el proceso que se haría:



Como se puede ver en las flechas de la figura se pensó en un esquema en donde se va dibujando dependiendo de los colores, para poder automatizarlo un poco más y no tener que estar coloreando pixel por pixel, esto da un total de 41 franjas de color, de modo que es la cantidad de veces que se repetirá el proceso.

Ya una vez explicado el proceso, se explicará de forma rápida las variables usadas para definir esta lógica, son las siguientes:

1. stripRow[i] define la fila donde va el rectángulo (Yc)
2. stripC0[i] define la columna inicial del rectángulo (Xc)
3. stripC1[i] define la columna final del rectángulo (Xc)
4. stripCol[i] define el color.

los arreglos que se muestran en los primeros 3 puntos toman los datos directamente de los arreglos const float Xc y const float Yc y se van definiendo dependiendo de las franjas y de la posición donde se encuentren mediante los índices de los arreglos Xc e Yc (se muestra un pequeño segmento de código a continuación)

```
int stripC0[franjas] = {  
    // -0.8  
    1,  
    // -0.7  
    1, 2, 4, 5, 7,  
    // -0.6  
    1, 2, 3, 5, 7,  
    // -0.5  
    1, 2, 7,  
    // -0.4  
    1, 2, 7,  
    // -0.3  
    1, 2, 7,  
    // -0.2  
    1, 2, 7,  
    // -0.1  
    1, 2, 7,  
    // 0  
    1, 2, 7,  
    // 0.1  
    1, 2, 7,  
    // 0.2  
    1, 2, 7,  
    // 0.3  
    1, 2, 7,  
    // 0.4  
    1, 2, 7,  
    // 0.5  
    1, 2, 7,  
    // 0.6  
    1, 2, 7,  
    // 0.7  
    1, 2, 7,  
    // 0.8  
    1, 2, 7,  
    // 0.9  
    1, 2, 7,  
    // 1  
    1, 2, 7,  
    // 1.1  
    1, 2, 7,  
    // 1.2  
    1, 2, 7,  
    // 1.3  
    1, 2, 7,  
    // 1.4  
    1, 2, 7,  
    // 1.5  
    1, 2, 7,  
    // 1.6  
    1, 2, 7,  
    // 1.7  
    1, 2, 7,  
    // 1.8  
    1, 2, 7,  
    // 1.9  
    1, 2, 7,  
    // 2  
    1, 2, 7,  
    // 2.1  
    1, 2, 7,  
    // 2.2  
    1, 2, 7,  
    // 2.3  
    1, 2, 7,  
    // 2.4  
    1, 2, 7,  
    // 2.5  
    1, 2, 7,  
    // 2.6  
    1, 2, 7,  
    // 2.7  
    1, 2, 7,  
    // 2.8  
    1, 2, 7,  
    // 2.9  
    1, 2, 7,  
    // 3  
    1, 2, 7,  
    // 3.1  
    1, 2, 7,  
    // 3.2  
    1, 2, 7,  
    // 3.3  
    1, 2, 7,  
    // 3.4  
    1, 2, 7,  
    // 3.5  
    1, 2, 7,  
    // 3.6  
    1, 2, 7,  
    // 3.7  
    1, 2, 7,  
    // 3.8  
    1, 2, 7,  
    // 3.9  
    1, 2, 7,  
    // 4  
    1, 2, 7,  
    // 4.1  
    1, 2, 7,  
    // 4.2  
    1, 2, 7,  
    // 4.3  
    1, 2, 7,  
    // 4.4  
    1, 2, 7,  
    // 4.5  
    1, 2, 7,  
    // 4.6  
    1, 2, 7,  
    // 4.7  
    1, 2, 7,  
    // 4.8  
    1, 2, 7,  
    // 4.9  
    1, 2, 7,  
    // 5  
    1, 2, 7,  
    // 5.1  
    1, 2, 7,  
    // 5.2  
    1, 2, 7,  
    // 5.3  
    1, 2, 7,  
    // 5.4  
    1, 2, 7,  
    // 5.5  
    1, 2, 7,  
    // 5.6  
    1, 2, 7,  
    // 5.7  
    1, 2, 7,  
    // 5.8  
    1, 2, 7,  
    // 5.9  
    1, 2, 7,  
    // 6  
    1, 2, 7,  
    // 6.1  
    1, 2, 7,  
    // 6.2  
    1, 2, 7,  
    // 6.3  
    1, 2, 7,  
    // 6.4  
    1, 2, 7,  
    // 6.5  
    1, 2, 7,  
    // 6.6  
    1, 2, 7,  
    // 6.7  
    1, 2, 7,  
    // 6.8  
    1, 2, 7,  
    // 6.9  
    1, 2, 7,  
    // 7  
    1, 2, 7,  
    // 7.1  
    1, 2, 7,  
    // 7.2  
    1, 2, 7,  
    // 7.3  
    1, 2, 7,  
    // 7.4  
    1, 2, 7,  
    // 7.5  
    1, 2, 7,  
    // 7.6  
    1, 2, 7,  
    // 7.7  
    1, 2, 7,  
    // 7.8  
    1, 2, 7,  
    // 7.9  
    1, 2, 7,  
    // 8  
    1, 2, 7,  
    // 8.1  
    1, 2, 7,  
    // 8.2  
    1, 2, 7,  
    // 8.3  
    1, 2, 7,  
    // 8.4  
    1, 2, 7,  
    // 8.5  
    1, 2, 7,  
    // 8.6  
    1, 2, 7,  
    // 8.7  
    1, 2, 7,  
    // 8.8  
    1, 2, 7,  
    // 8.9  
    1, 2, 7,  
    // 9  
    1, 2, 7,  
    // 9.1  
    1, 2, 7,  
    // 9.2  
    1, 2, 7,  
    // 9.3  
    1, 2, 7,  
    // 9.4  
    1, 2, 7,  
    // 9.5  
    1, 2, 7,  
    // 9.6  
    1, 2, 7,  
    // 9.7  
    1, 2, 7,  
    // 9.8  
    1, 2, 7,  
    // 9.9  
    1, 2, 7,  
    // 10  
    1, 2, 7,  
    // 10.1  
    1, 2, 7,  
    // 10.2  
    1, 2, 7,  
    // 10.3  
    1, 2, 7,  
    // 10.4  
    1, 2, 7,  
    // 10.5  
    1, 2, 7,  
    // 10.6  
    1, 2, 7,  
    // 10.7  
    1, 2, 7,  
    // 10.8  
    1, 2, 7,  
    // 10.9  
    1, 2, 7,  
    // 11  
    1, 2, 7,  
    // 11.1  
    1, 2, 7,  
    // 11.2  
    1, 2, 7,  
    // 11.3  
    1, 2, 7,  
    // 11.4  
    1, 2, 7,  
    // 11.5  
    1, 2, 7,  
    // 11.6  
    1, 2, 7,  
    // 11.7  
    1, 2, 7,  
    // 11.8  
    1, 2, 7,  
    // 11.9  
    1, 2, 7,  
    // 12  
    1, 2, 7,  
    // 12.1  
    1, 2, 7,  
    // 12.2  
    1, 2, 7,  
    // 12.3  
    1, 2, 7,  
    // 12.4  
    1, 2, 7,  
    // 12.5  
    1, 2, 7,  
    // 12.6  
    1, 2, 7,  
    // 12.7  
    1, 2, 7,  
    // 12.8  
    1, 2, 7,  
    // 12.9  
    1, 2, 7,  
    // 13  
    1, 2, 7,  
    // 13.1  
    1, 2, 7,  
    // 13.2  
    1, 2, 7,  
    // 13.3  
    1, 2, 7,  
    // 13.4  
    1, 2, 7,  
    // 13.5  
    1, 2, 7,  
    // 13.6  
    1, 2, 7,  
    // 13.7  
    1, 2, 7,  
    // 13.8  
    1, 2, 7,  
    // 13.9  
    1, 2, 7,  
    // 14  
    1, 2, 7,  
    // 14.1  
    1, 2, 7,  
    // 14.2  
    1, 2, 7,  
    // 14.3  
    1, 2, 7,  
    // 14.4  
    1, 2, 7,  
    // 14.5  
    1, 2, 7,  
    // 14.6  
    1, 2, 7,  
    // 14.7  
    1, 2, 7,  
    // 14.8  
    1, 2, 7,  
    // 14.9  
    1, 2, 7,  
    // 15  
    1, 2, 7,  
    // 15.1  
    1, 2, 7,  
    // 15.2  
    1, 2, 7,  
    // 15.3  
    1, 2, 7,  
    // 15.4  
    1, 2, 7,  
    // 15.5  
    1, 2, 7,  
    // 15.6  
    1, 2, 7,  
    // 15.7  
    1, 2, 7,  
    // 15.8  
    1, 2, 7,  
    // 15.9  
    1, 2, 7,  
    // 16  
    1, 2, 7,  
    // 16.1  
    1, 2, 7,  
    // 16.2  
    1, 2, 7,  
    // 16.3  
    1, 2, 7,  
    // 16.4  
    1, 2, 7,  
    // 16.5  
    1, 2, 7,  
    // 16.6  
    1, 2, 7,  
    // 16.7  
    1, 2, 7,  
    // 16.8  
    1, 2, 7,  
    // 16.9  
    1, 2, 7,  
    // 17  
    1, 2, 7,  
    // 17.1  
    1, 2, 7,  
    // 17.2  
    1, 2, 7,  
    // 17.3  
    1, 2, 7,  
    // 17.4  
    1, 2, 7,  
    // 17.5  
    1, 2, 7,  
    // 17.6  
    1, 2, 7,  
    // 17.7  
    1, 2, 7,  
    // 17.8  
    1, 2, 7,  
    // 17.9  
    1, 2, 7,  
    // 18  
    1, 2, 7,  
    // 18.1  
    1, 2, 7,  
    // 18.2  
    1, 2, 7,  
    // 18.3  
    1, 2, 7,  
    // 18.4  
    1, 2, 7,  
    // 18.5  
    1, 2, 7,  
    // 18.6  
    1, 2, 7,  
    // 18.7  
    1, 2, 7,  
    // 18.8  
    1, 2, 7,  
    // 18.9  
    1, 2, 7,  
    // 19  
    1, 2, 7,  
    // 19.1  
    1, 2, 7,  
    // 19.2  
    1, 2, 7,  
    // 19.3  
    1, 2, 7,  
    // 19.4  
    1, 2, 7,  
    // 19.5  
    1, 2, 7,  
    // 19.6  
    1, 2, 7,  
    // 19.7  
    1, 2, 7,  
    // 19.8  
    1, 2, 7,  
    // 19.9  
    1, 2, 7,  
    // 20  
    1, 2, 7,  
    // 20.1  
    1, 2, 7,  
    // 20.2  
    1, 2, 7,  
    // 20.3  
    1, 2, 7,  
    // 20.4  
    1, 2, 7,  
    // 20.5  
    1, 2, 7,  
    // 20.6  
    1, 2, 7,  
    // 20.7  
    1, 2, 7,  
    // 20.8  
    1, 2, 7,  
    // 20.9  
    1, 2, 7,  
    // 21  
    1, 2, 7,  
    // 21.1  
    1, 2, 7,  
    // 21.2  
    1, 2, 7,  
    // 21.3  
    1, 2, 7,  
    // 21.4  
    1, 2, 7,  
    // 21.5  
    1, 2, 7,  
    // 21.6  
    1, 2, 7,  
    // 21.7  
    1, 2, 7,  
    // 21.8  
    1, 2, 7,  
    // 21.9  
    1, 2, 7,  
    // 22  
    1, 2, 7,  
    // 22.1  
    1, 2, 7,  
    // 22.2  
    1, 2, 7,  
    // 22.3  
    1, 2, 7,  
    // 22.4  
    1, 2, 7,  
    // 22.5  
    1, 2, 7,  
    // 22.6  
    1, 2, 7,  
    // 22.7  
    1, 2, 7,  
    // 22.8  
    1, 2, 7,  
    // 22.9  
    1, 2, 7,  
    // 23  
    1, 2, 7,  
    // 23.1  
    1, 2, 7,  
    // 23.2  
    1, 2, 7,  
    // 23.3  
    1, 2, 7,  
    // 23.4  
    1, 2, 7,  
    // 23.5  
    1, 2, 7,  
    // 23.6  
    1, 2, 7,  
    // 23.7  
    1, 2, 7,  
    // 23.8  
    1, 2, 7,  
    // 23.9  
    1, 2, 7,  
    // 24  
    1, 2, 7,  
    // 24.1  
    1, 2, 7,  
    // 24.2  
    1, 2, 7,  
    // 24.3  
    1, 2, 7,  
    // 24.4  
    1, 2, 7,  
    // 24.5  
    1, 2, 7,  
    // 24.6  
    1, 2, 7,  
    // 24.7  
    1, 2, 7,  
    // 24.8  
    1, 2, 7,  
    // 24.9  
    1, 2, 7,  
    // 25  
    1, 2, 7,  
    // 25.1  
    1, 2, 7,  
    // 25.2  
    1, 2, 7,  
    // 25.3  
    1, 2, 7,  
    // 25.4  
    1, 2, 7,  
    // 25.5  
    1, 2, 7,  
    // 25.6  
    1, 2, 7,  
    // 25.7  
    1, 2, 7,  
    // 25.8  
    1, 2, 7,  
    // 25.9  
    1, 2, 7,  
    // 26  
    1, 2, 7,  
    // 26.1  
    1, 2, 7,  
    // 26.2  
    1, 2, 7,  
    // 26.3  
    1, 2, 7,  
    // 26.4  
    1, 2, 7,  
    // 26.5  
    1, 2, 7,  
    // 26.6  
    1, 2, 7,  
    // 26.7  
    1, 2, 7,  
    // 26.8  
    1, 2, 7,  
    // 26.9  
    1, 2, 7,  
    // 27  
    1, 2, 7,  
    // 27.1  
    1, 2, 7,  
    // 27.2  
    1, 2, 7,  
    // 27.3  
    1, 2, 7,  
    // 27.4  
    1, 2, 7,  
    // 27.5  
    1, 2, 7,  
    // 27.6  
    1, 2, 7,  
    // 27.7  
    1, 2, 7,  
    // 27.8  
    1, 2, 7,  
    // 27.9  
    1, 2, 7,  
    // 28  
    1, 2, 7,  
    // 28.1  
    1, 2, 7,  
    // 28.2  
    1, 2, 7,  
    // 28.3  
    1, 2, 7,  
    // 28.4  
    1, 2, 7,  
    // 28.5  
    1, 2, 7,  
    // 28.6  
    1, 2, 7,  
    // 28.7  
    1, 2, 7,  
    // 28.8  
    1, 2, 7,  
    // 28.9  
    1, 2, 7,  
    // 29  
    1, 2, 7,  
    // 29.1  
    1, 2, 7,  
    // 29.2  
    1, 2, 7,  
    // 29.3  
    1, 2, 7,  
    // 29.4  
    1, 2, 7,  
    // 29.5  
    1, 2, 7,  
    // 29.6  
    1, 2, 7,  
    // 29.7  
    1, 2, 7,  
    // 29.8  
    1, 2, 7,  
    // 29.9  
    1, 2, 7,  
    // 30  
    1, 2, 7,  
    // 30.1  
    1, 2, 7,  
    // 30.2  
    1, 2, 7,  
    // 30.3  
    1, 2, 7,  
    // 30.4  
    1, 2, 7,  
    // 30.5  
    1, 2, 7,  
    // 30.6  
    1, 2, 7,  
    // 30.7  
    1, 2, 7,  
    // 30.8  
    1, 2, 7,  
    // 30.9  
    1, 2, 7,  
    // 31  
    1, 2, 7,  
    // 31.1  
    1, 2, 7,  
    // 31.2  
    1, 2, 7,  
    // 31.3  
    1, 2, 7,  
    // 31.4  
    1, 2, 7,  
    // 31.5  
    1, 2, 7,  
    // 31.6  
    1, 2, 7,  
    // 31.7  
    1, 2, 7,  
    // 31.8  
    1, 2, 7,  
    // 31.9  
    1, 2, 7,  
    // 32  
    1, 2, 7,  
    // 32.1  
    1, 2, 7,  
    // 32.2  
    1, 2, 7,  
    // 32.3  
    1, 2, 7,  
    // 32.4  
    1, 2, 7,  
    // 32.5  
    1, 2, 7,  
    // 32.6  
    1, 2, 7,  
    // 32.7  
    1, 2, 7,  
    // 32.8  
    1, 2, 7,  
    // 32.9  
    1, 2, 7,  
    // 33  
    1, 2, 7,  
    // 33.1  
    1, 2, 7,  
    // 33.2  
    1, 2, 7,  
    // 33.3  
    1, 2, 7,  
    // 33.4  
    1, 2, 7,  
    // 33.5  
    1, 2, 7,  
    // 33.6  
    1, 2, 7,  
    // 33.7  
    1, 2, 7,  
    // 33.8  
    1, 2, 7,  
    // 33.9  
    1, 2, 7,  
    // 34  
    1, 2, 7,  
    // 34.1  
    1, 2, 7,  
    // 34.2  
    1, 2, 7,  
    // 34.3  
    1, 2, 7,  
    // 34.4  
    1, 2, 7,  
    // 34.5  
    1, 2, 7,  
    // 34.6  
    1, 2, 7,  
    // 34.7  
    1, 2, 7,  
    // 34.8  
    1, 2, 7,  
    // 34.9  
    1, 2, 7,  
    // 35  
    1, 2, 7,  
    // 35.1  
    1, 2, 7,  
    // 35.2  
    1, 2, 7,  
    // 35.3  
    1, 2, 7,  
    // 35.4  
    1, 2, 7,  
    // 35.5  
    1, 2, 7,  
    // 35.6  
    1, 2, 7,  
    // 35.7  
    1, 2, 7,  
    // 35.8  
    1, 2, 7,  
    // 35.9  
    1, 2, 7,  
    // 36  
    1, 2, 7,  
    // 36.1  
    1, 2, 7,  
    // 36.2  
    1, 2, 7,  
    // 36.3  
    1, 2, 7,  
    // 36.4  
    1, 2, 7,  
    // 36.5  
    1, 2, 7,  
    // 36.6  
    1, 2, 7,  
    // 36.7  
    1, 2, 7,  
    // 36.8  
    1, 2, 7,  
    // 36.9  
    1, 2, 7,  
    // 37  
    1, 2, 7,  
    // 37.1  
    1, 2, 7,  
    // 37.2  
    1, 2, 7,  
    // 37.3  
    1, 2, 7,  
    // 37.4  
    1, 2, 7,  
    // 37.5  
    1, 2, 7,  
    // 37.6  
    1, 2, 7,  
    // 37.7  
    1, 2, 7,  
    // 37.8  
    1, 2, 7,  
    // 37.9  
    1, 2, 7,  
    // 38  
    1, 2, 7,  
    // 38.1  
    1, 2, 7,  
    // 38.2  
    1, 2, 7,  
    // 38.3  
    1, 2, 7,  
    // 38.4  
    1, 2, 7,  
    // 38.5  
    1, 2, 7,  
    // 38.6  
    1, 2, 7,  
    // 38.7  
    1, 2, 7,  
    // 38.8  
    1, 2, 7,  
    // 38.9  
    1, 2, 7,  
    // 39  
    1, 2, 7,  
    // 39.1  
    1, 2, 7,  
    // 39.2  
    1, 2, 7,  
    // 39.3  
    1, 2, 7,  
    // 39.4  
    1, 2, 7,  
    // 39.5  
    1, 2, 7,  
    // 39.6  
    1, 2, 7,  
    // 39.7  
    1, 2, 7,  
    // 39.8  
    1, 2, 7,  
    // 39.9  
    1, 2, 7,  
    // 40  
    1, 2, 7,  
    // 40.1  
    1, 2, 7,  
    // 40.2  
    1, 2, 7,  
    // 40.3  
    1, 2, 7,  
    // 40.4  
    1, 2, 7,  
    // 40.5  
    1, 2, 7,  
    // 40.6  
    1, 2, 7,  
    // 40.7  
    1, 2, 7,  
    // 40.8  
    1, 2, 7,  
    // 40.9  
    1, 2, 7,  
    // 41  
    1, 2, 7,  
    // 41.1  
    1, 2, 7,  
    // 41.2  
    1, 2, 7,  
    // 41.3  
    1, 2, 7,  
    // 41.4  
    1, 2, 7,  
    // 41.5  
    1, 2, 7,  
    // 41.6  
    1, 2, 7,  
    // 41.7  
    1, 2, 7,  
    // 41.8  
    1, 2, 7,  
    // 41.9  
    1, 2, 7,  
    // 42  
    1, 2, 7,  
    // 42.1  
    1, 2, 7,  
    // 42.2  
    1, 2, 7,  
    // 42.3  
    1, 2, 7,  
    // 42.4  
    1, 2, 7,  
    // 42.5  
    1, 2, 7,  
    // 42.6  
    1, 2, 7,  
    // 42.7  
    1, 2, 7,  
    // 42.8  
    1, 2, 7,  
    // 42.9  
    1, 2, 7,  
    // 43  
    1, 2, 7,  
    // 43.1  
    1, 2, 7,  
    // 43.2  
    1, 2, 7,  
    // 43.3  
    1, 2, 7,  
    // 43.4  
    1, 2, 7,  
    // 43.5  
    1, 2, 7,  
    // 43.6  
    1, 2, 7,  
    // 43.7  
    1, 2, 7,  
    // 43.8  
    1, 2, 7,  
    // 43.9  
    1, 2, 7,  
    // 44  
    1, 2, 7,  
    // 44.1  
    1, 2, 7,  
    // 44.2  
    1, 2, 7,  
    // 44.3  
    1, 2, 7,  
    // 44.4  
    1, 2, 7,  
    // 44.5  
    1, 2, 7,  
    // 44.6  
    1, 2, 7,  
    // 44.7  
    1, 2, 7,  
    // 44.8  
    1, 2, 7,  
    // 44.9  
    1, 2, 7,  
    // 45  
    1, 2, 7,  
    // 45.1  
    1, 2, 7,  
    // 45.2  
    1, 2, 7,  
    // 45.3  
    1, 2, 7,  
    // 45.4  
    1, 2, 7,  
    // 45.5  
    1, 2, 7,  
    // 45.6  
    1, 2, 7,  
    // 45.7  
    1, 2, 7,  
    // 45.8  
    1, 2, 7,  
    // 45.9  
    1, 2, 7,  
    // 46  
    1, 2, 7,  
    // 46.1  
    1, 2, 7,  
    // 46.2  
    1, 2, 7,  
    // 46.3  
    1, 2, 7,  
    // 46.4  
    1, 2, 7,  
    // 46.5  
    1, 2, 7,  
    // 46.6  
    1, 2, 7,  
    // 46.7  
    1, 2, 7,  
    // 46.8  
    1, 2, 7,  
    // 46.9  
    1, 2, 7,  
    // 47  
    1, 2, 7,  
    // 47.1  
    1, 2, 7,  
    // 47.2  
    1, 2, 7,  
    // 47.3  
    1, 2, 7,  
    // 47.4  
    1, 2, 7,  
    // 47.5  
    1, 2, 7,  
    // 47.6  
    1, 2, 7,  
    // 47.7  
    1, 2, 7,  
    // 47.8  
    1, 2, 7,  
    // 47.9  
    1, 2, 7,  
    // 48  
    1, 2, 7,  
    // 48.1  
    1, 2, 7,  
    // 48.2  
    1, 2, 7,  
    // 48.3  
    1, 2, 7,  
    // 48.4  
    1, 2, 7,  
    // 48.5  
    1, 2, 7,  
    // 48.6  
    1, 2, 7,  
    // 48.7  
    1, 2, 7,  
    // 48.8  
    1, 2, 7,  
    // 48.9  
    1, 2, 7,  
    // 49  
    1, 2, 7,  
    // 49.1  
    1, 2, 7,  
    // 49.2  
    1, 2, 7,  
    // 49.3  
    1, 2, 7,  
    // 49.4  
    1, 2, 7,  
    // 49.5  
    1, 2, 7,  
    // 49.6  
    1, 2, 7,  
    // 49.7  
    1, 2, 7,  
    // 49.8  
    1, 2, 7,  
    // 49.9  
    1, 2, 7,  
    // 50  
    1, 2, 7,  
    // 50.1  
    1, 2, 7,  
    // 50.2  
    1, 2, 7,  
    // 50.3  
    1, 2, 7,  
    // 50.4  
    1, 2, 7,  
    // 50.5  
    1, 2, 7,  
    // 50.6  
    1, 2, 7,  
    // 50.7  
    1, 2, 7,  
    // 50.8  
    1, 2, 7,  
    // 50.9  
    1, 2, 7,  
    // 51  
    1, 2, 7,  
    // 51.1  
    1, 2, 7,  
    // 51.2  
    1, 2, 7,  
    // 51.3  
    1, 2, 7,  
    // 51.4  
    1, 2, 7,  
    // 51.5  
    1, 2, 7,  
    // 51.6  
    1, 2, 7,  
    // 51.7  
    1, 2, 7,  
    // 51.8  
    1, 2, 7,  
    // 51.9  
    1, 2, 7,  
    // 52  
    1, 2, 7,  
    // 52.1  
    1, 2, 7,  
    // 52.2  
    1, 2, 7,  
    // 52.3  
    1, 2, 7,  
    // 52.4  
    1, 2, 7,  
    // 52.5  
    1, 2, 7,  
    // 52.6  
    1, 2, 7,  
    // 52.7  
    1, 2, 7,  
    // 52.8  
    1, 2, 7,  
    // 52.9  
    1, 2, 7,  
    // 53  
    1, 2, 7,  
    // 53.1  
    1, 2, 7,  
    // 53.2  
    1, 2, 7,  
    // 53.3  
    1, 2, 7,  
    // 53.4  
    1, 2, 7,  
    // 53.5  
    1, 2, 7,  
    // 53.6  
    1, 2, 7,  
    // 53.7  
    1, 2, 7,  
    // 53.8  
    1, 2, 7,  
    // 53.9  
    1, 2, 7,  
    // 54  
    1, 2, 7,  
    // 54.1  
    1, 2, 7,  
    // 54.2  
    1, 2, 7,  
    // 54.3  
    1, 2, 7,  
    // 54.4  
    1, 2, 7,  
    // 54.5  
    1, 2, 7,  
    // 54.6  
    1, 2, 7,  
    // 54.7  
    1, 2, 7,  
    // 54.8  
    1, 2, 7,  
    // 54.9  
    1, 2, 7,  
    // 55  
    1, 2, 7,  
    // 55.1  
    1, 2, 7,  
    // 55.2  
    1, 2, 7,  
    // 55.3  
    1, 2, 7,  
    // 55.4  
    1, 2, 7,  
    // 55.5  
    1, 
```

en la segunda celda ( si lo vemos como una matriz 10x10). Lo mismo aplica para stripC1, y para el caso de stripRow es más fácil, simplemente se define dada la altura donde este. En el caso de stripCol simplemente es un arreglo de índices de color y los arreglos que se muestran a continuación son la paleta de colores RGB que use:

```
float CR[4] = { 0.0f, 1.0f, 0.678f, 1.0f };  
float CG[4] = { 0.0f, 0.0f, 0.847f, 1.0f };  
float CB[4] = { 0.0f, 0.0f, 0.902f, 1.0f };
```

A continuación se muestra el segmento de código que da la lógica necesaria para poder ejecutar esto de forma correcta al momento de dibujar las franjas:

```
int starts[franjias];  
int counts[franjias];  
int v = 0;  
for (int i = 0; i < franjias; ++i) {  
    int r = stripRow[i];  
    int c0 = stripC0[i];  
    int c1 = stripC1[i];  
    int col = stripCol[i];  
  
    float xl = Xc[c0] - DX * 0.5f;  
    float xr = Xc[c1] + DX * 0.5f;  
    float yb = Yc[r] - DY * 0.5f;  
    float yt = Yc[r] + DY * 0.5f;  
  
    float rr = CR[col], gg = CG[col], bb = CB[col];  
  
    starts[i] = v;  
    counts[i] = 4;  
  
    float quad[4][6] = {  
        { xl, yb, 0.0f, rr, gg, bb },  
        { xl, yt, 0.0f, rr, gg, bb },  
        { xr, yb, 0.0f, rr, gg, bb },  
        { xr, yt, 0.0f, rr, gg, bb }  
    };  
  
    for (int k = 0; k < 4; ++k) {  
        int base = v * 6;  
        vertices[base + 0] = quad[k][0];  
        vertices[base + 1] = quad[k][1];  
        vertices[base + 2] = quad[k][2];  
        vertices[base + 3] = quad[k][3];  
        vertices[base + 4] = quad[k][4];  
        vertices[base + 5] = quad[k][5];  
        ++v;  
    }  
}
```

Con este ciclo se dota al programa de la capacidad para poder “empaquetar” todas las franjas en un solo arreglo de vértices y recordar dónde empieza cada franja para así dibujarla con `glDrawArrays(GL_TRIANGLE_STRIP, starts, counts)`

El resultado de ejecutar este script es que se guardan los vértices, se hace un análisis del color dada la paleta. En resumidas cuentas convierte toda la información contenida en los arreglos antes explicados para poder dibujar de forma correcta las franjas en cada iteración.

A continuación se muestra un pequeño ejemplo del proceso al iterar con “1 franja”, “2 franjas” y “3 franjas”:

1 franja:



2 franjas:



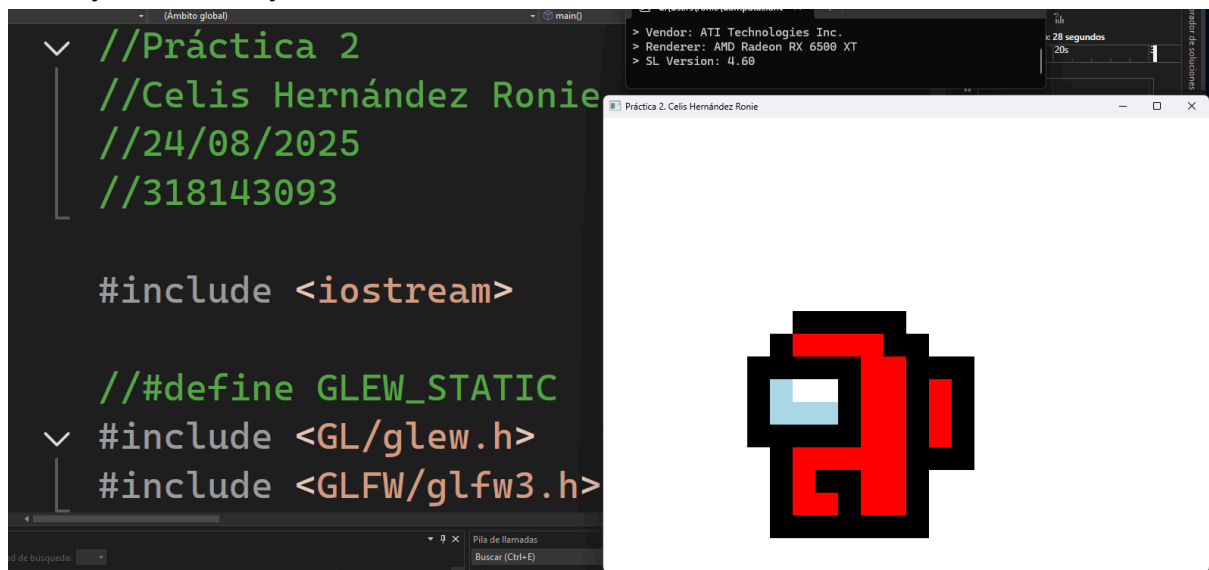
3 franjas:



*Resultado final.*

```
for (int i = 0; i < franjas; ++i) {  
    glDrawArrays(GL_TRIANGLE_STRIP, starts[i], counts[i]);  
}
```

Al dibujar las 41 franjas en la iteración del ciclo for:



**Repositorio de Github.**

- <https://github.com/ronniecelis123/ComputacionGrafica/tree/main/Practicas/Practica2>

**Bibliografía.**

- stevewhims. (2023, June 13). *Función glDrawArrays (Gl.h) - Win32 apps.*

Microsoft.com.

<https://learn.microsoft.com/es-es/windows/win32/opengl/gldrawarrays>