

Trabajo relleno por líneas de rastreo, semestre enero-junio-2025

Algoritmo de relleno por líneas de rastreo.

Desarrollar un documento y defensa que explique de manera didáctica el algoritmo de relleno por **líneas de rastreo** y su implementación en C.

Auxiliarse en el video.

<https://www.youtube.com/watch?v=kphMHaXtOoc>

implementation of scan line polygon filling algorithm

Criterios de calificación.

1. Síntesis, palabras propias.
2. Complejidad del polígono a rellenar.
3. Presentación.

Entrega. **Martes 06 de mayo 6c**

Jueves 08 de mayo 8b

Defensa directa en el salón

70-90	90-100
	Relleno de polígonos de más de 10 lados con todo tipo de ángulos y polígono "artístico".

- **main()**
- Define un arreglo de puntos `estrella` con coordenadas (x, y) para los vértices de un polígono con forma de estrella.
- Llama a `iniciar_graficos()` para iniciar el modo gráfico.
- Dibuja el contorno de la estrella con `poligono()`.
- Rellena la estrella con el algoritmo `scanlineFill()` y una animación lenta para visualizar el efecto.
- Usa `getch()` para esperar una tecla antes de cerrar la ventana gráfica.
- ---
- **iniciar_graficos()**
- Inicializa el sistema gráfico utilizando la librería BGI de Turbo C.
- `DETECT` permite detectar automáticamente el driver gráfico.
- `"c:\\tc\\bgi"` es la ruta al archivo BGI necesario para activar el modo gráfico.
- ---
- **linea(x1, y1, x2, y2)**
- Implementa una variante del **algoritmo de Bresenham** para dibujar líneas píxel por píxel.
- Calcula las diferencias entre coordenadas (`delta_x`, `delta_y`) para saber cómo avanzar en X e Y.
- Usa errores acumulados (`xerr`, `yerr`) para decidir cuándo cambiar de fila o columna.
- Usa `putpixel()` para dibujar píxeles en pantalla uno a uno.
- ---
- **poligono(n, puntos)**
- Dibuja un polígono de `n` puntos conectando cada vértice consecutivo.
- Usa `linea()` para dibujar cada lado del polígono.
- ---
- **scanlineFill(n, puntos)**
- Este es el **núcleo del efecto visual de relleno**:
- Calcula los valores mínimo y máximo de Y entre los vértices (`minY`, `maxY`).
- Para cada fila de pantalla (y desde `minY` hasta `maxY`):
 - Calcula las intersecciones entre esa línea horizontal y los lados del polígono.
 - Ordena las coordenadas X de esas intersecciones de menor a mayor.
 - Dibuja líneas horizontales entre pares de intersecciones (`intersecciones[i]` y `intersecciones[i+1]`), logrando así el relleno.
 -