



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERÍA
Escuela Profesional de Ingeniería de Sistemas

PROYECTO DE UNIDAD I
“Informe de Proyecto de Unidad I”

Curso: Diseño y Modelamiento Virtual

Docente: MSc. Ing. Hugo Manuel Barraza Vizcarra

RIVAS GUTIERREZ, SOANNY RIVAS (2023078694)

Tacna – Perú
2025

Índice

Resumen.....	3
Introducción.....	3
Objetivos:.....	3
Marco Teórico.....	3
Diseño del sistema.....	4
Arquitectura.....	4
Módulos principales.....	4
Diagramas de flujo (algoritmos).....	5
Especificación de menú y atajos.....	5
Menú contextual (clic derecho):.....	5
Atajos de teclado:.....	6
Casos de pruebas y resultados.....	6
Conclusiones y trabajo futuro.....	8

Resumen

Este trabajo presenta el desarrollo de un software de dibujo asistido por computadora (CAD 2D) implementado en C++ utilizando la librería gráfica FreeGLUT/OpenGL. El sistema integra algoritmos clásicos de rasterización de primitivas gráficas como la línea por el método directo, la línea mediante el algoritmo DDA (Digital Differential Analyzer), el círculo por el algoritmo del punto medio y la elipse por el método del punto medio. Además, incorpora funcionalidades interactivas como la visualización de cuadrícula y ejes, la selección de colores y grosores de línea, y la exportación básica de imágenes.

Los resultados muestran que la implementación permite comprender el funcionamiento interno de los algoritmos gráficos, así como su aplicación en un entorno interactivo. Este prototipo constituye un recurso pedagógico y una base inicial para proyectos más avanzados en gráficos por computadora y sistemas CAD.

Palabras clave: C++, FreeGLUT, OpenGL, rasterización, CAD 2D, algoritmos gráficos, punto medio, DDA.

Introducción

El diseño de herramientas CAD (Computer Aided Design) ha sido históricamente un área de aplicación fundamental de los gráficos por computadora. En este contexto, aprender e implementar algoritmos de dibujo de primitivas gráficas básicas permite a los estudiantes comprender los fundamentos del renderizado digital y la rasterización de figuras geométricas en pantalla.

Este proyecto busca desarrollar un sistema CAD 2D básico en C++ que permita dibujar líneas, círculos y elipses utilizando diferentes algoritmos clásicos, en un entorno gráfico interactivo.

Objetivos:

- Implementar algoritmos de rasterización clásicos: línea directa, DDA, círculo y elipse por punto medio.
- Diseñar una interfaz básica en OpenGL que permita la interacción mediante ratón, teclado y menús contextuales.
- Incorporar funciones adicionales como cuadrícula, ejes cartesianos, colores y grosores de línea.
- Proporcionar una herramienta educativa que facilite la comprensión de los fundamentos de la rasterización.

Marco Teórico

- **Método Directo para líneas:** Se basa en la ecuación de la recta $y = mx + b$. Se calcula el valor de y a partir de x (o viceversa, dependiendo de la pendiente). Aunque sencillo, presenta problemas de redondeo y precisión, además de ineficiencia en pendientes pronunciadas.

- **Algoritmo DDA (Digital Differential Analyzer):** Divide la línea en “pasos” determinados por la diferencia mayor entre dx y dy . Calcula incrementos constantes en x y y , dibujando píxel por píxel. Es más preciso que el método directo, aunque todavía requiere operaciones de punto flotante.
- **Algoritmo del Punto Medio para círculos:** Se basa en la simetría de los círculos y en una función de decisión que determina si el siguiente píxel se coloca hacia el este o hacia el sureste. Reduce cálculos costosos usando solo sumas y restas. Es uno de los algoritmos más eficientes y utilizados en rasterización.
- **Algoritmo del Punto Medio para elipses:** Similar al de los círculos, aprovecha la simetría y utiliza dos regiones de decisión (una más cercana al eje mayor y otra al eje menor). Permite rasterizar elipses de manera eficiente usando únicamente operaciones enteras.

Diseño del sistema

Arquitectura

El sistema se basa en una arquitectura modular:

- **Módulo principal (main.cpp):** Se encarga de inicializar la librería GLUT, crear la ventana principal y establecer los *callbacks* que coordinan las diferentes acciones del programa. Este módulo actúa como núcleo de la aplicación, ya que organiza la ejecución de los demás componentes y asegura el correcto funcionamiento del entorno gráfico.
- **Módulo de dibujo:** Implementa los algoritmos de rasterización fundamentales, como el método de la línea directa, el algoritmo DDA, y los algoritmos de punto medio para círculos y elipses. Su correcta implementación permitió comprobar el comportamiento de estos métodos clásicos y comparar sus ventajas y limitaciones, convirtiéndose en la base visual del sistema.
- **Módulo de interacción:** Administra la entrada del ratón, teclado y menús, lo que otorga al usuario la capacidad de interactuar con el lienzo de manera dinámica. Gracias a este módulo, se implementaron opciones de selección y ejecución de algoritmos, así como la navegación por menús que facilitan la usabilidad del programa.
- **Módulo de utilidades:** Contiene funciones auxiliares que complementan la experiencia de uso, como la generación de cuadrículas y ejes de referencia, la limpieza del lienzo, el manejo de coordenadas y la exportación de resultados. Este módulo brinda soporte a los demás y contribuye a mantener el código más claro y reutilizable.

Módulos principales

- **Entrada/Interacción:** ratón, teclado, menús.

- **Algoritmos gráficos:** rasterización de primitivas.
- **Renderizado:** OpenGL (cuadrícula, ejes, figuras).
- **Persistencia:** exportación en formato PPM.

Diagramas de flujo (algoritmos)

a) Línea directa

- Calcular pendiente m y b .
- Recorrer coordenadas incrementando x (o y).
- Dibujar píxel redondeado en cada paso.

b) DDA

- Calcular dx , dy , y pasos = $\max(|dx|, |dy|)$.
- Calcular incrementos $xInc$, $yInc$.
- Repetir hasta pasos: $x += xInc$, $y += yInc$.

c) Círculo punto medio

- Iniciar en $(0, r)$ con decisión $d = 1 - r$.
- Dibujar puntos en los 8 octantes.
- Actualizar decisión: si $d < 0$, mover este; si no, mover sureste.

d) Elipse punto medio

- Iniciar en $(0, ry)$.
- Región 1: mover en x hasta que $px < py$.
- Región 2: mover en y hasta $y = 0$.

Especificación de menú y atajos

Menú contextual (clic derecho):

- Dibujo:
 - Línea directa: Traza una línea utilizando el método matemático directo (ecuación de la recta).
 - Línea DDA: Dibuja una línea con el algoritmo DDA (Digital Differential Analyzer), más eficiente y preciso para la rasterización.
 - Círculo: Permite dibujar un círculo empleando el algoritmo del punto medio.
 - Elipse: Dibuja una elipse utilizando el algoritmo del punto medio para curvas elípticas.
- Color:
 - Negro, Rojo, Verde, Azul: Cambia rápidamente el color de dibujo a uno de los predeterminados.

- Personalizado. Permite seleccionar un color específico definido por el usuario.
- Grosor:
 - 1 px, 2 px, 3 px, 5 px.: Establece el grosor de las líneas y figuras dibujadas.
- Vista:
 - Mostrar/Ocultar cuadrícula: Activa o desactiva la visualización de la cuadrícula para referencia espacial.
 - Mostrar/Ocultar ejes: Muestra u oculta los ejes cartesianos en el lienzo.
 - Mostrar/Ocultar coordenadas: Habilita o deshabilita la visualización de coordenadas del puntero en pantalla.
- Herramientas:
 - Limpiar lienzo: Borra todo el contenido del área de dibujo.
 - Borrar última figura: Elimina solo la última figura dibujada, permitiendo correcciones rápidas.
 - Exportar imagen: Guarda el contenido actual del lienzo en un archivo de imagen.
- Ayuda:
 - Atajos de teclado: Muestra una lista de teclas rápidas disponibles en el programa.
 - Acerca de.: Despliega información sobre la aplicación, autores o versión del sistema.

Atajos de teclado:

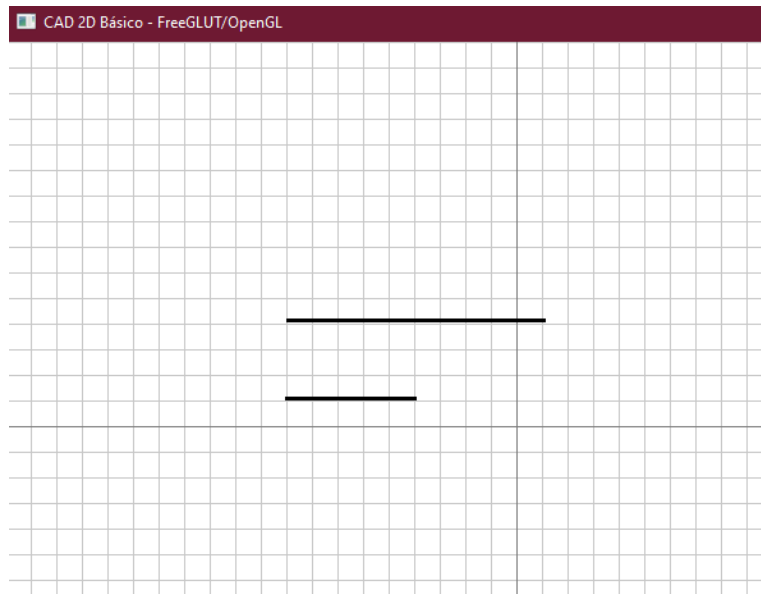
- G: Mostrar u ocultar la cuadrícula en el lienzo.
- E: Mostrar u ocultar los ejes cartesianos.
- C: Limpiar por completo el lienzo.
- S: Exportar el dibujo actual como imagen.
- Z: Deshacer (eliminar) la última figura dibujada.
- ESC: Salir del programa.

Casos de pruebas y resultados

Prueba 1: Línea Directa

Entrada: Colocamos los puntos a nuestra preferencia

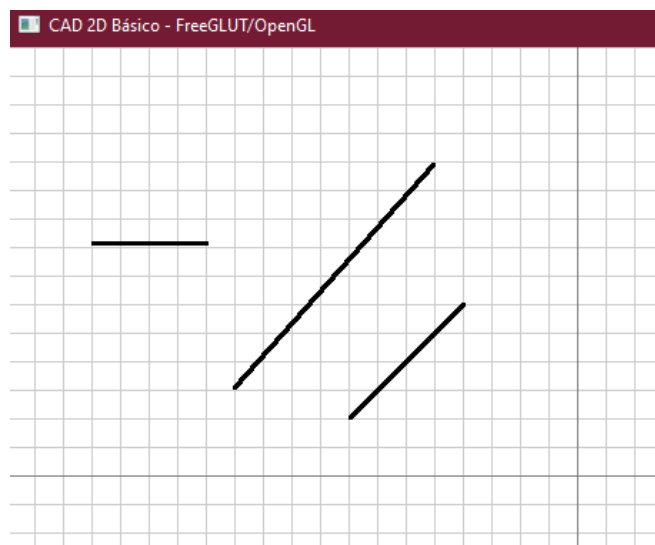
Resultado: Línea suave, aunque con saltos en pendientes altas.



Prueba 2: DDA

Entrada: Colocamos los puntos a nuestra preferencia

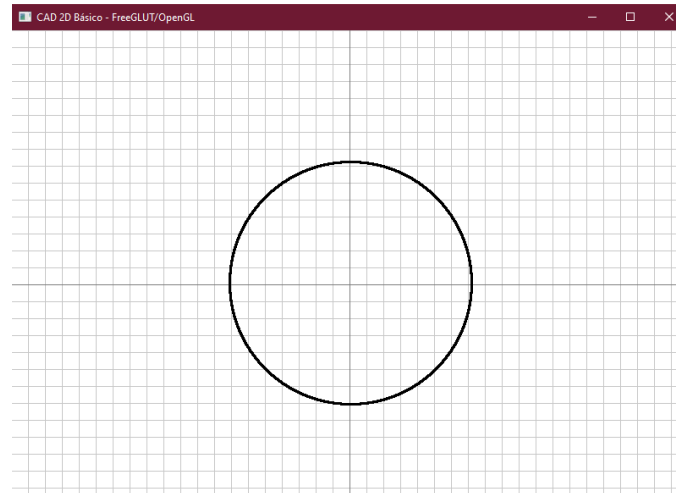
Resultado: Línea más uniforme, sin distorsión.



Prueba 3: Círculo

Entrada: Colocamos los puntos a nuestra preferencia.

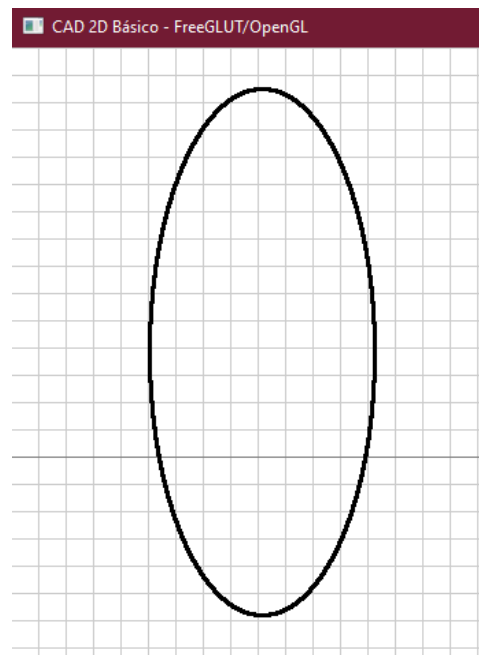
Resultado: Círculo completo en ocho octantes, bien rasterizado.



Prueba 4: Elipse

Entrada: Colocamos los puntos a nuestra preferencia

Resultado: Elipse precisa, simétrica.



Conclusiones y trabajo futuro

El desarrollo de este CAD 2D básico permitió implementar y comprobar el funcionamiento de algoritmos fundamentales de rasterización, confirmando su relevancia en el aprendizaje y aplicación de los gráficos por computadora. Los resultados obtenidos demuestran que cada algoritmo posee ventajas y limitaciones según el contexto de uso, lo que permite a los estudiantes comprender mejor los criterios de selección y optimización en proyectos reales. Asimismo, se evidenció que el uso de librerías gráficas como OpenGL no solo simplifica la representación visual, sino que también facilita la construcción de entornos interactivos que mejoran la experiencia del usuario y potencian las posibilidades de experimentación.

Como trabajo futuro, se plantea ampliar las capacidades del sistema mediante la incorporación de nuevas funcionalidades. Entre ellas, destacan el soporte para zoom y desplazamiento del lienzo, que otorgaría mayor flexibilidad en la edición de figuras; la inclusión de un sistema de capas y herramientas de edición avanzada, que permitiría organizar y modificar de manera más eficiente los elementos gráficos; y la implementación de algoritmos adicionales, como el de Bresenham y otros métodos de rasterización, que aportarían mayor precisión y diversidad de técnicas al programa. Estas mejoras no solo fortalecerían la utilidad del software como herramienta académica, sino que también lo proyectarían hacia un entorno más cercano a las aplicaciones profesionales, consolidando su valor como plataforma de aprendizaje y experimentación en el área de gráficos computacionales.