



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas

INFORME TÉCNICO – PROYECTO DE UNIDAD

Curso: DISEÑO Y MODELAMIENTO VIRTUAL

Docente: Hugo Barraza Vizcarra

Ludeña Bedoya, Milagros Shadu (2023078331)

**Tacna – Perú
2025**

INFORME TÉCNICO DEL PROYECTO DE UNIDAD	3
Resumen	3
Palabras clave	3
Introducción	4
Objetivos	4
Objetivo general	4
Objetivos específicos	4
Marco Teórico	4
Algoritmo Directo para líneas	4
Algoritmo Digital Differential Analyzer (DDA)	4
Algoritmo de Punto Medio para círculos	5
Algoritmo de Punto Medio para elipses	5
Diseño del Sistema	5
Arquitectura general	5
Módulos principales	5
Especificación del Menú y Atajos	5
Menú contextual (clic derecho)	5
Casos de Prueba y Resultados	6
Dibujo de líneas	6
Dibujo con algoritmo DDA	6
Círculo con punto medio	6
Elipse con punto medio	6
Opciones de color y grosor	6
Herramientas de edición	6
Discusión	6
Conclusiones	7

INFORME TÉCNICO DEL PROYECTO DE UNIDAD

Resumen

El presente proyecto tiene como finalidad el desarrollo de una aplicación gráfica en lenguaje C++ utilizando la librería OpenGL con GLUT, con el propósito de implementar un entorno básico de dibujo tipo CAD 2D. La aplicación permite al usuario construir figuras geométricas elementales mediante algoritmos clásicos de rasterización: líneas (algoritmo directo y DDA), círculos y elipses (ambos con el método del punto medio). Además, se ofrecen opciones para cambiar el color y grosor de línea, activar o desactivar visualización de cuadrícula y ejes, y gestionar el dibujo con herramientas de edición como deshacer y rehacer. También se incluye la capacidad de exportar la imagen resultante en formatos PPM y PNG. Este proyecto busca fortalecer la comprensión práctica de los algoritmos estudiados, integrándolos en un sistema interactivo y modular.

Palabras clave

OpenGL, GLUT, Rasterización, CAD 2D, C++, Algoritmos de líneas, Círculo punto medio, Elipse punto medio.

Introducción

El desarrollo de herramientas CAD asistidas por computadora requiere una base sólida en algoritmos gráficos para la generación precisa de figuras. En este proyecto se implementó un sistema básico en C++ con OpenGL y GLUT que permite la creación de dibujos 2D utilizando algoritmos de rasterización clásicos. Dicha implementación facilita la exploración didáctica de las técnicas gráficas fundamentales, a la vez que proporciona una interfaz gráfica accesible y funcional para el usuario.

Objetivos

Objetivo general

Desarrollar una aplicación interactiva en C++ usando OpenGL y GLUT para dibujar figuras geométricas básicas, aplicar herramientas de edición, y exportar dibujos, consolidando el aprendizaje de los algoritmos de rasterización.

Objetivos específicos

- Implementar algoritmos para rasterizar líneas mediante el método directo y DDA.
- Implementar algoritmos del punto medio para dibujo de círculos y elipses.
- Desarrollar un sistema de interacción con menús contextuales y atajos de teclado.
- Permitir la configuración dinámica de color, grosor, visualización de cuadrícula y ejes.
- Integrar funcionalidades de edición: limpiar lienzo, deshacer y rehacer.
- Incorporar la exportación de imágenes en formatos PPM y PNG.

Marco Teórico

Algoritmo Directo para líneas

Basado en la ecuación de la recta

$$y=mx+b$$

$y=mx+b$, calcula los valores de y incrementando x en pasos unitarios.

Aunque es sencillo, puede generar discontinuidades por efectos de redondeo.

Algoritmo Digital Differential Analyzer (DDA)

Mejora el método directo al usar incrementos fraccionarios en x o y , calculando cada punto sucesivamente para lograr líneas más continuas y suaves, aunque con operaciones en punto flotante.

Algoritmo de Punto Medio para círculos

Utiliza la simetría del círculo y un parámetro de decisión para determinar el siguiente punto a dibujar, reduciendo costo computacional y garantizando figuras continuas.

Algoritmo de Punto Medio para elipses

Divide la elipse en regiones para optimizar el cálculo, usando decisiones basadas en parámetros para avanzar horizontal o verticalmente, aprovechando su simetría para eficiencia.

Diseño del Sistema

Arquitectura general

El sistema se estructura en módulos claramente definidos:

- Interfaz de usuario: Menús configurados con GLUT y atajos de teclado para facilitar la selección de herramientas y configuración.

- Módulo de entrada: Captura eventos de mouse y teclado para obtener las coordenadas de puntos y activar las funciones de dibujo o edición.
- Módulo de rasterización: Implementa los algoritmos para calcular los puntos a dibujar y renderizar mediante OpenGL.
- Módulo de edición: Maneja funciones para limpiar el lienzo y la gestión de operaciones de deshacer/rehacer usando pilas.
- Módulo de exportación: Permite guardar la imagen gráfica en formato PPM o PNG.
- Módulo de visualización: Controla la representación de cuadrícula, ejes y muestra las coordenadas del puntero en tiempo real.

Módulos principales

- Menú principal con opciones de dibujo, color, grosor, vista, y edición.
- Gestor de eventos para mouse y teclado.
- Rasterizador que dibuja utilizando los algoritmos indicados.
- Administrador de estructuras de datos para figuras y edición.
- Exportador para formatos de imagen.

Especificación del Menú y Atajos

Menú contextual (clic derecho)

Opciones organizadas en submenús para selección de

- Dibujo: línea directa, línea DDA, círculo punto medio, elipse punto medio.
- Color: negro, rojo, verde, azul y personalizado.
- Grosor: 1px, 2px, 3px, 5px.
- Vista: activar/deshacer cuadrícula, ejes, coordenadas.
- Edición: limpiar lienzo, deshacer, rehacer.

Casos de Prueba y Resultados

Dibujo de líneas

Se verificó que las líneas con algoritmo directo y DDA generan trazados continuos, aunque DDA produce líneas visualmente más suaves.

Dibujo con algoritmo DDA

Confirmada la precisión y continuidad en líneas con pendientes variadas.

Círculo con punto medio

Los círculos generados son simétricos y sin huecos, cumpliendo la eficiencia esperada.

Elipse con punto medio

Las elipses son consistentes en forma y simetría, validando el diseño del algoritmo.

Opciones de color y grosor

El sistema responde correctamente a cambios dinámicos en color y grosor de figura.

Herramientas de edición

Deshacer y rehacer operaron según lo esperado, facilitando interactividad.

Discusión

Los resultados muestran que la implementación de los algoritmos clásicos junto con la interfaz interactiva posibilita un aprendizaje efectivo de rasterización gráfica y programación en OpenGL. La modularidad del sistema permite su extensión futura.

Conclusiones

El sistema cumple con los objetivos de implementar algoritmos de rasterización para líneas, círculos y elipses, junto con herramientas de edición y exportación en un interfaz amigable. Este entorno refuerza la teoría gráfica a través de la experiencia práctica con OpenGL y C++, adecuándose a los requerimientos académicos y ampliando capacidades de programación gráfica.