

Universidad Privada de Tacna



CURSO : DISEÑO DE MODELAMIENTO VIRTUAL
DOCENTE : HUGO MANUEL BARRAZA VIZCARRA
ALUMNA : MILAGROS CONDORI ACERO
SECCION : “A”

Tacna - Perú

202

INDICE

RESUMEN.....	3
palabras clave.....	3
Introducción y objetivos.....	3
Marco teórico breve de Directo, ADD, Punto Medio (círculo y elipse).....	4
Diseño del sistema: arquitectura, módulos, diagramas de flujo por algoritmo.....	5
Especificación del menú y atajos.	
Menú contextual con opciones para:.....	6
Casos de prueba y resultados (capturas con rejilla/zoom + discusión).....	7
Conclusiones y trabajo futuro.....	7
Referencias.....	8

RESUMEN

El presente proyecto desarrolla un sistema básico CAD 2D que permite dibujar líneas (directo y DDA), círculos (incremental y punto medio) y elipses (punto medio), utilizando OpenGL y GLUT. Incluye funcionalidades de visualización con cuadrícula, ejes y coordenadas, además de soporte para menú contextual y atajos de teclado para facilitar la interacción. Se implementa un sistema de historial para deshacer/rehacer acciones, y exportación de la escena a imagen en formato PPM. Se presentan los algoritmos gráficos, diseño del sistema, pruebas y resultados con discusión.

palabras clave.

CAD 2D, OpenGL, GLUT, algoritmo DDA, algoritmo punto medio, círculo, elipse, gráfico por computadora, historial undo/redo, menú contextual.

Introducción y objetivos.

Introducción:

El diseño asistido por computadora (CAD) es fundamental en ingeniería para la creación digital de planos y figuras geométricas. Este proyecto se enfoca en la implementación de un sistema CAD 2D básico utilizando OpenGL/GLUT para representar primitivas gráficas mediante algoritmos clásicos de dibujo.

Objetivos:

- Implementar los algoritmos directo y DDA para líneas.

- Implementar algoritmos punto medio e incremental para círculos.
- Implementar algoritmo punto medio para elipses.
- Proveer interacción gráfica con menú, atajos, cuadrícula y ejes.
- Incorporar historial de dibujo para acciones undo/redo.
- Permitir exportar la imagen en formato PPM.

Marco teórico breve de Directo, ADD, Punto Medio (círculo y elipse).

- **Algoritmo Directo:** Calcula los puntos de una línea usando la ecuación $y = mx + b$ con resolución punto a punto.
- **Algoritmo DDA (Digital Differential Analyzer):** Incrementa el punto en la dirección más larga e interpola el otro eje para suavizar la línea.
- **Algoritmo Punto Medio Círculo:** Usa un parámetro de decisión para elegir el próximo punto en la circunferencia, optimizando cálculos para sólo valores enteros y simetría de octantes.
- **Algoritmo Punto Medio Elipse:** Similar al círculo, pero calcula regiones diferenciadas para manejar radios x e y independientes.

Diseño del sistema: arquitectura, módulos, diagramas de flujo por algoritmo.

Arquitectura: Basada en OpenGL y GLUT para interfaz gráfica y eventos. El sistema mantiene vectores de figuras con sus atributos.

Módulos principales:

- Entrada usuario (mouse, teclado, menú)
- Renderizado (primitivas y escena)
- Gestión de historial
- Exportación imagen

Diagramas de flujo:

Para cada algoritmo:

1. Inicialización parámetros
2. Cálculo punto siguiente según condición
3. Dibujar píxel/segmento
4. Repetir hasta completar figura

Especificación del menú y atajos.

Menú contextual con opciones para:

- Tipo de figura (líneas, círculos, elipse)
- Color (negro, rojo, verde, azul, personalizado)
- Grosor (1, 2, 3, 5 px)
- Vista (mostrar/ocultar cuadrícula, ejes, coordenadas)
- Herramientas (limpiar lienzo, borrar última figura, exportar imagen)

→ Ayuda (mostrar atajos, acerca de)

Atajos de teclado:

- G: alternar cuadrícula
- E: alternar ejes
- C: limpiar lienzo
- Z: deshacer
- Y: rehacer
- S: exportar imagen PPM
- H: mostrar/ocultar ayuda
- A: mostrar/ocultar acerca de
- ESC: cerrar diálogo o salir

Casos de prueba y resultados (capturas con rejilla/zoom + discusión).

Se realizaron pruebas dibujando líneas con ambos algoritmos (directo/DDA), comparando precisión. Círculos y elipses se verificaron con radios y ejes variables.

Se capturaron pantallas con cuadrícula y zoom para validar la exactitud visual.

El historial fue probado con series de acciones confirmando correcto undo/redo.

Undo (Deshacer): Revertir la última acción realizada, por ejemplo, borrar una figura que se acaba de dibujar. Es como un "retroceso" en el historial de acciones.

Redo (Rehacer): Restaurar una acción que se deshizo con undo. Es como avanzar otra vez en el historial si decides que quieres mantener la acción revertida.

Exportación generó imagen PPM correcta abierta en visualizadores estándar.

Conclusiones y trabajo futuro.

El sistema cumple con los objetivos básicos de un CAD 2D educativo, incluyendo algoritmos fundamentales y funcionalidad interactiva. Se logró implementar un diseño modular facilitando ampliaciones futuras. El trabajo futuro incluye incorporación de figuras poligonales y curvas Bézier, interfaz gráfica avanzada, y mejoras en exportación a formatos modernos.

Referencias

-Foley, van Dam et al., "Computer Graphics: Principles and Practice", 1995.

-OpenGL Programming Guide (Red Book).

-www.opengl.org documentation.

-Recursos académicos internos y tutoriales oficiales GLUT.