Proyecto 1: Líneas rectas

Profesor: Ernesto Rivera Alvarado

I. Introducción

En este proyecto, Ud. deberá desarrollar y comparar algoritmos para dibujar líneas rectas. Toda la programación debe realizarse en C sobre Linux.

II. Las líneas

Este programa recibirá sus argumentos de ejecución de la línea de comando, de esta forma:

`<programa> <resolucion> <# lineas> <# veces>`

Al ejecutar, `<programa>` creará una ventana cuadrada de `<resolucion>` × `<resolucion>` píxeles. Después, generará `<# lineas>` líneas rectas aleatorias, donde cada línea es definida por dos puntos $[x_0, y_0]$ y $[x_1, y_1]$ escogidos dentro de la resolución establecida, estos pares de puntos serán conservados en una tabla en memoria.

Para cada uno de los algoritmos a ser evaluados, el programa dibujará las mismas `<# lineas>` líneas `<# veces>` veces y **tomará tiempos de ejecución con la mayor precisión posible (*i.e.*, ¿cuánto dura cada versión de cada algoritmo?)**.

III. Algoritmos

Para este proyecto, contrastaremos los algoritmos estudiados en clase:

- Algoritmo de Fuerza Bruta
- Algoritmo Incremental
- Algoritmo Incremental Versión 2
- Algoritmo Bresenham (Punto Medio)
- **Trabajo extra opcional 1:** Hacer una quinta versión con el Algoritmo de Bresenham programado en ensamblador.

Para la rutina **`plot(x,y)`** se harán 3 versiones. La primera es una rutina vacía que simplemente retorna (esta versión no desplegará nada y servirá para comparar tiempos de ejecución puros de los 4 algoritmos). La segunda guardará en disco la imagen de las líneas generadas usando formato PPM. La tercera será una rutina que despliega usando `glut / MESA / SDL / OpenGL / Vulkan`. Para cada uno de los casos se debe mostrar el tiempo de ejecución.

IV. Modos de despliegue

En las versiones que sí usan `glut / MESA` cada algoritmo desplegará todas las líneas en la ventana usando un color diferente y único. La ventana **NO** será limpiada entre un algoritmo y otro. Ya que un criterio mínimo de que el programa está bueno es que todos los algoritmos escojan exactamente los mismos píxeles, debería ocurrir que todos los algoritmos superpongan sus líneas sobre las del algoritmo anterior. Su programa hará una pausa entre un algoritmo y otro hasta que el usuario presione alguna tecla para continuar. Independiente del despliegue en la ventana gráfica, en la consola aparecerá el nombre del algoritmo ejecutándose.

Al final, su programa debe desplegar a consola (con mucha calidad) los resultados finales con los tiempos de los 4 algoritmos en sus 2 versiones (8 resultados).

Todos los proyectos serán ejecutados en clase. Es su responsabilidad verificar que su máquina con Linux nativo funciona para la revisión del proyecto.

V. Sobre la revisión y completitud del proyecto

La evaluación del proyecto se dividirá de la siguiente manera:

- Versión 1 de los 4 algoritmos implementada correctamente con comparación de los tiempos de ejecución 20%.
- Versión 2 de los 4 algoritmos implementada correctamente con comparación de los tiempos de ejecución 30%.
- Versión 3 de los 4 algoritmos implementada correctamente con comparación de los tiempos de ejecución 50%.

VI. Requisitos indispensables

La ausencia de uno solo de los siguientes requisitos vuelve al proyecto "no revisable" y recibe un 0 de calificación inmediata:

- Todo el código debe de estar escrito en C sobre Linux.
- Las funciones de OpenGL/Vulkan o SDL se limitan a crear una ventana, gestionar la interacción con el usuario (mouse y teclado) y apagar y encender pixeles en colores definidos.
- No debe presentarse "segmentation fault" por ninguna razón.
- La colaboración entre grupos se considera fraude académico.
- Todo el código debe estar escrito en C (no C++).
- El proyecto debe compilar y ejecutar en Linux. Todo debe estar **integrado**, explicaciones del tipo "*todo está bien pero no pudimos pegarlo*"[1] provocan la cancelación automática de la revisión.
- La presentación debe ser de mucha calidad.
- No debe dar "Segmentation Fault" bajo ninguna circunstancia.
- **La única interacción válida con el programa es por medio de argumentos de línea de comando.**
- Hacer la demostración en una máquina que levante Linux de manera real (puede ser dual), es decir no usar máquinas virtuales.
- La presentación debe hacerse en clase en la computadora del estudiante.

[1] esto incluye los supuestos casos cuando alguien del grupo de trabajo no hizo su parte – el profesor no está interesado en sus problemas de organización.

- Debe seguir a cabalidad lo descrito en la **Sección II** del presente documento.
- Los estudiantes deben demostrar dominio completo de su proyecto en la presentación oral.
- Los estudiantes deben utilizar el sistema de control de versiones Git tal como se describe en este documento.

La evaluación de este proyecto está sujeta a la presentación oral del mismo, en la que el estudiante debe mostrar un dominio completo del trabajo realizado a nivel de código. En caso de que el estudiante no muestre un dominio completo del trabajo, se considerará el trabajo como "no revisable" y se asignará la nota de cero. La defensa del proyecto será grabada.

En caso de trabajar en pareja, ambos integrantes deberán tener un dominio completo del trabajo que están presentando. El profesor podrá hacerle preguntas a cualquiera de los integrantes y este deberá contestar acordemente. En caso de que se evidencia que uno de los integrantes desconoce el funcionamiento de alguna parte del trabajo presentado, se le asignará la nota de cero a ambos.

VII. Desarrollo del proyecto

El proyecto está pensado para desarrollarse individualmente, sin embargo, los estudiantes que deseen reforzar la habilidad de trabajo en equipo pueden entregar el proyecto en parejas. El profesor les hace la aclaración de que el trabajo con un compañero conlleva dificultades de coordinación, división de trabajo y sobre todo de "pegar o juntar ambas partes". En experiencias propias del profesor, se les comenta que en ocasiones el trabajo de juntar, acoplar y corregir partes desarrolladas por diferentes personas conlleva más tiempo y trabajo que la realización individual.

VIII. Git

Los estudiantes deberán utilizar el sistema de control de versiones Git, específicamente a través de GitLab. Para ello configurarán desde sus máquinas Gitlab, para poder hacer cualquier acción (ejemplo *push*, *pull*, *commit*) desde **consola de texto**, sin la necesidad de ingresar la contraseña constantemente. Además agregarán al repositorio del proyecto al usuario Ernesto.cursos, donde el profesor podrá observar los avances y desarrollos del proyecto por semana, día, hora, minuto y segundo.

IX. Consideraciones sobre plagio

El trabajo presentado por los estudiantes (ya sea a nivel individual y de parejas) debe ser de su propia autoría. No se permite utilizar código realizado por un tercero (sin importar la licencia de dicho código) o entre compañeros del mismo curso, y en caso de que esto ocurra, será considerado plagio por lo que se seguirá el proceso correspondiente. Códigos encontrados en "github" de los cuales solo se tomaron "partes" será considerado plagio, el estudiante es el responsable de desarrollar absolutamente todo su código, a excepción del código que se menciona como base en el presente documento. De igual manera, cualquier código que el estudiante no tenga la capacidad de explicar, será considerado como plagio.

X. Fecha de entrega

Las demostraciones se harán en la **semana 5** asignadas por citas distribuidas aleatoriamente en las dos lecciones de la semana.

La carpeta comprimida `.zip` de su proyecto debe contener los archivos fuentes `.c` y complementarios, además del archivo `Makefile` que genera el ejecutable del código fuente. El nombre de la carpeta comprimida debe de ser los apellidos de integrantes del grupo de trabajo (por ejemplo `rivera-alvarado.zip`) y este será subido al correo del profesor y al Tec Digital el día de la revisión a antes de las 6 am. Los estudiantes deberán defender el trabajo realizado y mostrar un dominio completo del trabajo desarrollado. El no tener conocimiento de cómo funcionan partes del proyecto invalida la entrega y se asignará la nota de cero. El título del correo será [Proyecto 1 - Gráficos por Computadora].