Gráficos por Ordenador

Shaders de Teselación

Brais Lagoa Rúa y Juan Maruri Pérez - 11/05/2025

Entre las técnicas más potentes y visualmente impactantes que ofrece la programación gráfica moderna se encuentra la teselación. Esta técnica permite subdividir superficies en tiempo real, incrementando el nivel de detalle geométrico de un modelo sin necesidad de modificar su malla base original. Al hacerlo en la etapa del pipeline de renderizado, permite aplicar transformaciones geométricas sobre modelos simples, mejorando su apariencia sin incurrir en un gran coste de almacenamiento o complejidad inicial en el modelado.

Este proyecto se centra en el uso de shaders de teselación interactivos, con el objetivo de explorar su potencial para mejorar visualmente modelos 3D de baja resolución. En particular, se ha desarrollado una aplicación interactiva que permite controlar, en tiempo real, dos parámetros clave del proceso de teselación:

1. El nivel de subdivisión de los patches, que determina cuántas veces se subdivide la geometría original, aumentando su resolución geométrica y permitiendo una representación más detallada del objeto.

2. La intensidad de deformación del relieve, que modifica la altura de los vértices tras la subdivisión según una función definida, generando variaciones en la superficie que simulan irregularidades o volúmenes añadidos al modelo base.

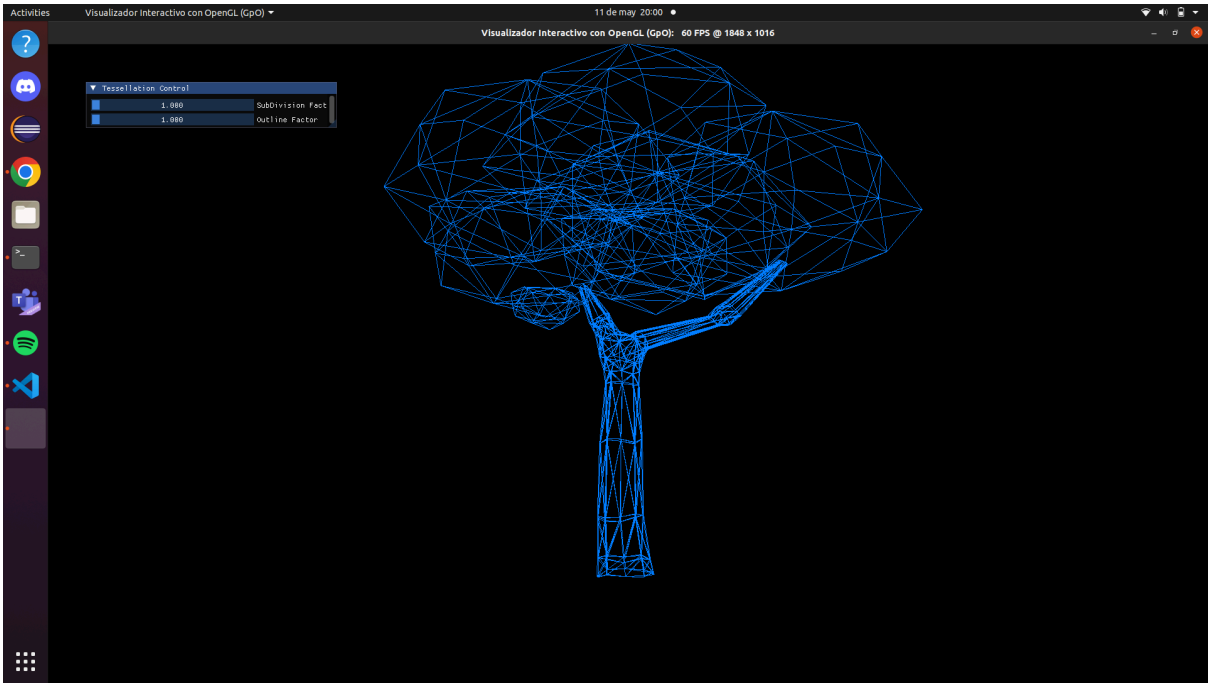
Como caso práctico, se ha seleccionado un modelo de árbol low poly, una figura sencilla pero representativa, con el fin de comprobar cómo la teselación puede aportar riqueza visual mediante el incremento del detalle geométrico y la generación de relieves dinámicos. La idea principal es transformar un modelo simple en uno más complejo y orgánico, sin alterar la malla original, aprovechando exclusivamente las capacidades del pipeline gráfico y de los shaders.

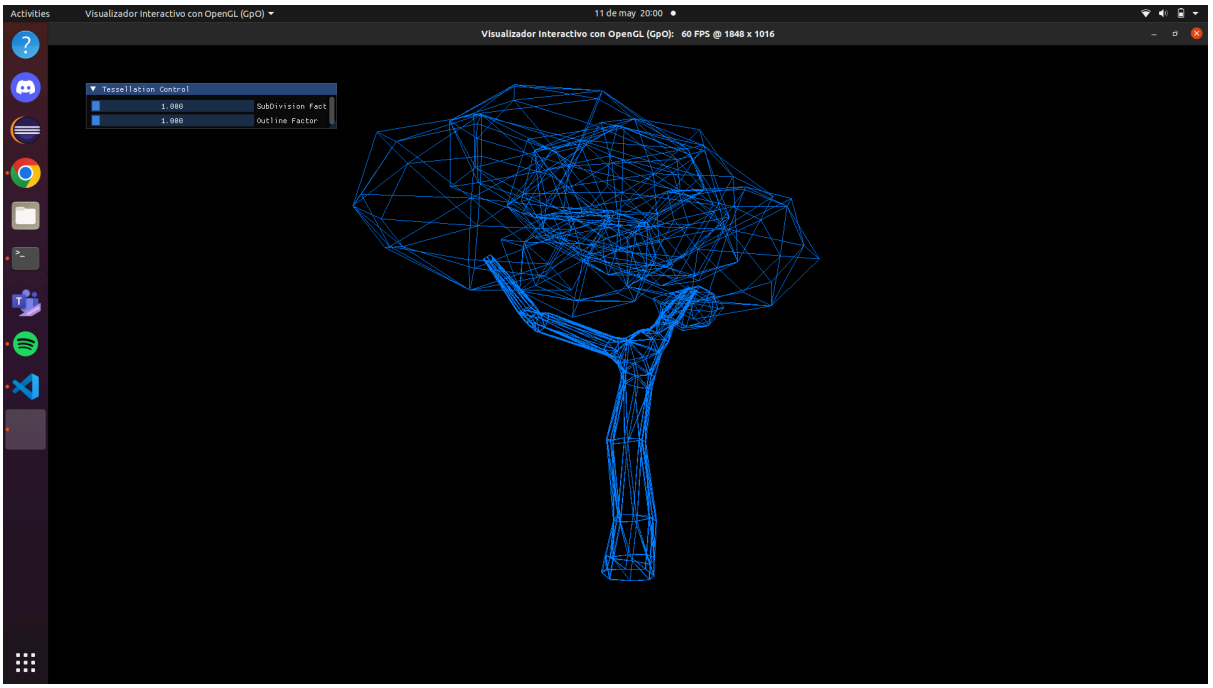
Durante las primeras fases del desarrollo fue necesario modificar manualmente el archivo `.obj` correspondiente al modelo del árbol, ya que presentaba múltiples errores. En concreto, las caras no estaban correctamente definidas, las normales estaban mal configuradas o no normalizadas, y al cargar las texturas se generaban numerosos problemas de visualización. A pesar de realizar múltiples pruebas —incluyendo la corrección de normales, la reasignación de coordenadas UV y el uso de distintos formatos de textura—, no se consiguió una visualización adecuada ni del tronco ni de la copa del árbol. Por este motivo, se optó por mostrar el modelo en modo *wireframe*, lo que se reveló como una ventaja desde el punto de vista experimental. Esta representación facilita la observación del efecto del primer parámetro interactivo —el nivel de subdivisión—, permitiendo ver con claridad cómo cambia la geometría del modelo a medida que se ajusta la densidad de los *patches*.

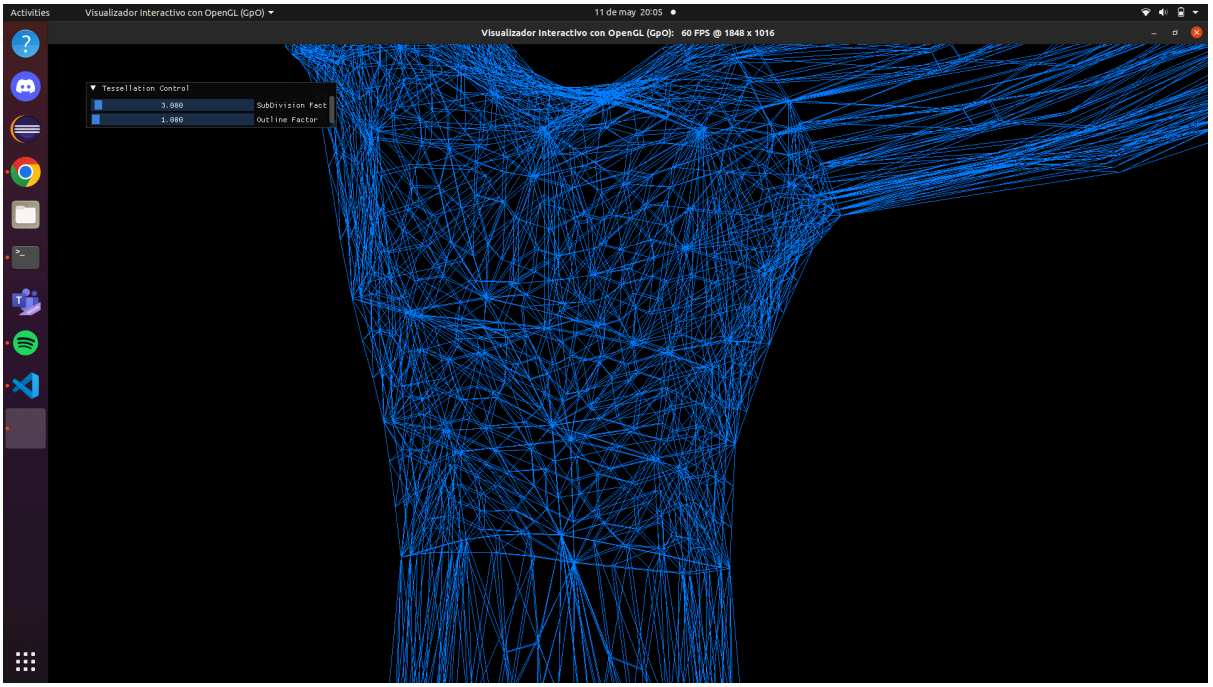
Para implementar el control interactivo de la subdivisión, se ha utilizado una variable global en el programa principal, que se actualiza mediante un callback vinculado a la interfaz gráfica de usuario, como una barra deslizante. Esta variable representa el nivel deseado de subdivisión y se envía al Tessellation Control Shader (TCS) mediante una *uniform*. Dentro del TCS, se utiliza para ajustar los factores de teselación interna y externa de los *patches*, lo que determina cuántos triángulos se generarán a partir de cada patch. De este modo, el usuario puede modificar en tiempo real el grado de subdivisión del modelo, obteniendo una visualización más o menos detallada según se desee, sin necesidad de recompilar el shader o reiniciar la aplicación. Esta capacidad de ajuste fluido permite experimentar rápidamente con distintos niveles de complejidad geométrica, lo que resulta especialmente útil para comprender visualmente el impacto de la teselación.

El segundo parámetro interactivo, relacionado con la deformación del relieve, se gestiona con una técnica similar. Se define otra variable global, también actualizada mediante un callback de la interfaz gráfica, y se pasa como *uniform* al Tessellation Evaluation Shader (TES). En este shader se manipulan las coordenadas de los vértices generados por la teselación, concretamente su componente vertical (o en la dirección normal a la superficie). Para ello, se puede aplicar una función procedural de desplazamiento —por ejemplo, basada en las coordenadas del modelo o en una textura de ruido— que modifique la altura de cada vértice evaluado. Esta deformación no afecta al modelo base, sino que se aplica al representar los vértices después de la subdivisión, generando así una ilusión visual de relieve o volumen que puede ajustarse en tiempo real.

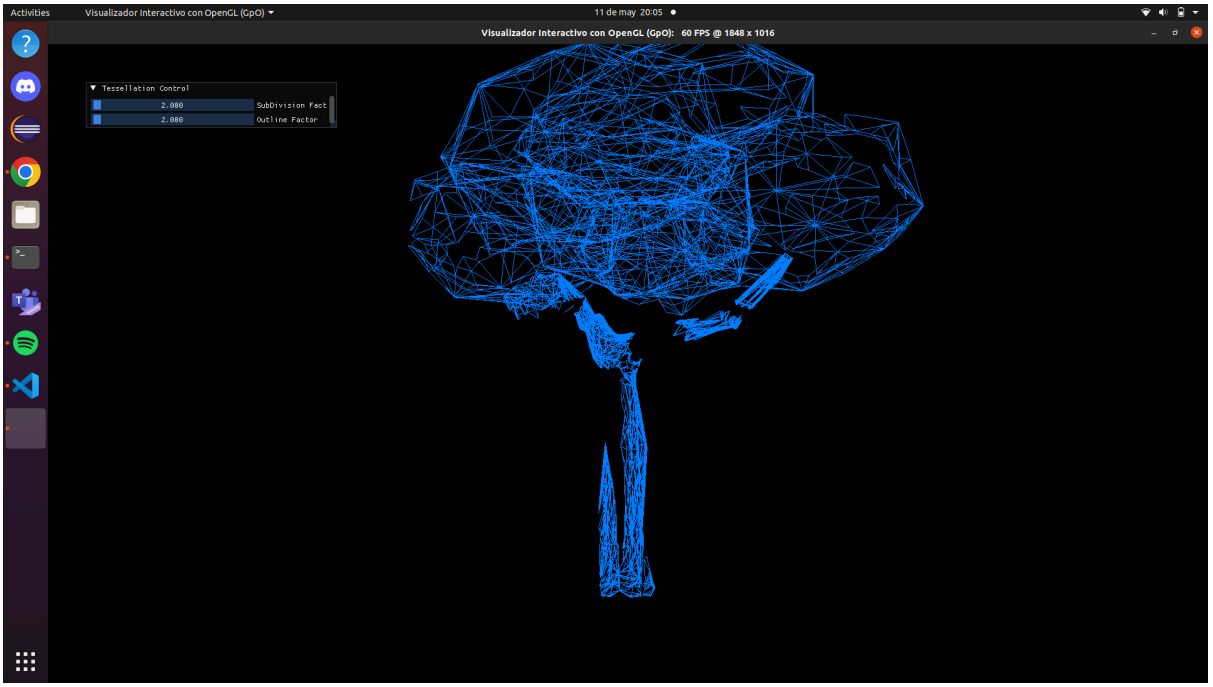
Este enfoque demuestra cómo el pipeline de teselación permite enriquecer modelos simples aplicando efectos geométricos avanzados sin penalizaciones importantes en el rendimiento, dado que la complejidad se gestiona en la GPU y de forma dinámica. Además, ofrece un punto de partida para explorar técnicas más sofisticadas como el *displacement mapping*, la teselación adaptativa o la generación procedural de superficies, todas ellas con aplicaciones en videojuegos, simulaciones y visualización científica.

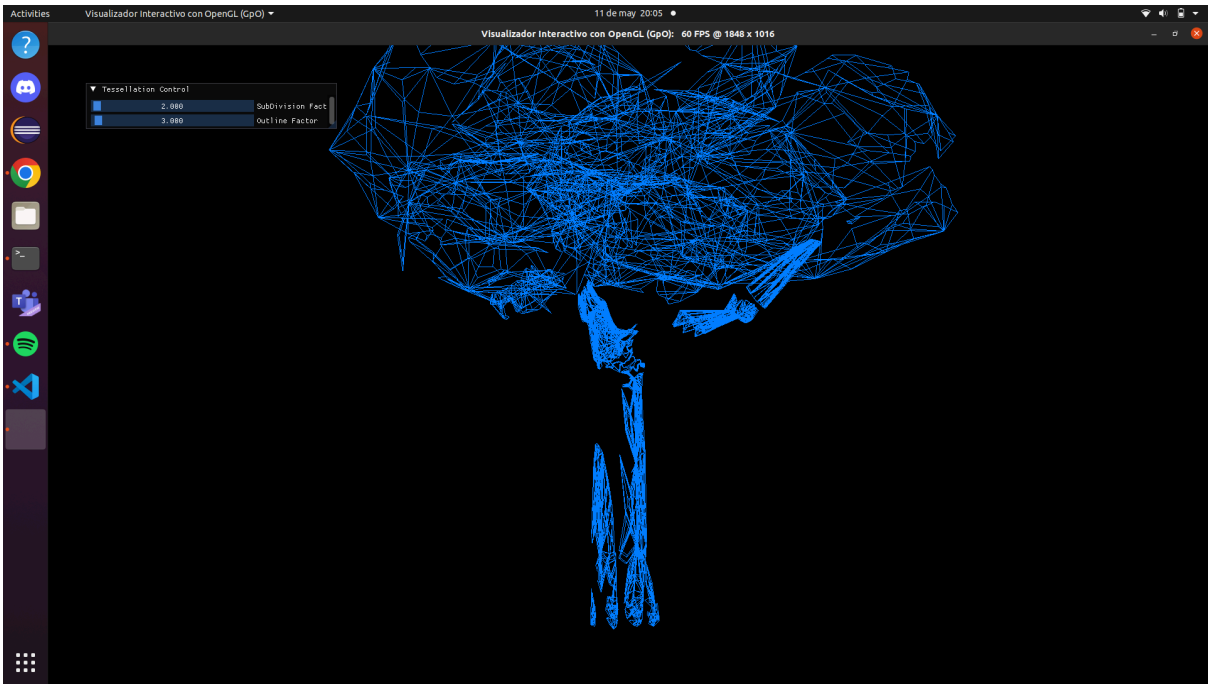

Activities
Visualizador Interactivo con OpenGL (GpO)
11 de may 20:00
Visualizador Interactivo con OpenGL (GpO): 60 FPS @ 1848 x 1016
Tessellation Control
1.000
SubDivision Fact
1.000
Outline Factor

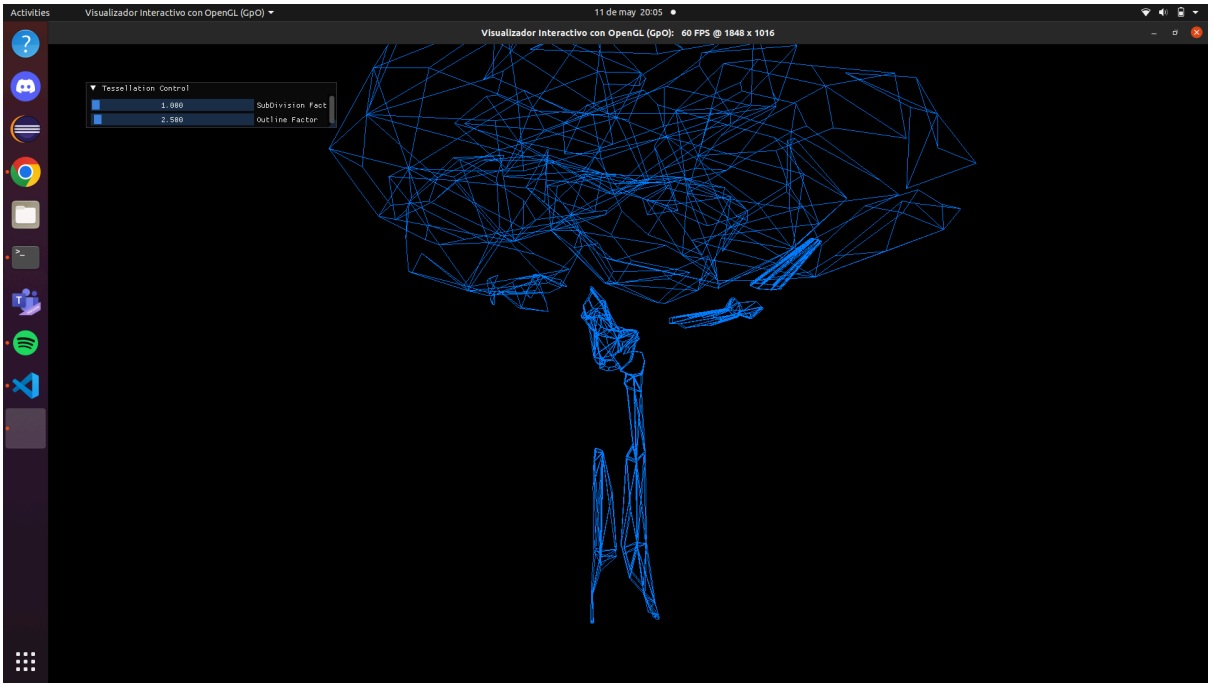

Activities
Visualizador Interactivo con OpenGL (GpO)
11 de may 20:00
Visualizador Interactivo con OpenGL (GpO): 60 FPS @ 1848 x 1016
Tessellation Control
1.000
SubDivision Fact
1.000
Outline Factor

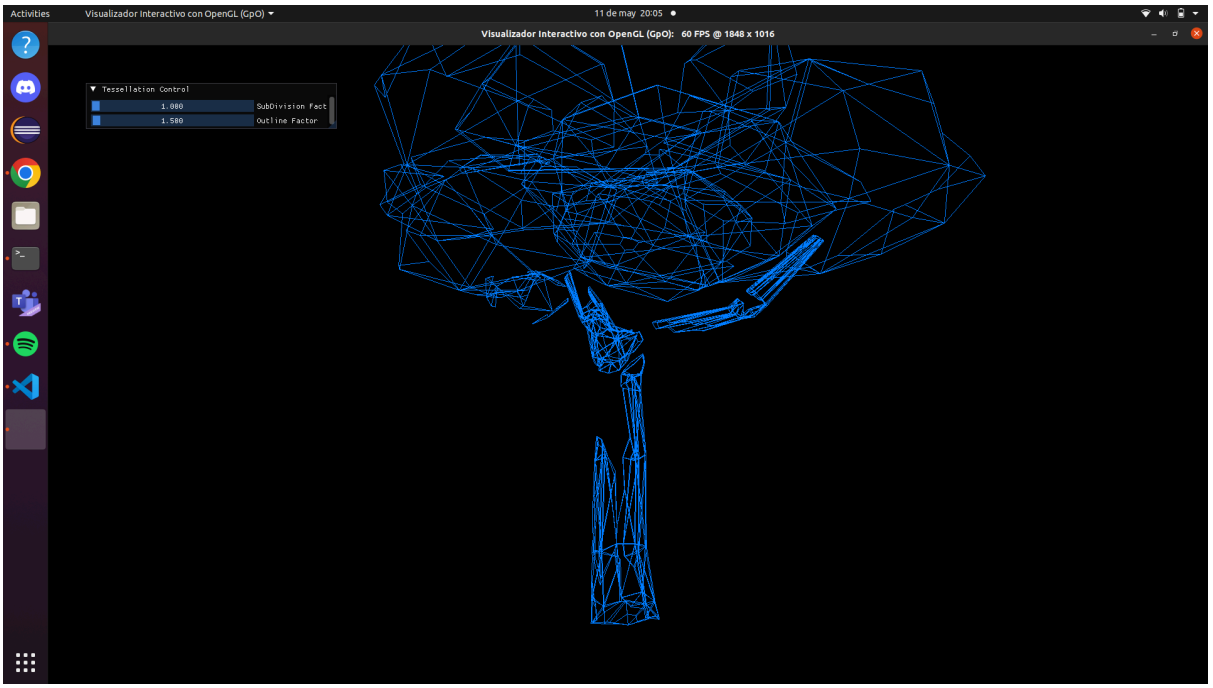

Activities
Visualizador Interactivo con OpenGL (GpO)
11 de may 20:05
Visualizador Interactivo con OpenGL (GpO): 60 FPS @ 1848 x 1016
Tessellation Control
3.000
SubDivision Fact
1.000
Outline Factor

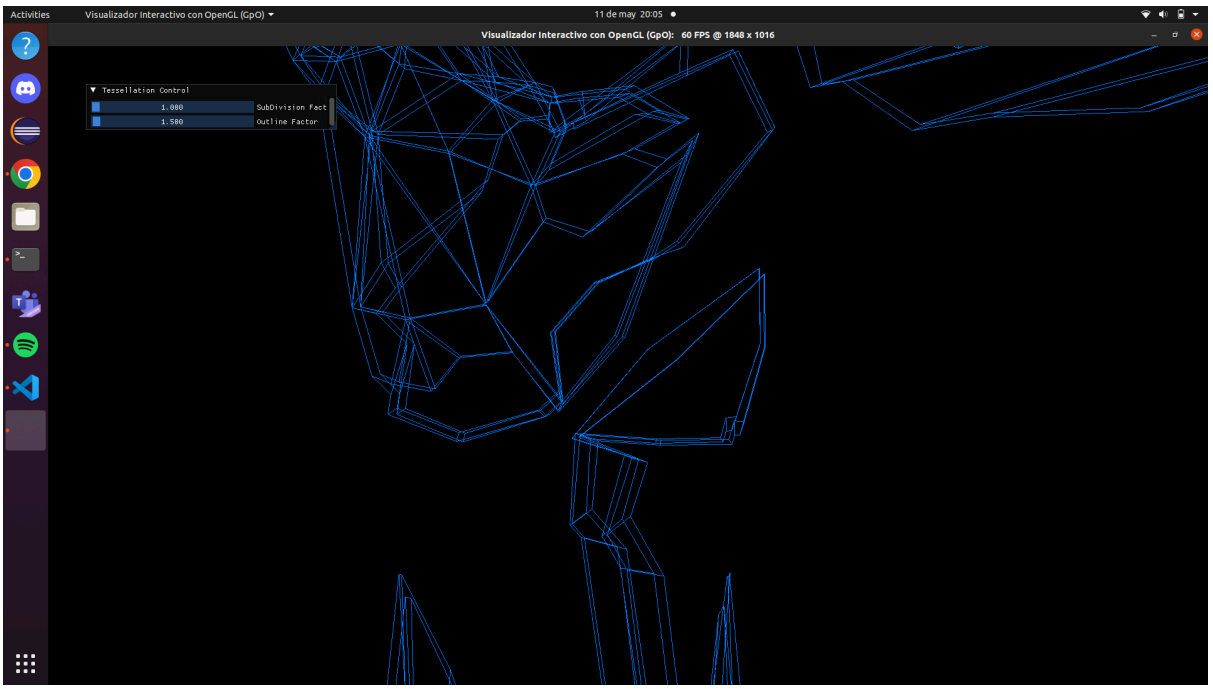

Activities
Visualizador Interactivo con OpenGL (GpO)
11 de may 20:04
Visualizador Interactivo con OpenGL (GpO): 60 FPS @ 1848 x 1016
Tessellation Control
3.000
SubDivision Fact
1.000
Outline Factor

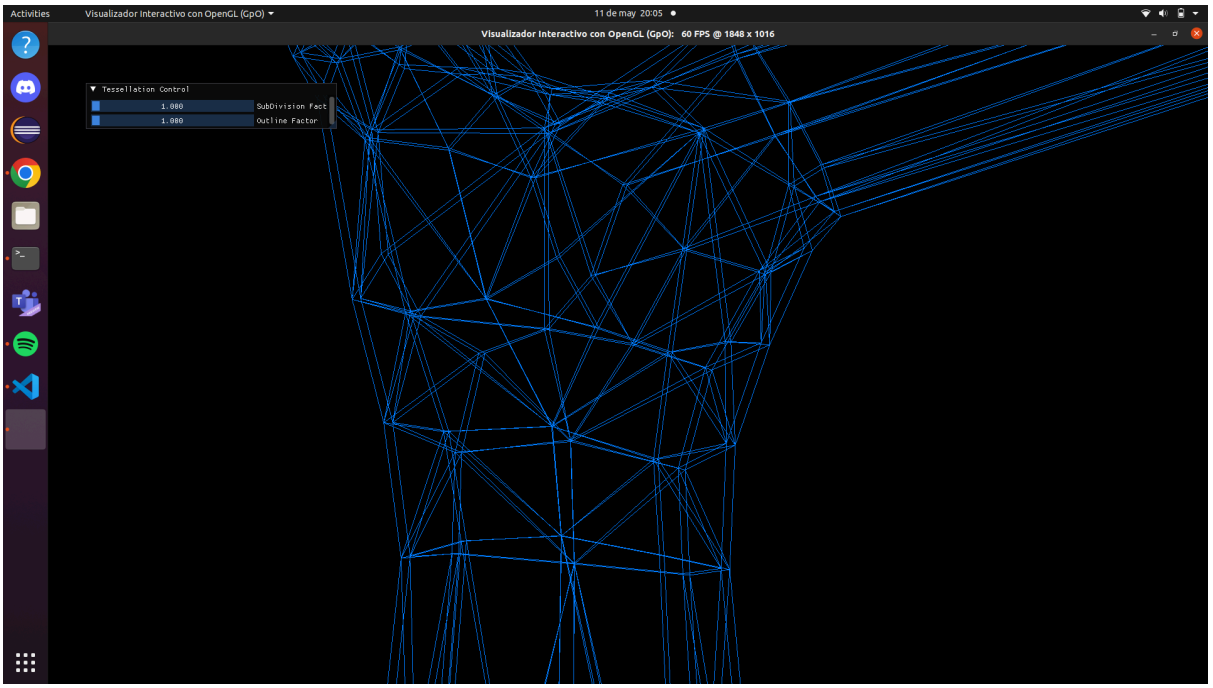

Activities
Visualizador Interactivo con OpenGL (GpO)
11 de may 20:05
Visualizador Interactivo con OpenGL (GpO): 60 FPS @ 1848 x 1016
Tessellation Control
2.000
SubDivision Fact
2.000
Outline Factor

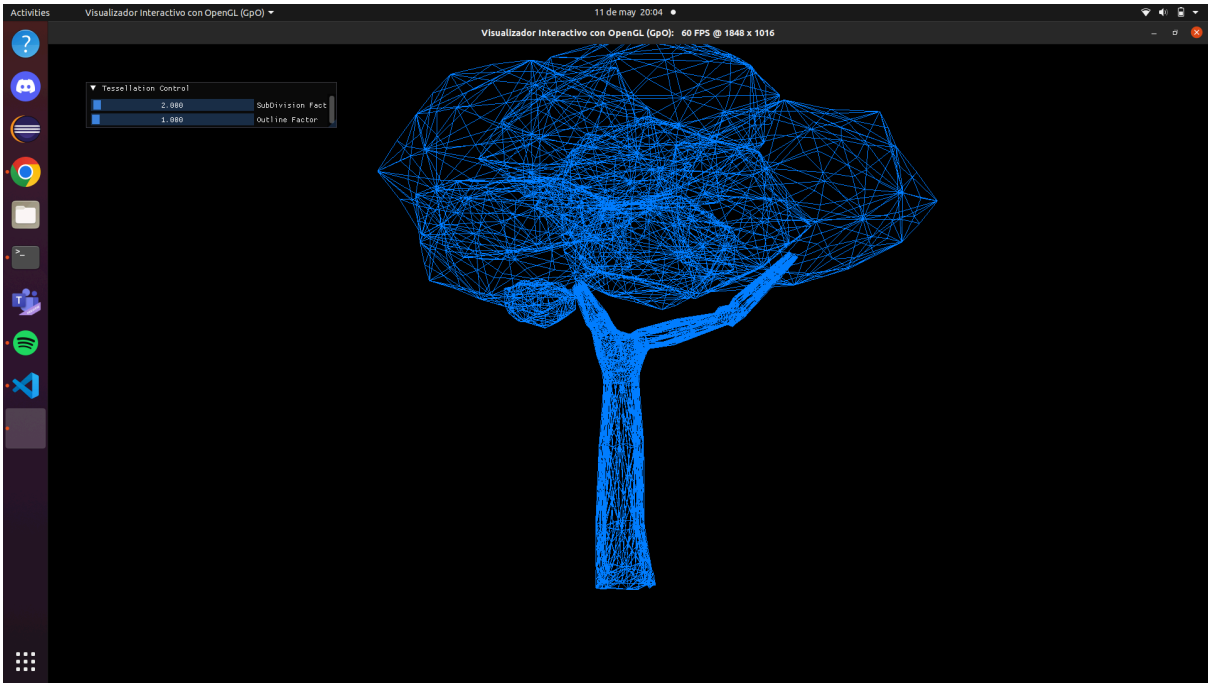

Activities
Visualizador Interactivo con OpenGL (GpO)
11 de may 20:05
Visualizador Interactivo con OpenGL (GpO): 60 FPS @ 1848 x 1016
Tessellation Control
2.000
SubDivision Fact
3.000
Outline Factor

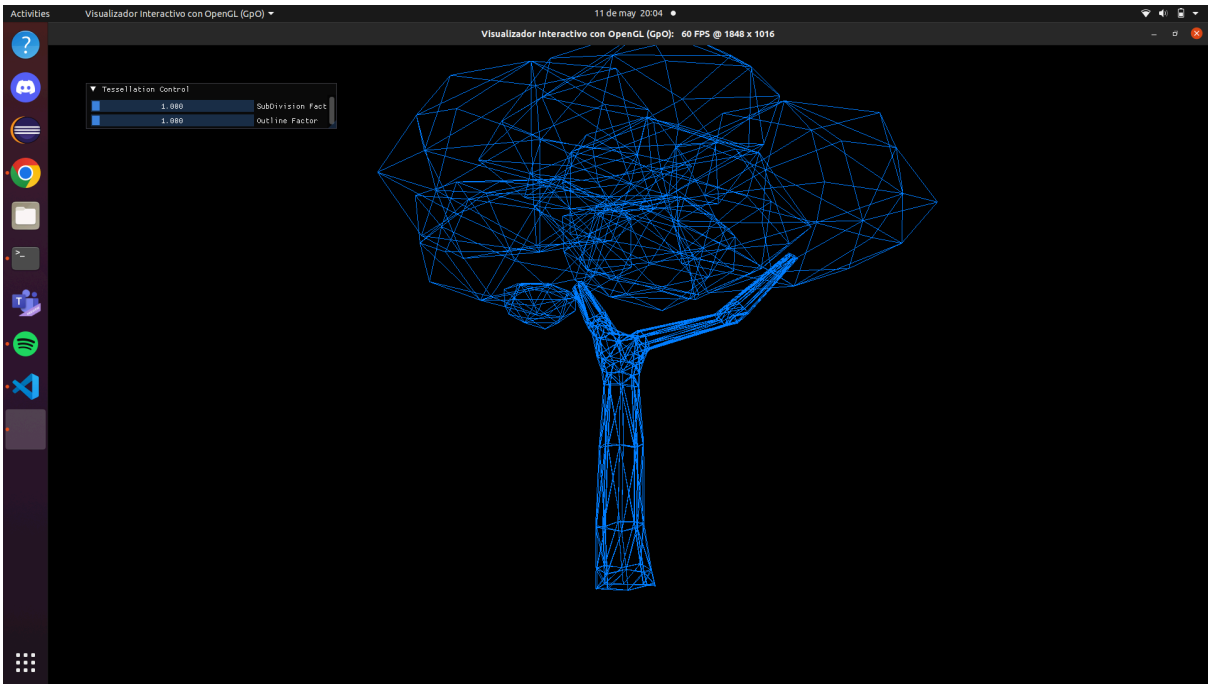

Activities
Visualizador Interactivo con OpenGL (GpO)
11 de may 20:05
Visualizador Interactivo con OpenGL (GpO): 60 FPS @ 1848 x 1016
Tessellation Control
1.000
SubDivision Fact
2.500
Outline Factor


Activities
Visualizador Interactivo con OpenGL (GpO)
11 de may 20:05
Visualizador Interactivo con OpenGL (GpO): 60 FPS @ 1848 x 1016
Tessellation Control
1.000
SubDivision Fact
1.500
Outline Factor


Activities
Visualizador Interactivo con OpenGL (GpO)
11 de may 20:05
Visualizador Interactivo con OpenGL (GpO): 60 FPS @ 1848 x 1016
Tessellation Control
1.000
SubDivision Fact
1.500
Outline Factor


Activities
Visualizador Interactivo con OpenGL (GpO)
11 de may 20:05
Visualizador Interactivo con OpenGL (GpO): 60 FPS @ 1848 x 1016
Tessellation Control
1.000
SubDivision Fact
1.000
Outline Factor


Activities
Visualizador Interactivo con OpenGL (GpO)
11 de may 20:04
Visualizador Interactivo con OpenGL (GpO): 60 FPS @ 1848 x 1016
Tessellation Control
2.000
SubDivision Fact
1.000
Outline Factor


Activities
Visualizador Interactivo con OpenGL (GpO)
11 de may 20:04
Visualizador Interactivo con OpenGL (GpO): 60 FPS @ 1848 x 1016
Tessellation Control
1.000
SubDivision Fact
1.000
Outline Factor