

# Proyecto de Computación Gráfica

ALEJANDRO KLEVER CLEMENTE C411

LAURA TAMAYO BLANCO C411

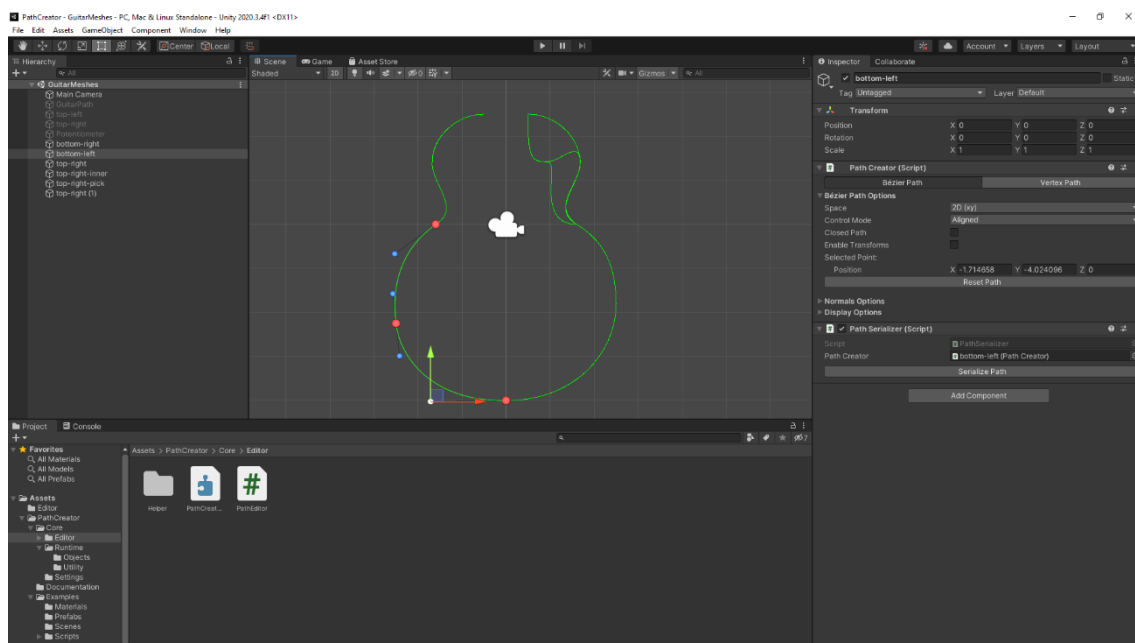
## Proceso de modelado:

El proceso de modelado fue el más complicado debido a la complejidad de los elementos de la escena. El cuerpo de la guitarra está compuesto de pequeños modelos que se intentaron reproducir de la forma más cercana posible a la realidad posible. Para separar estas tareas usamos un esquema simple de composición, creamos una clase `SceneObject` que contendría la información de las mallas que componen este objeto y sus materiales. En complemento con una clase `Transform` podemos definir posición, escala y rotación un objeto padre para aplicar las transformaciones del padre a sus hijos.

Modelamos cada pieza del cuerpo de guitarra con una clase que hereda de `SceneObject`. Así factorizábamos el código y era más fácil complementar las distintas partes de la escena.

Para el modelado de la guitarra usamos Splines de Bézier, para aprovechar todo el potencial de esta estructura para hacer curvas más complejas. Este spline de Bézier es una concatenación de segmentos cúbicos de Bezier y para evaluarlos usamos el polinomio cubico de Bezier. Para calcular las normales de la curva usamos la primera y segunda derivada de este polinomio.

Para no trabajar a ciegas hicimos uso de una implementación antigua que teníamos de Curvas de Bézier en Unity y usamos el motor para diseñar las curvas más complejas:



## Redondeando Cubos:

Como parte del modelado necesitábamos crear cubos con los bordes redondeados. Para eso decidimos primeramente usar un algoritmo de subdivisión de mallas. Loops era el algoritmo ideal para mallas triangulares, pero tiene la desventaja que convierte al cubo en una esfera, Catmull-Clark tenía un efecto similar, solo que tuvimos que agregar el concepto de quads a la malla, donde un quad está formado por 4 vértices, o sea, 2 triángulos. Así que el número de triángulos debía ser par y se consideraba que 2 triángulos consecutivos eran un quad. Pero como el resultado era muy similar al de Loops seguimos investigando. Finalmente dimos con el algoritmo de Doo-Sabin pero resolvía nuestro problema... pero su implementación nos fue costosa con el sistema de mallas basado en triángulos. Finalmente encontramos una solución en un tutorial de Unity3D donde se obtenía el resultado deseado aplicando un redondeo en los

vértices de los bordes de un cubo y atrayéndolos al centro del mismo en la dirección de la normal entre ese vértice y el correspondiente a un cubo interior de un tamaño menor, pero igual proporcionalidad en sus dimensiones.

Una vez terminado esto solo necesitábamos armar la escena, que si fue el proceso más costoso por la construcción de cada pieza de la guitarra. En cuanto a la iluminación y materiales no hubo mucha complicación ya que la escena a imitar no tenía un nivel de detalle muy elevado en cuanto luces se trataba, aunque no pudimos conseguir las él sombreado de las paredes como aparece en el de la foto original.

## Organización del proyecto:

El proyecto esta tiene todo organizado en una carpeta llamada Scene. Se intentó no tocar las clases inicialmente otorgada, y en su lugar decidimos extender los funcionamientos creando nuestros propios tipos.

La carpeta de Textures tiene todas las texturas usadas en este proyecto y al momento de ejecutar el programa estas son copiadas en la carpeta del ejecutable.

En la raíz del proyecto se encuentran los resultados del raytracing y el pathtracing (es probable que la luz no esté bien ajustada, pedimos disculpas por esto)