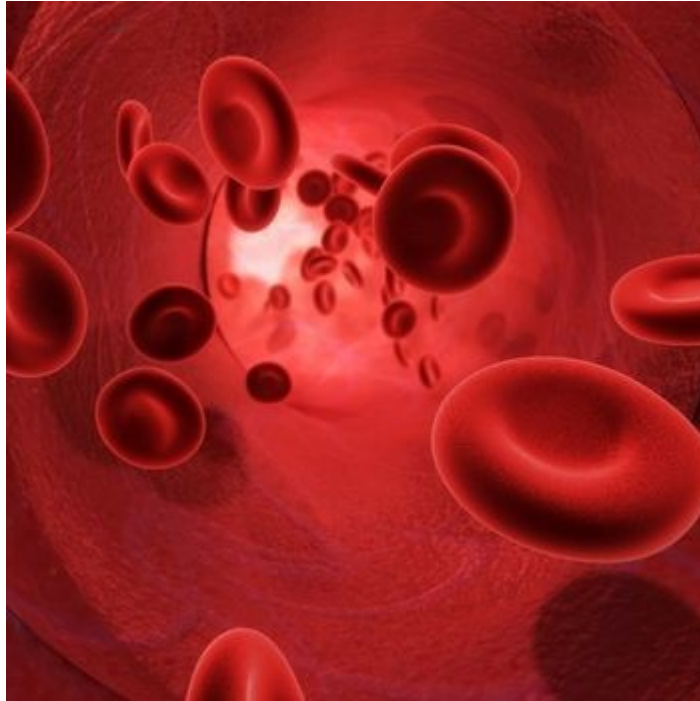


Práctica 1: El viaje alucinante

--- Adaptado del material de Alexei Efros
(Carnegie Mellon University)



Objetivo de la práctica

El objetivo de esta práctica consiste en reforzar los conceptos de OpenGL que se han introducido en la asignatura de Graficos 3D y explorar las técnicas que se han estudiado en Rendering Avanzado para iluminación local; para lograrlo, aplicaremos estos conocimientos en la **simulación de un recorrido a través del sistema circulatorio humano por el torrente sanguíneo** (que requerirá la realización de dicho entorno mediante geometría y texturizado).

Los objetivos principales de esta práctica son que el alumno aprenda a:

- Iluminación y sombreado en OpenGL
- Utilizar al menos una técnica en la que la información de una textura no codifique simplemente el color a aplicar, sino información que afecte a la geometría de la escena.
- Aplicación de técnicas de renderizado sobre texturas
- Manejo y animación de la cámara
- Utilización de funciones para definir curvas (spline / bezier...)

Con todo ello se compondrá en OpenGL el escenario de un conjunto de capilares en un recorrido cerrado en la que se observen las imperfecciones de los vasos sanguíneos, en el que el visitante virtual podrá deslizarse a modo de tobogán tubular gigantesco.

Todo lo que en este guión se propone como tareas de esta práctica es susceptible de mejora por parte del alumno, lo que se valorará positivamente en la nota de la práctica.

Planteamiento

La presente práctica consta de 3 partes obligatorias, que se corresponden con los objetivos mínimos, y una serie de mejoras opcionales que pretenden dar libertad y estimular la creatividad de los alumnos de la asignatura.

Sois libres de utilizar el lenguaje de programación que queráis, pero sólo vamos a dar soporte a las dudas de código en lenguaje C y OpenGL, y tenéis que tener en cuenta que el código resultante tiene que compilarse sin problemas en el ordenador de referencia (Pentium Quad Core con Windows 7, 16Gb RAM, OpenGL 4.5, Visual Studio 2012, GeForce 570 GTX).

IMPORTANTE:

- En esta práctica habrá que entregar una memoria escrita en la que se describa su realización, el planteamiento del código escrito y se incluyan ejemplos de capturas de pantalla.
- Es necesario incluir el código fuente (sin los ficheros generados automáticamente por Intellisense).
- Será absolutamente **obligatorio** comentar los resultados obtenidos.
- Se debe incluir un vídeo corto con el resultado de las renderizaciones (preferentemente como un enlace a YouTube con el video).

Lectura recomendada

En Rendering Avanzado se presupone un cierto conocimiento de OpenGL, que debería haber sido adquirido en la asignatura de Gráficos 3D, así como cierta base en programación en C (en asignaturas de programación del curso 0).

Si tenéis dudas en esto temas podéis consultar los libros de la bibliografía

- Introducción a OpenGL, de José Luis Bosque, Susana Mata, Marcos García y Sofía Bayona
[http://www.dac.escet.urjc.es/rvmaster/asignaturas/IG/practicas/Apuntes_docentes_OpenGL.pdf]
- El libro Rojo de OpenGL [<http://fly.cc.fer.hr/~unreal/theredbook/>]
- Real Time Rendering, Tomas Akenine Möller, 3ª edición. AK. Peters
- 3D computer graphics: a mathematical introduction with OpenGL *Buss, S.R.* Cambridge U. Press
- Interactive computer graphics: A top-down approach with OpenGL *Angel, E.* Addison Wesley
- OpenGL Bible, R. S. Wright, 6ª edición (2013). Addison-Wesley Professional

Tabla de contenido

Objetivo de la práctica.....	1
Planteamiento	2
Lectura recomendada.....	2
Normativa del laboratorio.....	3
1. Parte 1: Mapas / Campos de altura e irregularidades en las paredes de las arterias.....	4
1.1. Irregularidades en las paredes de los vasos sanguíneos	5
2. Parte 2: Splines y los vasos sanguíneos.....	6
2.1 Código de base	7
2.2 Animación de la cámara	7
3. Parte 3: Mutación	7
4. Requerimientos de la práctica.....	8
4.1 Mínimo exigido (parte obligatoria)	8
3.2 Parte “opcional” (sólo si se cursa la asignatura completa)	8
3.3 Posibles extras	8
3.3 Para los muy muy valientes	9
Recomendaciones	9
Forma y fecha de entrega.....	10
Nota aclaratoria.....	10

Normativa del laboratorio

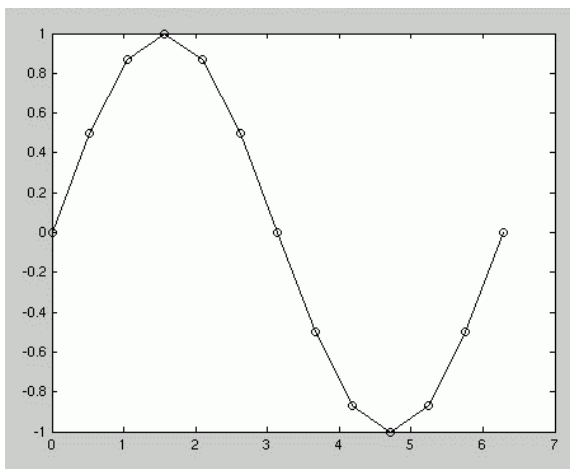
- El criterio más importante es la funcionalidad: un programa que funciona siempre tiene más posibilidades de llevarse una buena puntuación; no se valorarán aquellos programas que no funcionen. Una práctica o proyecto modesto será evaluada mucho más favorablemente que un "proyecto" ambicioso que sólo da *core-dumps*. Los siguientes criterios que se tendrán en cuenta (y que hay que cuidar al realizar las prácticas) son:
 - La manera de resolver el problema con el programa
 - Estructuras de datos y diseño de los algoritmos
 - Claridad y documentación en el código
 - Eficiencia y elegancia en la implementación.
- ¡Por favor, no hagáis trampas! Se procura alentar el diálogo y el trabajo en equipo, pero por favor trabajad de forma independiente (a menos que el trabajo sea en grupos). Trabajos muy similares serán considerados como copias, a menos que la naturaleza de lo pedido sea tan restrictiva que justifique las similitudes. Y una copia implica el suspenso automático. Simplemente piénsalo de esta manera: hacer trampas dificulta el aprendizaje y la diversión de conseguir hacerlo. Es vuestra responsabilidad proteger vuestro trabajo y asegurarnos que no se convierte en el de otro.
- Si se utiliza (o mejora) código fuente u otro material obtenido a través de internet, la biblioteca... debe darse el crédito a los autores y pedir permiso de ser necesario (si tiene una licencia restrictiva). Tomar código de un libro o de internet sin tener en cuenta estas consideraciones será considerado copia.

- Está terminantemente prohibido la práctica de técnicas de *overclocking* en las tarjetas gráficas del laboratorio, así como desbloquear los procesadores de los chips gráficos. Este tipo de acciones pueden dañar físicamente el equipo del laboratorio y los alumnos responsables serán amonestados severamente.
- Las prácticas (código y memoria explicativa) deberán entregarse en los plazos indicados mediante la herramienta correspondiente en el Campus Virtual.

1. Parte 1: Mapas / Campos de altura e irregularidades en las paredes de las arterias

Los campos de altura (*height fields*) son muy habituales en gráficos por ordenador, tanto en visualización científica como en la creación de terrenos en videojuegos. Una de sus aplicaciones más típicas es como mapas de normales en efectos de *bumpmapping* y *displacement mapping*.

En esta práctica los vamos a utilizar para generar la geometría de la rugosidad de los vasos sanguíneos en el que nos moveremos con la aplicación. Para ello partiremos de una textura ya creada ó bien de una textura en la que previamente se ha creado un efecto de plasma ó ruido Perlin como el explicado en clase cuando vimos las texturas procedurales.



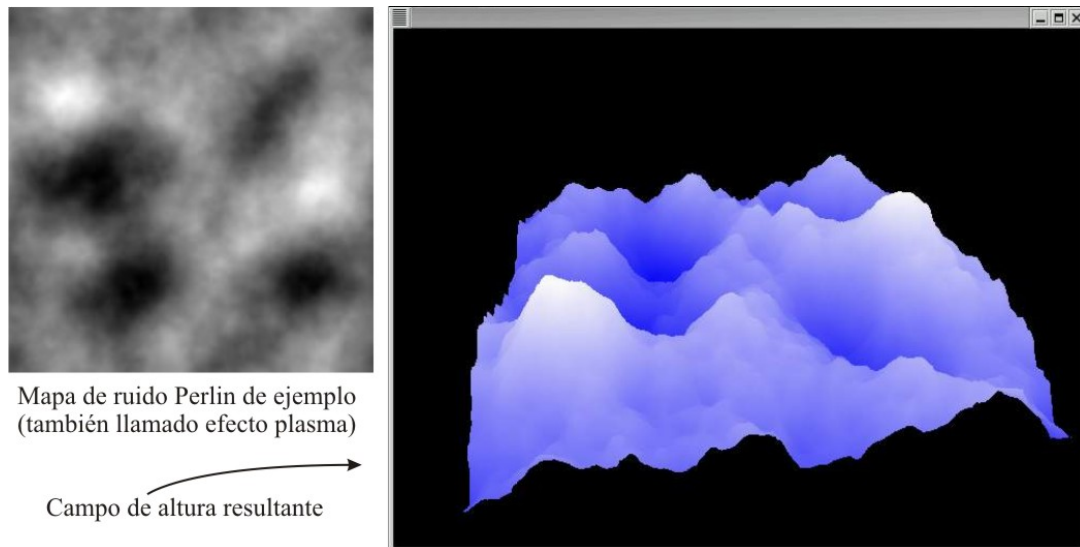
Un campo de altura es una función que toma como entrada una coordenada bidimensional (x e y) y nos devuelve un tercer valor que representa la distancia ortogonal a la superficie inicial.

$$z = f(x,y)$$

El caso general es un poco complicado, así que supondremos que nuestra función está muestreada de forma uniforme y que aproximamos la superficie de forma lineal.

En la figura de la izquierda podemos ver una aproximación en una sola dimensión

Con dos dimensiones tendremos una superficie bidimensional con triángulos en un espacio 3D como los que se ven en las figura inferior (generada a partir de un plasma ó un mapa de ruido Perlin).

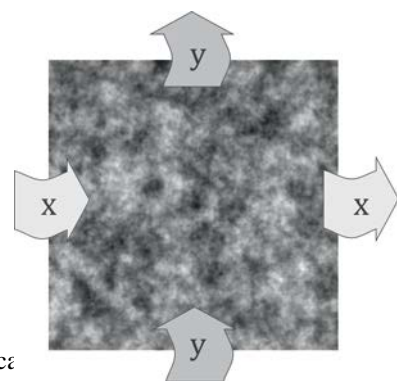


La imagen base con ruido Perlin será generada “al vuelo” en cada ejecución del programa (si fijáis la semilla del generador de números pseudoaleatorios, podéis asegurar que el ruido generado será siempre el mismo)¹. La textura debe ser generada con código creado por los alumnos. A la hora de colorear, los valores de las componentes rojo, verde y azul pueden ser procesados de la manera en el que el alumno considere, pero todos los valores de gris (aquellos en los que las tres componentes son idénticas) deben tener una transformación monótonica².

Dado que en esta asignatura no se centra en la programación de *shaders*, la conversión del mapa de alturas a su correspondiente visualización como mapa de alturas puede realizarse como un preproceso en el que se genere geometría o se modifique la ya existente en un array de vértices. Aquellos alumnos que deseen aplicar los conocimientos adquiridos en la asignatura de Procesadores Gráficos y Aplicaciones en Tiempo Real pueden tratar de abordar técnicas más sofisticadas como el *displacement mapping real* con autooclusiones ó *shaders* de geometría a su propio riesgo, ya que la dificultad de estas técnicas excede lo que se pretende ilustrar con esta práctica.

1.1. Irregularidades en las paredes de los vasos sanguíneos

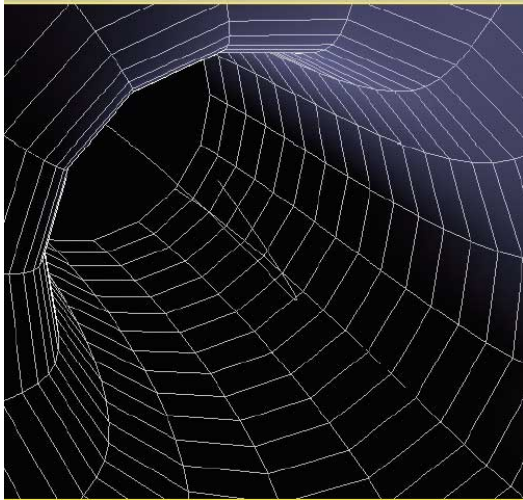
Para que el ruido de Perlin que va a definir las protuberancias tenga un parecido a las imperfecciones de las paredes de una arteria (debidos a la musculatura de los vasos), es necesario definirla de forma cilíndrica. Para ello, los distintos mapas que utilizamos para generar la texturas del ruido de Perlin deben estar lógicamente conectados en sus extremos de forma toroidal, al hacer el cálculo de la



¹ Para facilitar la depuración de programa suele ser una buena práctica probar a cargar una textura de un fichero para comprobar que el mapa se genera correctamente. Pero en la entrega, el mapa del ruido de Perlin debe generarse en tiempo de ejecución.

² La forma más sencilla de lograr este efecto es cogiendo uno sólo de los canales y operar sólo con él (mediante un escalado lineal).

interpolación bilineal. De esa manera se consigue que la textura no tenga “costuras” (seamless).



La geometría extruida al introducir los mapas de altura creada puede colorearse a nivel de vértice de acuerdo con un degradado de colores que sea función de la altura como se ve en el dibujo de la derecha, ó algún algoritmo más sofisticado que nos permita controlar algunas características. Con ello podemos lograr efectos interesantes, como por ejemplo: que se entrevean pegotes de colesterol, dar cierta transparencia y color blanquecino a los vasos, cicatrices, e incluso las válvulas presentes en las venas.

El rasterizador se encargará de que las transiciones de color sean suaves al interpolar los

valores de color de cada vértice.

Dado que la cavidad del túnel que forma la arteria es función de la textura inicial, conoceremos de antemano la altura del terreno y podemos situar pequeños elementos (*quads*) con texturas transparentes que simulen elementos ambientales en el medio líquido, como por ejemplo: plaquetas, azúcares, virus.... Estos elementos, normalmente conocidos como *impostors*, se suelen colocar de forma vertical a la pared del túnel y perpendicular a la cámara (de manera que ofrezcan la mayor superficie posible hacia el espectador) y tienen una textura mapeada sobre ellos en los que parte del dibujo es transparente. Estos elementos pueden ser colocados de forma más o menos aleatoria y elegir un tipo u otro y su tamaño en función de la irregularidad del terreno.



Opcionalmente, además se puede aplicar un mapa procedural celular para dar un aspecto más realista a las paredes de los vasos sanguíneos, que quedaría superpuesto a la paleta degradada de colores debida a las protuberancias del ruido de Perlin.

Se recomienda el uso de la librería GLUT, GLFM y GLM.

2. Parte 2: Splines y los vasos sanguíneos

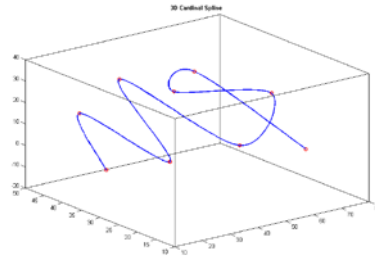
Para crear la trayectoria que siguen los vasos sanguíneos y el camino del observador es necesario renderizar una secuencia de splines³ que recorran la



³ También es posible llevarlo a cabo con curvas cuya descripción es más sencilla, como las de Bezier.

longitud del camino a seguir. En lugar de ofrecer una función para hacer el cálculo, tendréis que implementar la spline desde cero. Esta función tomará al menos 4 puntos de control (de p_0 a p_3) y un valor entre 0 y 1 para determinar la longitud de la línea entre los puntos mencionados, y seguirá el esquema Catmull-Rom. Así se encadenarán segmentos de splines que nos puedan permitir crear una trayectoria más elaborada.

Es importante entender bien las ecuaciones de Catmull-Rom, podéis consultar el PDF adjunto en esta práctica y también a los profesores de la asignatura de Modelado Geométrico.



2.1 Código de base

También en esta parte se ofrece un código de base que no tiene porque utilizarse si los alumnos prefieren hacer la práctica de cero. En él se presentan algunos ejemplos de posibles trayectorias con formas simples (en formato .sp), con un poco de código para leerlo y convertirlo en secuencias de puntos de control que se puedan usar.

Es interesante, como primer paso, escribir el código para dibujar y poder visualizar estos ejemplos antes de continuar con el resto de esta parte.

2.2 Animación de la cámara

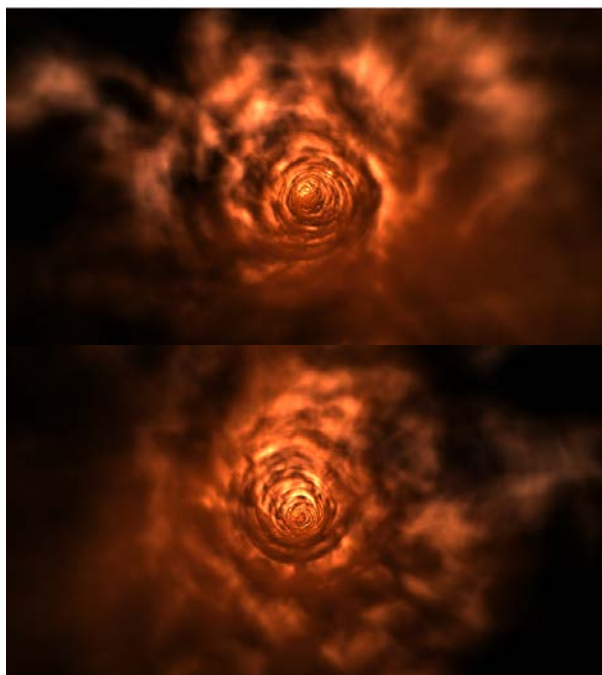
Una vez establecida la trayectoria parece fácil que la cámara la siga con una velocidad constante. Sin embargo, para imprimir algo de realismo, la cámara debería orientarse de tal manera que siempre mire hacia delante; para lo cual hay que cambiar su orientación y que siga la tangente de la spline a medida que se mueva. El documento de Christopher Twigg puede ser útil para hallar una solución al problema.

Un añadido opcional interesante sería variar la velocidad en función de las condiciones de la trayectoria para que resulte más realista. No es necesario hacer los cálculos físicos en función de la gravedad ó el flujo del torrente sanguíneo en cada latido, pero al menos se puede tener en cuenta la longitud del arco y su pendiente.

3. Parte 3: Mutación

En algún momento del recorrido, las células del torrente sanguíneo son atacadas por un agente externo a elección de los alumnos. Este agente debe provocar una transformación de los vasos y las células efectista.

Se deja a criterio de los alumnos qué



transformación realizar. Sois libres de hacer efectos de postproceso (siempre que encajen en la temática de esta práctica), idear nuevas técnicas y dar un toque artístico que diferencie notablemente vuestra práctica de la de vuestros compañeros.

4. Requerimientos de la práctica

4.1 Mínimo exigido (parte obligatoria)

- Manejar imágenes de al menos 256x256 para el campo de altura que se aplica al cilindro que forma el túnel a un ritmo interactivo (> 15 fps en una ventana de 640x480)
- Textura de Perlin *seamless* para las protuberancias
- La posibilidad de renderizar en forma de puntos, modo alámbrico, ó polígonos rellenos. Y el poder cambiar de uno a otro mediante teclas.
- El código debe renderizar una ventana en perspectiva y utilizar el buffer Z (profundidad) para eliminar las superficies ocultas
- Dar al usuario la posibilidad de moverse a través del escenario con determinadas teclas:
 - o Rotaciones
 - o Translaciones
 - o Escalados (Zooms)
 - o **Y, por supuesto, “deslizarse a través de los vasos sanguíneos”** (para ver la trayectoria en perspectiva de primera persona)
 - Lo que implica que cámara debe estar adecuadamente orientada en cada momento, esto es, **debe apuntar en la misma dirección que la tangente de la trayectoria**
- El recorrido debe ser circular en el sentido de que el principio y el final coincidan en el mismo punto (como en el caso de las atracciones de montañas rusas)
- Colorear los vértices de acuerdo a algún gradiente suave
- Irregularidades en la pared de los vasos mediante ruido de Perlin (sin reflejo)
- Incluir glóbulos rojos (con forma globular)
- A partir de cierto punto en la animación, los vasos sanguíneos y sus células deben sufrir una transformación efectista a discreción de los alumnos. La originalidad y el carácter artístico serán valorados positivamente.
- Comentar el código de forma razonable y entendible (lo leeremos)
- Documentación de las características y el control desde teclado/ratón (por ejemplo en un fichero `LEEME.txt`). Es importante que destaquéis a los usuarios (profesores) los extras que hayáis incluido

3.2 Parte “opcional” (sólo si se cursa la asignatura completa)

- Combinar el ruido de Perlin con un mapa celular para crear una pared más realista
- Incluir elementos decorativos en la arteria en forma de *impostors* con texturas: glóbulos blancos, plaquetas, virus...
- Glóbulos rojos más realistas
- Glóbulos rojos fluyendo a un ritmo distinto que el usuario (por ejemplo al ritmo de los latidos)
- 3 extras a elegir entre las dos siguientes listas

3.3 Posibles extras

- Experimentar con propiedades de los materiales y las luces
- Permitir al usuario deformar interactivamente el recorrido de los vasos

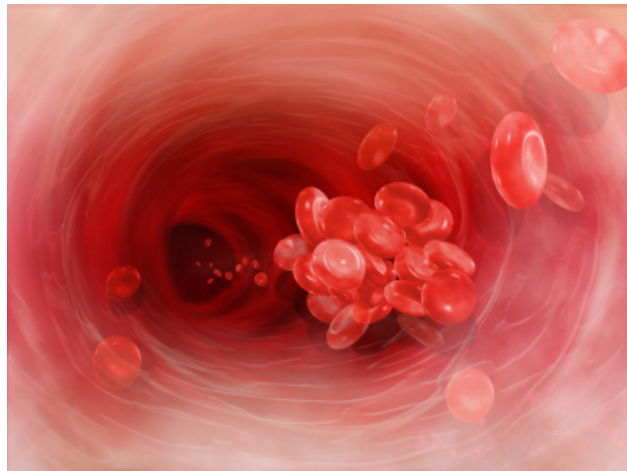
- Colorear los vértices a partir de los valores tomados a partir de otra imagen del mismo tamaño
- Zonas en las que la pared de la arteria sea tan fina que parezca translúcida
- Sombras proyectadas por los glóbulos rojos
- Iluminar la escena con luces y proyectar las sombras producidas por las irregularidades del túnel.
- Instanciación para los glóbulos
- Múltiples posibles caminos
- Bamboleo del usuario
- Todo lo que se os ocurra

3.3 Para los muy muy valientes

- Una parte del recorrido puede ser realizado en el exterior (no más del 30%)
 - o Del estilo de las prácticas del año pasado, hay que establecer una forma de entrar y salir del cuerpo humano (por ejemplo por el lacrimal como en la novela de Asimov).
 - o Crear una superficie en el terreno que tenga varios niveles de detalle, de manera que el campo de altura tenga una resolución significativamente superior
- Zona con neuronas como en la segunda parte de la novela de Asimov
- Pulmones y alveolos en los que tiene lugar el intercambio de O_2 y CO_2
- Geometrías no rectilíneas
- *Displacement mapping* con autooclusiones
- Shaders de geometría
- Shaders de teselación

Recomendaciones

Comienza con la práctica tan pronto como te sea posible. Es una práctica que requiere bastante dedicación y tiene muchos pasos intermedios. Si lo dejas para unos días antes de la fecha tope es poco probable que tengas margen para poder terminarla a tiempo.



Es aconsejable que estéis familiarizados con las transformaciones de vista en OpenGL antes de afrontar esta práctica. Si los alumnos consideran que se encuentran un poco oxidados desde la última vez que programaron con OpenGL, es conveniente que repasen las prácticas de G3D. También podéis probar con geometrías simples (como un cubo) antes de tratar de renderizar el campo de alturas sobre un cilindro.

Es preferible terminar la parte 1 antes de comenzar a programar la parte 2 de animación de la cámara. Aunque se ha desacoplado en la medida de lo posible para facilitar su programación en paralelo.

No lo dejéis para el último día. Esta práctica está pensada para que sea fácil y relativamente sencilla, pero la presión del tiempo puede fastidiar por completo esta sensación.

En principio, la práctica se puede hacer en parejas, si alguien no encuentra pareja o, por problemas de horario tiene complicado el poder compaginarlo con el trabajo, también es posible hacerla de forma individual, esta práctica es sencilla y no debería suponer una dificultad añadida el no tener compañero/a para realizarla.

¡Experimenta con tus propias ideas y diviértete con la práctica!

Esta práctica no requiere de hardware especializado para su realización. Los alumnos pueden realizarla cuando le resulte más cómodo en el aula de libre acceso S09 del edificio de Laboratorios II (Campus de Móstoles) que dispone de 15 ordenadores con tarjetas gráficas. Indícale al responsable del aula que necesitáis un ordenador os dirá el que podéis utilizar (procurad que sea uno de los ordenadores con las tarjetas para que vuestros trabajos vayan más rápido).

Forma y fecha de entrega

Se entregará la memoria en el formato de cualquier procesador de textos estándar (ó PDF) debidamente identificada junto con los ficheros con los programas en C en formato texto.

Puede enviarse directamente mediante la aplicación correspondiente en el Campus Virtual dentro del plazo, e indicando “[RA Practica01]”. No olvidéis indicar vuestros nombres y apellidos en el cuerpo del mensaje.

La fecha tope de entrega de esta práctica será el **6 de Marzo** de 2015.

Planificaros bien, pues se solapa con el periodo de realización del resto de las prácticas y los exámenes de mitad de cuatrimestre. Esta práctica puede llevar una cantidad considerable de tiempo, pero también es muy divertida.

Nota aclaratoria

Todas las marcas y productos mencionados en este enunciado de prácticas están registradas por sus respectivas compañías, y su uso es de carácter descriptivo con fines docentes.