

Ejercicios de Teoría

Sistemas Gráficos

Grado en Ingeniería Informática

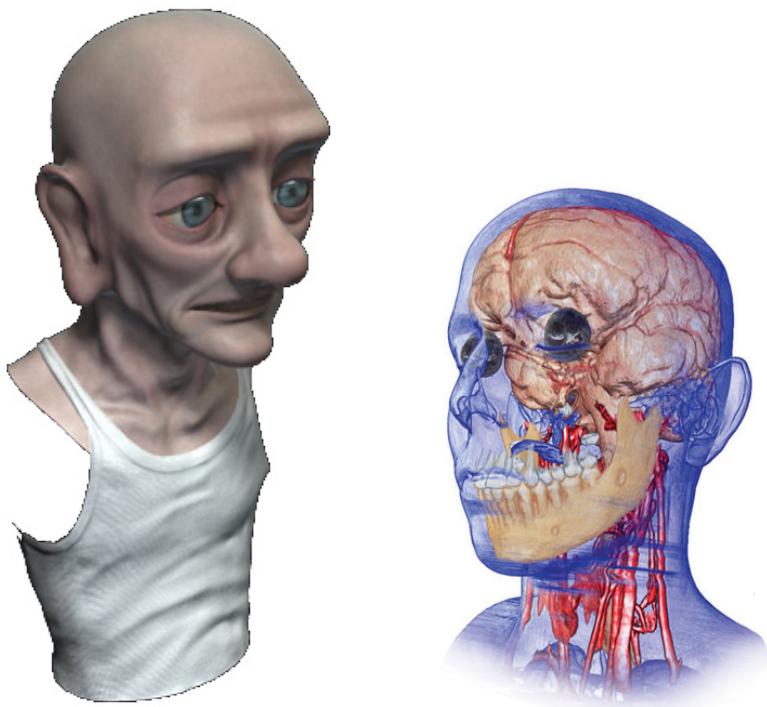
Curso 2019/2020

Introducción a los Sistemas Gráficos

1. En clase se han citado varios ámbitos en los que los sistemas gráficos forman parte fundamental, como el diseño industrial, el cine, la medicina o la arquitectura. Describir otras cuatro disciplinas profesionales o científicas que requieran de la participación inexcusable de los gráficos por ordenador para su correcto desempeño.
2. ¿Por qué no se han incluido disciplinas como “procesamiento de imágenes” y “fotografía computacional”?
3. Describir brevemente en lenguaje natural con el máximo detalle la figura mostrada.



4. Cuando ves una película de animación en el cine, ¿en qué te fijas, en el guión o en los efectos más o menos realistas? ¿Recuerda el movimiento del pelo de “Sulley” en “Monstruos SA”? ¿Y el resbalar de la capa de “Encantador” sobre su caballo al llegar al castillo en “Shrek”?
5. Compartir con los amigos una sesión de cine en la que se hayan usado elementos virtuales. Comentar dichos elementos.
6. Enumerar las diferencias que hay entre modelar un edificio y digitalizarlo en 3D.
7. Comentar las diferencias entre el modelo de una persona para un videojuego (imagen de la izquierda) y el modelo de una persona para una aplicación médica (imagen de la derecha).

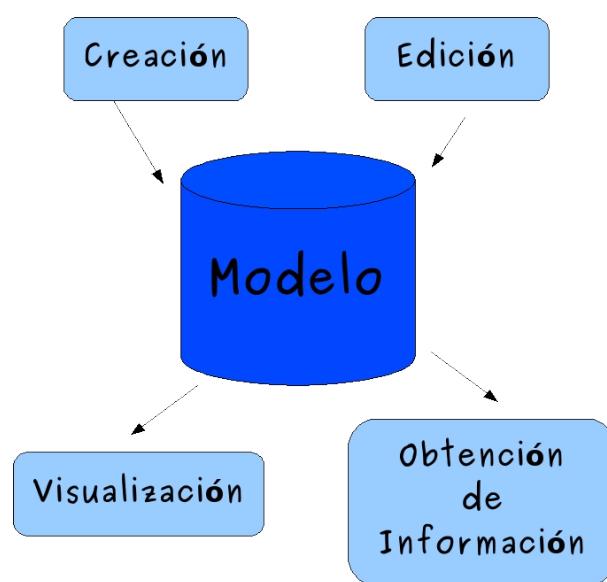


8. De las disciplinas vistas en el tema, ¿cuáles intervienen en un juego como Fortnite?. Comentarlas.

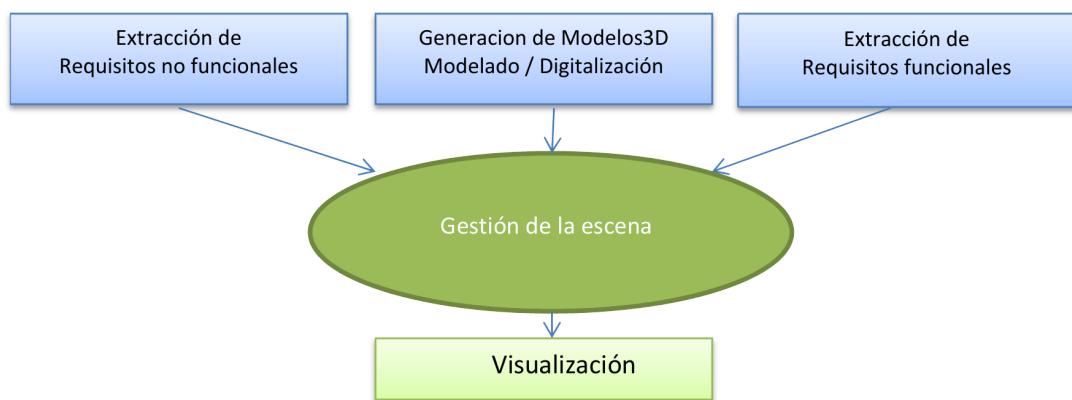


Desarrollo de un Sistema Gráfico

9. En clase se han mostrado modelos relacionados con procesos industriales, arte, arquitectura, ingeniería del software y otras disciplinas. Exponer y comentar modelos relacionados con otras 2 disciplinas.
10. Explicar el diagrama de la figura siguiente



11. Explicar qué es un grafo de escena y qué elementos intervienen.
12. Proponer 2 formas, distintas a las vistas en el tema, de representar una malla de polígonos.
13. Proponer una manera de representar el tablero de un Tetris 2D (solo el tablero, no las piezas).
14. Suponer un programa de decoración de habitaciones a partir de muebles y complementos de una biblioteca de elementos ya existente. Explicar cómo diseñarías el modelo principal y los subsistemas de creación, visualización y edición.
15. ¿A cuál de los bloques de la siguiente figura pertenecerían las siguientes situaciones? ¿Por qué?
 - a) El cliente nos explica que quiere un videojuego ambientado en la Alhambra
 - b) Modelamos con 3DStudio Max las partes arrasadas del Palacio de los Abencerrajes
 - c) Digitalizamos en 3D el Patio de los Leones para tener un modelo fiel de la realidad
 - d) El cliente nos explica que quiere el juego para PC (Windows) y Android.
 - e) Creamos un nivel ambientado en el patio de los leones, con 100 millones de polígonos y una serie de avatares humanos animados.
 - f) Dibujamos un frame para enseñárselo al cliente
 - g) Desarrollamos una versión específica para la tarjeta Quadro K5000
 - h) Cambiamos la velocidad de los avatares en función de la potencia del equipo
 - i) Controlamos que no se supere un tope de polígonos en la escena.



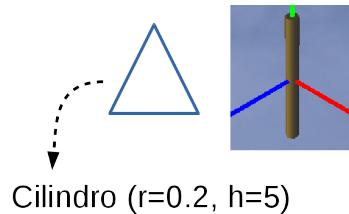
16. ¿La definición de las coordenadas de textura, pertenece a la generación del modelo 3D o la visualización?
17. ¿En qué fase del sistema gráfico interviene OpenGL?
18. Leer y comprender el código suministrado de la aplicación de la grapadora.
19. Comentar las diferencias más relevantes entre los ejemplos “hola mundo” subidos a Prado relativos a los lenguajes WebGL y Three.js.

Modelos jerárquicos

Requisitos en los ejercicios sobre modelos jerárquicos

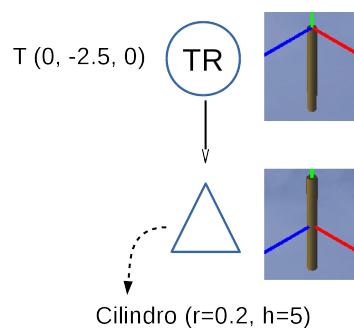
Estos requisitos **deben cumplirse** en todos los ejercicios de esta sección. Tener en cuenta que **en los exámenes**, en las preguntas sobre modelos jerárquicos y grafos de escena, se piden estos mismos requisitos.

- Ponerle dimensiones a todas las partes de la figuras.
- Simplificar toda la geometría usando solo cajas, conos, cilindros, esferas, etc.
- Cada geometría básica se considera que se crea centrada en el origen.
- Cada geometría debe crearse indicando sus dimensiones y dibujándola al lado del nodo junto con el sistema de referencia. Por ejemplo, para crear un cilindro de radio 0.2 y altura 5 se indicará según el siguiente gráfico.



Cilindro ($r=0.2$, $h=5$)

- Cada transformación debe estar etiquetada con la transformación concreta de que se trate, incluyendo sus medidas exactas.
- Las transformaciones a usar serán las siguientes.
 - *Traslación*, indicando sus valores en X, Y, Z. Por ejemplo, $T(2, 3, 4)$
 - *Rotación*, indicando el eje y el valor del ángulo en grados sexagesimales. Por ejemplo, para indicar una rotación por el eje X de 90° se indicaría mediante $RX(90)$
 - *Escalado*, indicando sus valores en X, Y, Z. Por ejemplo, $S(2, 2, 2)$
- Al lado de cada nodo de transformación debe ir un dibujo de lo que representa, incluyendo el sistema de referencia. Véase el ejemplo del siguiente gráfico.



Cilindro ($r=0.2$, $h=5$)

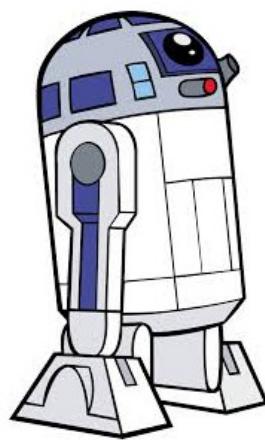
- Por tanto, **cada nodo del grafo debe estar convenientemente etiquetado**.

20. Dibujar el grafo de escena completo correspondiente al maniquí visto en clase. Sus articulaciones y movimientos serían los naturales en este tipo de figuras.
21. Dibujar el grafo de escena completo correspondiente a la imagen de abajo considerando las articulaciones y movimientos indicados.



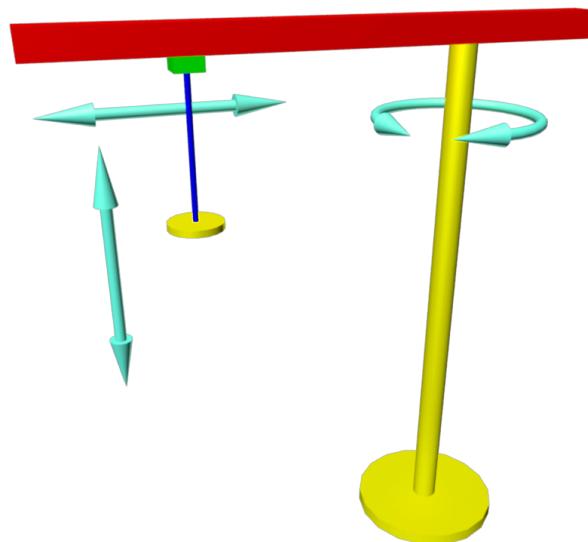
Articulaciones y movimientos

- El flexo completo se puede mover por el tablero de la mesa
 - Cada barra del flexo y la cabeza del mismo podrán girar por sus respectivas rótulas. En total son 3 rótulas de giro.
22. Dibujar el grafo de escena completo correspondiente a la imagen de abajo considerando las articulaciones y movimientos indicados.



Articulaciones y movimientos

- El robot completo puede desplazarse por el suelo.
 - Los brazos del robot pueden estirarse y encogerse, con lo que el conjunto cuerpo-cabeza del robot subirán y bajarán respectivamente.
 - El conjunto cuerpo-cabeza del robot se puede balancear (como si fuese un columpio) con respecto a los hombros.
 - La cabeza podrá girar de lado a lado, como si el robot dijera No.
23. Dibujar el grafo de escena completo correspondiente a la imagen de abajo considerando las articulaciones y movimientos indicados.



Articulaciones y movimientos

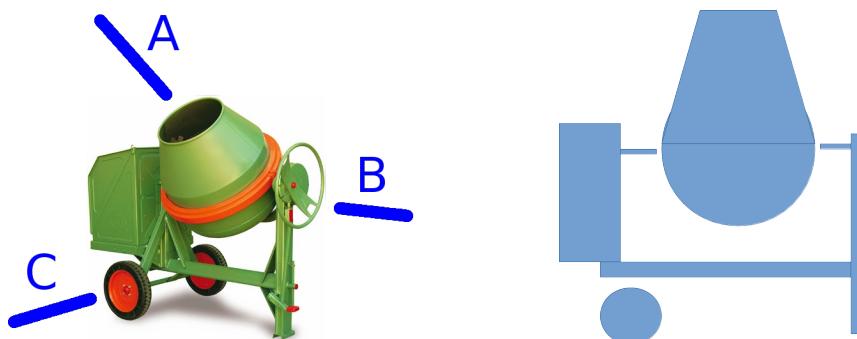
- El brazo (pieza roja) podrá girar según se indica.
- La pluma (pieza verde) podrá desplazarse por el brazo.
- La cuerda (pieza azul) podrá encogerse y alargarse. Provocando que el cilindro de la parte de abajo de la cuerda (el gancho) suba y baje respectivamente.

24. Dibujar el grafo de escena completo correspondiente a la imagen de abajo considerando las articulaciones y movimientos indicados.



Articulaciones y movimientos

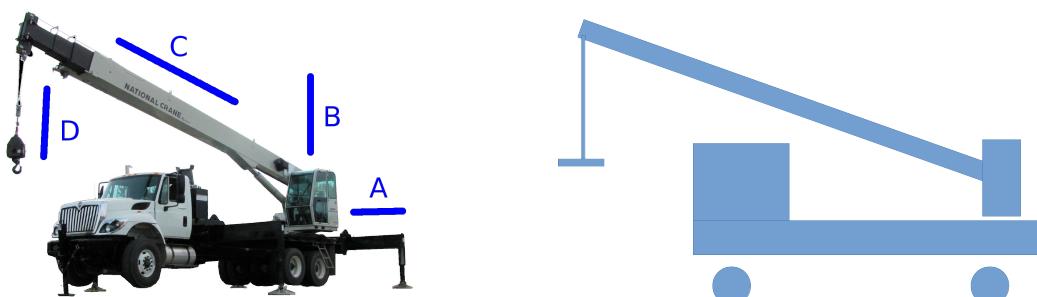
- La excavadora completa podrá desplazarse hacia adelante y hacia atrás.
 - El conjunto cabina-motor-brazo puede girar con respecto al eje vertical indicado.
 - Las distintas partes del brazo y la pala pueden girar con respecto a los ejes horizontales que se indican.
25. Dibujar el grafo de escena completo correspondiente a la imagen de abajo considerando las articulaciones y movimientos indicados. Realizar una versión simplificada en cuanto a la geometría, como se muestra en la figura de la derecha.



Articulaciones y movimientos

- El contenedor girará con respecto a su eje “A”, para realizar la mezcla.
- El contenedor también girará con respecto a su eje “B”, para vaciar su contenido.
- La hormigonera completa podrá girar con respecto al eje de las ruedas, eje C. Ele-vándose del suelo su pata de apoyo.
- La hormigonera completa podrá desplazarse por el suelo.

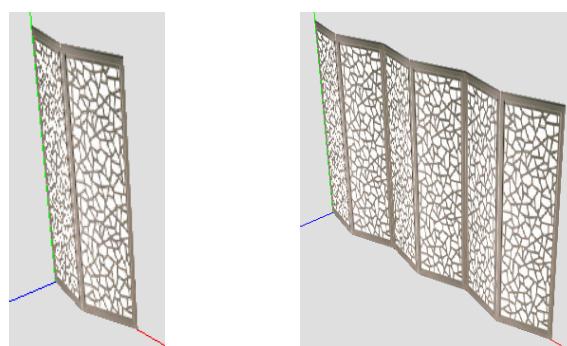
26. Dibujar el grafo de escena completo correspondiente a la imagen de abajo considerando las articulaciones y movimientos indicados. Realizar una versión simplificada en cuanto a la geometría, como se muestra en la figura de la derecha.



Articulaciones y movimientos

- El contenedor girará con respecto a su eje “A”, para realizar la mezcla.
- El brazo girará con respecto a su eje horizontal “A”
- El conjunto brazo-cabina de mando girará con respecto a su eje vertical “B”
- El brazo se alargará y acortará según se indica en la línea “C”
- El gancho subirá y bajará (línea “D”), encogiéndose y alargándose la cuerda, la cuerda debe mantenerse siempre vertical.
- El camión completo podrá trasladarse por el suelo.

27. Dibujar el grafo de escena completo correspondiente a un módulo de biombo (imagen de abajo a la izquierda). Las dimensiones de cada hoja del biombo serán fijas: 50 x 200 cm. La rama que represente a un módulo de biombo, debe tener un nodo, llamado ampliación, de modo que si se cuelga de ahí una rama de biombo exactamente igual, se puedan representar biombo de más hojas, tal como se muestra en la imagen de abajo a la derecha.



Articulaciones y movimientos

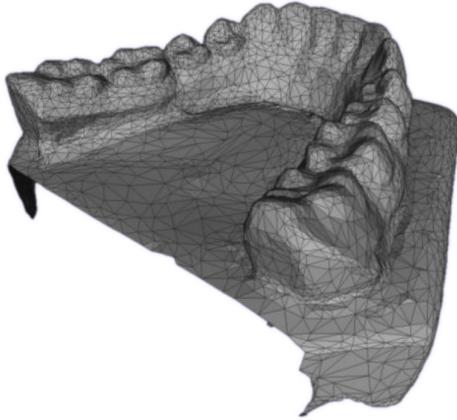
- El biombo se plegará y extenderá como un acordeón.

Creación de geometría

28. Comentar las diferencias entre un modelo geométrico y un modelo volumétrico. Poner un par de ejemplos de cada tipo.
29. Poner ejemplos de objetos del mundo real que modelarías mediante líneas, superficies y sólidos. Poner un par de ejemplos de cada tipo.

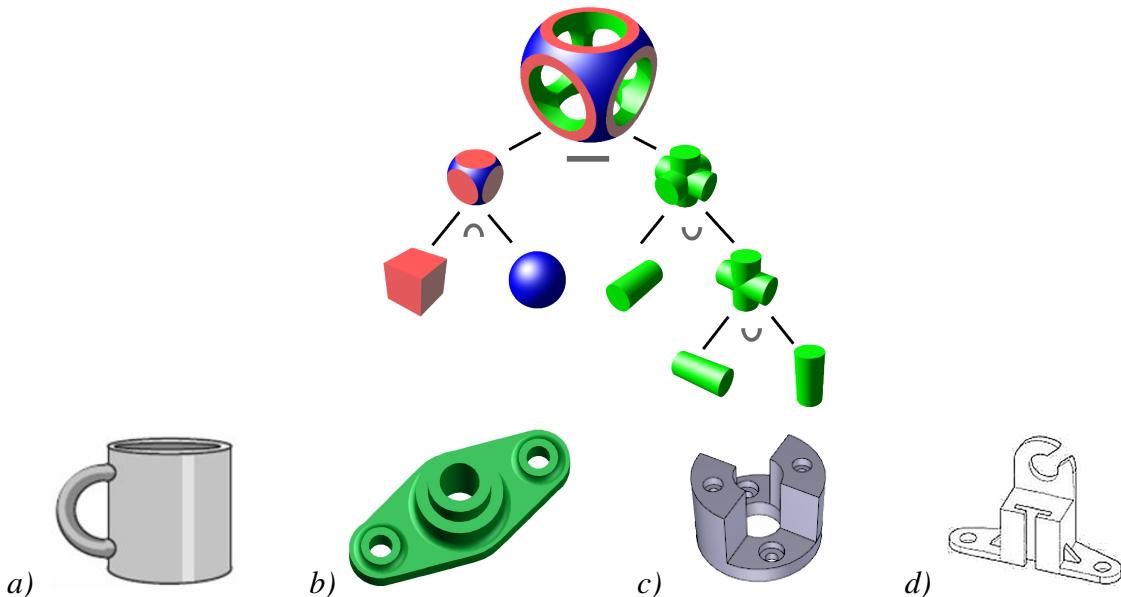
Sólidos

30. El modelo de la dentadura que se muestra en la imagen, ¿es una superficie o un sólido? Razonar la respuesta



31. En el caso de que el modelo de la dentadura del ejercicio anterior sea una superficie, ¿qué podría hacerse para representarla como un sólido?
32. Indicar ejemplos de sólidos del mundo real que modelarías como un sólido por barrido y como un sólido CSG.

33. Partiendo de primitivas geométricas básicas, dibujar el árbol CSG correspondiente a los sólidos de las siguientes imágenes: (la primera imagen es un ejemplo de cómo hay que realizar el ejercicio.



34. Explicar en qué circunstancias usarías un esquema de representación por descomposición espacial para representar un sólido.
35. Explicar en qué circunstancias usarías un esquema de representación B-rep para representar un sólido.
36. Escribir en pseudocódigo un método que reciba como entrada un float, el tamaño de un lado de un cubo, y devuelva una lista de floats que represente la geometría no indexada de un cubo centrado en el origen con el tamaño recibido como parámetro.
La lista de floats de salida se interpreta como sigue: Cada 3 floats, es un vértice 3D, y cada 3 vértices 3D es un triángulo. Por tanto, si un cubo tiene 6 caras, cada cara son 2 triángulos, cada triángulo son 3 vértices y cada vértice son 3 coordenadas (floats), debe salir una lista con 108 floats.
37. Realizar el ejercicio anterior pero considerando en esta ocasión que se representa el cubo mediante geometría indexada.
Deben rellenarse, por tanto, 2 listas:
- Una lista de vértices (8 vértices x 3 coordenadas/vértice), es decir, con 24 floats.
 - Y otra de índices (6 caras x 2 triángulos/cara x 3 vértices/triángulo), es decir, con 36 ints.
38. Implementar, en pseudocódigo, un método para crear la geometría indexada de una esfera centrada en el origen. Los parámetros de entrada son el radio de la esfera y la resolución.

39. Describir qué operaciones usarías, y en qué orden, para modelar el sólido de la imagen.



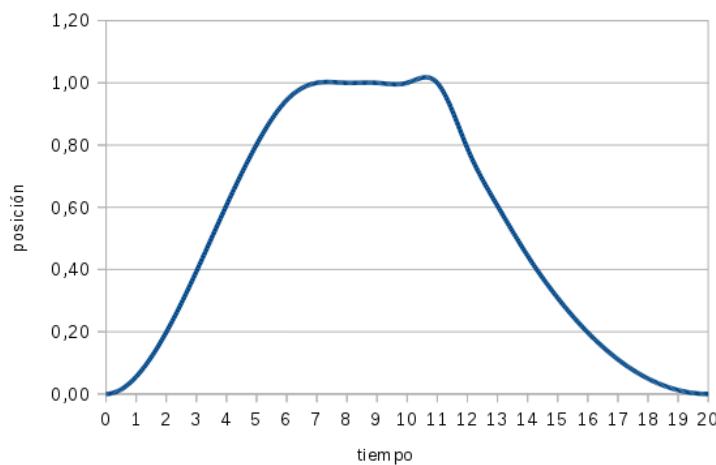
40. Describir qué operaciones usarías, y en qué orden, para modelar cada pieza del ajedrez mostrado.



41. Implementar en Three.js alguno de los modelos jerárquicos que se han planteado entre los ejercicios 20 y 27, ambos inclusive.

Animación

42. Explicar en qué consiste la técnica *Motion capture*.
43. Explicar cómo es el movimiento, con respecto a la velocidad, de un determinado objeto que se mueve según la gráfica siguiente.



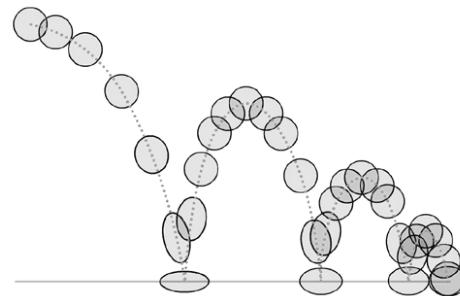
44. Definir las escenas claves que sean necesarias para realizar las siguientes animaciones.

Indicar:

- Qué parámetros intervienen.
- Qué valores tienen en cada escena clave.
- Qué función controla el valor de cada parámetro entre cada dos escenas clave.
- Qué tiempo transcurre entre cada dos escenas clave.

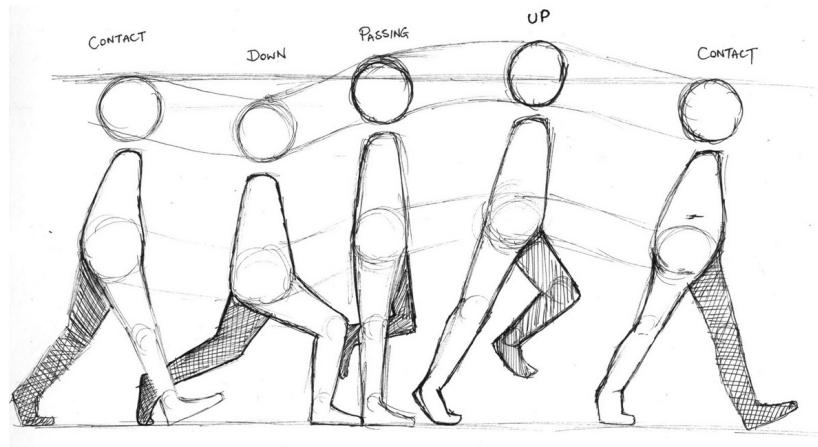
a) Una pelota botando contra el suelo, en vertical. La pelota debe deformarse (aplantarse un poco) el tiempo que esté en contacto con el suelo antes de volver a subir.

b) Una pelota botando contra el suelo mientras avanza.



c) Un péndulo oscilando de un lado a otro.

d) Un personaje caminando



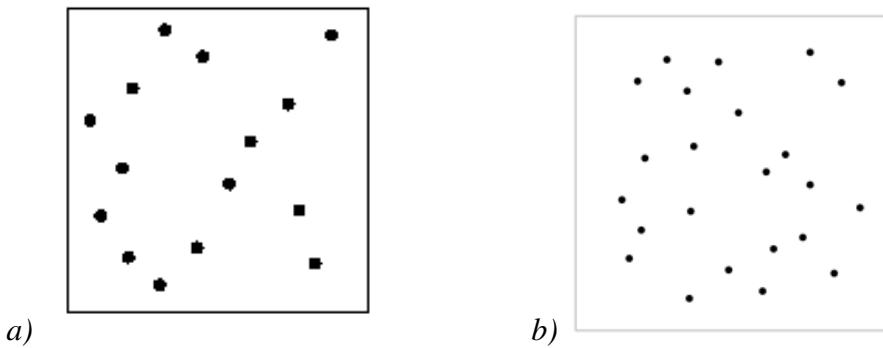
45. Suponer que se desea animar un cochecito de una montaña rusa. Explicar cómo realizarías la montaña rusa y cómo harías para realizar la animación del coche.

Interacción

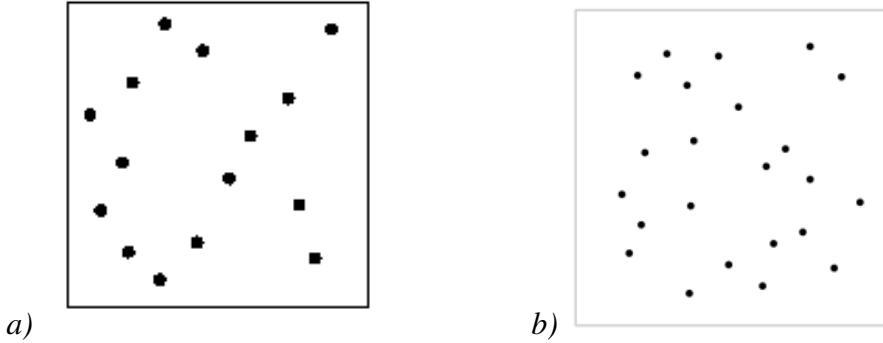
46. Explicar cómo diseñarías la parte de interacción de una aplicación que permita: añadir diferentes figuras geométricas en un escenario al hacer clic con el ratón en dicho escenario; también debe permitir mover las figuras por el escenario con el ratón y girarlas sobre sí mismas si se han seleccionado y se hace mover la rueda del ratón. Considerar que la interacción se realiza exclusivamente con el ratón.
47. Repetir el ejercicio anterior pero considerando que la interacción se hace exclusivamente con el teclado.
48. Repetir el ejercicio anterior considerando que la interacción se realiza combinando ratón y teclado.
49. Explicar el procesamiento a seguir para obtener la figura que ha seleccionado un usuario al hacer clic con el ratón en un determinado punto de la ventana de la aplicación (picking).
50. Imaginar una escena con varios personajes. Cada personaje está compuesto por varias figuras geométricas que representan la cabeza, las extremidades, etc. Explicar cómo se puede saber a qué personaje concreto pertenece la geometría concreta sobre la que un usuario ha hecho clic con el ratón.
51. Además de las técnicas vistas en clase para hacer un feedback ante un picking. Explicar otro modo de mostrar al usuario el hecho de que ha seleccionado un elemento de la escena de modo que dicho elemento siga viéndose con todas las características de los materiales que tiene asignados.

Detección de colisiones

52. Explicar cómo diseñarías un módulo para la detección de colisiones entre objetos de una escena.
53. Explicar cómo diseñarías un algoritmo para la detección exacta de colisiones entre dos sólidos representados mediante b-rep.
54. Diseñar un algoritmo para calcular la caja englobante de un objeto cualquiera a partir de su lista de vértices.
55. Diseñar un algoritmo para calcular la esfera englobante de un objeto cualquiera a partir de su lista de vértices.
56. Escribir en pseudocódigo un método para indexar elementos de una escena mediante un octree. El dato que se tiene de cada elemento para la indexación es un punto 3D. De la escena se conocen sus dimensiones totales. Hay que subdividir un voxel cada vez que tenga 2 objetos.
57. Escribir en pseudocódigo un método para, a partir del octree del ejercicio anterior, y a partir de un punto 3D como dato de entrada, nos devuelva el objeto que se encuentre en el voxel donde se sitúe el punto de entrada.
58. Sobre las escenas mostradas a continuación, realizar la partición que se corresponde con un quadtree:

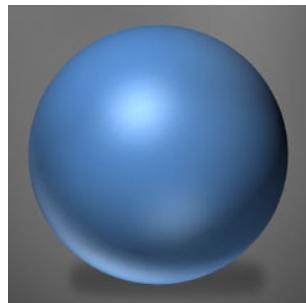


59. Sobre las escenas mostradas a continuación, realizar la partición que se corresponde con un k-d tree:



Materiales

60. Explicar qué son los rayos de reflexión glossy y para qué sirven.
61. Explicar qué son los rayos de refracción glossy y para qué sirven.
62. Explicar las diferencias entre la reflexión especular y la difusa.
63. Explicar la importancia del vector normal en el cálculo de iluminación.
64. En la siguiente imagen, indicar en qué zonas se aprecia más la intervención de la componente ambiental, en qué zona se aprecia más la componente difusa, y en qué zona se aprecia más la componente especular.



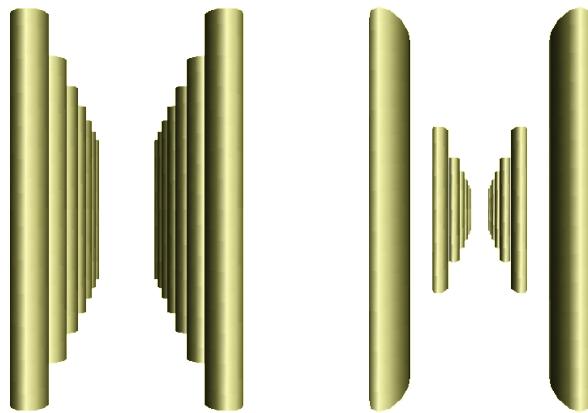
65. Explicar qué utilidad tiene la componente emisiva de un material.
66. Indicar ejemplos en los que sería más adecuado un material Lambert frente a un material Phong y viceversa.
67. Explicar qué colores y/o texturas usarías y en qué canales, para definir un material que se asemeje al de las siguientes imágenes. Puedes considerar los objetos completos o dividirlos en partes y definir materiales para cada componente.



(la alambrada)

Vistas

68. Indicar situaciones en la que emplearías estos tipos de vistas: primer plano, plano medio, plano general en perspectiva, plano general en planta, vista subjetiva.
69. En las imágenes mostradas. Explicar qué parámetros de la vista han cambiado y cómo, en una imagen con respecto a la otra.



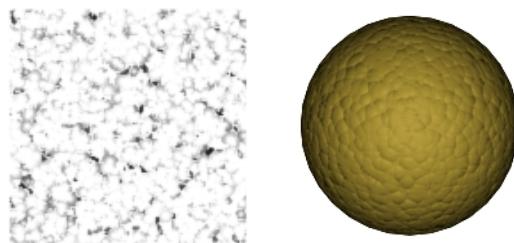
70. Explicar la diferencia que existe entre una vista y un viewport. Poner un ejemplo que ayude a explicarlo.

Luces

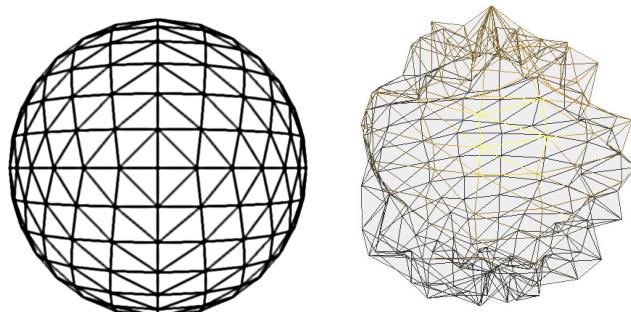
71. Indicar situaciones en las que emplearías estos tipos de luces: puntual, focal, direccional y subjetiva.
72. Explicar cómo afecta la luz ambiental a una escena. ¿Qué efecto tendría si no se pusiera la luz ambiental en una escena?
73. Suponer que estás modelando una escena consistente en un aula como la que usamos en clase de teoría, y necesitas iluminarla. Explicar qué tipo de luces usarías, cuántas y dónde. Razonar la respuesta.
74. Explicar cómo se calculan las sombras arrojadas mediante Ray-Tracing.
75. Explicar cómo se calculan las sombras arrojadas mediante Shadow Mapping.

Programación de shaders en GPU

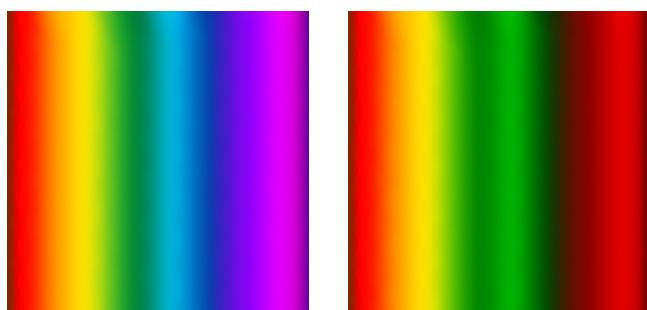
76. Explicar qué hay que hacer para crear y usar un programa shader.
77. Explicar qué realiza el rasterizador.
78. Si se quisiera programar un shader para implementar un efecto relieve alterando las normales según una textura en tonos de gris (ver imágenes de abajo). ¿Dónde se haría, en el vertex shader o en el fragment shader? Razonar la respuesta.



79. Si se quisiera programar un shader para implementar el efecto que se muestra en la imagen de la derecha. El shader recibe una esfera como la mostrada en la imagen de la izquierda y sin embargo el resultado es el de la imagen de la derecha. ¿Dónde se haría, en el vertex shader o en el fragment shader? Razonar la respuesta.



80. Si se quisiera programar un shader para eliminar la componente azul de cualquier color. (Ver imágenes abajo) ¿Dónde se haría, en el vertex shader o en el fragment shader? Razonar la respuesta.



colores originales

colores sin componente azul

81. Investigar tres formas distintas de calcular la distancia entre 2 puntos usando operadores GLSL.
82. Investigar la manera de convertir un color RGBA en un tono de gris conservando el mismo alfa y usando operadores GLSL.
83. Suponer que se desea realizar un shader para mostrar la temperatura de piezas mecánicas en una escala de color que va desde el azul (baja temperatura) al rojo (alta temperatura). Ver las imágenes abajo.

El shader se va a usar con un programa que va a asignar a cada vértice del b-rep que representa la pieza un float con la temperatura que el programa ha calculado. El shader debe visualizar la pieza mecánica *traduciendo* esas temperaturas en color.

Explicar cómo diseñarías ese shader, explicando qué harías en el vertex shader y qué harías en el fragment shader para realizar esa visualización.



Visualización con un material gris

Visualización con falso color según temperatura