

accelerating targets

La aceleración del ratón...

☐

Afecta al parámetro **b** de la ley de Fitts.

☐

Afecta al parámetro **a** de la ley de Fitts.

☐

Afecta a la variable **D** de la ley de Fitts.

☐

Afecta a la variable **W** de la ley de Fitts

☐

NO CONTESTO

Cámara 1a P

Tenemos una cámara perspectiva en primera persona montada en la cabeza de un Patricio, ajustada perfectamente para ver objetos cercanos dentro de una habitación.

Se quiere girar al Patricio 90 grados, para que quede mirando justo a través de una ventana, desde la que se ven objetos exteriores mucho más alejados.

Indica cuáles son los parámetros mínimos que habrá que ajustar, **como mínimo**, para hacer el giro de la cámara adecuadamente.

☐

Zfar y VRP

☐

VRP, FOV, zNear

☐

VRP, FOV, zFar

☐

VRP, FOV, ra, zFar

☐

NO CONTESTO

Camara1 (cast)

Tenemos una función `pinta_esfera()` que envía a visualizar el VAO que contiene el modelo de una esfera de radio 1 centrada en el punto (0, 0, 0). Tenemos una cámara con una óptica ortogonal con matriz `viewMatrix` identidad y con `window = (0, 10, 0, 10)`, `zn=0` y `zf=3`. Si antes de la llamada a `pinta_esfera()`, inicializamos una `TG=Translate(0,0,-2)`, ¿qué se vería?

Observación: VS i FS correctamente programados.

☐

La parte superior derecha de la esfera.

☐

La parte superior de la esfera.

☐

La parte inferior izquierda de la esfera

☐

La parte derecha de la esfera

☐

NO CONTESTO

EPA1 (cast)

En OpenGL 3.3. podemos activar dos técnicas de eliminación de caras ocultas al observador: el *back-face culling* y el *z-fuffer*. Tenemos una escena formada por un cubo y una pirámide. Indica la respuesta correcta:

☐

Hay que activar el *z-buffer* y podemos escoger si activar el *back-face culling* o no.

☐

Si no hacemos cálculos de iluminación podemos no activar ninguna técnica de eliminación de partes ocultas.

☐

Como los dos objetos son convexos con el *back-face culling* es suficiente para obtener una visualización correcta.

☐

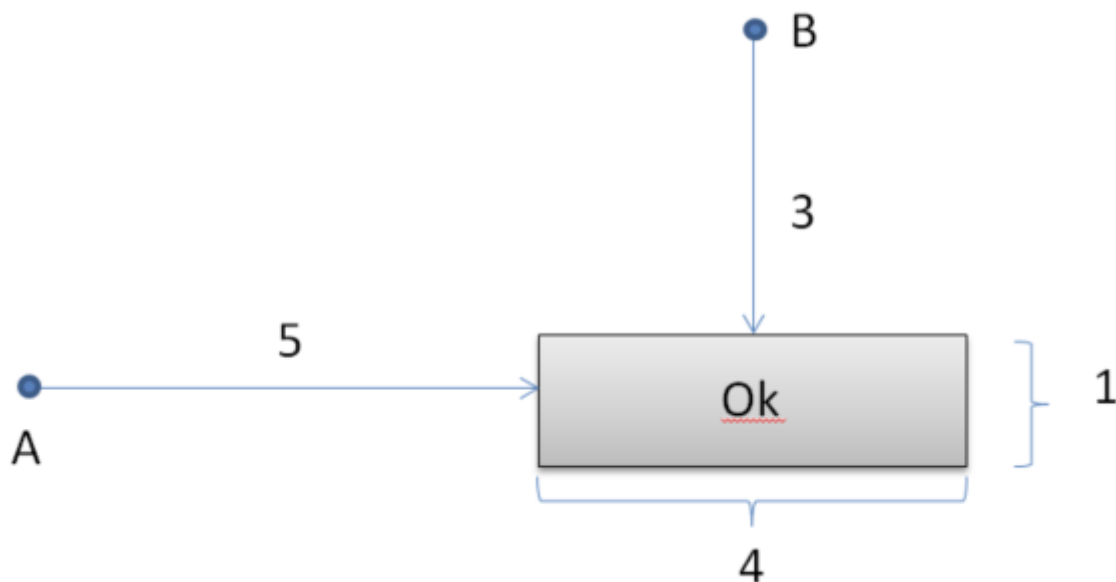
Si hacemos los cálculos de iluminación en el FS, necesariamente hay que tener activado el *z-buffer* de lo contrario con el *back-face culling* es suficiente.

☐

NO CONTESTO

Fitts - fórmula

Queremos apretar el botón Ok, situado en medio de la pantalla. Imaginemos dos posibles puntos de partida del cursor, **A** y **B**. Según la formulación de MacKenzie de la ley de Fitts podemos decir que:

☐

$MT_A < MT_B$

☐

$MT_A = MT_B$

☐

$MT_A > MT_B$

☐

No podemos saberlo sin conocer los parámetros **a** y **b** de la formula.

☐

NO CONTESTO

Gestalt / Menú swipe

Respecto a la interficie mostrada a continuación, el menú formado por bolitas rojas sobre las que se puede hacer un *swipe* para ver más opciones...



☐

No es aconsejable porque se muestra un elemento de la interficie parcialmente.

☐

Según la ley de Closure, nuestra percepción percibe la bolita incompleta sin ningún problema.

☐

No es aconsejable según el criterio de la ley de Similaridad, ya que la última opción (la bolita incompleta) se interpretaría como diferente.

☐

Por la ley de Simetría, la bolita incompleta rompe la sensación de grupo, porque no es simétrica con la primera.

☐

NO CONTESTO

gestos móvil

La interacción con gestos en los dispositivos móviles:

☐

Se puede utilizar para ahorrar espacio en pantalla, pero no está exenta de problemas de usabilidad.

☐

Es fácil de recordar y utilizar.

☐

Es más adecuada para dispositivos con pantallas grandes como los tablets que con los teléfonos, porque tenemos más espacio para mover el dedo.

☐

Ninguna de las otras respuestas es correcta.

☐

NO CONTESTO

Il·luminació 2 \ fragment vs vertex 2 (cast)

Si comparamos la iluminación calculada en el *fragment shader* respecto a la calculada en el *vertex shader* podemos decir que ...

☐

La iluminación en el *fragment shader* es más realista, mientras que en el *vertex shader* es más eficiente computacionalmente.

☐

La iluminación en el *fragment shader* es más realista y más eficiente computacionalmente que en el *vertex shader*.

☐

La iluminació en el *fragment shader* es visualmente equivalente al *vertex shader*, pero es más eficiente computacionalmente.

☐

La iluminación en el *fragment shader* es visualmente equivalente al *vertex shader*, pero es menos eficiente computacionalmente.

☐

NO CONTESTO

Il·luminació - constants materials i llums (cast)

Tenemos una esfera de radio 3 centrada en el origen de coordenadas y un foco de luz situado en la posición (0, 3, 5) de color (1, 1, 0). No hay luz ambiente. Un observador mira esta escena desde la posición (0, 0, 5) y mirando hacia el centro de la esfera, y lo que observa es una esfera que tiene una parte cercana a la silueta por la parte de bajo de la esfera de color negro, un degradado de colores verdes que son más claros por la parte de arriba de la esfera y más oscuros por la parte de debajo, y una mancha de color amarillo hacia la parte del medio de la semiesfera superior. ¿Qué constantes de material de la esfera permiten que se pueda ver esta escena de la forma descrita?

☐

$K_a = (0, 0.2, 0.2)$, $K_d = (0, 0.8, 0.8)$, $K_s = (1, 1, 1)$ y $N = 100$

☐

$K_a = (0, 0.2, 0.2)$, $K_d = (0, 0.8, 0.8)$, $K_s = (0, 1, 1)$ y $N = 100$

☐

$K_a = (0, 0.2, 0)$, $K_d = (0, 0.8, 0)$, $K_s = (0, 0, 0)$ y $N = 100$

☐

$K_a = (0.2, 0.2, 0.2)$, $K_d = (0.8, 0.8, 0.8)$, $K_s = (1, 1, 1)$ y $N = 100$

☐

NO CONTESTO

Il·luminació - coordenades pos llum (cast)

Tenemos una escena que nos pinta un suelo y un Patricio de altura 2 con el centro de la base de su caja contenedora en la posición (0,0,0) inicialmente. A este Patricio se le aplica la transformación geométrica TG y puede moverse con las teclas sobre el eje X, siendo en cada momento su posición final (posx,0,0). Supón que queremos poner un foco de luz a una altura de 2.5 encima de este Patricio y que se tiene que

mover con él. ¿Qué transformación se le tiene que aplicar a la posición del foco de luz (posx,2.5,0) para tener esta posición en coordenadas del observador (SCO)?

☐

posFocusSCO = view * vec4 (posx,2.5,0,1);

☐

posFocusSCO = TG * view * vec4(posx,2.5,0,1);

☐

posFocusSCO = view * TG * vec4(posx,2.5,0,1);

☐

No es necesario recalcular la posición del foco en ningún momento.

☐

NO CONTESTO

impresoras2 (cast)

Si queremos imprimir la bandera de Lituania que tiene colores amarillo, verde y rojo. ¿Cómo lo harías?

☐

En papel de color amarillo y con tintas CMY (0,0,0), (1,0,0) y (0,1,0) respectivamente.

☐

En papel de color verde y con tintas CMY (0,0,1), (1,0,1), y (0,1,1) respectivamente.

☐

En papel de color blanco y con tintas CMY (0,0,1), (1,0,1), y (1,1,1) respectivamente.

☐

En papel de color amarillo y con tintas CMY (1,1,0), (1,0,1) y (0,1,1) respectivamente.

☐

NO CONTESTO

Leyes Interacción \ steering

Para analizar como diseñar menús podemos:

☐

Utilizar la ley de steering para comparar cuál de los dos menús jerárquicos con la misma estructura y las mismas opciones pero anchuras y alturas diferentes se puede recorrer mejor.

☐

Utilizar la ley de crossing para determinar si el recorrido en un menú jerárquico necesita demasiado tiempo.

☐

Utilizar la ley de Fitts para saber si hemos puesto o no demasiadas opciones en el menú.

☐

Utilizar indistintamente la ley de Hick-Hyman o la de Fitts para evaluar el tamaño de los ítems del menú.

☐

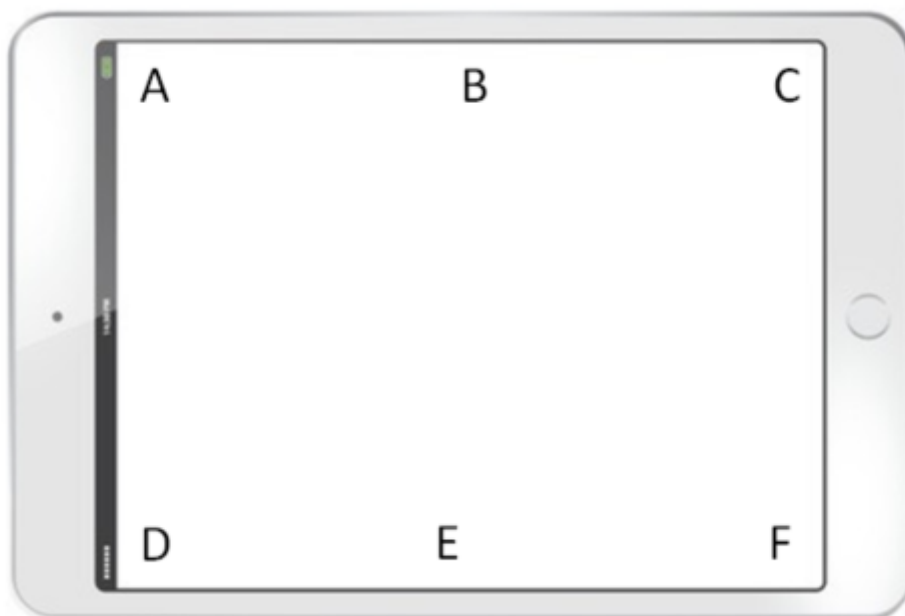
NO CONTESTO

Mobile Design / Àrees (cast)

En una aplicación de agenda queremos situar dos botones:

1. Botón para eliminar las tareas de toda la semana.
2. Botón para crear una tarea nueva.

Segúns la plantilla mostrada a continuación de una tablet en modo de uso horizontal, indica dónde pondrías cada botón.



☐

Pondremos: 1-B , 2-F

☐

Pondremos: 1-F , 2-C

☐

Pondremos: 1-D , 2-F

☐

Pondremos: 1-B , 2-E

☐

NO CONTESTO

RV

En Realidad Virtual:

☐

Si usamos un display fijo de realidad virtual semi-inmersiva (powerwall o CAVE), el volumen de visión para cada ojo se tiene que ir actualizando con los movimientos de la cabeza del usuario.

☐

Si usamos cascos de realidad virtual inmersiva, el volumen de visión para cada ojo se tiene que ir actualizando con los movimientos de la cabeza del usuario.

☐

Si es un sistema inmersivo (cascos) necesitamos actualizar la matriz de visión (viewmatrix) cada vez que el usuario mueve la cabeza, en cambio en los semi-inmersivos (display fijo) no es necesario.

☐

Ninguna de las otras es correcta.

☐

NO CONTESTO

Saturació/Luminància \ fatiga visual (cast)

En cuanto a la percepción del color, nos provoca fatiga visual...

☐

Un único color para el texto cuando el fondo tiene un color no saturado.

☐

Una alta saturación.

☐

Una gama pequeña de colores.

☐

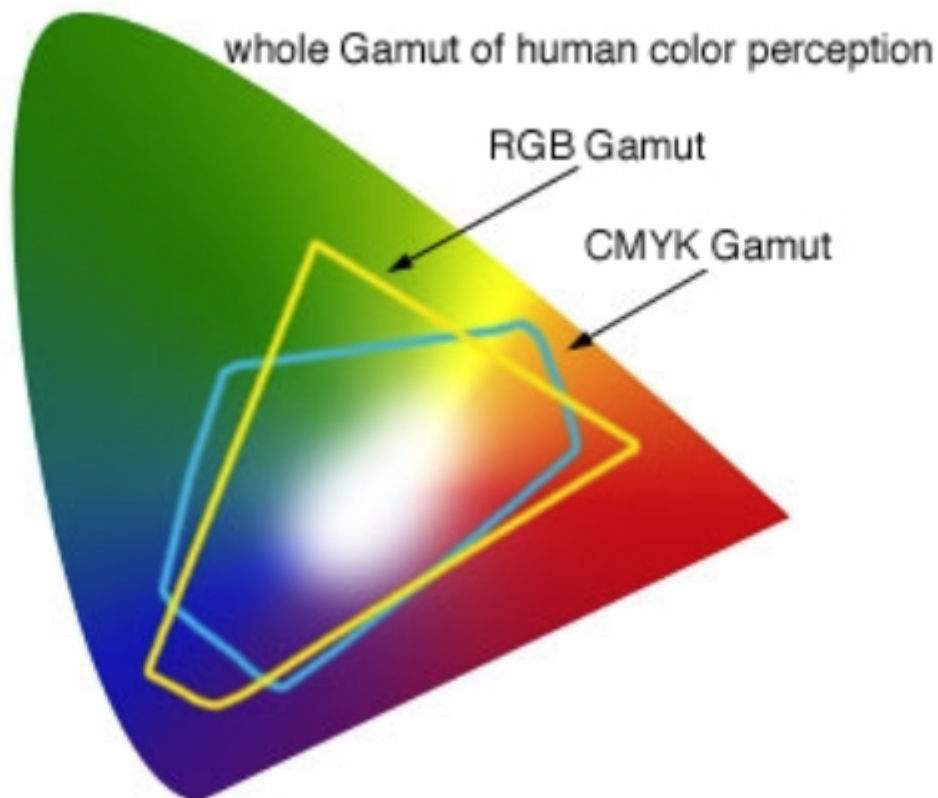
Una baja saturación.

☐

NO CONTESTO

Sistema visual humana - RGB (bis) (cast)

Respecto a la siguiente figura siguiente, selecciona la afirmación cierta.

☐

La figura explica por qué los diseñadores no pueden escoger cualquier color en pantalla si quieren imprimir su trabajo.

☐

La figura explica que los colores menos saturados no son reproducibles en un monitor RGB.

☐

La figura se ha manipulado y es incorrecta, con RGB se puede reproducir cualquier color visible.

☐

No hay ninguna respuesta correcta.

☐

NO CONTESTO

teclados

Teclear en un dispositivo móvil:

☐

Requiere más atención visual que un teclado mecánico

☐

Es fácil y eficiente si empleamos principalmente la técnica lift-off.

☐

No es problemático si el teclado es del tipo QWERTY.

☐

Escribir mediante gestos es siempre más rápido que haciendo click con el dedo en cada letra.

☐

NO CONTESTO

TGs2 (cast)

Tenemos un método `pinta_ninot()` que pinta un muñeco de nieve a partir de una esfera de radio 10 con el centro de la base de su caja contenedora en el origen, una segunda esfera de radio 5 centrada en el punto (0,15,0) y una tercera esfera de radio 1 centrada en el punto (5,15,0). Para construir una escena con un muñeco de nieve, un estudiante calcula la transformación a realizar sobre el muñeco que es:

$$TG = \text{Translate}(0,0,5) * \text{Rotate}(-90,1,0,0) * \text{Rotate}(-90,0,1,0) * \text{Scale}(0.5,0.5,0.5)$$

¿Cuál es el nuevo centro del muñeco una vez transformado y cuáles son sus

medidas en X, Y i Z respectivamente?

☐

Centro = (0,0,2.5), tamañoX = 10, tamañoY = 10 y tamañoZ = 15.

☐

Centro = (0,0,0), tamañoX = 0.5, tamañoY = 0.5 y tamañoZ = 0.5.

☐

Centro = (0,7.5,0), tamañoX = 10, tamañoY = 15 y tamañoZ = 10.

☐

Ninguna de las otras es correcta.

☐

NO CONTESTO

usabilidad

Según la definición de Usabilidad (ISO 9241) indica cual de las siguientes sentencias es cierta:

☐

Usabilidad siempre hace referencia a un grupo concreto de usuarios y a un entorno específico.

☐

Para que un programa sea usable, sólo hay que asegurarse de que sea eficiente desde el punto de vista de la implementación.

☐

La usabilidad tiene que ver únicamente con aspectos de informática, diseño de software, y artes gráficas.

☐

Usabilidad se define de manera general para cualquier usuario de un producto.

☐

NO CONTESTO

usabilidad

Cuando hacemos un estudio de usabilidad de un producto, cuál de las siguientes respuestas es correcta:

☐

Ninguna de las otras respuestas es correcta.

☐

No se puede hacer siguiendo la técnica de guerrilla testing.

☐

Se debe hacer al final del desarrollo del producto y con muchos usuarios.

☐

Se tiene que hacer siempre comparando con productos competidores.

☐

NO CONTESTO

Enviar