

Tria l'espai de coordenades en que ha d'estar P per tal que la transformació **projectionMatrixInverse*P** tingui sentit

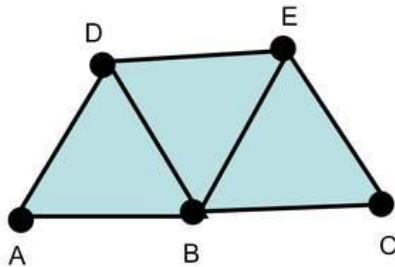
Select one:

- ☒ clip space
- ☐ eye space
- ☐ object space
- ☐ world space

Assigna a cada crida/tasca l'ordre relatiu (1,2,3,4) en que s'executa en un pipeline d'OpenGL sense GS:

glGenVertexArrays	1
S'escriu gl_Position	2
Backface culling	3
Rasterització	4

Indica un ordre adient per emetre els vèrtexs dels triangles de la figura, en un GS, usant una única primitiva:



Select one:

- ☒ ADBEC
- ☐ CEBAD
- ☐ DABEC
- ☐ ABDEC

Indica el punt que segur que serà FORA de la piràmide de visió d'una càmera perspectiva:

Select one:

- ☒ (1.00, 5.00, 6.00, 2.00) en clip space
- ☐ (3.00, 4.00, 1.00, 11.00) en clip space
- ☐ (2.00, 0.00, -5.00, 1.00) en eye space
- ☐ (-2.00, -4.00, -8.00, 1.00) en eye space

Indica el valor que retorna aquesta expressió GLSL:

`mix(9, 3, 0.7)`

Answer:

Correct answers: [4.8]

Indica el valor que retorna aquesta expressió GLSL:

`mod(7.9, 4)`

Answer:

Correct answers: [3.9000000000000004]

Tenim una primitiva que ocupa tot un viewport de 2048x2048 pixels. Indica quin codi ens dona un cercle blanc de radi 295 pixels centrat al viewport:

Select one:

- ☒ fragColor = vec4(1-step(295, distance(gl_FragCoord.xy, vec2(1024))))
- ☐ fragColor = vec4(step(1024, distance(gl_FragCoord.xy, vec2(295))))
- ☐ fragColor = vec4(step(295, distance(gl_FragCoord.xy, vec2(2048))))
- ☐ fragColor = vec4(step(295, distance(gl_FragCoord.xy, vec2(1024))))

Soposa que

P és un punt,

N és la normal unitària en el punt,

(a,b,c,d) és el pla perpendicular a N que conté P,

L és un vector unitari cap a la font de llum,

R és el vector reflectit de L,

V és un vector unitari en direcció cap a la càmera, i

Q és un punt arbitrari.

Quina interpretació té l'expressió **cross(dFdx(P), dFdy(P))**?

Select one:

- ☒ Vector normal
- ☐ Projecció del vector posició sobre una esfera unitària
- ☐ Vector de reflexió especular de la llum directa
- ☐ Vector tangent a la superfície en el punt

Indica, en un FS, quina transformació de la z en window space té un efecte equivalent a invertir el depth test amb `glDepthFunc(GL_GREATER)`:

Select one:

- ☒ `gl_FragDepth = 1 - gl_FragCoord.z;`
- ☐ `gl_FragDepth = -1 * gl_FragCoord.z;`
- ☐ `gl_FragDepth = 0.5 * gl_FragCoord.z;`
- ☐ `gl_FragCoord.z = 1 - gl_FragCoord.z;`

Diposem d'aquesta textura:



Indica amb quina opció el FS de sota obté aquest resultat amb l'objecte plane:



Recorda que plane.obj té coordenades de textura en [0,1].

```
fragColor = texture(colorMap, factor*vtexCoord + offset)
```

Select one:

- ☒ factor=vec2(0.1, 1.0); offset=vec2(0.1, 0.0);
- ☐ factor=vec2(1.0, 1.0); offset=vec2(0.1, 0.1);
- ☐ factor=vec2(0.1, 1.0); offset=vec2(0.0, 1.0);
- ☐ factor=vec2(0.1, 0.1); offset=vec2(0.1, 1.0);

La matriu que representa una reflexió respecte un mirall triangular definit pels vèrtexs (6.00, 0.00, 9.00), (9.00, 0.00, 1.00), (7.00, 0.00, 1.00) és...

Select one:

- ☒ $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- ☐ $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- ☐ $\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- ☐ $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Assigna a cada tasca l'ordre relatiu (1,2,3,4) en que s'executa, per simular reflexions especulars utilitzant la tècnica de sphere mapping en eye space:

El VS passa P, N a eye space

1

Es calcula el vector reflectit

2

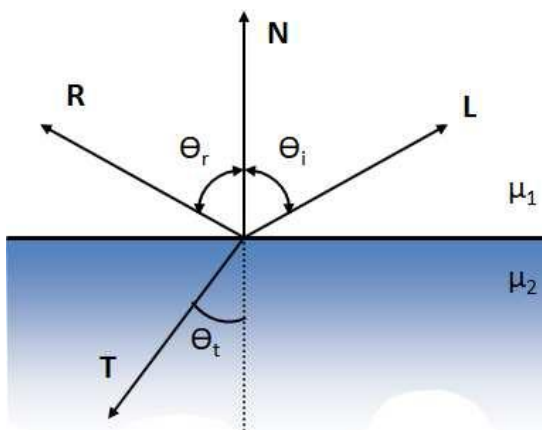
Es calculen les coordenades (s,t) del fragment

3

Es pren una mostra de la textura que conté el sphere map

4

Tenint en compte la llei de Snell, indica quina parella de valors (μ_1 , μ_2) explicarien de forma aproximada la direcció del raig trasmès que s'observa a la figura:



Select one:

- ☒ (1.09, 1.43)
- ☐ (1.43, 1.09)
- ☐ (0.09, 1.43)
- ☐ (1.09, 0.43)

Tenim activat alpha blending amb la funció

`glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);`

El color RGBA d'un fragment és (0.30, 0.20, 0.50, 0.10), i el color associat al frame buffer és (0.70, 0.50, 0.80, 0.60). Indica quin