

Examen Teoria (2^{on} parcial): Gràfics i Visualització de Dades
14 de juny de 2018

curs 2017-2018

Nom: _____

DNI: _____

Aula: _____ **Fila:** _____ **Columna:** _____

-
- Per marcar una resposta vàlida poseu
 - Per rectificar una resposta ja marcada poseu un sobre la i marqueu la correcta amb una
 - **Puntuació:** No contestada: 0 punts. Correcta: 1 punt. Incorrecta: -0.25 punts.
-

Test (40 punts): Temps 45 min.

1. Si es vol fer un ZBuffer amb Flat shading per visualitzar objectes representats amb malles triangulars i es té un mètode que calcula el test d'intersecció que entre un raig i un triangle, quina és la millor representació de la malla tenint en compte l'eficiència en memòria i en temps?
 - a. La representació explícita modificada per a guardar una llista de normals associada a cada vèrtex.
 - b. **La representació per vèrtexs indexats modificada per a guardar una llista de normals per triangle.**
 - c. La representació per adjacència de cares.
 - d. El model *winged edge*.
2. Quan s'utilitza RayTracing per visualitzar una escena, quina de les següents afirmacions és **falsa**?
 - a. Un objecte opac lambertià i sense reflexió espelular genera rajos secundaris.
 - b. **Un punt d'un objecte està a l'ombra si i només si la recta infinita que passa pel punt i la llum interseca amb algun objecte de l'escena.**
 - c. Si no es fa *anti-aliasing*, el nombre de rajos primaris ve determinat pel nombre de píxels del viewport.
 - d. Si només es té en compte el raig primari ($r = p_0 + \lambda * v$), el valor del color calculat depèn de les llums i de l'objecte més proper al punt p_0 amb el què interseca el raig r .
3. Quina diferència hi ha entre el Phong shading i el shading de Gouraud
 - a. **El Phong shading dóna resultats més suaus que en el shading de Gouraud, ja que a en el Phong shading es realitza interpolació de normals i no de colors.**
 - b. En el shading de Gouraud es calculen les normals en coordenades de món, mentre que en el Phong shading es calculen en coordenades de càmera, ja que es calcula en el fragment shader.
 - c. Cap, els resultats visuals d'ambdós shadings són iguals ja que el Phong shading és una optimització de Gouraud.
 - d. El shading de Gouraud es calculen els colors interpolats en el vertex shader mentre que a Phong shading es calcula la interpolació de les normals en el vertex shader.
4. Si es té una escena formada per un objecte format per 4 cares i té associat un material. Cada cara, però, té un color difús diferent. Quan es vol fer un ZBuffer de forma que es calculi la il·luminació per Phong shading....
 - a. Per fer eficient el *fragment shader*, quan es fa el draw de l'objecte, s'ha de passar per a cada vèrtex de l'objecte, tot el material a la GPU i rebre'l en el *fragment shader* com una variable IN.
 - b. Cal rebre tot el material en el *vertex shader* com UNIFORM i passar-lo al *fragment shader* en una variable OUT per tal que s'interpolin els seus valors a cada píxel.
 - c. Per fer eficient el *fragment shader*, cal passar els quatre materials a la GPU i decidir en la GPU quin material usar dependent de la cara que s'estigui dibuixant.
 - d. Per fer eficient el *fragment shader*, **cal enviar primer el material de l'objecte a la GPU com UNIFORM, després fer un draw de cada cara i a cada draw enviar a la GPU el color difús com a UNIFORM.**

Examen Teoria (2^{on} parcial): Gràfics i Visualització de Dades
14 de juny de 2018

curs 2017-2018

5. En la pel·lícula de Cars es va fer servir l'algorisme de RayTracing per a visualitzar escenes formades pels diferents escenaris, els edificis i els cotxes. Es feia servir un màxim de 1 rebot en cotxes que estaven prou lluny de l'observador. A part, per a les reflexions en les superfícies de tots els cotxes, només es calculaven les interseccions que distaven 12 metres de la superfície, sinó s'usava el color d'un *environmental map* per defecte i es parava la recursió. Quina de les següents afirmacions és **certa**?



- a. Si l'escena està representada en metres, cal utilitzar un t_{max} a 12.0, en el hit de tots els rajos reflectits
b. Es podria substituir el RayTracing per un Zbuffer i simular totes les reflexions en el *fragment shader* utilitzant només el *environmental map* basat en un CubeMap en el *fragment shader*
c. En el mètode ComputeColor del RayTracing cal calcular sempre el raig reflectit.
d. Cal accedir a l'*environmental map* directament amb les coordenades de textura corresponents als punts dels cotxes que intersequen amb el raig primari.
6. En el joc Portal 2, que és un joc en primera persona, el jugador disposa d'un aparell que li permet obrir portals en les parets d'un cert escenari. Un portal permet teletransportar-se d'un punt origen a un punt destí, punts de la mateixa habitació o d'habitacions diferents. Aquests punts es simbolitzen amb cercles de diferents colors. Quan el jugador obre un portal, veu en un cercle l'habitació on anirà, i quan el creua, veu l'habitació d'on ve. Per exemple, en la foto (1) de la figura de sota, el jugador està en una habitació i darrera seu té el que es veu a la foto. A la foto (2), el jugador ha obert un portal en aquesta habitació que el teletransporta a la mateixa habitació. Així, si entrés pel portal pel cercle de l'esquerra, estaria veient el fons de l'habitació on aniria a parar pel cercle de la dreta. En la foto (3) es veu l'efecte d'un jugador entrant pel portal pel cercle de sota i com es teletransporta al de dalt. Fixa't que en el cercle de dalt es veu el fons de l'escenari d'on ve. Quina de les següents afirmacions és **falsa**?

(1) Part de l'escena que està darrera del jugador en la figura (2)	(2) Com que el portal va a parar a la mateixa habitació, si el jugador passés pel portal per l'esquerra, veuria l'escena on aniria a parar.	(3) El robot, que simula el jugador, està passant pel portal des de la part d'abaix de l'escenari fins a la part de dalt.

- a. Per a generar la imatge (2) de la figura caldría tenir tres càmeres, una per a cada cercle del portal que només tindrien diferent la matriu *modelView* i una tercera càmera pel jugador que tindria les seves pròpies matrius *modelView* i *projection*, diferents a les dues càmeres anteriors.
b. Per a generar la imatge (2) de la figura es poden fer tres visualitzacions diferents usant el Zbuffer: Primer es calcularan les dues visualitzacions dels dos cercles del portal. En cada cercle l'observador es pot situar al mig de l'altre cercle. Després es calcularà una tercera visualització usant com a observador la posició del jugador i s'usaran com a textures dels dos cercles, les dues visualitzacions anteriors.
c. Per a generar la imatge (2) de la figura es poden fer tres visualitzacions diferents usant el Zbuffer: Primer es calcularan les dues visualitzacions dels dos cercles del portal. En cada cercle la direcció de visió, o *Look*, es el raig reflectit del raig que va del jugador al centre de l'altre cercle. Després es calcularà una tercera visualització usant com a observador la posició del jugador i s'usaran com a textures dels dos cercles, les dues visualitzacions anteriors.

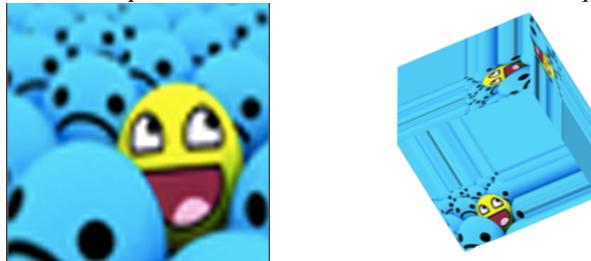
**Examen Teoria (2^{on} parcial): Gràfics i Visualització de Dades
14 de juny de 2018**

curs 2017-2018

- d. Si s'utilitzés RayTracing, es podria calcular la visualització d'una sola passada, calculant els rajos secundaris cadascun dels dependent de la posició del jugador i de la posició de l'altre cercle.
7. Sobre el zBuffer a GL, és **fals** que:
- Es fa la comprovació del ZBuffer abans del fragment shader per a descartar ja els triangles que no es veuran
 - La matriu projection conserva les z's a cada punt per a després poder fer la comprovació del z-Buffer
 - La distància entre el znear i el zfar determina la precisió que pot donar la comprovació de dues z's diferents de dos punts
 - Per a malles triangulars molt subdividides, el ZBuffer implementat en GPU permet generar visualitzacions més ràpides que el RayTracing
8. Es volen simular les visualitzacions que s'obtenen des de dues càmeres de videovigilància situades en una casa. Les dues càmeres estan col·locades a 2m de terra i enfoquen cap a 1.5 m des de la seva posició de forma fixa, amb un angle d'obertura de la càmera de 90° i un aspect ratio de 1.5. Quan detecten moviment enfoquen l'objecte que es mou, intenten visualitzar-lo sencer i el segueixen. Quina de les següents afirmacions és **certa**?



- La matriu *projection* ha de ser diferent a cada càmera ja que estan enfocant a llocs diferents.
 - La matriu *modelView* de les dues càmeres és la mateixa.
 - La *window* de les dues càmeres es defineix amb una alçada de 3m i una amplada de 4.5 metres però els seus orígens són diferents, ja que les càmeres estan situades en dos punts diferents.
 - Quan una càmera detecta moviment, les seves matrius *modelView* i *projection* canvien per què el VRP pot variar segons l'objecte que es mou i els seus plans de retallat laterals també.
9. Es vol mapejar la foto de l'esquerra en les cares d'un cub utilitzant tècniques de textures i ...



- La figura de la dreta s'obté per què algunes de les coordenades de textures no estan en l'interval (0, 1) i s'ha utilitzat la tècnica de wrapping `GL_CLAMP_TO_BORDER` en la funció de projecció amb una `smax, tmax = (2.0, 2.0)`.
- La figura de la dreta s'obté per què algunes de les coordenades de textures no estan en l'interval (0, 1) i s'ha utilitzat la tècnica de wrapping `GL_CLAMP_TO_EDGE` en la funció de projecció amb una `smax, tmax = (2.0, 2.0)`.
- La figura de la dreta s'obté per què algunes de les coordenades de textures no estan en l'interval (0, 1) i s'ha utilitzat la tècnica de wrapping `GL_CLAMP_TO_EDGE` en la funció de projecció amb una `smax, tmax = (0.2, 0.2)`.
- La figura de la dreta s'obté per què s'ha utilitzat una tècnica de mapeig indirecta de textures utilitzant com a figura intermitja un pla i per això es veuen les línies en lesarestes de la foto.

Examen Teoria (2^{on} parcial): Gràfics i Visualització de Dades
14 de juny de 2018

curs 2017-2018

10. Quan es calcula el raig en l'algorithm de RayTracing és **cert** que:

 - a. El punt inicial del raig es pot calcular a partir de les coordenades de càmera (xc , yc , $znear$), on xc , yc s'obtenen aplicant la transformació *viewport-window* al píxel (i, j) pel que passa el raig i la inversa de la matriu *projection*.
 - b. Si la càmera del RayTracing té definit com a pla de *clipping* posterior $zfar$, es pot posar en la crida al hit, $tmax = zfar$, tant per rajos primaris com per a rajos secundaris.
 - c. El pipeline de càlcul de les coordenades de món dels dos punts que definiran el raig és: Coordenades de dispositiu → Coordenades de món → Coordenades de càmera
 - d. Cap de les anteriors és certa.

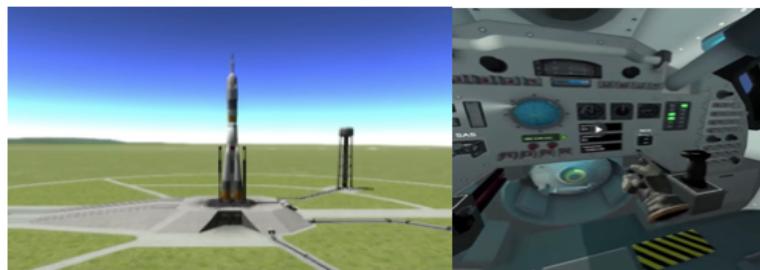
Respostes:

Examen Teoria (2^{on} parcial): Gràfics i Visualització de Dades 14 de juny de 2018

curs 2017-2018

Problema (60 punts): Temps 2:30h

El joc de "KERBAL Space" planteja el repte de construir coets que van a l'espai. El jugador construeix el coet i el llença a l'espai per a veure l'eficàcia del seu disseny. Fins ara, es visualitzava el llançament des de fora del coet. En aquest exercici es proposa una millora en la interfície per a donar-hi més vistositat. Es volen tenir dues visualitzacions simultànies on es vegi el llançament des de fora i des de la cabina de l'astronauta. Per això s'han construït dues escenes independents que es visualitzaran simultàniament en una pantalla de 1600 x 1200 píxels, de forma que en 3/4 parts de la pantalla es visualitzaran la nau exterior en el seu entorn durant l'enlairament i en 1/4 part restant es visualitzarà l'interior de la nau amb els seus tripulants, de forma semblant al que es veu en la figura següent:



A l'interior de la nau hi han dues finestres per on els tripulants veuen l'exterior. A la figura següent es veuen els dos costats de la cabina interior.



El dissenyador ha decidit modelar una escena amb els objectes exteriors, la terra com una esfera de radi de 63710 m i centrada en el 0,0,0. A més a més, utilitza un CubeMap com a textura de fons. Aquest CubeMap anirà canviant segons l'alçada de la nau en relació al mar que està en la superfície de l'esfera que modela la terra. Aquesta escena també conté els objectes que estan arran de terra, el coet i les textures que s'utilitzen per a visualitzar el terreny de la terra i les parets exteriors de la nau.

Es vol generar la visualització del joc tenint en compte que es tenen definits tots els objectes continguts en una escena **exterior** de mida 150000 x 150000 x 150000, centrada en el (0, 0, 0). El jugador només té visible a cada moment de l'animació un submón d'amplada 4500 metres, profunditat 8000 metres i que conservi l'aspect ratio del viewport on anirà la visualització de l'escena exterior en projecció paral·lela. Tots els objectes de l'escena exterior són malles triangulars i tenen els seus materials definits com lambertians, metà·lics o transparents. Només es considera una llum direccional corresponent al sol.

En relació a **l'interior** de la nau, el dissenyador ha considerat que tots els objectes estan continguts en una cabina cúbica de 5x5x5 metres, centrada en el 0,0,0. El seu interior es visualitza en perspectiva enfocant a 8 metres, amb un angle d'obertura vertical de 90° i mantenint el mateix aspect ratio del viewport on es visualitzarà l'interior de la cabina. La càmera està fixada en el centre del sostre de la cabina enfocant cap al mig de les parets de la cabina. Durant l'enlairament, el jugador podrà canviar la visualització per a veure les parts laterals o frontal de la cabina, però sense possibilitat de fer zoom. Suposa que la part frontal de la cabina està en les Z+ i les parts laterals a X- i X+. Tots els objectes i els personatges de la cabina també són malles triangulars i tenen els seus materials definits com lambertians i metà·lics, amb textures associades. Es considera que hi ha una llum puntual amb atenuació amb profunditat constant situada en el mateix punt que la càmera.

Des de l'interior de la nau es pot veure l'escena exterior per dues finestres centrades en les X- i a les X+ de la nau respectivament. Les finestres fan una mida de 0.5 m x 0.5 m. Per a simular el que es veu per la finestra es considera que es el que es veu des de la càmera interior enfocant cap al mig de la finestra. El terra de la cabina està situat a 75 metres del terra de la nau.

Examen Teoria (2^{on} parcial): Gràfics i Visualització de Dades 14 de juny de 2018

curs 2017-2018

Tot suposant que tens els dos codis realitzats a les pràctiques amb les dues aproximacions, RayTracing i ZBuffer, amb el càlcul de la intersecció raig-triangle implementat, respon les següents preguntes:

- (30 punts). Suposa que has de generar les visualitzacions de l'escena exterior i la interior. Inicialment la base del coet està situat al punt 63710, 0, 0, suposant que l'eix de la terra és l'eix Y. L'alçada del coet és de 80 metres. Per a visualitzar l'enlairament de la nau des de fora, inicialment es situa una càmera a la mateixa alçada del coet i allunyada 100 metres del coet en positiu en la resta de dimensions. A més, està enfocant distància 100 metres cap al centre del coet. Quantes càmeres són necessàries per a generar les visualitzacions del joc? Detalla els valors inicials de TOTS els atributs de la/les càmera/es, definint especialment en quines coordenades els estàs detallant. Si hi han càmeres diferents, indica per què són diferents i si tenen valors en comú, i per què.
- (40 punts) Utilitzant la visualització en ZBuffer com el del codi de la pràctica i activant un *Timer* per a simular els diferents frames d'enlairament, es desitja utilitzar diferents textures de *CubeMap* per a anar-les canviant quan el coet està a diferents alçades sobre el nivell del mar. Quan estigui entre 0 i 6000 m s'utilitzarà el CubeMap0, quan estigui a més de 6000 m però a menys de 20 m s'usarà el CubeMap1, quan estigui entre 20 m i 5000 m, el CubeMap2, entre 5000 m i 8500 m el CubeMap4 i a més de 8500 m, el CubeMap5. En la Figura es veuen diferents visualitzacions de com quedaria la visualització final a diferents *frames* de l'escena exterior usant els diferents CubeMaps.



Es desitja tenir una visualització de *Phong Shading* de l'escena exterior de forma que s'usi el CubeMap corresponen de fons i que el coet reflecteixi els colors del CubeMap en el seu fuselatge. Disposes d'una funció ja implementada (i que no has de definir ni detallar en l'examen) que calcula la matriu de la transformació geomètrica que pertoca a cada frame segons la trajectòria del coet, definida com:

```
mat4 calculaTG(iframe)
```

on *iframe* és el número de frame i *mat4* es el valor de retorn de la matriu. Aquest matriu defineix la Transformació Geomètrica absoluta a cada frame que cal aplicar al coet considerant la seva posició inicial.

També disposes d'una funció que donada l'alçada especificada en metres del punt mig del coet et retorna quin CubeMap has d'utilitzar (suposant que els CubeMaps estan guardat en un array de 5 QOpenGLTexture, on estan carregades de fitxer les textures de tipus cubeMap)

```
int getCubeMap (float alcada)
```

A cada frame, com canviaries el codi per a que fos el més eficient possible? Indica en pseudo-codi els passos que has de realitzar a cada pas del *timer*. Detalla el que es calcularia en la CPU i a cadascun dels *shaders*.

- (30 punts). Suposa que es vol fer la visualització de l'interior de la cabina l'algorisme de RayTracing, considerant que es veu una de les parts laterals on hi ha finestra. Es vol fer una visualització de *Toon Shading* pels objectes de l'interior de la cabina, de forma que es vegin ben marcades les diferències entre els colors que es calculen els colors en el *Toon Shading*. A les finestres es visualitzarà tot l'exterior, dependent de l'alçada on estigui el coet i sense tenir en compte la refracció del raig, visualitzant l'exterior amb *Phong Shading*. Descriu la teva aproximació en un dibuix primer i després detalla el pseudo-codi del Raytracing que cal canviar. Justifica quins mètodes canviaries (*Render*, *Hit*, *ComputeColor*, ...). Raona i justifica el pseudo-codi que has proposat.