Examen Teoria (2on parcial): Gràfics i Visualització de Dades
4 de juny de 2014

curs 2013-2014

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: -0.1 punts.

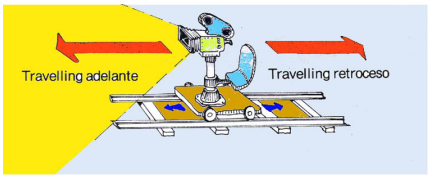
Test (40 punts): Temps 40 min.

1. Quina de les següents afirmacions és certa?
   a) En projeccions perspectives, en aplicar la matriu *projection* als punts de l'escena, es pot alterar la distància en x o en y entre les seves posicions relatives.
   b) Donada una escena formada per diferents objectes opacs, si hi ha un objecte que tingui coordenades normalitzades al punt (-0.5, -0.5, -0.5), sempre és visible.
   c) Els punts (10, 25, 5, 10) i (8, 20, 14, 8), després d'homogenitzar-los, es corresponen al mateix punt.
   d) En aplicar la matriu *projection* sobre els punts de l'escena i passar-los després a coordenades de dispositiu, es pot canviar l'ordre relatiu de les z's dels objectes.

2. Donada una *window* de coordenades x_{min} = 0.0, x_{max} = 2.0, y_{min} = 0.0 i y_{max} = 3.0 i un *viewport* de 400 píxels x 600 píxels amb el seu origen a (100, 300). Quina és la matriu de transformació window-viewport?
   a) $M = \begin{bmatrix} 200 & 0 & 100 \\ 0 & -200 & -300 \\ 0 & 0 & 1 \end{bmatrix}$
   b) $M = \begin{bmatrix} 200 & 0 & 100 \\ 0 & -200 & 300 \\ 0 & 0 & 1 \end{bmatrix}$
   c) $M = \begin{bmatrix} 200 & 0 & 100 \\ 0 & 200 & -300 \\ 0 & 0 & 1 \end{bmatrix}$
   d) Cap de les anteriors.

3. Si es desitja canviar la window que defineix els plans laterals del *frustum*,
   a) Cal tornar a definir només la matriu *projection*, tant en projeccions paral·leles com en projeccions perspectives.
   b) Cal tornar a definir la matriu *model-view* i la matriu *projection*
   c) Cal tornar a definir només la matriu *model-view*.
   d) Cal tornar a calcular la matriu de projecció perspectiva, però no la matriu de normalització quan es calcula la matriu *projection*.

4. En alguns jocs, es vol fer un moviment de tràveling de la càmera automàtic, de forma que s'acosta al personatge progressivament en una projecció perspectiva. Suposa que inicialment el personatge està al punt (5, 0, 0) i la càmera al punt (20, 0, 0). Es vol simular el moviment de la càmera traslladant-se al llarg de l'eix X, segons el vector (-1,0,0). Quina de les següents afirmacions és certa?

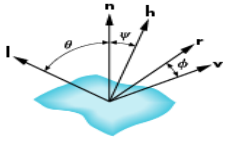


   a) A cada *frame,* cal canviar la posició de l'observador i per tant, cal recalcular tantes matrius *projection* com *frames* tingui l'animació.
   b) Els plans de *clipping* antero-posterior poden ser els mateixos a tots el *frames*, ja que es defineixen en coordenades de càmera.
   c) Cal recalcular tantes matrius *projection* com a *frames* es vulguin gravar en l'animació, però la matriu *modelView* és comuna a tots els *frames,* ja que el VRP es manté sempre al punt (5, 0, 0) i els angles de visió són el mateixos.
   d) Cap de les anteriors en certa.

5. En relació al model de Blinn-Phong, quina de les següents afirmacions és certa?
   a) La intensitat de reflexió difusa està influïda pel vector reflectit i l'angle entre la direcció de la llum i la direcció de visió.
   b) La intensitat de reflexió difusa depèn del coeficient especular del material.
   c) La intensitat de reflexió difusa es independent del vector de visió.
   d) La intensitat de reflexió ambient depèn de l'angle entre el vector de llum i la normal de la superfície.

6. Segons el model de Phong shading, quina de les següents afirmacions és falsa?
   a) En el Phong shading s'interpolen les normals entre els vèrtexs d'una cara.
   b) Es pot realitzar el càlcul del model de Phong tant en coordenades de món, com en coordenades de càmera.
   c) El Phong shading és significativament més lent que el Gouraud shading.
   d) El Phong shading requereix més cares en l'objecte per a tenir un aspecte més suau que el Gouraud shading.

7. Suposa que es vol il·luminar una esfera de centre (0,0,0) i radi 1 amb una llum direccional amb direcció L = (1, 1, 1) i l'observador està posicionat al punt (5, 5, 0). La intensitat de la llum és I_d = (0.8, 0.8, 0.8), I_r = (1.0, 1.0, 1.0) i I_a = (0.2, 0.2, 0.2). El material de l'esfera és k_d=(0.4, 0.8, 0.4) i k_s = (1.0, 1.0, 1.0) amb un coeficient de *shineness* de 500. Com es veurà l'esfera utilitzant Blinn-Phong? Suposa que no hi ha intensitat ambient global, ni atenuació en profunditat.



   a) L'esfera es veurà de color verd, amb una taca petita de color blanc en el reflex especular.
   b) L'esfera es veurà de color vermell amb una taca petita de color blanc que correspon al reflex especular.
   c) Phong-Blinn no es pot aplicar a llums direccionals.
   d) Els valors del material estan mal definits ja que la suma de k_d i k_s no pot superar al color (1.0, 1.0, 1.0)

8. En relació al procés de “culling”...
   a) Es realitza com a part del zbuffer i no té sentit aplicar-lo en el raytracing, ja que el mateix raig ja descarta les cares no visibles.
   b) Serveix per a calcular la visibilitat de les cares d'un objecte en relació a la seva posició amb altres objectes.
   c) Aplicar-lo als objectes en coordenades de món fa que el *pipeline* gràfic sigui més eficient que si s’aplica als objectes en coordenades de càmera.
   d) No té sentit realitzar-lo si ja s'aplica algun mètode d’eliminació de parts amagades.

9. En relació als algorismes d’eliminació de parts amagades es pot afirmar que:
   a) els mètodes del Pintor i de depth-sorter s'apliquen en coordenades de món i els mètodes de z-buffer, scan-line i ray-tracing s'apliquen en coordenades de càmera.
   b) El mètode de z-buffer es pot integrar en el mètode de rasterització de scan-line per a optimitzar càlculs.
   c) No garanteixen la resolució de casos complicats com el d’ensolapament cíclic de dos objectes.
   d) El mètode de z-buffer, dins del *pipeline* de GL, s'aplica abans de la rasterització.

10. El mètode de Cohen-Sutherland .....
   a) ... només es pot aplicar amb coordenades normalizades o en projeccions paral·leles.
   b) ... s’aplica a coordenades de càmera no normalitzades.
   c) ... es pot aplicar tant a coordenades de càmera com en coordenades de món.
   d) ... resolt els casos trivials de codificació de segments totalment interiors o totalment exteriors, però pels casos d’intersecció ha d’utilitzar altres mètodes.

Respostes:

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
a	X		X				X			X
b				X					X	
c					X			X		
d		X				X				

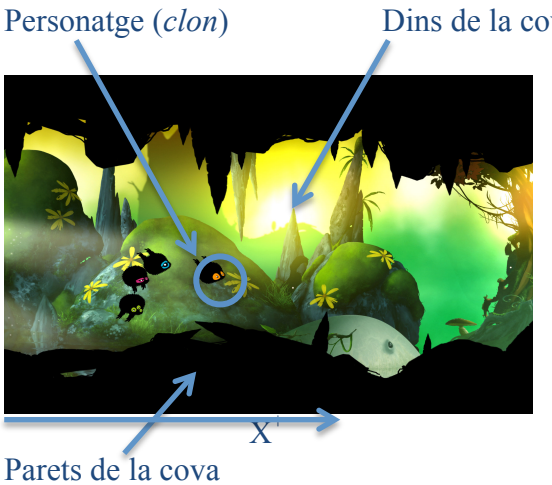
Examen Teoria (2on parcial): Gràfics i Visualització de Dades
4 de juny de 2013

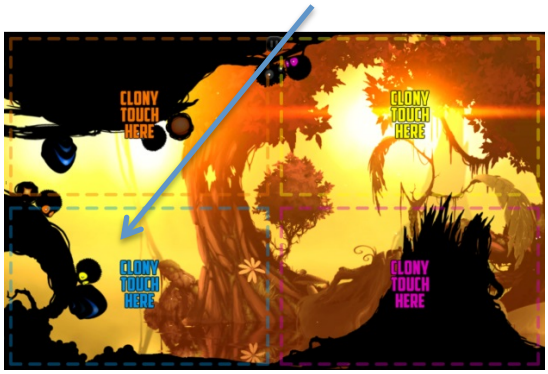
curs 2013-2014

Problema (60 punts): Temps 2h

Es vol simular una variació del joc per tabletes de Badland on hi han 4 jugadors portant els seus respectius personatges esfèrics (*clons*) per una cova de colors limitada per parets fosques i esquerpes. Els personatges només poden moure's la part dels colors (la cova). Pots veure les parts del joc en el dibuix de l'esquerra.

El moviment de cada *clon* és de translació. Quan els *clons* col·lisionen amb alguna de les parets de la cova, reboten cap a una altra direcció. Quan el joc s'inicia, l'efecte visual és que el clon està parat i tot l'escenari es desplaça cap a les X's negatives de pantalla. El jugador fa un click a la pantalla i desplaça el dit fins a aixecar-lo per donar moviment al seu *clon.* El clon es trasllada proporcionalment al desplaçament fet amb el dit per la pantalla. Si es deixa de picar la pantalla, el *clon* no es mou. L'usuari fa moure el personatge tocant la part de la pantalla que és del color del seu *clon,* tal i com es veu al dibuix de la dreta.
Quan un personatge ja no es veu en la pantalla, queda eliminat del joc. Quan un *clon* queda com a únic *clon* visible a la pantalla o arribar el primer al final de la cova, guanya el joc.





Per a modelar tot l'escenari es tenen dues escenes: una per modelar les parets de la cova i una per a modelar els objectes de dins de la cova i els *clons*. Suposa que les dues escenes estan definides i alineades en el mateix sistema de coordenades, amb origen al punt (0,0,0) i de dimensió de 100000x100x1000. Durant el joc, els jugadors només veuen un part parcial les escenes de mida 100x100x1000 en un únic *GLWidget* de 1400x800 píxels. L'àrea de joc de cada jugador és una quarta part de la mida del *widget*, tal i com es veu a la figura de la dreta. Cada àrea de joc es correspon la *window* visible de l'escena. Per simplificar, un *clon* esta modelat per una esfera, de centre *(x, y, z)* i radi, *r* i la seva direcció inicial de moviment és (1, 0, 0).

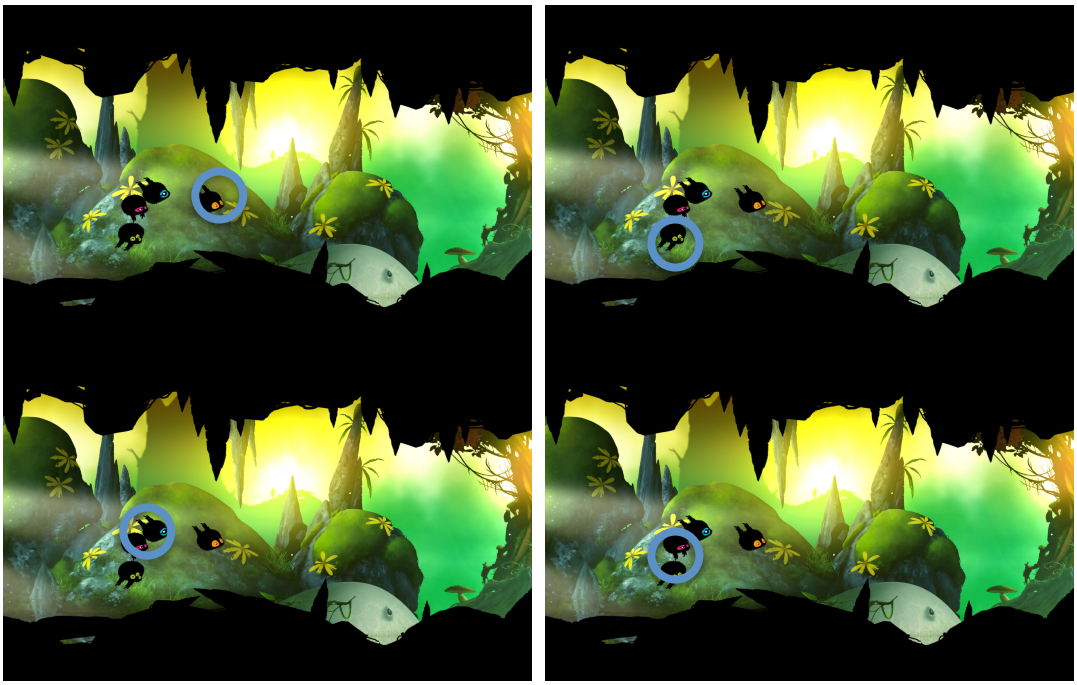
Tot suposant que tens una aplicació gràfica estructurada com el codi de la pràctica, respon les següents preguntes:

a) Defineix les classes i els atributs necessaris per a obtenir les vistes en les classes corresponents, així com les relacions entre elles. A quina classe es crearien les escenes? Quantes càmeres són necessàries? (10 punts)

b) El joc comença visualitzant frontalment la part de món de 100x100x1000 de les X's més petites. Suposa que l'observador enfoca al mig d'aquesta part de món des d'una distància de 800. Defineix els atributs de la/es càmera/es necessaris per a definir aquesta primera visualització.

Quan comença el joc, s'activa un *timer* i a cada pas de temps, es canvia la visualització per a donar l'efecte que la cova i les seves parets es desplacen cap a les X negatives del *widget*, un desplaçament de 10 unitats de món. Si cap usuari no interacciona amb el joc, és possible fer aquest efecte canviant només paràmetres de la càmera? Quins? Raona quines matrius hauries de re-calcular i quines etapes del *pipeline* hauries de tornar a executar. (10 punts)

c) Quan l'usuari interacciona amb el seu *clon*, ho fa des de la àrea d'interacció corresponent i només afecta al *clon* concret. El desplaçament fet per l'usuari amb el dit es correspon a un desplaçament mesurat en píxels. En cada àrea d'interacció (*viewport*) es mapeja tot el joc visible, tal i com mostra la figura.



Suposa que el mètode mousePressEvent(QMouseEvent *event) tracta els events de prémer la tableta i captura la posició on s'ha clicat a l'atribut *position*. Amb el mètode GLWidget::*keyReleaseEvent*(QKeyEvent *event), es tracten els events de deixar de prémer la pantalla. Fixa't que les dues posicions estan donades en píxels.

Quina transformació viewport-window serà necessària per a obtenir els punts corresponents als píxels en coordenades de món? Detalla les transformacions necessàries per a actualitzar el *clon* número 3 després de capturar el desplaçament fet pel jugador blau (el de sota i a l'esquerra), suposant que el clon es mou directament d'una posició a l'altra, segons la direcció definida pels punts inicial i final i sense tenir en compte els rebots contra les parets de la cova. (25 punts)

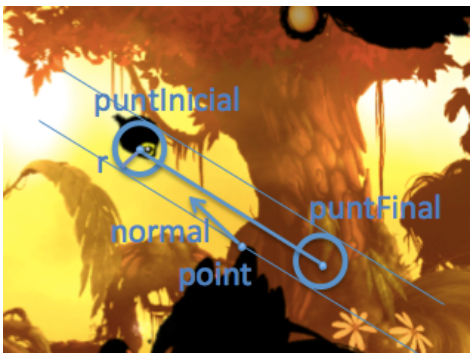
d) Cada escena té la seva il·luminació pròpia però comparteixen la mateixa llum puntual. Les parets de la cova només tenen un comportament difús, ja sigui directe o indirecte, i s'utilitza la il·luminació de Gouraud. Com quedarà simplificat el model de Blinn-Phong per a només modelar aquesta il·luminació?. La part central de la cova s'il·lumina amb una modificació

del càlcul del *shading* on només es volen remarcar les siluetes 2D dels objectes, és a dir, només tenen color el píxels tals que l'angle entre la normal i la direcció de visió sigui diferent de 0. El color del píxel es calcula directament amb la component difusa del material de l'objecte multiplicada per (1- cos (α)), sent α l'angle entre el vector de visió i la normal associada al píxel. Raona on es realitza aquest càlcul de la il·luminació. Implementa els *shaders* que es necessiten. (15 punts)

OPCIONAL:

e) Addicionalment, si entremig del moviment, el *clon* col·lisiona amb una paret de la cova, es trenca el moviment en dues parts. Suposa que tens ja implementat el mètode següent:

```
bool escena:: intersecta      ( vec4 puntInicial, vec4 puntFinal, float r,
                                vec4 &point, vec3 &normal)
```



que a partir del segment de recta definit entre el puntInicial i el puntFinal en coordenades de món, retorna cert hi ha alguna intersecció entre l'escena i el camí de gruix 2r que hi ha entre el puntInicial i el puntFinal. En el cas que retorni cert, retorna també el primer punt d'intersecció, point, i la normal associada a aquest punt. Retorna fals, en el cas que no hi hagi intersecció.

Per a calcular la direcció de rebot, pots utilitzar la fórmula de càlcul del vector reflectit utilitzada a Phong ($r = 2 (d \cdot n) n – d$, sent d la direcció del vector incident i n la normal a la superfície on es col·lisiona). Suposa que el moviment de rebot té sempre una longitud de 2 unitats de món. Quins passos s'han de realitzar per a calcular la posició final del clon? (10 punts)