

Players (players.*)

Pes:60%

Escriu VS+FS per a representar gràficament una pista de pàdel i la posició dels quatre jugadors:

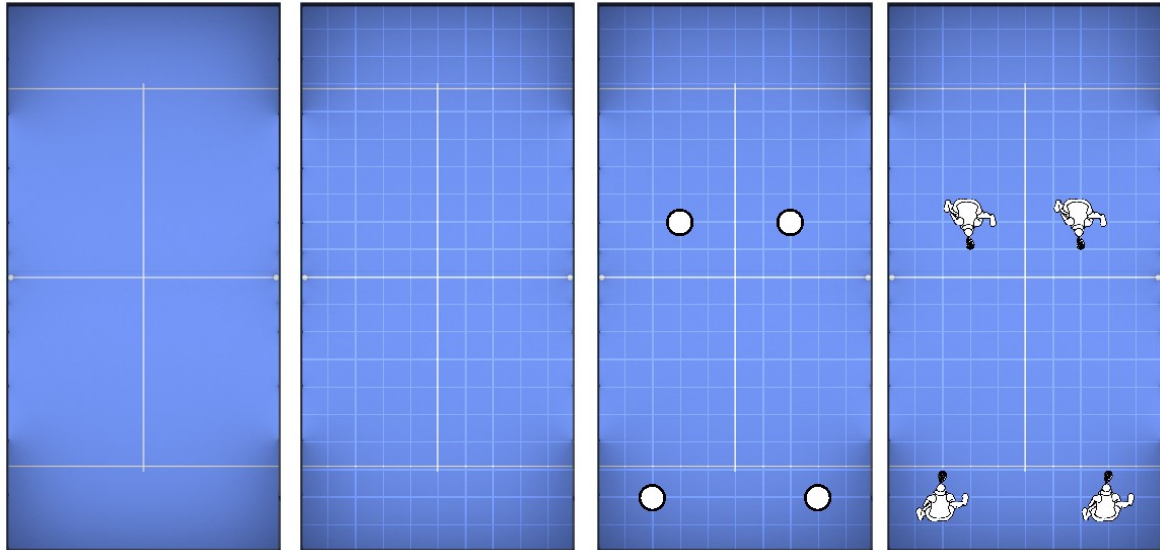


Figura 1: Resultat esperat amb l'objecte plane, pels modes 0, 1, 2 i 3.

En aquest problema només usarem l'objecte **plane.obj**, que com sabeu té vèrtexs de la forma $[\pm 1, \pm 1, 0]$.

El VS farà les tasques imprescindibles, que inclouen:

- Abans d'escriure `gl_Position` en clip space, cal aplicar-li al vèrtex (en object space) un escalat que només modifiqui la coordenada Y, de format que el quadrat es converteixi en un rectangle amb relació d'aspecte 1:2 (el doble d'alçada que amplada).
- El VS passarà al FS un **out vec2 P** amb les coordenades XY del vèrtex original en object space, però escalades de forma que omplin les dimensions d'una pista de pàdel (10m x 20m). És a dir, cal escalar XY amb els factors adients per passar de coordenades XY inicialment en $[-1, 1]$ a una coordenada X en $[-5, 5]$ i una coordenada Y en $[-10, 10]$. Observeu que l'origen en aquest espai (diguem-li, **court space**) és al centre de la pista, i que les seves unitats són metres.

El FS mostrarà una representació gràfica de la pista de pàdel i dels jugadors, depenent d'una variable **uniform int mode = 0**; les puntuacions màximes que teniu per cada mode són orientatives.

Si mode és 0 **[2 punts]**, simplement cal aplicar a l'objecte una primera textura (**uniform sampler2D courtMap**, usarem **court.jpg**), usant les coordenades de textura habituals.

Si mode és ≥ 1 **[3 punts]**, volem dibuixar una graella rectangular a sobre de la textura de la pista (vegeu figura), amb cel·les de mida 1 metre. Per fer-ho, us heu de basar en la variable P, que indica la posició del fragment en court space. Un fragment amb posició P correspon a una de les línies verticals de la graella si $P.x$ pertany a algun dels intervals $[0, 0.05)$, $[1, 1.05)$, $[2, 2.05)$... de forma que les línies tindran un gruix de 5cm. De forma anàloga, podeu saber si el fragment pertany a alguna de les línies horitzontals. Per aquests fragments que pertanyen a la graella, el seu color serà el de la textura **court.jpg** però multiplicant les components RGB per 1.2 (per tal que es vegin més clares).

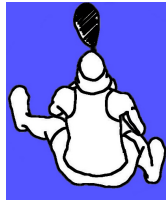
Si mode és 2 [**2 punts**], dibuixarem un cercle de radi 0.5m (en court space) centrat en cada jugador. La posició XY de cada jugador, en court space, vindrà indicada per les variables:

```
uniform vec2 p1 = vec2(-3, -8); // 3m cap a l'esquerra (-X), 8m cap a baix (-Y)
uniform vec2 p2 = vec2( 3, -8);
uniform vec2 p3 = vec2(-2, 2);
uniform vec2 p4 = vec2( 2, 2);
```

El cercle al voltant de cada jugador estarà format per un cercle interior (de color blanc i radi 0.4m) i un altre exterior (de color negre i radi 0.5m):



Finalment, si mode és 3 [**3 punts**], en comptes de representar els jugadors amb cercles, farem servir una segona textura (**uniform sampler2D player1**, usarem **player3.jpg**):



En comptes de cercles de radi 0.5m, la textura s'aplicarà en un quadrat de costat 2m centrat en la posició del jugador. Si el fragment pertany al quadrat d'un jugador, haureu de calcular unes coordenades de textura adjacents per prendre la mostra de la textura, de forma que la textura ompli tot aquest quadrat (pista: pel jugador que és a p1, bàsicament heu d'escalar i traslladar (p1-P). El color del fragment serà el color C de la textura només si C.r > 0.5 o C.b < 0.5 (el fons blau de player3.jpg no el volem dibuixar); altrament el color del fragment serà el mateix que amb mode 1. Observeu que la coordenada de textura t l'heu d'invertir, si escau, de forma que els jugadors sempre mirin cap a l'altre costat de la pista, com a la figura (podeu usar el signe de la coordenada Y del jugador per determinar si cal o no invertir la t).

Identificadors obligatoris:

players.vert, players.frag (en minúscules!)

```
uniform int mode = 0;
uniform vec2 p1 = vec2(-3, -8);
uniform vec2 p2 = vec2( 3, -8);
uniform vec2 p3 = vec2(-2, 2);
uniform vec2 p4 = vec2( 2, 2);
```

La resta d'uniforms necessaris segons l'enunciat.

(*) Podeu lliurar els .vert, .frag de forma individual, o en un zip sense carpetes a dins.

Outliner (outliner.*)

Pes:40%

Escriu VS+FS per dibuixar el model (per exemple, **player.obj**) amb una il·luminació que ressalti la seva silueta.

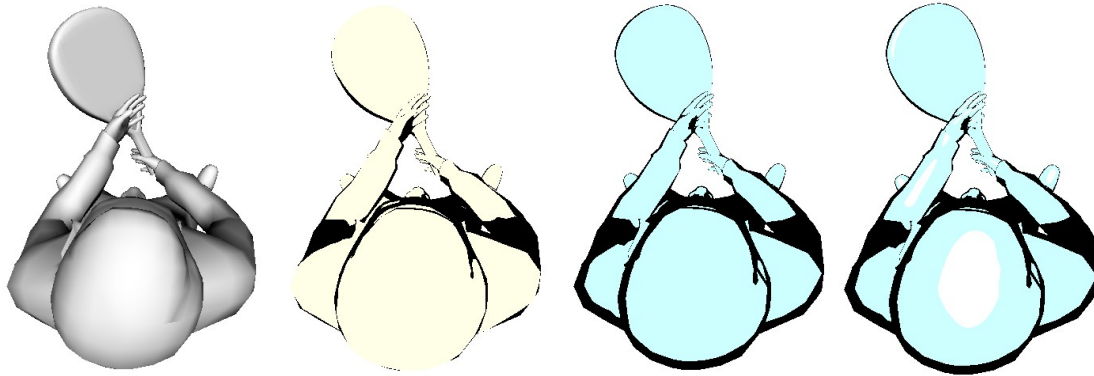


Figura 2: Resultat esperat (player.obj) amb els modes 0, 1, 2 i 3.

La puntuació de cada apartat és orientativa

El VS **[5 punts]** farà les tasques imprescindibles, incloent-hi:

- Aplicar al vèrtex (quan encara és en object space) una rotació de 90 graus al voltant de l'eix X. Aquesta serà la transformació de modelat de l'objecte (teniu la matrix de rotació a sota)
- De forma similar, cal aplicar la mateixa rotació a la normal.
- Passar-li al FS dues variables vec3, una amb la normal rotada N i una altra amb el vèrtex rotat P, **tots dos en eye space**.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

El FS calcularà el color del fragment d'acord amb **uniform int mode = 0**. Sigui z' la component Z de la normal, en eye space.

Si mode és 0 **[1 punt]**, el color del fragment serà, simplement, el color RGBA que té per components z' .

Si mode és 1 **[1 punt]**, primer calculareu un color serà blanc o negre, depenent de si z' supera o no el valor 0.4. El color final serà aquest blanc/negre, però multiplicant la component blava per 0.9 (el blanc quedarà com un groc clar).

Si mode és 2 **[2 punts]**, cal que calculeu el view vector V (vector unitari en la direcció del punt a la càmera, en eye space). Primer calculareu un color serà blanc o negre, depenent de si $V \cdot N$ supera o no el valor 0.4. El color final serà aquest blanc/negre, però multiplicant la component vermella per 0.8 (el blanc quedarà com un cian clar).

Si mode és 3 **[1 punt]**, el color es calcularà com al mode 2, amb una modificació: si $V \cdot N > 0.95$ (és a dir, si la cara és molt perpendicular al vector de visió), el color serà blanc, simulant una taca especular.

Identificadors obligatoris:

outliner.vert, outliner.frag (en minúscules!)

Els uniforms de l'enunciat.(*) Podeu lliurar els .vert, .frag de forma individual, o en un zip sense carpetes.