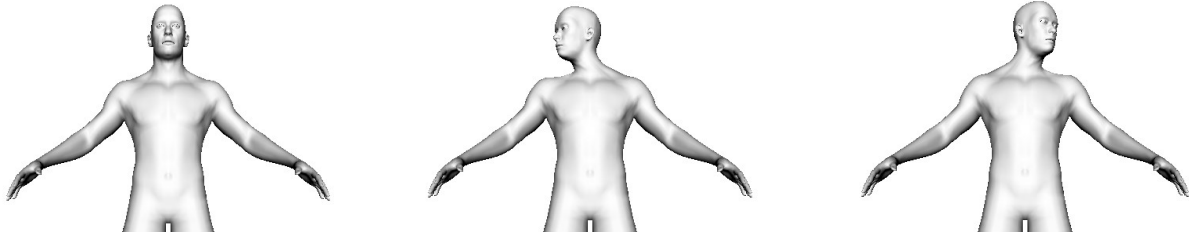


## Look2 (look2.\*)

Escriu VS+FS per a girar el cap del model **man.obj** al voltant de l'eix Y:



El VS farà les següents tasques. L'angle de rotació, en radians, estarà determinat per

`uniform float angle = 0.5;`

Sigui P el vèrtex original, i sigui P' el resultat d'aplicar-li al vèrtex la rotació respecte l'eix Y (tots dos en *object space*). Volem que la rotació afecti només a la part superior del cos (coll i cap). Assumirem que el coll comença a Y=1.45 i acaba a Y=1.55.

La posició del nou vèrtex, encara en *object space*, l'heu de calcular com la interpolació lineal entre P i P' segons un paràmetre d'interpolació t. Aquest paràmetre ha de variar suaument (smoothstep) entre 0 i 1, de forma que

- $t=0$  quan  $\text{vertex.y} \leq 1.45$  (la rotació no tindrà efecte),
- $t=1$  quan  $\text{vertex.y} \geq 1.55$  (rotació completa),
- i hi hagi una interpolació suau pels vèrtexs amb coordenada Y entre 1.45 i 1.55.

La nova normal l'heu de calcular interpolant la normal original amb la normal rotada, usant el mateix paràmetre d'interpolació t. Aquesta nova normal, un cop transformada a *eye space*, us donarà la direcció de la normal N que usareu per calcular el color del vèrtex, que tindrà per components el valor de N.z.

Recordeu que la matriu de rotació respecte l'eix Y té la forma:

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

El FS farà les tasques per defecte.

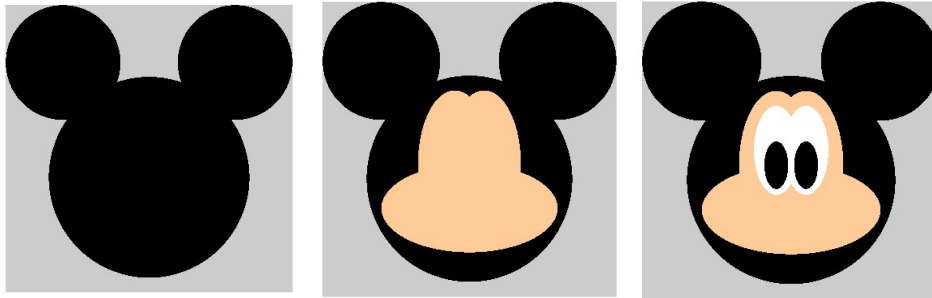
**Identificadors obligatoris:**

`look2.vert`, `look2.frag`

Tots els uniform's de l'enunciat.

## Mouse (mouse.\*)

Escriu **VS+FS** que, amb l'objecte **plane.obj**, dibuixi de forma procedural un ratolí similar a aquest:

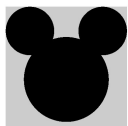


El VS farà les tasques imprescindibles.

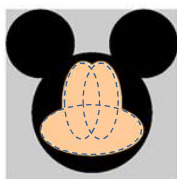
El FS calcularà el color per tal reproduir quelcom *semblant* a la figura. El color resultant serà blanc, negre, gris (0.8) o color pell (1.0, 0.8, 0.6), depenent de les coordenades de textura del fragment. El nivell de detall dependrà de

```
uniform int mode=2;
```

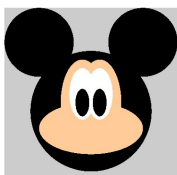
**Si mode és 0 [6 punts de la nota]:** només cal reproduir la silueta de la figura, formada per tres cercles negres sobre el fons gris:



**Si mode és 1 [2 punts de la nota],** cal afegir tres òvals de color pell, com a la figura. Una forma d'obtenir aquests òvals és tractar-los com cercles escalats. Per saber si un determinat punt P és dins d'un cercle centrat a C amb radi R, compareu la longitud del vector  $v=P-C$  amb el radi R. Pels òvals, podeu escalar el vector v (per exemple, multiplicant/dividint la component X) abans de calcular-ne la longitud.



**Si mode és 2 [2 punts de la nota],** cal afegir els quatre òvals per definir els ulls:



**Identificadors obligatoris:**

mouse.vert, mouse.frag

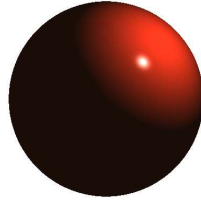
Tots els uniform's de l'enunciat.

(\*) Test orientatiu. No cal una sortida idèntica al test.

## Sphere (sphere.\*)

---

Escriu VS+FS que, amb l'objecte **plane.obj**, dibuixi de forma procedural una esfera il·luminada:

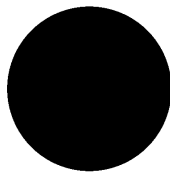


El VS farà les tasques per defecte que resultin imprescindibles. A més a més, el VS escalarà les coordenades de textura per tal que siguin a l'interval  $[-1, 1]$  (originalment pel plane són a  $[0, 1]$ ).

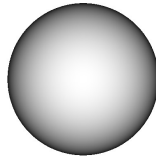
El FS s'encarregarà de determinar els fragments que pertanyen a l'esfera (proceduralment), i de calcular-ne la il·luminació. El resultat dependrà de la variable

`uniform int mode=2;`

**Si mode=0 [2 punts de la nota]**, el FS pintarà de negre els fragments que, d'acord amb les seves coordenades de textura (s,t), siguin dins d'un cercle de radi 1 centrat a l'origen. El FS descartarà els fragments fora del cercle:

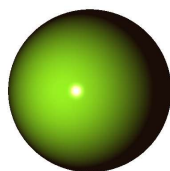


**Si mode=1 [3 punts de la nota]**, el FS calcularà, pels fragments no descartats, una posició P i un vector normal N calculats com  $P = N = (s, t, \sqrt{1 - s^2 - t^2})$ . El color del fragment serà simplement el gris que té per components N.z:



**Si mode = 2 [5 punts de la nota]**, el FS calcularà la il·luminació usant la posició P i el vector N calculats abans (els quals haureu de considerar que estan en *object space* i per tant haureu de passar a *eye space*). El model d'il·luminació a aplicar serà el de Phong, com a l'exercici Lighting(4):

$$K_a I_a + K_d I_d (N \cdot L) + K_s I_s (R \cdot V)^s$$



### Identificadors obligatoris:

```
sphere.vert, sphere.frag
uniform int mode=2;
uniform vec4 matAmbient, matDiffuse, matSpecular;
uniform float matShininess;
uniform vec4 lightAmbient, lightDiffuse, lightSpecular, lightPosition;
```