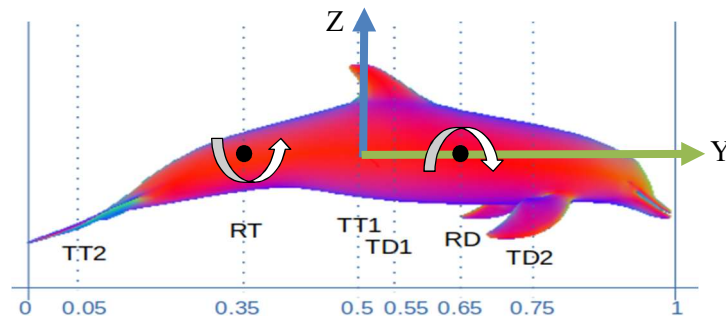


Dolphin (dolphin.*)

Volem simular l'animació d'un dofí nadant. Per aconseguir-ho cal que el VS deformi el model tenint en compte les següents indicacions. L'animació durarà **un segon** i s'anirà repetint en el temps (caldrà fer servir una funció sinusoïdal amb un període apropiat). Per tal d'aplicar les transformacions corresponents dividirem el dofí en dues meitats i aplicarem una rotació a cada meitat en funció del temps.

Model: Per al model **dolphin.obj** amb els següents punts (tingueu en compte que els valors de cada punt de la figura són en relació a la llargària de la caixa contenidora en direcció Y):



Punts de rotació: La rotació de cada meitat serà al voltant d'un eix paral·lel a l'eix X i que passa per un punt de la forma $(0, y, 0)$, on la y varia segons la meitat. Per la meitat davantera, el punt de rotació serà RD, i per la meitat posterior el punt de rotació serà RT.

Transició de la deformació: La transformació s'aplicarà de manera suau (combinant amb smoothstep els vèrtexs originals i els transformats) en l'eix Y des del punt TD1 (on no hi haurà cap transformació) fins al punt TD2 (on la transformació serà màxima) per la part davantera, i des de TT1 fins a TT2 per la part posterior.

Angles de la rotació: L'angle de rotació per la part davantera variarà en $[-\pi/32, \pi/32]$, i per la part posterior en $[-\pi/4, 0]$. Tingueu en compte que les rotacions seran en sentits oposats, fent que el cap i la cua pugin i baixin a l'hora amb un petit offset de temps.

Offset entre parts: L'animació de la part davantera començarà 0.25 segons abans que la de la part posterior. És a dir, si usem $\text{time} = 0$ en l'animació de la part posterior, la part davantera es comportarà com si $\text{time} = 0.25$.

El color del dofí serà el gris clar $(0.8, 0.8, 0.8)$, al qual el VS aplicarà il·luminació bàsica tenint en compte la component Z de la normal en *eye space*.

El FS farà els càlculs imprescindibles per a la visualització. Aquí tens els resultats esperats amb el model **dolphin.obj** per a diferents instants de temps ($t=0s$, $1.25s$, $2.75s$):



$t = 0s$



$t = 1.25s$



$t = 2.75s$

Identificadors (ús obligatori):

dolphin.vert, dolphin.frag

uniform float time;

const float PI = 3.1416;

Flag (no hi ha test)

Escriu **VS+FS** que, amb l'objecte **plane.obj**, dibuixi de forma procedural una bandera similar a aquesta:



El VS farà les tasques imprescindibles, escalant la coordenada Y per tal que la relació d'aspecte sigui 2:1.

El FS calcularà el color per tal reproduir quelcom *semblant* a la figura.

Identificadors obligatoris:

flag.vert, flag.frag

Beach (beach.*)

Escriu un **VS** i un **FS** per tal de simular una finestra a través de la qual es pot veure una palmera moguda pel vent, i un fons. Usarem les textures **interior.png**, **palm-tree2.png** i **dunes.jpg**:



La composició que volem obtenir (amb l'objecte **plane.obj**) la teniu a la dreta.

El VS, a banda de les tasques imprescindibles, li passarà al FS la **normal N** en eye space.

El FS usará tres textures, que haureu de declarar així:

```
uniform sampler2D window;           // interior
uniform sampler2D palm1;             // palm-tree
uniform sampler2D background2;      // dunes
```

Primer accedirà a la textura **window** (amb les coordenades de textura habituals) per obtenir un color que li direm **C**.

Si la component alfa de **C** és 1.0 (part opaca de la finestra), el color del fragment serà **C**.

Si la component alfa de **C** és inferior a 1.0, per calcular el color del fragment s'accedirà a la textura **palm1** amb coordenades de textura **vtexCoord + 0.25*N.xy + vec2(0.1*sin(2*time)*vtexCoord.t, 0)** per obtenir un color que li direm **D**.

Si la component alfa de **D** és superior o igual a 0.5 (part opaca de la palmera), el color del fragment serà **D**. Altrament, el color del fragment serà el color de la textura **background2** al punt **vtexCoord + 0.5*N.xy**.

Observeu que estem usant un offset en les coordenades de textura que depèn de les components de la normal en eye space. Degut a aquest offset, la part visible de la palmera i del fons dependrà de l'orientació del model. En el cas de la palmera, també hi ha un desplaçament horitzontal que depèn del temps, per simular l'ondulació produïda pel vent.

Identificadors (ús obligatori):

```
beach.vert, beach.frag (minúscules!)
uniform float time;
uniform sampler2D window;
uniform sampler2D palm1;           // observeu el dígit 1 al final
uniform sampler2D background2;    // observeu el dígit 2 al final
```

