

**Picking di oggetti con il mouse  
in 3D**

Dopo la chiamata della call-back function all'evento mouse con il tasto sinistro down, xpos e ypos sono le coordinate in pixel ( $[0, width]$ ,  $[0, height]$ )

Dobbiamo trasformarle in coordinate Normalized Device Coordinates (NDC):

$$\begin{aligned} NDC_x & \quad [0, width] \rightarrow [-1, 1] \\ NDC_y & \quad [0, height] \rightarrow [-1, 1] \end{aligned}$$

$$NDC_x = 2 \frac{x}{width} - 1$$

$$NDC_y = 2 \frac{y}{height} - 1$$

Ma le coordinate dello schermo nella direzione y sono invertite:

$$NDC_x = 1 - 2 \frac{y}{height}$$

Adesso abbiamo un punto 3D nelle coordinate Normalized Device Coordinates (NDC).  
Indichiamo con  $P_{clip}$  questo punto

$$P_{clip} = \begin{bmatrix} NDC_x \\ NDC_y \\ -1 \\ 1 \end{bmatrix}$$

$$P_{clip} = Projection \cdot View \cdot Model \cdot P$$

Il passo successivo è calcolare un raggio nelle coordinate del mondo che abbia origine nella posizione della telecamera e direzione individuata dalla posizione del mouse.

Dobbiamo trovare le coordinate del mondo che corrispondono a  $P_{clip}$

$$Projection^{-1} P_{clip} = View \cdot Model \cdot P$$

$$P_w = View^{-1} \cdot Projection^{-1} P' = Model \cdot P$$

```
// Coordinate nel clip space
vec4 P_clip = vec4(ndc.x, ndc.y, ndc.z, 1.0);

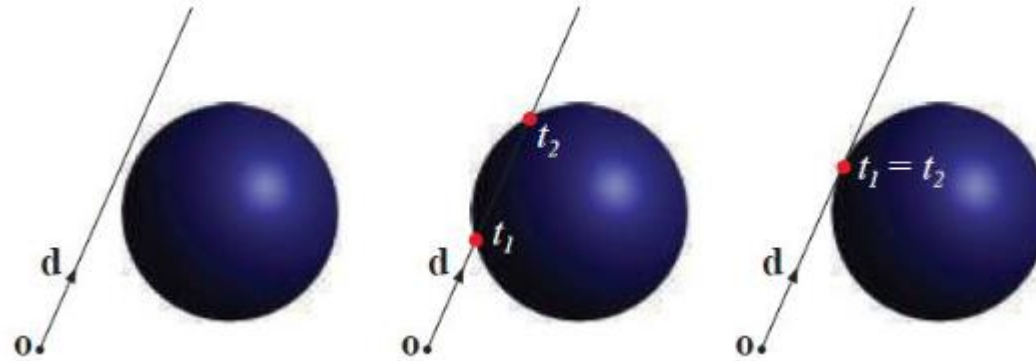
// Le coordinate Nell' eye space si ottengono premoltiplicando per
l'inversa della matrice Projection.

vec4 ViewModelp = inverse(Projection) * P_clip;

ViewModelp.w = 1;
// le coordinate nel world space: si ottengono premoltiplicando per
l'inversa della matrice di Vista

vec4 Pw = inverse(View) * ViewModelp;
//Direzione dalla posizione della telecamera al punto Pw
vec3 ray_wor = normalize(vec3(Pw) - vec3(SetupTelecamera.position));
```

Successivamente, calcoliamo l'intersezione tra il raggio che parte dalla telecamera nella direzione precedentemente calcolata e la sfera di raggio prefissato che ha centro nell'ancora di ogni oggetto.



Una sfera può essere definita matematicamente da un centro  $\mathbf{c}$  e da un raggio  $r$ .  
Una formula più compatta per rappresentare una sfera è la seguente:

$$f(p) = \|p - c\|_2 - r = 0$$

Per calcoliar l'intersezione tra una sfera ed il raggio  $r(t)$ , sostituiamo  $r(t)$  al posto di  $p$  nell'equazione precedente:

$$f(r(t)) = \|r(t) - c\|_2 - r = 0$$

L'equazione del raggio che parte dalla posizione della telecamera  $\mathbf{O}$  nella direzione  $\mathbf{d}$  individuata con il click del mouse si esprime come:

$$r(t) = O + t \cdot d$$

Sostituendo questa espressione nella precedente:  $\|O + t \cdot d - c\|_2 - r = 0$

Di seguito indichiamo con  $a \cdot b$  il prodotto scalare tra due vettori

$$(O + t \cdot d - c) \cdot (O + t \cdot d - c) = r^2$$

$$(O - c + td) \cdot (O - c + td) = r^2$$

$$t^2 \cdot d^T d + 2t(d \cdot (O - c)) + (O - c) \cdot (O - c) - r^2 = 0$$

Poiché  $d$  è un vettore normalizzato  $d^T d = \|d\|_2^2 = 1$

$$t^2 + 2t(d \cdot (O - c)) + (O - c) \cdot (O - c) - r^2 = 0$$

L'equazione risultante è un'equazione di secondo grado in  $t$ .

Se il raggio interseca la sfera la interseca al massimo in due punti, per due valori del parametro  $t$ .

Se le soluzioni sono immaginarie, il raggio non interseca la sfera.

$$t^2 + 2tb + q = 0$$

$$b = d \cdot (O - c)$$

$$q = (O - c) \cdot (O - c) - r^2$$

Calcoliamo il discriminante dell'equazione di secondo grado:

$$\Delta = b^2 - q$$

$b^2 - q < 0$       Il raggio non interseca la sfera, si evita di fare i calcoli.

$$b^2 - q \geq 0 \qquad t_0 = -b - \sqrt{\Delta} \qquad t_1 = -b + \sqrt{\Delta}$$

Interessa l'intersezione più vicina all'osservatore, quella con valore di  $t$  inferiore

$$t_0 = -b - \sqrt{\Delta}$$

Se  $t_0 < 0$ , l'intersezione avviene dietro l'osservatore e si restituisce FALSE

Costruire la funzione

```
bool ray_sphere(vec3 O, vec3 d, vec3 sphere_centre_wor, float sphere_radius,
float* intersection_distance
```

Questa funzione controlla se il raggio che parte dall'occhio  $O$  della telecamera nella direzione  $d$ , corrispondente al click effettuata dall'utente, interseca la sfera che ha centro nell'ancora dell'oggetto in coordinate del mondo  $\text{sphere\_centre\_wor}$  e raggio  $\text{sphere\_radius}$ . In caso negativo, restituisce false. Rigetta le intersezioni dietro l'origine del raggio, e pone  $\text{intersection\_distance}$  all'intersezione più vicina.



Per ogni oggetto della scena:

- Interseco il raggio che esce dalla camera nella direzione del mouse con la sfera centrata nell'ancora di tutti gli oggetti posizionati nel mondo per individuare se c'è intersezione con l'oggetto

L'oggetto selezionato è quello che è più vicino all'osservatore, quello che è caratterizzato da un'intersezione corrispondente al valore di  $t$  minimo