Laboratorio 7

Scaricare i file del progetto

Laboratorio7Curve_Studenti

- Implementare l'algoritmo di De Casteljeau per la valutazione di una curva di Bezier a partire da vertici di controllo selezionati con il mouse sulla viewport sullo schermo.
- Dare la possibilità all'utente di visualizzare il poligono di controllo.
- Verificare sperimentalmente che le curve di Bezier non permettono il controllo locale della curva.

Scaricare i file del progetto

Laboratorio7_Studenti

a) Implementare la possibilità di selezionare un oggetto della scena, facendo click con il tasto sinistro sull'oggetto da selezionare, nel seguente modo:
    - Definire un punto di ancoraggio per ogni oggetto nella scena.
    - Tracciare il raggio che parte dalla posizione della telecamera nella direzione individuata con il click del mouse.
    - Calcolare l'intersezione del raggio con una sfera centrata nel punto di ancoraggio di ogni oggetto nella scena e raggio prefiddato. Se c'è intersezione del raggio con uno di questi oggetti si individua quale oggetto sia stato selezionato.

Si tratta di:

Scrivere la funzione:

```
vec3 get_ray_from_mouse(float mouse_x, float mouse_y)
```

che prese in input le coordinate x,y del mouse nel sistema di coordinate dello schermo, ne restituisce in vec3 le corrispondenti coordinate nel sistema di coordinate del mondo.

(ricordare di porre z=-1 nello spazio di clipping).

1) Dalle coordinate di schermo alle coordinate normalizzate (NDC)

    #Mappatura di x che appartiente all'intervallo [A,B) in [l,r]

    xm=(x-A)*(r-l)/(B-A)+l

    Mappare xmouse da [0,width] in [-1,1]

    Mappare ymouse da [heigth,0] in [-1,1]

2) Dalle coordinate normalizzate alle coordinate di clipping (z=-1),
3) Trasformare le coordinate di clipping in coordinate di Vista, premoltiplicando per l'inversa della matrice di Proiezione.
4) Trasformare le coordinate di vista in coordinate del mondo premoltiplicando le coordinate di vista per l'inversa della matrice di Vista.

Scrivere la funzione

```
bool ray_sphere(vec3 ray_origin_wor, vec3 ray_direction_wor, vec3
sphere_centre_wor, float sphere_radius, float* intersection_distance)

*controlla se un raggio interseca una sfera. In caso negativo,
restituisce false. Rigetta
le intersezioni dietro l'origine del raggio, e pone
intersection_distance all'intersezione p iù vicina.
*/
```

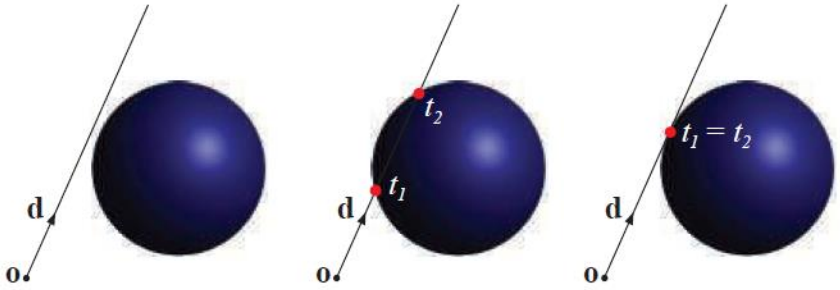

Figure 22.10. The left image shows a ray that misses a sphere and consequently $b^2 - c < 0$. The middle image shows a ray that intersects a sphere at two points ($b^2 - c > 0$) determined by the scalars t_1 and t_2. The right image illustrates the case where $b^2 - c = 0$, which means that the two intersection points coincide.

22.6.1 Mathematical Solution

A sphere can be defined by a center point, $\mathbf{c}$, and a radius, r. A more compact implicit formula (compared to the one previously introduced) for the sphere is then

$$f(\mathbf{p}) = ||\mathbf{p} - \mathbf{c}|| - r = 0, \tag{22.5}$$

where $\mathbf{p}$ is any point on the sphere's surface. To solve for the intersections between a ray and a sphere, the ray $\mathbf{r}(t)$ simply replaces $\mathbf{p}$ in Equation 22.5 to yield

$$f(\mathbf{r}(t)) = ||\mathbf{r}(t) - \mathbf{c}|| - r = 0. \tag{22.6}$$

Using Equation 22.1, that $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, Equation 22.6 is simplified as follows:

$$\begin{gathered}
||\mathbf{r}(t) - \mathbf{c}|| - r = 0 \\
\Longleftrightarrow \\
||\mathbf{o} + t\mathbf{d} - \mathbf{c}|| = r \\
\Longleftrightarrow \\
(\mathbf{o} + t\mathbf{d} - \mathbf{c}) \cdot (\mathbf{o} + t\mathbf{d} - \mathbf{c}) = r^2 \\
\Longleftrightarrow \\
t^2(\mathbf{d} \cdot \mathbf{d}) + 2t(\mathbf{d} \cdot (\mathbf{o} - \mathbf{c})) + (\mathbf{o} - \mathbf{c}) \cdot (\mathbf{o} - \mathbf{c}) - r^2 = 0 \\
\Longleftrightarrow \\
t^2 + 2t(\mathbf{d} \cdot (\mathbf{o} - \mathbf{c})) + (\mathbf{o} - \mathbf{c}) \cdot (\mathbf{o} - \mathbf{c}) - r^2 = 0.
\end{gathered} \tag{22.7}$$

The last step comes from the fact that $\mathbf{d}$ is assumed to be normalized, i.e., $\mathbf{d} \cdot \mathbf{d} = ||\mathbf{d}||^2 = 1$. Not surprisingly, the resulting equation is a polynomial of the second order, which means that if the ray intersects the sphere, it does so at up to two points. See Figure 22.10. If the solutions to the equation are imaginary, then the ray misses the sphere. If not, the two solutions t_1 and t_2 can be inserted into the ray equation to compute the intersection points on the sphere.

The resulting Equation 22.7 can be written as a quadratic equation:

$$t^2 + 2bt + c = 0, \tag{22.8}$$

where $b = \mathbf{d} \cdot (\mathbf{o} - \mathbf{c})$ and $c = (\mathbf{o} - \mathbf{c}) \cdot (\mathbf{o} - \mathbf{c}) - r^2$. The solutions of the second-order equation are shown below:

$$t = -b \pm \sqrt{b^2 - c}. \tag{22.9}$$

Note that if $b^2 - c < 0$, then the ray misses the sphere and the intersection can be rejected and calculations avoided (e.g., the square root and some additions). If this test is passed, both $t_0 = -b - \sqrt{b^2 - c}$ and $t_1 = -b + \sqrt{b^2 - c}$ can be computed. An additional comparison needs to be done to find the smallest positive value of t_0 and t_1. See the collision detection chapter at realtimerendering.com for an alternate way of solving this quadratic equation that is more numerically stable [1446].

If these computations are instead viewed from a geometric point of view, then better rejection tests can be discovered. The next subsection describes such a routine.

All'interno della funzione di risposta all'evento pressione di un tasto sul mouse

Testare per ogni oggetto nella scena se il raggio che ha origine nella camera e direzione che punta verso la posizione del mouse interseca la sfera di raggio predefinito che ha per centro il centro dell'oggetto, e poi prendere l'oggetto che è più vicino all'osservatore.

b) Implementare

Telecamera che si muove avanti, (tasto w), indietro (tasto s), a sinistra (tasto a), a destra (tasto d), lungo una direzione.

c) Telecamera la cui direzione viene definita mediante l’evento passivemotion.

d) Costruire una scena formata da 30 pupazzi neve equidistribuiti in una griglia 5x6 e visitare la scena con la telecamera.