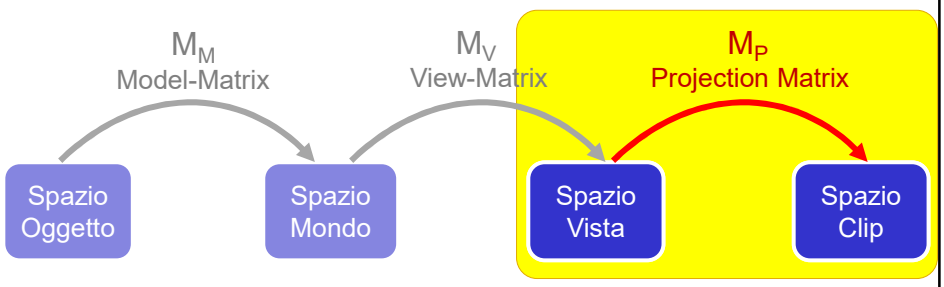
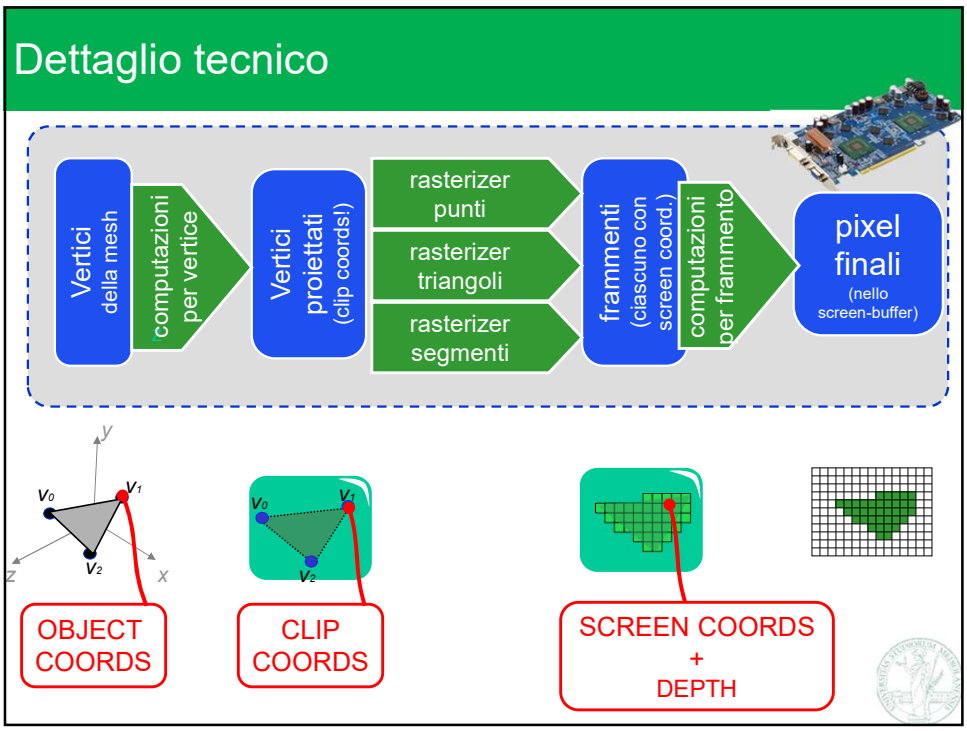
Note sulla matrice di proiezione prospettica



122

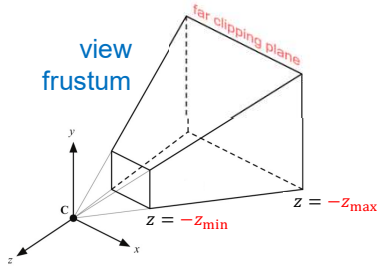
Dettaglio tecnico

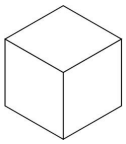


123

Limiti dello spazio visibile (inquadrato)

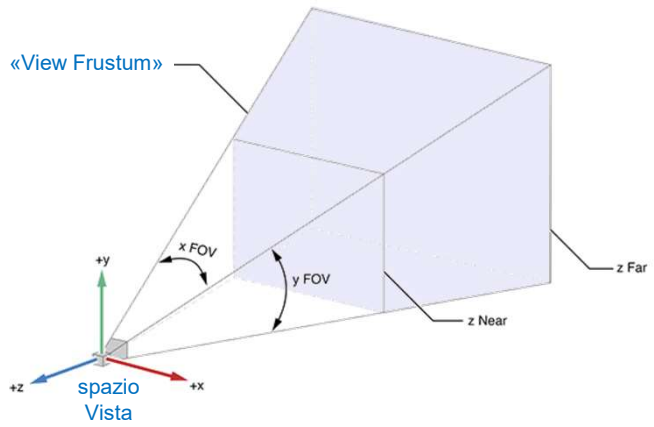
	Spazio Vista	Spazio Clip	Spazio Schermo	
x	«left clipping plane»	-1	0	pixel coordinates
	«right clipping plane»	$+1$	res_x	
y	«bottom clipping plane»	-1	0	
	«top clipping plane»	$+1$	res_y	
z	$-z_{min}$	-1	0	pixel depth
	$-z_{max}$	$+1$	1	





124

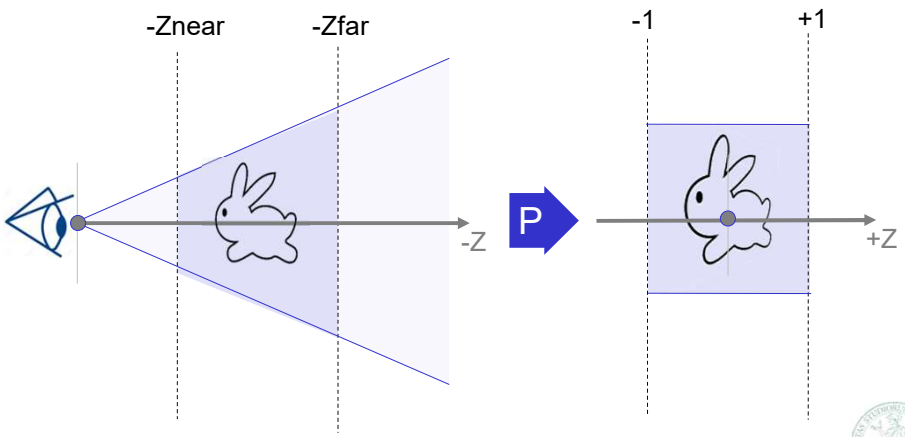
View Frustum



125

Modificare la matrice di proiezione per consentire il depth test

In Spazio Vista: In spazio Clip:



126

Depth test in openGL / webGL: passo per passo

- ✓ Il **processing per vertice** :
  ha il compito di produrre
  le **clip coords** del vertice
  - ⇒sotto forma di coordinate omogenee (x·w, y·w, z·w, w)
  - ⇒se X,Y,Z sono in [-1..+1], il vertice è nell'immagine
  - ⇒quando X,Y, **oppure Z** fuori da [-1..+1] → fuori da immagine!
  - ⇒cioè: se Z > +1 troppo lontano, non disegno
    cioè: se Z < -1 troppo vicino, non disegno
  - ⇒La matrice di proiezione deve quindi occuparsi anche di produrre la Z corretta in spazio clip
- ✓ Il rasterizzatore (hard wired) trasforma le coordiante clip x,y,z in coordinate schermo *x,y* e *depth*

127

Modificare la matrice di proiezione per consentire il depth test

- ✓Scelgo arbitrariamente
  - ⇒z_near : la distanza della cosa più *vicina* inquadrata nell'immagine
  - ⇒z_far : la distanza della cosa più *lontana* inquadrata nell'immagine
  - ⇒sono distanze definite in spazio vista!
    - sono due reali > 0
- ✓La trasf di proiezione deve portare l'intervallo Z (in spazio vista): [-z_near .. –z_far] nell'interallo Z (in spazio clip): [-1..+1]

128

Modificare la matrice di proiezione per consentire il depth test

- ✓Matrice di proiezione prospettica fin'ora:

divisione per la coord affine

$$\begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} d \cdot x \\ d \cdot y \\ d \cdot z \\ -z \end{pmatrix} \rightarrow \begin{pmatrix} -x \cdot d/z \\ -y \cdot d/z \\ -d \\ 1 \end{pmatrix}$$

perspective matrix — view coords — clip coords

valore di z costante
formalmente corretto (il punto finisce sempre sul piano immagine) ma inutile

129

Modificare la matrice di proiezione per consentire il depth test

✓Voglio modificarla in modo che la Z in output sia una qualche funzione della Z input

$$\begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & A & B \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} d \cdot x \\ d \cdot y \\ Az + B \\ -z \end{pmatrix} \rightarrow \begin{pmatrix} -x \cdot d/z \\ -x \cdot d/z \\ -A - B/z \\ 1 \end{pmatrix}$$

view coords

clip coords

che valori usare qui? (per A e B)

130

Modificare la matrice di proiezione per consentire il depth test

✓Troviamo i valori per A e B

✓Vogliamo che:

⇒se z vista = $-zNear$, z clip deve essere -1

⇒se z vista = $-zFar$, z clip deve essere $+1$

⇒quindi

$$-A + \frac{B}{zNear} = -1 \qquad -A + \frac{B}{zFar} = +1$$

✓Soluz:

$$A = \frac{zFar+zNear}{zFar-zNear} \qquad B = \frac{2 \cdot zFar \cdot zNear}{zFar-zNea}$$

(verificare!)

131

X e Y da Clip Space a Screen Space

$$f_{to\_screen}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} (x+1)/2 \cdot RES_X \\ (y+1)/2 \cdot RES_Y \\ (z+1)/2 \end{pmatrix}$$

in $[-1, +1]^3$

in $[0, 1]^3$

Se $RES_X \neq RES_Y$ questo corrisponde ad una scalatura non uniforme

132

Correzione dell'aspect ratio

✓ Matrice di proiezione prospettica con correzione dell'aspect ratio $a = \frac{RE_X}{RE_Y}$

$$\mathrm{P} = \begin{pmatrix} d & 0 & 0 & 0 \\ 0 & da & 0 & 0 \\ 0 & 0 & A & B \\ 0 & 0 & -1 & 0 \end{pmatrix} \quad \text{oppure} \quad \mathrm{P} = \begin{pmatrix} d/a & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & A & B \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

Nota: la trasformaz. di proiezione dipende solo dai parametri INTRINSECI della macchina fotografica… compreso l'aspect ratio del suo frame!

133

Matrice di Proiezione Prospettica finale

$$P = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d \cdot a & 0 & 0 \\ 0 & 0 & \dfrac{(z_F + z_N)}{(z_F - z_N)} & \dfrac{2 \cdot z_F \cdot z_N}{(z_F - z_N)} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

a aspect ratio
d lunghezza focale

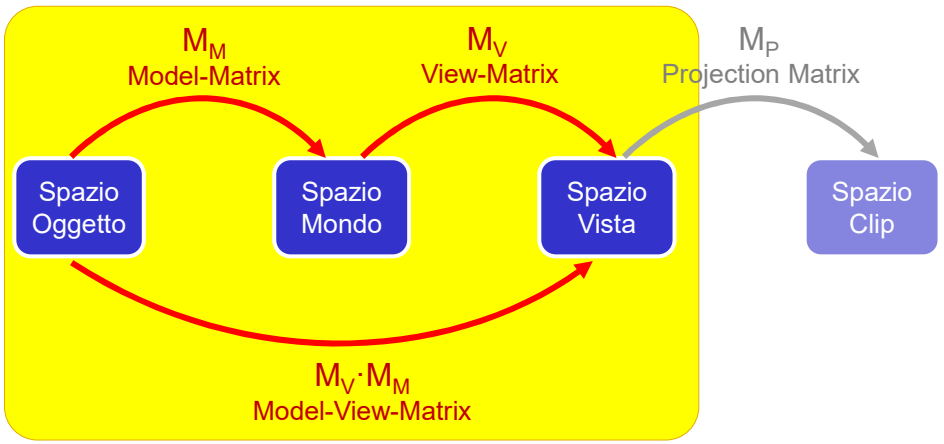
134

In three.js: oggetto di tipo camera (macchina fotografica virtuale)

- ✓ E' passato al renerer per disegnare la scena
- ✓ Parametri estrineseci:
  - ⇒ Determinati dai campi posizione e rotazione
  - ⇒ Producono la matrice di vista che viene memorizzata nel campo "camera.worldMatrixInverse"
- ✓ Parametri intriseci
  - ⇒ Determinati da FOV, aspect ratio, zmin e zfar
  - ⇒ Passati al costruttore
  - ⇒ Producono la matrice di proiezione, che viene memorizzata nel campo camera.projectionMatrix
- ✓ Può essere controllato da un controls
  - ⇒ Es. orbiting control

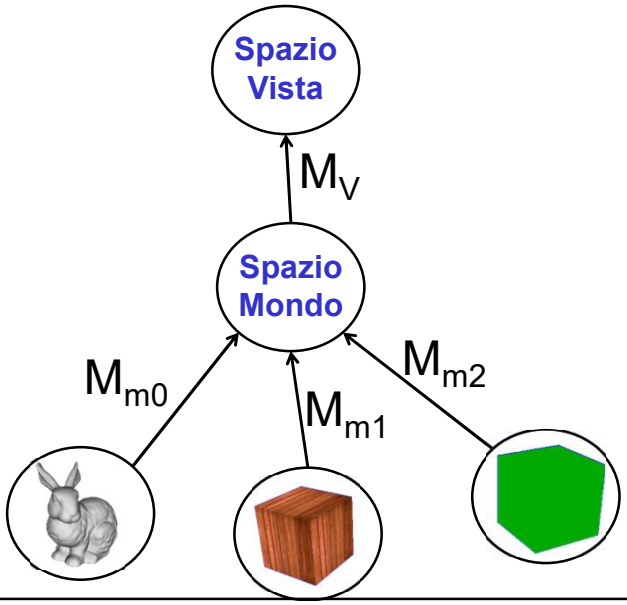
135

Nota sulla matrice di Model-view



136

Ogni oggetto della scena ha la propria matrice di modellazione



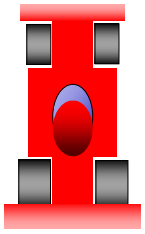
137

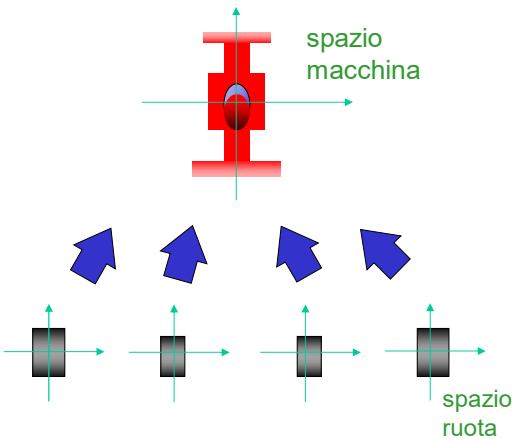
Scene graph a più livelli (albero della scena)

- ✓ Ad ogni nodo è associato uno spazio
  - ⇒ Radice: spazio mondo
  - ⇒ Nodi interni e foglie: spazi oggetto (di un dato oggetto)
  - ⇒ Ad ogni nodo associo le istanze delle mesh associate a quel nodo
- ✓ Ad ogni nodo N, associo la matrice «locale» che porta al padre di N
- ✓ La matrice di modellazione di N si ottiene cumulando trasformazioni dal nodo N alla radice
  - ⇒ Nota: le matrici vengono cumulate dal basso verso l'alto:
  - ⇒ La matrice del nodo più profondo viene eseguita per prima
- ✓ Lo scenegraph può avere qualsiasi profondità
  - ⇒ Vediamo un esempio con due livelli sotto la radice

138

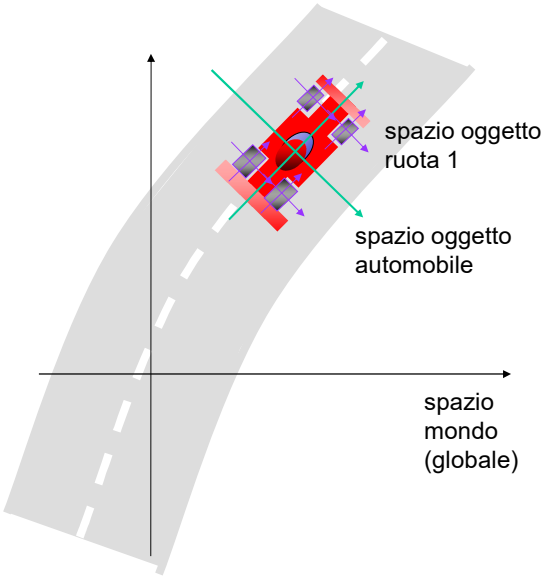
Scene composite (gerarchicamente)





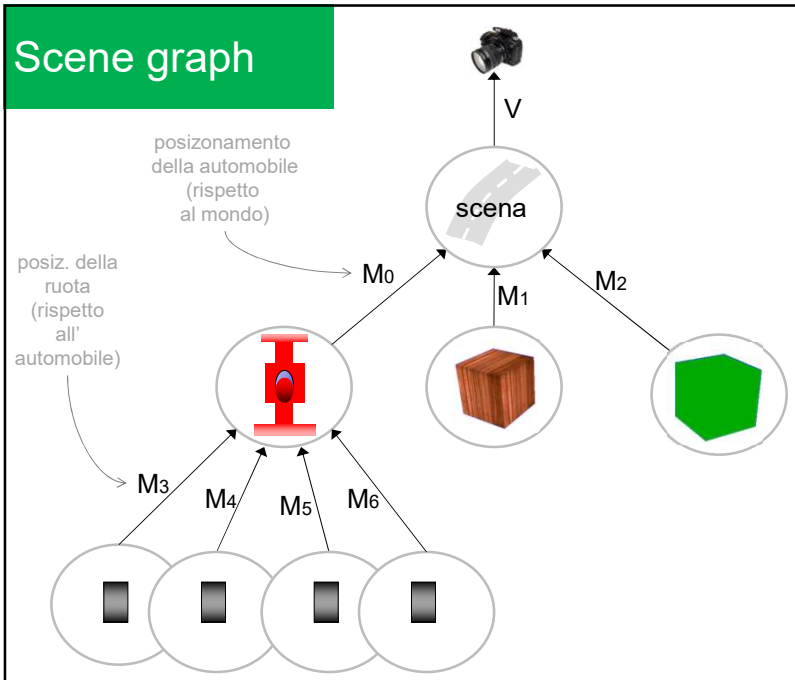
139

Rendering di scene composite

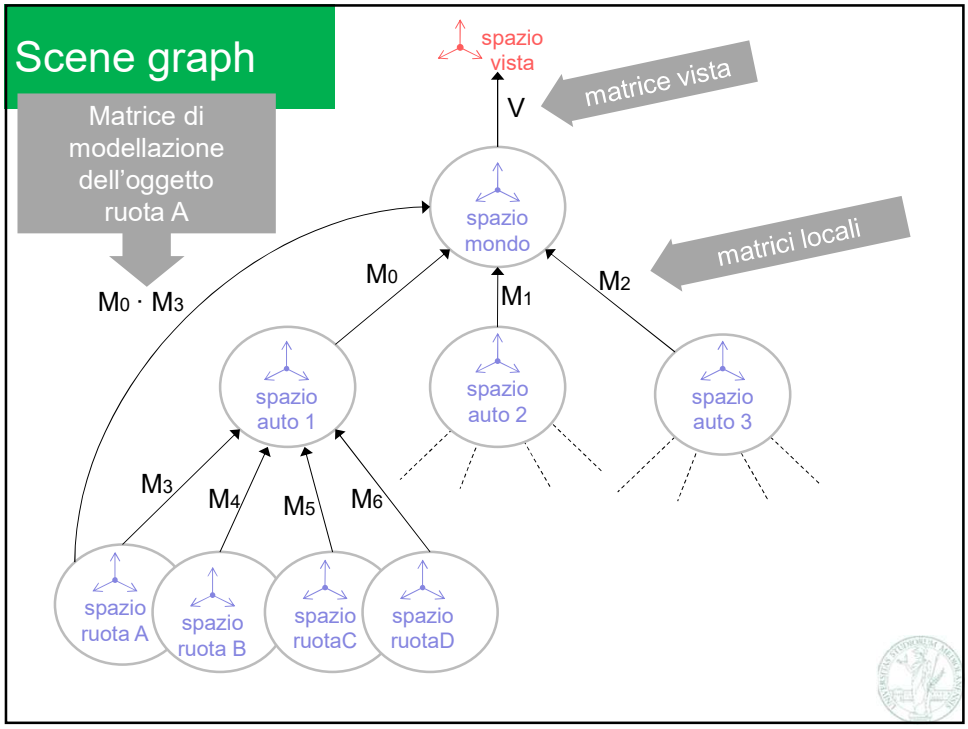


140

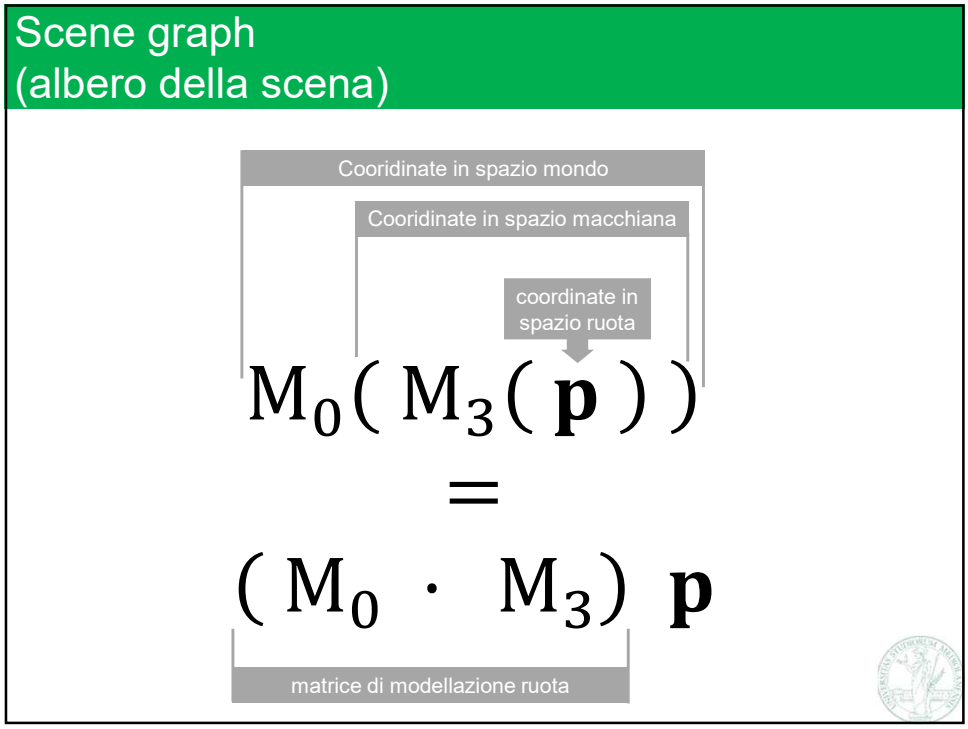
Scene graph



141


Scene graph
Matrice di modellazione dell'oggetto ruota A
M0 · M3
spazio vista
V
matrice vista
spazio mondo
matrici locali
M0
M1
M2
spazio auto 1
spazio auto 2
spazio auto 3
M3
M4
M5
M6
spazio ruota A
spazio ruota B
spazio ruotaC
spazio ruotaD

142


Scene graph
(albero della scena)
Cooridinate in spazio mondo
Cooridinate in spazio macchiana
coordinate in spazio ruota
$M_0(M_3(\mathbf{p}))$
=
$(M_0 \cdot M_3)\,\mathbf{p}$
matrice di modellazione ruota

143

Scene graph a più livelli (albero della scena)

- ✓ Vantaggio1:
  - ⇒ Modificando la matrice di modellazione della macchina (M_0), viene modificata anche in modo corretta anche quella della ruota ($M_0 \cdot M_3$)
  - ⇒ Cioè: spostando (traslando) la macchina, le ruote la seguono!
  - ⇒ Idem per qualsiasi altra trasformazione affine (scalatura, rotaizone, etc)
- ✓ Vantaggio 2:
  - ⇒ Le matrici di modellazione locali sono definite in termini del sistema di riferimento del padre, in modo intuitivo
  - ⇒ Per es, è facile determinare in quale posizione (traslazione) debba essere la ruota, rispetto alla macchina

144

In three.js

- ✓ La scena è un oggetto di tipo Scene
- ✓ Ogni nodo della scena (per es una mesh) può essere appeso con un comando «add» alla scena, oppure ad un altro nodo, costruendo la scena gerarchica
- ✓ La matrice locale è memorizzata nel campo «matrix» di ogni nodo
  - ⇒ È possibile settala determinando i campi
    position (traslazione)
    scale (scalatura)
    rotation (rotazione)
- ✓ La matrice di modellazione è memorizzata nel campo «worldMatrix» di ogni nodo
  - ⇒ Viene automaticamente aggiornata (di default)

145