

# Curve e Superfici per il Design

## Laboratorio - 1

Prof. Nicola Parolini

17 Ottobre 2019

## Il FranzPlot in breve

FranzPlot nasce come strumento software specificamente come supporto per la didattica di questo corso.

- ▶ Costruisce e renderizza mesh, superfici e curve parametriche.
- ▶ La creazione e la trasformazione degli oggetti rappresentati avviene assemblando grafi composti da nodi.
- ▶ Ogni elemento del grafo ha una funzione specifica, il suo output viene usato come input negli altri nodi.
- ▶ FranzPlot è disponibile come eseguibile per Windows (8, 10) e OS X (10.11+) su Beep.
- ▶ É disponibile un eseguibile a parte compatibile con Windows 7.
- ▶ Descrizione dettagliata di tutte le funzionalità nella guida utente (in Inglese).

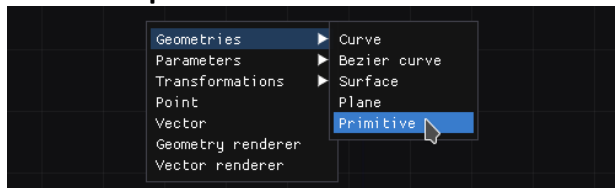
# Come avviare FranzPlot

- ▶ Una volta scaricato l'eseguibile `franzplot.exe` spostatelo nella cartella dove si intende lavorare (possibilmente sul disco locale - nei computer del laboratorio informatico è usualmente indicato con Z:). È sufficiente fare doppio click per avviare il programma, non è necessaria installazione.
- ▶ Il sistema operativo potrebbe chiedere conferma prima di aprire un qualsiasi software scaricato da internet. In tal caso è necessario sbloccare l'eseguibile prima di avviarlo (vedi primo paragrafo della guida per le istruzioni).

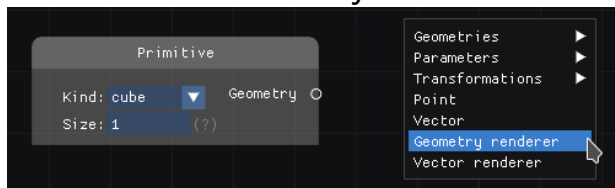
# Interfaccia utente

All'avvio viene mostrato il Node Graph Editor. Per creare un nodo fare **click destro** e scegliere un nodo dal menù. Esempio:

- ▶ Creare una **primitiva**



- ▶ Creare un nodo di **Geometry Renderer**

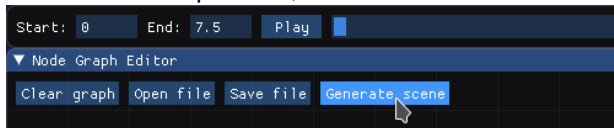


# Interfaccia utente

- Collegare i due nodi



- Dalla barra dei pulsanti, clickare su **Genera Scena**



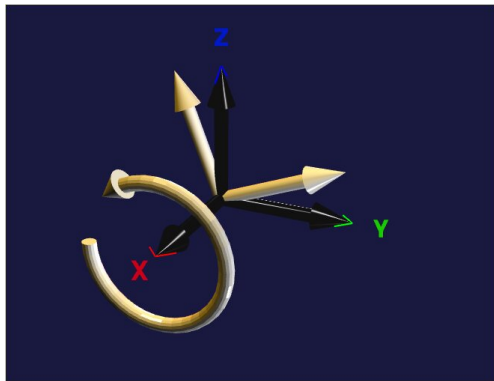
Per zoomare **nell'editor** tenere premuto ctrl e usare la rotella del mouse. Per resettare il livello di zoom clickare usando la rotella.

# Nota sulle trasformazioni

Nelle prossime slide seguirà l'elenco delle trasformazioni usate nell'intero corso, la maggior parte di queste non saranno usate in questa esercitazione.

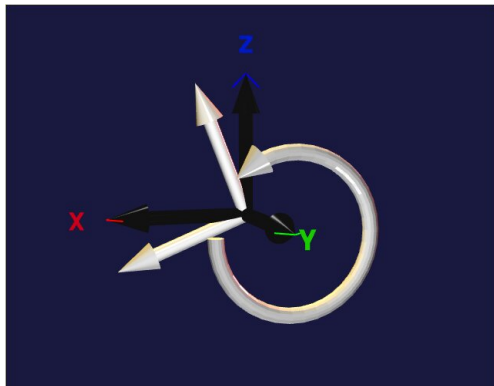
## Rotazioni: Asse x

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (1)$$



## Rotazioni: Asse y

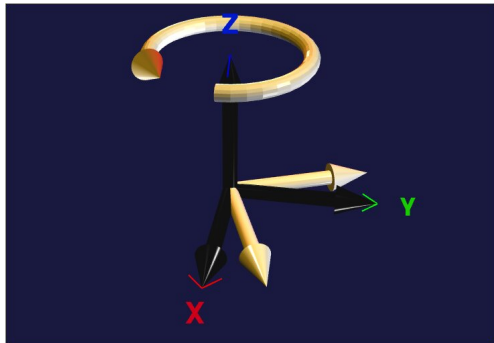
$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (2)$$





## Rotazioni: Asse z

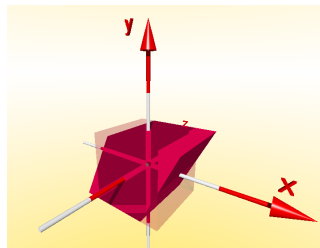
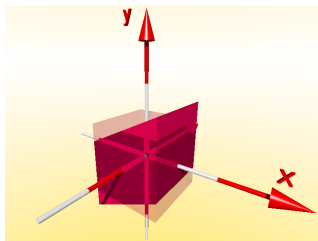
$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$



# Tagli

Taglio in direzione x sulle facce con normale y:

$$T_{xy} = \begin{bmatrix} 1 & k_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$



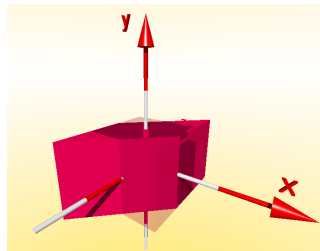
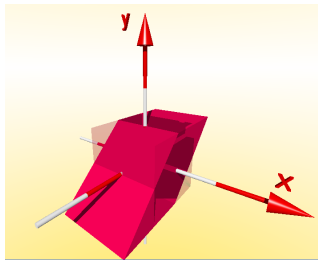
Taglio in direzione y sulle facce con normale x:

$$T_{yx} = \begin{bmatrix} 1 & 0 & 0 \\ k_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

## Tagli[2]

Taglio in direzione z sulle facce  
con normale x:

$$T_{zx} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ k_z & 0 & 1 \end{bmatrix} \quad (6)$$



Taglio in direzione z sulle facce  
con normale y:

$$T_{zy} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & k_z & 1 \end{bmatrix} \quad (7)$$

# Scalatura, Riflessione, Proiezione

## ► Scalatura

$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} \quad (8)$$

## ► Riflessione

$$F = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - 2 \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \begin{bmatrix} n_x & n_y & n_z \end{bmatrix} \quad (9)$$

## ► Proiezione

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \begin{bmatrix} n_x & n_y & n_z \end{bmatrix} \quad (10)$$

# Coordinate omogenee

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & t_1 \\ a_{21} & a_{22} & a_{23} & t_2 \\ a_{31} & a_{32} & a_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11}x + a_{12}y + a_{13}z + t_1 \\ a_{21}x + a_{22}y + a_{23}z + t_2 \\ a_{31}x + a_{32}y + a_{33}z + t_3 \\ 1 \end{bmatrix}$$

# Coordinate omogenee

FranzPlot fa uso di matrici di trasformazione in coordinate omogenee, che sono matrici  $4 \times 4$ .

- Per l'uso odierno, ci basta sapere che quando scriviamo una matrice di trasformazione  $3 \times 3$  dobbiamo inserirla nel blocco in alto a sinistra, lasciando invariato il resto:

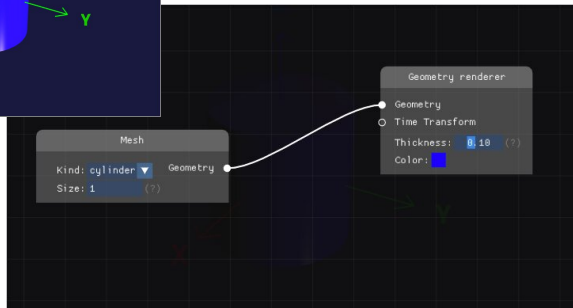
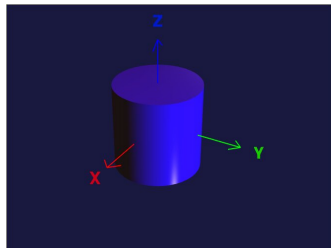
$$T = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \Rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# Esercizio 1: prendere familiarità con FranzPlot

Il nodo **Primitive** ci fornisce una piccola libreria di figure geometriche primitive, figure solide già costruite: sfera, cubo, cilindro, ecc... Vediamo un altro esempio di utilizzo:

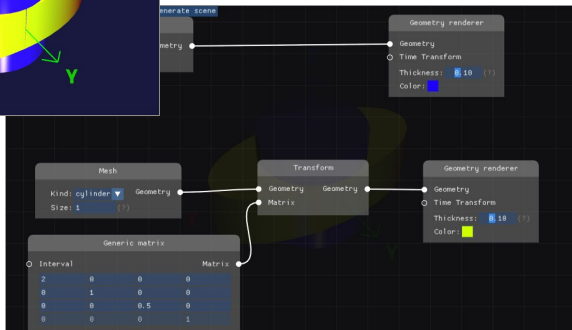
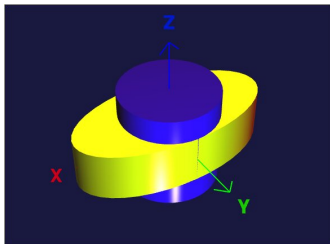
- ▶ Disegnare un cilindro con asse parallelo a  $Z$  ed altezza 1.
  - ▶ *Elementi da utilizzare:*  
Geometries → Primitives,  
Geometry Renderer.
- ▶ Effettuare uno scaling con  $S_x = 2$ ,  $S_y = 1$ ,  $S_z = 0.5$ .
  - ▶ *Elementi aggiuntivi da utilizzare:*  
Transformations → Generic Matrix,  
Transformations → Transform

# Esercizio 1 - i





# Esercizio 1 - ii



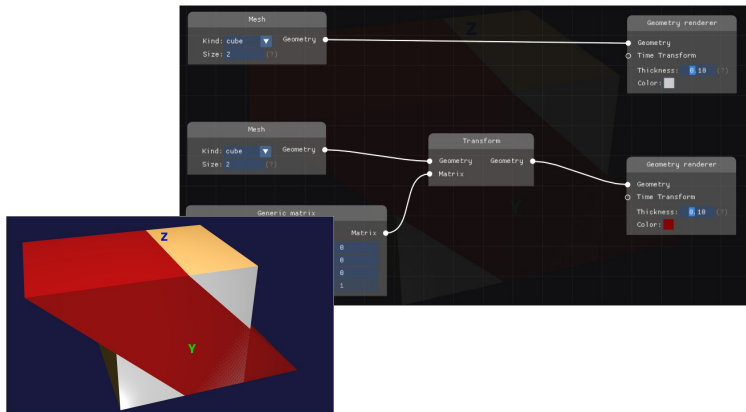
## Esercizio 2: Identificare una trasformazione

- ▶ Disegnare un cubo centrato sull'origine, con lati di misura 2 (notare che nella primitiva è possibile fissare un parametro che influenza la dimensione dell'oggetto).
- ▶ Applicare al cubo la seguente matrice:

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- ▶ Descrivere la deformazione che è stata applicata al cubo. Il volume del cubo è cambiato?

## Esercizio 2 - i

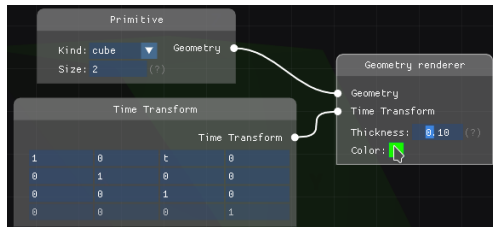


## Esercizio 3: Trasformazioni tempo dipendenti

- ▶ Partiamo nuovamente dal cubo dell'es. 2, questa volta applichiamo una trasformazione nel tempo del tipo:

$$A = \begin{bmatrix} 1 & 0 & t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- ▶ Utilizzare l'elemento Transformations → Time Transform  
*Attenzione: in questa matrice è obbligatorio usare la lettera t*



- ▶ Generare la scena e fare click sul tasto **Play** dalla Top Bar

## Esercizio 4 - Rotazioni e traslazioni

Per applicare una trasformazione di rotazione o traslazione abbiamo due alternative:

- ▶ Scrivere gli elementi nella matrice generica

oppure

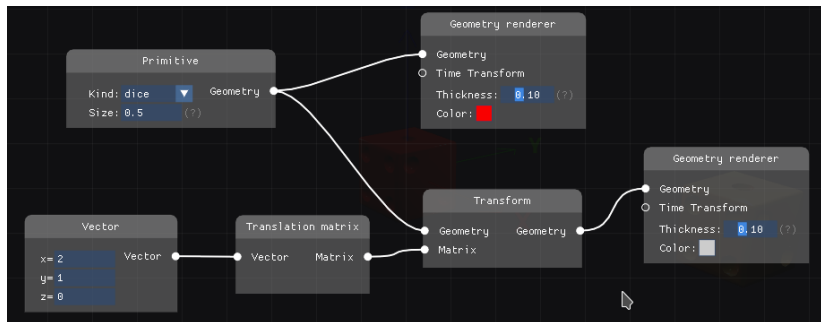
- ▶ Usare due nodi contenenti le trasformazioni “preconfezionate” (utili per fare test o esperimenti più velocemente)

Andiamo a vedere come usare i nodi Translation Matrix e Rotation Matrix

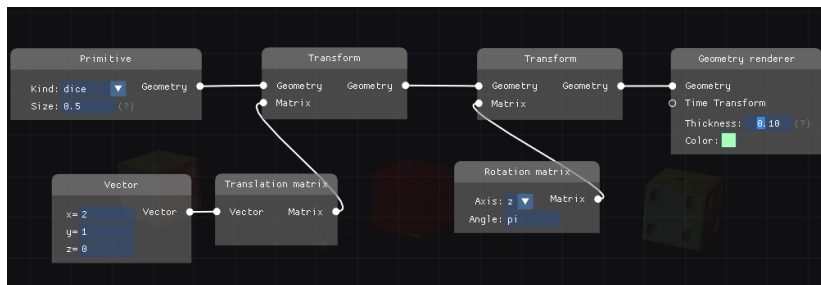
## Esercizio 4 - Rotazioni e traslazioni

- ▶ Creare un oggetto di tipo 'dado' ('*dice*') utilizzando il comando `Primitive`, con fattore di scala 0.5.
- ▶ Traslare il centro dell'oggetto in  $\langle 2, 1, 0 \rangle$  (combinando gli elementi `Transformations` → `Translation Matrix` e `Vector`).
- ▶ Ruotare di 180 gradi il dado traslato intorno all'asse  $Z$  (usando l'elemento `Transformations` → `Rotation Matrix`).
- ▶ Rappresentare i tre oggetti (iniziale, traslato, traslato e ruotato).

## Esercizio 4 - i - Dado traslato

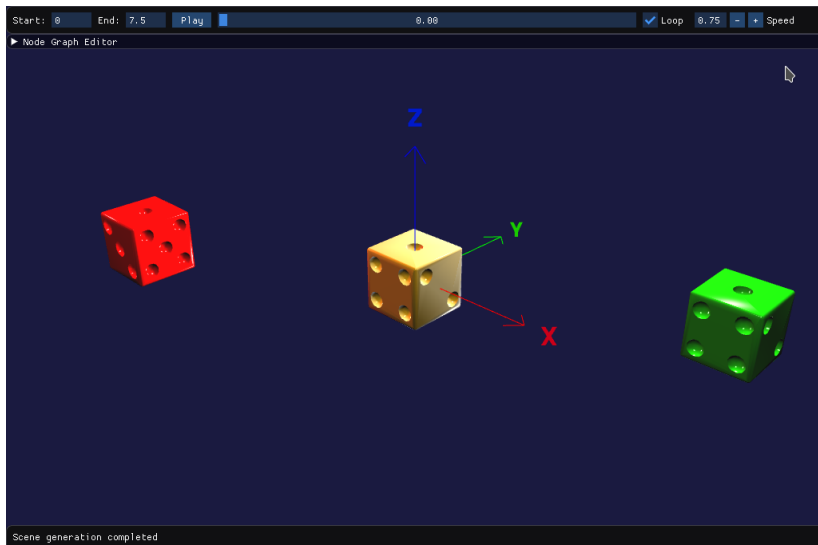


## Esercizio 4 - ii - Trasformazioni in cascata





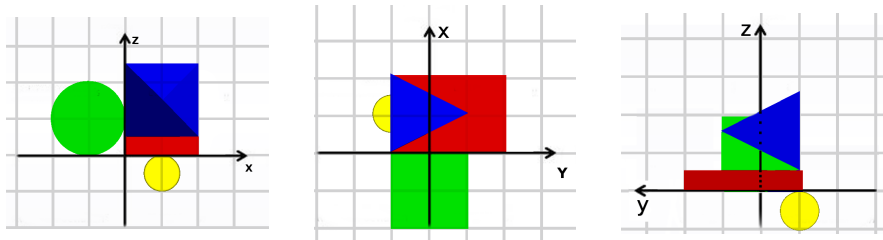
## Esercizio 4 - iii - Risultato



## Esercizio 4 - iv - Considerazioni finali

- ▶ È possibile verificare che il dado sia stato ruotato correttamente controllando come sono orientate le facce.
- ▶ Se vogliamo animare la rotazione usando la Time Transform, sarà necessario scrivere la matrice “a mano” (non è possibile usare il nodo Rotation Matrix).
- ▶ Cambiando l'ordine delle trasformazioni (applicando la rotazione *prima* della traslazione), il risultato finale è lo stesso o cambia?

## Per Casa: Proiezioni ortogonali



Creare un'organizzazione di oggetti le cui proiezioni sui piani cartesiani riproducono le figure sovrastanti.

Per visualizzare la scena da diverse angolazioni predefinite è possibile premere i tasti da 1 a 7. In questo caso specifico possono essere utili rispettivamente le view 6, 7 e 5.