

# GÉOMÉTRIE ALGORITHMIQUE

---

TRAVAUX PRATIQUES I

# TP PROJECTION 3D

---

PROJETER UN MODEL 3D SUR UN ECRAN 2D

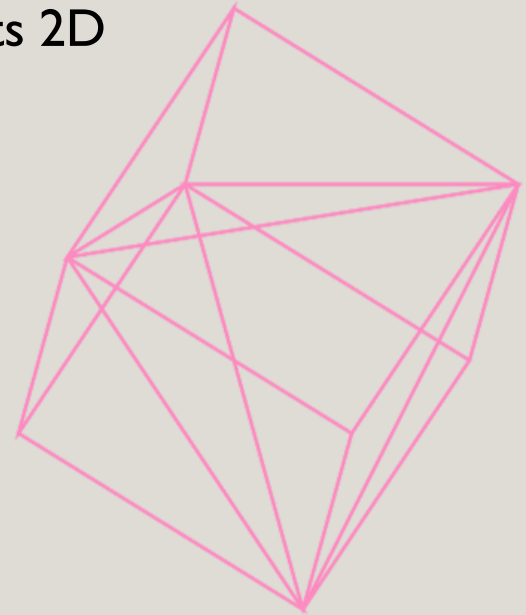


# OBJECTIF

---

- L'objectif de ce TP est de dessiner un cube en 3D fil de fer de  $1 \times 1 \times 1$  unité de côté
- Les traits seront dessinés sur un Canvas web avec des segments 2D
- Les matrices monde, vue et projection seront fournies
- Ci-dessous le lien vers le squelette du projet

<https://jsitor.com/LnylOjNMIX>



# DESCRIPTION EXERCICE – CREATION CUBE

---

- Définir une fonction de transformation d'un vecteur par une matrice
- Remplir le tableau **positions** des sommets du cube de manière optimale. Réutiliser les sommets communs aux triangles sur une même face.
- Remplir le tableau **indices** (triangles) avec les index des sommets du 12 triangle du cube.

# DESCRIPTION EXERCICE – PROJECTION

---

- Création d'une boucle de parcours du tableau indices
  - Obtenir les indexes des 3 sommets d'un triangle
  - Obtenir dans le tableau de positions les coordonnées de ces 3 sommets
  - Transformer les coordonnées des sommets avec successivement
    - la matrice monde
    - la matrice vue
    - la matrice projection

# DESCRIPTION EXERCICE – DESSIN 2D

---

- Toujours dans la boucle du tableau indices
  - Logger les coordonnées  $x, y$  projetées à l'étape précédente dans la console avec `console.log()`
  - Adapter les coordonnées  $x, y$  des sommets projetés pour que le cube soit centré dans le Canvas HTML de 700x700
  - Dessiner les 3 segments du triangle dans le Canvas



# BONUS

---

- Implémenter la fonction de multiplication de Matrice
- Multiplier la matrice monde par une matrice de rotation autour de l'axe X pour faire tourner le cube
- Animer le cube en appliquant plusieurs fois une matrice de rotation autour de l'axe X à la matrice monde

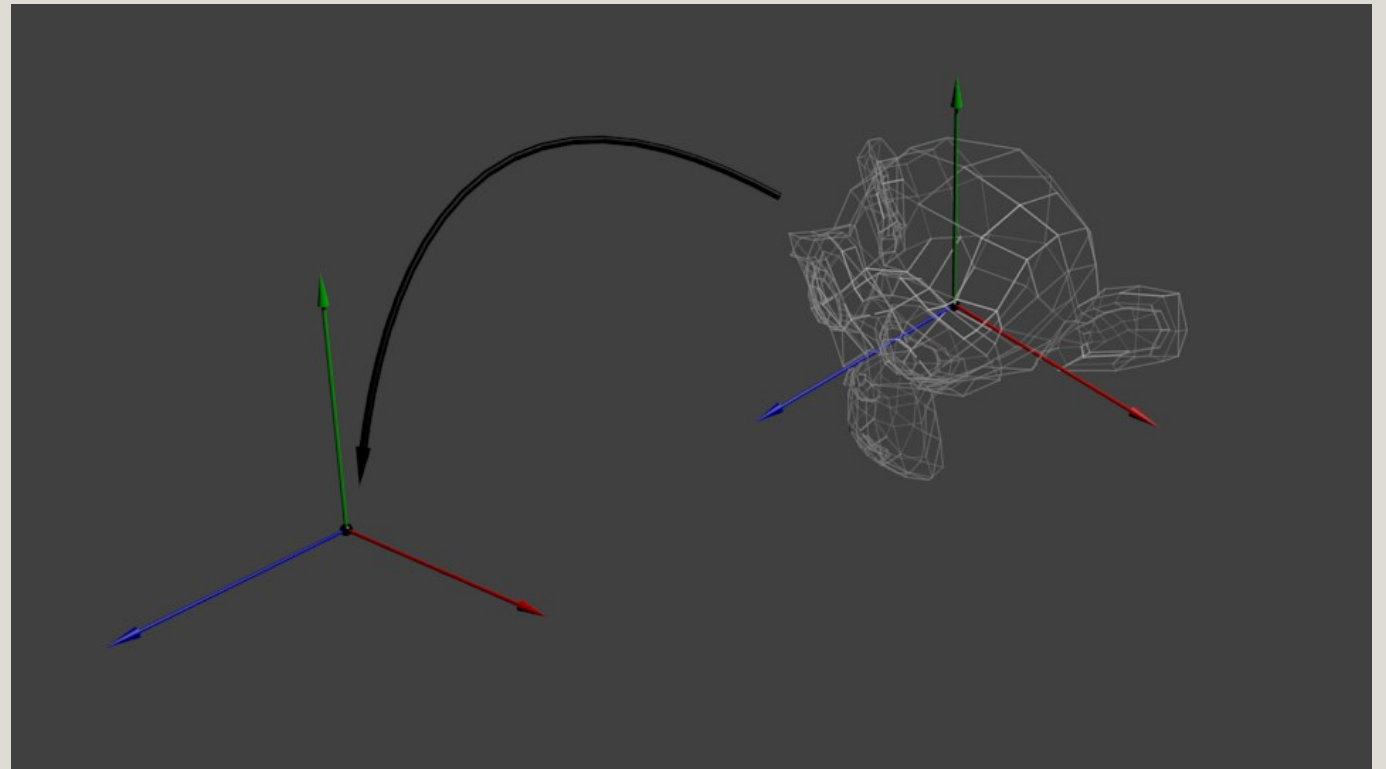
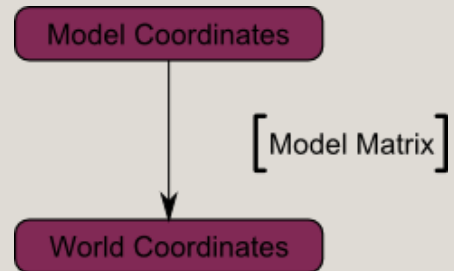
# ANNEXES

---



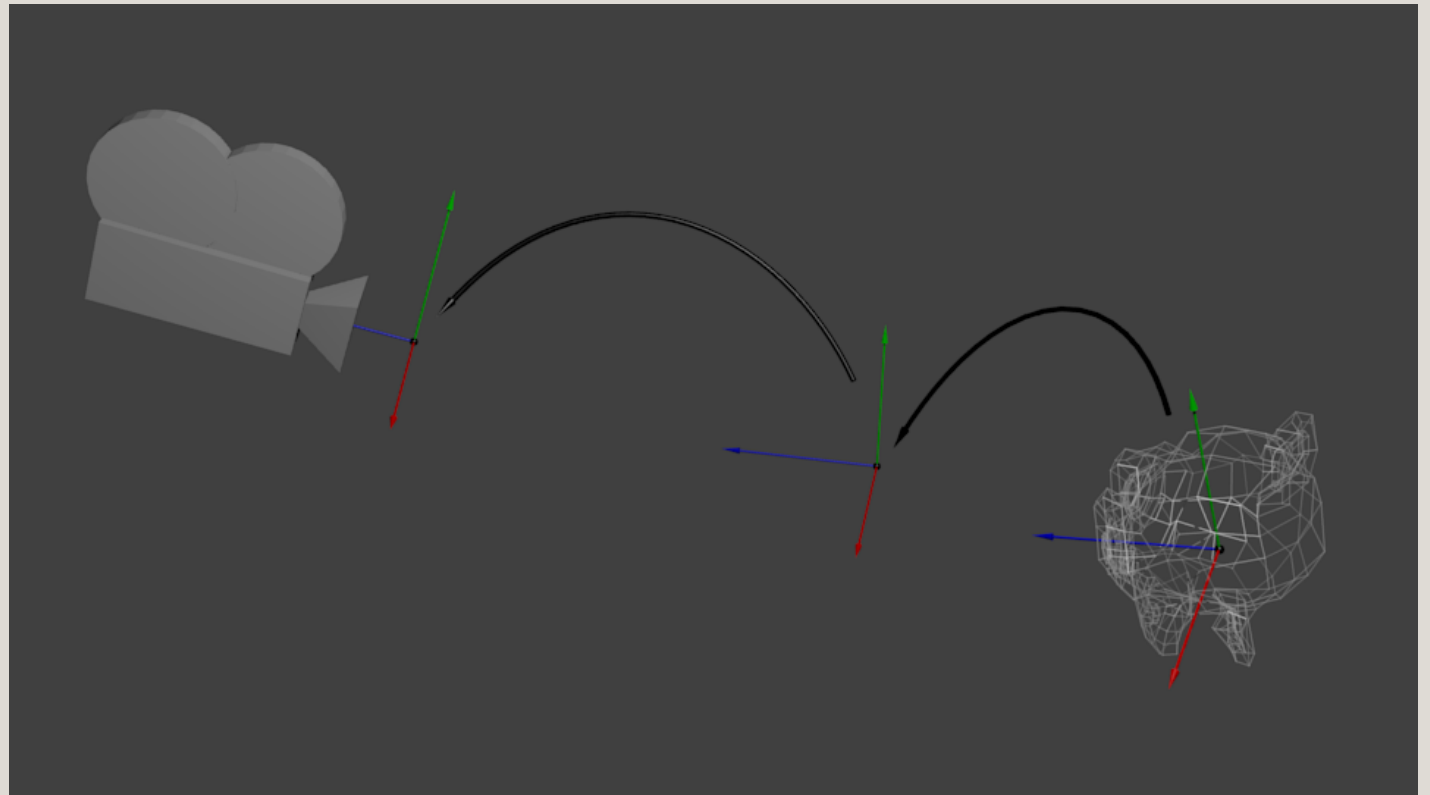
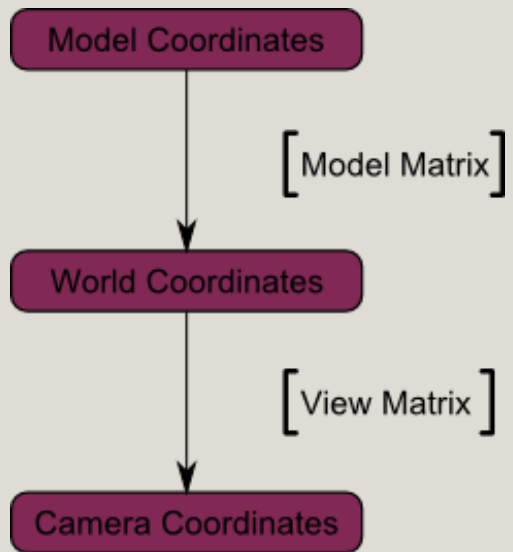
# MATRICE MONDE

---



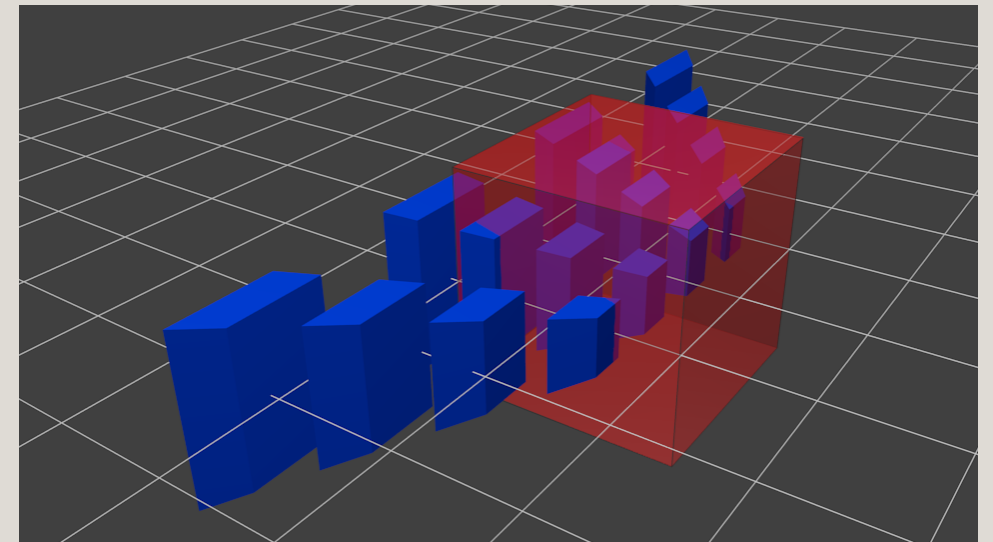
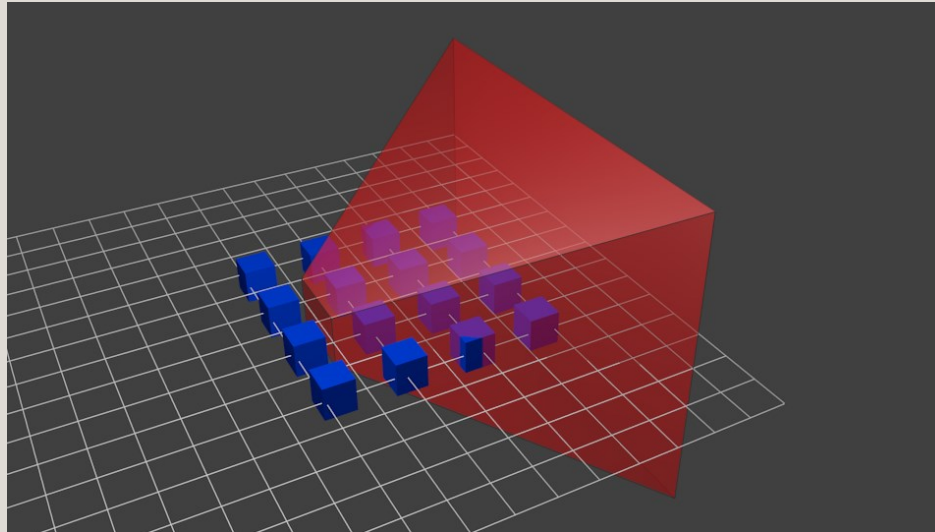
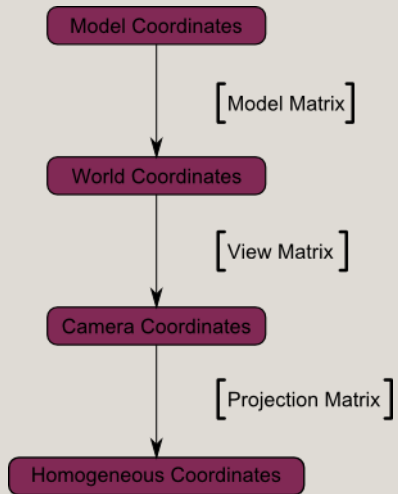
# MATRICE DE VUE

---



# MATRICE DE PROJECTION

---



# MULTIPLICATION MATRICE ET TRANSFORMATION D'UN POINT PAR UNE MATRICE

Transformation d'un vecteur par une matrice

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} ax + by + cz + dw \\ ex + fy + gz + hw \\ ix + jy + kz + lw \\ mx + ny + oz + pw \end{bmatrix}$$

Transformation d'une matrice par une matrice  
Addition l'effet des deux matrices

$$\begin{bmatrix} ax_1 & ax_2 & ax_3 & ax_4 \\ ay_1 & ay_2 & ay_3 & ay_4 \\ az_1 & az_2 & az_3 & az_4 \\ aw_1 & aw_2 & aw_3 & aw_4 \end{bmatrix} \times \begin{bmatrix} bx_1 & bx_2 & bx_3 & bx_4 \\ by_1 & by_2 & by_3 & by_4 \\ bz_1 & bz_2 & bz_3 & bz_4 \\ bw_1 & bw_2 & bw_3 & bw_4 \end{bmatrix} = \begin{bmatrix} ax_1*bx_1+ax_2*by_1+ax_3*bz_1+ax_4*bw_1 & ax_1*bx_2+ax_2*by_2+ax_3*bz_2+ax_4*bw_2 & ax_1*bx_3+ax_2*by_3+ax_3*bz_3+ax_4*bw_3 & ax_1*bx_4+ax_2*by_4+ax_3*bz_4+ax_4*bw_4 \\ ay_1*bx_1+ay_2*by_1+ay_3*bz_1+ay_4*bw_1 & ay_1*bx_2+ay_2*by_2+ay_3*bz_2+ay_4*bw_2 & ay_1*bx_3+ay_2*by_3+ay_3*bz_3+ay_4*bw_3 & ay_1*bx_4+ay_2*by_4+ay_3*bz_4+ay_4*bw_4 \\ az_1*bx_1+az_2*by_1+az_3*bz_1+az_4*bw_1 & az_1*bx_2+az_2*by_2+az_3*bz_2+az_4*bw_2 & az_1*bx_3+az_2*by_3+az_3*bz_3+az_4*bw_3 & az_1*bx_4+az_2*by_4+az_3*bz_4+az_4*bw_4 \\ aw_1*bx_1+aw_2*by_1+aw_3*bz_1+aw_4*bw_1 & aw_1*bx_2+aw_2*by_2+aw_3*bz_2+aw_4*bw_2 & aw_1*bx_3+aw_2*by_3+aw_3*bz_3+aw_4*bw_3 & aw_1*bx_4+aw_2*by_4+aw_3*bz_4+aw_4*bw_4 \end{bmatrix}$$



# ROTATION

---

- Une rotation fait tourner un point autour d'un axe X,Y ou Z
- Matrice 4x4 équivalente:

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

Rotation X

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

Rotation Y

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

Rotation Z