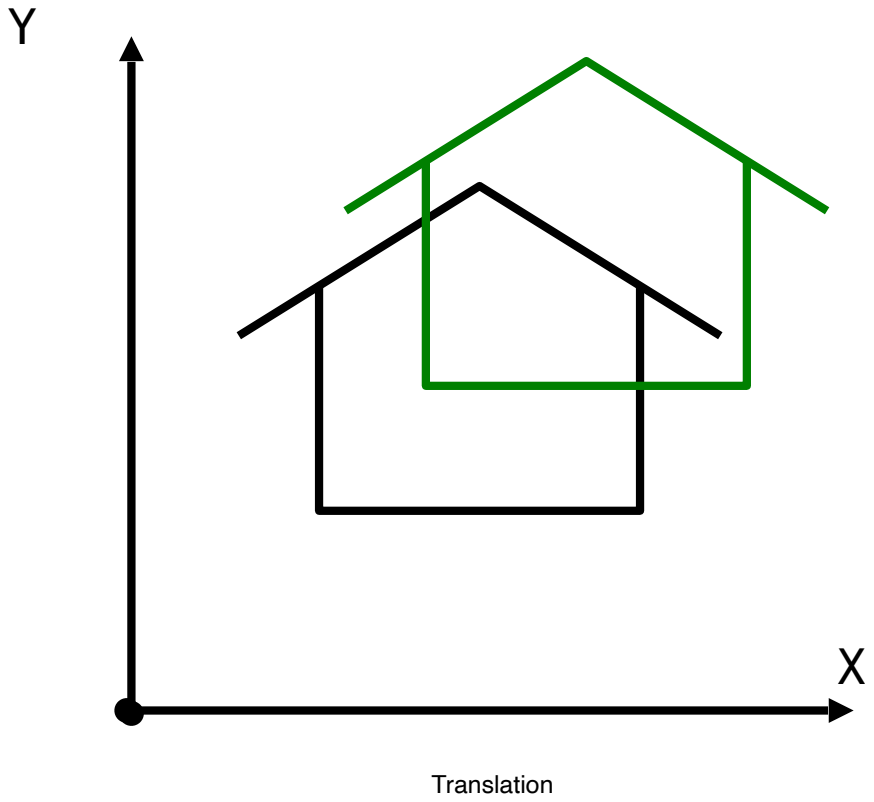
Translation

Vecteur de translation : $t = \begin{pmatrix} x \\ y \\ z \\ 0 \end{pmatrix}$

Matrice : $T = \begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$



gl4dmMatrixTranslate (float x, float y, float z)

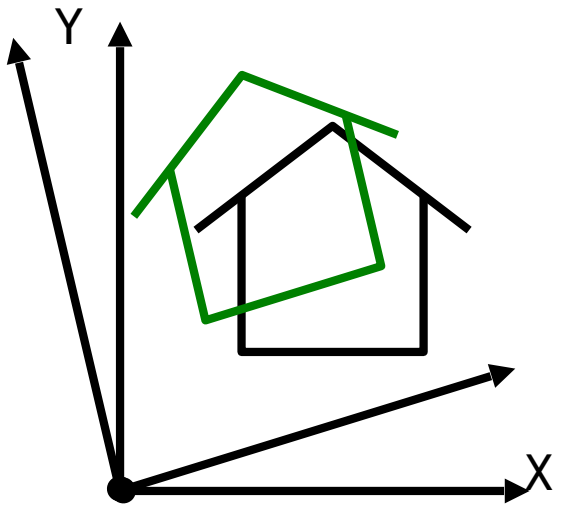
UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

Rotation

Matrice de rotation R d'angle θ autour d'un axe de direction r :

$$r = \begin{pmatrix} x \\ y \\ z \\ 0 \end{pmatrix}, c = cos\theta, s = sin\theta, R = \begin{pmatrix} x^2(1-c)+c & xy(1-c)-zs & xz(1-c)+ys & 0 \\ xy(1-c)+zs & y^2(1-c)+c & yz(1-c)-xs & 0 \\ xz(1-c)-ys & yz(1-c)+xs & z^2(1-c)+c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

gl4dmMatrixRotate (float angle, float x, float y, float z)

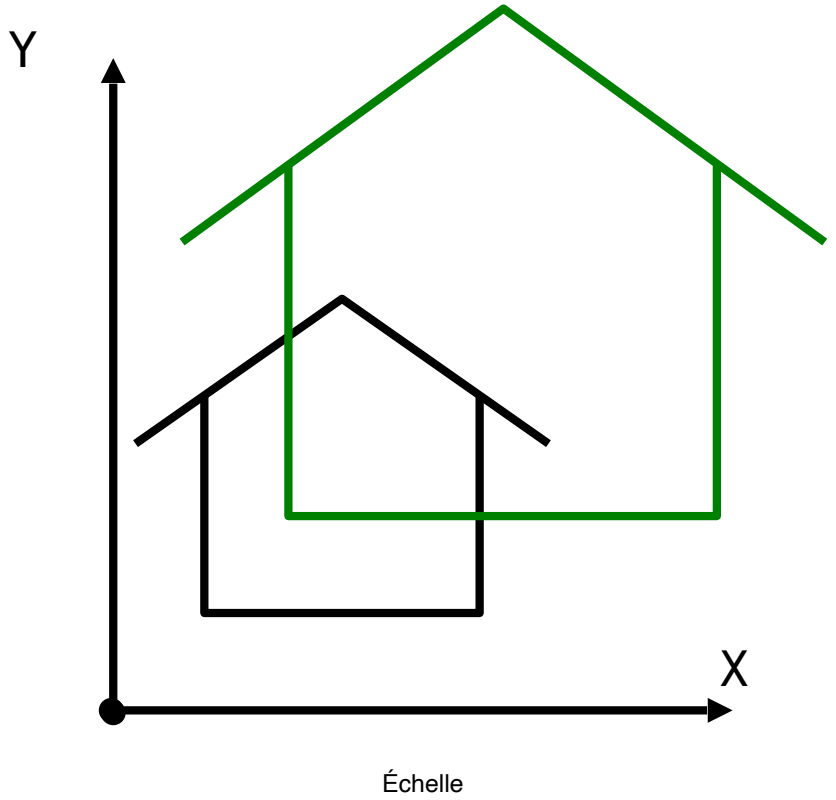


Rotation

Homothéthie

Facteur d ‘échelle $s = \begin{pmatrix} x \\ y \\ z \\ 0 \end{pmatrix}$

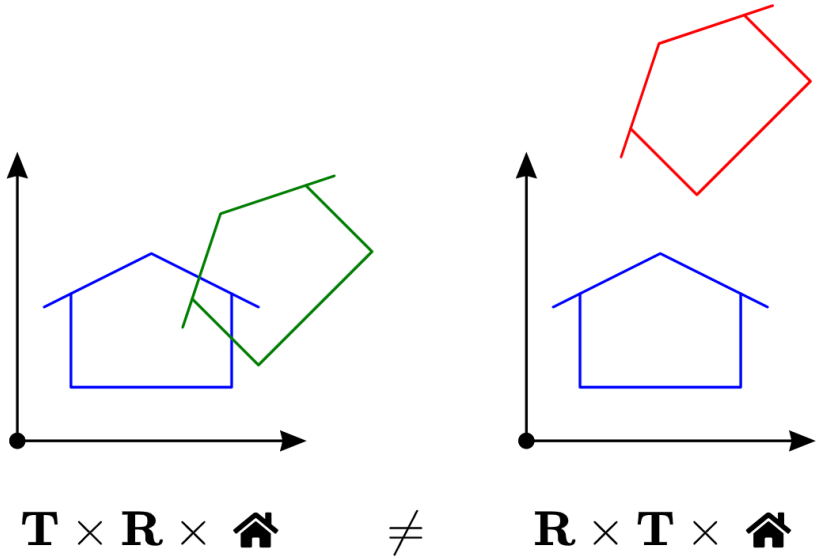
Matrice $S = \begin{pmatrix} x & 0 & 0 & 0 \\ 0 & y & 0 & 0 \\ 0 & 0 & z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$



gl4dmMatrixScale (float x, float y, float z)

Transformation de modélisation

☠ La multiplication n'est pas commutative !



🫖 Convention : T × R × S × 🏠

1. gl4dmMatrixTranslate (T.x, T.y, T.z);
2. gl4dmMatrixRotate (a, R.x, R.y, R.z);
3. gl4dmMatrixScale (S.x, S.y, S.z);

UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

Transformation de vision

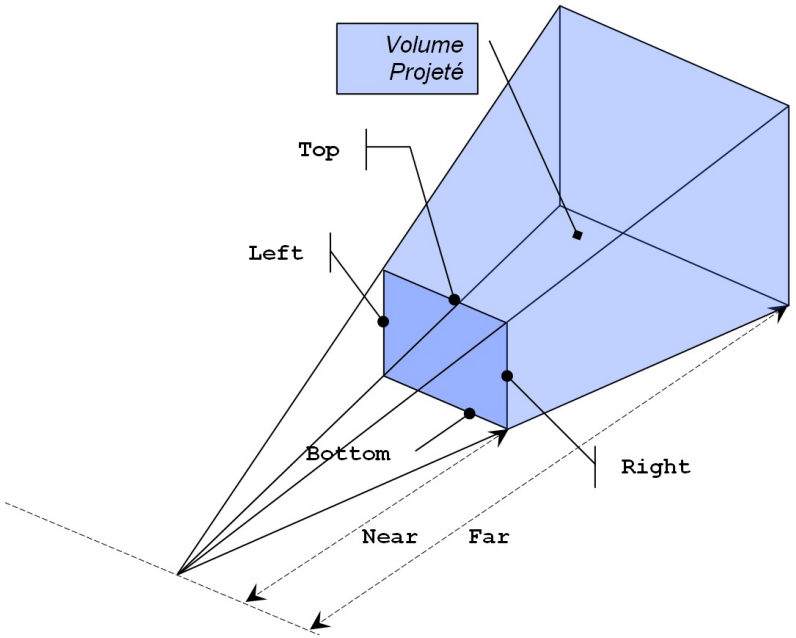
- Position relative de la caméra par rapport aux objets :
  translater la caméra de t équivaut à translater tous les objets de –t

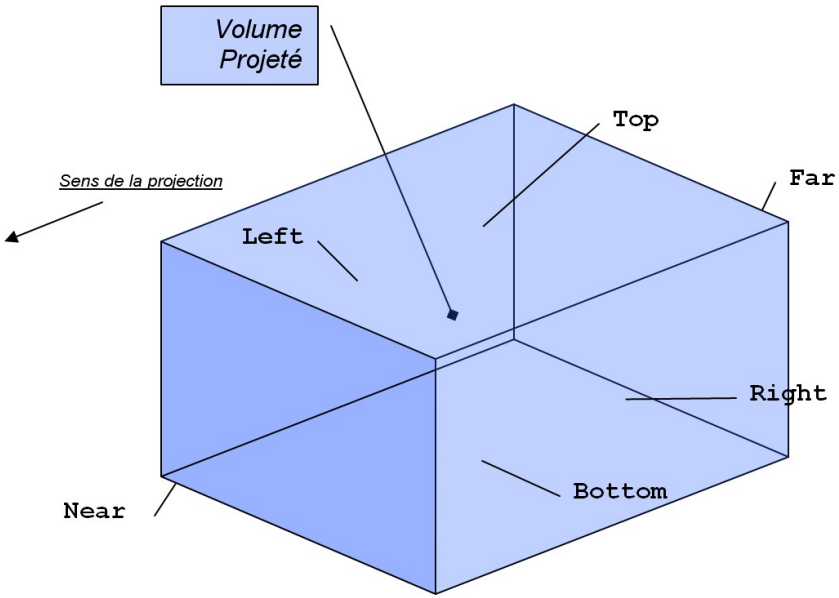
- glViewport (0, 0,w, h) : Surface utilisée lors du dessin, correspond aux coordonnées dans le repère de la zone de dessin.

- gl4dmMatrixLookAt (px,py,pz, cx,cy,cz, ux,uy,uz)
  Positionne la caméra au point [px py pz],
  oriente les Z vers la cible [cx cy cz]
  et les Y selon le vecteur [ux uy uz].

Transformation de projection

- **`gl4dmMatrixPerspective (fovy, aspect, near, far)`**
  Définit le volume de projection perspective.
- **`gl4dmMatrixOrtho(left,right, bottom,top, near,far)`**
  Définit le volume de projection orthogonale.





UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

Projections - suite

- **`gl4duPerspectivef (fovy, aspect, near, far)`**
  Création d'une matrice de projection perspective selon l'ancienne fonction gluPerspective et la multiplie dans la matrice en cours.
- **`gl4duOrthof(left,right, bottom,top, near,far)`**
  Création d'une matrice de projection orthogonale selon l'ancienne fonction glOrtho et la multiplie dans la matrice en cours.
- **`gl4duFrustrumf(left,right, bottom,top, near,far)`**
  Création d'une matrice de projection perspective selon l'ancienne fonction glFrustum et la multiplie dans la matrice en cours

UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

GL4D - Manipulation des matrices

- **`gl4duGenMatrix(type, name)`**
  Génère une matrice 4x4 liée au nom *name* et de type *type*. *Le type peut être GL_FLOAT ou GL_DOUBLE* .
- **`gl4duBindMatrix(name)`**
  Active la matrice liée au nom passé en argument.
- **`gl4duLookAtf(eyeX, eyeY, eyeZ, centerX, centerY,centerZ, upX, upY, upZ)`**
  Définit des transformations pour simuler un point de vue avec direction de regard et orientation.
- **`gl4duLoadIdentityf()`**
  Chargement de la matrice identité dans la matrice actuellement active.
- **`gl4duPushMatrix()`**
  Sauvegarde/empile la matrice courante et utilise une nouvelle matrice dont le contenu est le meme que celle empilée.
- **`gl4duPopMatrix()`**
  Dépile la matrice courante en restaurant l'état précédemment sauvegardé à l'aide de **`gl4duPushMatrix.`**

UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

Objets-géométrie GL4D

- **`gl4dgQuadf ()`**
  Génère un objet-géométrie de type Quad (plan vertical en z=0) et renvoie son identifiant (référence). La géométrie est décrite par 4 sommets reliés par un triangle_strip.
- **`gl4dgSpheref(slices,stacks)`**
  Génère un objet-géométrie de type sphère et renvoie son identifiant (référence).Cette sphère est obtenue par transformation des coordonnées polaires en coordonnées cartésiennes. slices donne le nombre de longitudes de la sphère et stacks le nombre de latitudes de la sphère.
- **`gl4dgCubef()`**
  Génère un objet-géométrie de type cube et renvoie son identifiant (référence). Ce cube est composé de 6 plans meshés par des triangle_strips. Le normales sont aux plans.
- **`gl4dgTorusf(slices, stacks, radius)`**
  Génère un objet-géométrie de type tore et renvoie son identifiant (référence). slices donne le nombre de longitudes du tore, stacks le nombre de « latitudes » (coupes verticales et radius le rayon d'une section verticale du tore.
- Il y en a d'autres !
- **`gl4dgDraw(id)`**
  Dessine un objet-géométrie dont l'identifiant est passé en argumente.
- **`gl4dgDelete(id)`**
  Détruit un objet-géométrie dont l'identifiant (référence) est passé en argument.