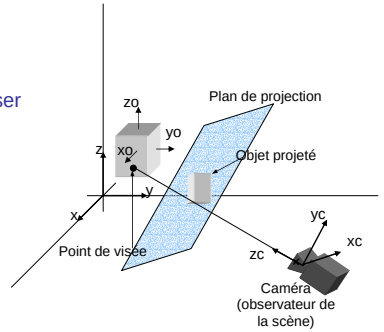
4. Transformations géométriques

- plusieurs repères :
  - objet,
  - scène,
  - observateur(caméra),
  - écran:

  transformations pour passer d'un repère à l'autre.

- description 3D pour un affichage 2D:

  projection de la scène sur l 'écran



4.1 Les transformations 2D

Soient $\mathbf{X} = \begin{pmatrix} x \\ y \end{pmatrix}$ point de départ et $\mathbf{X}' = \begin{pmatrix} x' \\ y' \end{pmatrix}$ le point final (transformé de $\mathbf{X}$)

Toutes les transformations ponctuelles dans le plan peuvent s'écrire :

$$\mathbf{X}' = \mathbf{AX} + \mathbf{B}$$

$\mathbf{A}$ est une matrice 2×2 inversible $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ $\mathbf{B}$ est un vecteur $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$

$$\begin{cases} x' = a_{11}x + a_{12}y + b_1 \\ y' = a_{21}x + a_{22}y + b_2 \end{cases}$$

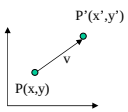
Les transformations 2D

- **Translation**

$\mathbf{A}$ est la matrice identité $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

$\mathbf{B}$ est le vecteur de translation $\mathbf{T}_v = \begin{pmatrix} x_v \\ y_v \end{pmatrix}$

$$\begin{cases} x' = x + x_v \\ y' = y + y_v \end{cases}$$



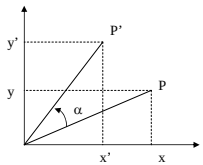
Les transformations 2D

- **Changement d'échelle**

$\mathbf{A}$ est une matrice diagonale $\mathbf{E}_{ex,ey} = \begin{pmatrix} e_x & 0 \\ 0 & e_y \end{pmatrix}$ $\mathbf{B}$ est nul

$$\begin{cases} x' = e_x x \\ y' = e_y y \end{cases}$$

- **Rotation par rapport à l'origine**



$\mathbf{A}$ est la matrice de rotation $\mathbf{R}_\alpha = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$ $\mathbf{B}$ est nul $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$$\begin{cases} x' = x\cos\alpha - y\sin\alpha \\ y' = x\sin\alpha + y\cos\alpha \end{cases}$$

Les transformations 2D

- **Symétrie par rapport à un axe**

A est la matrice symétrie par rapport à l'axe des x $\mathbf{S}_x = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ **B** est nul

$$\begin{cases} x' = x \\ y' = -y \end{cases}$$

A est la matrice symétrie par rapport à l'axe des y $\mathbf{S}_y = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$ **B** est nul

$$\begin{cases} x' = -x \\ y' = y \end{cases}$$

- **Application aux objets**

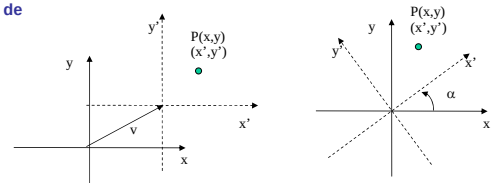
en théorie : on applique la transformation ponctuelle en chaque point de l'objet
en pratique : seulement quelques points de référence

Transformations inverses

- **Transformations inverses**

Translation : $\mathbf{T}_v^{-1} = \mathbf{T}_{-v}$
Rotation : $\mathbf{R}_\theta^{-1} = \mathbf{R}_{-\theta}$
Changement d'échelle : $\mathbf{E}_{e_x,e_y}^{-1} = \mathbf{E}_{1/e_x,1/e_y}$
Symétrie : $\mathbf{S}_x^{-1} = \mathbf{S}_x$ et $\mathbf{S}_y^{-1} = \mathbf{S}_y$

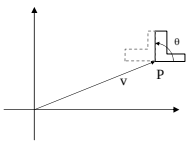
- **Transformations de coordonnées**



$$\begin{cases} x' = x - x_v \\ y' = y - y_v \end{cases} \qquad \begin{cases} x' = x\cos\alpha + y\sin\alpha \\ y' = -x\sin\alpha + y\cos\alpha \end{cases}$$

Composition des transformations

- Toute transformation peut se décomposer en composition de transformations élémentaires



- Comment exprimer de manière simple une transformation non élémentaire?

Exemple : la rotation par rapport à un point P

Transformation : translation vers l'origine $\mathbf{T}_{-v}$
rotation : $\mathbf{R}_\theta$
translation vers le point $\mathbf{P}$: $\mathbf{T}_v$

Coordonnées homogènes

Une transformation peut se représenter sous forme d'une seule matrice **M**

$$\mathbf{X}' = \mathbf{MX}$$

C'est possible pour toutes les transformations élémentaires sauf la translation. Pour cela on utilise les coordonnées dites homogènes

- **Définition**

A tout point (x, y) de $\Re^2$, on fait correspondre un triplet (h_1, h_2, s)
si $s = 1$ alors $h_1 = x$ et $h_2 = y$ (normalisées)
si $s = 0$ alors le point se transforme à l'infini
sinon s est un facteur d'échelle : $h_1 = s \cdot x$ et $h_2 = s \cdot y$

une transformation géométrique peut s'écrire

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

A : rotation, échelle, symétrie
B : translation

Transformations en coordonnées homogènes

$$\text{translation}: \begin{bmatrix} 1 & 0 & v_x \\ 0 & 1 & v_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{rotation}: \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{échelle}: \begin{bmatrix} e_x & 0 & 0 \\ 0 & e_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{symétrie/x}: \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{symétrie/y}: \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Représentation matricielle des transformations

Soient $\mathbf{X}$ le point de départ et $\mathbf{X}'$ le point final (transformé de $\mathbf{X}$).
$\mathbf{X}'$ et $\mathbf{X}$ sont des vecteurs colonnes
Les transformations sont représentées par des multiplications matricielles à gauche.

$\mathbf{X}'=\mathbf{AX}$

On trouve une autre notation en infographie

$\mathbf{Y}'$ et $\mathbf{Y}$ sont des vecteurs lignes
Les transformations sont représentées par des multiplications matricielles à doite.
$\mathbf{Y}'=\mathbf{YB}$

équivalence

si $\mathbf{X}=\mathbf{Y}^T$ et $\mathbf{X}'=\mathbf{Y}'^T$ alors $\mathbf{A}=\mathbf{B}^T$

Composition des transformations en coordonnées homogènes

- **Composition de transformations : produit matriciel**
  - **Transformations successives**
    1) $\mathbf{M}_1$
    2) $\mathbf{M}_2$
    3) $\mathbf{M}_3$
  - **Appliquée à n points**

$$\mathbf{P}_1' = \mathbf{M}_3 \cdot \mathbf{M}_2 \cdot \mathbf{M}_1 \cdot \mathbf{P}_1$$
$$\mathbf{P}_2' = \mathbf{M}_3 \cdot \mathbf{M}_2 \cdot \mathbf{M}_1 \cdot \mathbf{P}_2$$
$$\ldots$$
$$\mathbf{P}_n' = \mathbf{M}_3 \cdot \mathbf{M}_2 \cdot \mathbf{M}_1 \cdot \mathbf{P}_n$$

  - **Composition des transformations : calcul d'une matrice**

$$\mathbf{M} = \mathbf{M}_3 \cdot \mathbf{M}_2 \cdot \mathbf{M}_1$$
$$\mathbf{P}_1' = \mathbf{M} \cdot \mathbf{P}_1$$
$$\mathbf{P}_2' = \mathbf{M} \cdot \mathbf{P}_2$$
$$\ldots$$
$$\mathbf{P}_n' = \mathbf{M} \cdot \mathbf{P}_n$$

Composition des transformations en coordonnées homogènes

- **exemple** Transformation $\mathbf{R}_{\theta,P}$:
  1) translation de $\mathbf{P}$ vers l'origine $\mathbf{T}_{-\mathbf{v}}$
  2) rotation de θ autour de l'origine : $\mathbf{R}_\theta$
  3) translation vers le point $\mathbf{P}$: $\mathbf{T}_\mathbf{v}$

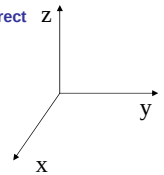
$$\mathbf{R}_{\theta,P} = \mathbf{T}_\mathbf{v} \cdot \mathbf{R}_\theta \cdot \mathbf{T}_{-\mathbf{v}}$$
$$\mathbf{X}' = \mathbf{R}_{\theta,P} \cdot \mathbf{X}$$

- **Opération de prélèvement**

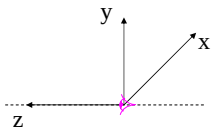
Soit un objet défini dans son propre repère.
Le placer dans une image consiste à :
1) effectuer une mise à l 'échelle
2) effectuer une rotation
3) effectuer une translation

4.2 Les transformations 3D

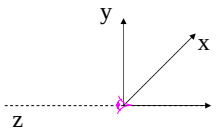
- **Repère direct**



- **Repère indirect lié à l'observateur**

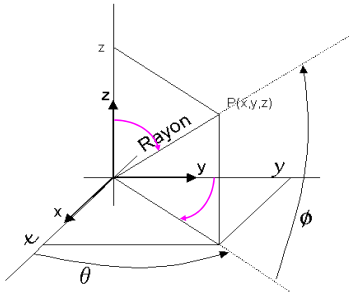


- **Repère direct lié à l'observateur**



Les transformations 3D

- **Coordonnées cartésiennes et sphériques**



Coordonnées cartésiennes (x, y, z)
Coordonnées sphériques (Rayon, θ, ϕ)

Angles azimuth et élévation dans certaines applications (OpenGL)

Les transformations élémentaires

- **Translations**

$$\mathbf{T}_{xt,yt,zt} = \begin{bmatrix} 1 & 0 & 0 & x_t \\ 0 & 1 & 0 & y_t \\ 0 & 0 & 1 & z_t \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **Changements d'échelle**

$$\mathbf{E}_{ex,ey,ez} = \begin{bmatrix} e_x & 0 & 0 & 0 \\ 0 & e_y & 0 & 0 \\ 0 & 0 & e_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **Symétries par rapport à un plan**

$$\mathbf{S}_{xy} : \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{S}_{xz} : \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{S}_{yz} : \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **Rotations par rapport à un axe**

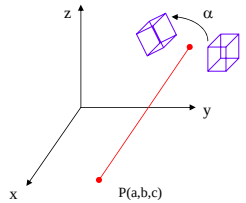
$$\mathbf{R}_{x,\alpha} : \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_{y,\alpha} : \begin{bmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_{z,\alpha} : \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Composition des transformations

- **<u>Principe</u>: le même qu'en 2D; on multiplie les matrices représentant les transformations élémentaires.**
- **Exemple: Rotation autour d'un axe // à l 'axe x.**



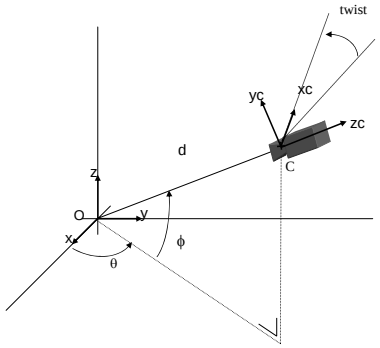
- **Matrice de transformation :**

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -a \\ 0 & 1 & 0 & -b \\ 0 & 0 & 1 & -c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & -b\cos\alpha + c\sin\alpha + b \\ 0 & \sin\alpha & \cos\alpha & -b\sin\alpha - c\cos\alpha + c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

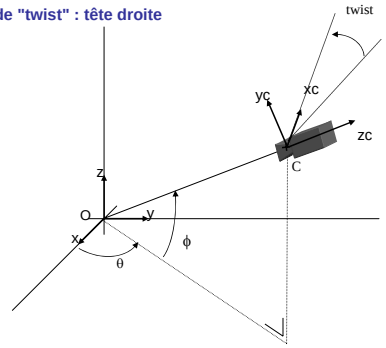
4.4 Transformations de coordonnées

- Opération de changement de répère
- Appliquée lorsqu'on passe du repère de la scène au repère observateur
- Un exemple : le polarview



Transformations de coordonnées

- **On suppose ici que l'observateur vise le centre de la scène** (polarview)
  - **position de l'observateur**
  - **point de visée : centre du repère scène**
  - **angle de "twist" : tête droite**



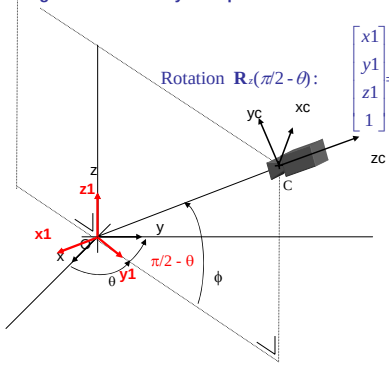
Le polarview (1)

- **Rotation de - (π/2 - θ) autour de z**

Alignement de l'axe y du repère observateur dans le plan OCz

Rotation $\mathbf{R}_z(\pi/2 - \theta)$:

$$\begin{bmatrix} x1 \\ y1 \\ z1 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\pi}{2}-\theta\right) & -\sin\left(\frac{\pi}{2}-\theta\right) & 0 & 0 \\ \sin\left(\frac{\pi}{2}-\theta\right) & \cos\left(\frac{\pi}{2}-\theta\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

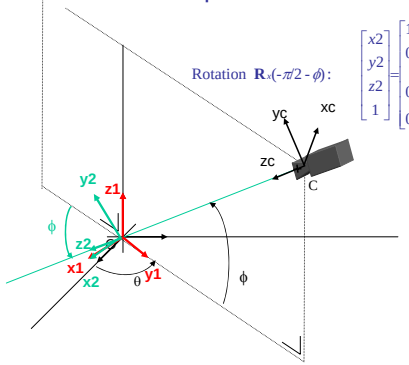


Le polarview (2)

- **Rotation de (π/2 + ϕ) autour de x1**
- **Pointe l'axe z du repère observateur vers le centre du repère scène**

Rotation $\mathbf{R}_x(-\pi/2 - \phi)$:

$$\begin{bmatrix} x2 \\ y2 \\ z2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\left(\frac{\pi}{2}-\phi\right) & -\sin\left(\frac{\pi}{2}-\phi\right) & 0 \\ 0 & \sin\left(\frac{\pi}{2}-\phi\right) & \cos\left(\frac{\pi}{2}-\phi\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x1 \\ y1 \\ z1 \\ 1 \end{bmatrix}$$

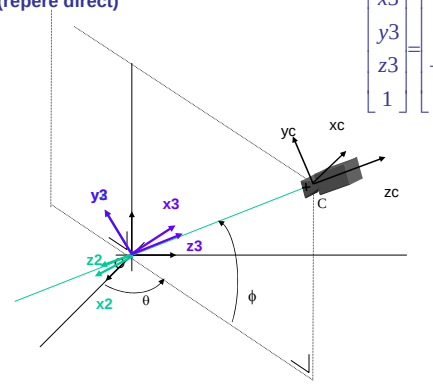


Le polarview (3)

- **Rotation de π autour de y2 (repère direct)**

Rotation $\mathbf{R}_y(\pi)$:

$$\begin{bmatrix} x3 \\ y3 \\ z3 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\pi & 0 & \sin\pi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\pi & 0 & \cos\pi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x2 \\ y2 \\ z2 \\ 1 \end{bmatrix}$$

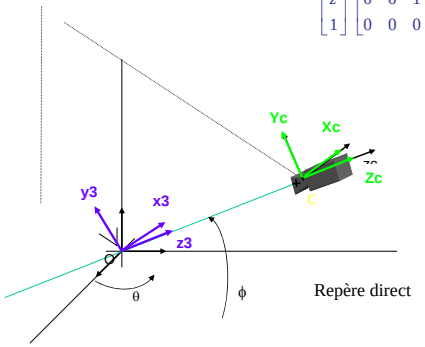


Le polarview (4)

- **Translation de d = distance de O vers C :**

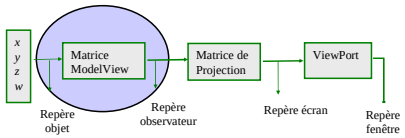
translation T(0,0,-d) :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -\sqrt{x_c^2+y_c^2+z_c^2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x3 \\ y3 \\ z3 \\ 1 \end{bmatrix}$$



4.5 Les transformations avec OpenGL

- **Une seul matrice : ModelView**



Affiche le point **X' = ModelView X**

- **Équivaut à un seul repère : repère observateur**
- **Les transformations de déplacement de scène et de positionnement de l'observateur sont combinées et stockées dans la même matrice**

Les transformations avec OpenGL

- **Tranformations élémentaires**

  glTranslate*(…);
  glRotate*(…);
  glScale*();
  …

- **Capacité de mémorisation : gestion d'une pile**

  glPushMatrix (…)
  glPopMatrix (…)
  glLoadMatrix (…)
  …

- **Produit de matrice à droite**

Les transformations avec OpenGL

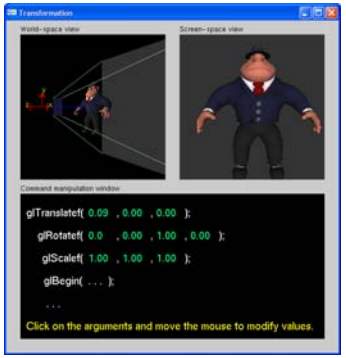
- **Remarque très importante :**

La multiplication des matrices se faisant à droite, il faut faire attention à l'enchaînement des transformations. Par exemple, pour réaliser la transformation composée d'une rotation puis une translation sur P, soit P' = T.R.P, il faut réaliser les appels suivants :

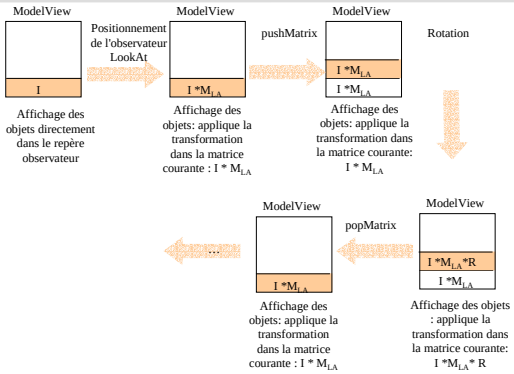
```
glTranslate*(…);
glRotate*(…);
afficheP(…)
```

Les transformations avec OpenGL

tutoriel de Nate Robins : *http://www.xmission.com/~nate/tutors.html*



Les transformations avec OpenGL



4.6 Les projections sur un plan

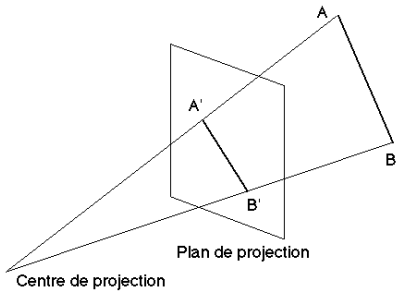
- **Transformation des coordonnées du point de vue de l'observateur dans le plan de l'écran**
- **Classification**
  - **projection non planaire (écran de projection non plan)**
  - **projection planaires**

projections planaires

- *perspectives*
  - *1 point de fuite principal*
  - *2 points de fuite principaux*
  - *3 points de fuites principaux*
- *parallèles*
  - *obliques*
    - *générale*
    - *cavalière*
    - *cabinet*
  - *orthogonales*
    - *dessin industriel*
      - *3 vues*
      - *vue auxiliaire*
      - *vue en coupe*
    - *axonométrique*
      - *isométrique*
      - *dimétrique*
      - *trimétrique*

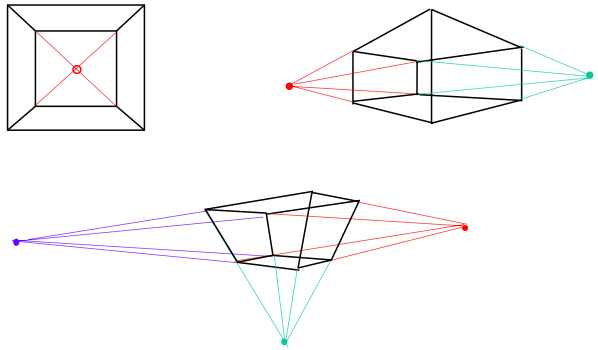
Les projections perspectives

- **Définition du centre de projection**
- **Définition du plan de projection**



Les projections perspectives

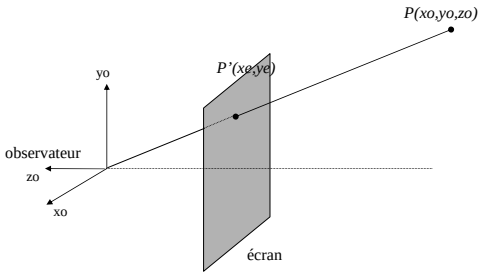
- **Points de fuite : 1, 2 ou 3**



Les projections perspectives

Généralement :

- **centre de projection = position de l 'observateur**
- **plan de projection (perpendiculaire à l'axe de visée de l'observateur : Z ou -Z)**



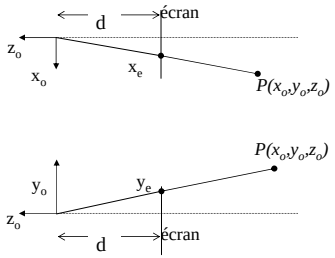
Les projections perspectives

- **Calcul des coordonnées écran**

$$\frac{xe}{d} = \frac{xo}{zo} \qquad \frac{ye}{d} = \frac{yo}{zo}$$

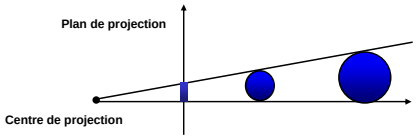
$$xe = \frac{xo}{zo}d \qquad ye = \frac{yo}{zo}d$$

- **Plan de projection est // au plan** $0x_oy_o$

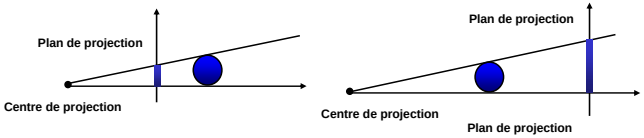


Influence de la position des objets

- **position de l'objet**

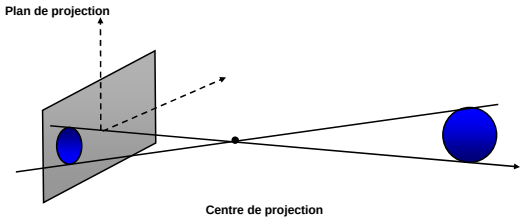


- **position de l'écran**



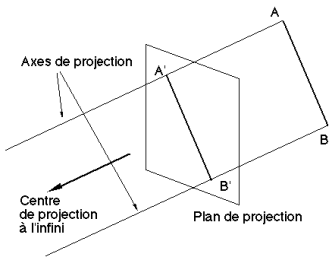
Influence de la position des objets

- **position du centre entre l'objet et le plan de projection**



Les projections parallèles

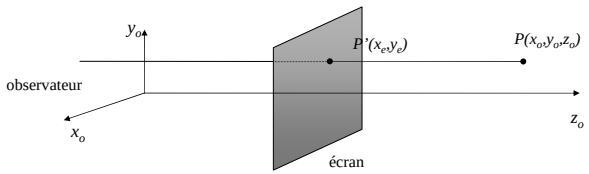
- **plan de projection**
- **centre rejeté à l'infini**



- **orthogonale ou obliques**
  - **orthogonales : axes de projection perpendiculaires au plan de projection.**
  - **obliques : axes de projection font un angle différent de 90° avec le plan de projection**

Les projections parallèles

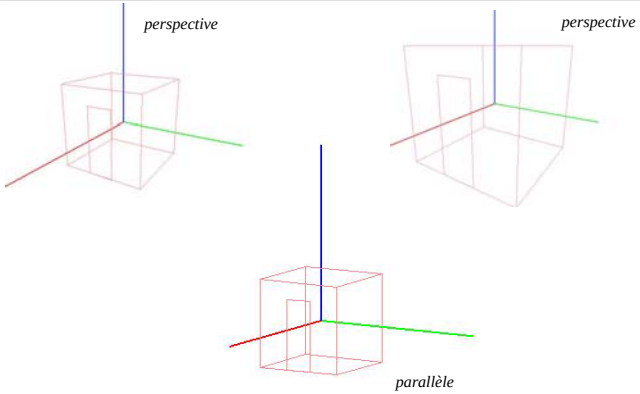
- **plan de projection (perpendiculaire à l'axe de visée de l'observateur : Z ou -Z)**



- **calcul des coordonnées écran**

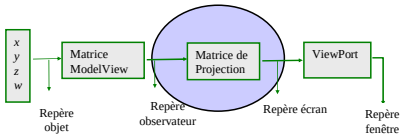
$$xe = xo \qquad ye = yo$$

Exemples



4.7 Les projections avec OpenGL

- **La matrice : GL_PROJECTION**



- **indépendante de la matrice du modèle**
- **initialisée par fonctions OpenGL**

Les projections avec OpenGL

- **Projection perspective**

  glFrustum(…);
  gluPerspective(…);
  …

- **Projection orthogonale**

  glOrtho (…)

Généralement, ces fonctions sont appelées une seule fois.
On ne compose pas les transformations de projection