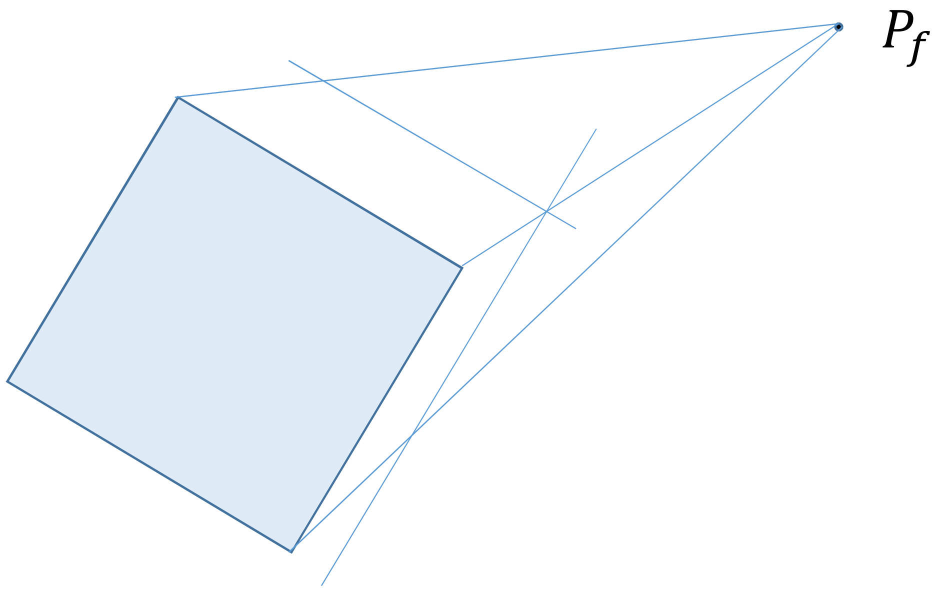
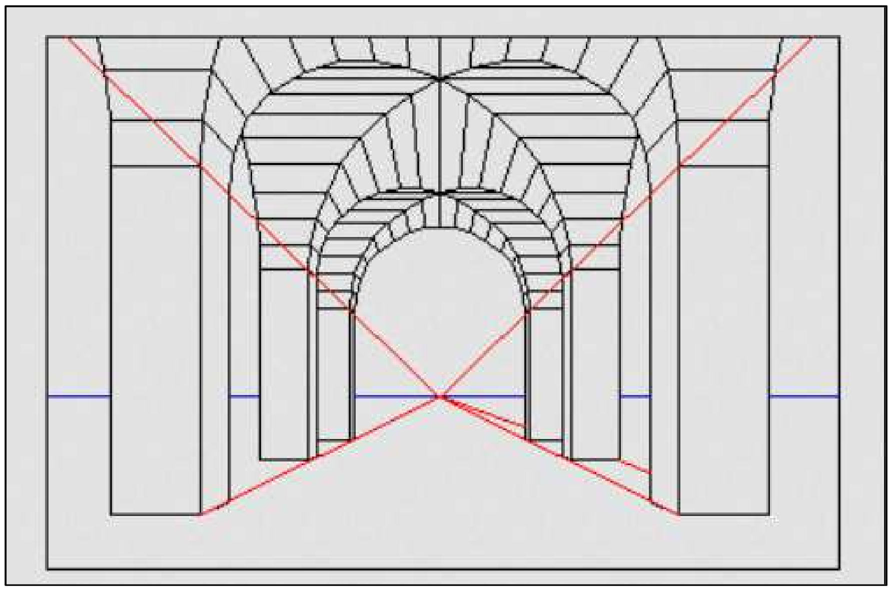
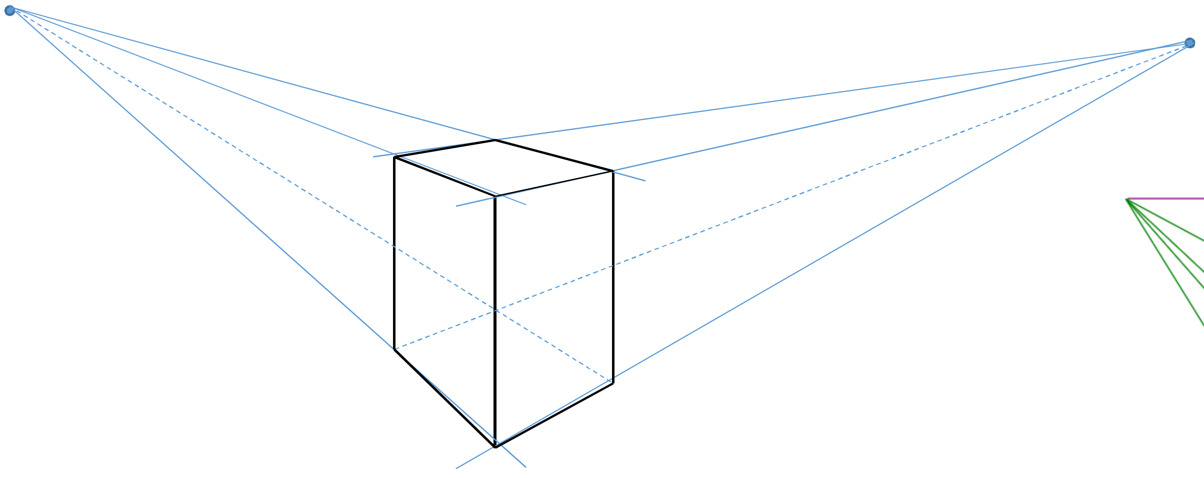
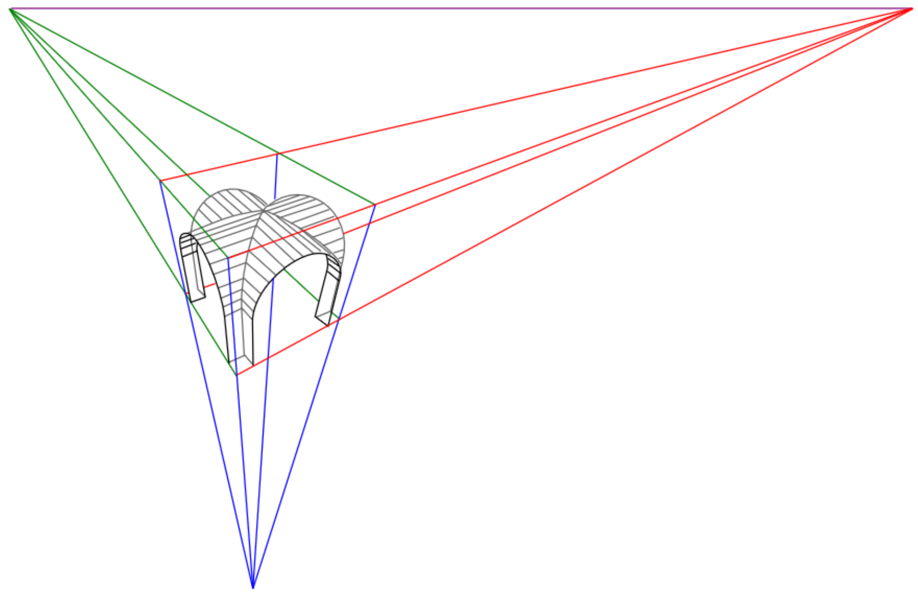
Perspectives à point(s) de fuite





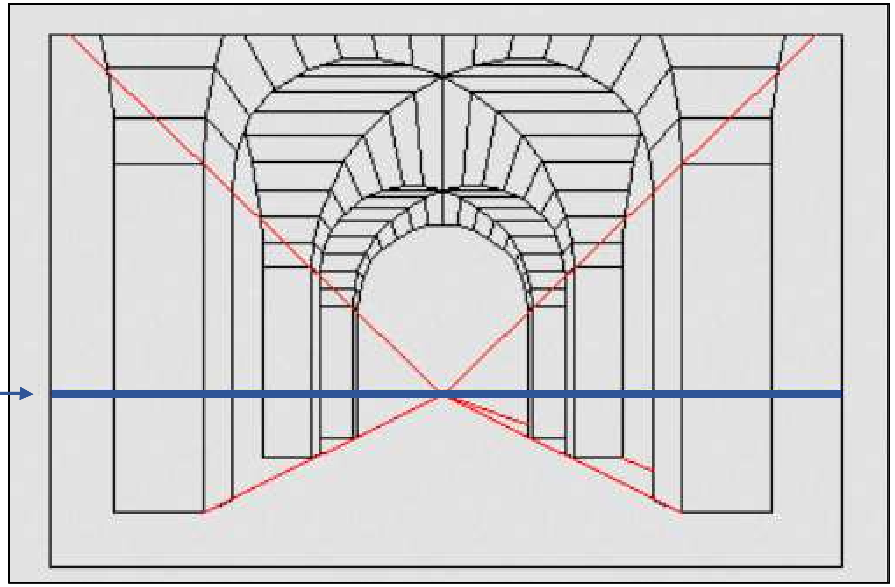
Avec 2 ou 3 points





Perspective conique

1 point de fuite : sur la ligne d'horizon, dans la direction du regard



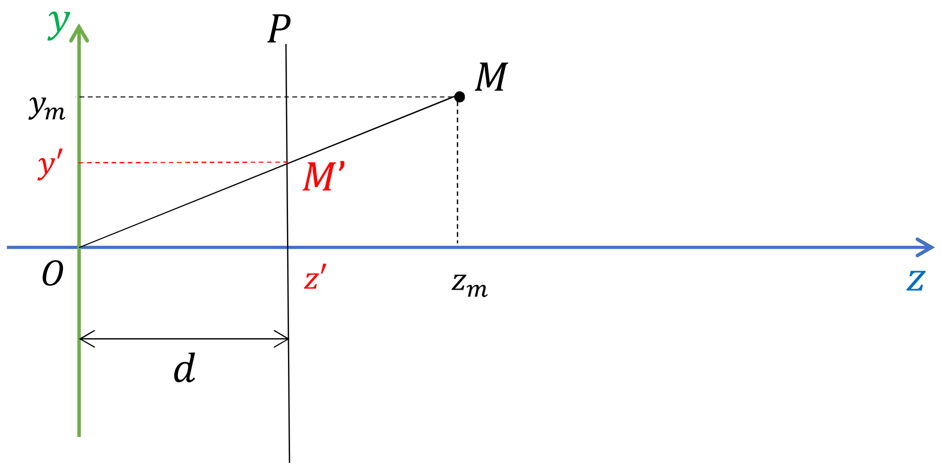
Perspective conique

1 point de fuite : sur la ligne d'horizon, dans la direction du regard

Comment calculer cette projection ?

Par une matrice de projection.

Matrice de projection perspective



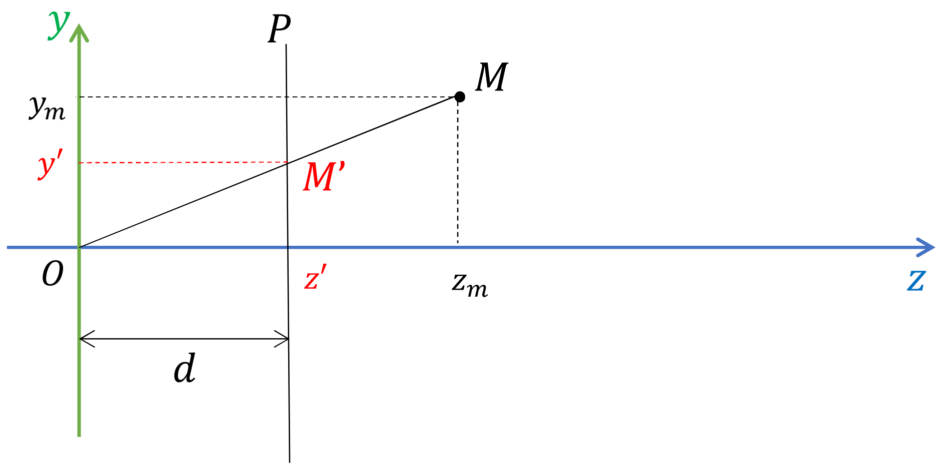
Soit O le centre de projection choisi pour la scène et P le plan de projection choisi pour former une image.

On appelle distance focale d la distance entre le point de projection et ce plan P

Soit M un point dans l'espace, de coordonnées $\begin{pmatrix} y_m \\ z_m \end{pmatrix}$

Quelles sont les coordonnées cartésiennes $\begin{pmatrix} y' \\ z' \end{pmatrix}$ de M', projection de M sur le plan P ?

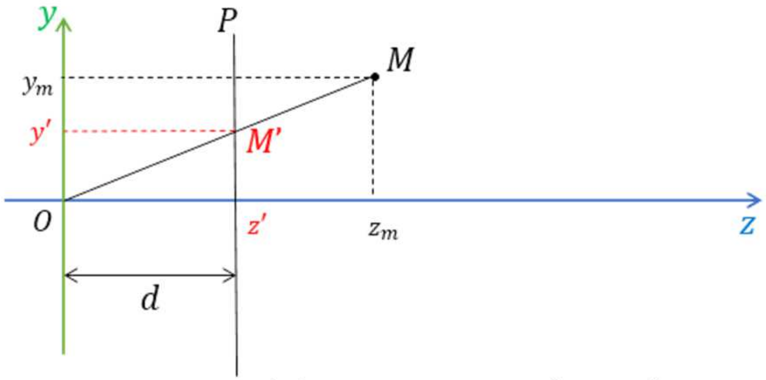
Matrice de projection perspective



en utilisant $\frac{z_m}{d}$, réécrivez la relation liant M' et M en coordonnées homogènes.

Exprimez la relation liant M' et M en coordonnées homogènes à l'aide d'un calcul matriciel simple (une matrice 3x3)

l'opération de projection est-elle réversible ? Quelle particularité de la matrice de projection traduit ce fait ?



$$z' = d$$

On utilise le théorème de Thalès pour y'

$$\boxed{\frac{y'}{y_m} = \frac{d}{z_m},\ y' = y_m.\frac{d}{z_m}}$$

On écrit M et M' en coordonnées homogènes

$$M: \begin{pmatrix} y_m \\ z_m \\ 1 \end{pmatrix}$$

$$M': \begin{pmatrix} y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} y_m.\dfrac{d}{z_m} \\ d \\ 1 \end{pmatrix} \times \frac{z_m}{d} = \begin{pmatrix} y_m.\dfrac{d}{z_m}.\dfrac{z_m}{d} \\ d.\dfrac{z_m}{d} \\ \dfrac{z_m}{d} \end{pmatrix} = \begin{pmatrix} y_m \\ z_m \\ \dfrac{z_m}{d} \end{pmatrix}$$

Exercice : Matrice de projection

On cherche P telle que $P.M \ = M'$

$$\begin{pmatrix} y_m \\ z_m \\ 1 \end{pmatrix} = M$$

$$P = \begin{pmatrix} & & & \\ & & & \\ & & & \end{pmatrix} \begin{pmatrix} y_m \\ z_m \\ \frac{z_m}{d} \end{pmatrix} = M'$$

Solution

Transposition en 3D : $M(x_m, y_m, z_m)$
Se projette en $M'(x', y', z' = d)$

Avec $x' = x_m.\frac{d}{z_m}$

$$\begin{pmatrix} y_m \\ z_m \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{d} & 0 \end{pmatrix} \quad \begin{pmatrix} y_m \\ z_m \\ \frac{z_m}{d} \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{pmatrix}$$

La matrice de projection n'est pas inversible

Caméra

Constitution de l'image à partir du modèle 3D

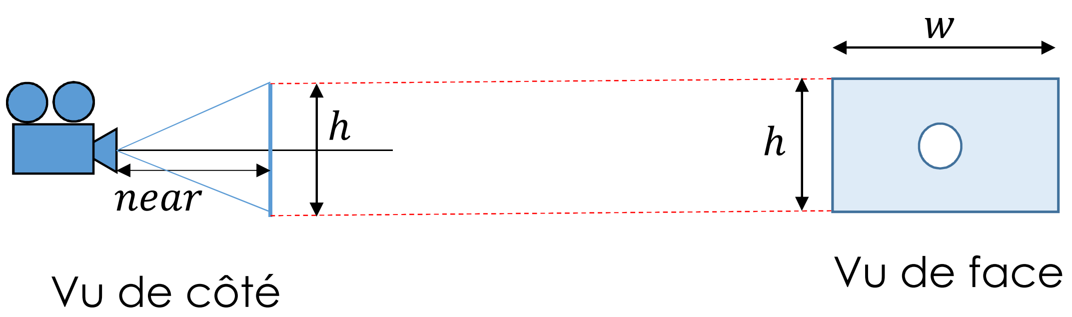
Passage 3D -> 2D

Quels paramètres définir ?

Champ de vue 'virtuel' pour la synthèse d'images

View frustum

On caractérise le **plan de projection** : (n, h, w)



Ou $(n, \alpha = \text{fov}_y, ar)$ avec $\tan(\alpha) = \dfrac{h}{2.near}$ et $ar = \dfrac{w}{h}$

Far plane en synthèse d'images

Définition d'un volume **utile** de projection

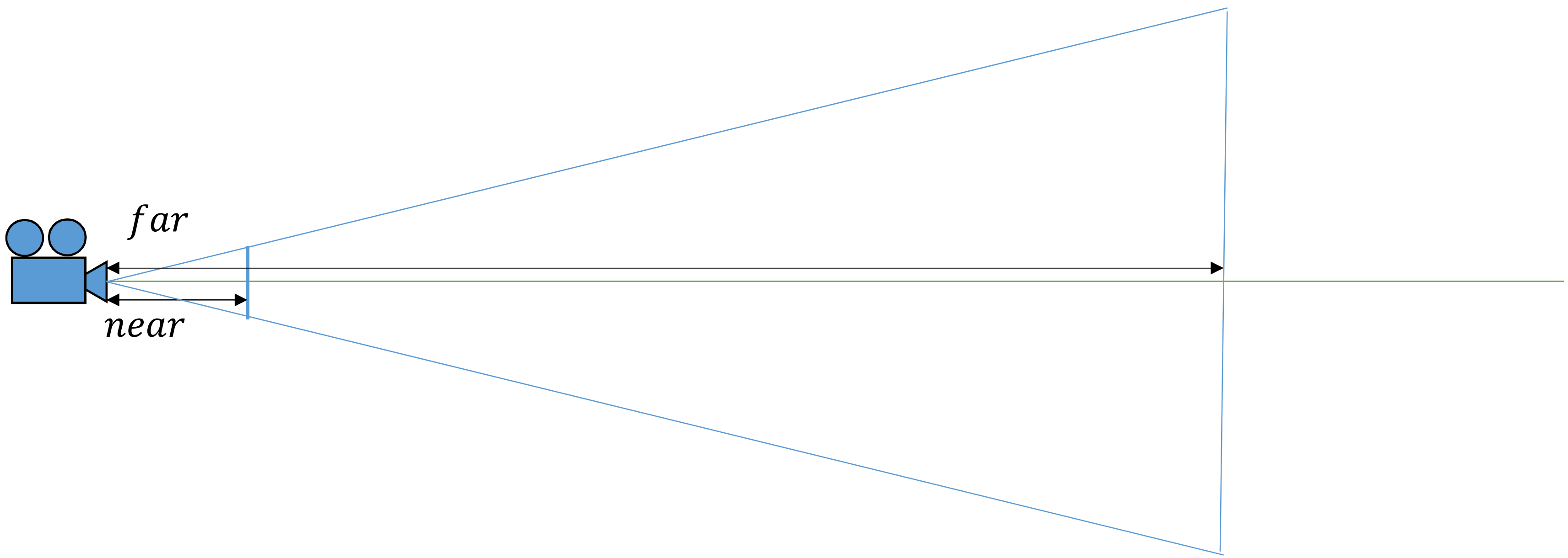
Au-delà d'une certaine distance à la caméra :

✓Objets très petits (quelques pixels)
✓Probablement cachés par d'autres (occlusion)

Le ratio d'utilité : taille occupée à l'écran/temps de calcul est très faible.

Far plane

Définition d'un volume utile pour la projection et le rendu

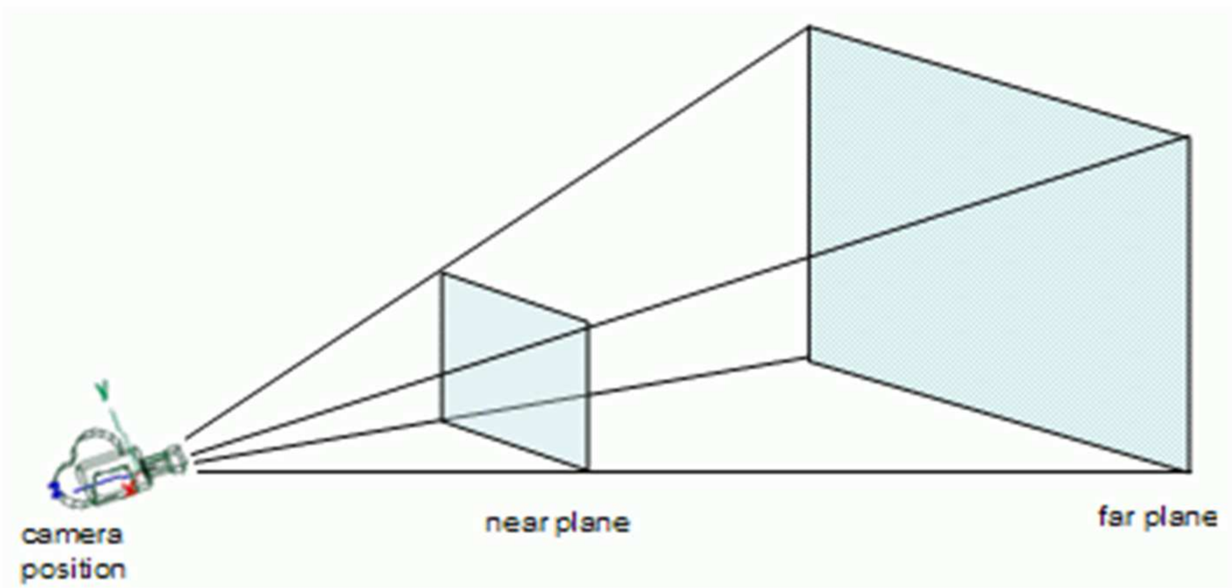


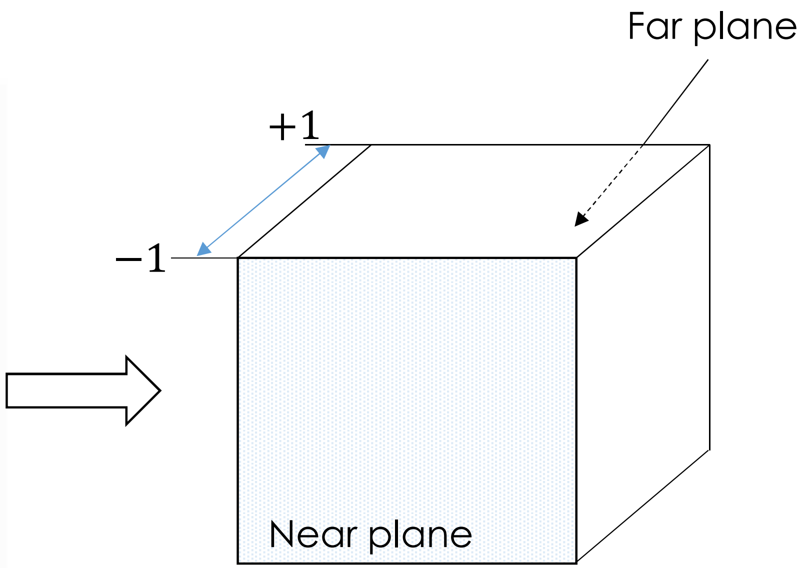
Influence sur la projection

Normer l'espace objet vers l'espace image

Transformation non linéaire pour la qualité de l'image

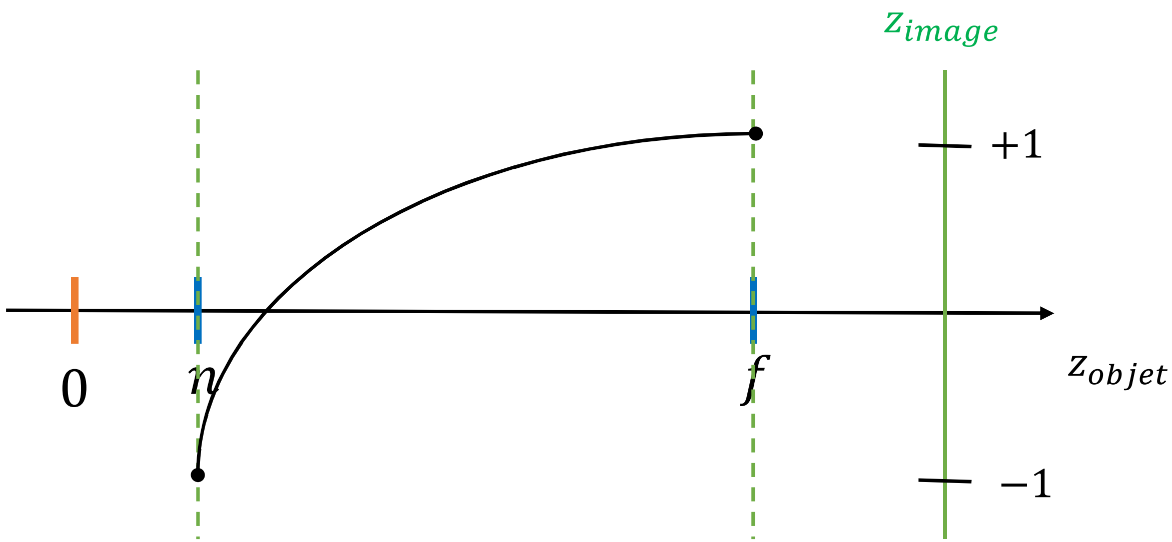
Frustum en 3D





de http://www.lighthouse3d.com/tutorials/view-frustum-culling/

Normalisation sur z



Plus de précision vers le near plane

$$z_i = f\left(\frac{1}{z_o}\right) = A.\frac{1}{z_0} + B$$

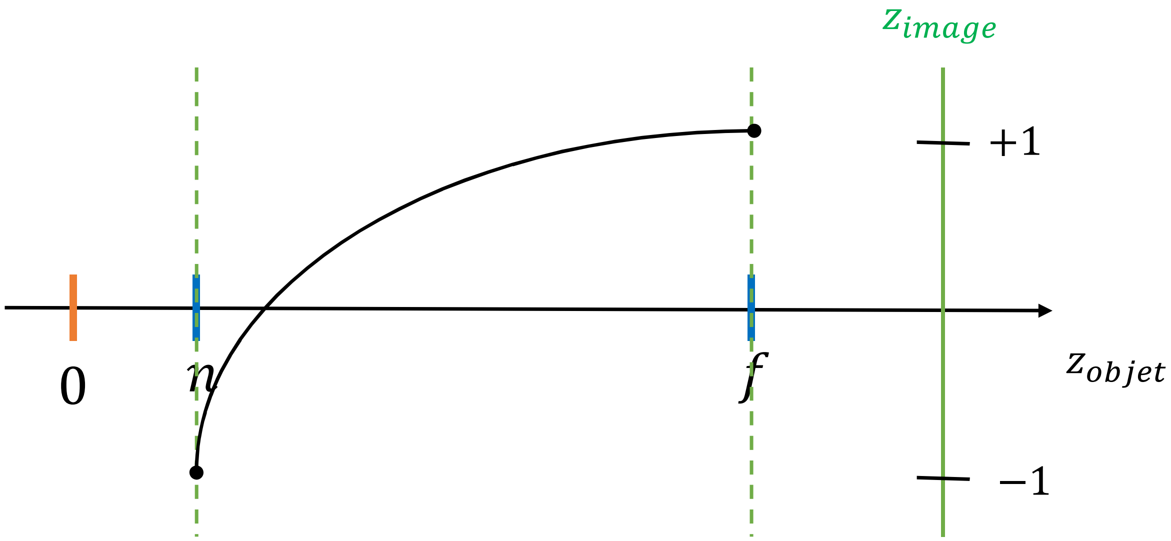
$$-1 = A.\frac{1}{n} + B$$

$$1 = A.\frac{1}{f} + B$$

$$A = -\frac{2fn}{f-n}$$

$$B = \frac{f+n}{f-n}$$

Normalisation sur z



$$z' = -\frac{2fn}{f-n}\frac{1}{z} + \frac{f+n}{f-n}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & B & A \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

En normalisant le frustum

Largeur w définit l (left) et r (right) : $w = l + r$

Hauteur h définit t (top) et b (bottom) : $h = t + b$

$$\begin{bmatrix} \frac{n}{r} & 0 & 0 & 0 \\ 0 & \frac{n}{t} & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Espace à projeter, éliminations

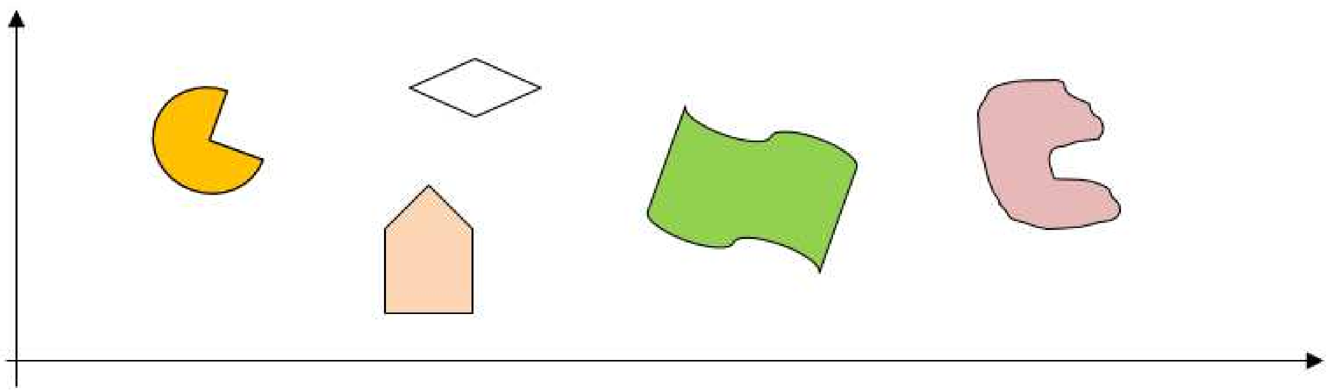
Ne projeter que les objets qui font partie du frustum

Procéder à des éliminations rapides : modéliser un objet de manière simple.

Représentation géométrique réduite. OBB, AABB, BS

AABB

Axis Aligned Bounding Box : le plus petit parallélépipède qui englobe l'objet, axes parallèles aux axes du repère



Dessinez les AABB des objets proposés.

Algorithmes AABB, BS

Ecrivez les algorithmes de construction d'une AABB et d'une BS

En entrée : liste de sommets **`s[i]`**, nombre de sommets **`n`**, chaque sommet a des coordonnées x, y et z accessibles par **`s[i].x`**, **`s[i].y`**, **`si.[z]`**

En sortie : les caractéristiques de l'AABB ou de la BS

Mise à jour d'une AABB/BS

Comment mettre à jour une BS ?

- Rotation
- Translation
- Scaling (homothétie différenciée)

Comment mettre à jour une AABB ?

- Translation
- Rotation
- Scaling

AABB et rotation

Méthode rapide mais sous optimale, pour ne pas tester tous les sommets

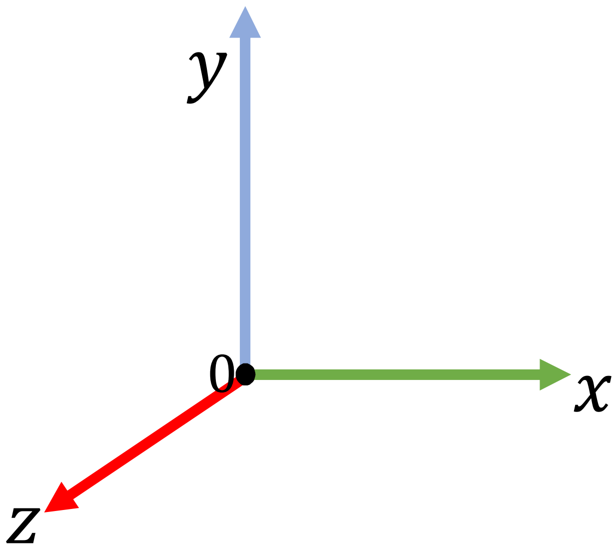
Rotation des sommets de l'AABB, puis recalcul d'AABB à partir de l'ancienne

AABB élargie mais couvre l'objet :

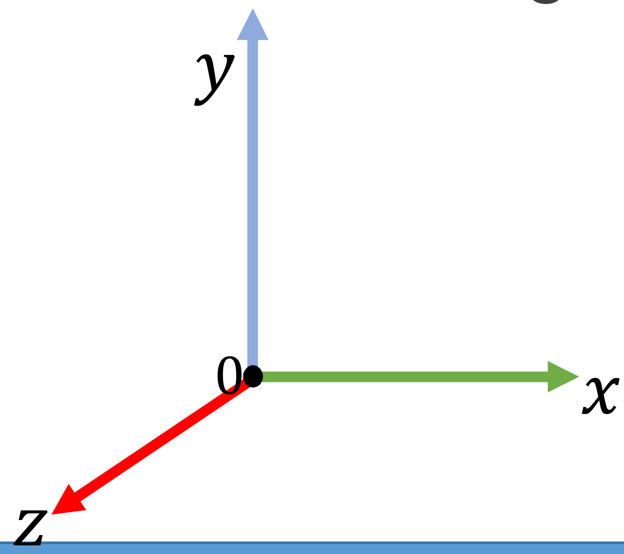
Exemple

Soit l'AABB $(2,-1,-1)-(4,1,1)$

Dessinez-la sur le schéma suivant :



Elle subit une rotation d'angle $\frac{\pi}{4}$ autour de l'axe Ox : dessinez-là, calculez la nouvelle AABB et dessinez-là également



Définition géométrique du frustum

6 plans,

Un plan = $(a, b, c, d) = (\overrightarrow{n}\begin{pmatrix} a \\ b \\ c \end{pmatrix}, d)$ $ax + by + cz + d = 0$

Rappel : position d'un point *P* par rapport à un plan (produit scalaire svp !)

BS : intersection sphère/plan

Ecrivez l'algorithme indiquant si un plan a une intersection avec une sphère

Elimination (step 1)

Ecrivez un algorithme qui indique si une sphère est totalement du ‘mauvais côté’ d’un plan du frustum

Ecrivez un algorithme indiquant si une sphère a une intersection avec un frustum de caméra

Choix de conception : garder ou rejeter ?

Elimination (step 2)

Ecrivez un algorithme qui indique si une AABB est totalement du ‘mauvais côté’ d’un plan du frustum

Ecrivez un algorithme indiquant si une AABB a une intersection avec un frustum de caméra

Alors, BS ou AABB ?

Utile pour le BSP, quadtrees, octrees

Graphe de scène

Arbre décrivant une hiérarchie, une dépendance d’objets

Exemple avec Ogre3D

Organisation d'une scène

```
Ogre::Root*         mRoot;
Ogre::Camera*       mCamera;
Ogre::SceneManager* mSceneMgr;
Ogre::RenderWindow* mWindow;
```

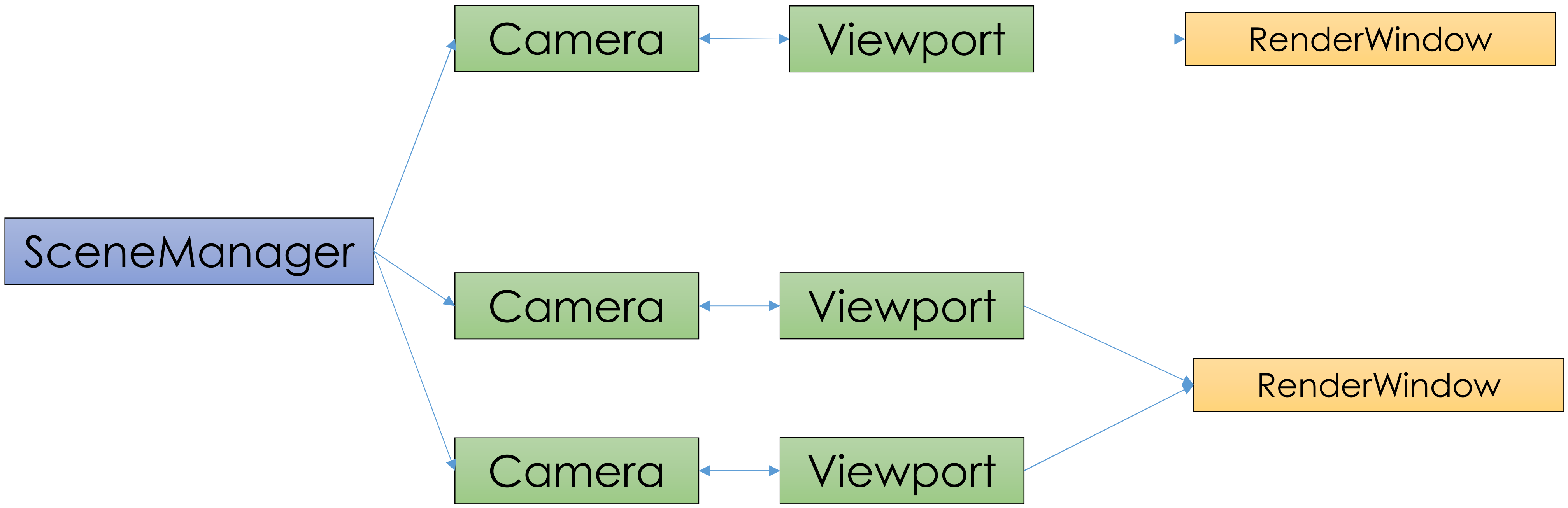
Objet **mRoot** – configuration d'Ogre3D (direct3D ou openGL ? Plein écran ou non ?)

Objet **mCamera** – SetPosition / lookAt / setNearClipDistance / setFOVy / setAspectRatio
cf : https://www.ogre3d.org/docs/api/1.9/class_ogre_1_1_camera.html

Frustum : accès aux Ogre::Plane

Objet **mSceneManager** – gère le culling (abattage/sélection) et le rendu d'une collection d'objets, en choisissant quelle(s) méthode(s) utiliser.

RenderWindow



Retour au graphe de scène

```
mRoot->createSceneManager(Ogre::ST_GENERIC);
Ogre::SceneNode *rootSceneNode = sceneManager->getRootSceneNode();
```

Sert de référence aux autres nœuds, position(0,0,0)

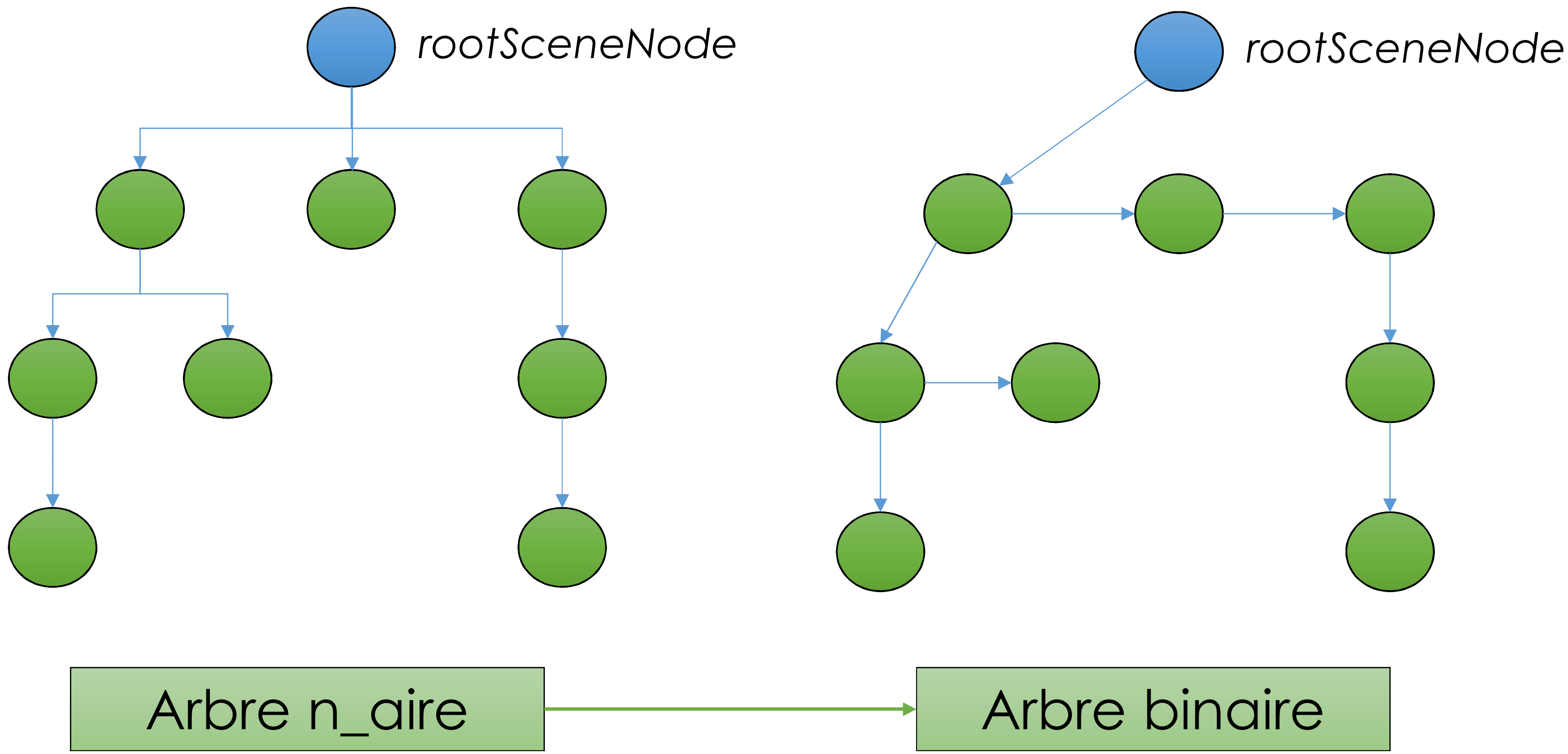
```
Ogre::SceneNode *someNode = anotherNode->createChildSceneNode("name");
```

nœud Enfant

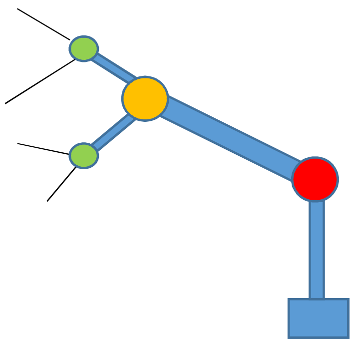
nœud Parent

Seule manière de créer un noeud

Structure d'arbre



Exemple : quel est le graphe de scène de l'objet ci-dessous ?



SceneNodes

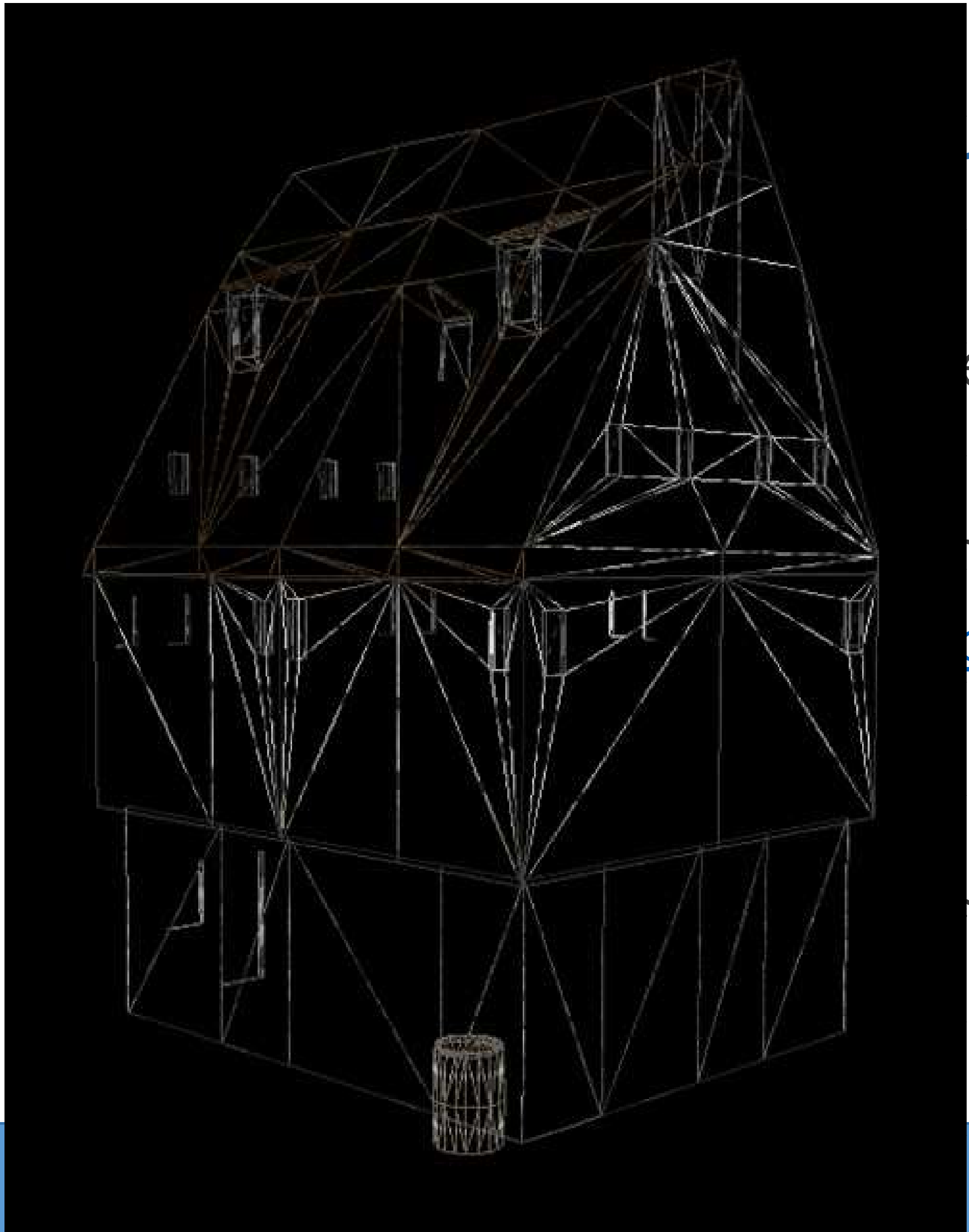
Les caractéristiques géométriques d'un nœud sont définies par rapport à celles de son nœud parent

Position (x, y, z) dans le repère du nœud parent

Taille $(s_x\ , s_y, s_z)$ par rapport à la taille du nœud parent

Orientation (q) dans le repère du nœud parent

Possibilité de les exprimer en fonction du *rootSceneNode*



ntités

e visualise rien :

une entité Ogre::Entity :

api/1.9/class_ogre_1_1_entity.html

maillage + un matériau

```
reateEntity("cube","tudorhouse.mesh");
```

Nom interne de l'entité

Nom du fichier contenant le maillage

```
","tudorhouse.mesh");

gleton().create("texturecube","General");

State("fw12b.jpg");
```

ar un fichier de texture
(jpg)
r déclaratif (.material)

AABB / Nodes / entities

Les AABB sont associées à des étendues dans l'espace

SceneNode ou Entity ?

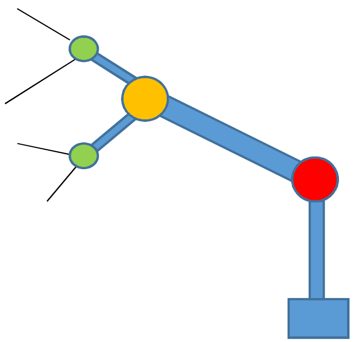
Elles peuvent être regroupées pour traitement rapide du frustum culling.

Sur quel critère regrouper ?

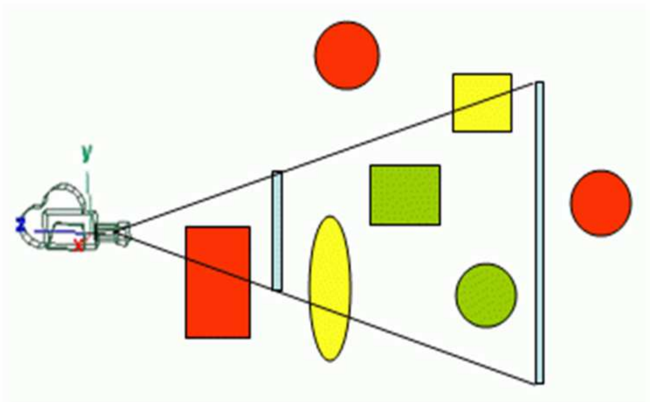
Comment les regrouper ?

Dessiner l'AABB de l'articulation rouge

Retour sur le frustum culling



Frustum culling, le retour



Sur le frustum proposé à gauche (vu de dessus), quels objets sont visibles ?

Comment traiter les objets partiellement dans le frustum ?

Subdivision puis clipping

On élimine encore

Transformations géométriques

Frustum culling

!

Projection

Clipping

?

?

?

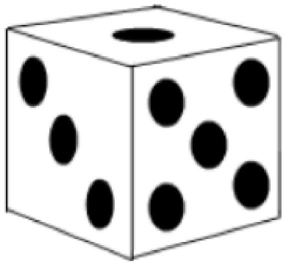
Visibilité

Être présent dans le frustum

≠

Être visible, être rendu

occlusion



Exercice

Soit un cube. Combien de facettes triangulaires sont nécessaires pour avoir un maillage d'un cube ?

Combien de faces d'un cube sont visibles à un instant donné ? Combien de triangles seront rendus au maximum ?

Stockage de maillages

Proposer un format d'écriture de données (format de fichier) pour stocker de manière simple les points du maillage d'un cube de centre (0,0,0) et de côté 2.

Contrainte : dans ce format, les coordonnées de sommets ne doivent apparaître qu'une seule fois

Stockage de maillages

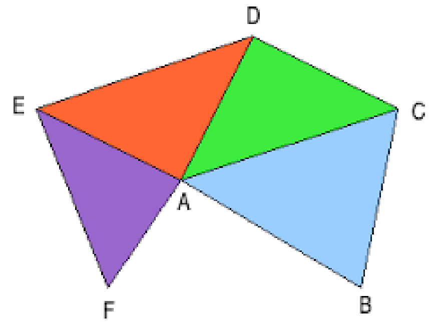
Proposer un format d'écriture de données (format de fichier) pour stocker la liste des facettes triangulaires composant le maillage

Contrainte : Vous devez utiliser la liste de sommets de l'étape précédente

Stockage de maillages (3)

Proposer un format compact pour les structures :

Trianglefan :



Trianglestrip :