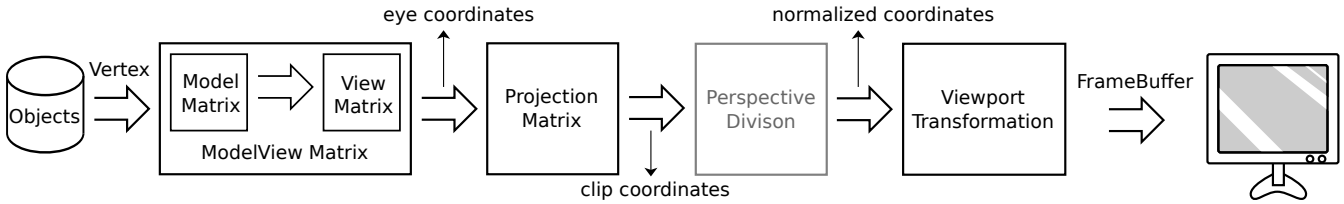
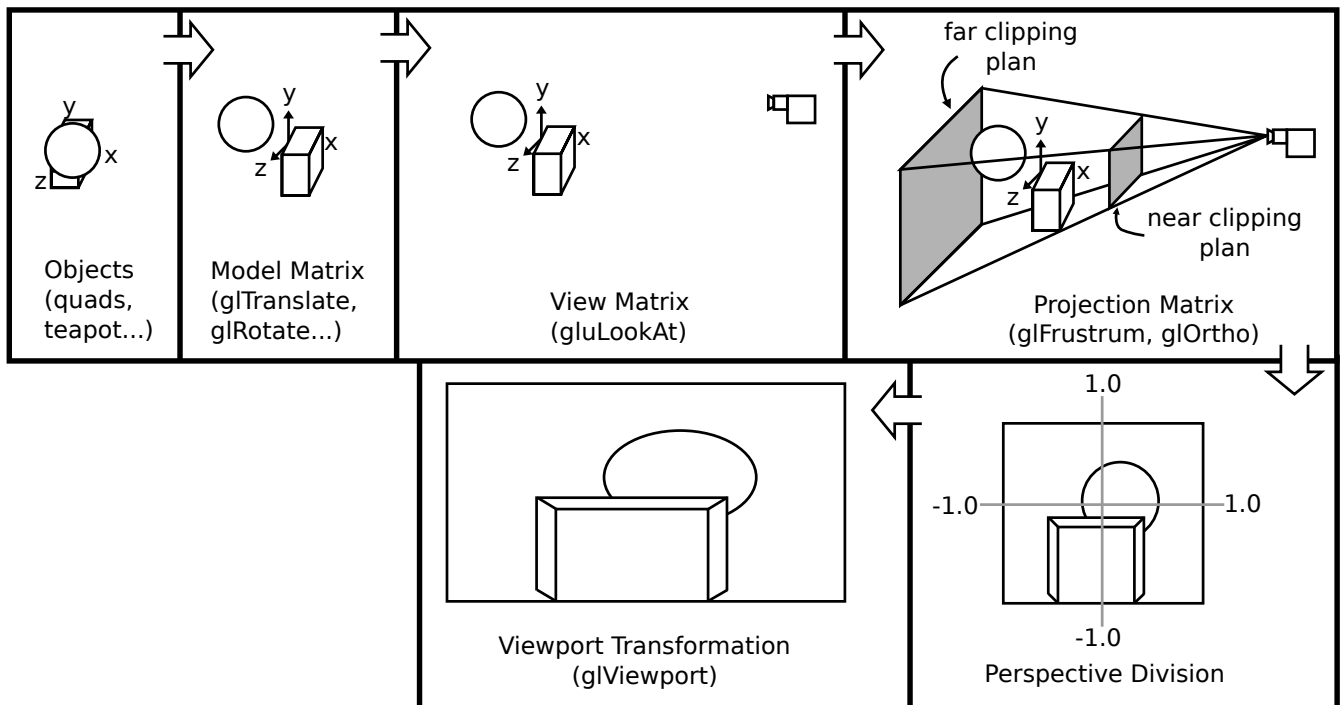


# Chaîne de rendu OpenGL

Voici un schéma illustrant les différentes étapes lors de la création d'un rendu OpenGL :



- **Objects** : Liste des objets à dessiner.
- **ModelView Matrix** : Elle regroupe le travail de deux matrices :
  - Model Matrix : Elle place les objets dans la scène 3D (glTranslate, glRotate...).
  - View Matrix : Elle place la caméra (gluLookAt).
- **Projection Matrix** : Elle définit le volume dans lequel la scène doit être dessinée (entre les deux plans de clipping *far* et *near*) et génère une image 2D en *projetant* la scène 3D sur le plan de clipping le plus proche de la caméra (near clipping plan) (glFrustum, glOrtho).
- **Perspective Division** : Cette étape correspond à une normalisation de l'image générée par la matrice de projection, de façon à ce que la nouvelle image soit comprise entre [-1,1] sur les trois axes (même si l'image est affichée en deux dimensions, OpenGL garde quand même une information de profondeur sur chaque pixel).
- **Viewport Transformation** : Cette dernière étape consiste à adapter l'image normalisée à la taille de la fenêtre dans laquelle la scène est affichée (glViewport).



## LE PICKING

Connaissant les coordonnées  $(x,y)$  d'un point de l'écran (clic de la souris par exemple) nous voulons remonter la chaîne de rendu OpenGL pour obtenir les coordonnées  $(x,y,z)$  de l'objet correspondant dans la scène 3D.

L'image affichée à l'écran est calculée dans le *Framebuffer* (littéralement un *tampon d'image*) qui correspond schématiquement à un ensemble d'attributs pour tous les pixels à afficher. Parmi ces attributs, il est intéressant de noter qu'OpenGL, bien qu'on lui demande d'afficher une image en 2D, calcule une profondeur pour chacun des pixels (attribut `GL_DEPTH_COMPONENT`). Cette profondeur est comprise entre 0.0 (le plan de clipping le plus proche), et 1.0 (le plan de clipping le plus loin). Cette valeur peut être récupérée à l'aide de la fonction **glReadPixels()**.

La fonction **gluUnProject()** permet, à partir des coordonnées  $(x,y,z)$  d'un pixel dans l'image finale et des différentes matrices, de *remonter* la chaîne de rendu de façon à obtenir les coordonnées  $(x,y,z)$  dans la scène 3D de départ (en inversant les différentes matrices).

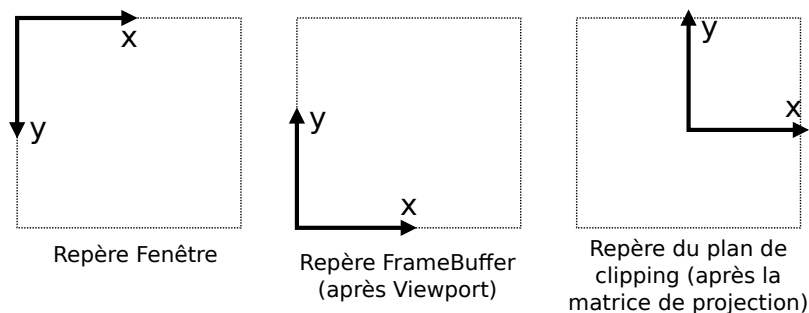
Pour résoudre le problème de picking il faut donc récupérer les trois matrices

```
GL_MODELVIEW_MATRIX,
GL_PROJECTION_MATRIX,
GL_VIEWPORT,
```

ainsi que la composante de profondeur du pixel considéré (**glReadPixels**) et utiliser la fonction **gluUnProject**.

### ATTENTION !

Le repère de l'écran n'est pas le même que celui du *Framebuffer* !



(le troisième repère est donnée à titre informatif)

Pour pouvoir utiliser la fonction **glReadPixels**, il faut donc transformer la coordonnée **y** du point sélectionné dans la fenêtre dans le repère du *Framebuffer*.

Pour cela on peut récupérer et utiliser la matrice viewport :

```
viewport = glGetIntegerv( GL_VIEWPORT )
```

sachant que :

`viewport[0]` : position en x de l'image dans la fenêtre (0 dans notre cas)

viewport[1] : position en y de l'image dans la fenêtre (0 dans notre cas)  
viewport[2] : largeur de l'image (x maximal)  
viewport[3] : hauteur de l'image (y maximal)