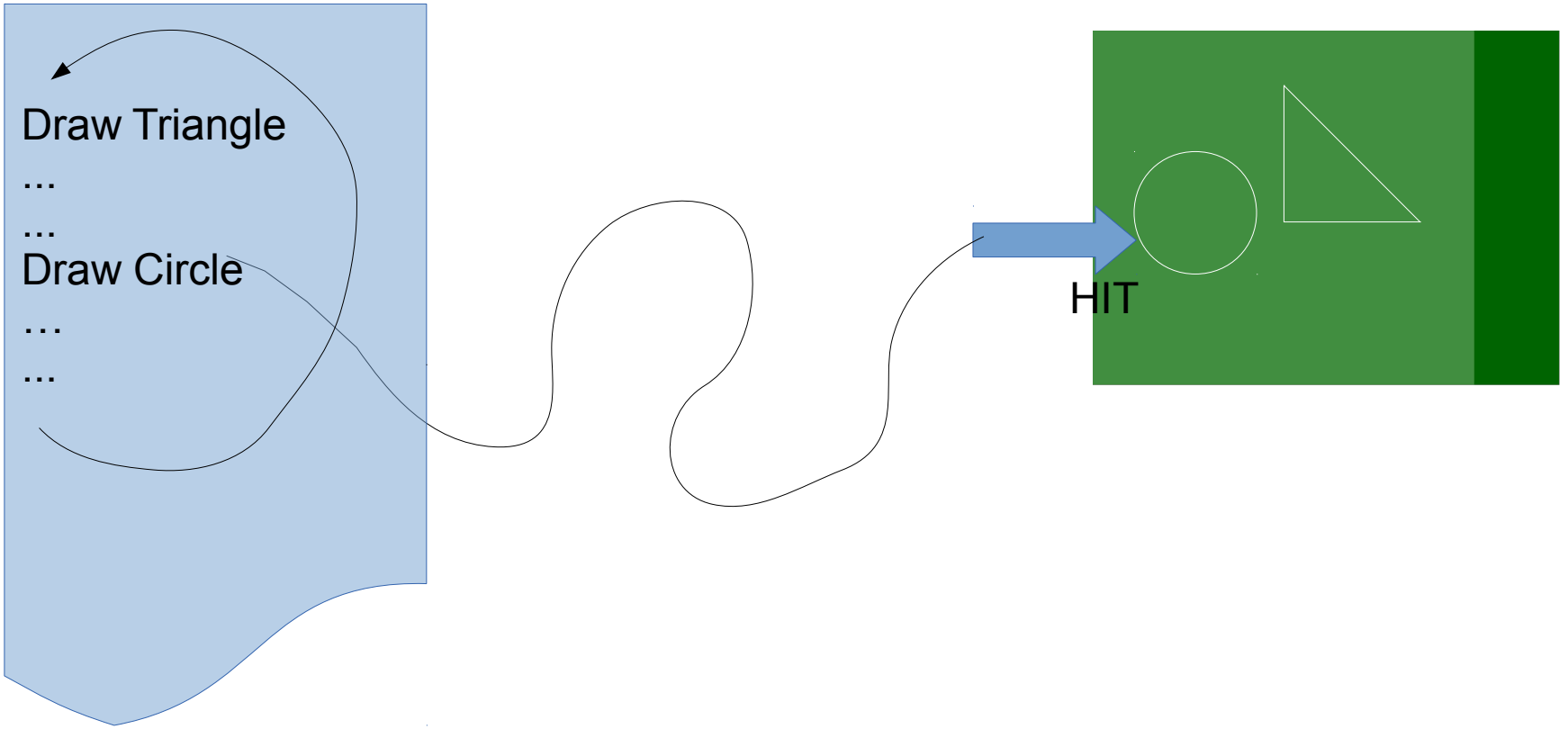
Désignation (Picking)

M Neveu

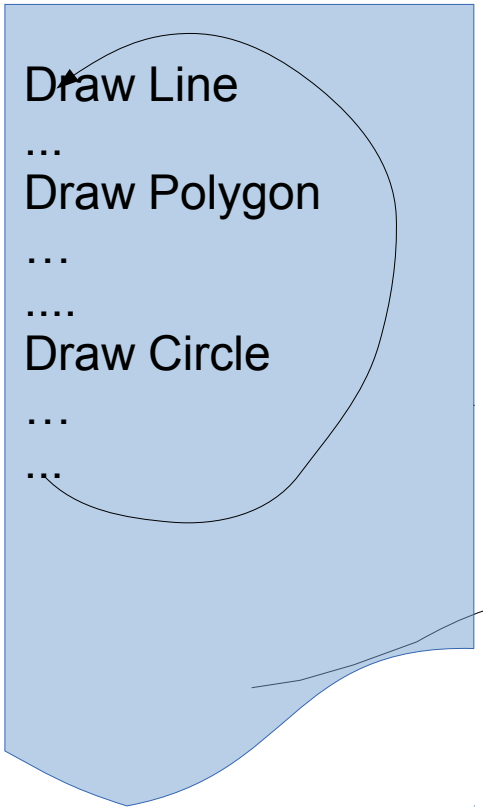
Au début

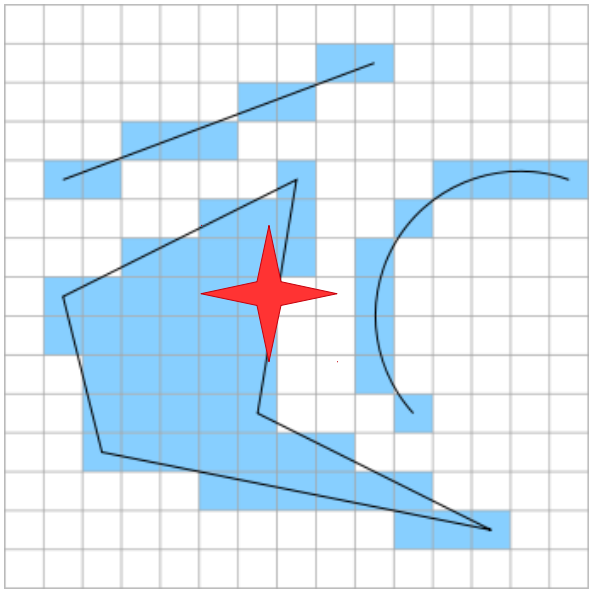
- Liste de visualisation et écrans à balayage



Maintenant

- Liste de visualisation et rasterization





OpenGL

1. Récupérer les coordonnées écran de la souri
2. Passer en mode « sélection »
3. Redéfinir le volume de visualisation autour du curseur
4. Tracer la scène (tout ou partie des primitives)
5. Sortir du mode sélection et récupérer les identifiants des objets.

OpenGL

Identifier = un nom par objet ou par primitive

En mode sélection, les objets ne sont pas tracés dans le framebuffer. Les noms des objets (plus z) sont gardés dans une liste. Pour les objets sans nom seul z est gardé.

Un événement « hit » intervient quand une primitive est tracée en mode sélection. Les Hits sont enregistrés dans le selection buffer.

Quand on sort du mode sélection, on récupère les hits + z ce qui permet de repérer l'objet le plus proche.

Exemple OpenGL

```
#define BODY    1
#define HEAD    2


...
void
renderInSelectionMode() {
1    glInitNames();

2    glPushName(BODY);
3    drawBody();
4    glPopName();

5    glPushName(HEAD);
6    drawHead();
7    drawEyes();
8    glPopName();

9    drawGround();
}
```

Pile de noms vide

1 nom dans la pile. Pas encore e hit => rien dans le selection buffer

OpenGL trace qqch. Si une primitive coupe le volume de visualisation, un hit est créé avec le nom BODY, plus le z min et le z max des primitives

Dépile → le hit (s'il existe) en 2 est gardé dans le selection buffer

Pile modifiée, pas encore de hit, rien dans le selection buffer

Idem 3

Si une primitive coupe le VV et déjà un hit en 6, màj zmin et zmax. Si pas de hit en 6, un hit créé.

Idem 4

Si une primitive coupe le VV, un hit créé sans nom, avec zmin et zmax

4 et 5 <=> glLoadName(HEAD);

Un ou plusieurs noms

UN :

```
for(int i = -3; i < 3; i++)
    for(int j = -3; j < 3; j++) {
      glPushMatrix();
      glPushName(i*6+j);
      glTranslatef(i,j,0);
      glCallList(OBJET);
      glPopName();
      glPopMatrix();
}
```

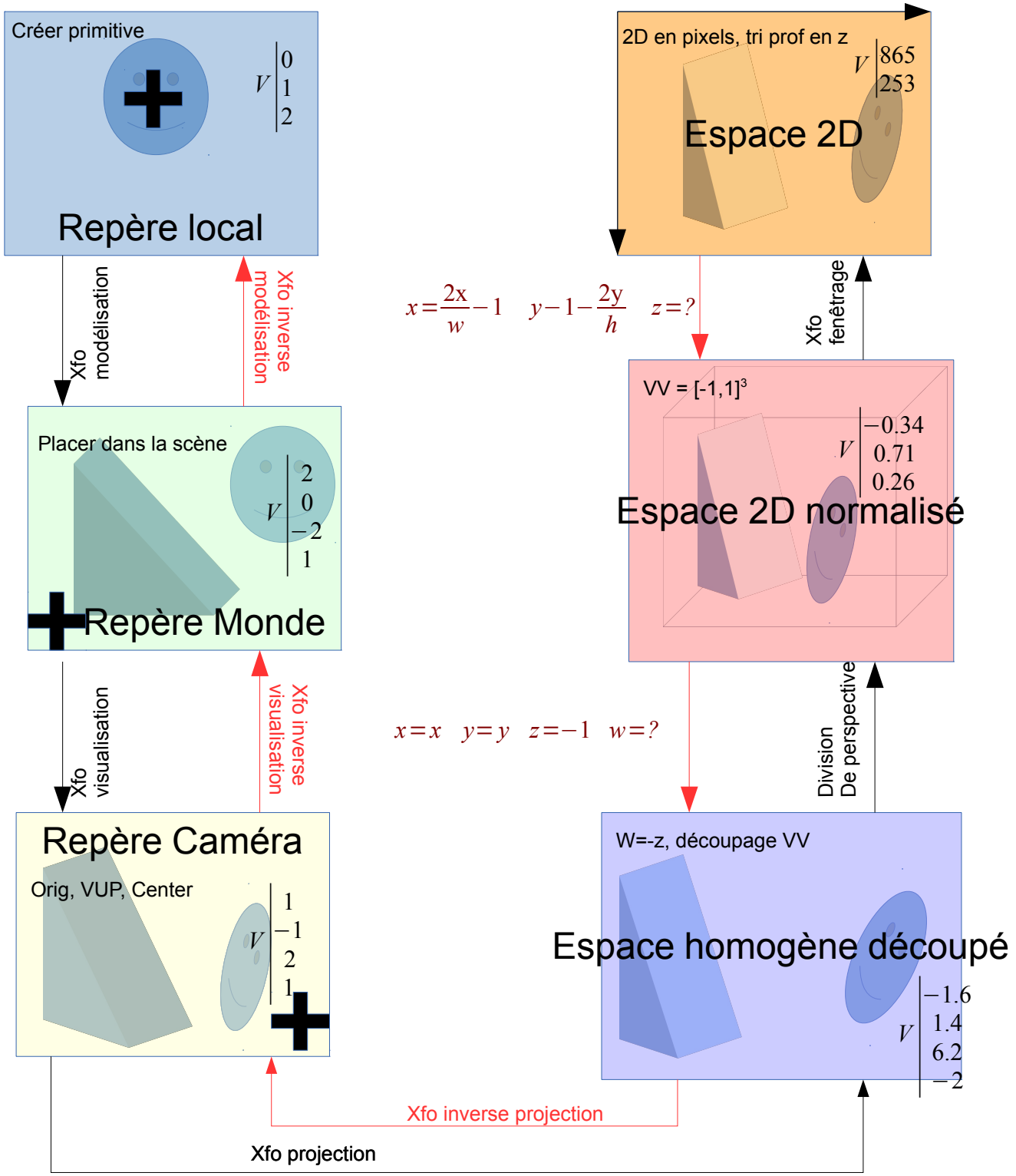
DEUX :

```
for(int i = -3; i < 3; i++) {
    glPushName(i);
    for(int j = -3; j < 3; j++) {
        glPushMatrix();
        glPushName(j);
        glTranslatef(i,j,0);
        glCallList(OBJET);
        glPopName();
        glPopMatrix();
        }
    glPopName();
}
```

Picking par lancer de rayon

- Principe : projeter un rayon depuis la souris à travers la caméra et on regarde les objets touchés dans la scène
- Le rayon a pour origine la caméra. Tout doit être ramené dans le même espace où se fait le calcul d'intersection (en général l'espace de la scène)
- Lancer de rayon classique avec des primitives
- Englobants : sphère ou cube

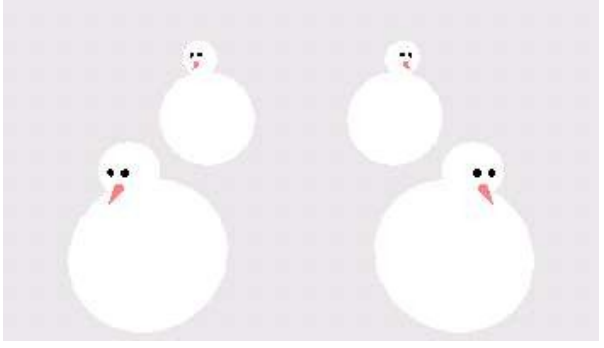
Picking par lancer de rayon

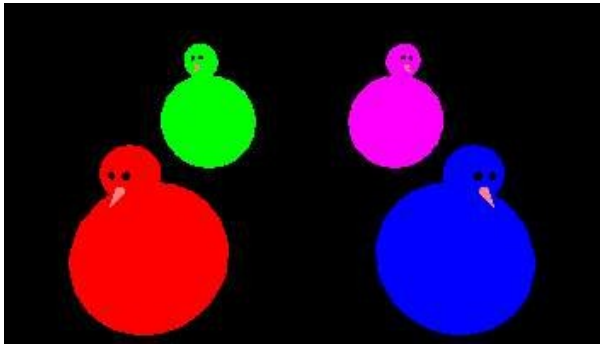


Picking par tampon de couleurs

1) On trace la scène dans un buffer, avec une couleur différente par objet mais sans lumières, textures etc...Utiliser les mêmes transformations (caméra) que pour le véritable affichage.
2) Ensuite, récupérer la couleur du pixel dans le buffer sous la souris : comme chaque objet a une couleur unique, on a l'objet.

NB: les couleurs en RGB sont des entiers de 0 (noir) à 16 777 216 (blanc). On identifie les objets par des entiers (table de hashage).Attention aux objets transparents





```
glDisable(GL
_DITHER);

     .....

glEnable(GL_
DITHER);
```