Synthèse d'Images

– IG3D –

Jonathan Fabrizio
LRDE-EPITA

UPMC
SORBONNE UNIVERSITÉS

EPITA
ÉCOLE D'INGÉNIEURS EN INFORMATIQUE

Rendu temps réel

Principe général
Algorithmes 3D fondamentaux
Algorithmes 2D fondamentaux

Rendu temps réel

Principe général

Principe général

- Modélisation des objets dans un repère local
- Modélisation de la scène dans un repère global
- Projection de la scène sur le plan image
  - passage repère global au repère caméra
  - projection sur le plan image (+dessin 2D)

Rendu temps réel

Algorithmes 3D fondamentaux

Algorithmes 3D fondamentaux

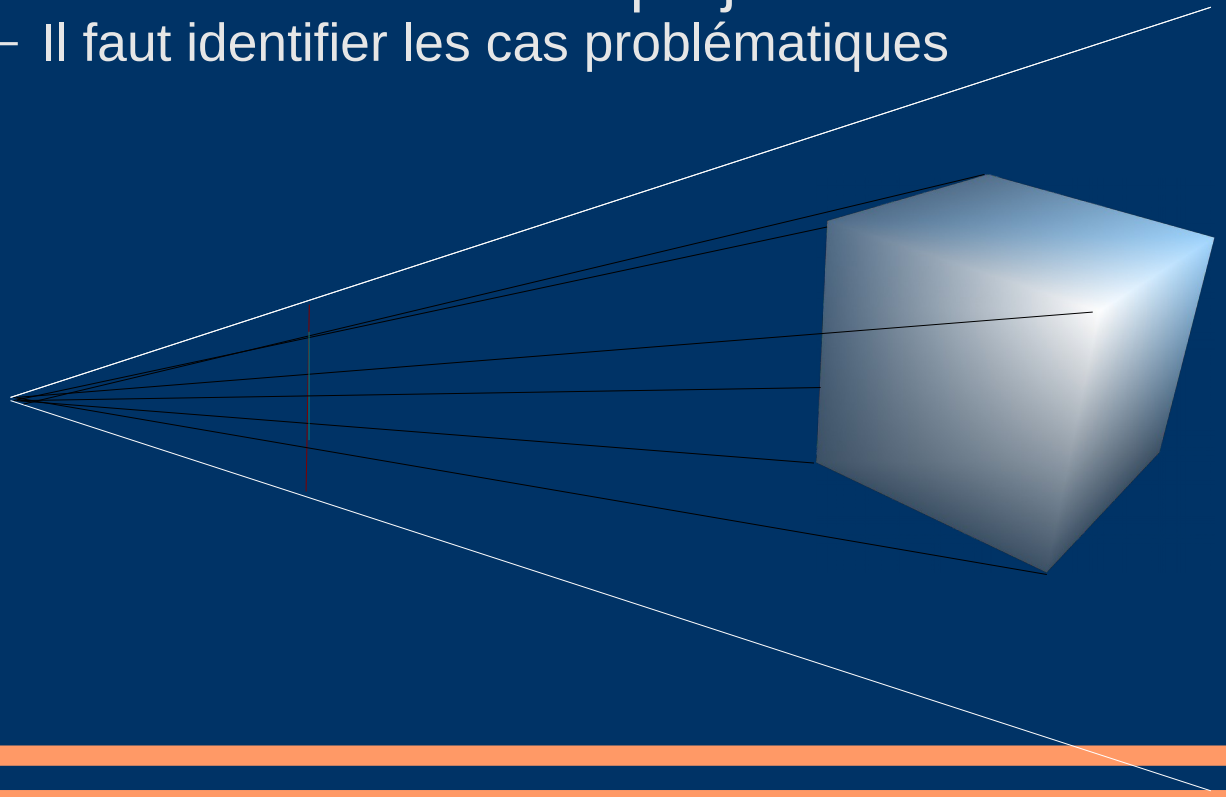
- Étape 1 : modélisation des objets dans un repère local
- Étape 2 : insertion des objets dans la scène

Algorithmes 3D fondamentaux

- Étape 1 : modélisation des objets dans un repère local
- Étape 2 : insertion des objets dans la scène
- Étape 3 : passage du repère mode au repère caméra afin de pouvoir réaliser la projection sur le plan image

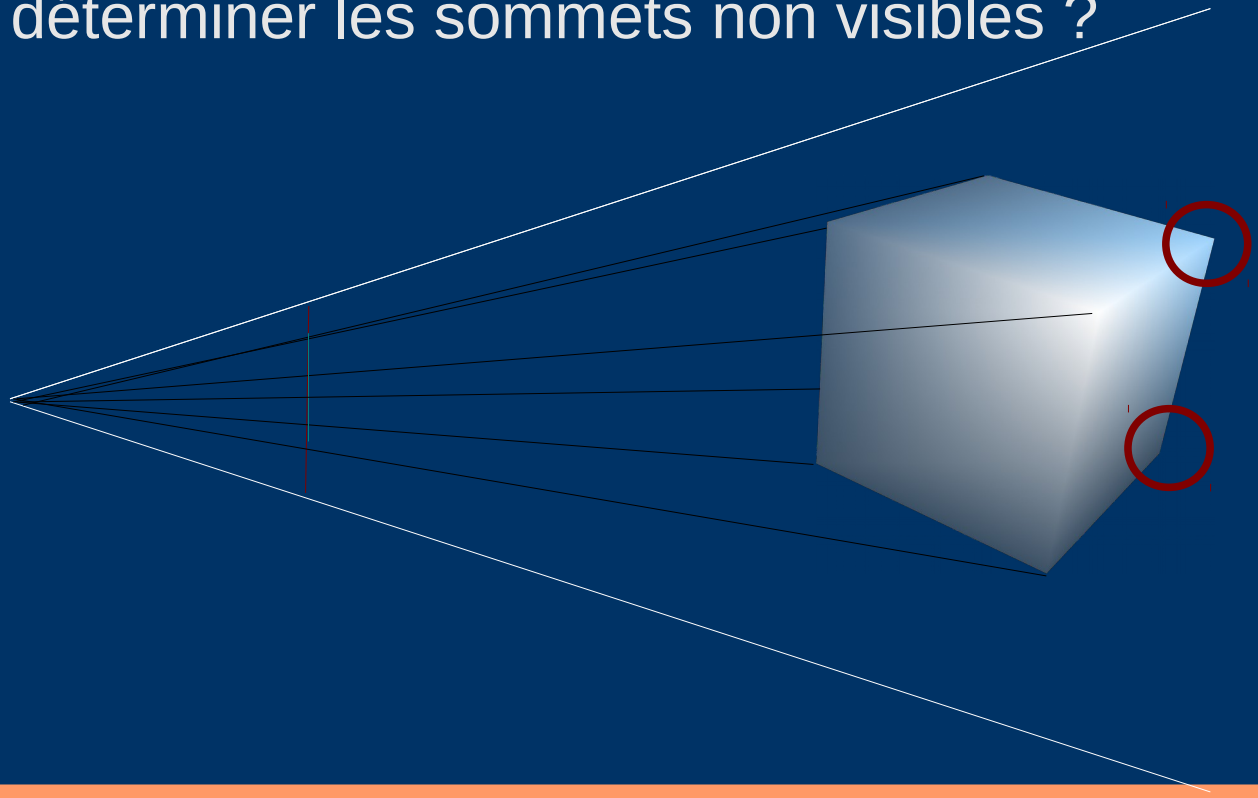
Algorithmes 3D fondamentaux

- Préparation de la projection
  - Projection des sommets
    - Comment réaliser cette projection efficacement ?
      - Il faut identifier les cas problématiques



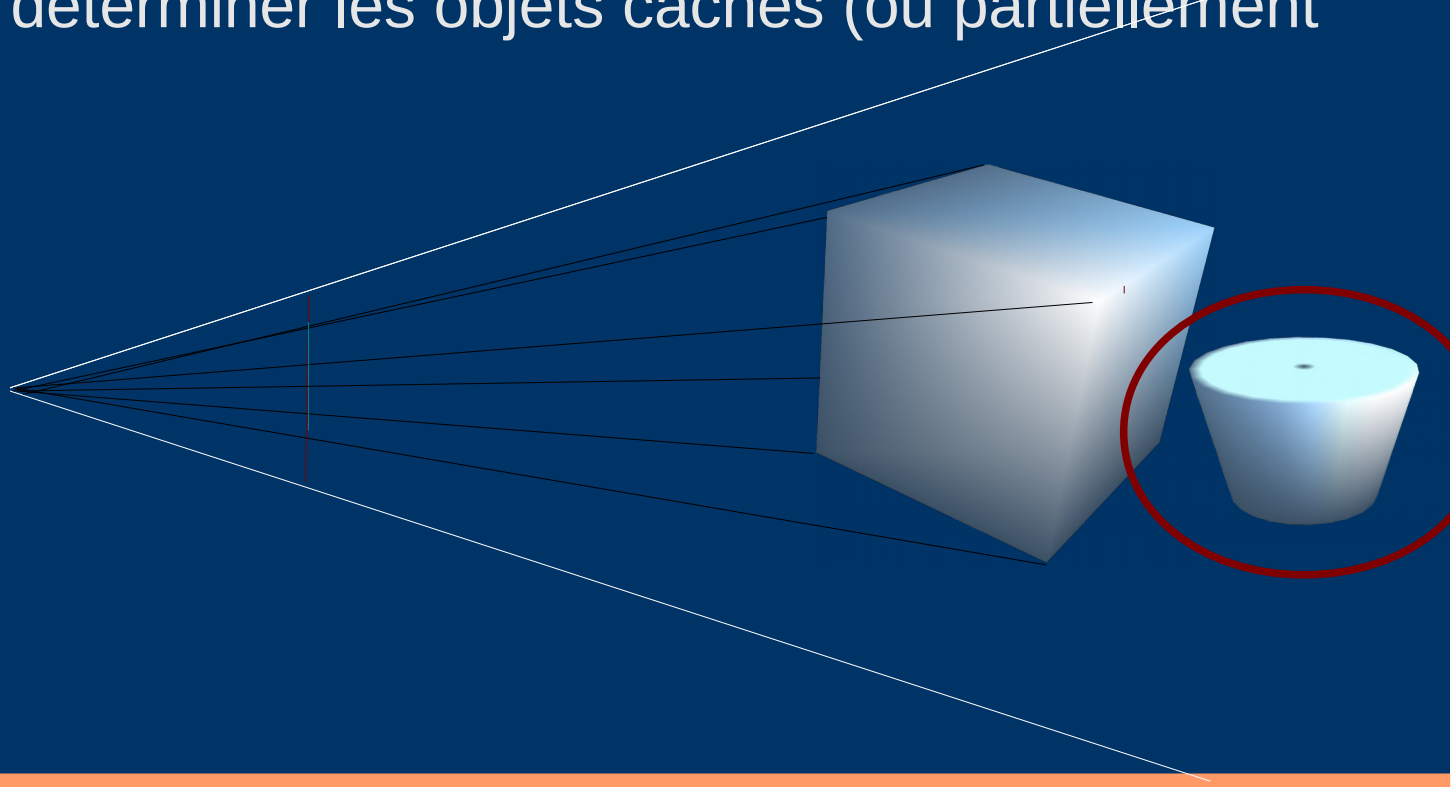
Algorithmes 3D : Clipping/Backface culling

- Préparation de la projection
  - Problèmes :
    - Comment déterminer les sommets non visibles ?



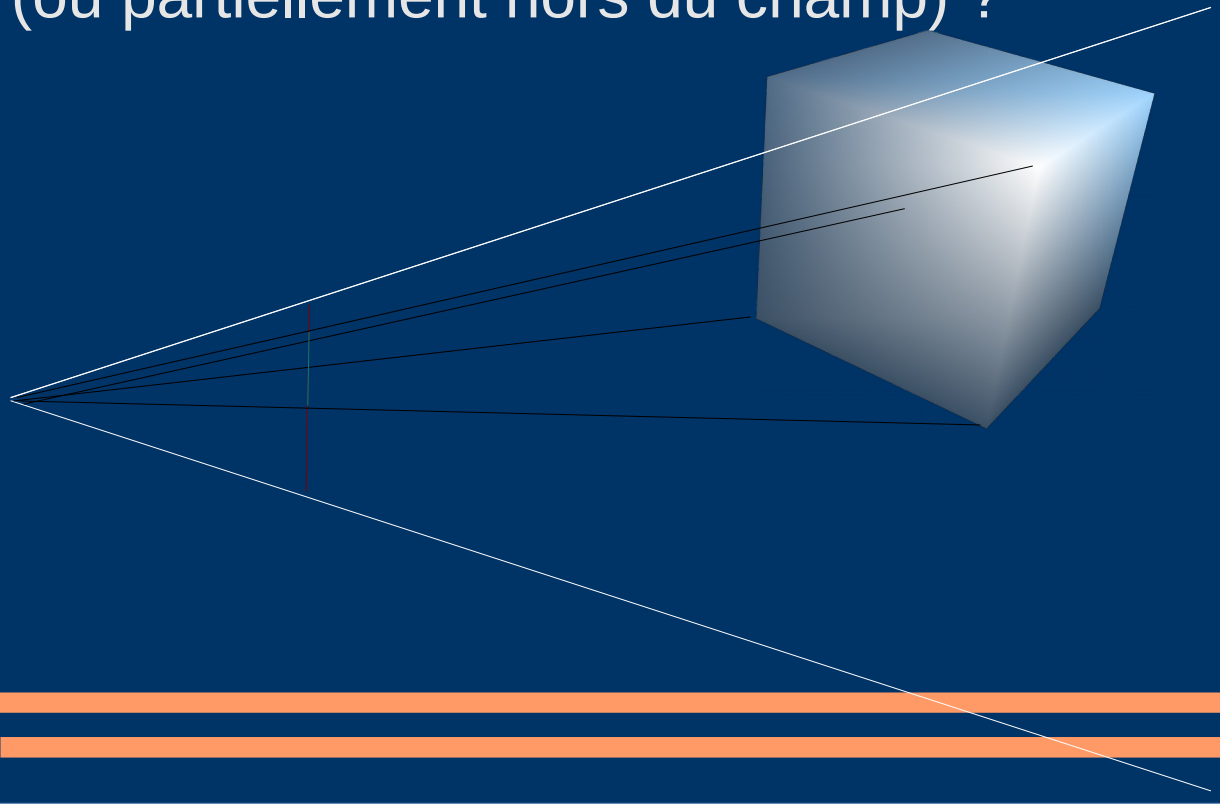
Algorithmes 3D : Clipping/Backface culling

- Préparation de la projection
  - Problèmes :
    - Comment déterminer les objets cachés (ou partiellement cachés)



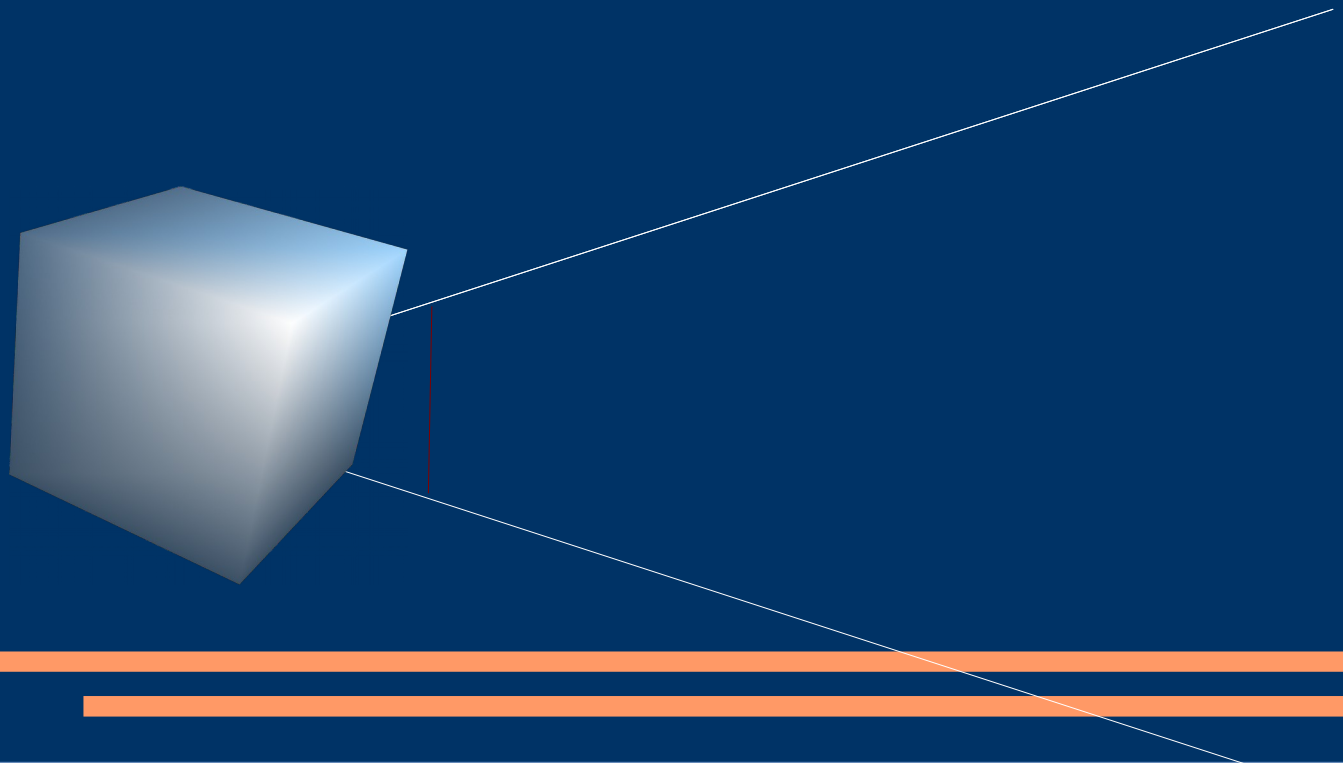
Algorithmes 3D : Clipping/Backface culling

- Préparation de la projection
  - Problèmes :
    - Comment déterminer les objets qui sont hors du champ (ou partiellement hors du champ) ?



Algorithmes 3D : Clipping/Backface culling

- Préparation de la projection
  - Problèmes :
    - Comment déterminer les objets qui sont derrière le plan image ou partiellement visible ?



Algorithmes 3D : Clipping/Backface culling

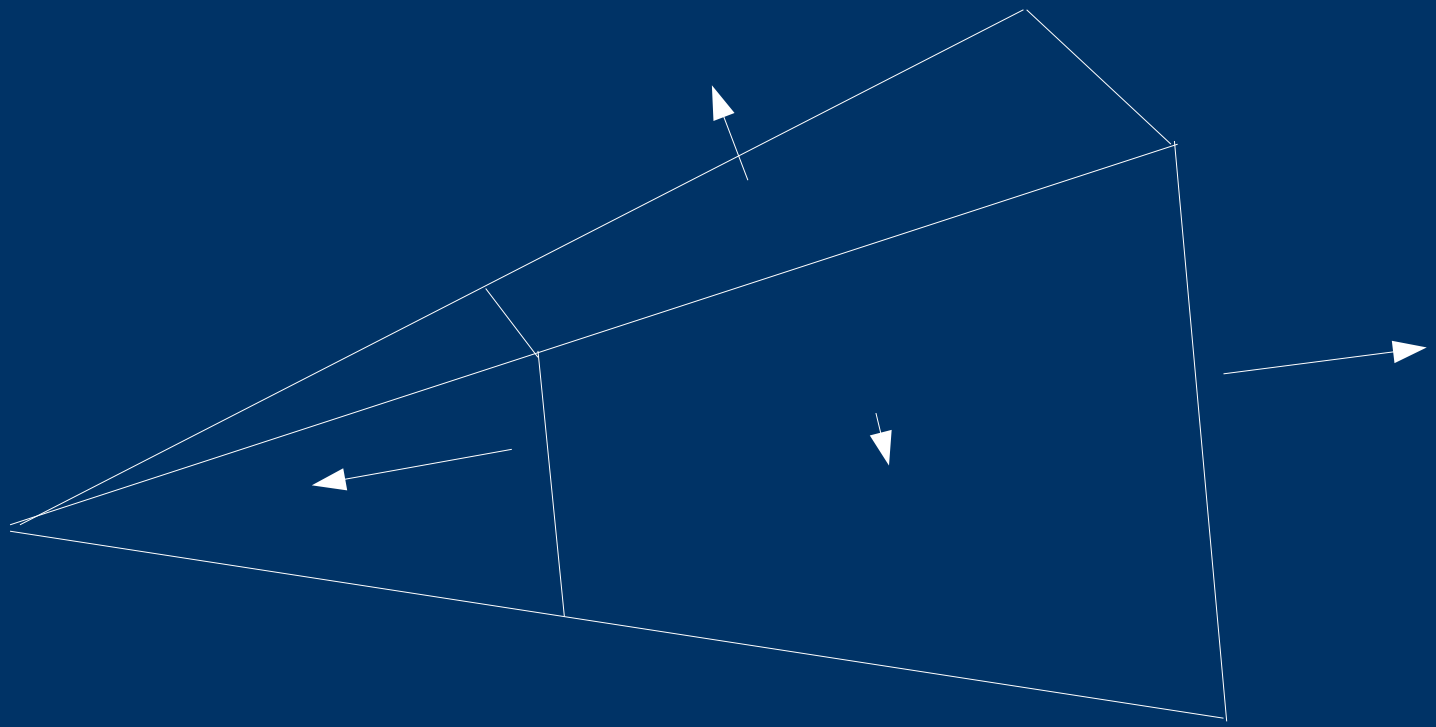
- Problèmes identifiés :
  - Comment déterminer les sommets non visibles ?
  - Comment déterminer les objets cachés (ou partiellement cachés)
  - Comment déterminer les objets qui sont hors du champ (ou partiellement hors du champ) ?
  - Comment déterminer les objets qui sont derrière le plan image ou partiellement visible ?

- Il faut résoudre ces problèmes afin
  - d'avoir une projection correcte
  - d'être efficace

Algorithmes 3D : Clipping 3D

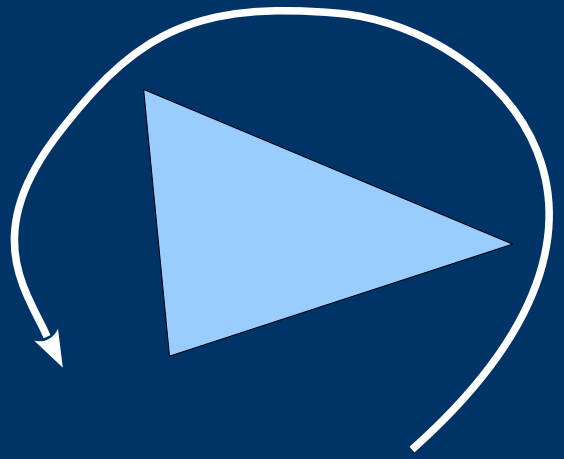
- Comment déterminer les objets qui sont hors du champ (ou partiellement hors du champ) ?
- Comment déterminer les objets qui sont derrière le plan image ou partiellement visible ?

Algorithmes 3D : Clipping 3D



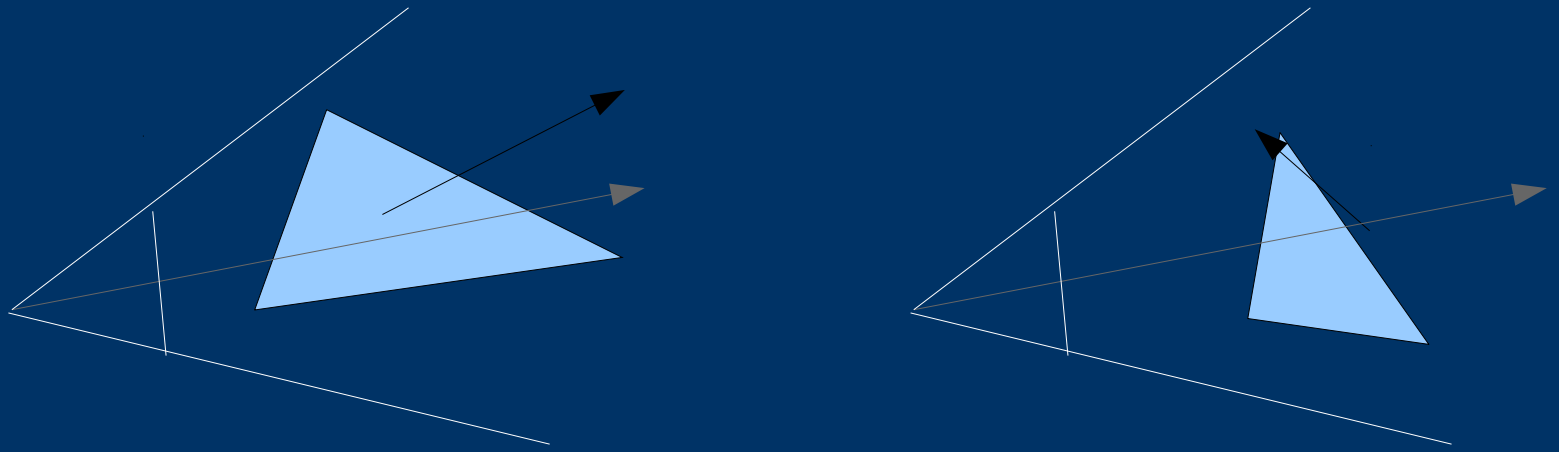
Algorithmes 3D : Backface culling

- Comment déterminer les sommets non visibles ?
  - 1. énumérer les sommets toujours dans le même sens



Algorithmes 3D : Backface culling

- Comment déterminer les sommets non visibles ?
  - 2. Déterminer l'orientation de la face par rapport à l'axe optique
    - Calculer le vecteur normal à la surface (produit vectoriel)
    - Déterminer l'angle entre le vecteur normal à la surface et le vecteur directeur de l'axe optique (produit scalaire)



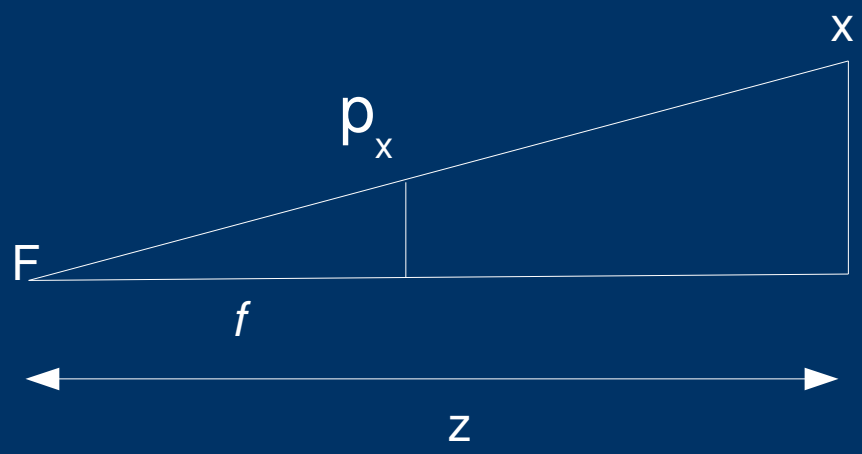
Algorithmes 3D fondamentaux

- Comment déterminer les objets cachés (ou partiellement cachés)
  - Sauvegarder la profondeur pour chaque pixel dessiné

  - Avantage
    - simple
  - Inconvénient
    - Oblige à projeter tout les polygones
    - Problème de résolution lors de l'encodage du Z

Algorithmes 3D fondamentaux

- Projection
  - Une fois que l'on a éliminé les éléments hors du champ de la caméra, les éléments qui ne sont pas de face
    - On projette les sommets et on dessine le polygone en tenant compte de la profondeur



$$P_x = f\,x / z$$

Rendu temps réel

Algorithmes 2D fondamentaux

Algorithmes 2D fondamentaux

- Suivant la projection des polygones, il faut dessiner/remplir le polygone
  - déterminer si une partie n'est pas visible
  - déterminer la couleur et l'éclairage
    - éventuellement plaquer une texture
  - ...
- Les données sont la liste des sommets

Algorithmes 2D : Remplissage de polygones

- Type de polygones : Tirangle -> Convexe
- Données : liste de sommets

Algorithmes 2D : Remplissage de polygones

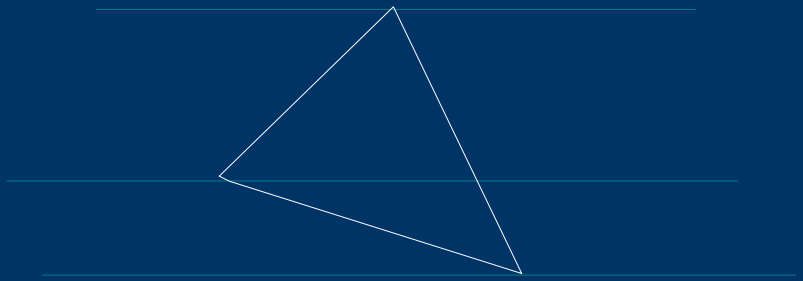
- Données : liste de sommets
- Algo :
  - Parcourir toutes les arrêtes de haut en bas et remplir horizontalement

Algorithmes 2D : Remplissage de polygones

- Algo :
  - Parcourir toutes les arrêtes de haut en bas et remplir horizontalement
- Problème :
  - Trier les sommets
  - Identifier les arrêtes actives
- Comment suivre les arrêtes actives
  - Tracé de segments

Algorithmes 2D : Remplissage de polygones

- Identifier les arrêtes actives



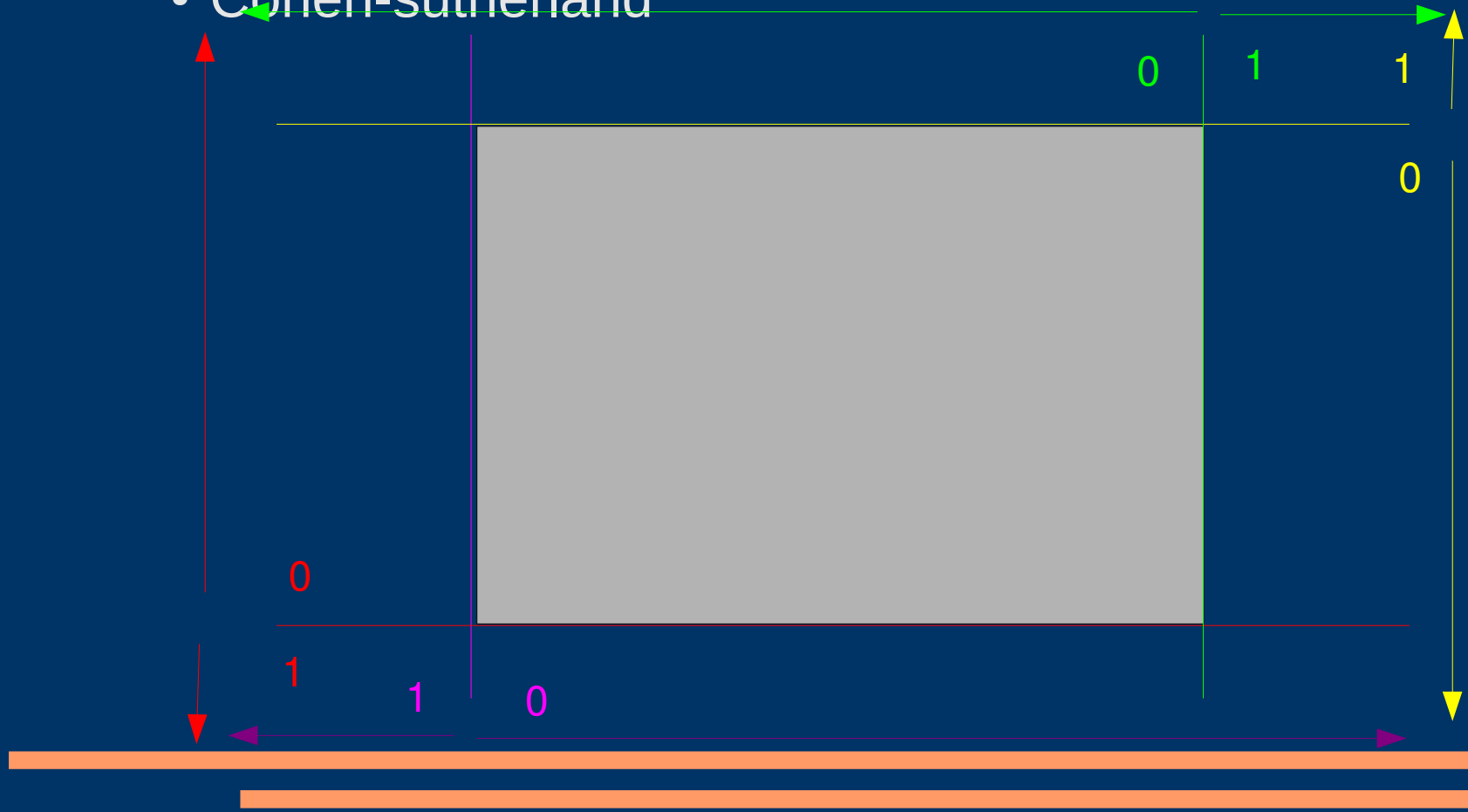
Algorithmes 2D : Clipping

- Même s'il est possible de fenêtrer les polygones dans l'espace, il faut parfois fenêtrer dans le plan

  - Fenêtrage rectangulaire de segments :
    - Cohen-sutherland
  - Fenêtrage d'un polygone à partir des segments :
    - Weiler – Atherton

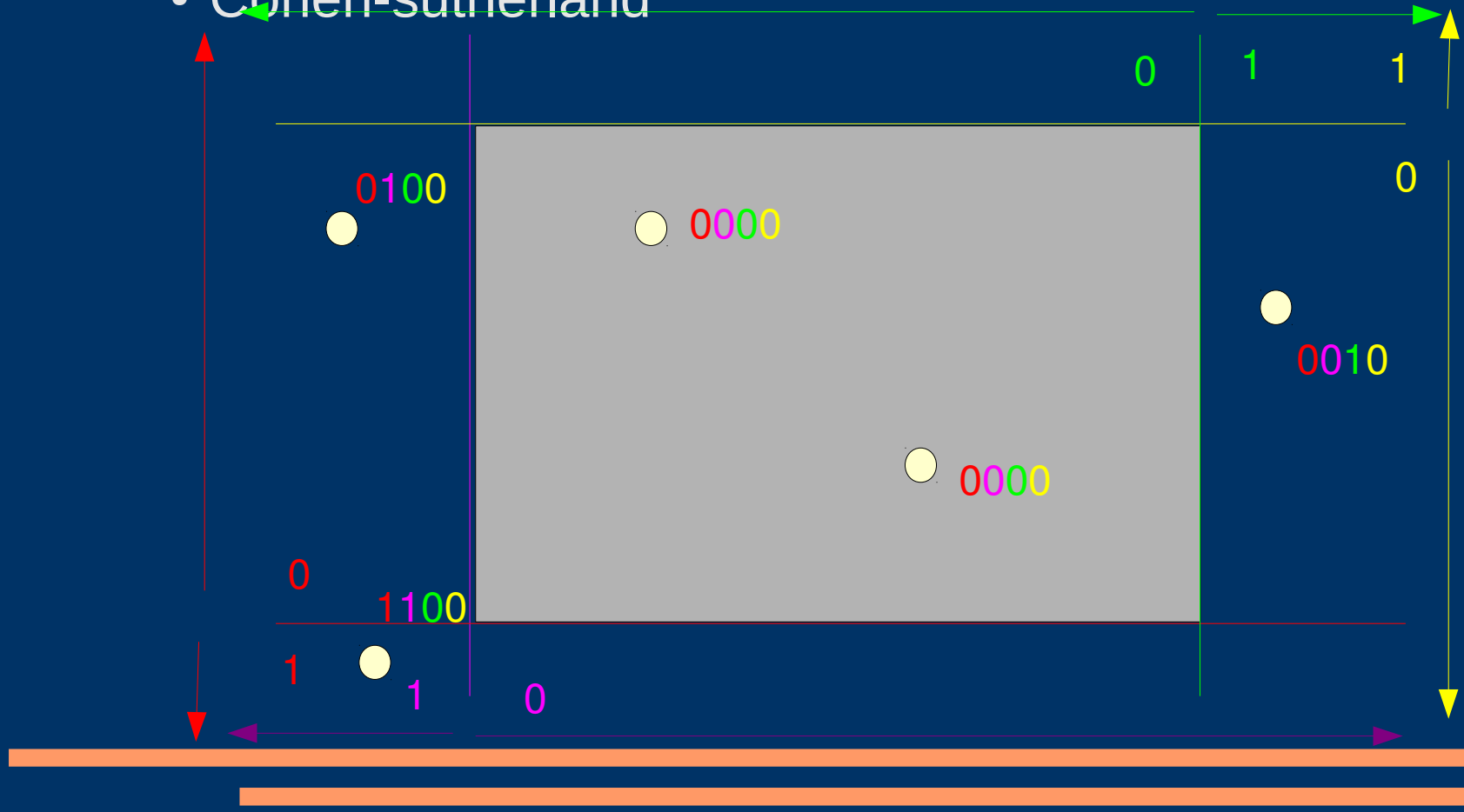
Algorithmes 2D : Clipping

- Fenêtrage rectangulaire de segments :
  - Cohen-sutherland



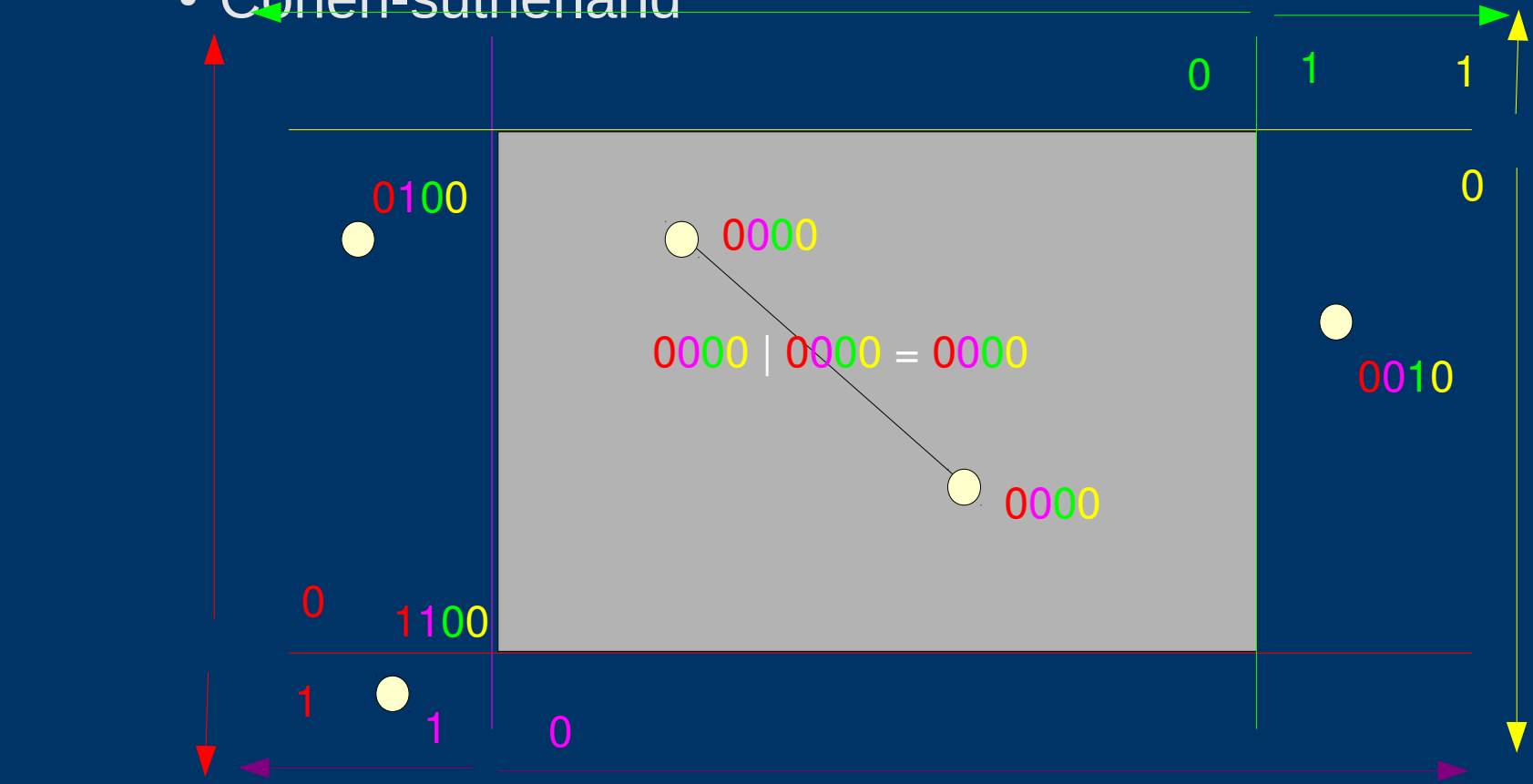
Algorithmes 2D : Clipping

- Fenêtrage rectangulaire de segments :
  - Cohen-sutherland



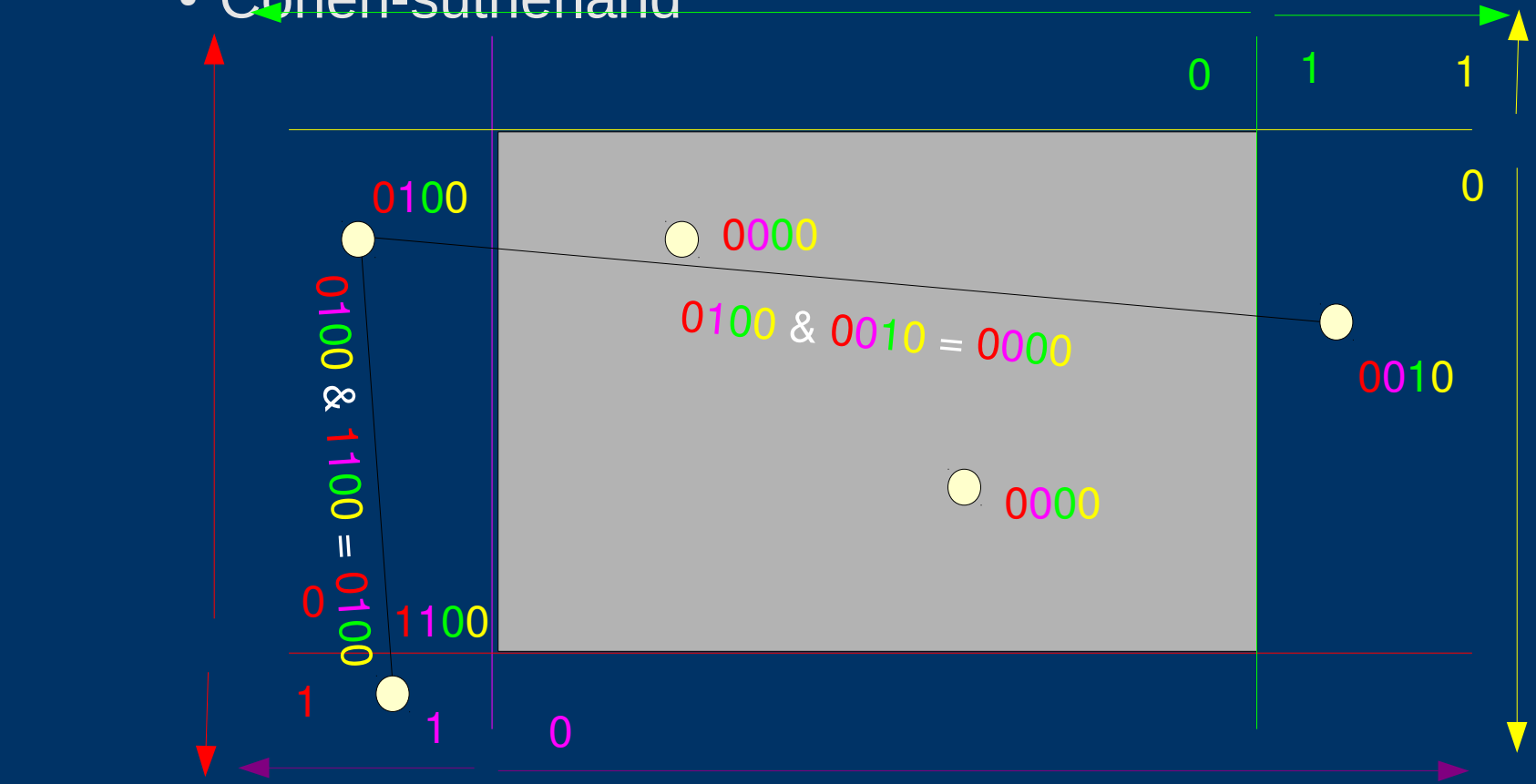
Algorithmes 2D : Clipping

- Fenêtrage rectangulaire de segments :
  - Cohen-sutherland



Algorithmes 2D : Clipping

- Fenêtrage rectangulaire de segments :
  - Cohen-sutherland

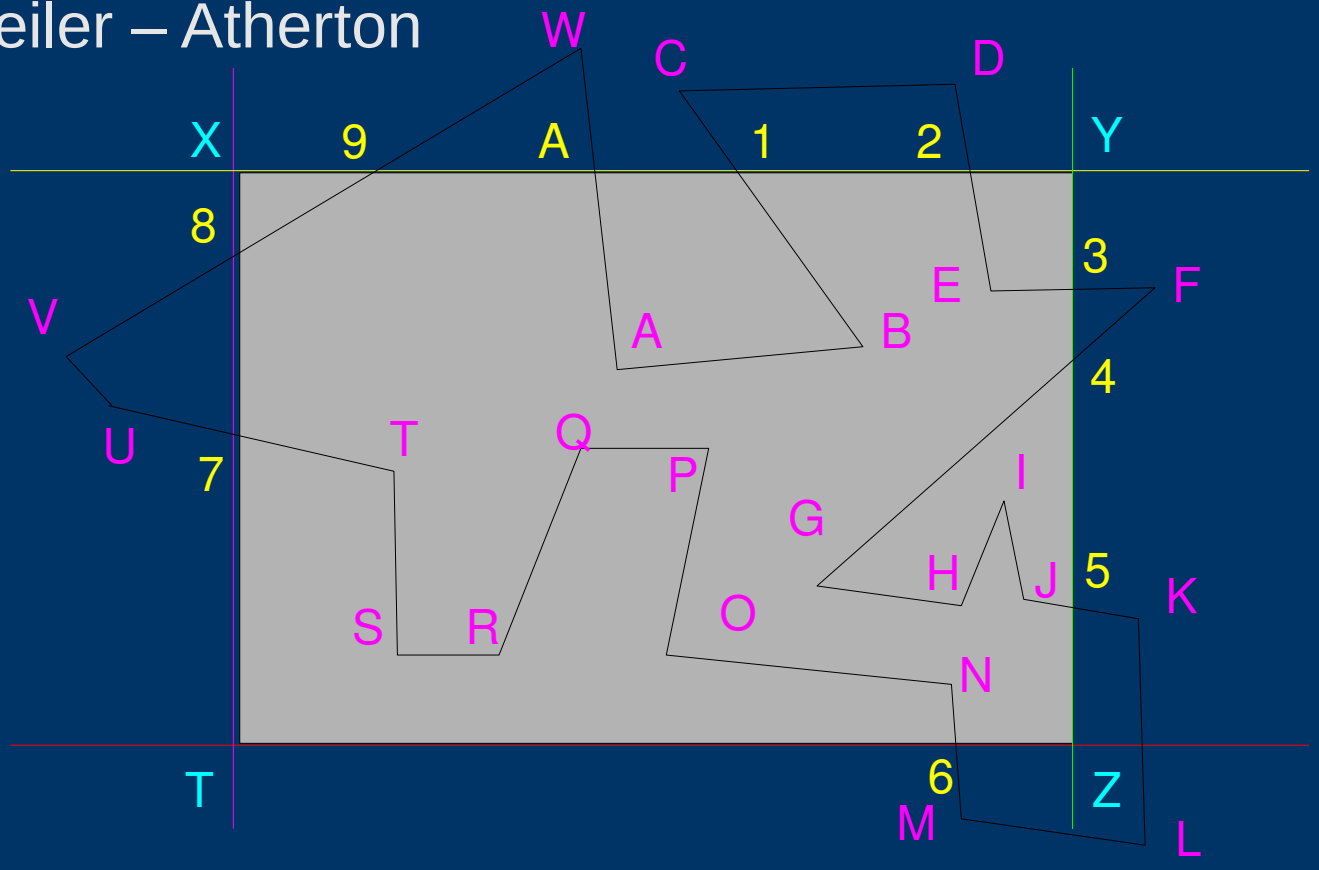


Algorithmes 2D : Clipping

- Fenêtrage rectangulaire de segments – reconstitution du polygone...
  - Weiler – Atherton

Algorithmes 2D : Clipping

- Fenêtrage rectangulaire de segments :
  - Weiler – Atherton

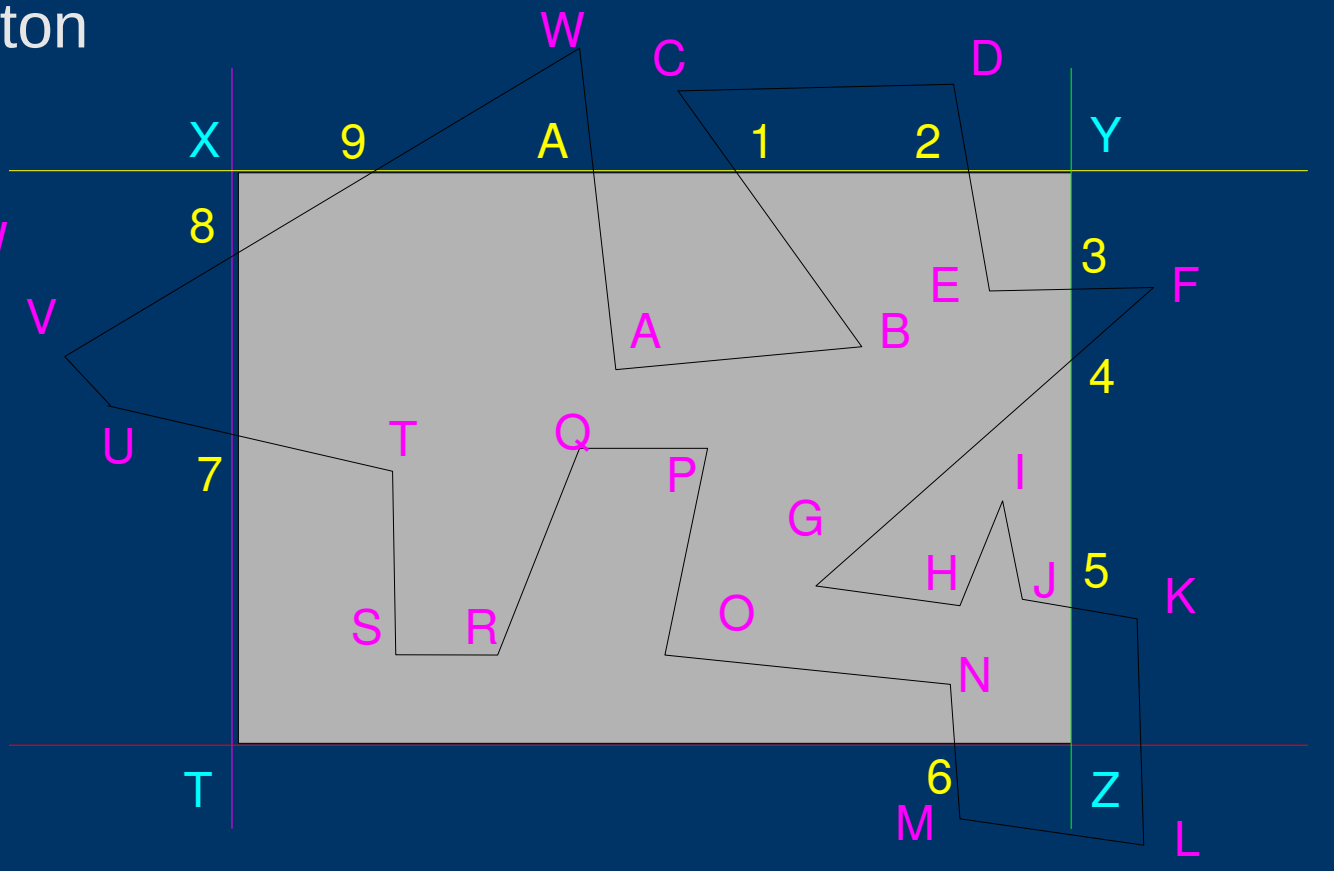


Algorithmes 2D : Clipping

- Fenêtrage rectangulaire de segments :
  - Weiler – Atherton

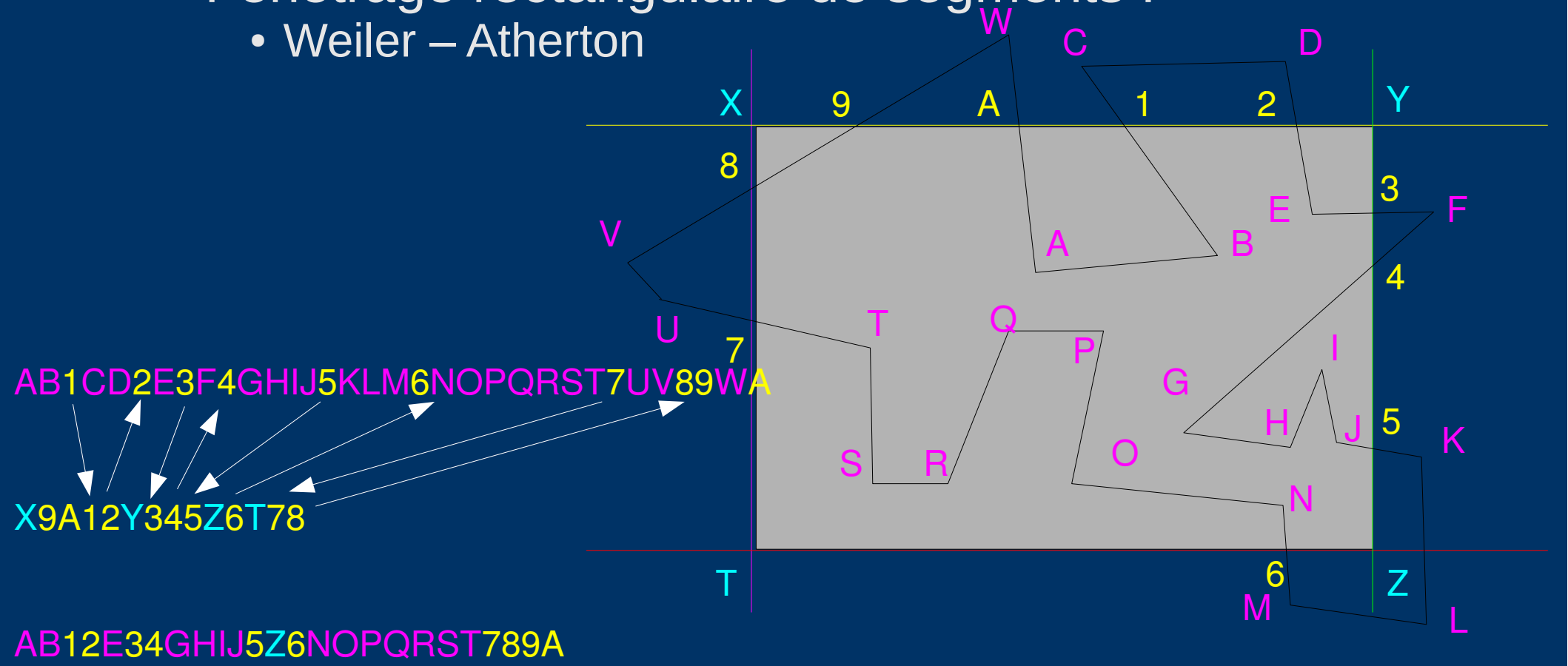
ABCDEFGHIJKLMNOPQRSTUVW

XYZT



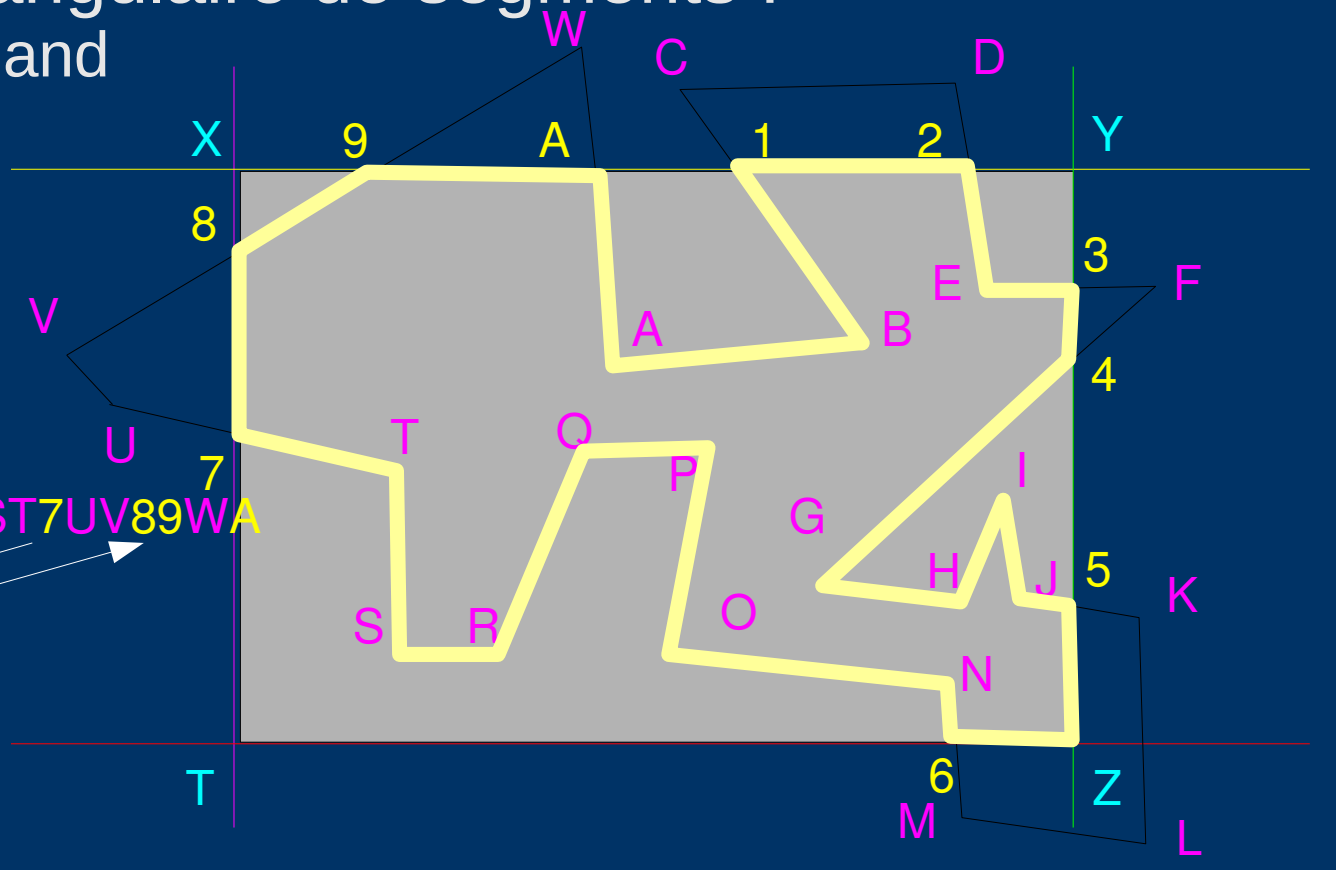
Algorithmes 2D : Clipping

- Fenêtrage rectangulaire de segments :
  - Weiler – Atherton



Algorithmes 2D : Clipping

- Fenêtrage rectangulaire de segments :
  - Cohen-sutherland



AB1CD2E3F4GHIJ5KLM6NOPQRST7UV89WA

X9A12Y345Z6T78

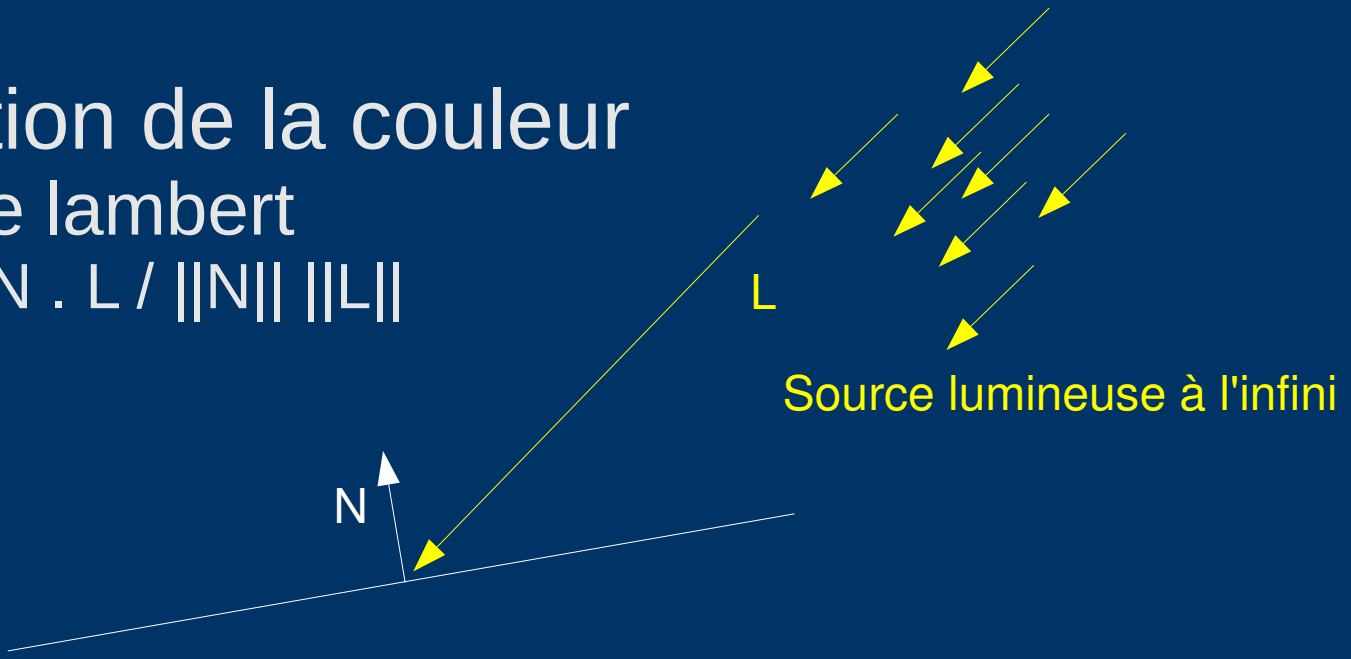
AB12E34GHIJ5Z6NOPQRST789A

Algorithmes 2D : Détermination de la couleur des pixels durant le remplissage

- Déterminer la couleur des pixels pendant le remplissage en fonction de :
  - La couleur de la face ?
  - L'éclairage ?
  - ...

Eclairage

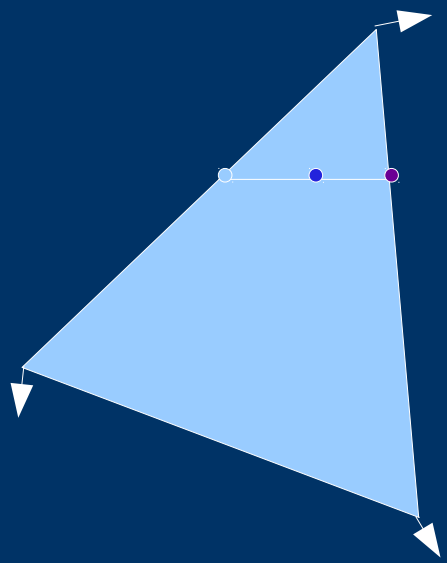
- Determination de la couleur
  - Modèle de lambert
    - Id = k * N . L / ||N|| ||L||



  - 1 couleur par face
  - Indépendant du point de vu

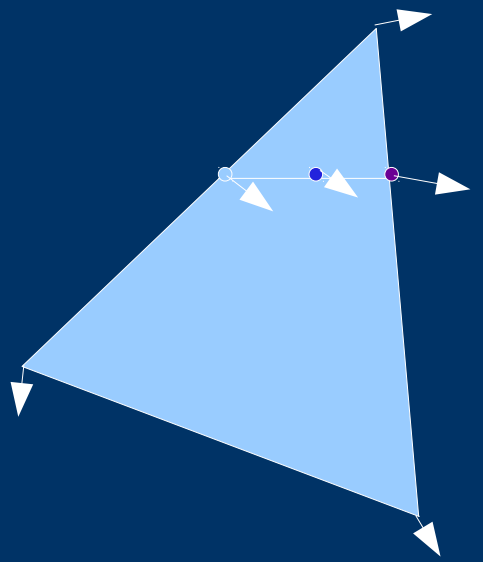
Eclairage

- Determination de la couleur
  - Modèle de Gouraud



Eclairage

- Determination de la couleur
  - Modèle de Phong



Eclairage

Génération des images

- THE END !