Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Moteurs Temps Réel
Optimisations de l'affichage

Guillaume Gilet
Guillaume.Gilet@unilim.fr

2012-2013

master ISICG limoges

Objectif : Temps réel

Temps réel

- Temps interactif : $\approx 2 - 10fps$
- Temps réel : $\approx 30fps$

Jusqu'ici on a vu

- Des modèles complexes
- Des apparences variées
- De l'illumination
- Utile pour une scène de petite taille

En général, on a souvent

- Modèles Nombreux
- Nombre de primitives élevé
- Changement d'états coûteux
- Calculs CPU (animation, collisions...)

Différentes accélérations

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Besoin d'optimisations

- Limiter le nombre d'objets, de primitives
- Limiter les changements d'états
- Éviter les calculs inutiles
- Réutiliser, pré-calculer...

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Différentes accélérations

Besoin d'optimisations

- Limiter le nombre d'objets, de primitives
- Limiter les changements d'états
- Éviter les calculs inutiles
- Réutiliser, pré-calculer...

Traitement au niveau de la scène

- Position et orientation de la caméra
- Visibilité
- *Culling*

Traitement de la géométrie

- Au niveau des primitives
- Niveau de détails

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Visibilité et *culling*

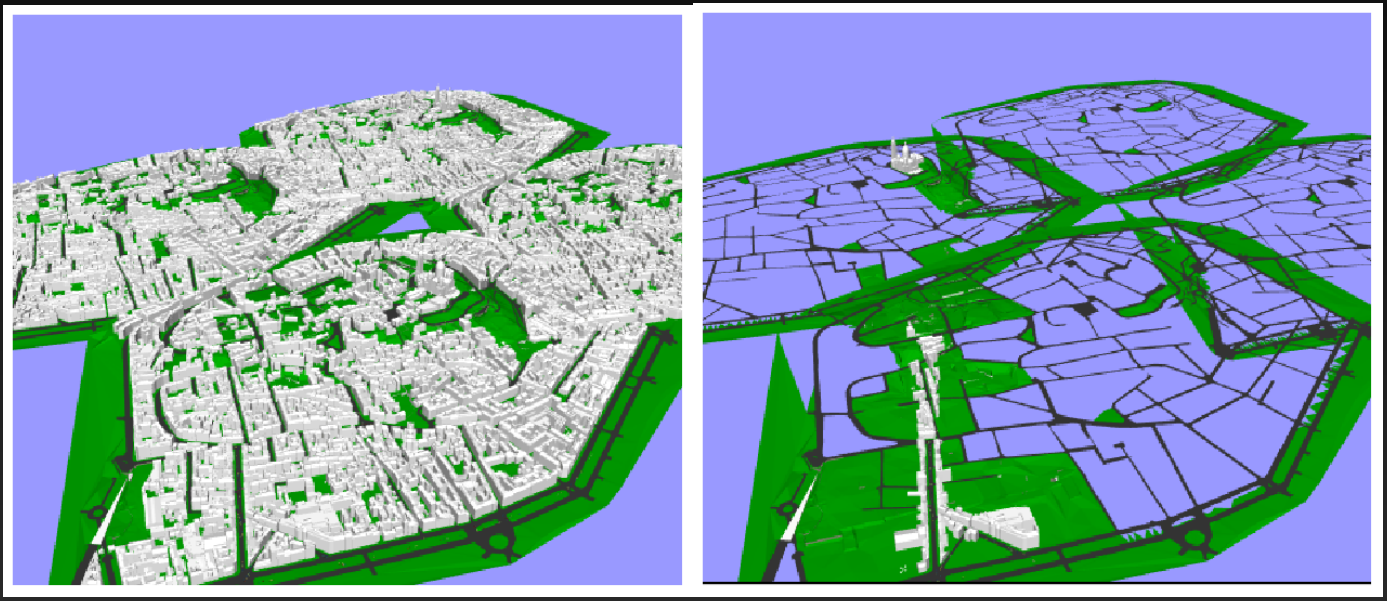
" *Couvrez ce sein que je ne saurais voir.* " (Tartuffe)

Traitement de la scène

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Traitement au niveau de la scène

- Pour un point de vue donné que voit-on ?



- Traitement indispensable pour de grands environnements *libres*
  - Assassin's Creed
  - The Elder Scrolls

Traitement au niveau de la scène

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Que ne voit-on pas ?

- Ce qui est en dehors du volume de vue

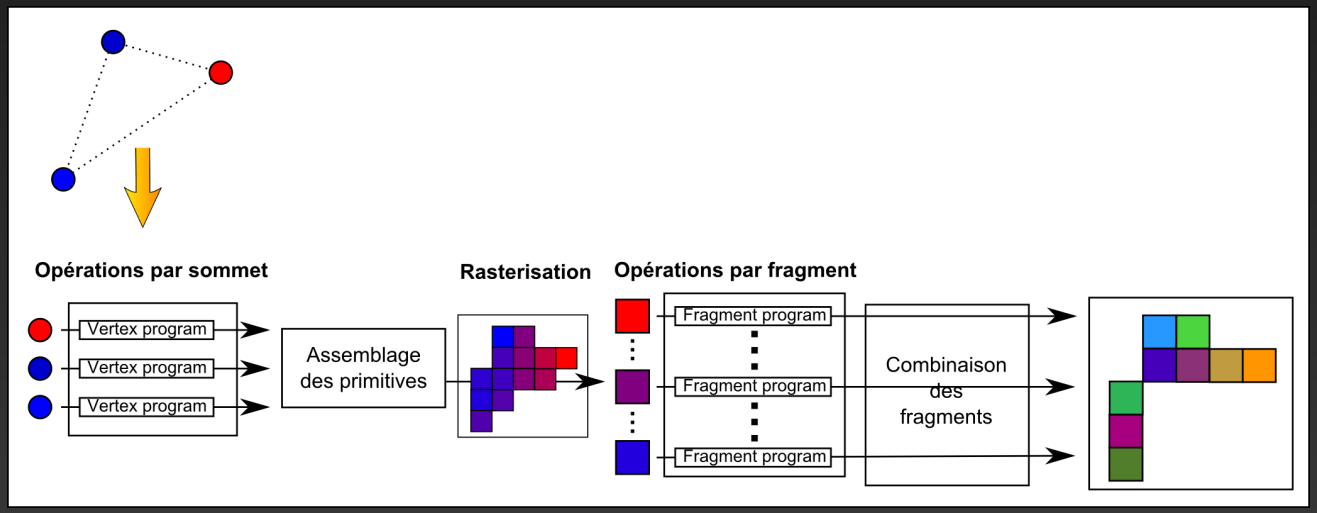
Traitement au niveau de la scène

Que ne voit-on pas ?

- Ce qui est en dehors du volume de vue

Culling

- Avec le pipeline graphique



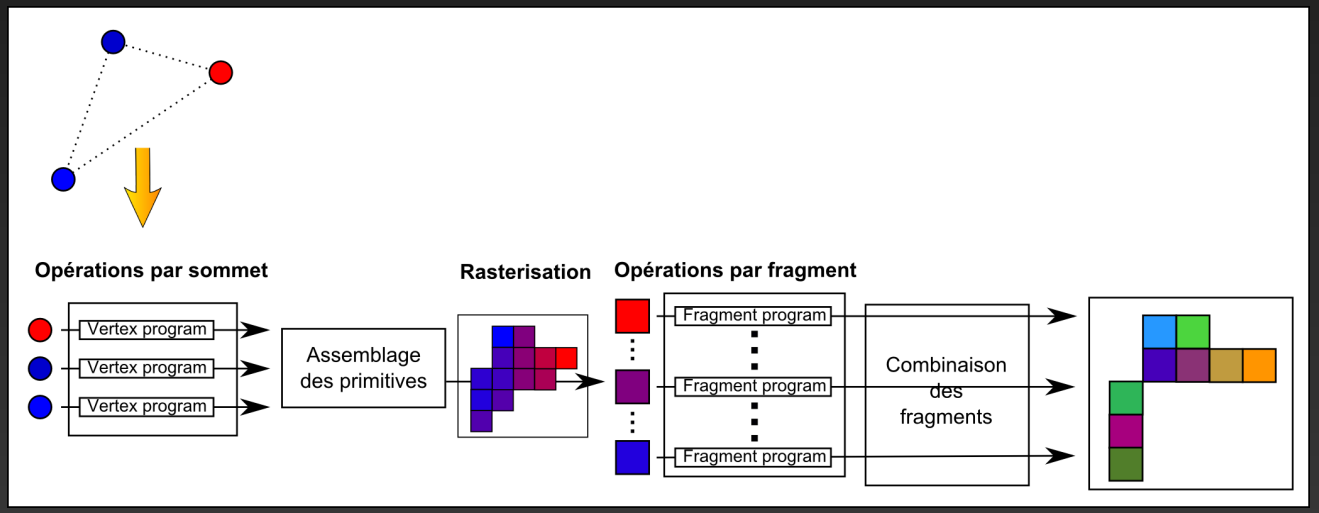
Traitement au niveau de la scène

Que ne voit-on pas ?

- Ce qui est en dehors du volume de vue Géometrie projetée

Culling

- Avec le pipeline graphique



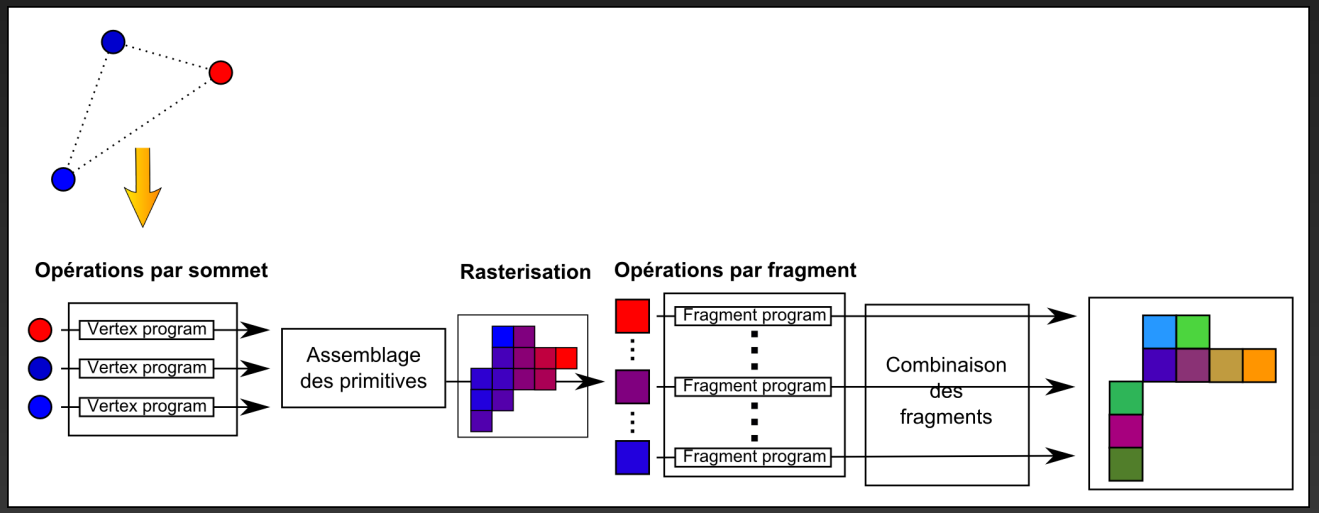
Traitement au niveau de la scène

Que ne voit-on pas ?

- Ce qui est en dehors du volume de vue Géometrie projetée
- Ce qui est caché par un objet

Culling

- Avec le pipeline graphique



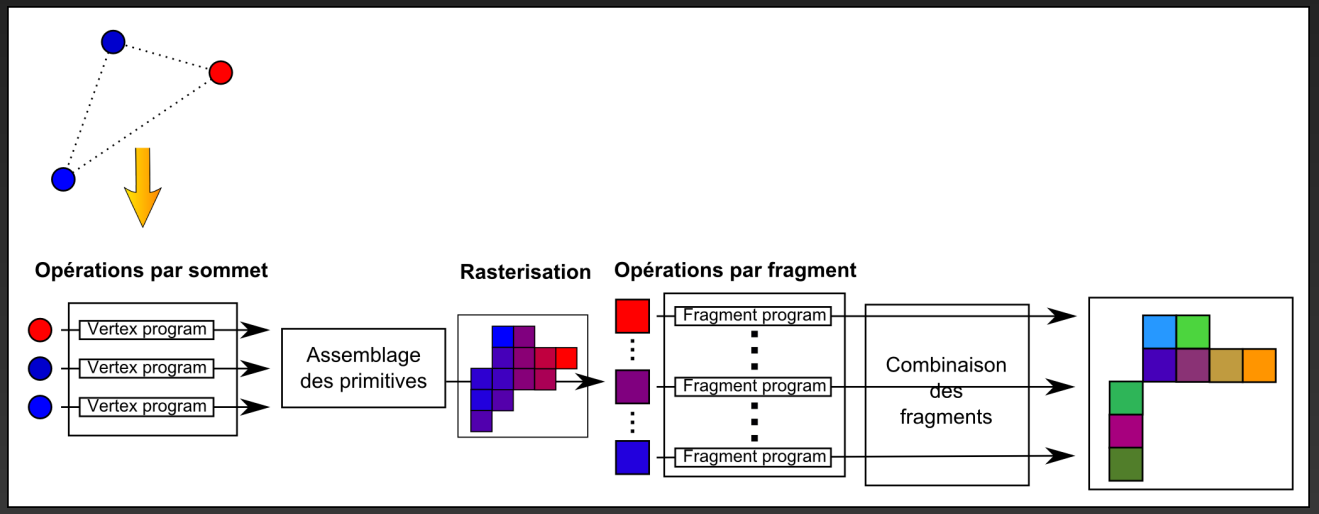
Traitement au niveau de la scène

Que ne voit-on pas ?

- Ce qui est en dehors du volume de vue Géometrie projetée
- Ce qui est caché par un objet Géometrie projetée et fragments traités

Culling

- Avec le pipeline graphique



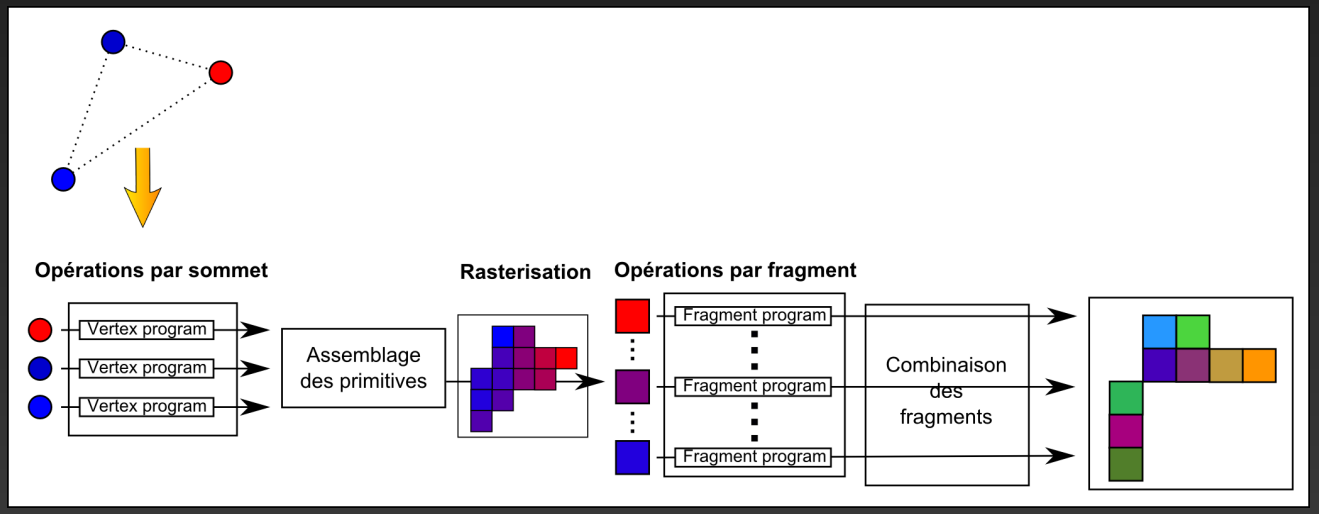
Traitement au niveau de la scène

Que ne voit-on pas ?

- Ce qui est en dehors du volume de vue Géometrie projetée
- Ce qui est caché par un objet Géometrie projetée et fragments traités
- Ce qui est plus petit qu'un pixel

Culling

- Avec le pipeline graphique



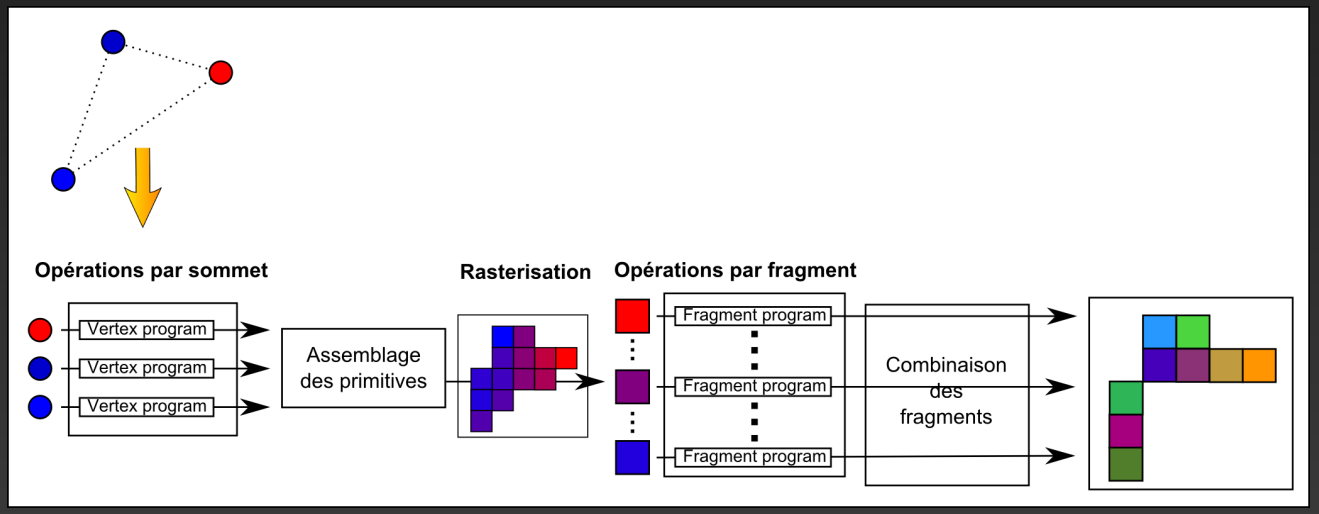
Traitement au niveau de la scène

Que ne voit-on pas ?

- Ce qui est en dehors du volume de vue Géometrie projetée
- Ce qui est caché par un objet Géometrie projetée et fragments traités
- Ce qui est plus petit qu'un pixel Probablement des calculs inutiles

Culling

- Avec le pipeline graphique



Traitement au niveau de la scène

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Que ne voit-on pas ?

- Ce qui est en dehors du volume de vue Géometrie projetée
- Ce qui est caché par un objet Géometrie projetée et fragments traités
- Ce qui est plus petit qu'un pixel Probablement des calculs inutiles

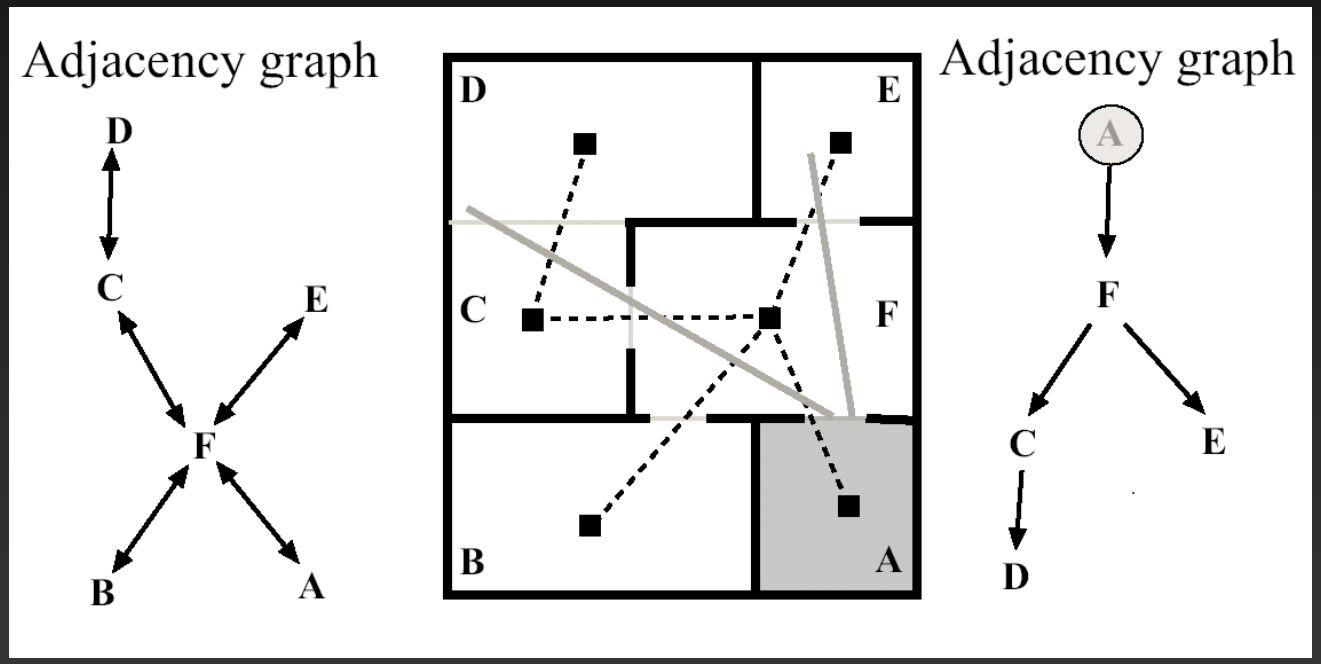
Eliminer les objets le plus tôt possible

- Pipeline graphique peu efficace
- Idéalement avant le rendu
- En utilisant le CPU
- Différentes optimisations

Traitement au niveau de la scène

Potentially Visible Set

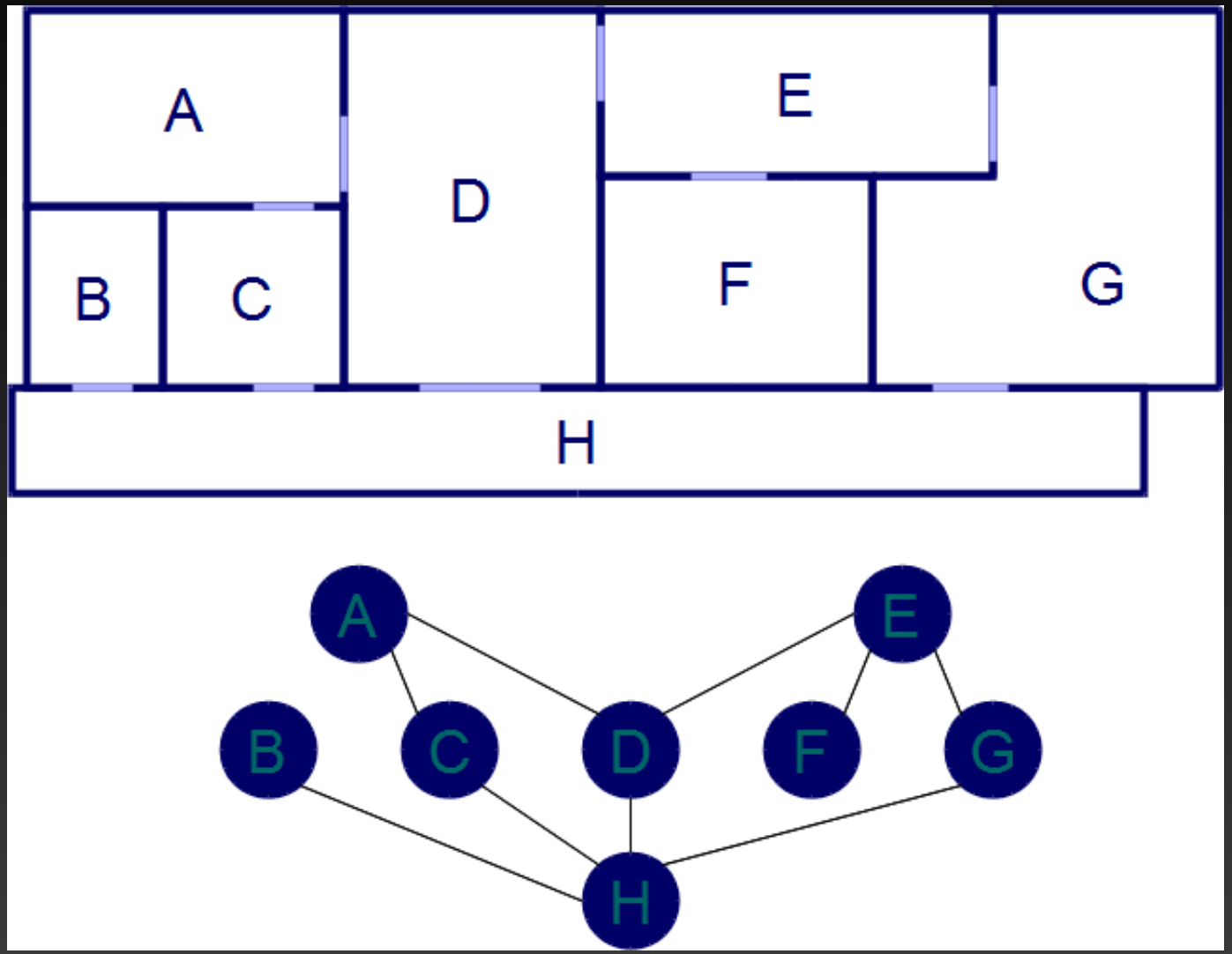
- Idée
  - Diviser le monde en cellule
  - Avec des liaisons (portails) entre cellules
  - Déterminer ce qui est **potentiellement** visible
  - Construire un graphe d'adjacence et de visibilité



Cellules et portails

Déterminer les objets potentiellement visibles

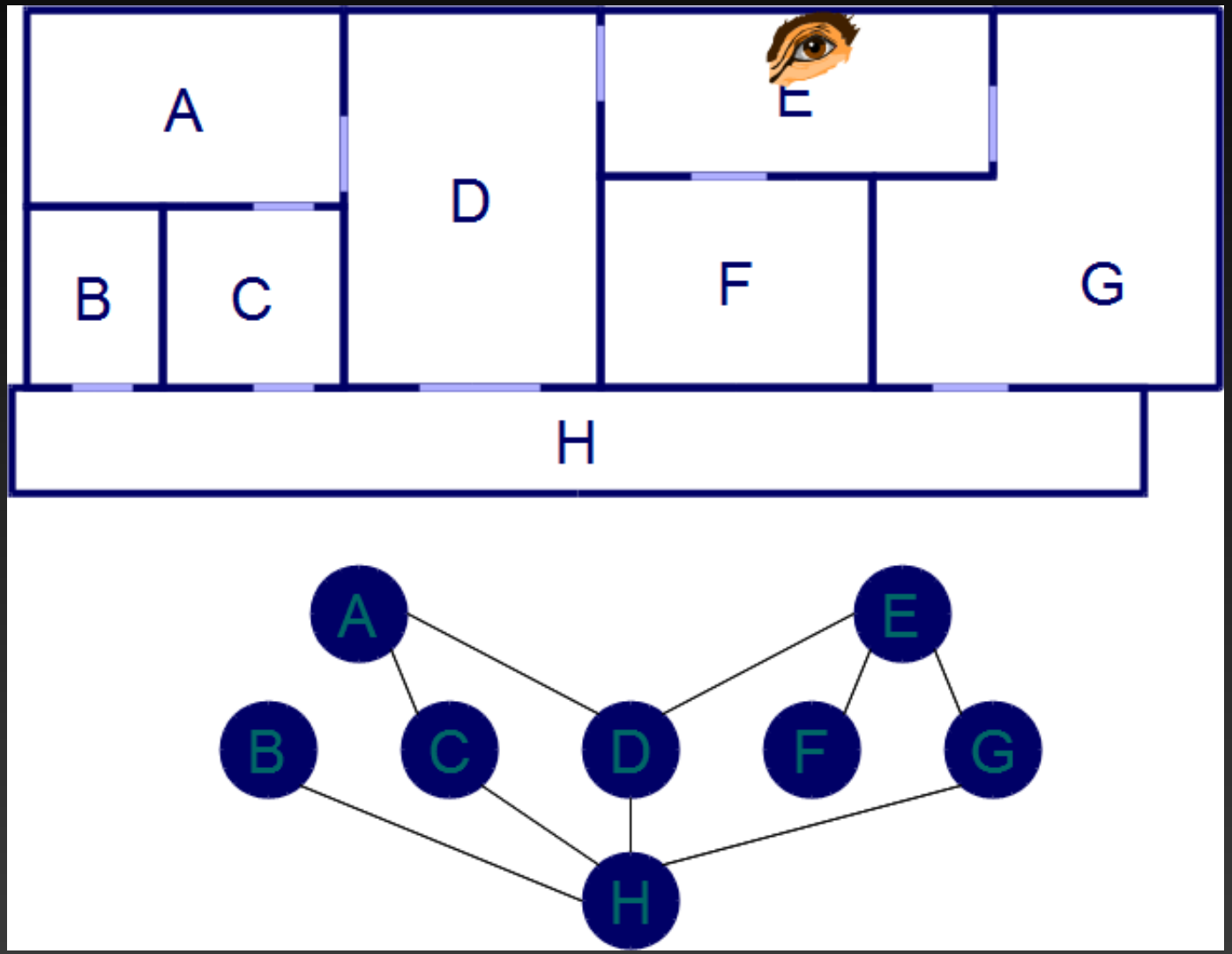
- Parcourir le graphe



Cellules et portails

Déterminer les objets potentiellement visibles

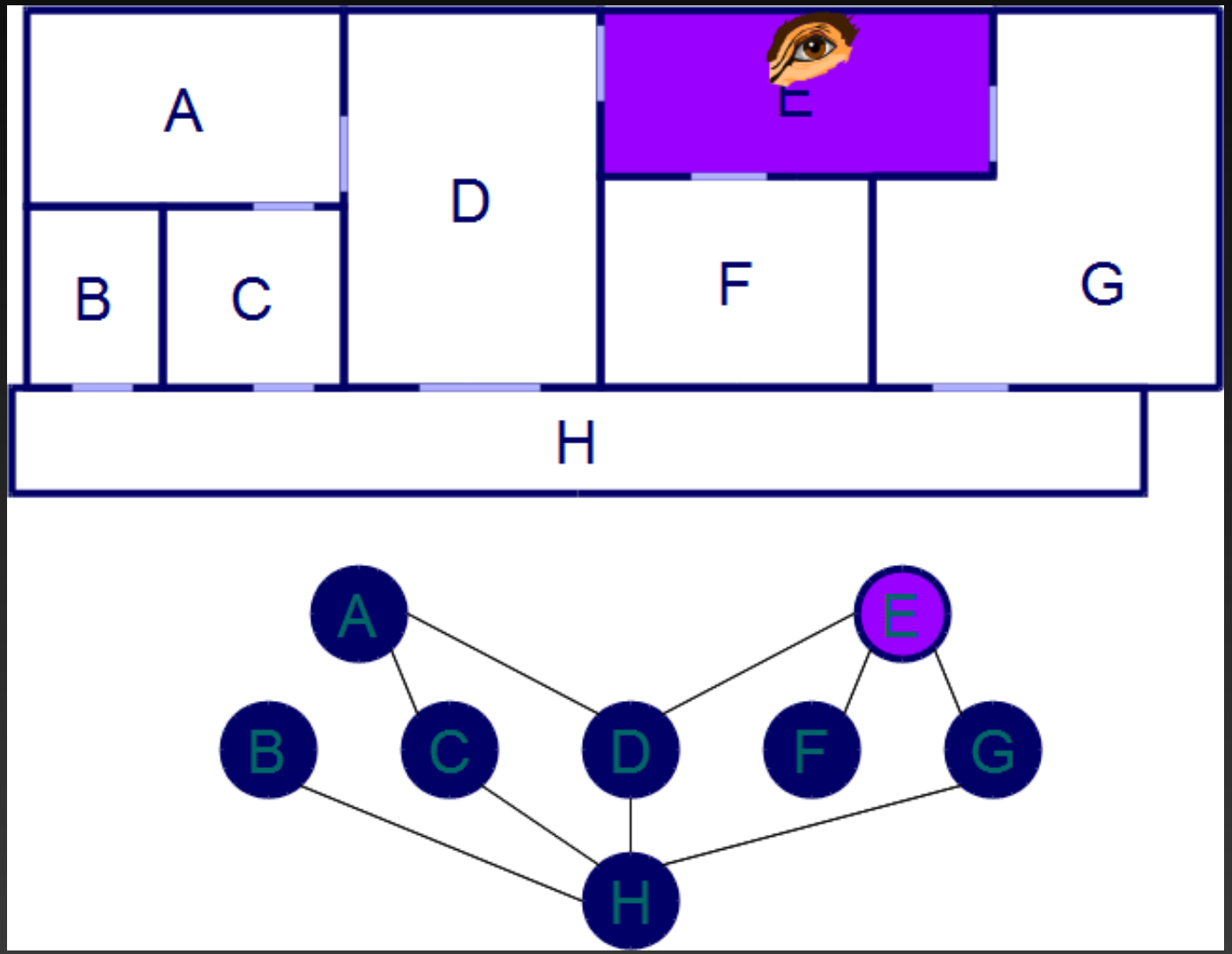
- Parcourir le graphe



Cellules et portails

Déterminer les objets potentiellement visibles

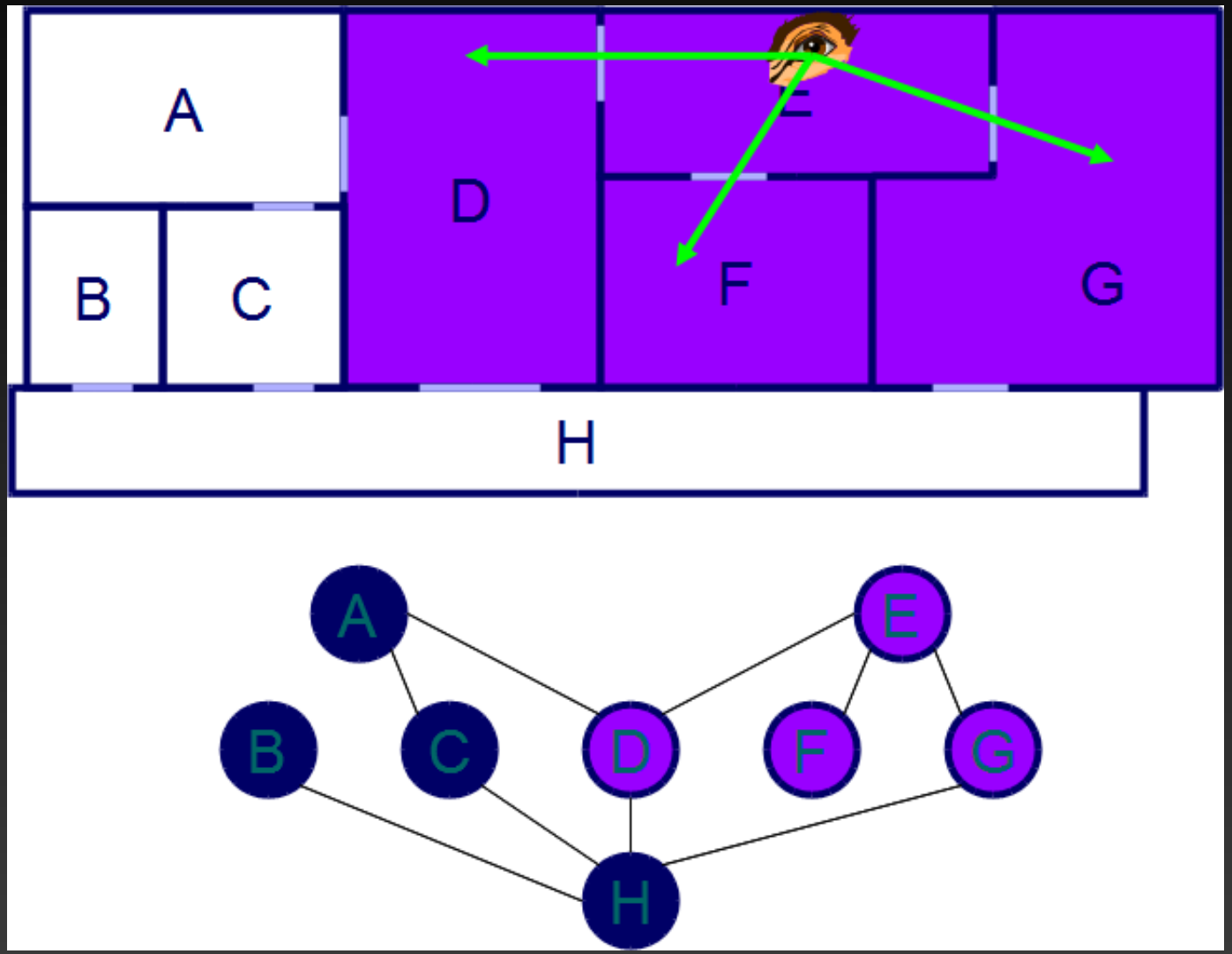
- Parcourir le graphe



Cellules et portails

Déterminer les objets potentiellement visibles

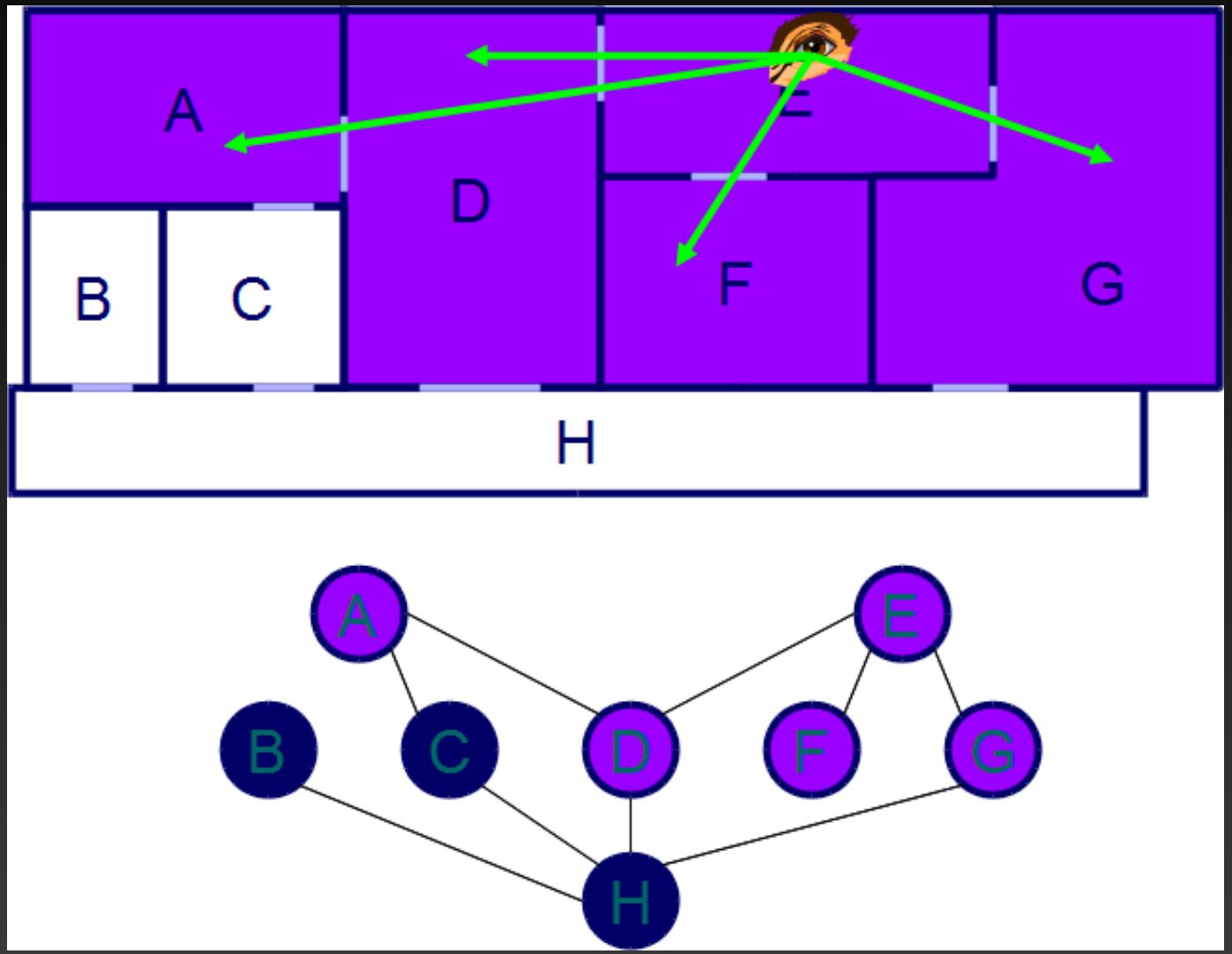
- Parcourir le graphe



Cellules et portails

Déterminer les objets potentiellement visibles

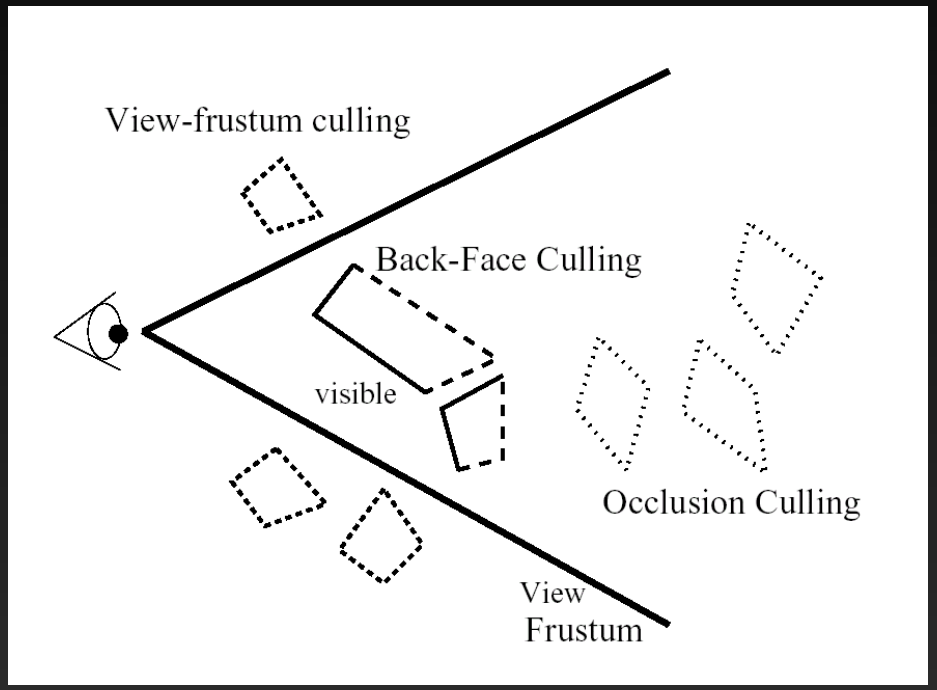
- Parcourir le graphe



Traitement des primitives

Culling

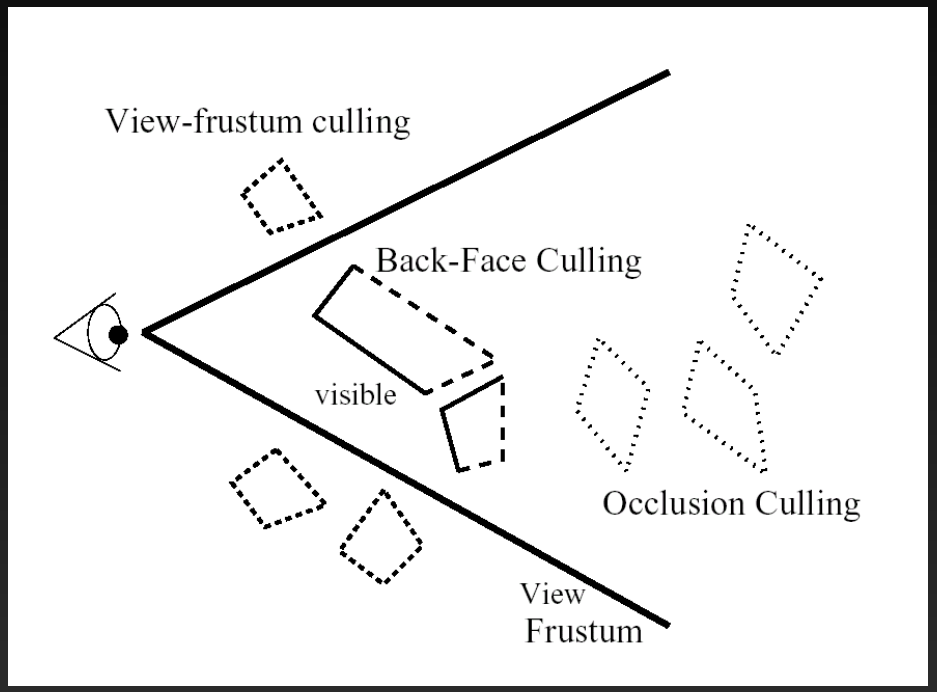
- Différentes raisons d'éliminer une primitive



Traitement des primitives

Culling

- Différentes raisons d'éliminer une primitive

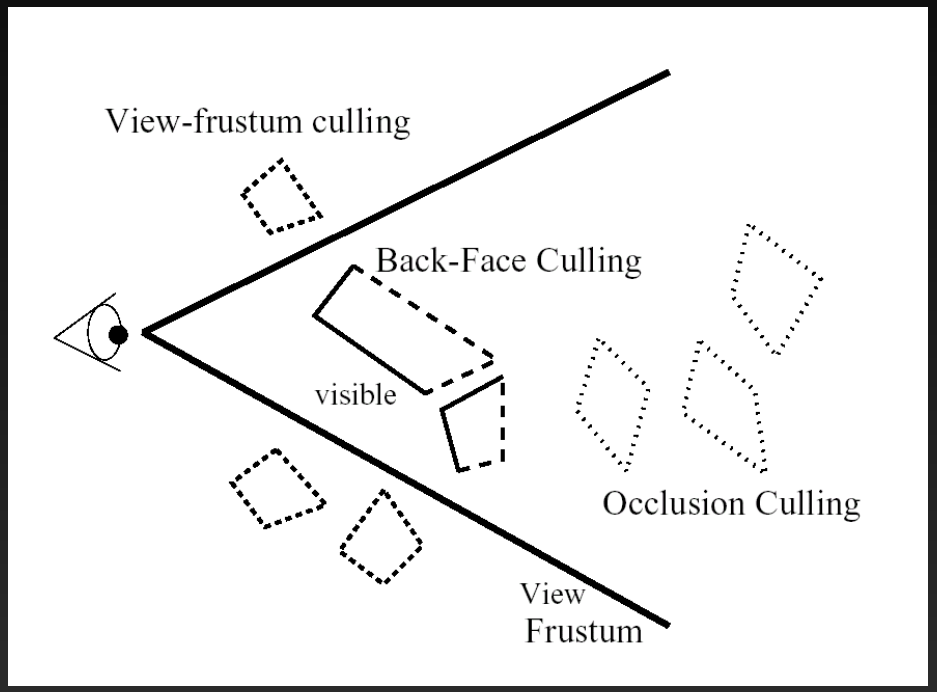


- *BackFace Culling* : Eliminer les primitives (faces) arrières

Traitement des primitives

Culling

- Différentes raisons d'éliminer une primitive

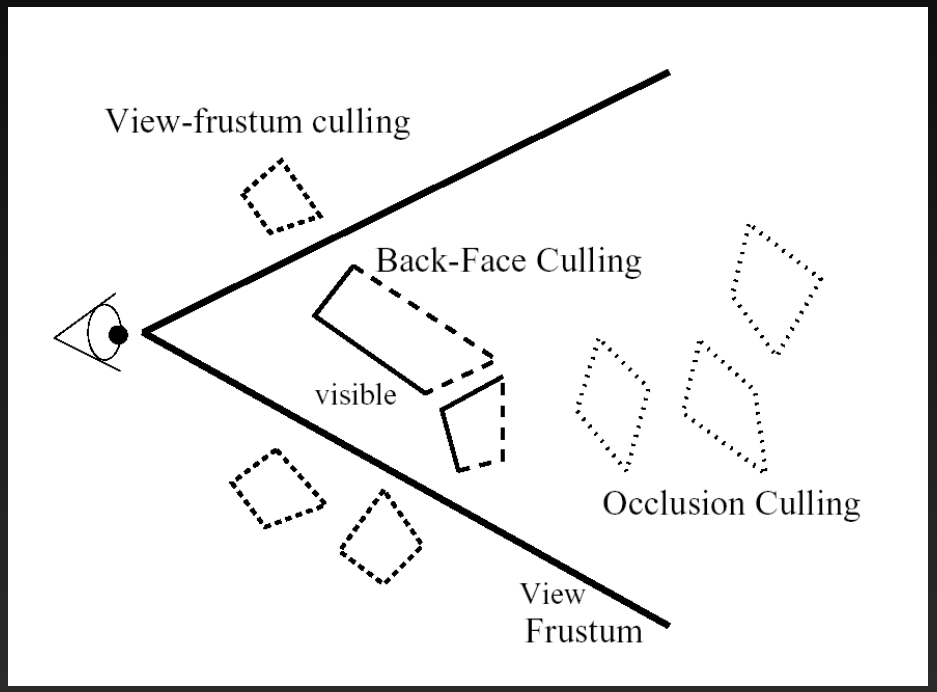


- *BackFace Culling* : Eliminer les primitives (faces) arrières
- *Occlusion Culling* : Eliminer les primitives occludées

Traitement des primitives

Culling

- Différentes raisons d'éliminer une primitive



- *BackFace Culling* : Eliminer les primitives (faces) arrières
- *Occlusion Culling* : Eliminer les primitives occludées
- *Frustum Culling* : Eliminer les primitives hors du volume de vue

Occlusion Culling

Occlusion Culling

- Qui cache qui ?
- Approche globale (toutes les primitives de la scène)
- On veut être conservatif (dans le doute un objet est visible)

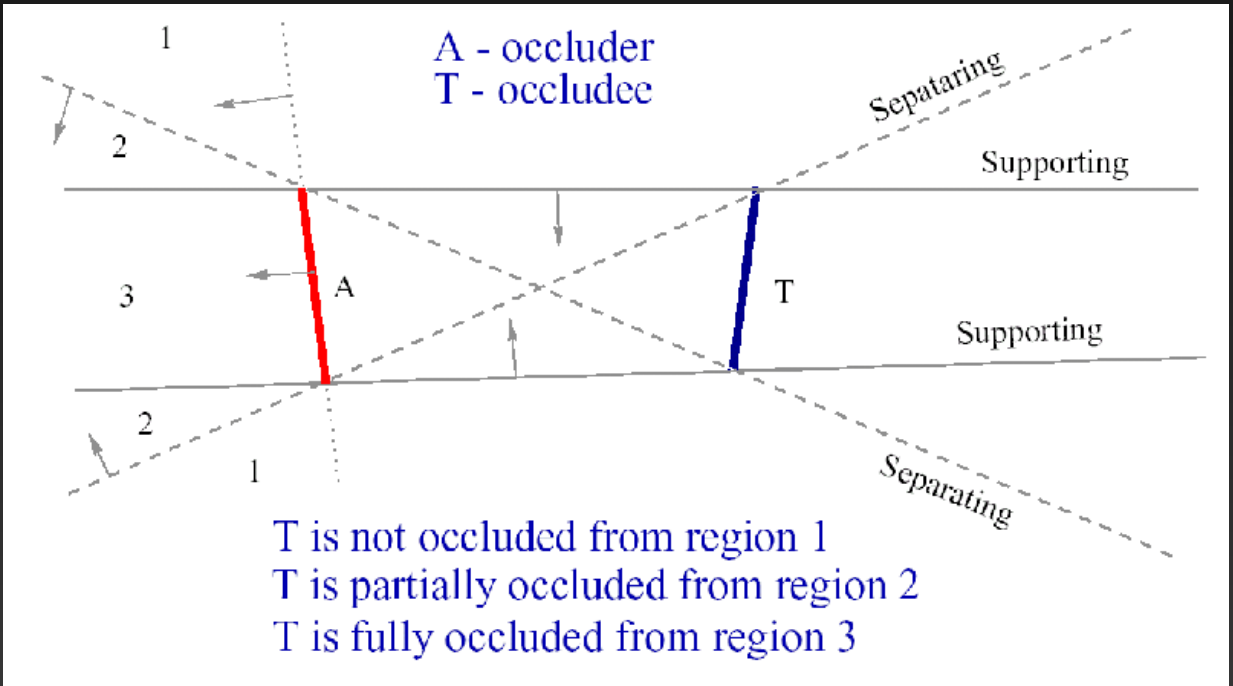
De nombreuses méthodes

- Point / région
- Espace image / espace objet
- Pré-calcul / scène dynamique

Occlusion Culling

Quelques pistes

- Espace object
- Partitions de l'espace
- Problématique des ombres douces
- Occluders virtuels approximatif

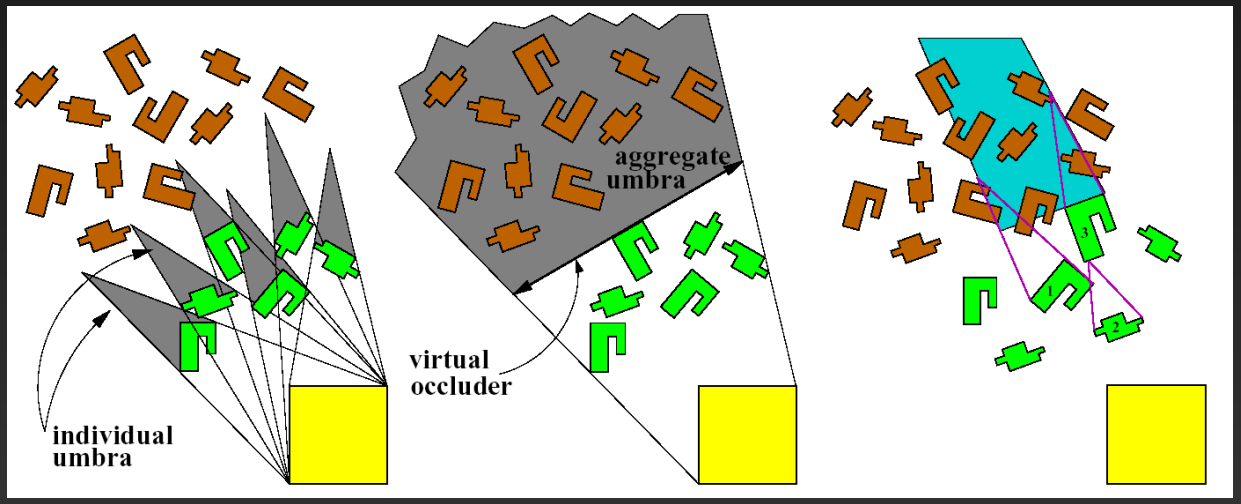


master ISICG limoges

Occlusion Culling

Quelques pistes

- Espace object
- Partitions de l'espace
- Problématique des ombres douces
- Occluders virtuels approximatif



Occlusion Culling

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Quelques pistes

- Espace image
- *Occlusion maps*
- Séparer *Occluder* et *occludee*

Occlusion Culling

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

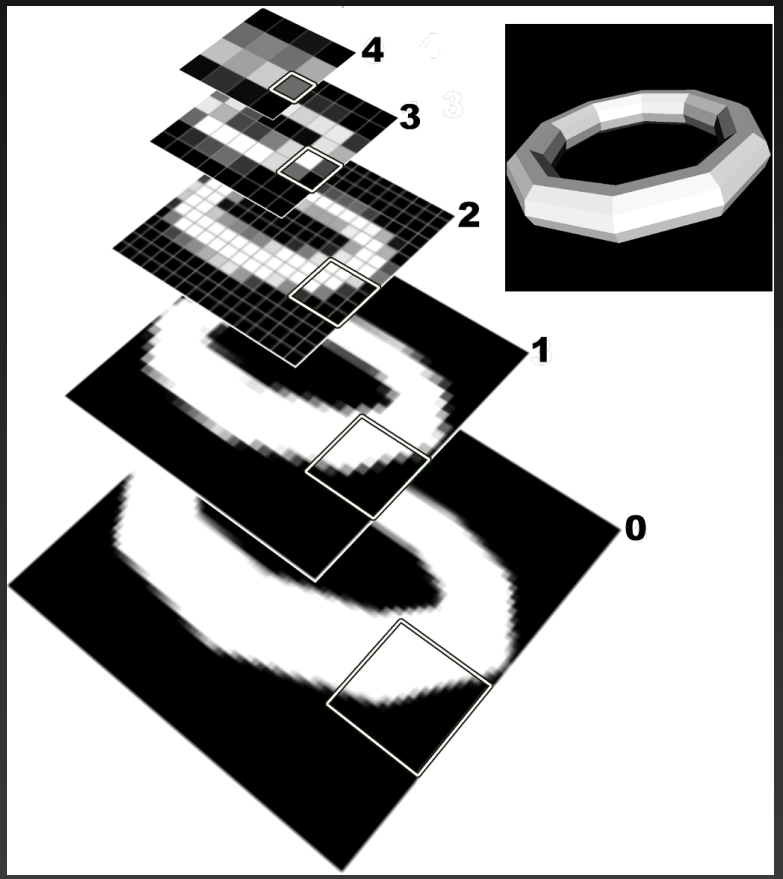
Quelques pistes

- Espace image
- *Occlusion maps*
- Séparer *Occluder* et *occludee*

Occlusion Culling

Quelques pistes

- Espace image
- *Occlusion maps*
- Séparer *Occluder* et *occludee*
- Cartes hierarchiques



Occlusion Culling

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Quelques pistes

- Espace image
- *Occlusion Query*
- Mécanisme GPU
- Renvoie le nombre de pixel couvert par une primitive

Occlusion Culling

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Quelques pistes

- Espace image
- *Occlusion Query*
- Mécanisme GPU
- Renvoie le nombre de pixel couvert par une primitive
- Principe

Occlusion Culling

Quelques pistes

- Espace image
- *Occlusion Query*
- Mécanisme GPU
- Renvoie le nombre de pixel couvert par une primitive
- Principe
  - Activer la requête d'occlusion

Occlusion Culling

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Quelques pistes

- Espace image
- *Occlusion Query*
- Mécanisme GPU
- Renvoie le nombre de pixel couvert par une primitive
- Principe
  - Activer la requête d'occlusion
  - Effectuer le rendu de l'objet *simplifié*

Occlusion Culling

Quelques pistes

- Espace image
- *Occlusion Query*
- Mécanisme GPU
- Renvoie le nombre de pixel couvert par une primitive
- Principe
  - Activer la requête d'occlusion
  - Effectuer le rendu de l'objet *simplifié*
  - Récupérer n le nombre de fragments couverts par cet objet

Occlusion Culling

Quelques pistes

- Espace image
- *Occlusion Query*
- Mécanisme GPU
- Renvoie le nombre de pixel couvert par une primitive
- Principe
  - Activer la requête d'occlusion
  - Effectuer le rendu de l'objet *simplifié*
  - Récupérer n le nombre de fragments couverts par cet objet
  - Afficher l'objet complexe si n positif

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Structures d'accélérations

" Oh ! Par mes moustaches, je suis en r'tard, en r'tard, en r'tard !"
(le lapin blanc)

Structures d'accélérations

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Structures d'accélérations

- On ne peut pas tester chaque primitives
- Il faut hiérarchiser et organiser la calculs

Méthodes classiques

- Boîtes englobantes
- Hiérarchie de boîtes englobantes
- Grilles d'accélération

Structures d'accélérations

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

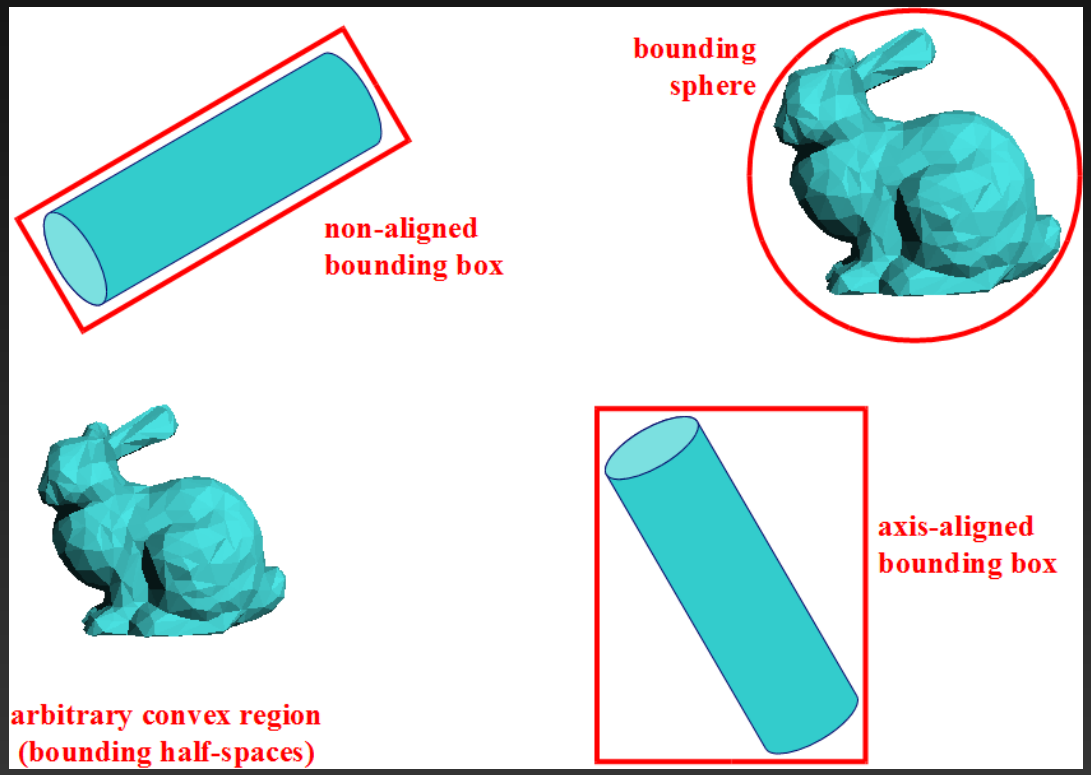
Boîtes englobantes

- Géométrie simplifiée englobant la géométrie d'un modèle

Structures d'accélérations

Boîtes englobantes

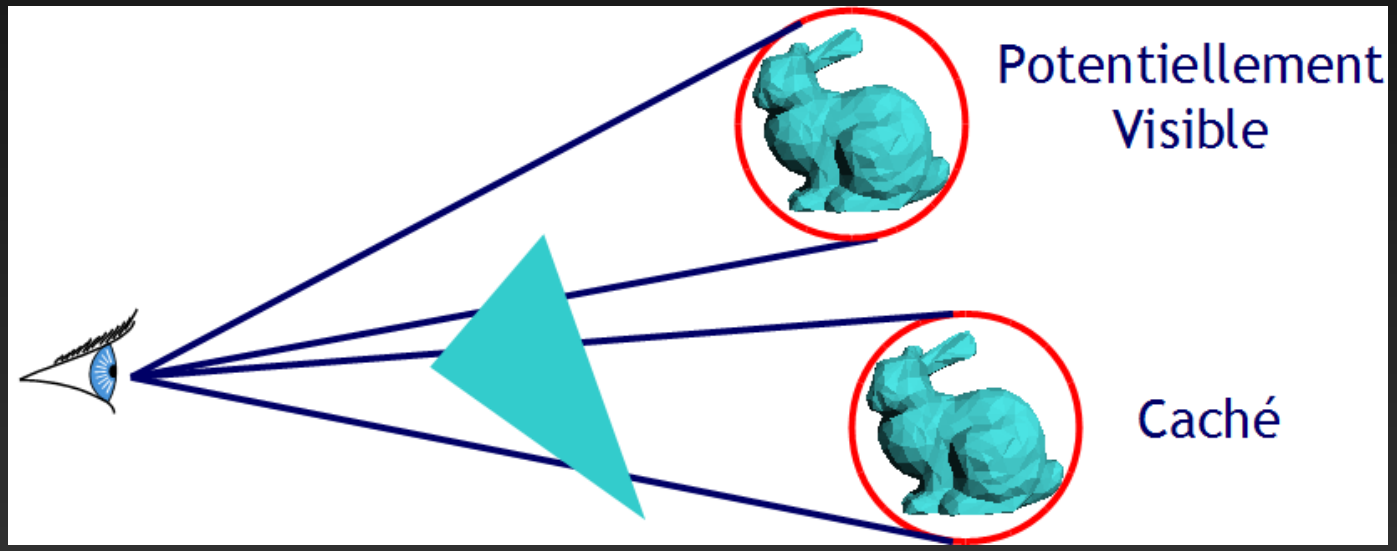
- Géométrie simplifiée englobant la géométrie d'un modèle
- Complexité/précision variable



Structures d'accélérations

Boîtes englobantes

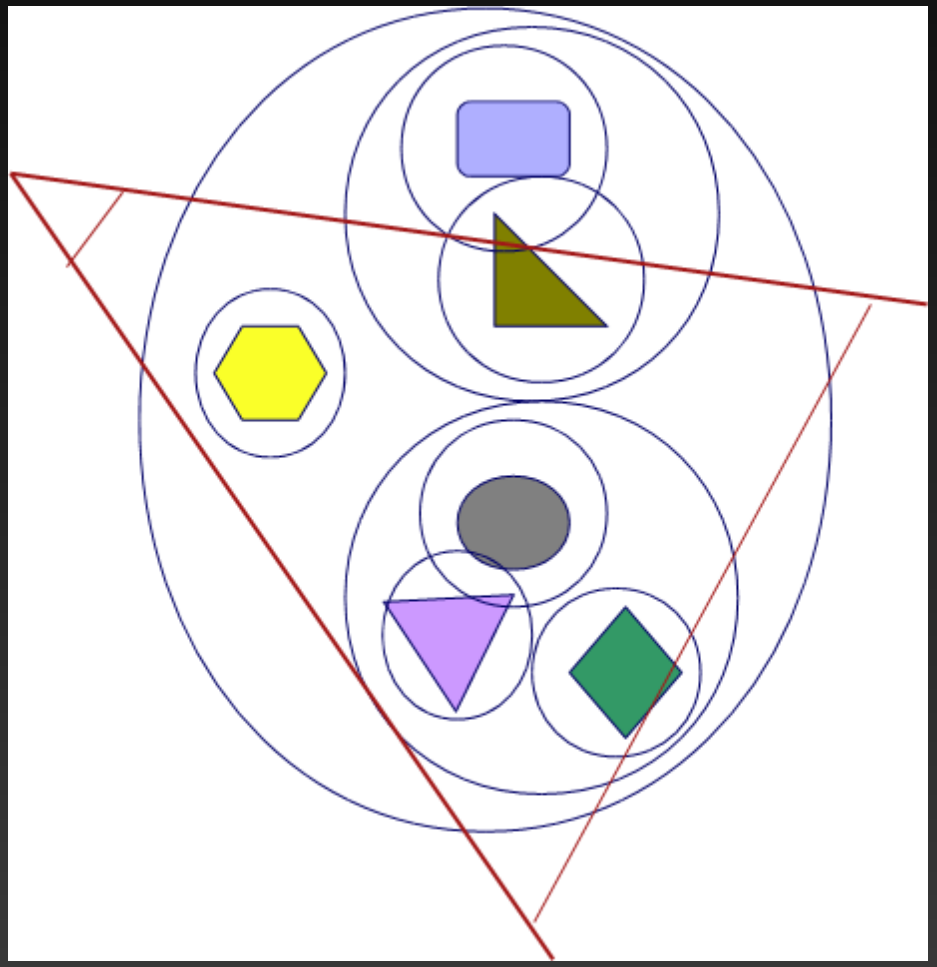
- Permet une réponse conservative rapide
  - Bout de boite visible $\Longrightarrow$ potentiellement visible
  - Boite cachée $\Longrightarrow$ sûrement caché



Structures d'accélérations

Hiérarchie de boîtes englobantes

- Testée de haut en bas pour raffiner les réponses
- Equivalent à un parcours d'arbre



Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

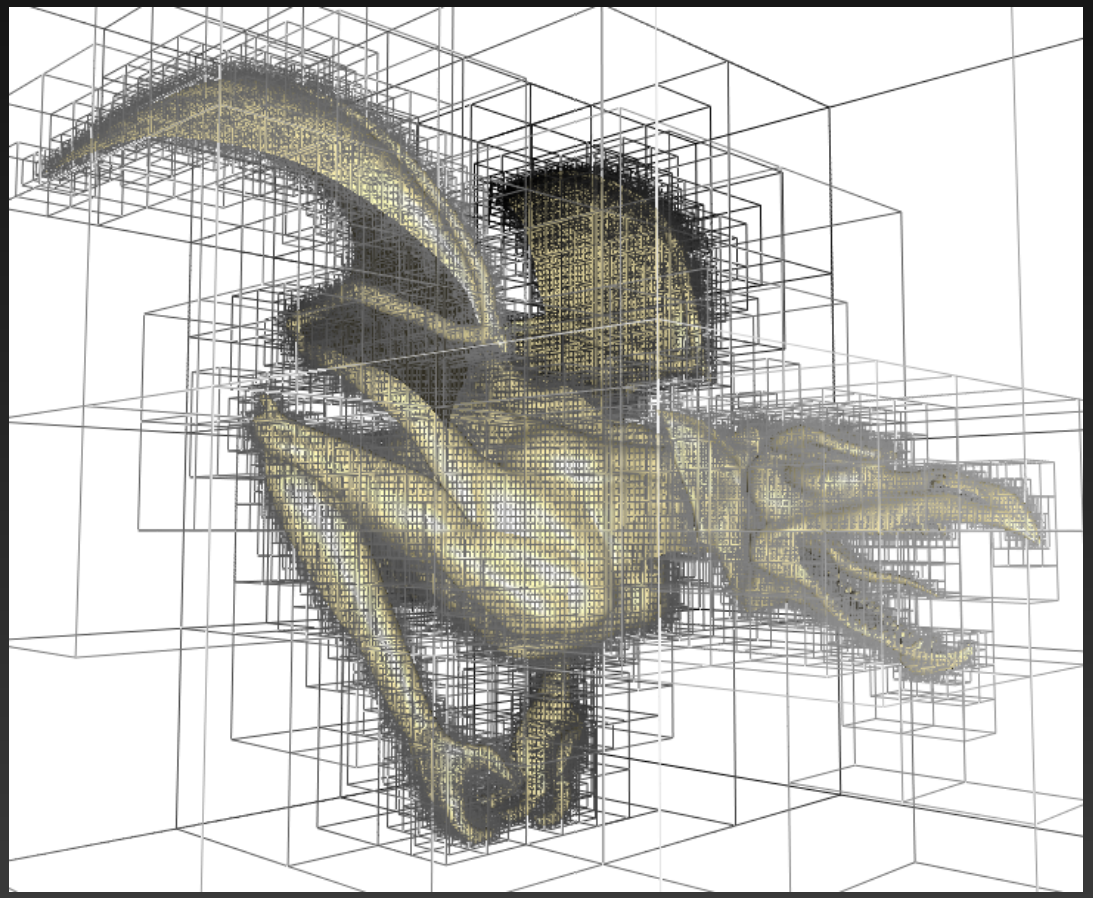
master ISICG limoges

Structures d'accélérations

Grilles et partitions de l'espace

- Partitionner l'espace
- Quad-Trees et Oct-Trees

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

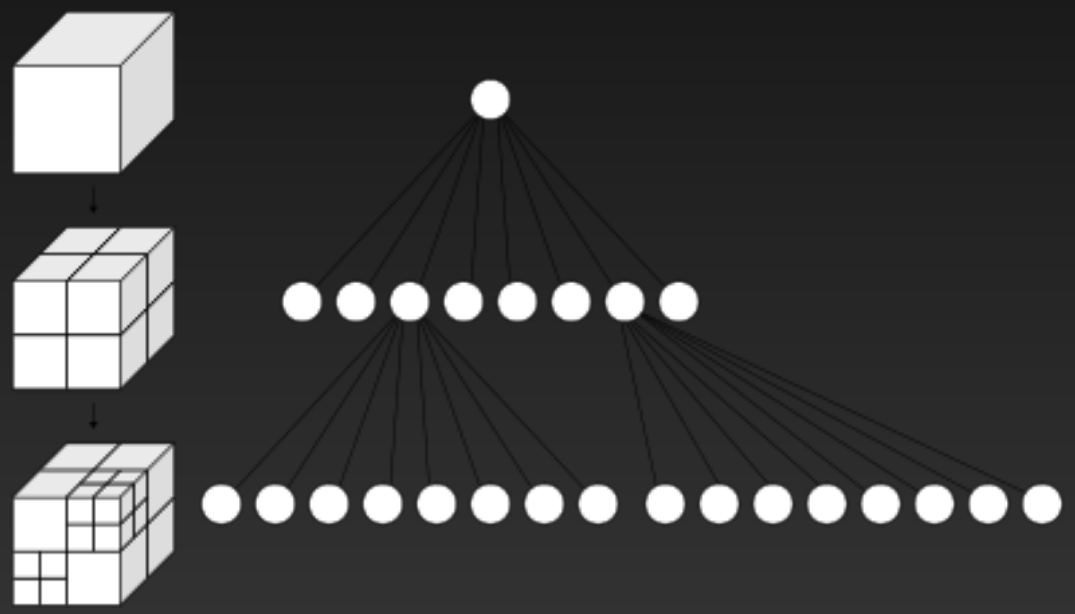


Structures d'accélérations

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Grilles et partitions de l'espace

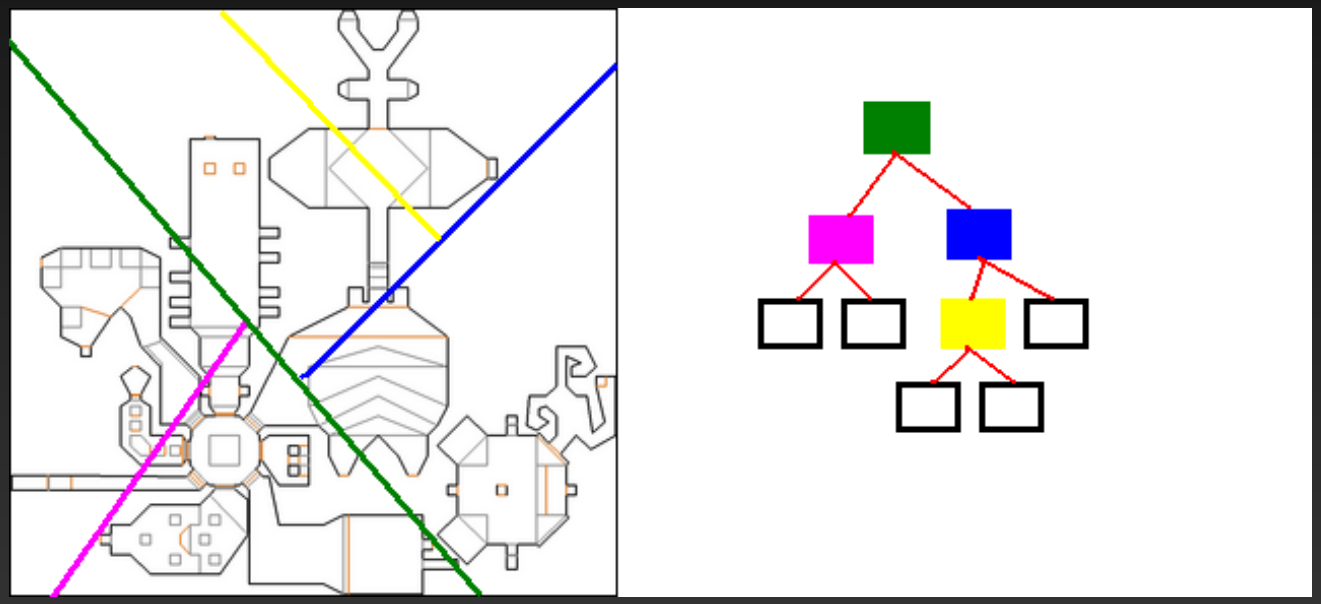
- Partitionner l'espace
- Quad-Trees et Oct-Trees



Structures d'accélérations

Grilles et partitions de l'espace

- Partitionner l'espace
- Quad-Trees et Oct-Trees
- BSP-Trees (Doom, Quake, Unreal...)

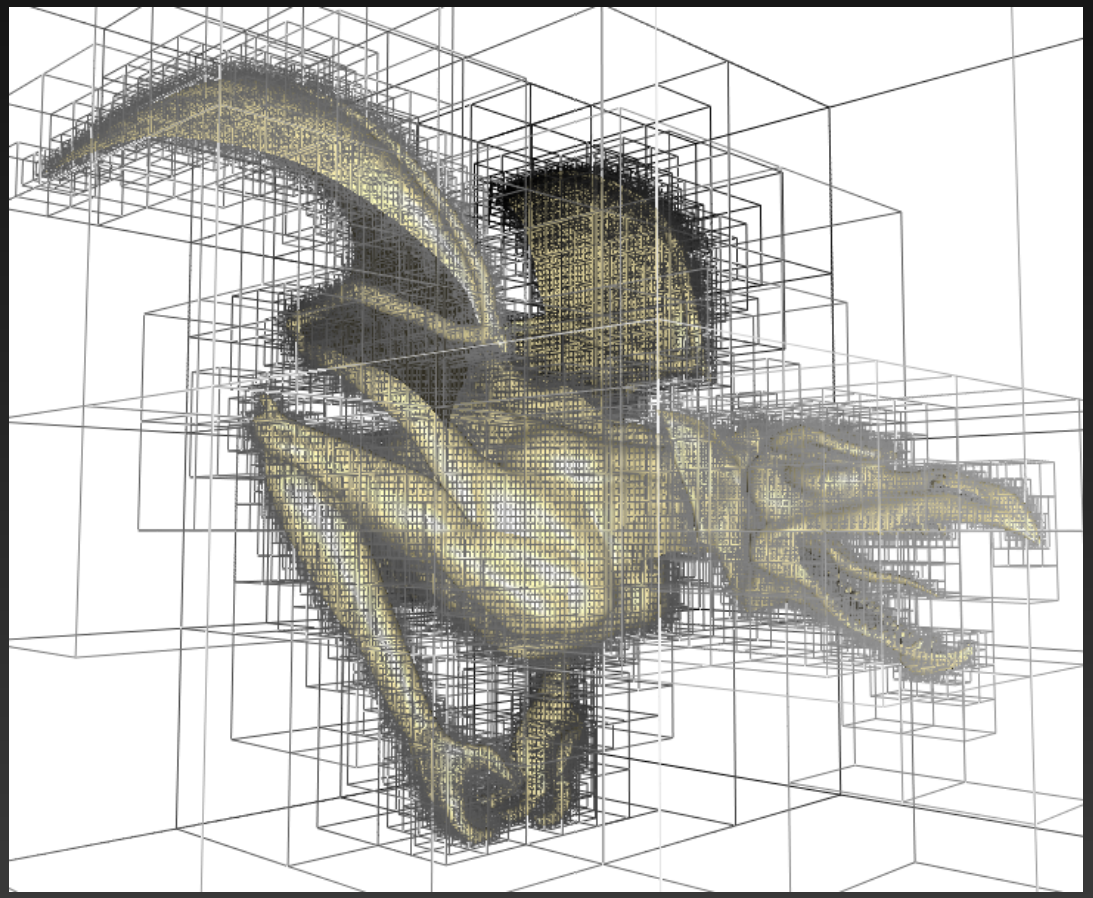


Structures d'accélérations

Grilles et partitions de l'espace

- Partitionner l'espace
- Quad-Trees et Oct-Trees

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

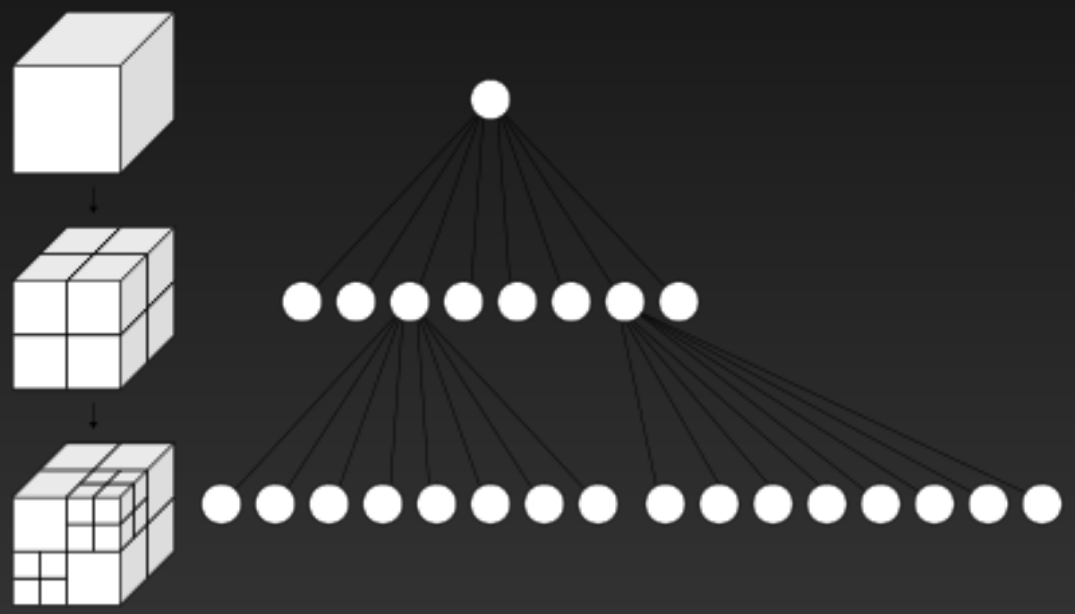


Structures d'accélérations

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Grilles et partitions de l'espace

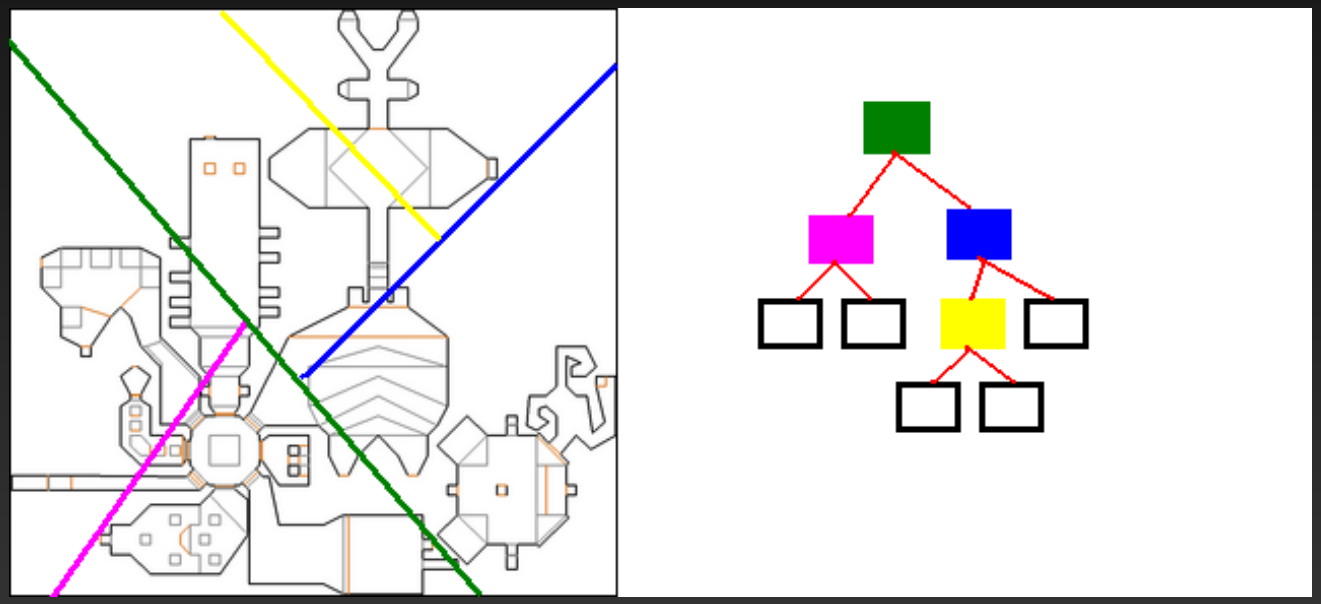
- Partitionner l'espace
- Quad-Trees et Oct-Trees



Structures d'accélérations

Grilles et partitions de l'espace

- Partitionner l'espace
- Quad-Trees et Oct-Trees
- BSP-Trees (Doom, Quake, Unreal...)



Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Niveau de détails

Niveau de détails

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Des (très) gros modèles

- Il existe toujours des plus gros modèles

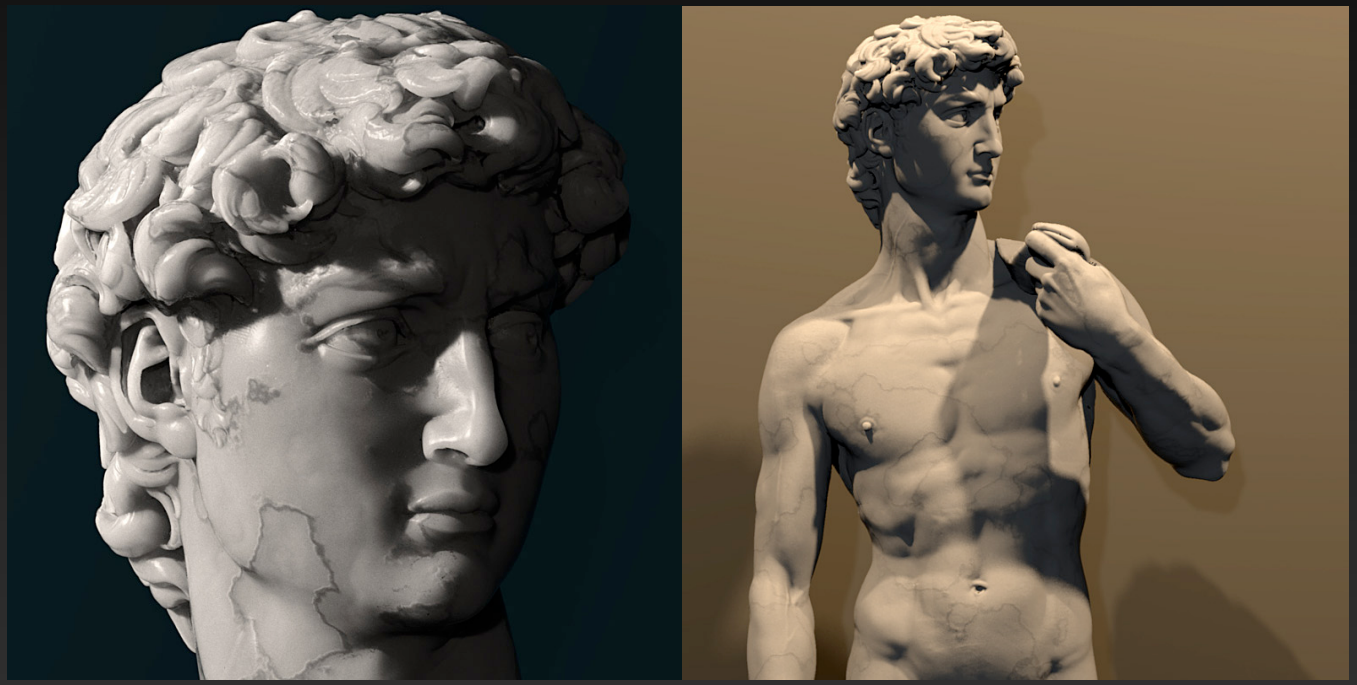
13 millions de faces

Niveau de détails

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Des (très) gros modèles

- Il existe toujours des plus gros modèles

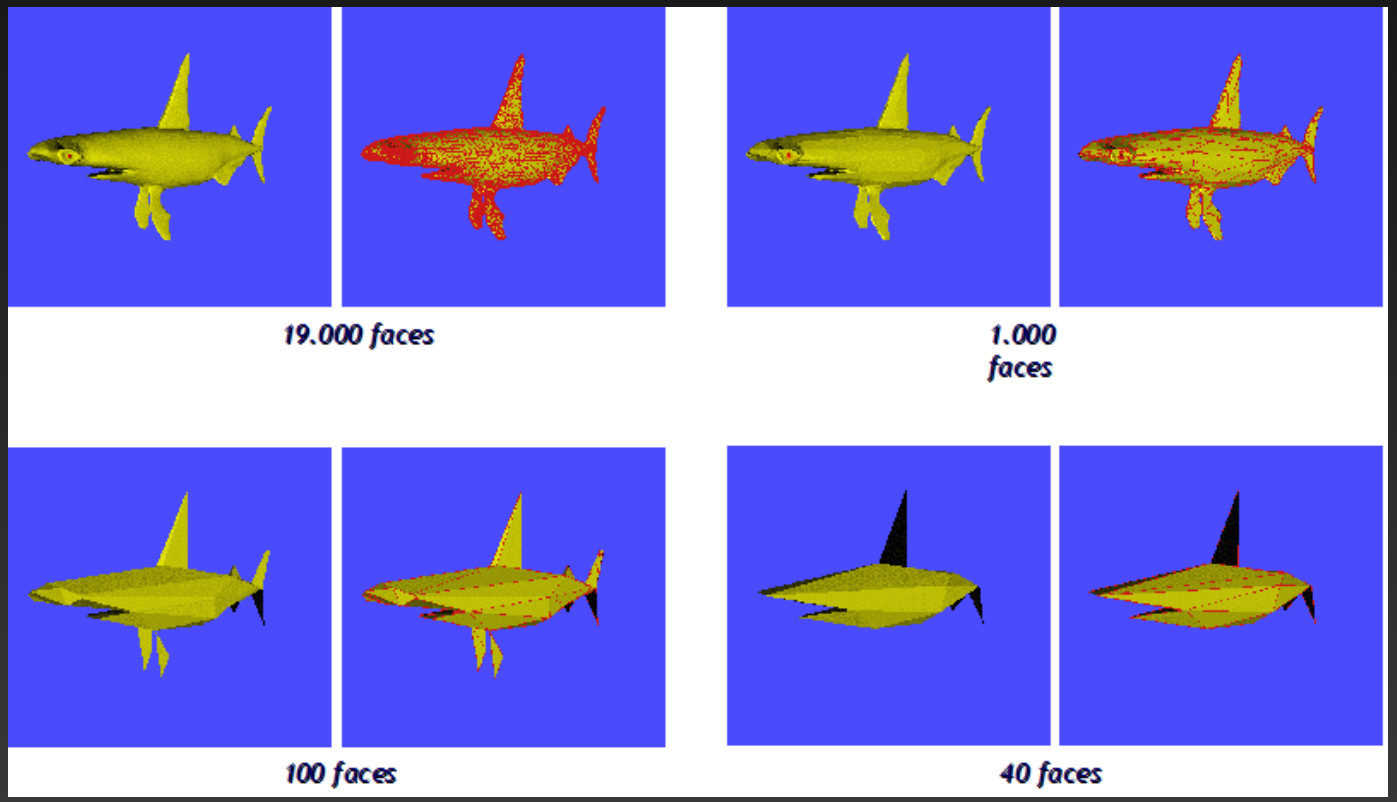


3 milliards de faces

Niveau de détails

Simplifier la géométrie

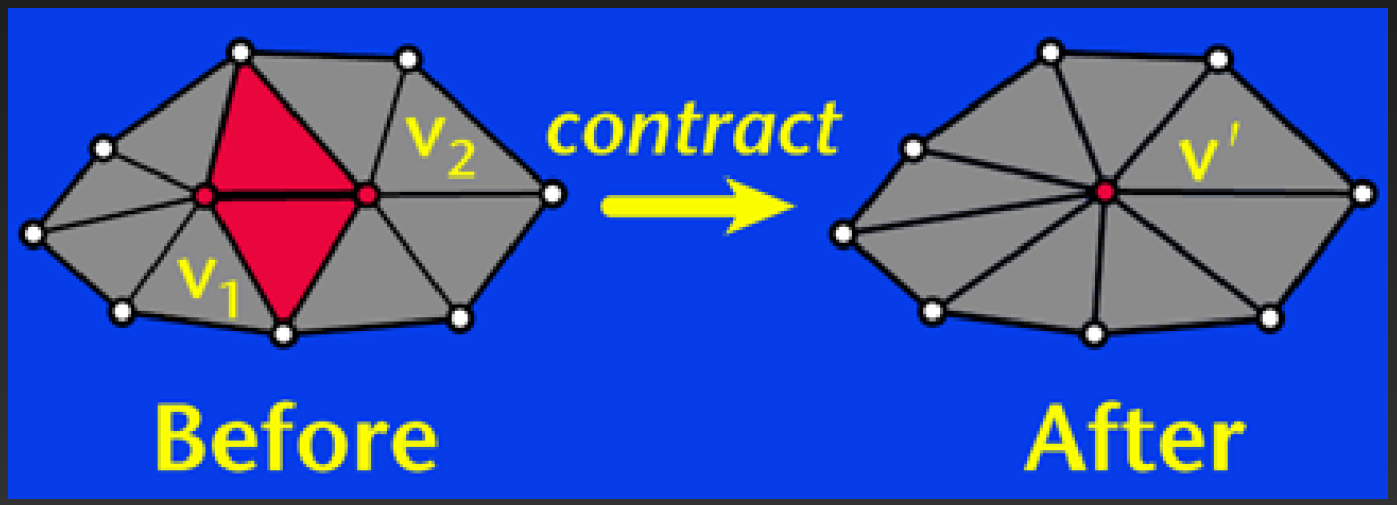
- Générer plusieurs représentations d'un modèle
- Contrôle du nombre final de polygones



Niveau de détails

Simplifier la géométrie

- Générer plusieurs représentations d'un modèle
- Contrôle du nombre final de polygones
- Plusieurs méthodes (pré-calcul, calcul au chargement, modeleur 3D ,...)



Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Niveau de détails

Simplifier la géométrie

- Générer plusieurs représentations d'un modèle
- Contrôle du nombre final de polygones
- Plusieurs méthodes (pré-calcul, calcul au chargement, modeleur 3D ,...)

Au cours du rendu

- Calculer des niveaux de détail pour chaque objet de la scène
- Choisir à chaque instant le bon niveau de détail et l'afficher
- Eviter les problèmes de *popping*

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Graphes de scène

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Graphes de scène

Motivation

- Organiser la scène
- Gestion à un plus haut niveau
- Mécanismes de rendus indépendants
- Manipulable sans connaître le code
- Les objets à modéliser existent rarement de manière isolée
  - Chaîne corps/épaule/bras/main/doigts
  - Les roues d'un véhicule y sont attachées
  - La chaise est devant la table, le téléphone est posé dessus
  - Les chaises et les tables sont alignées
- Une organisation hiérarchique permet d'exprimer les relations de dépendances entre les objets

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Graphes de scène

Structure

- Un graphe de scène est un arbre
- Chaque noeud possède un unique parent
- Chaque noeud contient des changement d'état (lumières, couleurs, sons) et/ou un modèle géométrique
- Arbre a parcourir pour effectuer les opérations

Opérations courantes

- Affichage
- Manipulation
- Interaction
- Animation
- Culling

Graphes de scène

Il en a existé et en existe encore beaucoup

- Inventor/VRML
- Performer
- OpenSG
- OpenScenegraph
- Java3D
- etc.

Le problème est de choisir ou d'en créer un nouveau...

Graphe de scène minimaliste

Un graphe de scène minimaliste

- Une hiérarchie existe déjà pour les transformations
  - Chaque **Frame** à un parent
  - Composition des transformations

En pratique :

- Chaque node est un noeud d'un graphe de scène (avec une racine : le noeud **Root**)
- Chaque node contient :
  - Un repère local exprimé dans le repère du noeud parent
  - Éventuellement un modèle et une méthode de rendu ...
  - Une liste de dépendances (noeuds fils)...

Optimisations
Visibilité et culling
Structures d'accélérations
Niveau de détails
Graphes de scène

Graphe de scène minimaliste

Dans le fonction *render* de la classe **Engine**

- On collecte les noeuds du graphe (**NodeCollector**)
- *collector->collect(scene->getRoot()) ;*
- On effectue le rendu de la liste des noeuds
- *for (unsigned int i = 0 ;i < collector->nodes.size() ;i++) collector->nodes[i]->render() ;*

classe **NodeCollector**

- Objectif : Parcoure et collecte les noeuds du graphe
- Pour l'instant : **StandardCollector**
  - Objectif : Collecter **tous** les noeuds
- Pour plus tard : **XXXXCollector**
  - Frustum Culling
  - Cellules et Portails, BSP...

Graphe de scène minimaliste

Pour aller plus loin

- Problème d'efficacité (changement d'états)
- Optimisations possibles (tri d'une liste de noeuds)
- Coût du tri vs coût des changements d'états
- Méthodes de visibilité potentielle
- Modifications dynamiques des graphes de scènes