

**Université de Sherbrooke**  
**Département d'informatique**  
**IMN 529 TP3 - Radiosité**

## **1. Objectif**

L'objectif de ce devoir est d'implanter l'illumination globale par la méthode de radiosité vue en classe.

## **2. N3D et la radiosité**

Pour réaliser ce travail pratique, vous devez utiliser le noyau graphique du cours (**N3D**).

### ***Fichiers scènes***

Ce TP utilise le même format de fichier scène que les autres TPs. Chaque objet atomique est transformé en une ou plusieurs pièces (« patches »). Un polygone (PG) est constitué d'une seule pièce tandis que les quadrilatères (QD) peuvent être subdivisés en  $n$  par  $m$  pièces.

Les sources lumineuses sont des objets de la scène. Pour rendre un objet lumineux, il faut spécifier un coefficient d'émission (CE).

*Exemple d'un rectangle lumineux subdivisé en 32 (4\*8) pièces (patches) :*

```
QD 4 8
  0 0 0 1 0 0 1 2 0 0 2 0
CE 20 20 20
ID
```

Plusieurs fichiers scène conçus pour ce TP sont fournis dans le répertoire « **working** » de N3D. Ces derniers se nomment *radio1.scn*, *radio2.scn*, ..., *radio7.scn*.

### ***Limitations***

La version actuelle de N3D a quelques limitations vis-à-vis la radiosité. Les cylindres (CL) et les sphères (SP) ne sont pas implantés. De plus, l'usage des quadrilatères est limité à des parallélogrammes.

### ***Calcul de la radiosité***

Pour lancer le calcul de la radiosité, il suffit de cliquer sur le bouton droit de la souris et de choisir l'option « Radiosité (Gauss-Seidel) ». Cette action a pour effet d'appeler la fonction **CalculRadiosite\_GS()** défini dans le fichier **radiosite.cpp**. Cette dernière déclenchera les 4 d'étapes suivantes :

1. Déterminer le nombre total de pièces de la scène et créer un tableau de pièces.
2. Calculer les facteurs de forme (*à faire*).
3. Résoudre le système d'équations par Gauss-Seidel (*à faire*).
4. Affecter les radiosités calculées aux objets atomiques de la scène.

## **3. Travail déjà fait**

### ***Informations sur les pièces***

La première étape est déjà faite. Cette étape est codée dans la fonction **DeterminerPieces(Objet\* scene)**.

Cette dernière parcourt tous les objets de la scène et compte le nombre total de pièces. De plus, cette fonction crée et initialise le tableau global **Piece\* lespieces**. La taille de ce tableau est égale au nombre total de pièces dans la scène.

Ainsi, chaque pièce de la scène possède une entrée dans le tableau **lespieces**. Ces entrées sont des instances de la classe **Piece**. La classe **Piece** est définie dans le fichier **radiosite.h**. Les principaux attributs de cette classe sont :

- **diffus** : est le coefficient diffus (CD) de la pièce.
- **emis** : est le coefficient émis (CE) de la pièce.
- **radiosite** : est la radiosité de la pièce (attribut à calculer; valeur initiale = emis).

### ***Infrastructure pour le calcul des facteurs de formes***

Une base pour l'implantation du calcul des facteurs de formes est également donnée. La fonction **CalculerFacteursFormes(...)** s'occupe de lancer le calcul des facteurs de formes. Cette fonction crée la matrice **reel \*\*FacteursDeFormes**. Cette matrice est une variable globale et elle est de taille  $n$  par  $n$  où  $n$  est le nombre total de pièces. De plus, cette matrice est initialisée à zéro. La valeur de **FacteurDeFormes[i][j]** (noté  $F_{ij}$  en classe) représentera la proportion de l'énergie quittant la pièce  $i$  et qui atteint la pièce  $j$ . Vous aurez à calculer tous les éléments de cette matrice.

Aussi, la fonction **CalculerFacteursFormes(...)** s'occupe de parcourir tous les objets de la scène. Pour chaque pièce de la scène, la fonction fait un appel à **FacteursFormePiece(...)** qui elle se chargera de calculer les facteurs de formes d'une pièce. Vous devrez coder cette dernière.

### ***Affectation des radiosités aux objets de la scène***

La dernière étape a été codée aussi. Ce travail est accompli par la fonction **AffecterRadiositeScene(...)**. Cette dernière affecte les radiosités calculées (**lespieces[i].radiosite**) aux objets atomiques de la scène. Cette étape permet l'affichage du résultat (bouger la caméra pour voir le résultat).

## **4. \*\*\*\* Travail à faire \*\*\*\***

Toutes vos modifications doivent être faites dans le fichier **radiosite.cpp**.

### ***Calcul des facteurs de formes***

La première tâche à effectuer est le calcul des facteurs de formes. Pour réaliser cette tâche, vous devez coder le corps de la fonction suivante :

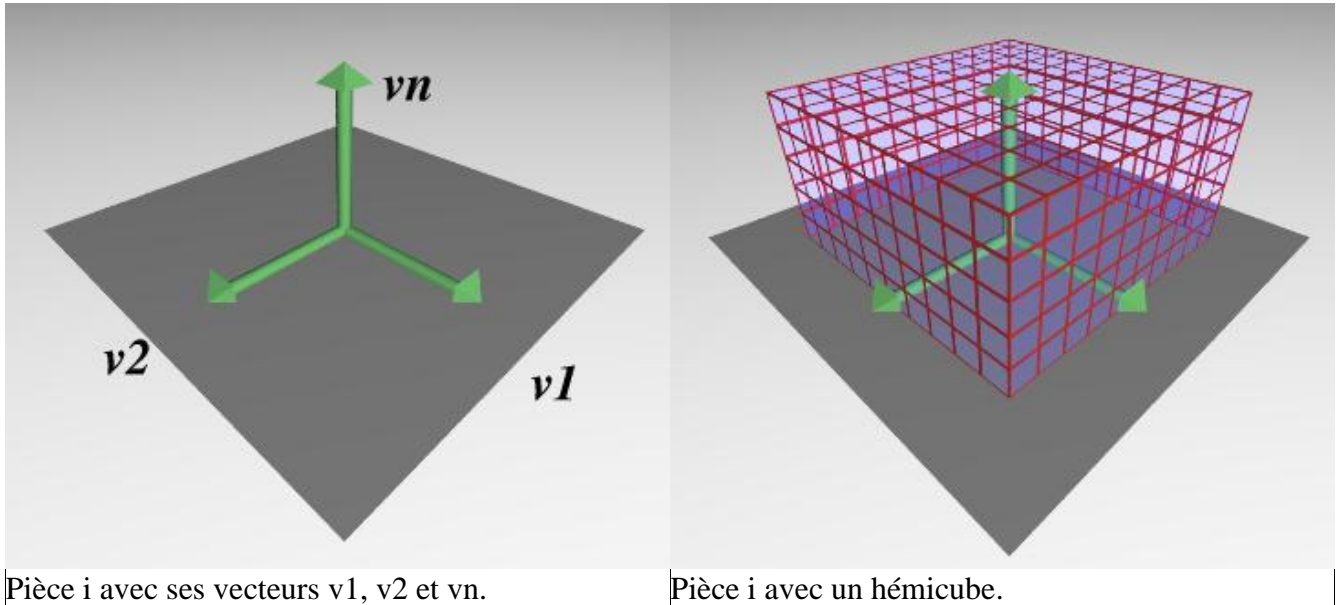
```
void FacteursFormePiece(Objet* scene, const point& centre, const vecteur& v1, const vecteur& v2,
    const vecteur& vn, reel* ligneFF)
```

Comme mentionné précédemment, cette fonction est appelée pour toutes les pièces de la scène. À chaque appel, la fonction doit calculer les facteurs de formes d'une pièce donnée (pièce  $i$ ). Les paramètres sont :

- scene** : est la scène originale. Elle est utilisée pour calculer les intersections.
- centre** : est le point centrale de la pièce  $i$ .
- v1, v2** : avec le centre,  $v1$  et  $v2$  définissent un plan sur lequel se trouve la pièce  $i$ .
- vn** : vecteur normal à la pièce  $i$ .
- ligneFF** : est un pointeur vers le début de la ligne  $i$  de la matrice **FacteursDeFormes**.

Les deux images suivantes illustrent bien les vecteurs  $v1$ ,  $v2$  et  $vn$ . Sur la deuxième image, on voit l'hémicube sur lequel vous devez envoyer des rayons. Pour chaque carré des surfaces de l'hémicube,

vous devez appeler la fonction **Objet\_InterPiece(Objet& scene, const point&, const vecteur&, int\* numero\_piece\_touchée)**. Voir le fichier *inter.h*.



Pour ce TP, utilisez une taille variable pour la précision de l'hémicube. Pour la remise du TP, fixez une définition de 100x100 pour la face haut et de 100x50 pour les 4 côtés (voir *const int PRECISION\_HEMICUBE 100* dans *radiosite.cpp*). Afin d'obtenir des résultats plus rapidement, vous pouvez réduire temporairement ce nombre à 30. Vous pouvez aussi réduire le nombre de subdivision des objets de la scène (fichiers .scn). Par contre, la réduction de ces paramètres aura pour effet de réduire la qualité de l'image résultante. Vous pouvez aussi obtenir des résultats beaucoup plus rapidement en mode « Release ».

### **Résolution du système d'équation par Gauss-Seidel**

Dans un second temps, vous devez résoudre itérativement l'équation suivante :

$$\mathbf{B_i} = \mathbf{E_i} + \mathbf{R_i} \Sigma (\mathbf{B_j} \mathbf{F_{ij}})$$

Dans le fichier *radiosite.cpp*, des noms de variables différents sont utilisés :

**$\mathbf{B_i}$**  = `lespieces[i].radiosite`  
 **$\mathbf{E_i}$**  = `lespieces[i].emis`  
 **$\mathbf{R_i}$**  = `lespieces[i].diffus`  
 **$\mathbf{B_j}$**  = `lespieces[j].radiosite`  
 **$\mathbf{F_{ij}}$**  = `FacteursDeFormes[i][j]`

Pour résoudre le système, utilisez l'algorithme Gauss-Seidel présenté en cours. Vous devez coder cette partie dans la fonction **Gauss\_Seidel()**.

### **Condition d'arrêt**

La condition d'arrêt de l'algorithme Gauss-Seidel dépend d'une certaine précision désirée. Si la solution

trouvée change peu entre deux itérations, on considère la solution comme suffisamment précise. La précision d'une solution est vérifiée par l'expression suivante :

$$\sqrt{(B_1 - B_{old1})^2 + (B_2 - B_{old2})^2 + \dots + (B_n - B_{oldn})^2} \leq \varepsilon$$

Notes :

- La constante reel EPS\_B (notée  $\varepsilon$  ci-dessus) est fixée à 0.000001 au début du fichier radiosite.cpp. Ne pas la changer.

- $B_i - B_{oldi}$  (i.e. couleur - couleur) calcul une norme dans n3d.

## **5. Remise du devoir**

Ne remettez que le fichier **radiosite.cpp** alors vous ne devez pas modifier d'autres fichiers du noyau. N'oubliez pas d'inscrire votre nom et CIP au début de radiosite.cpp.

***Bon Travail !***