

Synthèse d'image par lancer de rayon



Julien Badie - Maxime Blaess - Anthony Foulfoin - Samuel
Kohn - Benoit Reulier

Plan

- 1 Environnement de développement
 - Logiciels utilisés
 - Organisation du travail
- 2 Conception
 - Packages
 - Diagramme de classe
- 3 Lancer de rayon
- 4 Tests
- 5 Interfaces
 - Interface graphique
 - Interface texte
- 6 Optimisation
- 7 Conclusion

Logiciels utilisés



- Eclipse version 3.4
- plugin subclipse permettant de synchroniser les sources du programme vers un repository SVN via eclipse 
- un serveur SVN privé permettant de synchroniser les sources du groupe 
- un système de gestion de projet en ligne semblable à Trac
- NetBeans 6.5.1 pour mettre au point l'interface graphique
- JUnit pour la réalisation des tests unitaires

Organisation du travail

- Samuel -> classes métiers
- Anthony -> classes applicatives
- Benoît et Maxime -> tests JUnit
- Julien -> interface graphique

Packages

- lancerRayon
- tests
- util
- interfaceGraphique
- interfaceTexte
- xmlParser

Architecture de l'application

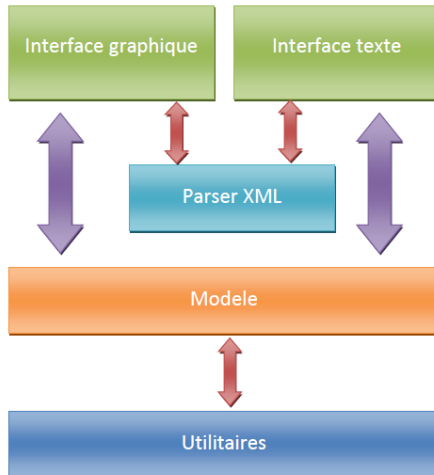
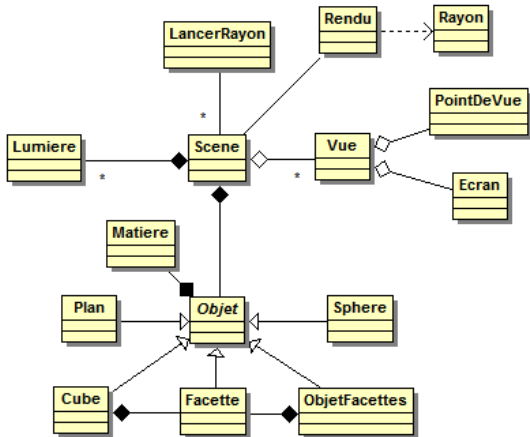
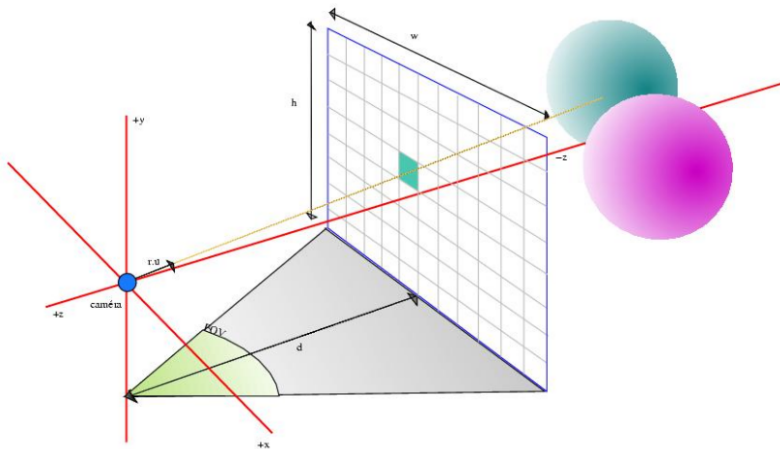


Diagramme de classe



Lancer de rayon

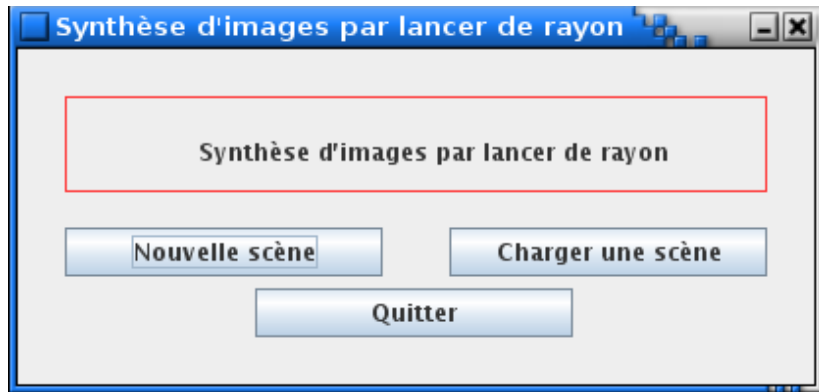


Tests

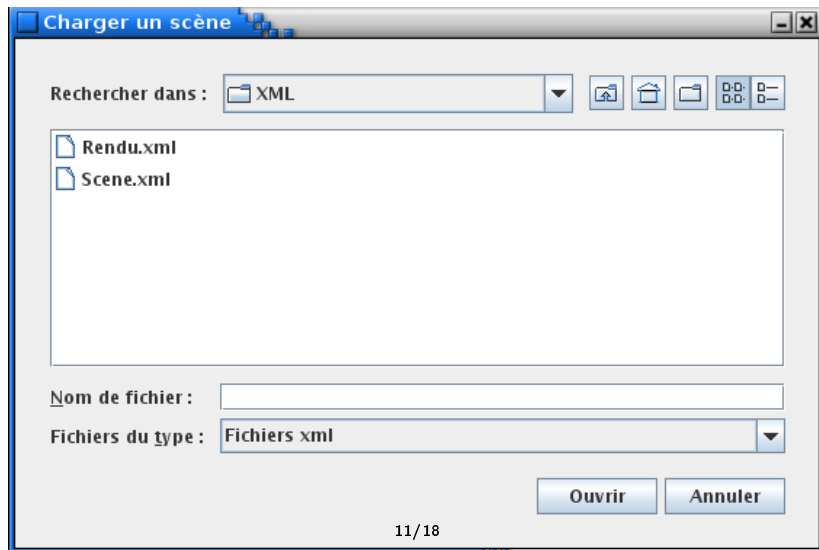
- développés en parallèle des classes
- permettent la détection d'erreurs
- toutes les classes métiers et applicatives testées individuellement

Tests réalisés par des personnes différentes de ceux chargés des classes métiers et applicative => double contrôle

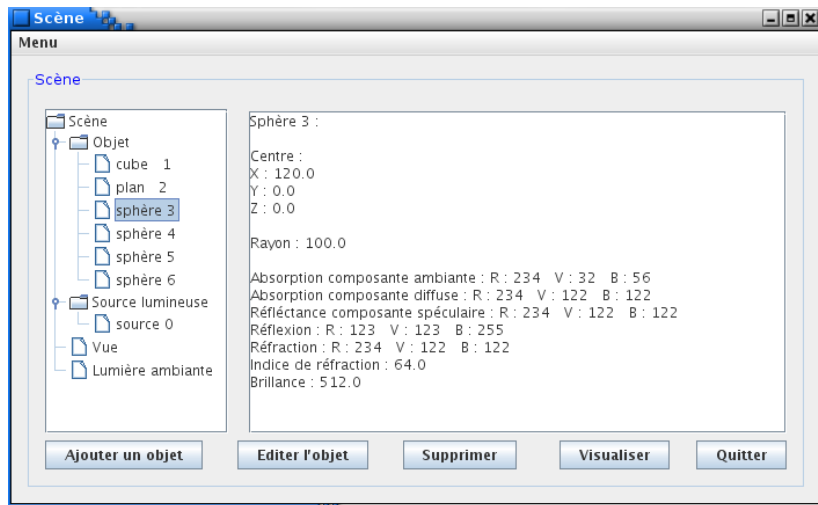
Menu principal



Charger une scène



Editer une scène



Editer un objet

Modifier un objet

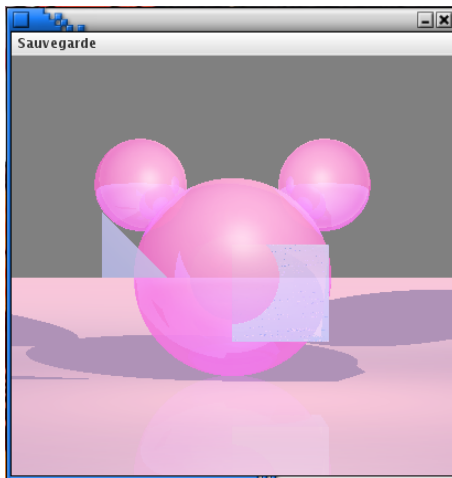
Ajout d'un objet

Sphère Plan infini Cube Source lumineuse Vue

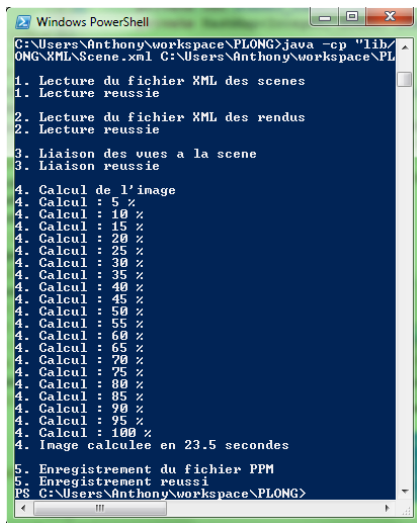
Coordonnées du centre : X: Rayon :
Y:
Z:

Absorption composante ambiante : R: V: B:
Absorption composante diffuse : R: V: B:
Réflectance composante spéculaire : R: V: B:
Réflexion : R: V: B:
Réfraction : R: V: B:
Indice de réfraction :
Brillance :

Fenêtre de rendu



Interface texte



```
Windows PowerShell
C:\Users\Anthony\workspace\PLONG>java -cp "lib/PLONG.XML\Scene.xml C:\Users\Anthony\workspace\PLONG>

1. Lecture du fichier XML des scenes
1. Lecture reussie

2. Lecture du fichier XML des rendus
2. Lecture reussie

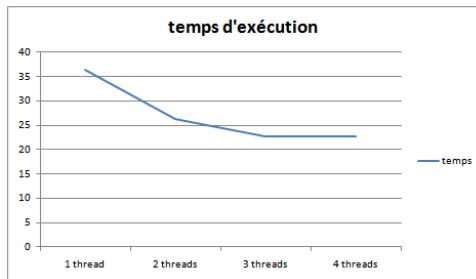
3. Liaison des vues a la scene
3. Liaison reussie

4. Calcul de l'image
4. Calcul : 5 %
4. Calcul : 10 %
4. Calcul : 15 %
4. Calcul : 20 %
4. Calcul : 25 %
4. Calcul : 30 %
4. Calcul : 35 %
4. Calcul : 40 %
4. Calcul : 45 %
4. Calcul : 50 %
4. Calcul : 55 %
4. Calcul : 60 %
4. Calcul : 65 %
4. Calcul : 70 %
4. Calcul : 75 %
4. Calcul : 80 %
4. Calcul : 85 %
4. Calcul : 90 %
4. Calcul : 95 %
4. Calcul : 100 %
4. Image calculee en 23.5 secondes

5. Enregistrement du fichier PPM
5. Enregistrement reussi
PS C:\Users\Anthony\workspace\PLONG>
```

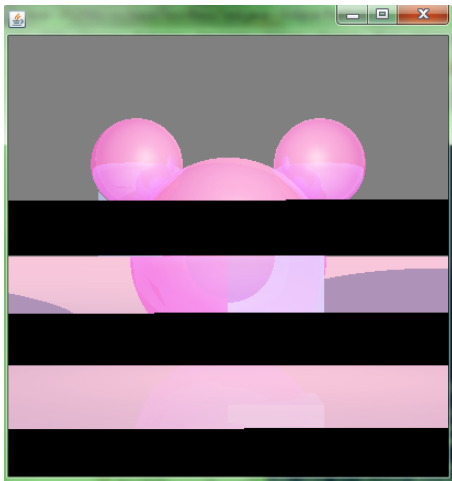
- compte rendu de l'avancée du traitement à chaque étape
- affichage du temps d'exécution
- affichage de l'avancée du calcul tous les 5%

Multithreading



- calcul de chaque pixel indépendant des autres => parallélisation possible
- passage de 1 à 2 threads => gain de 30% en rapidité

Exemple d'image en construction avec 4 threads



Conclusion

- gestion d'autres objets géométriques
- amélioration du multithreading par mise en place d'une liste d'attente
- optimisation de l'interface graphique avec ajout de de fonctionnalités