|  |
| --- |
| Universite de technologie de belfort-montbeliard |
| TO52 – Travail Operationnel |
| Implementation d’algorithmes de calcul de visibilite sur GPU |
|  |
| **Nicolas Said** |
| **12/16/2009** |

|  |
| --- |
| [Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.] |

# Introduction

Dans le cadre de l’UV TO52 (Travail Operationnel) propose a l’UTBM, j’ai etudie la reecriture d’une bilbiotheque logicielle implementant le calcul de visibilite dans un champs de vision (Frustum Culling) pour de nombreux champs de vision, et ce en parallele, en utilisant les capacites des processeurs graphiques.

Un premier travail auquel j’ai participle le semester dernier dans le cadre de l’UV TX52 m’avait déjà permis de travailler sur le problem. Il est vite apparu que les resultants que nous avons obtenus n’etaient ceux que nous esperions. La method utilisee etait certes simple, mais avait de bonnes chances d’aboutir a de bons resultants.

Nous avions egalement propose l’utilisation d’une structure de partitionnement de l’espace afin d’accelerer les operations de Frustum Culling et d’Occlusion Culling. Ayant manqué de temps, nous n’avons malheureusement pas pu implementer cette fonctionalite.

C’est justement l’objet de ce projet. Reecrire la bibliotheque de calcul de visibilite sur GPU afin d’y integrer la gestion d’une structure de partitionnement de l’espace et d’etudier l’impact de cette modification sur les performances atteintes. La gestion de la structure de partitionnement comporte sa creation sur le GPU, ainsi que son utilization pour l’acceleration des operations de Culling.

Dans une premiere partie, je ferai quelques rappels sur le travail qui avait ete effectue par moi-meme et mon binome dans le cadre de la TX52 du semester dernier.

Je presenterai ensuite les differentes methodes de construction de differentes structures de partitionnement de l’espace sur GPU existantes dans la literature. Puis je presenterai plus en details celle qui a ete choisie pour ce projet, ainsi que les details de son implementation.

Puis je lolol

Enfin, je comparerai les resultants obtenus en utilisant cette nouvelle method de calcul de Frustum Culling avec ceux obtenus en utilisant la premiere method.

Je finirai, avant de conclure, par

# Rappel travail precedent

# Etat de l’art

# Algorithme de construction

## Presentation

## Implementation

# Algorithme de Frustum Culling

## Presentation

## Implementation

# Resultats

# Approfondissements

# Documentation

# Conclusion

# Bibliographie