

Fire Monkeys

Modélisation de feu en temps réel

Benjamin Aupetit - Julien Champeau - Arnaud Emilien

Jun 14 2010

- 1 Objectif
- 2 Le modèle
 - Le fluide
 - Les objets
- 3 Portage du modèle de fluide sur GPU
- 4 Démonstration
- 5 Conclusion

Objectif

Réaliser un modèle de combustion d'objets en 3D temps réel.

image

- Le feu
- La fumée
- Interaction avec des objets
- Propagation sur l'objet
- Combustion d'objet

Les étapes de notre démarche

- découvrir le milieu scientifique
- étudier différents articles
- Concevoir notre propre modèle
- Implémenter le modèle
- Améliorer le modèle GPU (objectif initial)

Modèle d'interaction

Interactions entre le modèle de feu et le modèle d'objet.

- Présence des objets (i.e. : pas de flamme a l'interieur des objets)
- Transimission des informations de chaleur d'un modèle à l'autre.
- Gestion de la “pyrolise” des objets.

Principe

- Modèle basé sur le travail de **Jos Stam**, notamment **Stable Fluids**(SIGGRAPH 99 Conference Proceedings).
- Résolution de manière approchée des équation de Navier-Stokes pour la dynamique des fluides incompressibles.
- Un rendu utilisant le principe de “BillBoard”.

La diffusion

La diffusion représente la capacité du fluide à se déplacer dans le milieu ambiant.

$$\frac{\partial \vec{u}}{\partial t} = \nu \nabla^2 \vec{u} \quad (1)$$

Figure: Equation de diffusion

Diffusion

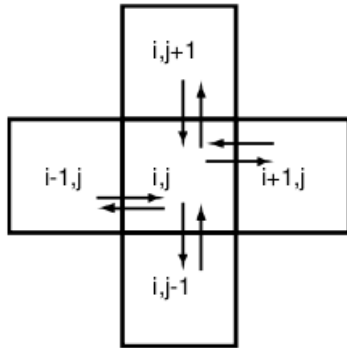


Figure: Diffusion sur une grille 2D

L'advection

Le champ de vitesse va servir à transporter la quantité de fluide dans l'espace. L'auto-advection du champ de vitesse est ce qui va permettre le mouvement des particules de gaz dans le milieu.

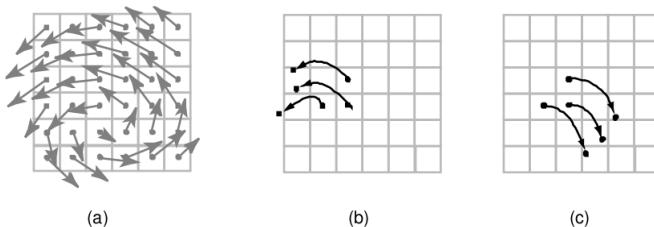


Figure: Principe de l'advection évoquée par **Jos Stam**

La projection

La projection permet de forcer le champ de vitesse à conserver la masse. Pour faire cela il suffit de soustraire le gradient au champ de vitesse après la diffusion et après l'advection de celui-ci.

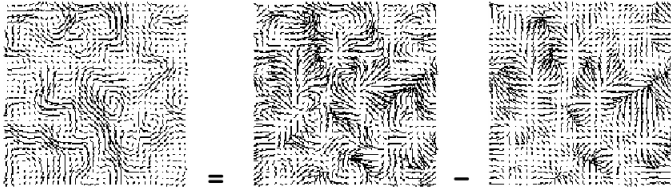


Figure: Correction du champ de vitesse par la soustraction du gradient

Le rendu

Pour le rendu, nous utilisons une technique de “billboard”, ce qui permet d’avoir un cout d’affichage assez faible.

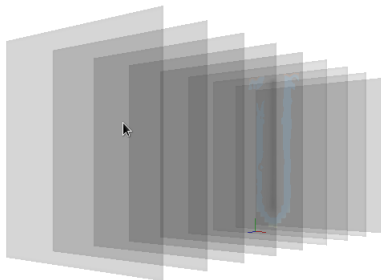


Figure: Vue des plans affichés



Figure: Affichage normal face caméra

Pour améliorer le rendu et le rendre plus réaliste nous avons aussi implémenté un bruit de Perlin qui agit sur l'image.

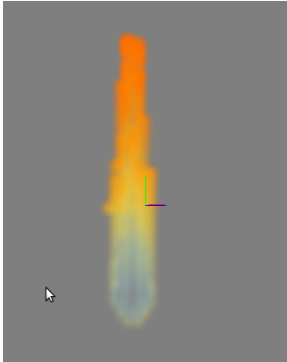


Figure: Sans le bruit de Perlin

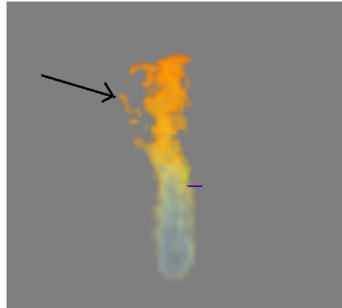


Figure: Avec le bruit de Perlin

Une représentation par voxel

Un objet = un champ de voxels

Interaction facilitée avec le modèle d'objet

Permet de gérer des objets non uniforme

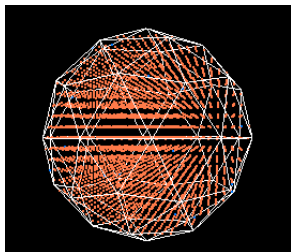


Figure: Exemple de champ de distance

Action des objets sur le fluide

Propagation du feu qui tiens compte des objets
⇒ champ de répulsion autour des objets.

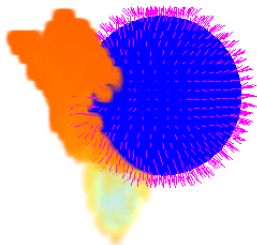


Figure: Objet afficher avec son champ de répulsion

Combustion des objet

Lors de la phase d'échange entre les modèles, les voxels sont détruit pour générer du combustible pour le modèle de fluide, ainsi que de la fumée.

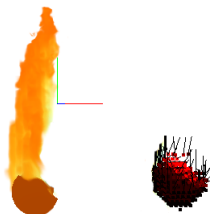


Figure: Combustion d'une sphere, avec diffusion de la chaleur

Rendu

Besoin de recalculer la surface de l'objet localement.
⇒ utilisation d'un algorithme de marching cube adaptatif.



Figure: Surface calculée par marching cube

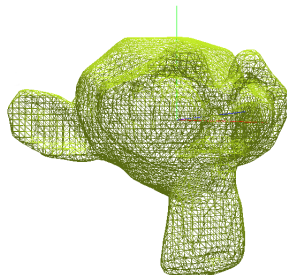


Figure: Arrêtes calculées de marching cube

Le calcul sur GPU

Principe du modèle indentique, calcul différent.

	CPU	GPU
données	tableau unidimensionnel	texture 3D
calcul	itératif	parallèle par cellule

Principe de fonctionnement

- dessin dans un **frame buffer**
- calculs fait par le **fragment shader**

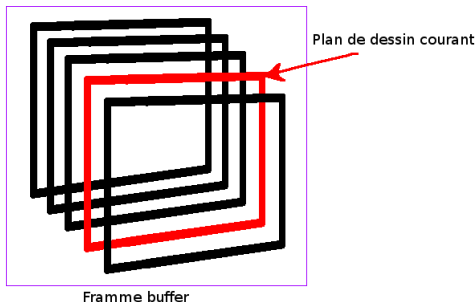


Figure: Frame Buffer Object

Les problèmes rencontrés

Réalisations

Démonstrations

Conclusion

Remerciements

- Marie-Paule Cani pour avoir accepté de nous encadrer, pour ses conseils et indices de recherche.
- Cyril Crassin pour nous avoir aidé à comprendre le fonctionnement de GLSL.
- Aurelie Catel pour le suivi de gestion de projet.
- Nintendo™ pour Super Smash Bros Melee ©.