

# Fire Monkeys

## Animation de feu en 3D temps réel

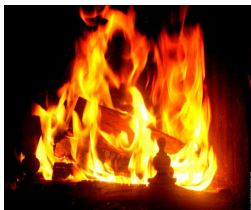
Benjamin Aupetit - Julien Champeau - Arnaud Emilien

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.



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

# Les étapes de notre démarche

Découverte du milieu scientifique :

- Apprendre une démarche de R&D
- Étudier différents articles
- Concevoir notre propre modèle
- Implémenter le modèle
- Paralléliser le modèle en l'implémentant sur GPU

# Le feu

- Résultat de la combustion d'un fluide
- Le fluide se propage dans l'air
- La couleur de la flamme =  $f(\text{chaleur}, \text{gaz})$



Figure: Gaz qui brûle

# 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.

# Représentation du fluide

## 4 grilles uniformes 3D

- Densité de gaz ( flamme )
- Densité de fumée
- Temperature
- Champs de vitesse de l'air

## Phase 1 - La diffusion

Échange de valeurs avec les cases voisines.

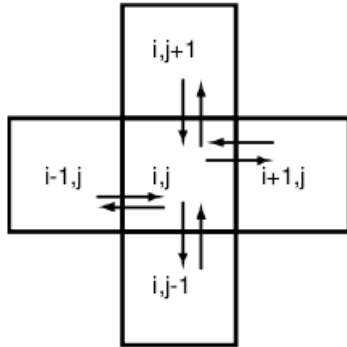
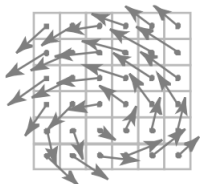


Figure: Diffusion sur une grille 2D

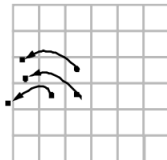


## Phase 2 - Le transport

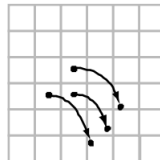
Déplacement des valeurs en fonction du champs de vitesse.



Champs de vitesse



Forward tracking



Backward tracking

**Figure:** Résolution de l'advection par Backtracking introduite par **Jos Stam**

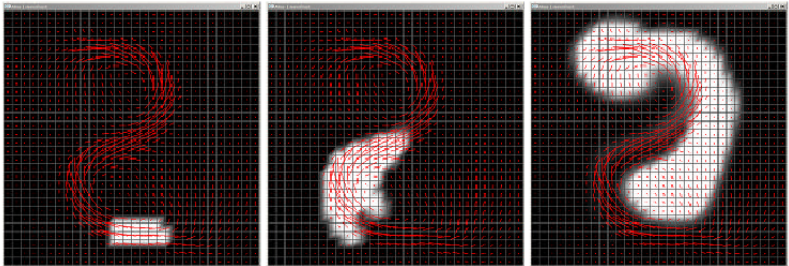


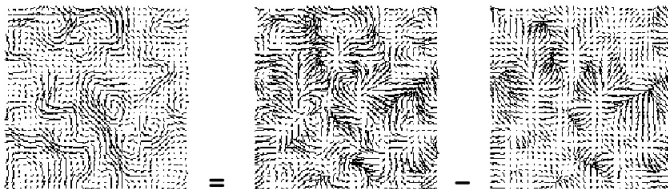
Figure: Advection d'une densité dans une grille 2D

## Phase 3 - La projection

But : avoir un fluide incompressible

Solution : gradient de la pression nul ( Décomposition de Hodge )

Effet visuel : des vortex se créent



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

# Apéritif

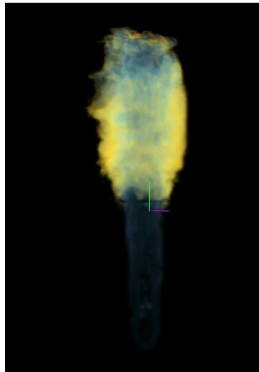


Figure: Rendu de flamme sur la version finale

## Le rendu

Problème : tourner autour de l'image

Solution : une texture 3D et une technique de “billboard”,  
⇒ affichage de plans face à la camera

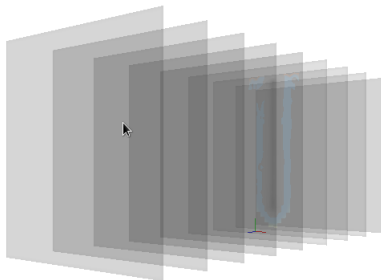
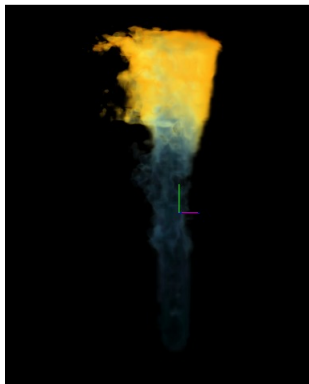


Figure: Vue des plans affichés



Figure: Affichage normal face caméra

# Coloration de la flamme

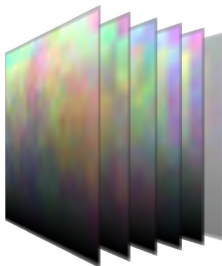


- Corps noir
- Formule de Planck

Figure: Coloration de la flamme

# Bruit de perlin 4D

- Bruit de perlin 3D lissé cosinusoidalement
- Translation des coordonnées de texture
- Continuité temporelle



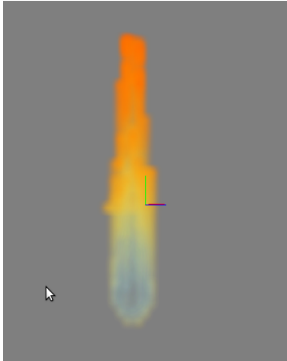


Figure: Sans le bruit de Perlin

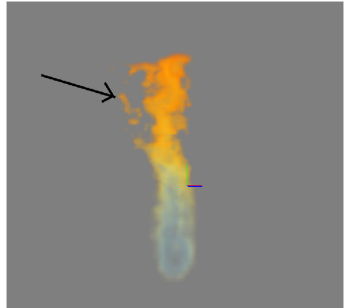


Figure: Avec le bruit de Perlin



# Entrée

# Description des objets

- Volumiques
- Différents matériaux
- Combustibles
- Température de combustion

# Modèle d'interaction

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

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

## Une représentation par voxel

Un objet = un champ de voxels

Interaction facilitée avec le modèle d'objet

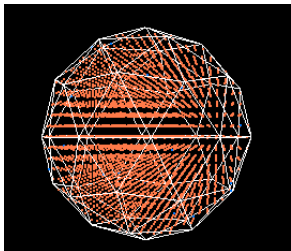


Figure: Exemple de champ de distance

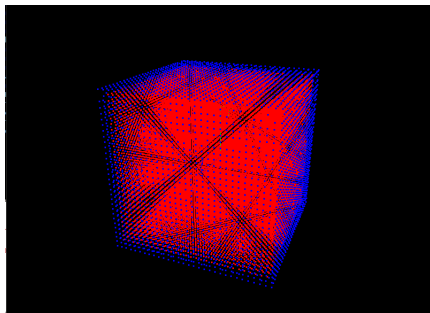


Figure: Exemple de champ de distance 2

## Rendu

Besoin : recalculer la surface de l'objet localement.

⇒ utilisation d'un algorithme de marching cube adaptatif.



Figure: Surface calculée par marching cube

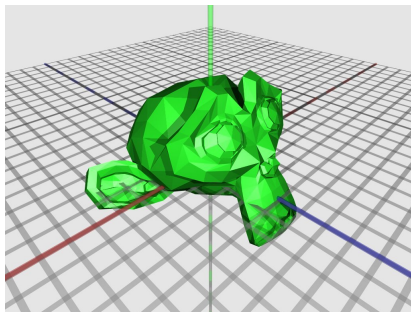


Figure: Mesh original

## Action des objets sur le fluide

Propagation du feu qui tient compte des objets  
⇒ champs de répulsion autour des objets.

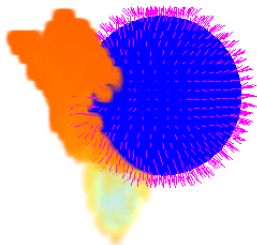


Figure: Objet affiché avec son champs de répulsion

# Combustion des objet

- 1 : combustion
- 2 : génération de gaz / chaleur / fumée
- 3 : si plus de matière : destruction du voxel et recalcul de la surface

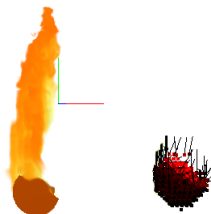


Figure: Combustion d'une sphère, avec diffusion de la chaleur

## Plat principal



# Le calcul sur GPU

Principe du modèle indentique, calcul différent.

	CPU	GPU
données	tableau unidimensionnel	texture 3D
calcul	itératif	parallèle par cellule
rendu	tableau vers texture 3D	texture 3D
coloration	function(chaleur) return RGB	couleur(texture 1D,chaleur)
stockage	min 1E-37, max 1E+37	min 0.0, max 1.0

## Trou Normand

## Problème rencontré

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

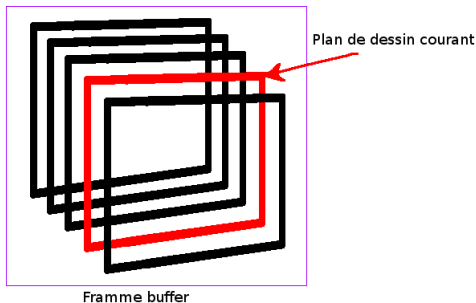


Figure: Frame Buffer Object

## Dessert

# Conclusion

- Résultats très satisfaisants
- Modèle complet
- Beaucoup de nouvelles connaissances
- Petite déception pour le GPU

# 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<sup>TM</sup> pour Super Smash Bros Melee ©.