

# GMINT317 - Moteurs de jeux – TP5

## Textures et GPU

---

Rémi Ronfard  [remi.ronfard@inria.fr](mailto:remi.ronfard@inria.fr)  <https://team.inria.fr/imagine/team/>

### Objectifs

Dans ce TP, nous allons nous intéresser à étudier les possibilités offertes par le GPU. Ainsi, ces travaux vont vous demander de vous familiariser avec différentes techniques :

- Pour habiller vos objets 3D
- Pour appliquer des transformation sur un objet
- Pour améliorer le rendu des TPs précédents
- Pour obtenir un rendu temps réel

### Bonus :

- Modifier votre gestionnaire de ressources pour supporter :
  - les textures
  - les shaders
- Utiliser le GPU pour réaliser les calculs d'interaction entre les particules (cf TP précédent : pluie, neige, eau)

### Gestionnaire de version

Cloner le cinquième TP :

<https://github.com/imagina-gmin317-2015/tp5>

Au plus tard la semaine prochaine, vous devrez rendre un compte rendu de ce TP avec votre code source.

### Texture d'un triangle

Dans cet exercice, nous allons reprendre le projet exemple de QT 5.5 :

<http://qt-project.org/doc/qt-5/qtgui-openglwindow-example.html>

Ici, nous allons texturer notre triangle, d'abord avec des couleurs, puis avec une image de votre choix.

Ensuite, modéliser une pyramide, et une surface plane qui représente le sol, soulever la pyramide et la faire tourner sur elle-même.

Texturer le sol ainsi que la pyramide (chaque face de la pyramide devra avoir une texture différente).

## Shader sur le triangle

Ajouter à votre scène une sphère et mettre en place différents shaders :

- Un éclairage diffus
- Une méthode de : *spherical environment mapping*
- Une méthode de *normal mapping*
- Et une déformation de la géométrie

## Texture et Shader sur le terrain

Maintenant nous allons reprendre les travaux précédents (TP3, TP4).

Texturer le terrain à partir d'une échelle de couleurs représentant l'altitude.

Appliquer une texture unique pour toutes les fenêtres (adaptée au terrain toujours).

Pour chaque saison (donc fenêtre) appliquer une coloration différente à l'aide de textures adaptées.

Ajouter un shader *normal mapping*, pour donner un rendu accidenté de votre scène.

Veuillez bien à conserver l'aspect temps réel de votre application.

## Compte rendu

- Présenter toutes vos nouvelles fonctionnalités
- Expliquer les points que vous n'êtes pas arrivés à réaliser et pourquoi.
- Expliquer votre démarche de développement.
- Présenter votre structure de données.
- Expliquer comment vous vous y prendriez pour les parties bonus.

## Bonus

- Mettre à jour le gestionnaire de ressources (TP4) pour charger les shaders, les textures et les objets de votre scène.
- Réaliser des calculs parallèles en utilisant le GPU

Dans un précédent TP, nous avons mis en place une chute de pluie, et neige. Vous pouvez maintenant utiliser le GPU pour calculer ces chutes à l'aide d'OpenCL.

Si vous avez également mis en place l'accumulation de particules, vous pouvez également vous servir du GPU afin de réaliser une simulation réaliste de fluide, notamment pour les rivières.