

# GMINT317 - Moteurs de jeux – TP1

## Rendu, gestionnaire de version et IHM

---

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

### Objectifs

Le but de ce TP est de réaliser votre première application de rendu 3D en utilisant OpenGL. Dans ce travail, chaque étudiant aura pour but de réaliser différentes tâches :

- Découvrir les environnements Git, Qt et OpenGL.
- Apprendre à gérer les événements (clavier, souris)
- Afficher une scène simple (une caméra et un triangle)
- Créer une surface carrée à base de triangles
- Appliquer une carte d'altitude sur cette surface
- Permettre les déplacements de caméra
- Avoir « commité » sur le serveur son projet en fin de séance

Bonus :

- Gérer les collisions caméra-terrain
- Texturer le terrain en utilisant des couleurs
- Jouer avec la lumière
- Afficher un brouillard localisé
- Créer un rendu terrain infini

### Framework

Pour nos différents TP, nous allons utiliser le Framework Qt. Il s'agit d'un Framework spécialisé dans les UI. Nous nous servons principalement de ce Framework afin de gérer les IO de notre application.

### Gestionnaire de version

La première étape à réaliser est de créer votre compte sur GitHub, puis de me donner vos identifiants pour que je vous ouvre les droits d'accès au repository du cours.

A partir de là, vous pourrez accéder au repository dans GitHub:

<https://github.com/imagina-gmin317-2015/tp1.git>

Les étapes à suivre seront les mêmes pour chaque tp :

1. Créer un « fork » du repository
2. Créer un « clone » du repository sur votre ordinateur
3. Modifiez les fichiers et faites votre TP.
4. Faites les commits de votre travail pour qu'ils soient visibles sur le repository.

5. Push/sync vos changements sur GitHub

6. A la fin du TP faites un « pull request » pour m'indiquer que je peux vérifier et corriger votre TP.

Vous pouvez maintenant travailler sur votre TP. N'oubliez pas de faire un commit de votre travail en fin de séance, puis à chaque fois que nécessaire jusqu'à la prochaine séance pour que vos modifications soient prises en compte. Chaque commit doit être accompagné d'un message de commentaire.

**Attention :** éviter les messages de commit inutiles, toujours décrire le travail réalisé.

## Votre première application

Afin de vous familiariser avec Qt et de simplifier votre développement, je vous recommande d'utiliser l'IDE fourni : Qt Creator. L'objectif de ce premier TP est principalement de vous familiariser avec cet environnement de programmation.

Pour vos premiers pas, avec OpenGL, votre travail sera d'étudier l'affichage d'un triangle en OpenGL avec Qt et l'étendre :

Ouvrir le projet openglwindow.pro dans QT creator, compiler et exécuter.

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

Après le rendu de votre premier triangle, vous allez devoir réaliser une surface plane (16\*16 sommets) composée de triangles. Grâce aux IO, vous pourrez afficher soit des triangles pleins, soit des triangles en fil de fer.

Une fois cette surface créée, appliquez des transformations aléatoires aux sommets pour réaliser un relief.

## Mouvements de caméra

Afin de vous guider, les interactions avec OpenGL seront réalisées avec les événements Qt, cette stratégie sera utilisée pour déplacer la caméra :

- QMouseEvent
- QKeyEvent

Ces IO permettront de déplacer la caméra dans l'espace : en avant (E), en arrière (D), à droite(F), à gauche (S).

## Cartes d'altitudes

Nous utiliserons ensuite une carte d'altitude (entrée fichier de l'application) grâce aux routines élémentaires de chargement d'une image dans Qt.

Pour chaque sommet de notre maillage de terrain, nous appliquerons un déplacement au sommet en corrélation avec la valeur du pixel correspondant dans la carte d'altitude.

Attention, il faudra augmenter la résolution du maillage à 240 x 240 pour répondre à cette question.

## Compte rendu

Présenter vos fonctionnalités.

Expliquer votre démarche de développement.

Présenter votre structure de données.

Expliquer comment vous vous y prendriez pour les parties bonus.

## Bonus

- Gérer les collisions caméra-terrain

Empêcher la caméra de traverser la surface du sol.

- Texturer le terrain en utilisant des couleurs

Utiliser la coloration de sommets pour texturer votre terrain.

Appliquer des textures en fonction de l'altitude des sommets (blanc pour les sommets très élevés par exemple).

- Jouer avec la lumière

Créer une lumière de type soleil.

Créer des lumières localisées.

Jouer avec les matériaux, les modèles de lumières.

- Afficher un brouillard localisé

En deux étapes, réaliser un brouillard global puis un brouillard localisé.

Utiliser la fonction **glFog()** pour afficher un brouillard.

Afficher un brouillard localisé dans les vallées.

- Créer un rendu terrain infini

Réfléchir à une stratégie multi-résolution du terrain pour permettre un rendu infini.

Générer aléatoirement un rendu cohérent.