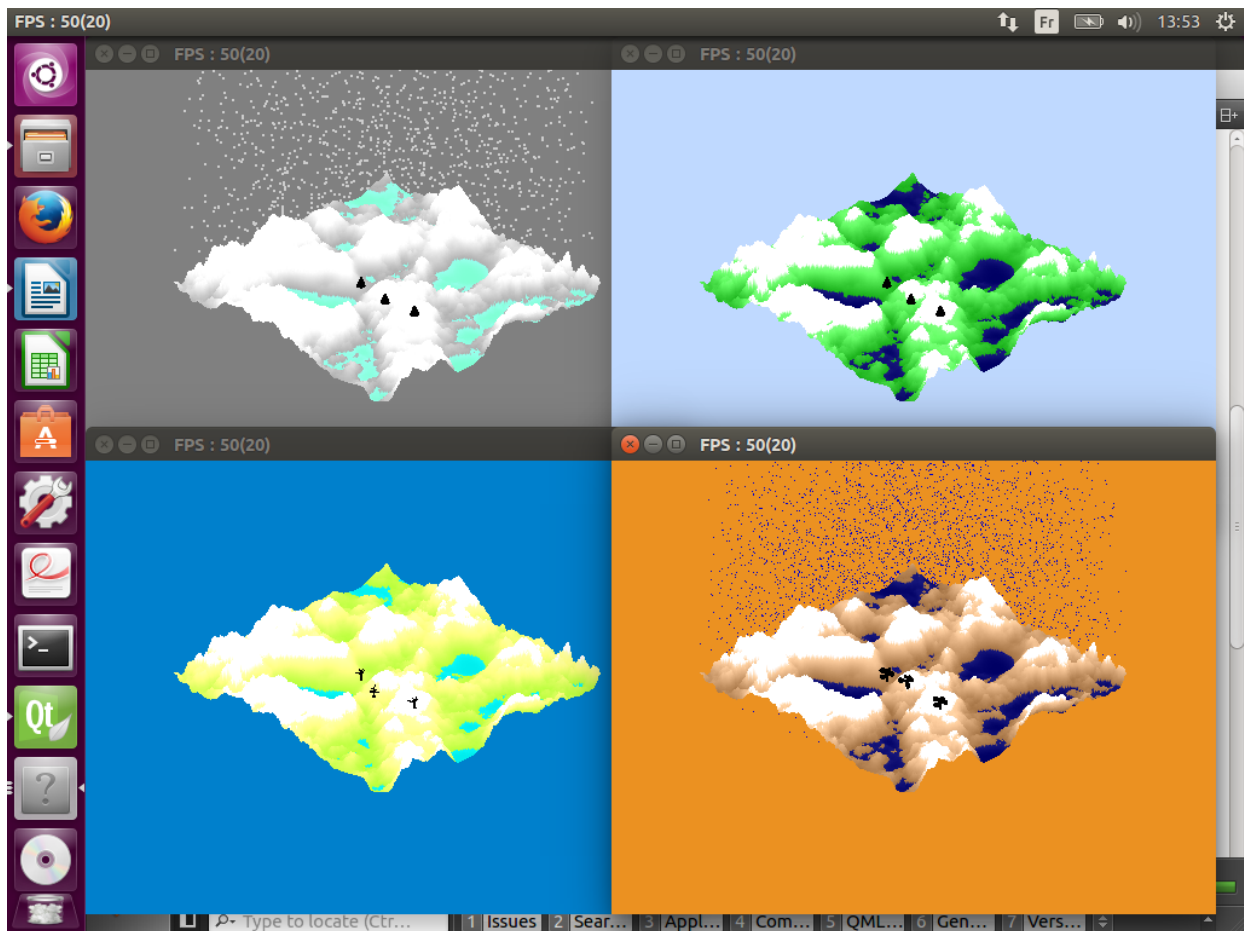


# Compte rendu : TP5 – MOTEUR DE JEUX

Vincent de Rousiers



## Résumé :

- Ajout d'une classe **Texture** afin de gérer le chargement d'un fichier de texture (PNG, JPEG, BMP) et de le stocker en mémoire
- Création de différents shaders (un shader pour la lumière diffuse, un autre pour le sphérique environnement et un dernier pour le normal mapping)

Ce n'est clairement pas mon tp de Moteur de Jeu le plus abouti, je n'ai pas bénéficié d'assez de temps pour faire tout ce que je voulais dans cette UE, d'autres projets devant être rendus entre temps.

## Question 1 :

```
class Texture{
public:
    Texture();
    Texture(QString);

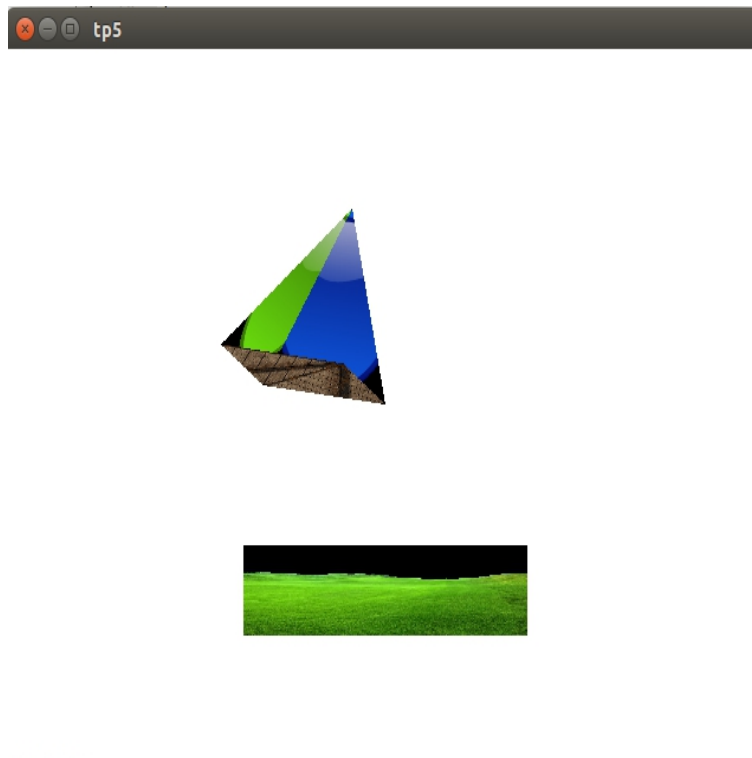
    void initTexture();
    void delTexture();

    QString getNom();
    void setNom(QString);
    GLuint getId();
    void setId(GLuint);

private:
    QString nom;
    GLuint id;
    QImage texture;
};
```

Cette classe va permettre de charger un fichier de texture (l'utilisation de la fonction load de `QImage` permet de gérer un grand nombre de type de fichier) et de le stocker directement en mémoire via la fonction `glBindTexture(GL_TEXTURE_2D, id);`.

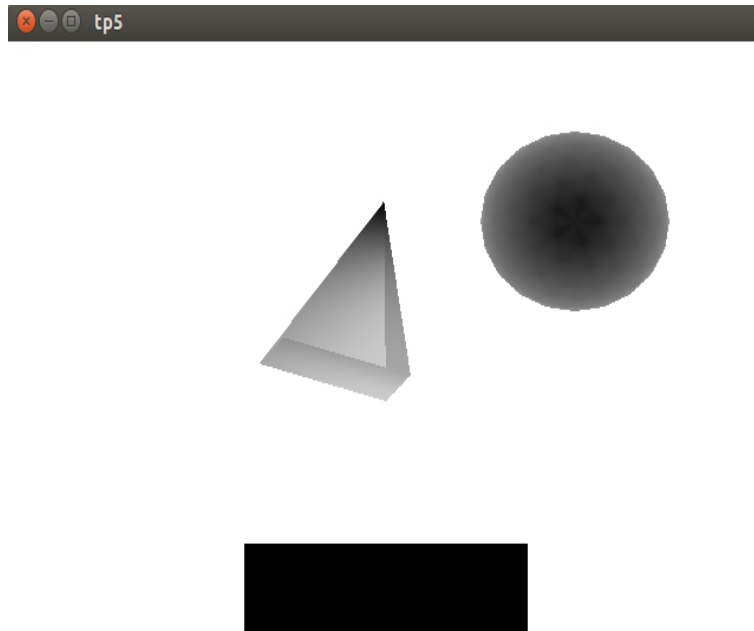
Il ne reste plus qu'au programme à stocker un tableau de `Texture` et de les utiliser via la fonction `void initTexture()` quand on le désire.



## Question 2 :

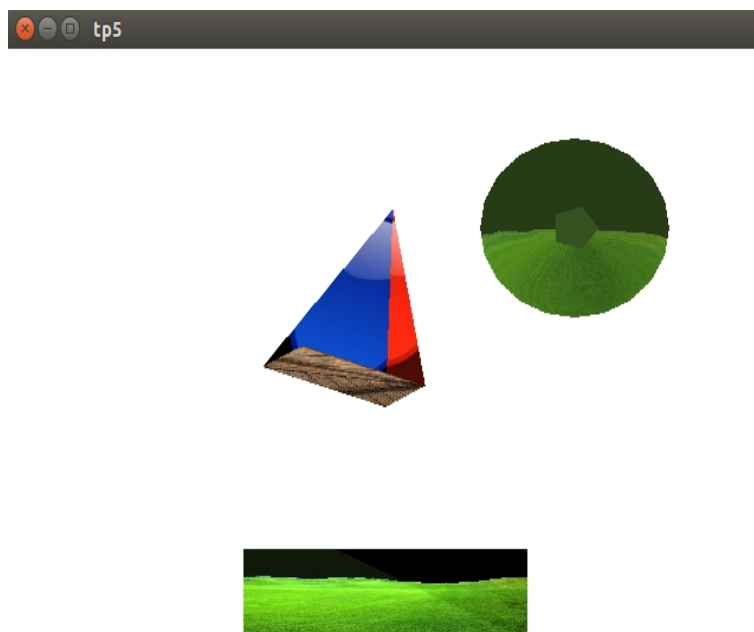
### Diffuse :

Lumière diffuse sans utiliser de texture afin de bien visualiser la répartition de la lumière sur les meshes.



### Spherical environment mapping

La sphère permet de bien visualiser l'effet de sphérique environnement mapping, la réflexion est bien présente.



## Normal mapping

En cours d'implémentation dans le projet de l'UE, la génération de château fort.

## Déformation géométrique

Je n'ai pas implémenté la déformation géométrique par manque de temps.

## Bonus :

- Je n'ai pas mis à jour le gestionnaire de ressources pour les shaders, uniquement pour les textures, par manque de temps.
- Idem pour déporter les calculs vers le GPU, là encore par manque de temps.