

GPGPU

geometry shaders

CPE

5ETI IMI

1 But

L'objectif de ce TP est d'apprendre à utiliser les geometry shaders.

2 Prise en main de l'environnement

2.1 Compilation

Question 1 Compilez le code, assurez-vous de voir un maillage gris.

Question 2 Dans `glhelper.h`, qui reprend le travail effectué en TP1, assurez-vous de comprendre le rôle de chaque fonction.

3 Gestion de plusieurs programmes

Vous pouvez utiliser un `std::vector<GLuint>` pour stocker les différents shaders utilisés par la suite, un compteur permettra de sélectionner le shader voulu.

Question 3 Créez un second programme contenant un fragment shader qui permet d'afficher un maillage avec sa texture.

4 Gestion des uniformes

Question 4 Regardez le passage de paramètres uniformes (paramètre commun à tous le programme GPU) pour la camera. Sur le même modèle, modifiez votre fragment shader pour qu'il évolue dans le temps. Vous pouvez utiliser la librairie `chrono` de la `std`.

```
auto t_start = std::chrono::high_resolution_clock::now();
...
auto t_now = std::chrono::high_resolution_clock::now();
float time = std::chrono::duration_cast<std::chrono::duration<float>>(t_now - t_start).count();
```

Question 5 Ajoutez un shader pour visualiser la texture.

5 Ajout des geometry shaders

Question 6 À la manière de `create_program_from_file(...)`, ajoutez une fonction pour créer un programme avec en plus, un geometry shader.

Question 7 Modifiez le programme principal afin de prendre en compte le geometry shader basic.gs. Compilez, lancez le programme, qu'obtenez-vous, est-ce prévisible ?

Question 8 Créez un geometry shader permettant de créer une vue éclatée de l'objet –il suffit de déplacer les sommets dans le sens de la normale.

Question 9 Créez un geometry shader afin de visualiser les normales sous la forme de ligne par-dessus le maillage. Il vous faudra utiliser deux programmes dans la fonction d'affichage, l'un pour le maillage, l'autre pour les normales.

Question 10 Créez un geometry shader qui calcule la normale à la surface et permet ensuite d'afficher la couleur associée dans fragment shader (passage de paramètre en shaders).

Question 11 Créez un geometry shader qui permet de "gonfler" le maillage : déplacement des sommets dans le sens de la normale de ceux-ci. Est-ce utile d'utiliser un geometry shader ?

Question 12 Créez un geometry shader permettant visualiser non plus des triangles plein mais les lignes des triangles.

Question 13 Créez un geometry shader permettant d'effectuer un "face culling", c'est-à-dire supprimer les triangles dont la normale n'est pas dans le sens de la caméra.

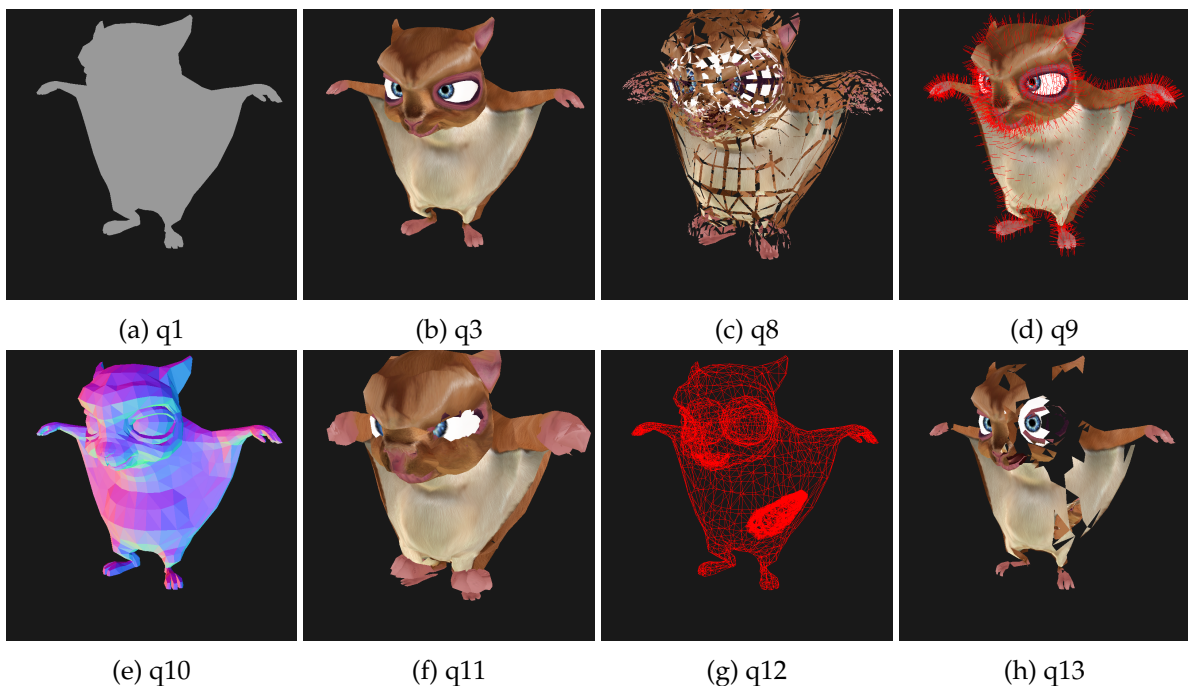


Figure 1: Images attendues pour chaque étape