

POGLA

TP6 : OpenGL 4 - Compute Shaders

Jonathan Fabrizio
LRDE - EPITA

Objectif

L'objectif de ce T.P. est de programmer un *compute shader*. Pour cela, le *compute shader* modifiera le contenu du *VBO* entre chaque rendu.

1 Travail préliminaire

Récupérez le squelette du tp précédent, décompressez l'archive et essayez de comprendre le fonctionnement du programme.

1.1 Un *Compute Shader* minimal

Modifiez le code pour ajouter un *compute shader*. Pour cela vous devrez :

- écrire un *compute shader* mais pour l'instant il ne fera rien (le `main` sera vide toutefois il faudra définir une taille de *local work group*),
- ajouter la compilation du *shader* et l'édition de lien du programme associé,
- invoquer le `glDispatchCompute()` juste avant le `glDraw()` (attention à bien sélectionner à chaque fois le bon programme - `glUseProgram()`).

2 Un lapin boursoufflé !

2.1 Préparez l'animation

Ajoutez un timer dans votre code :

```
void anim() {
    GLint anim_time_location;
    glUseProgram(program_id);
    anim_time_location =
        glGetUniformLocation(program_id, "anim_time");
    glUniform1f(anim_time_location, anim_time);
    anim_time += 0.1;
    // testez une borne max.
    glutPostRedisplay();
}
```

```
void timer(int value) {
    anim();
    glutTimerFunc(33,timer,0);
}
```

```
void init_anim() {
    glutTimerFunc(200,timer,0);
}
```

2.2 Les SSBOs

Vous devez maintenant déclarer deux *SSBOs* dans votre *compute shader*, l'un que vous associez au *VBO* des normales et l'autre que vous associez au *VBO* des sommets. Modifier aussi le code c++ pour faire cette association (c.f. `glBindBufferBase`).

Intuitivement votre *SSBOs* devrait être un tableau de `vec3` :

```
layout(std430, binding = xxx) buffer point_pos_buffer
{
    vec3 point[];
};
```

2.3 Le Compute Shader

Écriture du *Compute Shader*

Écrivez (enfin) le contenu du *compute shader* (c'est pas trop tôt!) pour que le lapin gonfle et dégonfle au cours du temps (en utilisant un `sin` sur `anim_time` et les deux *SSBOs*).

Note : je vous propose de faire gonfler le lapin mais vous êtes libre de faire d'autres modifications, vous pouvez imaginer ce que vous souhaitez.

Correction du *Compute Shader*

Le résultat semble pas tout à fait correct. En effet, le tableau de `vec3` contient du *padding* (comme si vous aviez du `vec4`). Corriger votre *Compute Shader* en conséquence.

