

---

# *TP Informatique Graphique*

<https://github.com/TomClabault/M1-TP-TinyMesh>

---

Tom CLABAULT - p2205453

18 octobre 2022

## **1 Fondamentaux**

mettre côte à cote plusieurs images d'occlusion ambiante pour illustrer le travail -> différents nombre de samples pour le même nombre de vertex et différent nombre de vertex pour le même nombre de sample comparer les perfs des intersections volumétriques avec les intersections de Mesh plot le nombre de triangles en même temps que le temps de calcul sur les plots de génération de primitives

Résultat de l'implémentation des maillages d'une sphere de subdivision, d'un tore, d'une capsule et d'un cylindre :

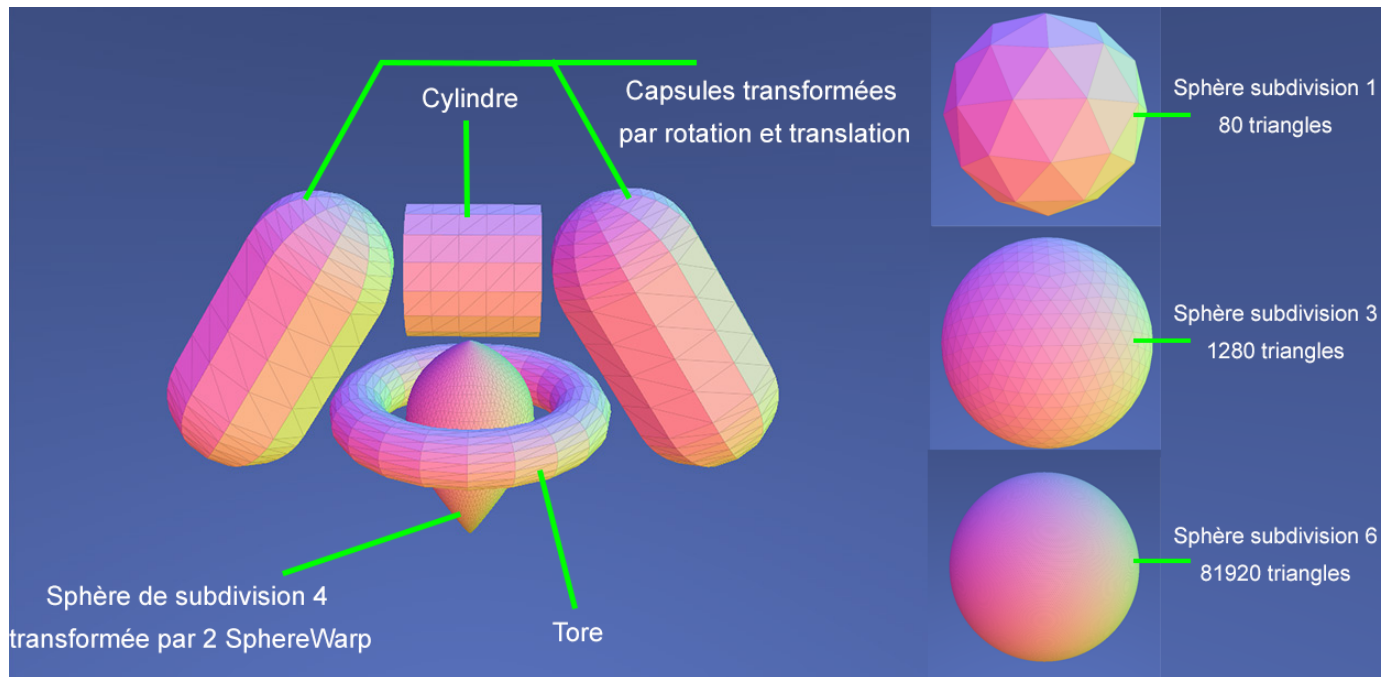


FIGURE 1 – Union de maillages grâce à Mesh : :Merge

### 1.1 Statistiques de génération

Rectifier le tir des maillages simples / lisses Virer les "maillages basiques"

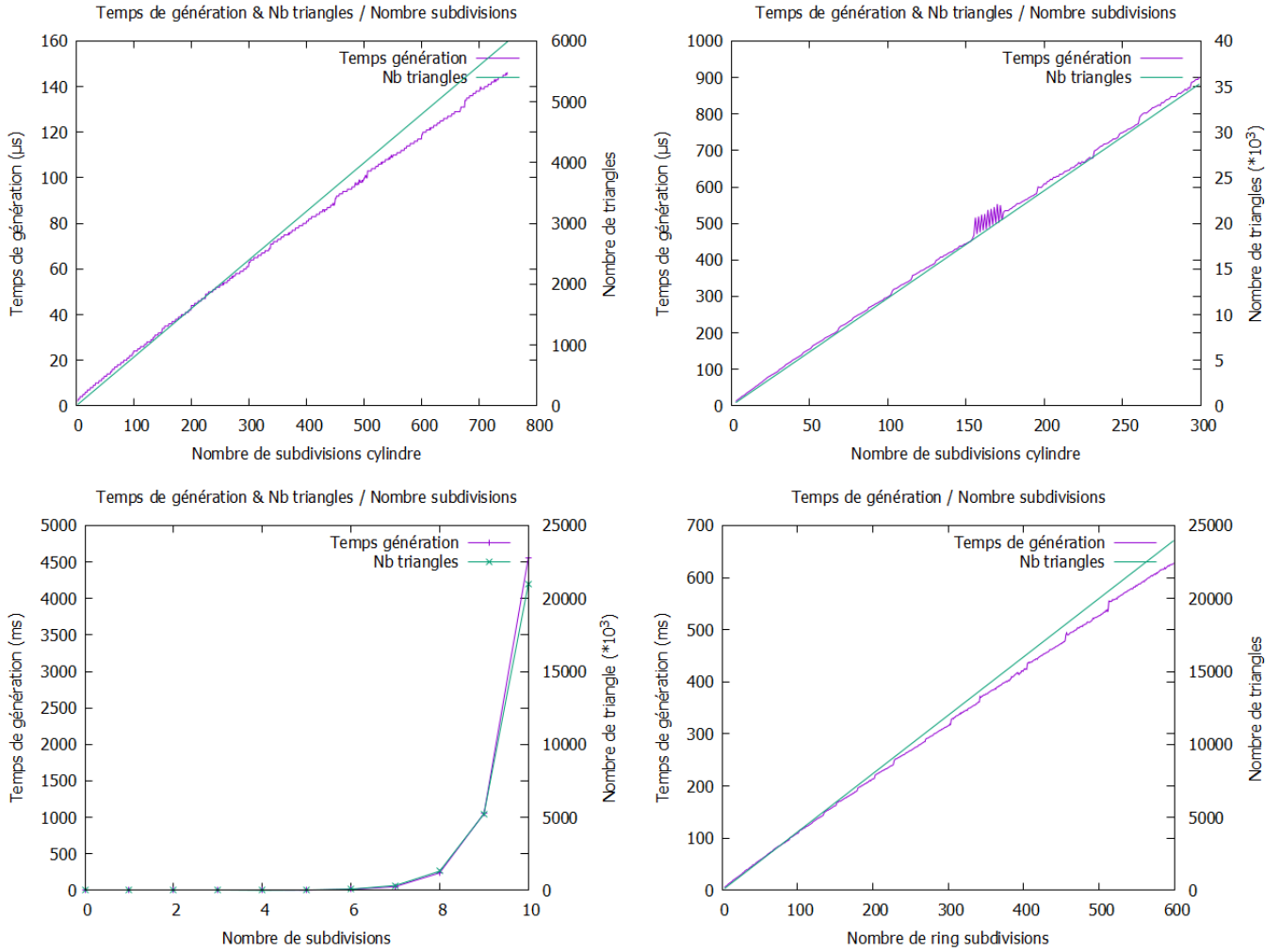


FIGURE 2 – Temps de génération pour les maillages simples implémentés en fonction du nombre de subdivisions

Toutes les primitives à l'exception de l'icosphère évoluent en temps linéaire quand leur nombre de subdivisions augmente et c'est le comportement auquel on pourrait s'attendre. L'icosphère évolue en  $4^n$  avec  $n$  le nombre de subdivisions (le graphique de l'icosphère a son axe des ordonnées en échelle logarithmique base 4). C'est également ce à quoi on s'attend puisque la subdivision dyadique multiplie par 4 le nombre de triangles à chaque subdivision.

Les courbes montrent des "accidents", elles ne sont pas totalement "lisses" et cela est dû à différents aspects techniques tels que la taille des caches du CPU, les réallocations des `std::vector` lors de `push_back()`, ...

## 2 Accessibilité et occlusion ambiante