

MEC1315 TI en ingénierie

Travail Synthèse 1 – Hiver 2024

Cahier des charges

Date de remise : le 19 février 2024 avant 17h00

Équipe de 3 ou 4 étudiants

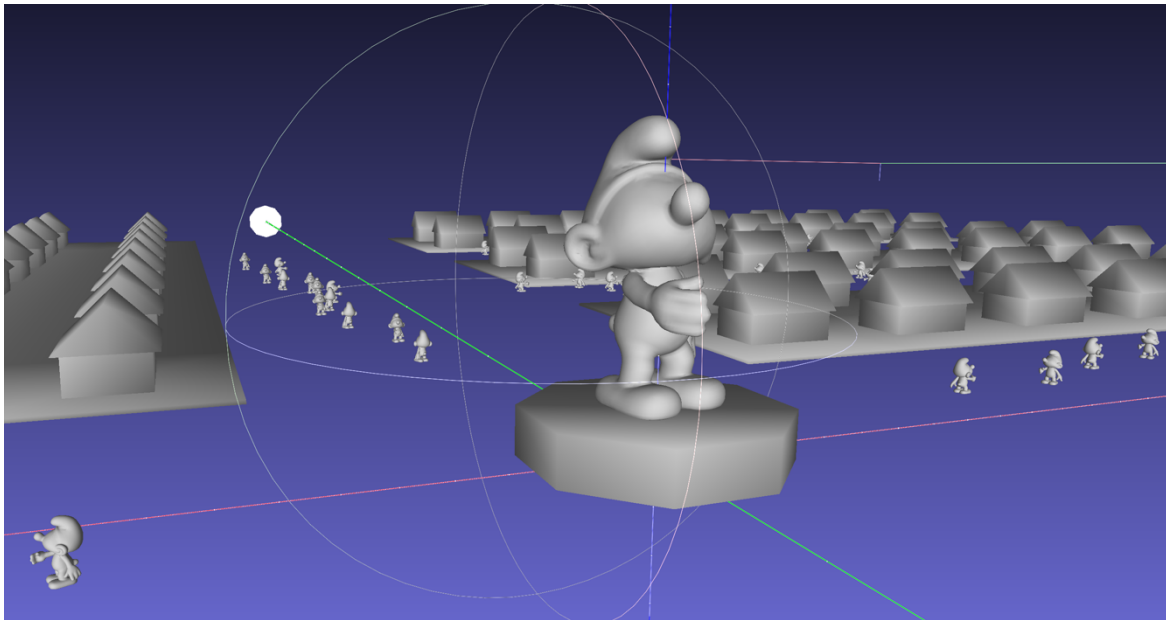


Figure 1 - Scène construite avec des fichiers STL

Polytechnique Montréal

Objectif

Le travail synthèse no. 1 (TS1) porte sur l'utilisation des structures de programmation Python pour effectuer des transformations géométriques 3D d'objets STL, tels que des mises à l'échelle, des translations, des rotations, et des fusions dans le but de construire une scène 3D originale, telle que celle montrée à la figure 1.

1. Objets STL de construction

La scène STL finale doit être construite par votre script Python à partir de 4 objets STL de base et de 1 objet STL original de votre choix. Le script **MEC1315_STL.py** contient des fonctions permettant la lecture de fichiers STL en format ASCII et binaire, ainsi que l'écriture de fichiers STL en format ASCII.

Les 4 objets STL de base

Tel que montré à la figure 2, les objets STL de base sont : 1) un cube; 2) un triangle; 3) un cylindre et 4) un schtroumpf. Le cube est de 1 x 1 x 1 unité (fichier **Cube.stl**). Le triangle est un cube de même dimension, tranché à la verticale en diagonale à 45 degrés (fichier **Triangle.stl**). Le cylindre est un octogone vertical de largeur et hauteur de 1 unité (**Cylindre.stl**). Le schtroumpf (fichier **Schtroumpf.stl**) est un fichier contenant 18246 facettes et 9121 vertex. Chaque objet STL de base doit apparaître, au moins une fois, dans votre scène finale.

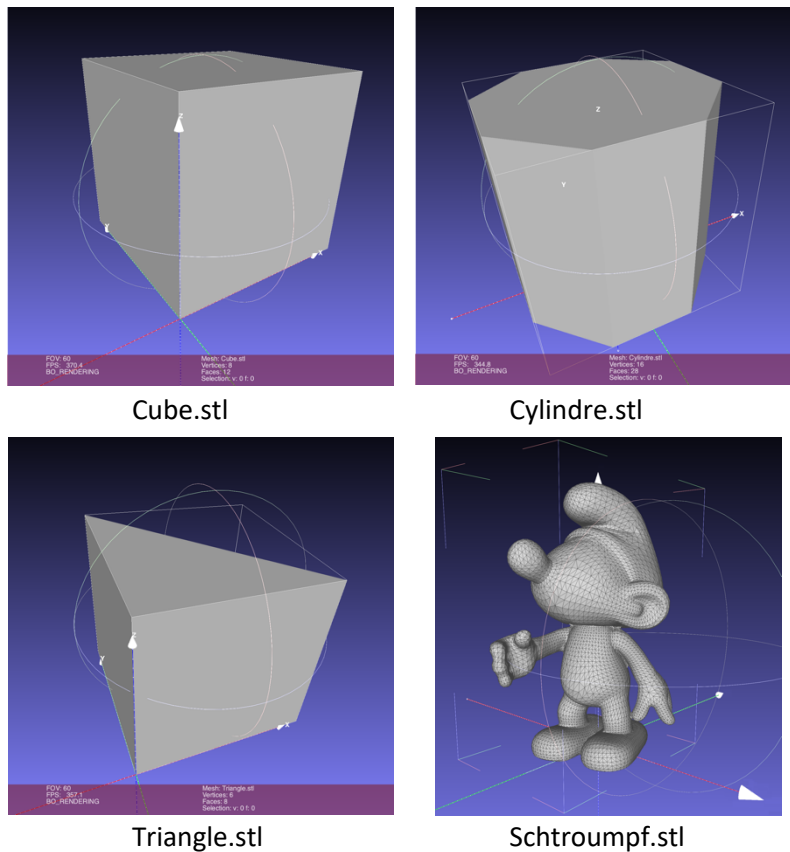


Figure 2 – Objets STL de base

Un objet STL original

Un objet STL original doit être ajouté. Il est possible de modéliser cet objet avec Catia, puis de l'exporter en format STL ou tout simplement télécharger un fichier STL à partir d'un site web, tel que cults3d.com, thingiverse.com, all3dp.com ou autres. Ces fichiers STL possèdent souvent une densité de maillage très élevée (plusieurs millions de facettes). Il est alors nécessaire de réduire cette densité avec un éditeur de maillage, tel que MeshLab (<https://www.meshlab.net/> disponible pour toutes les plateformes). Le temps de construire de votre scène finale ne devrait pas dépasser ~4-5 minutes. Par exemple, la scène de la figure 1 est construite en 10 secondes sur un disque SSD.

2. Script principal Python

Vos scripts Python doivent être concis, structurés et faciles à comprendre par un collègue, comme le script principal de la figure 3. Vous devez préparer des fonctions permettant de manipuler des objets STL. Vos fonctions Python doivent permettre d'effectuer les transformations géométriques 3D, tels que la mise à l'échelle, la translation et la rotation d'objet STL. Vous devez préparer 3 fonctions de répétition (rectiligne, circulaire et personnalisée) en utilisant des boucles Python (while, for, ...). Finalement, une fonction de fusion d'objet STL est nécessaire. Une répétition personnalisée se fait en spécifiant l'objet STL à répéter et le nom d'un fichier (.txt) contenant les points d'insertion (x,y,z) et l'orientation (angle) des instances. D'autres fonctions peuvent être nécessaire, afin de faciliter la création de votre scène 3D.

Attention, la programmation orientée objet n'est pas permise!

```
# MEC1315: Exemple de Travail Syntèse 1
# Transformations géométriques 3D d'objet STL
import numpy as np
from MEC1315_STL import *
from MesFonctions import *

#Construire le voisinage
maison = Construire(30,20,10,60,60) # Maison 30pi x 20pi x 10pi
voisinage = RepetitionRect(maison,'x', 8, 60) # terrain 60pi x 60pi
voisinage = RepetitionRect(voisinage,'y', 2, 60)
voisinage = RepetitionRect(voisinage,'y', 3, 180)
voisinage = Translation(voisinage,[85, 100, 0])
voisinage = RepetitionCirc(voisinage,'z', 4, 90)

Schtroumpf = LireSTL('Schtroumpf.stl')
Schtroumpf = Centrer(Schtroumpf,'x')
Schtroumpf = Centrer(Schtroumpf,'y')
Schtroumpf1 = MiseEchelle(Schtroumpf,0.25,0.25,0.25)
groupe = RepetitionPerso(Schtroumpf1,'data.txt',True)
groupe = RepetitionCirc(groupe,'z', 4, 90)

...

#Fusion et export du fichier STL
scene=Fusion(voisinage,socle,statut,groupe)
f,v,n=scene[0],scene[1],scene[2]
nom_fichier='Scene.stl'
EcrireSTLASCII(nom_fichier, f, v, n)
```

Figure 3 – Exemple d'un script principal en Python

3. Remise du TS1

Chaque équipe doit remettre sur Moodle avant le 19 février 2024 à 17h00 un fichier ZIP (nommé **TS1_Equipe_xx.zip**) d'un maximum de 20 Mo contenant :

- 1- votre script principal (**Equipe_xx.py**);
- 2- votre scène STL finale (**Scene_xx.stl**);
- 3- votre capture de la scène finale sur fond blanc (**Capture_xx.png**) selon les spécifications;
- 4- le % de contribution de chaque membre de l'équipe (**Contribution_xx.txt**);
- 5- vos fichiers secondaires **.py** , **.txt** et **.stl** (incluant votre objet STL original).

4. Directives et Barème d'évaluation

Ce travail synthèse 1 (TS1) compte pour 20% de la note globale du cours et doit être fait en équipe de 3 ou 4 étudiants selon les directives suivantes :

- 1- Scène finale originale qui inclus chacun des 4 objets de base et l'objet original (20%);
- 2- Scène finale qui montre les 3 types de répétition d'objets (5%);
- 3- Scripts Python fonctionnels, reconstruction de la scène finale en moins de 4 minutes (pénalité);
- 4- Scripts Python concis, structurés et faciles à comprendre par un collègue (15%);
- 5- Ensemble de fonctions Python permettant la manipulation directe d'objet STL (10%);
- 6- Qualité des fonctions de transformation géométrique (translation, rotation et fusion) (20%);
- 7- Qualité des 3 type de répétitions (rectiligne, circulaire et personnalisée) (30%);
- 8- Utilisation optimale des listes de valeurs et des boucles de contrôle Python (pénalité);
- 9- Respect des directives sur les documents à remettre (pénalité et voir barème préliminaire).

Les équipes de moins de 3 étudiants, non autorisées par l'enseignant, reçoivent une pénalité de -30%. Le correcteur ne fait qu'extraire votre contenu du fichier Zip et exécute et il ne modifiera pas le contenu, si l'exécution échoue faute d'oubli de fichier, votre scripte sera considéré non fonctionnel et recevra des pénalités.

5. Capture de votre scène finale avec Meshlab

Afin de créer une compilation vidéo des scènes 3D de toutes les équipes, il est demandé de remettre une capture sur fond blanc (fichier image **Capture_xx.png**), telle que montré à la figure 4.

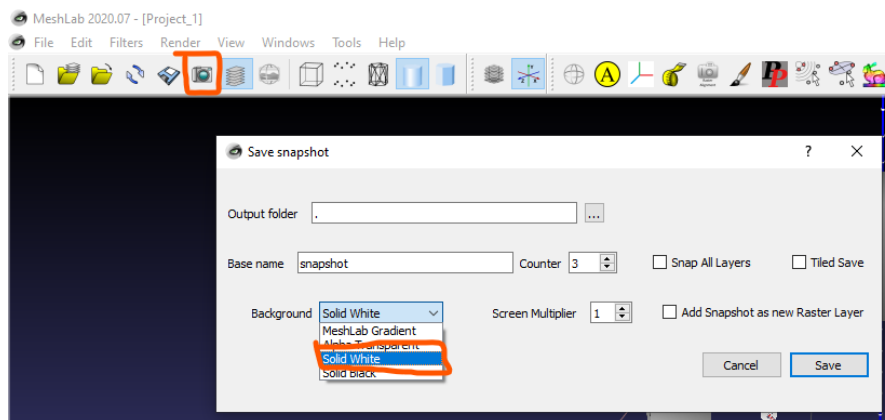


Figure 4 – Sauvegarde de mise en scène en format PNG