

BACCALAURÉAT

SESSION 2026

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°9

DURÉE DE L'ÉPREUVE : 1 heure

**Le sujet comporte 4 pages numérotées de 1 / 4 à 4 / 4
Dès que le sujet vous est remis, assurez-vous qu'il est complet.**

Cette situation d'évaluation comporte ce document ainsi que des fichiers de codes et de données présents sur l'ordinateur à la disposition du candidat. Le candidat doit restituer ce document avant de sortir de la salle d'examen. Le candidat doit agir en autonomie et faire preuve d'initiative tout au long de l'épreuve.

En cas de difficulté, le candidat peut solliciter l'examineur afin de lui permettre de continuer la tâche. Des moments privilégiés pour solliciter l'examineur sont indiqués dans le document sous la forme d'appels professeur.

L'examineur peut intervenir à tout moment, s'il le juge utile.

Les fichiers OBJ constituent l'un des formats les plus utilisés dans le domaine de la modélisation 3D.

Étant donné un repère orthonormé, ils servent à décrire une forme géométrique en listant ses sommets (les points dans l'espace) et ses faces (les surfaces reliant ces points). Grâce à ce principe, il est possible de représenter aussi bien des objets simples, comme un cube, que des modèles complexes issus de logiciels de modélisation.

Ce format est particulièrement apprécié parce qu'il est simple à lire, facile à manipuler et compatible avec la majorité des outils 3D. Cette simplicité vient du fait qu'un fichier OBJ n'est rien d'autre qu'un fichier texte, dans lequel on décrit ligne par ligne les différents éléments : le nom de l'objet (o), les sommets (v), les faces (f)...

Voici un exemple du contenu d'un fichier OBJ permettant de construire une seule face d'un cube.

```
o cube
v 0.0 0.0 0.0
v 0.0 1.0 0.0
v 1.0 1.0 0.0
v 1.0 0.0 0.0
f 1 2 3 4
```

Figure 1 – Contenu d'un fichier OBJ

Dans ce sujet, on ne considère que des coordonnées entières.

Représentation de l'information

Pour manipuler ces fichiers OBJ, nous avons à disposition trois classes Python :

- la classe `Sommet` : elle représente un sommet, défini par trois entiers précisant les coordonnées x, y et z du sommet ;
- la classe `Face` : elle représente une face d'un élément 3D, définie par une liste d'indices. Cet indice identifie un sommet présent dans la liste des sommets d'un `Objet3D` ;
- la classe `Objet3D` : elle représente un élément 3D, défini par une liste d'objets de type `Sommet` et une liste d'objets de type `Face`.

Chaque classe est définie dans son propre module Python. Ainsi, la classe `Sommet` est définie dans `Sommet.py`.

Calcul du volume d'un élément 3D

On s'intéresse au calcul du volume d'un objet afin d'obtenir une estimation du temps d'impression à l'aide d'une imprimante 3D. Pour cela, on va chercher l'arête la plus longue présente sur une de ses faces et approcher le volume de l'objet par celui d'un cube de même arête.

Pour calculer la longueur d'une arête, il suffit de calculer la distance entre les deux sommets qui la composent.

Question 1

Écrire la méthode `distance` de la classe `Sommet`. Elle prend en paramètre un objet de type `Sommet`. La méthode renvoie la distance entre l'objet courant et l'objet de type `Sommet` passé en paramètre.

On rappelle que la distance entre deux points A et B dans un repère choisi s'obtient grâce à la formule

$$\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}$$

avec $(x_A; y_A; z_A)$ les coordonnées du point A et $(x_B; y_B; z_B)$ les coordonnées du point B . Pour calculer le résultat d'une racine carrée en Python, on peut utiliser la fonction `sqrt` du module `math`.

Pour obtenir l'arête la plus longue, on va considérer tous les couples de sommets formant une arête et calculer le couple de plus grande longueur.

Quand deux sommets sont les arêtes d'une face, on dit qu'ils sont adjacents.

Question 2

Écrire la méthode `sommets_adjacents` de la classe `Objet3D` qui prend en entrée deux points donnés par leurs coordonnées et renvoie `True` s'ils représentent une arête de l'objet 3D et `False` sinon.

Attention, on considère que les arêtes ne sont pas ordonnées, ainsi `sommets_adjacents(s1, s2) == sommets_adjacents(s2, s1)`.



Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

On dispose d'une méthode `volume_cube_englobant` dans la classe `Objet3D` qui utilise la fonction précédente pour estimer le volume d'un objet 3D.

Dans le fichier `imprimante3D.py`, on définit une classe `Imprimante3D` qui contient deux attributs :

- l'attribut `remplissage` précise le taux de remplissage, flottant entre `0.0` et `1.0`, du plastique lors de l'impression.
- l'attribut `vitesse_extrusion` précise la vitesse d'impression de l'imprimante. Elle est représentée sous la forme d'un entier, précisant la vitesse en mm^3/s .

On considère que l'unité géométrique est de 1mm. Ainsi, le sommet de coordonnées (2, 0, 0) est à 2mm de l'origine du repère.

Question 3

Écrire une méthode `estimation_impression` pour la classe `Imprimante3D` qui prend en paramètre un `Objet3D` et renvoie une estimation de son temps d'impression en secondes en procédant ainsi :

- Dans un premier temps, il faut calculer le volume d'impression de l'objet, c'est-à-dire en multipliant le volume réel par le taux de remplissage.
- Dans un second temps, il faut calculer le temps d'impression en secondes. On rappelle que le temps s'obtient en divisant le volume d'impression de l'objet par la vitesse d'extrusion de l'imprimante.

Manipulation d'objets 3D

On souhaite agrandir ou rétrécir un objet 3D selon un rapport (agrandissement ou réduction). On dispose dans la classe `Objet3D` de la méthode `transformer`, qui permet de transformer l'instance courante selon le rapport passé en paramètre.

Question 4

Utilisez cette méthode pour doubler la dimension du cube présent dans `Objet3D.py` en l'affichant avant et après l'appel. Le cube n'apparaît pas deux fois plus grand, analyser le fonctionnement de la méthode `transformer` et proposer une correction.



Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

Description du dossier

Le dossier fourni au candidat sur l'ordinateur comporte les éléments suivants :

- une version PDF de l'énoncé ;
- Plusieurs fichiers de classes Python : `Objet3D.py`, `Face.py`, `Sommet.py`, `Imprimante3D.py`

Préparation de l'environnement

Pour faire fonctionner le code fourni dans le dossier, les bibliothèques suivantes doivent être présentes : `matplotlib`, `math`.