

Sapinisateur

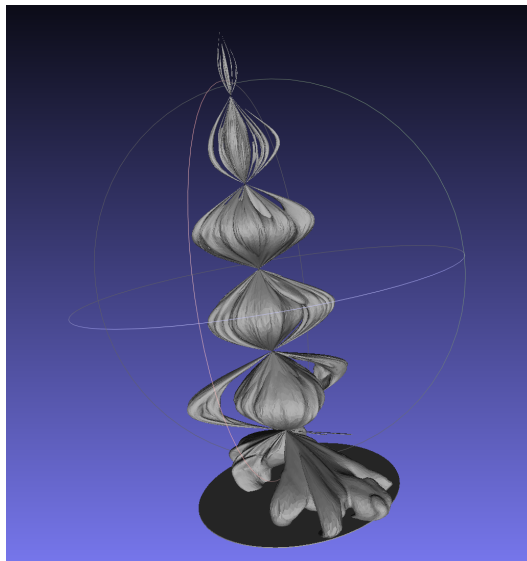
Bases de la programmation impérative

4 décembre 2017

Bienvenue dans le projet de BPI. Ce projet est à réaliser par équipe de un ou deux étudiants. Il n'est pas noté et ne sera pas corrigé individuellement mais il y aura une correction globale et vous pouvez bien sûr discuter entre étudiants des difficultés rencontrées ainsi qu'avec vos enseignants.

Même si le projet n'est pas noté, nous vous demandons de respecter les règles usuelles pour les projets qui sont les suivantes :

- rendre le projet à temps sur TEIDE ;
- on peut discuter entre étudiants mais pas montrer ou donner du code.



Pour ce projet, on se propose d'écrire un filtre prenant en entrée un modèle 3D au format STL (binaire), le modifiant avant de le re-sauvegarder. On cherche à transformer le modèle en sapin de Noël.

1 Le format STL

Le format STL est un format très utilisé pour les imprimantes 3D. Il s'agit en fait de deux formats : un format ascii et un format binaire.

On travaillera ici uniquement avec le format binaire.

Voici un pointeur sur une documentation du format. https://fr.wikipedia.org/wiki/Fichier_de_st%C3%A9r%C3%A9olithographie

À vous de jouer ! L'objectif du projet est de pouvoir gagner en autonomie.

En python, la lecture de fichiers binaires sera réalisée à l'aide du module *struct*.

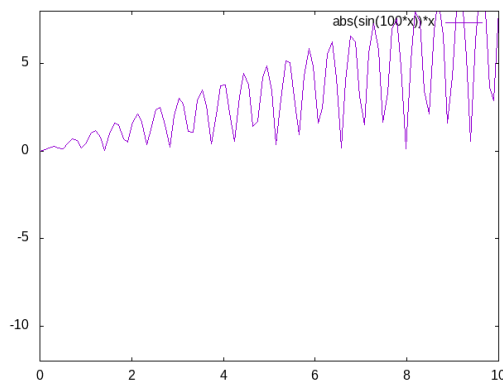
Notez que les normales aux surfaces ne sont en général pas nécessaires et peuvent être simplement mises à zéro lors de la sauvegarde du fichier. En revanche l'ordre des sommets de chaque triangle est important car il détermine l'orientation de la facette.

Nous vous conseillons d'utiliser le logiciel *meshlab* pour la visualisation des fichier STL. Il existe également de nombreux autres visualiseurs.

2 Sapinisateur

Une fois le fichier chargé, on cherche à le sapiniser.

Pour ce faire, nous allons chercher une forme naturelle de sapin. Considérons par exemple la fonction $f(x) = x|\sin(s \times x)|$. On distingue bien un demi sapin.



On va donc transformer notre modèle 3D en réalisant une déformation du modèle à l'aide de notre fonction sapin. L'algorithme est donné en pseudo-code figure 1.

Vous êtes bien entendu libre de paramétrer la fonction à votre guise ou même de fournir une fonction différente.

Nous vous fournissons quelques fichiers de tests mais de nombreux fichiers sont disponibles sur <http://www.thingiverse.com>. À vous de voir sous quelles conditions les résultats peuvent s'avérer probants.

```

on calcule un centre approximatif  $c$  comme la moyenne des points;
soit  $g$  une fonction d'agrandissement multipliant les coordonnées  $x$  et
 $y$  d'un point par un facteur donné ( $z$  ne changeant pas);
pour chaque point  $p$  faire
    | soit  $z_p$  sa hauteur relative (plutot en bas, mi-hauteur, ...);
    | on calcule un facteur d'agrandissement  $a_p = f(z_p)$ ;
    | soit  $p_2 = g(p - c, a_p) + c$ ;
    | remplacer  $p$  par  $p_2$ ;
fin

```

FIGURE 1 – Algorithme de sapinisation

3 Attendu

On se place dans un contexte professionnel et on cherche donc à nettoyer son/ses fichier(s), à proposer une interface sympathique en ligne de commande...

Le rendu est fixé au vendredi 22 décembre 2017.