

Informatique graphique

Le projet devra être codé en C++. Un code de maillage et de visualisation avec Qt6 est accessible sur <https://perso.liris.cnrs.fr/eric.galin/teaching.html>. Quatre sujets sont au choix, traiter **deux sujets au minimum** (deux rendus), traiter trois sujets permet de viser une très bonne note.

Le rendu contiendra un lien vers l'archive contenant le code, un PDF expliquant les classes implémentées et des images des maillages générés avec les paramètres de génération, et surtout **une analyse critique**.

Modélisation à l'aide de Blobs

L'objectif est de modéliser des Blobs définis par des nuages de points.

Concepts : surfaces implicites, arbres de construction.

Modèle : créer un Blob permettant de traiter un grand nombre de primitives simples (point, segment). Créer différentes formes et les visualiser à l'aide de l'algorithme de maillage, comparer la qualité de maillage obtenu. Généraliser de manière à traiter des primitives ayant différentes textures. Calculer la borne de Lipschitz d'un Blob de différentes manières.

Performance : comparer les performances selon le nombre de primitives. Comparer les Blobs de Wyvill à support compact et les Blobs de Blinn à support infini en termes de performance.

Modélisation à l'aide de distances signées

L'objectif est de modéliser des SDF à l'aide d'un arbre de construction hiérarchique.

Concepts : surfaces implicites, arbres de construction.

Modèle : créer une hiérarchie de nœuds N permettant de représenter des primitives (sphère, capsule, tore, boîte) et binaires (mélange, union, intersection). Implémenter pour chaque classe une fonction d'évaluation de type `double Node::Value(const Vector&) const`. Créer différentes formes, les visualiser à l'aide de l'algorithme de maillage, comparer la qualité de maillage obtenu.

Conversion : si les Blobs sont également implémentés, proposer un nœud permettant de convertir un Blob en primitive SDF.

Performance : comparer les performances des différentes primitives pour l'évaluation de 10^7 appels à la fonction de distance signée.

Génération procédurale de modèles représentés à l'aide surfaces implicites

L'objectif est de simuler le vieillissement d'objets modélisés à partir de SDF.

Concepts : intersection avec une droite.

Intersection : dans la classe représentant la surface implicite écrire une fonction de type `bool Signed::Intersect(const Ray&,double&) const` en utilisant l'algorithme de Sphere Tracing.

Vieillessement : écrire un algorithme érodant objet en projetant des sphères qui modifieront localement la surface. Comparer les performances lorsque l'érosion est faite de manière incrémentale, ou par paquets.

Accélération des requêtes

L'objectif est d'accélérer les traitements grâce à des fonctions Lipschitziennes par morceaux.

Concepts : volumes englobants, accélération de calcul.

Modèle : définir un nœud définissant un volume V englobant un sous arbre N , lui-même sous la forme d'une SDF, permettant d'accélérer les calculs de la distance $d(\mathbf{p}, N)$.

Performance : Comparer la performance du calcul de la distance signée et de l'intersection avec un rayon pour un objet en utilisant des volumes englobants ou non.