

TP4 - Hiver 2019

IMN 428

Infographie

Objectifs

1. Afficher des courbes paramétriques cubiques. En particulier, Bézier, Hermite, Catmull-Rom et B-Spline uniforme non-rationnelle.
2. Manipuler les courbes de Bézier et d'Hermite en fonction d'une continuité donnée (C0,G1 ou C1).
3. Gérer la précision des courbes.

Description

Partant d'un code fourni à l'avance, vous devez rédiger les fonctions permettant d'afficher une courbe (paramétrique ou non) avec ses points de contrôle. Vous devez également écrire le code nécessaire à la manipulation des points de contrôle et ce, en respectant la continuité choisie par l'utilisateur (C0,G1 ou C1 pour les courbes de Bézier et Hermite). Voir les endroits contenant le commentaire

AJOUTER CODE ICI dans le fichier tp4.cpp.

Pour effectuer ce TP, il vous est INTERDIT d'utiliser les fonctions d'OpenGL gérant automatiquement les courbes paramétriques comme `glMapX`, `glEvalCoordX` et `gluNurbsX`. Il revient à vous d'afficher les courbes cubiques à l'aide de l'algorithme de traçage de courbes par intervalles réguliers vu en classe.

Affichage des points de contrôle

Les points de contrôle sont contenus dans le tableau global `ControlPointArray`. Ce tableau contient les points que l'utilisateur a entré en cliquant dans la fenêtre à l'aide du bouton gauche de la souris.

Affichage des courbes

Pour afficher une courbe, il vous faut implanter l'équation paramétrique $P = TMG$ vu en classe où $T = (t^3, t^2, t, 1)$, M est la matrice 4×4 propre à la courbe (Bézier, Hermite, Catmull-Rom ou BSpline) et G est le vecteur de paramètres. La courbe est affichée en rouge et d'épaisseur 2.0. Dans le cas de la courbe d'Hermite, veuillez afficher les vecteurs R_1 et R_2 en bleu. L'utilisateur doit également pouvoir ajuster la précision des courbes à l'aide des touches "p" et "P" du clavier ainsi que la continuité avec la touche "c".

NOTE : considérant quatre points consécutifs p_a, p_b, p_c et p_d , les contraintes utilisées pour afficher une courbe d'Hermite sont les suivantes :

$$\begin{aligned} P_1 &= p_a \quad /* \text{ point de départ } */ \\ P_2 &= p_d \quad /* \text{ point d'arrivée } */ \\ R_1 &= 3.0(p_b - p_a) \quad /* \text{ tangente au point de départ.} */ \\ R_2 &= 3.0(p_d - p_c) \quad /* \text{ tangente au point de d'arrivée.} */ \end{aligned}$$

Évaluation

Ce travail doit être fait par **équipe de TROIS**. Au moment de soumettre votre travail, assurez-vous que votre code compile bien sous *Microsoft Visual Studio*. Utilisez le **turnin WEB** pour soumettre votre travail : <http://opus.dinf.usherbrooke.ca/>