

IMN428 – Infographie

Travail pratique 4

Objectifs

1. Affichage des points de contrôle (fonction `drawControlPoints()`)
2. Interpolation linéaire (fonction `drawLines()`)
3. Implantation des courbes de Bézier (fonction `drawBezierCurves()`)
4. Implantation des courbes B-Splines (fonction `drawBSplineCurves()`)

Interpolation linéaire

Vous devez relier les points de contrôle par des segments de droite.

Courbes de Bézier

Vous devez implanter l'équation

$$\mathbf{p}(u) = \sum_{i=0}^n \mathbf{P}_i B_{i,n}(u)$$

C'est donc dire que le degré de votre courbe dépendra du nombre de points affichés : si vous avez n points, votre courbe sera de degré $n - 1$. Notez que l'exécutable fourni souffre de quelques problèmes d'instabilité numérique lorsque le nombre de points de contrôle est élevé.

Courbes B-Splines

Vous devez implanter les équations des courbes B-Splines pour $k = 3$ et $k = 4$, qui sont respectivement

$$\mathbf{p}_i(u) = \frac{1}{2} \left((1-u)^2 \mathbf{P}_i + (-2u^2 + 2u + 1) \mathbf{P}_{i+1} + u^2 \mathbf{P}_{i+2} \right)$$

et

$$\mathbf{p}_i(u) = \frac{1}{6} \left((-u^3 + 3u^2 - 3u + 1) \mathbf{P}_{i-1} + (3u^3 - 6u^2 + 4) \mathbf{P}_i + (-3u^3 + 3u^2 + 3u + 1) \mathbf{P}_{i+1} + u^3 \mathbf{P}_{i+2} \right)$$

Vous n'êtes pas tenus de gérer les extrémités des segments. Pour $k = 3$, vous pouvez vous en tenir à $i \in [1 : n - 1]$ et $i \in [1 : n - 2]$ pour $k = 4$.

Précisions

Dans les deux cas, la variable *curvePrecision* sert à déterminer la précision de la courbe. Tant pour les courbes de Bézier que pour les B-Splines, le paramètre u varie dans l'intervalle $[0, 1]$. *curvePrecision* doit déterminer le nombre de points qui seront véritablement évalués dans cet intervalle.

Évaluation

Ce travail doit être fait par **équipe de TROIS ou QUATRE**. Au moment de soumettre votre travail, assurez-vous que votre code compile bien sous Visual Studio. Utilisez la commande **turnin** pour soumettre votre travail. La remise doit être faite au plus tard le **16 avril 2012**.