

Enseignant(s)

BIANCHINI Marc

Email(s)

mbianchi@myges.fr

Courbe de Bézier

1 Matières, formations et groupes

Matière liée au projet :

Formations : -

Nombre d'étudiant
par groupe :

1 à 4

Règles de constitution des groupes: **Libre**

Charge de travail
estimée par étudiant :

7,00 h

2 Sujet(s) du projet

Type de sujet : **Imposé**

3 Détails du projet

Objectif du projet (à la fin du projet les étudiants sauront réaliser un...)

Engendrer des courbes de Bézier et des raccordements de courbes de Bézier.

Utiliser une structure adaptée pour la gestion des courbes.

Appliquer des transformations géométriques par les matrices.

Utilisation d'OpenGL sous C++.

Pour un groupe de 4 hypothétique, faire une visionneuse graphique qui gère la sauvegarde de fichier ainsi que les entrées et les sorties 2D. Adapter le logiciel pour la partie 3D les extrusions graphiques du projet suivant.

Descriptif détaillé

Rajouter ce projet au fenêtrage et remplissage vus au S1. Faire un menu pour toutes les fonctionnalités implémentées.

Ce projet consistera à:

- 1) Cliquer $n+1$ points de contrôle à l'écran et afficher le polygone de contrôle
- 2) Afin d'engendrer la courbe de Bézier d'ordre n associé à ce nuage de points, faire deux méthodes:
 - a) utiliser la formule directe avec le calcul des combinaisons à l'aide du triangle de Pascal (vu en complément de cours)
 - b) utiliser l'algorithme de DeCasteljau, celui du cours en version itérative

Dans les deux cas, tracer la courbe à l'écran avec un pas fixé et comparer les temps de calculs pour un nombre important de points (> 50)

Remarque importante: à l'aide de segments, il faudra relier 2 à 2 tous les points consécutifs de la courbe de Bézier.

- 3) Possibilité de modifier le pas, avec les touches + et - afin de pouvoir (ou non) lisser la courbe
- 4) Pouvoir engendrer un nombre illimité de courbes, pour cela utiliser une liste chaînée
- 5) Pouvoir parcourir la liste et supprimer n'importe quelle courbe
- 6) Pouvoir déplacer et supprimer un ou plusieurs points de contrôle et afficher en temps réel la courbe
- 7) Modifier les courbes: utiliser les matrices de translation, scaling, rotation et cisaillement sur les points de contrôle d'une Bézier et tracer en temps réel la courbe en fonction des points de contrôle modifiés. Effectuer une gestion à la souris, éventuellement clavier.
- 8) Effet de la multiplicité d'un point de contrôle: répéter un point de contrôle dans la liste des points de contrôle et vérifier que la courbe se rapproche de ce point
- 9) Implémenter une méthode pour calculer l'enveloppe convexe d'un nuage de points (ex: méthode de Jarvis). Calculer les enveloppes convexes des lignes polygonales de contrôle de 2 Bézier. Tester si les deux enveloppes s'intersectent. Si c'est le cas calculer le point d'intersection éventuel entre les 2 Bézier
- 10) Effectuer des découpages et remplissages, via les algo vus au S1, de courbes de Bézier fermées (le premier point de contrôle est confondu avec le dernier + éventuellement faire du raccordement C_0 , C_1 et C_2 pour fermer la courbe sans point anguleux)
- 11) Raccorder des Courbes de Bézier, faire les raccords C_0 , C_1 et C_2 .
- 12) Construire au moins 10 courbes de Bézier cubiques raccordées C_2 entre elles. Vérifier le contrôle local de la courbe.
- 13) Bonus1: Engendrer des courbes Bsplines cubiques raccordées C_2 par l'algorithme de Cox-de-Boor. Vérifier que le contrôle local fonctionne.
- 14) Bonus2: Engendrer des courbes NURBS (classe mère de toute courbe)

Ouvrages de référence (livres, articles, revues, sites web...)

Outils informatiques à installer

OpenGL, visual C++

4

Livrables et étapes de suivi

1	Etape intermédiaire	Constitution groupes	mardi 04/03/2025 19h00
2	Rendu final	Présentation et commentaire du code Pas de ppt Quest/Réponse tour par tour	mercredi 21/05/2025 19h00

5 Soutenance

Durée de présentation par groupe : **15 min** Audience : **A huis clos**

Type de présentation : **Démonstration**

Précisions :