

# Compte Rendu TP3

Hadrien Marquez

08/05/2017

## Installation

In the project directory :  
Run 'cmake CMakeLists.txt' to create makefile.  
Run 'make' to build project.  
Run './bin/toto' to start the application.

## Table des matières

<b>1</b>	<b>Consigne</b>	<b>1</b>
<b>2</b>	<b>Structure du programme</b>	<b>1</b>
2.1	Point . . . . .	1
2.2	Courbe . . . . .	1
2.3	CourbeLin . . . . .	1
2.4	CourbeSCR . . . . .	2
2.5	CourbeBezi . . . . .	2
<b>3</b>	<b>Tests</b>	<b>3</b>
3.1	Sur les Points . . . . .	3
3.2	Le nombre de Point . . . . .	3
3.3	La précision paramétrique . . . . .	3
3.4	La tension de Catmull-Rom . . . . .	3

## 1 Consigne

Le but de l'exercice était de dessiner des courbes dans un fichier eps. Pour cela, il fallait rentrer des points critiques et effectuer une interpolation entre chacun de ces points. Plusieurs méthodes d'interpolation devaient être proposées dans le programme. Cette exercice avait pour objectif de renforcer notre compréhension de concepts objets tels que l'héritage et le polymorphisme. L'architecture et la logique interne du programme étant donc des points importants.

Nous ne détaillerons pas les trois méthodes d'interpolations ici. Il fallait produire des courbes linéaires, de Catmull-Rom ou bien de Bézier.

## 2 Structure du programme

Pour cet exercice une courbe étant une suite de points, nous avons réutiliser notre classe Point. Ensuite nous sommes partis du principe que même si elles utilisaient des méthodes différentes, une courbe restait une courbe. Nous avons donc créé une classe général Courbe qui ensuite se spécifie suivant la méthode à utiliser. Au total nous avons : une classe Point, une classe Courbe et trois classes héritant de Courbe. Nous avons décidé que la fonction de dessin n'appartiendrait pas à une classe.

### 2.1 Point

La classe Point, est un point 2-D composé de deux coordonnées (x,y). Elle est identique (ou presque) aux classes Point de nos TP précédents.

### 2.2 Courbe

La classe Courbe est une classe abstraite (non-instanciable) caractérisant de manière générale ce qu'est une Courbe. Il s'agit donc d'une suite de points critiques avec pour chacun d'eux une abscisse paramétrique. La classe délivre aussi des méthodes communes qui seront utiles pour le calcul des points interpolés. Elle définit aussi une méthode virtuelle pure, `calcQ`. Elle sera remplie dans les classes héritantes de Courbe. On y trouvera l'algorithme de calcul de la courbe à dessiner. On remarque que c'est cette méthode qui rends la classe abstraite.

### 2.3 CourbeLin

Il s'agit de la première spécification de Courbe. Elle hérite de Courbe. Sa différence réside dans la méthode de calcul `calcQ`, qui renvoie un tableau contenant tous les points de la courbe à dessiner suite à une interpolation linéaire entre points critiques. La figure 1 montre le dessin d'une courbe linéaire.

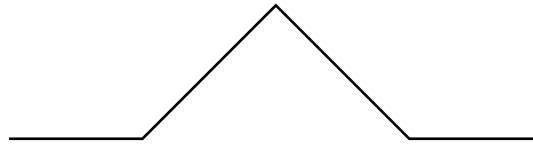


FIGURE 1

## 2.4 CourbeSCR

Deuxième spécification de Courbe. Elle hérite de Courbe. En plus de sa propre méthode de calcul `calcQ`, la courbe de Catmull-Rom introduit un paramètre de tension. Celui-ci servira lors de l'interpolation à rendre la courbe plus ou moins "molle". On a aussi rajouté une méthode nommée `calcMatriceCR`, qui ne sert qu'à rendre plus lisible `calcQ`. Son intérêt est donc limité. La figure 2 montre le dessin d'une courbe de Catmull-Rom.

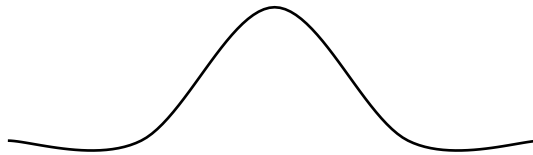


FIGURE 2

## 2.5 CourbeBezi

Dernière spécification de Courbe. Hérite de Courbe. La méthode de calcul `calcQ` contient l'algorithme pour une interpolation de Bézier. En plus, il a fallu surcharger la fonction d'initialisation de courbe de manière à ce que l'utilisateur ne rentre que des paquets de quatre points. La figure 3 montre le dessin d'une courbe de Bézier.

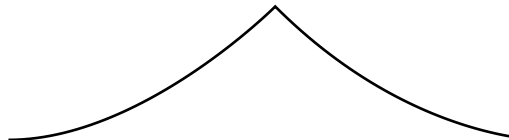


FIGURE 3

### 3 Tests

Nous avons effectué une batterie de test pour découvrir les limites du programmes. Pour avoir un dessin probant, il ne faut pas utiliser des valeurs petites. Le résultat (même si fonctionnelle) est tout de même décevant. Voici quelques tests qui ont fait échouer notre programme.

#### 3.1 Sur les Points

Rentrer des points non-conforme, c'est à dire autres choses que des coordonnées chiffrées, provoque une suite de 0 sur toutes les coordonnées de point suivants.

#### 3.2 Le nombre de Point

Rentrer un nombre de point non-conforme, c'est à dire non-chiffrées, ou égale à 0 provoque le lancement d'une exception et la fin du programme.

#### 3.3 La précision paramétrique

Rentrer une précision paramétrique non-conforme, c'est à dire non-chiffrées, ou égale à 0 provoque un 'freeze' du programme. Il faut alors sortir du programme manuellement avec Ctr+C.

#### 3.4 La tension de Catmull-Rom

Une tension inférieure à 0 et supérieure à 1 provoque des bizarreries comme le montre les images suivantes. La figure 4 montre une courbe avec  $tension = -1$ . La figure 5 montre une courbe avec  $tension = 10$ .



FIGURE 4



FIGURE 5