

# Compte Rendu TP2

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	Polygone . . . . .	1
2.3	Vec3 . . . . .	1
<b>3</b>	<b>Tests</b>	<b>1</b>

# 1 Consigne

Le but de l'exercice était de calculer l'aire d'un polygone non croisé. Encore une fois, cette excercice est une excuse pour développer notre compréhension de la programmation objet. Le calcul de l'aire du polygone se fait à partir d'une somme de produits vectoriels.

## 2 Structure du programme

Ici, trois objets sont nécessaires pour le bon fonctionnement du programme. Tout d'abord, il nous faut pouvoir utiliser des points, qui composeront le polygone. Le polygone aussi doit être un objet. Finalement, nous allons faire des produits vectoriels, il est intéressant d'avoir une classe représentant un vecteur. En plus de ces classes vient la boucle principale, le main, contenant les instructions du programme.

### 2.1 Point

La classe Point, est un point 3-D composé de trois coordonnées  $(x,y,z)$ . Elle est très identique à la classe Point utilisée dans le TP1. La seule différence provenant de la troisième coordonnée. On a donc un accesseur (en lecture) en plus, ainsi que de légère modification de méthodes pré-existante.

### 2.2 Polygone

La classe Polygone est caractérisé par la collection de point le composant. Elle possède des méthodes pour ajouter un ou plusieurs points à la figure. Elle peut aussi calculer son aire, renvoyant alors le résultat.

### 2.3 Vec3

Afin de calculer l'aire du polygone il nous fallait faire un produit vectoriel. Nous avons donc créer cette classe comme outil pour la classe Polygone. La classe Vec3 est un vecteur trois coordonnées  $(x,y,z)$ . Il possède plusieurs méthodes utiles pour le calcul de l'aire tel qu'une méthode de produit vectoriel ou bien une méthode pour normer les coordonnées du vecteurs. En plus, il possède des méthodes pour en faciliter l'utilisation tel que des accesseurs (en lecture) ou bien une surcharge sur l'opérateur de flux de sortie '«'. On a également surchargé l'opérateur '\*' pour que celui-ci fasse automatiquement appel à notre méthode de produit vectoriel.

## 3 Tests

Nous avons effectué une batterie de test pour découvrir les limites du programmes.

Nous avons essayé l'aire d'un carré de côté 1, puis de côté 10 000, le résultat est bon. De même nous avons essayé diverses triangles, chacun d'eux donnant un résultat juste. De même les polygones construits avec des valeurs négatives donnent des résultats cohérent. Nous nous sommes arrêté à des polygones composés de six points avec cette méthode.

Le seul test raté est quand nous avons soumis une lettre à la place d'une valeur numérique. Le point étant alors ramené directement à l'origine (0,0,0). Si l'utilisateur a spécifié plusieurs point à rentrer, alors tout les points non-saisie sont aussi ramené à l'origine. Le calcul se fait avec tout les points, ceux à l'origine compris.