*HMIN321: Scociété Virtuelle*

Rendu : Argumentation

Tianning MA

M2 IMAGINA

10/11/2020

Table de matière

[1. Introduction 2](#_Toc53497535)

[2. Rendu et explication des exercices 2](#_Toc53497536)

[Question 1 & 2 : La structure de 2-G-carte et les parcours de la structure 2](#_Toc53497537)

[Question 3 : Le plongement géométrique 3](#_Toc53497538)

[Question 4 : La couture (lier deux éléments) 3](#_Toc53497539)

[Question 5 : Caractéristique d’Euler-Poincaré 5](#_Toc53497540)

[Question 6 : Visualisation 6](#_Toc53497541)

# Introduction

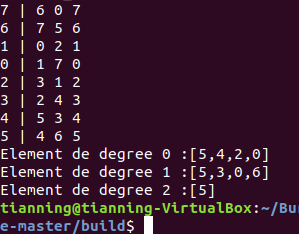
Ce compte rendu est dédié au TP Carte combinatoire.

J’ai rencontré les diffcultés lors de la visualisation et un peu bloqué dessus. Donc je n’ai pas réussi à partir de la question 6.

# Rendu et explication des exercices

## Question 1 & 2 : La structure de 2-G-carte et les parcours de la structure

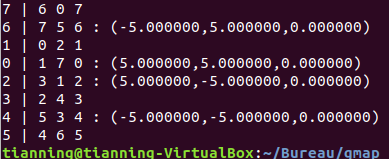
L’affichage du résultat est comme l’image ci-dessous.



On constate que pour chaque brin, la liste de l’alpha relation est affichée.

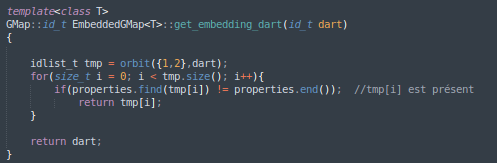
Les brins sont liées l’un et les autres, et la fonction éléments détermine les i-cellules de la carte.

## Question 3 : Le plongement géométrique



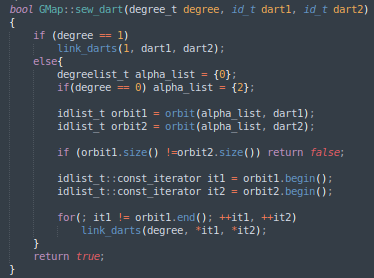
Après avoir implémenté la fonction « get\_embedded\_dart », j’ai obtenu le résultat d’affichage comme l’image ci-dessus.

La fonction que j’ai implémenté est comme l’image ci-dessous :



Pour chaque brin de l’orbit 1,2, s’il possède une propriété dans « properties » de classe EmbeddedGMap, on le retourne, sinon retourne le brin d’entré.

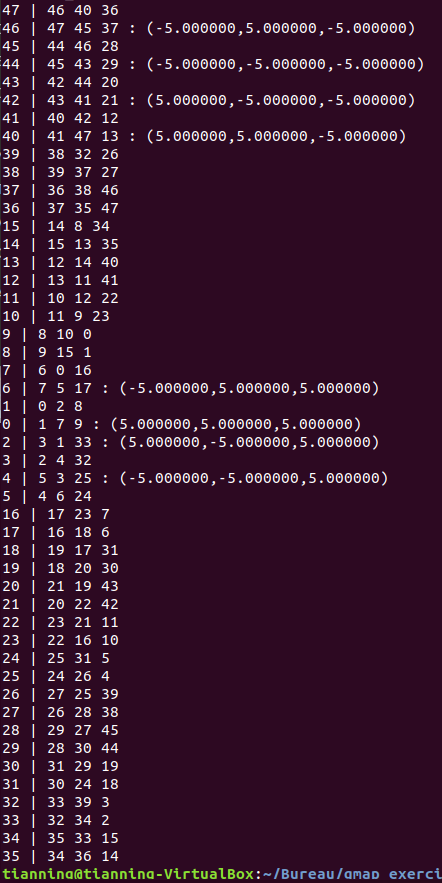
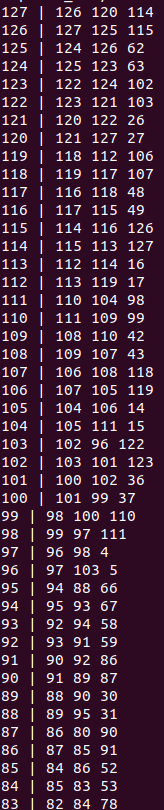
## Question 4 : La couture (lier deux éléments)

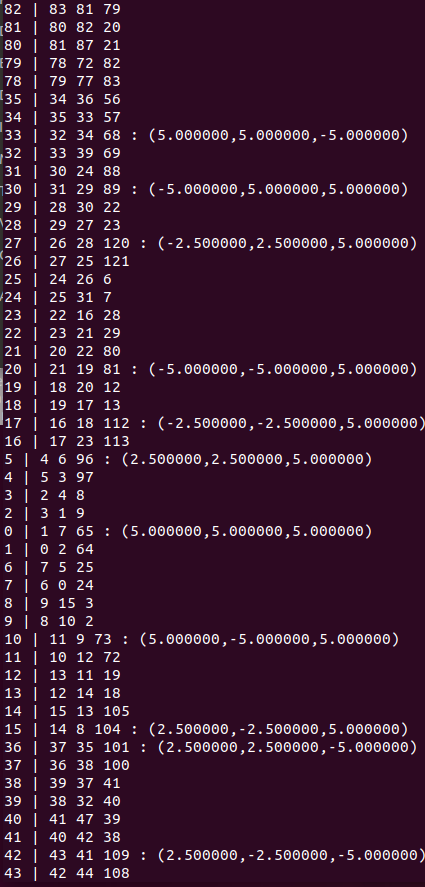


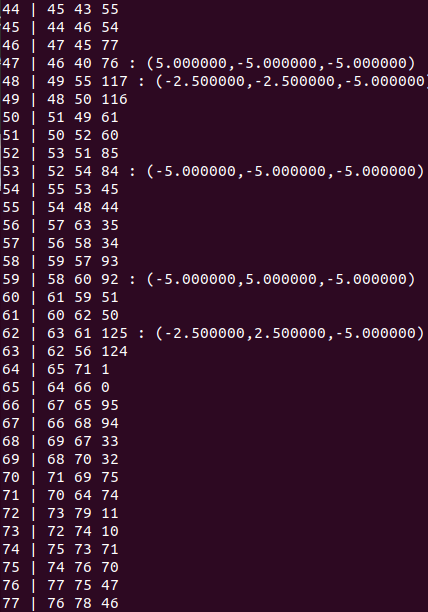
Pour la coutures des éléments, Tout abord j’ai récupéré les orbites de degree « degree » pour les deux brins.

Ensuite, j’ai fait lier les deux éléments une fois il sont compatibles (de meme taille) (en utilisant la fonction link\_dart).

Le résultat de « question4a » et « question4b » est comme les images suivantes :

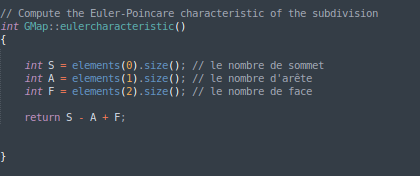
Question 4a : Question 4b :

Question 4b (suite)

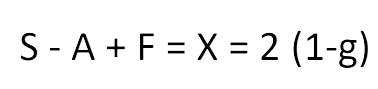


## Question 5 : Caractéristique d’Euler-Poincaré

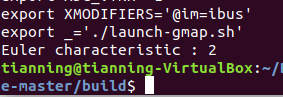
Grâce à la fonction « elements() » dans la classe Gmap, on peut obtenir le nombre de sommet, d’arête et de face.



Selon la formule d’Euler-Poincaré, la relation entre ces trois éléments sont comme suivante :



Donc pour un cube, on a bien le Euler characteristic = 2



Comme indiqué après l’exécution de la question5()

## Question 6 : Visualisation

Lors de la visualisation, il consiste à transformer Gmap en un vector de vertex / triangles etc.

Tout d’abord, pour les vertexs, cela est relativement évident. Ce sont les positions dans Gmap, et qui correspondant les coordonées des vertex. Donc on pourra sauvegarder dans le vector de index\_vertices.



Ensuite pour les indices, après plusieurs essaies, je n’ai toujours pas trouvé une méthode pour chercher les brins correspondant puis récupérer les indices. Pour avoir une affichage, (plus précisément pour un cube), j’ai juste tapé manuellement pour le vector d’indice pour avoir un affichage de cube.

(j’ai aussi modifié la position de caméra en (0.0f, 0.0f, 6.0f) pour avoir un rendu plus complèt) 