



## **Description des modules et leurs interactions**

### **Plan :**

- a. Présentation des différents modules
- b. Présentation du contenu des modules
- c. Présentation des interactions entre modules



## I. Présentation des différents modules

### **1. MODULE\_GLOBALE :**

Ce module a pour objectif principal de définir la structure ville, il sert de fichier principal contenant toutes les bibliothèques et contient les fonctions globales utiles à tous les autres modules. Par exemple, le module MODULE\_DIJKSTRA utilise la fonction distance pour calculer la distance entre 2 villes qui est contenu dans ce module.

### **2.MODULE\_FICHER :**

Ce module a pour objectif principal de traiter le fichier(En format .csv) et d'écrire les X plus grandes villes en France sur Google Map.

### **3.MODULE\_DIJKSTRA :**

Ce module a pour objectif principal de réaliser le plus court chemin entre 2 villes en France sur Google Map, en utilisant l'algorithme DIJKSTRA

### **4.MODULE\_ARBRE :**

Ce module a pour objectif principal de réaliser l'arbre aérien couvrant minimal entre 2 villes et de l'afficher sur Google Map, en utilisant l'algorithme Prim.

### **5.MODUE\_APPENDICE :**

Ce module est destiné à faciliter la saisie des villes lorsque utilisateur choisit la ville de départ et la ville de destination.

### **6.MODULE\_ANTL(R) :**

L'objectif de ce module est d'écrire une grammaire qui, après être générée par ANTLR, transforme un fichier XML en fichier de format csv.

---

## II. Présentation du contenu des modules

### **1.MODULE\_GLOBALE**

```
#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <string.h>
#define PI 3.1415926
0)---struct ville{
    char nom_ville[100];
    int pop;
    float latitude;
    float longitude;
};
typedef struct ville Ville;

1)----float change_format(char* str);
2)----float distance(Ville ville1, Ville ville2);
3)----Ville* tri_ville(Ville *ville, int flag, int nombre_de_ville);
4)----Ville* x_premieres_villes(Ville *ville_triee, int x);
5)----void affiche_tableau(Ville* ville, int n);
```

### **2.MODULE\_FICHER**

```
#include "MODULE_GLOBALE.h"

1)----FILE* lire_fichier(char* path_fichier);
2)----Ville* read_ville(FILE* fp, int* nombre_de_ville);
3)----void ecrire_fichier(Ville *ville, FILE *fp_origine_html, FILE *fp_new_html, int x);
```

### **3.MODULE\_DIJKSTRA**

```
#include "MODULE_FICHER.h"
#ifndef MODULE_DIJKSTRA_H_INCLUDED
#define MODULE_DIJKSTRA_H_INCLUDED
1)----float* creer_table_distance(Ville ville_depart, Ville *ville, int nombre_de_ville);
//crée la table distance pour calculer DIJKSTRA
2)----int* creer_table_marque(int nombre_de_ville);
//crée la table marque pour DIJKSTRA
```

```

3)----int* creer_table_pere(int nombre_de_ville);
//crée la table père pour DIJKSTRA
4)----int extraire_min(float *table_distance, int *table_marque, int nombre_de_ville);
//retourne le numéro de ville à extraire..
5)----int* extraire_update_table_marque(float *table_distance, int *table_marque, int
nombre_de_ville);
//met à jour la table_marque après extraire_min
6)----float *relacher_table_distance(float *table_distance, int x, int y, Ville *ville);
// modifie la table distance avant de la renvoyer
7)----int* relacher_table_pere(float *table_distance, int *table_pere, int X, int Y, Ville *ville);
// modifie la table père avant de la renvoyer
8)----int toutes_villes_extraites(int *table_marque, int nombre_de_ville);
//est-ce que toutes les villes sont extraites ?
9)----int cherche_chemin_1(Ville *ville, FILE *fp_new_html, int *table_pere, int
No_de_depart, int No_de_dest, int i);
//fonction antérieure dans ecrire_fichier_chemin
10)----int cherche_chemin_2(FILE *fp_new_html, int *table_pere, int No_de_depart, int
No_de_dest, int i);
//fonction antérieure dans ecrire_fichier_chemin
11)----void ecrire_fichier_chemin(Ville *ville, FILE *fp_origine_html, FILE *fp_new_html_2,
int *table_pere, int No_de_depart, int No_de_dest);
//écrit la solution de DIJKSTRA sur le Google Map

```

## 4.MODULE\_ARBRE

```

#include"MODULE_FICHER.h"
#ifndef MODULE_ARBRE_H_INCLUDED
#define MODULE_ARBRE_H_INCLUDED
1)----float* creer_table_prim_distance(Ville *ville, int nombre_de_ville);
//crée la table_prim_distance pour calculer Prim
2)----int* creer_table_prim_marque(int nombre_de_ville);
//crée la table_prim_marque pour Prim
3)----int* creer_table_prim_pere(int nombre_de_ville);
//crée la table_prim_pere pour Prim
4)----int extraire_prim_min(float *table_prim_distance, int *table_prim_marque, int
nombre_de_ville);
//retourne le numéro de ville à extraire..
5)----int* extraire_update_table_prim_marque(float *table_prim_distance, int
*table_prim_marque, int nombre_de_ville);
// met à jour la table_prim_marque après extraire_min
6)----int tous_villes_extraire_prim(int *table_prim_marque, int nombre_de_ville);
//est-ce que toutes les villes sont extraites ?
7)----float*relacher_table_prim_distance(float *table_prim_distance, int
*table_prim_marque, int X, int Y, Ville *ville);

```

```
// modifie la table table_prim_distance avant de la renvoyer
8)----int* relacher_table_prim_pere(float *table_prim_distance, int *table_prim_marque,
int *table_prim_pere, int X, int Y, Ville *ville);
/// modifie la table table_prim_pere avant de la renvoyer
9)----void ecrire_fichier_couvrant_minimal(FILE *fp_origine_html, FILE *fp_new_html_3, int
*table_prim_pere, int N, Ville *x_plus_grand_villes);
//écrit l'arbre couvrant minimal sur Google Map
```

## **5.MODULE\_APPENDICE**

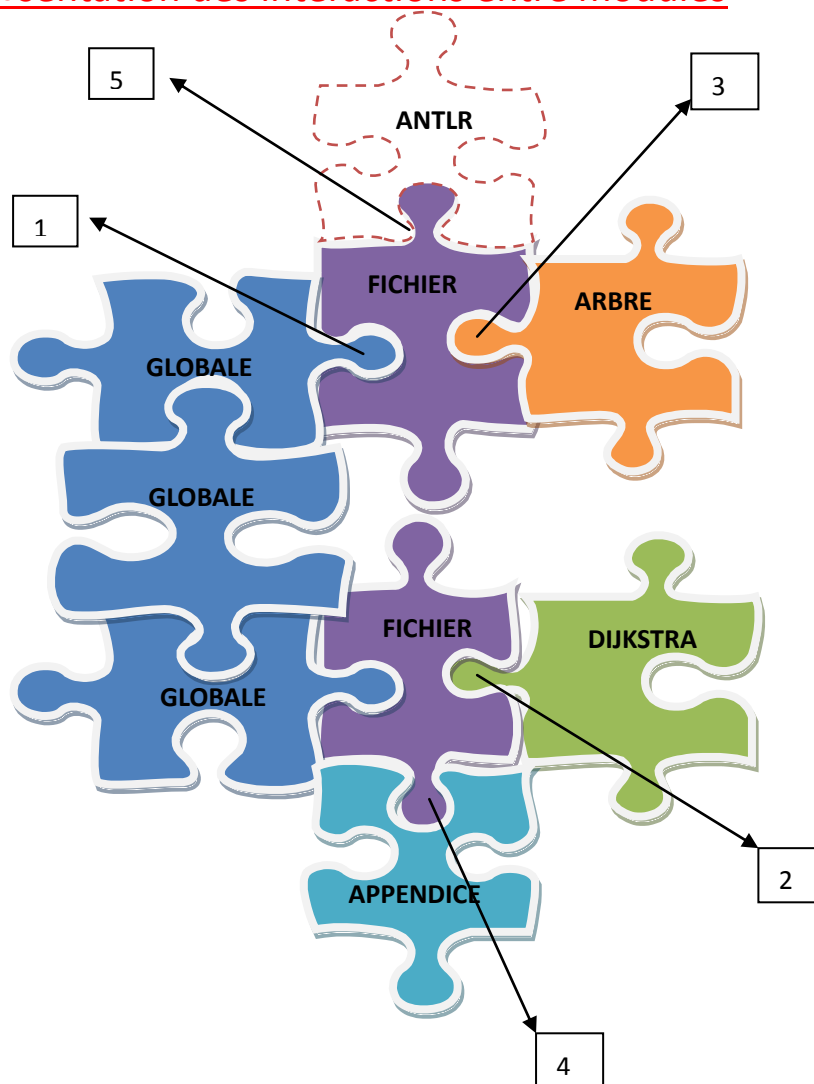
```
#include"MODULE_FICHER.h"
1)----Ville choisir_ville_depart(char *path_fichier_ville);
//choisir la ville de départ
2)----Ville choisir_ville_dest(char *path_fichier_ville);
//choisir la ville de destination
```

## **6.MODULE\_ANTL**

### **Grammaire**

---

### III. Présentation des interactions entre modules



#### **Explications :**

1. Entre le module **MODULE\_GLOBALE** et **MODULE\_FICHIER** :  
Le module MODULE\_FICHIER est utilise principalement la fonction change\_format et la structure de Ville présente de MODULE\_GLOBALE.
2. Entre le **MODULE\_FICHIER** et **MODULE\_DIJKSTRA**  
Dans la fonction MAIN, MODULE\_DIJKSTRA utilise les fonctions lire\_fichier et read\_ville hérité du MODULE\_FICHIER, et distance, tri\_ville, x\_premieres\_villes héritées du MODULE\_GLOBALE.
3. Entre le **MODULE\_FICHIER** et **MODULE\_ARBRE**  
Dans la fonction MAIN, MODULE\_ARBRE utilise les fonctions lire\_fichier et read\_ville provenant du MODULE\_FICHIER, et distance, tri\_ville, x\_premieres\_villes héritées du MODULE\_GLOBALE.

4. Entre le **MODULE\_FICHIER** et **MODULE\_APPENDICE** :

Le module MODULE\_APPENDICE utilise les fonctions lire\_fichier et read\_ville héritées de MODULE\_FICHIER.

5. Entre le **MODULE\_FICHIER** et **MODULEANTLR** :

Après avoir correctement analysé le fichier xml, le programme le renvoi en format csv, et les autres modules peuvent ensuite marcher correctement.