

---

# TP #2 – PANDEMIC IBERIA

---

## 8INF259 -- Structure de données

### Thématique

Pandemic Iberia est un jeu de plateau coopératif dans lequel les joueurs tentent d'enrayer la propagation de quatre maladies sur la péninsule ibérique au XIXe siècle. Le plateau de jeu est un **graphe** de 48 villes reliées par des routes terrestres et maritimes.

Dans ce TP, vous implémenterez un moteur simplifié qui modélise ce plateau comme un graphe et exploite des algorithmes de parcours pour simuler la propagation des maladies et calculer des plus courts chemins.

### Objectifs pédagogiques

Ce travail pratique permet de :

- Modéliser un problème réel sous forme de graphe (liste d'adjacence)
- Implémenter un parcours en largeur (BFS) dans différents contextes
- Manipuler des conteneurs de la STL (**VECTOR**, **MAP**, **SET**, **QUEUE**)
- Lire et parser un fichier texte structuré
- Concevoir l'architecture interne d'un programme à partir d'une interface imposée

Fichiers fournis

Fichier	Description
<b>MAIN.CPP</b>	Programme principal — <b>ne pas modifier</b>
<b>VILLE.H</b>	Déclaration de la classe <b>VILLE</b> — <b>ne pas modifier</b>
<b>PLATEAU.H</b>	Déclaration de la classe <b>PLATEAU</b> — <b>compléter la section privée</b>
<b>BOARD.TXT</b>	Fichier décrivant le plateau (48 villes, ~90 liens)

À créer : **VILLE.CPP** et **PLATEAU.CPP** (et tout autre fichier que vous jugez nécessaire).

Le code du **MENU()** est fourni en commentaire dans **MAIN.CPP** — intégrez-le dans votre **PLATEAU.CPP**.

### Contraintes

- Les signatures **publiques** des .h ne doivent pas être modifiées
- Vous devez ajouter vos **attributs et méthodes privées** dans Plateau.h
- Vous êtes libres de créer des fichiers supplémentaires

## Règles du jeu simplifié

### Les maladies

Chaque ville a une **couleur** associée à une maladie :

- **BLUE** = Choléra, **RED** = Typhus, **YELLOW** = Fièvre jaune, **BLACK** = Malaria

Chaque ville peut contenir entre 0 et 3 **cubes au total** (toutes couleurs confondues). Lorsqu'on **infecte** une ville, on ajoute 1 cube de la couleur de cette ville.

### Les éclosions (Outbreaks)

Si une ville contient déjà **3 cubes** et qu'on tente d'y ajouter un cube, une **éclosion** se déclenche :

1. Le compteur d'éclosions augmente de 1
2. Chaque ville **voisine** reçoit 1 cube de la couleur de l'éclosion
3. Si une ville voisine est elle-même déjà à 3 cubes, une **éclosion en chaîne** se produit
4. Une ville ne peut éclore qu'**une seule fois** par résolution (éviter les boucles infinies)
5. À **8 éclosions**, la partie est perdue

### Les transports

- **Route terrestre** : se déplacer vers une ville adjacente (1 action)
- **Train** : se déplacer vers n'importe quelle ville reliée par une chaîne continue de rails (1 action)
- **Bateau** : se déplacer d'un port à n'importe quel autre port (1 action)

Les trois modes sont **cumulables** pour trouver le chemin optimal.

## Exercices

### Exercice 1 : Classe **VILLE** et chargement du plateau

Implémentez la classe Ville et la méthode **PLATEAU::CHARGER()**.

Le fichier **BOARD.TXT** contient deux sections : **VILLES** (format **NOM;COULEUR;PORT**) et **LIENS** (format **VILLE1;VILLE2**). Les liens sont **bidirectionnels**. Le fichier contient des commentaires (#) et des lignes vides à ignorer.

Le chargement doit permettre au programme d'afficher un message de confirmation (ex : **48 VILLES CHARGEES**).

### Exercice 2 : Infection et éclosions en chaîne

Implémentez les options 1, 2 et 5 du menu.

L'option 1 place directement 3 cubes de la couleur de la ville (pratique pour préparer un scénario de test). L'option 2 ajoute 1 cube et déclenche une éclosion si nécessaire. L'option 5 affiche l'état courant du plateau.

Vous devez concevoir un **sous-menu de sélection de ville** — l'utilisateur ne doit pas avoir à taper le nom d'une ville manuellement.

Exemple de sortie attendue (3e infection consécutive de Madrid) :

```
Madrid est déjà à 3 cubes total -> ECLOSION !
** ECLOSION #4 à Madrid (Typhus)
|-- Salamanque : 2 -> 3
|-- Valladolid : 2 -> 3
```

```

|-- Saragosse : 2 -> 3
|-- Tolede : 3 cubes total -> eclosion en chaine !
** ECLOSION #5 a Tolede (Typhus)
|-- Caceres : 0 -> 1
|-- Madrid : ignore (deja en eclosion)
|-- Ciudad Real : 3 cubes total -> eclosion en chaine !
** ECLOSION #6 a Ciudad Real (Typhus)
|-- Madrid : ignore (deja en eclosion)
|-- Albacete : 1 -> 2
|-- Tolede : ignore (deja en eclosion)
|-- Badajoz : 1 -> 2
|-- Cordoue : 1 -> 2
|-- Cuenca : 2 -> 3

```

### Exercice 3 : Placement de rails

Implémentez l'option 3 du menu. Le jeu est limité à **20 rails**, et un rail ne peut pas être posé en double. L'utilisateur choisit une ville de départ, puis un voisin.

### Exercice 4 : Plus court chemin

Implémentez l'option 4 du menu. Le programme doit calculer et afficher le plus court chemin entre deux villes selon **3 modes cumulatifs** :

Mode	Déplacements autorisés
[ROUTE]	Route terrestre uniquement
[ROUTE+TRAIN]	Route + Rails (composante rail = 1 action)
[TOUT]	Route + Rails + Mer (port à port = 1 action)

Chaque segment du chemin doit être annoté (**[TRAIN]** ou **[MER]**) et le gain d'actions affiché.

Exemple de sortie attendue (avec rails posés : Badajoz↔Évora, Badajoz↔Ciudad Real, Ciudad Real↔Madrid) :

```

==== Gerone -> Evora ===

[ROUTE] Terrestre uniquement (6 actions) :
Gerone -> Barcelone -> Saragosse -> Madrid -> Ciudad Real -> Badajoz -> Evora

[ROUTE+TRAIN] Terrestre + rails (4 actions) :
Gerone -> Barcelone -> Saragosse -> Madrid -> Evora [TRAIN]
=> Gain de 2 action(s) grace aux rails !

[TOUT] Terrestre + rails + mer (3 actions) :
Gerone -> Barcelone -> Huelva [MER] -> Evora
=> Gain de 3 action(s) au total !

```

## Barème

Exercice	Points
<b>Ex. 1 — Ville + chargement du graphe</b>	25
<b>Ex. 2 — Infection, éclosions en chaîne, affichage</b>	30
<b>Ex. 3 — Placement de rails</b>	10
<b>Ex. 4 — Plus court chemin (3 modes)</b>	30
<b>Qualité du code</b>	5
<b>Total</b>	<b>100</b>

## Format de rendu

Pour être valide, il faut rendre un unique fichier .zip **comportant le dossier racine**. Ce fichier devra être nommé : « TP2\_NOM\_PRENOM.zip » en remplaçant bien entendu les « NOM » et « PRENOM » par vos noms et prénoms respectifs. Les fichiers devront directement être à la racine du fichier .zip (si vous avez des sous-dossiers, n'inclure que le code source, et garder votre structure). Le fichier « board.txt » sera ajouté directement à côté du main.cpp.

## Date de rendu

Le projet devra être remis avec **un seul fichier .zip au plus tard le 13 mars 2026 à 23h59**. Le rendu se fera sur Moodle via l'option créée à cet effet.

## Plateau.h

```
#pragma once
#include "Ville.h"
#include <set>

class Plateau {
    // Attributs et méthodes privées : à vous de décider

public:
    Plateau();

    bool charger(const std::string& fichier);

    void actionPlacerTroisCubes();
    void actionInfecter();
    void actionPlacerRail();
    void actionPlusCourtChemin();
    void afficherEtat();

    void menu();
};
```

## Ville.h

```
#pragma once
#include <string>
#include <vector>
#include <map>

class Ville {
public:
    std::string nom;
    std::string couleur;
    bool port;
    std::map<std::string, int> cubes;
    std::vector<std::string> voisins;

    Ville();
    Ville(const std::string& nom, const std::string& couleur, bool port);

    int totalCubes() const;
    int cubesDe(const std::string& couleur) const;
};
```

## Main.cpp

```
#include "Plateau.h"
#include <iostream>
#include <climits>

#ifndef _WIN32
#include <windows.h>
#endif

int main(int argc, char* argv[]) {
#ifndef _WIN32
```

```

SetConsoleOutputCP(CP_UTF8);
#endif

std::string fichier = "board.txt";
if (argc > 1) fichier = argv[1];

Plateau plateau;
if (!plateau.charger(fichier)) return 1;

plateau.menu();
return 0;
}

// =====
// Code du menu (à intégrer dans votre Plateau.cpp)
// =====
//
// void Plateau::menu() {
//     while (true) {
//         std::cout << "\n";
//         std::cout << "+======" <<
std::endl;
//         std::cout << "|      PANDEMIC IBERIA -- MENU      |" <<
std::endl;
//         std::cout << "+======" <<
std::endl;
//         std::cout << "| 1. Placer 3 cubes sur une ville    |" <<
std::endl;
//         std::cout << "| 2. Infecter une ville (+1 cube)    |" <<
std::endl;
//         std::cout << "| 3. Placer un rail                  |" <<
std::endl;
//         std::cout << "| 4. Plus court chemin                |" <<
std::endl;
//         std::cout << "| 5. Afficher l'état du plateau     |" <<
std::endl;
//         std::cout << "| 0. Quitter                         |" <<
std::endl;
//         std::cout << "+======" <<
std::endl;
//         std::cout << "> Choix : ";
//
//         int choix;
//         if (!(std::cin >> choix)) {
//             std::cin.clear();
//             std::cin.ignore(INT_MAX, '\n');
//             continue;
//         }
//
//         switch (choix) {
//             case 1: actionPlacerTroisCubes(); break;
//             case 2: actionInfecter();           break;
//             case 3: actionPlacerRail();        break;
//             case 4: actionPlusCourtChemin();   break;
//             case 5: afficherEtat();           break;
//             case 0:
//                 std::cout << "Au revoir !" << std::endl;
//         }
//     }
// }

```

```

//           default:
//           std::cout << "Choix invalide." << std::endl;
//       }
// }

```

### board.txt

```

# Pandemic Iberia – Fichier plateau
# 48 villes, 4 couleurs de maladie
#
# Maladies :
#   blue    = Choléra          – nord-ouest / Portugal
#   red     = Typhus           – nord / centre Espagne
#   black   = Malaria          – sud / Andalousie
#   yellow  = Fièvre jaune    – est / Méditerranée
#
# Format VILLES : nom;couleur;port (port = 1 si ville portuaire, 0 sinon)
# Format LIENS : ville1;ville2 (bidirectionnel)

VILLES
La Corogne;blue;1
Saint-Jacques-De-Compostelle;blue;0
Vigo;blue;1
Orense;blue;0
Braga;blue;0
Porto;blue;1
Salamanque;blue;0
Coimbra;blue;0
Évora;blue;0
Albufeira;blue;1
Lisbonne;blue;1
Caceres;blue;0
Tolède;black;0
Badajoz;black;0
Huelva;black;1
Cadix;black;1
Séville;black;0
Cordoue;black;0
Gibraltar;black;1
Malaga;black;1
Jaén;black;0
Ciudad Real;black;0
Grenade;black;0
Almérie;black;1
Saint-Sébastien;red;1
Pampelune;red;0
Huesca;red;0
Bilbao;red;0
Vitoria-Gasteiz;red;0
Soria;red;0
Madrid;red;0
Valladolid;red;0
Burgos;red;0
Santander;red;1
Leon;red;0
Gijon;red;1
Andorre-La-Vieille;yellow;0

```

```
Gérone;yellow;0
Barcelone;yellow;1
Tarragone;yellow;1
Saragosse;yellow;0
Teruel;yellow;0
Cuenca;yellow;0
Albacete;yellow;0
Valence;yellow;1
Alicante;yellow;1
Cartagène;yellow;1
Palma de Majorque;yellow;1

LIENS
# === BLUE ===
La Corogne;Saint-Jacques-De-Compostelle
La Corogne;Gijon
Saint-Jacques-De-Compostelle;Vigo
Saint-Jacques-De-Compostelle;Orense
Vigo;Orense
Vigo;Braga
Vigo;Porto
Orense;Leon
Braga;Porto
Braga;Salamanque
Porto;Coimbra
Porto;Lisbonne
Salamanque;Leon
Salamanque;Valladolid
Salamanque;Caceres
Salamanque;Madrid
Coimbra;Lisbonne
Coimbra;Caceres
Caceres;Tolède
Caceres;Badajoz
Lisbonne;Évora
Lisbonne;Albufeira
Évora;Huelva
Évora;Badajoz
Albufeira;Huelva

# === RED ===
Gijon;Santander
Gijon;Leon
Santander;Bilbao
Santander;Valladolid
Bilbao;Saint-Sébastien
Bilbao;Vitoria-Gasteiz
Saint-Sébastien;Pampelune
Vitoria-Gasteiz;Pampelune
Vitoria-Gasteiz;Burgos
Vitoria-Gasteiz;Saragosse
Pampelune;Huesca
Burgos;Valladolid
Burgos;Soria
Soria;Saragosse
Valladolid;Madrid
Huesca;Saragosse
Huesca;Andorre-La-Vieille
```

```
Madrid;Saragosse
Madrid;Tolède
Madrid;Ciudad Real
Madrid;Cuenca

# === YELLOW ===
Andore-La-Vieille;Gérone
Gérone;Barcelone
Saragosse;Barcelone
Saragosse;Teruel
Barcelone;Tarragone
Barcelone;Palma de Majorque
Tarragone;Valence
Tarragone;Teruel
Teruel;Cuenca
Cuenca;Valence
Cuenca;Albacete
Valence;Palma de Majorque
Valence;Albacete
Valence;Alicante
Alicante;Cartagène
Albacete;Cartagène
Albacete;Ciudad Real
Albacete;Jaén
Cartagène;Almérie

# === BLACK ===
Tolède;Ciudad Real
Badajoz;Ciudad Real
Badajoz;Cordoue
Cordoue;Ciudad Real
Cordoue;Séville
Cordoue;Malaga
Cordoue;Jaén
Jaén;Grenade
Grenade;Malaga
Grenade;Almérie
Almérie;Malaga
Malaga;Gibraltar
Gibraltar;Cadix
Cadix;Séville
Huelva;Séville
Malaga;Séville
```

Une image de la carte (en anglais, désolé) :

