

3A

Coding Week

du 23/01 au 25/01/2023



2022 – 2023

Thomas Raimbault

Mise en œuvre



Pour réaliser les différentes missions proposées durant cette *Coding Week*,

- un certain nombre de **ressources** vous sont **fournies** sur **Moodle**
- et certaines **contraintes** vous sont **imposées**.

Contraintes générales

Contraintes à respecter :

1. travail en équipe : **équipe imposée**
2. langage de programmation : **Java**
3. application en mode "**Console**"
4. une carte sera représentée dans le code par une **matrice d'entiers**
 - les (différents états des) éléments manipulés sont donc des valeurs entières
 - par **convention** : la **1^{ère} dimension** \Leftrightarrow **lignes**, la **2^{nde} dimension** \Leftrightarrow **colonnes**
 - rappel : un index commence à 0 (zéro)

Environnement de développement (pour coder)

- **Replit**

- besoin uniquement d'un navigateur
- [squelette de code](#) Java et autres fichiers déjà disponibles
- permet de coder de façon **collaborative**



```
static void afficherMatriceEnConsole(int[][] matrice) {  
    // parcourir par ligne  
    for(int lig = 0 ; lig < matrice.length ; lig++) {  
        // parcours colonne par colonne  
        for(int col = 0 ; col < matrice[0].length ; col++) {  
            // contenu base  
            System.out.print(matrice[lig][col]);  
            System.out.print(" ");  
        }  
        System.out.println();  
    }  
}
```

Principe général

Principe général

1. **charger** un fichier texte contenant une **carte**
 - cette carte est **convertie** en une **matrice d'entiers**



fichier **texte**



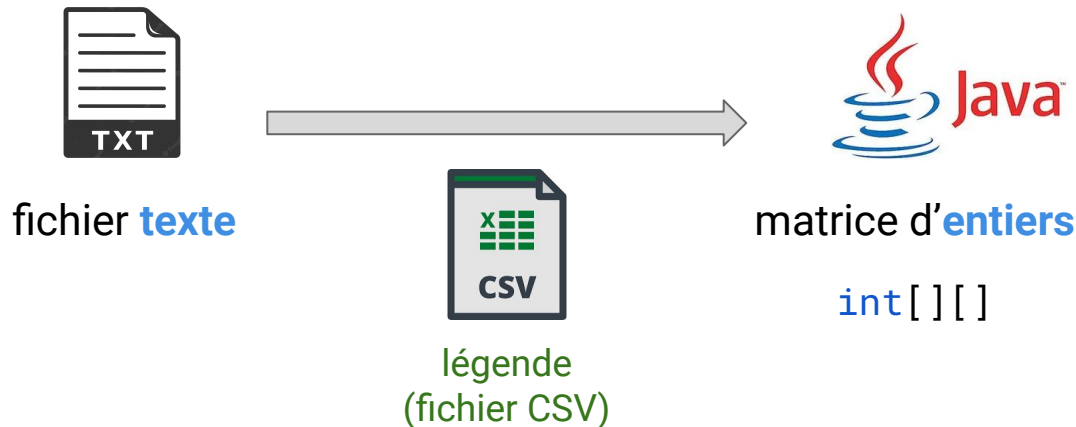
matrice d'**entiers**

`int[][]`

Principe général

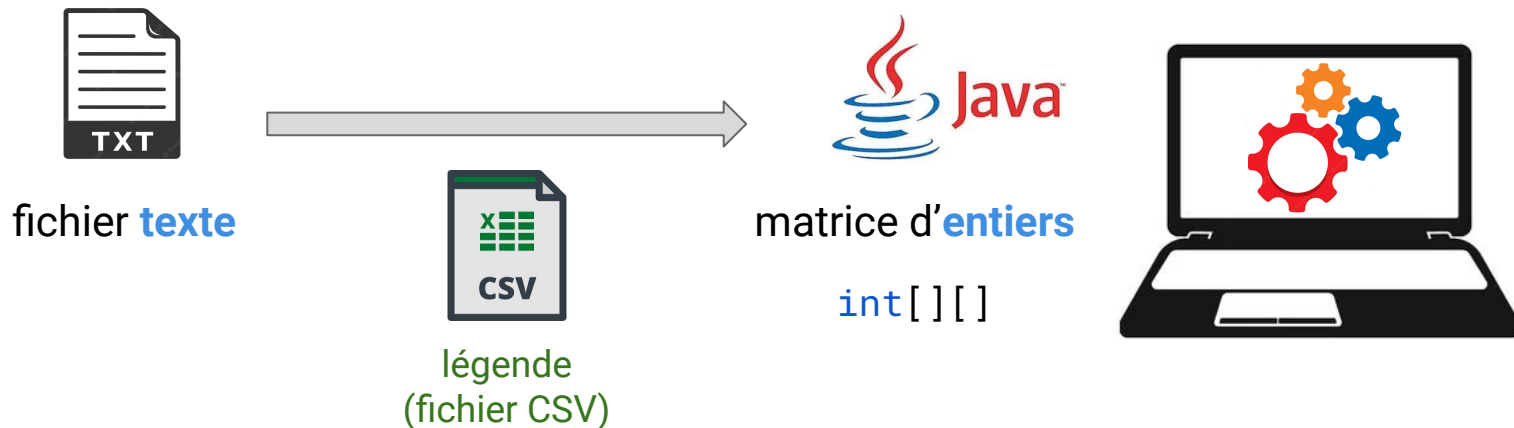
1. **charger** un fichier texte contenant une **carte**

- cette carte est **convertie** en une **matrice d'entiers**
- conversion obtenue selon une **légende** (correspondance lettre ↔ entier)



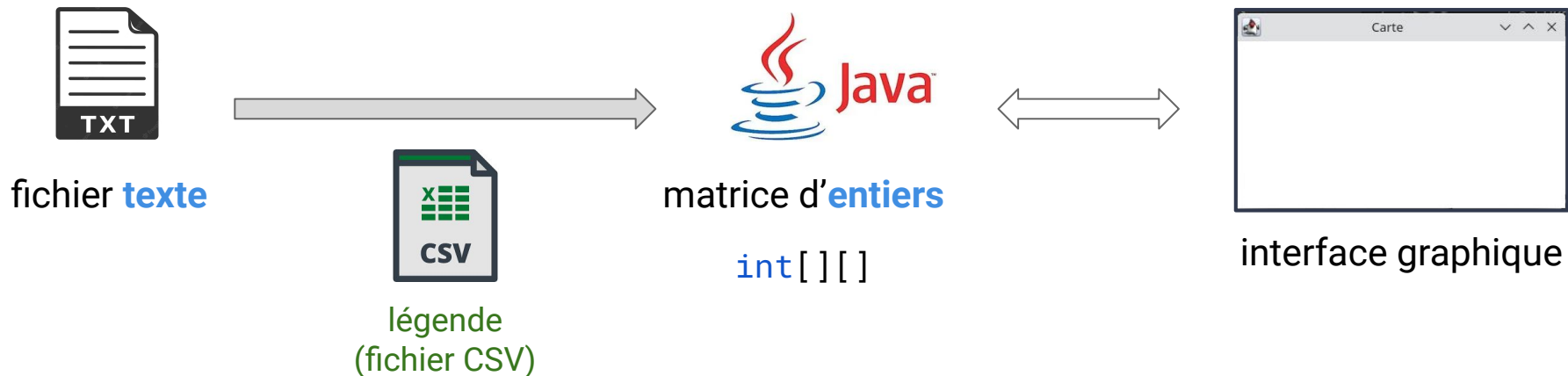
Principe général

1. **charger** un fichier texte contenant une **carte**
 - cette carte est **convertie** en une **matrice d'entiers**
 - conversion obtenue selon une **légende** (correspondance lettre ↔ entier)
2. **manipuler** cette matrice d'entiers via du code Java



Principe général

1. **charger** un fichier texte contenant une **carte**
 - cette carte est **convertie** en une **matrice d'entiers**
 - conversion obtenue selon une **légende** (correspondance lettre ↔ entier)
2. **manipuler** cette matrice d'entiers via du code Java
 - une **interface graphique** est associée à la matrice d'entiers



Principe général



périmètre d'action



manipulation de la matrice
d'entiers via du **code Java**



fichier **texte**

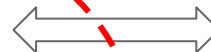


légende
(fichier CSV)



matrice d'**entiers**

`int[][]`



interface graphique

Principe général

Exemple

carte.txt

```
4 5  
rrrrr  
rhhhr  
rhhhr  
rrrrr
```

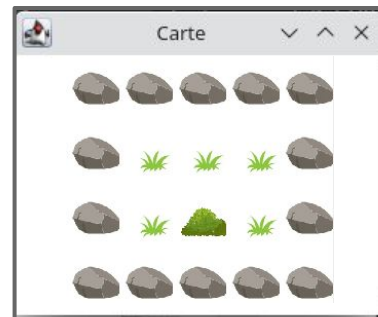
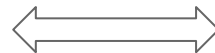
chargerCarte

```
lettre,entier,image  
e,1,images/eau.png  
r,2,images/rocher.png  
h,3,images/herbe.png  
...  
b,7,images/buisson.png  
...
```

legende.csv

int[][]

```
( 22222  
 23332  
 23732  
 22222 )
```



Fichier carte – fichier texte formaté

Une **carte** fournie est un fichier texte (extension .txt) respectant le format suivant :

- le première ligne est composée de
 - un entier = nombre de lignes dans la carte
 - un espace
 - un entier = nombre de colonnes dans la carte
- les lignes suivantes renseignent les éléments de la carte
 - chaque élément est identifié par une **lettre**

```
4 5  
r r r r r  
r h h h r  
r h b h r  
r r r r r
```



Postulat : Afin de simplifier l'écriture du code, les éléments en bordure d'une carte sont nécessairement des éléments non inflammables.

Fichier carte – fichier texte formaté

Une **carte** fournie est un fichier texte (extension .txt) respectant le format suivant :

- le première ligne est composée de
 - un entier = nombre de lignes dans la carte
 - un espace
 - un entier = nombre de colonnes dans la carte
- les lignes suivantes renseignent les éléments de la carte
 - chaque élément est identifié par une **lettre**

```
4 5  
r r r r r  
r h h h r  
r h b h r  
r r r r r
```



Remarque : possibilité de créer ses propres cartes (fichiers .txt spécifiques) pour tester des situations particulières.

Légende – fichier CSV

La **correspondance**, entre les **lettres (imposées)** dans un fichier carte et les **entiers** dans la matrice à manipuler en Java, est renseignée dans un fichier **CSV** ^W (par défaut “legende.csv”).

- **lettre** ← dans le fichier .txt de la carte
- **entier** ← à manipuler dans la matrice d’entiers
- **image** ← pour la fenêtre graphique

```
lettre,entier,image  
e,1,images/eau.png  
r,2,images/rocher.png  
h,3,images/herbe.png  
...  
b,7,images/buisson.png  
...
```

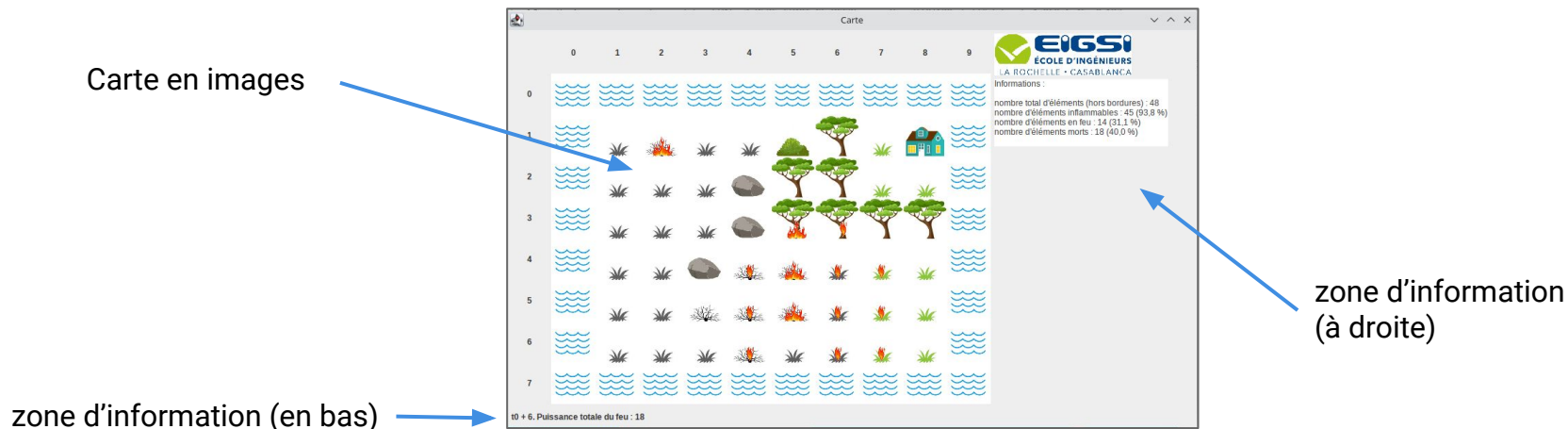
Remarques :

- la valeur des **entiers** sont **libres** de choix !
- en Java, les valeurs pour un **int** sont dans l’intervalle $[-32\,767, 32\,766]$
... ça laisse du choix !!



Fenêtre graphique

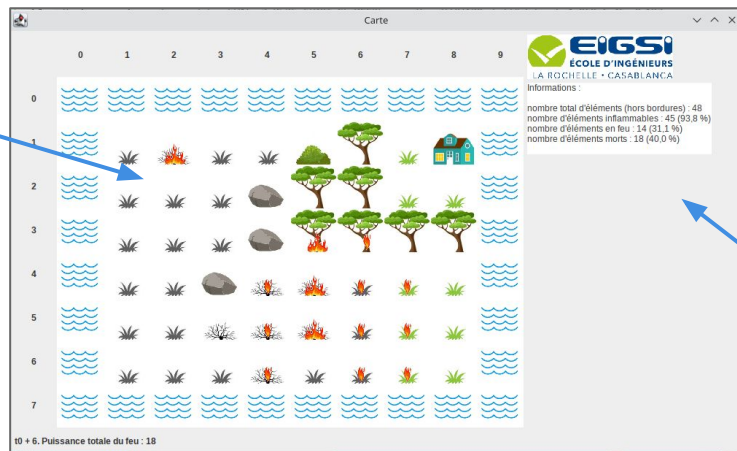
Pour rendre le visuel de votre programme plus exploitable et « moderne » (que simplement l'affichage en Console), une **fenêtre graphique** est associée à une carte (matrice d'entiers).



Fenêtre graphique

Pour rendre le visuel de votre programme plus exploitable et « moderne » (que simplement l'affichage en Console), une **fenêtre graphique** est associée à une carte (matrice d'entiers).

Carte en images



zone d'information
(à droite)

zone d'information (en bas)

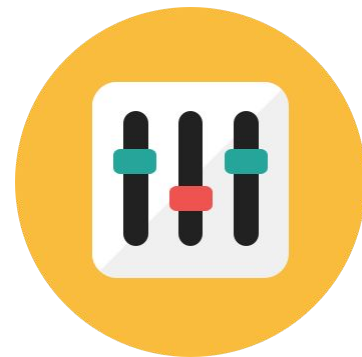


Postulat : pas de création de composant graphique Java dans votre code (on reste sur une application Console) ; seule la fenêtre graphique mise à disposition est à utiliser via les méthodes associées fournies.

Fichier Configuration.java

Pour *customiser* son programme, il est possible de modifier certaines *variables* via le fichier "Configuration.java".

- titre de la fenêtre
- taille des images (des éléments)
- fichier "légende"
- ...



Go !



Have fun!

#CodingWeek #Java #Simulation #Incendie

