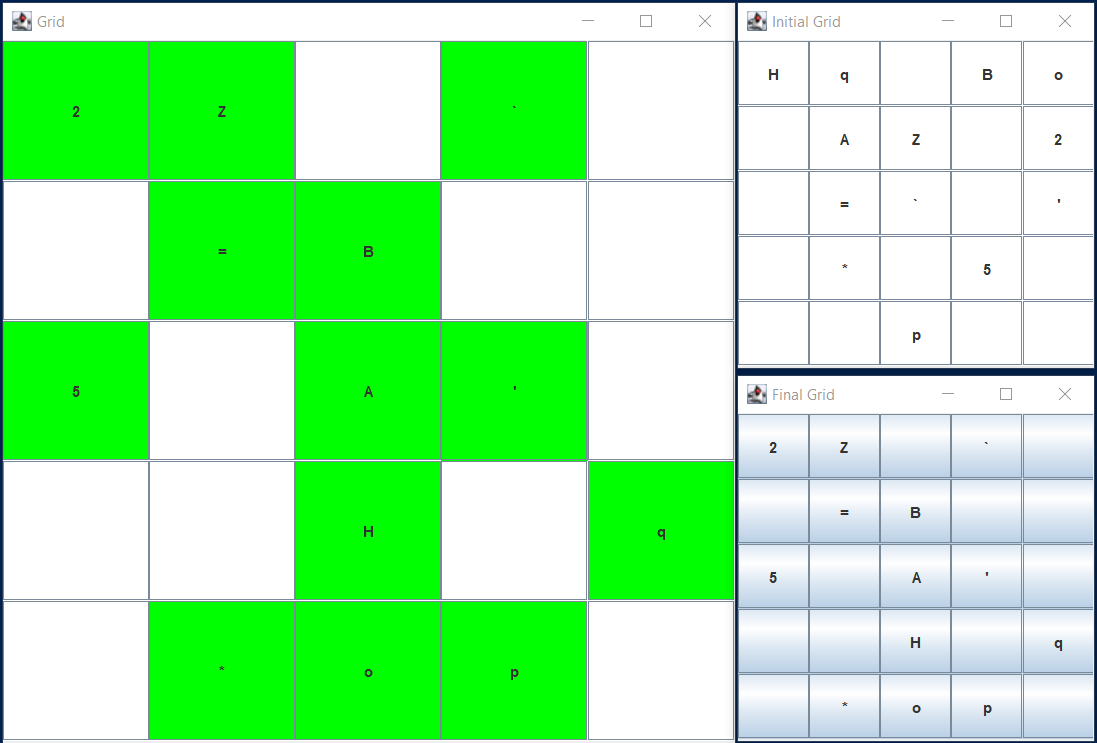
Charles ENG - p1403762

Kévin TANG - p1501263

**TP Techniques IA - Système multi-agents**

****

**Année 2022-2023**

**Enseignant : Samir AKNINE**

1. **Fonctionnement**

Au lancement du jeu, 3 fenêtres apparaissent simultanément à l’écran : la plus grande est la fenêtre principale du jeu, là où tout se déroule, la deuxième fenêtre est une capture de l’état initial du jeu et la dernière fenêtre affiche l’état final du jeu, c’est-à-dire comment doit se terminer la partie.

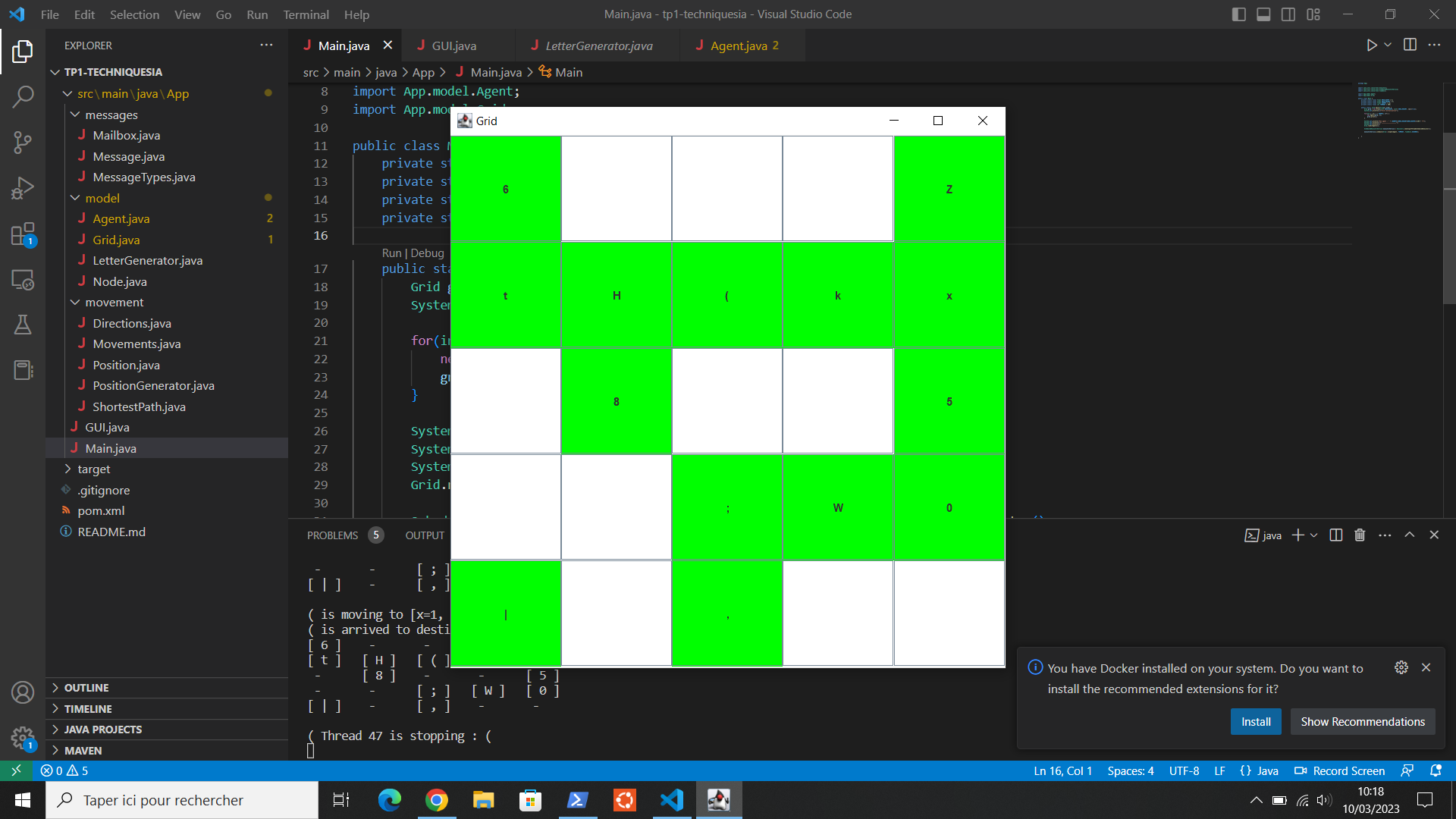
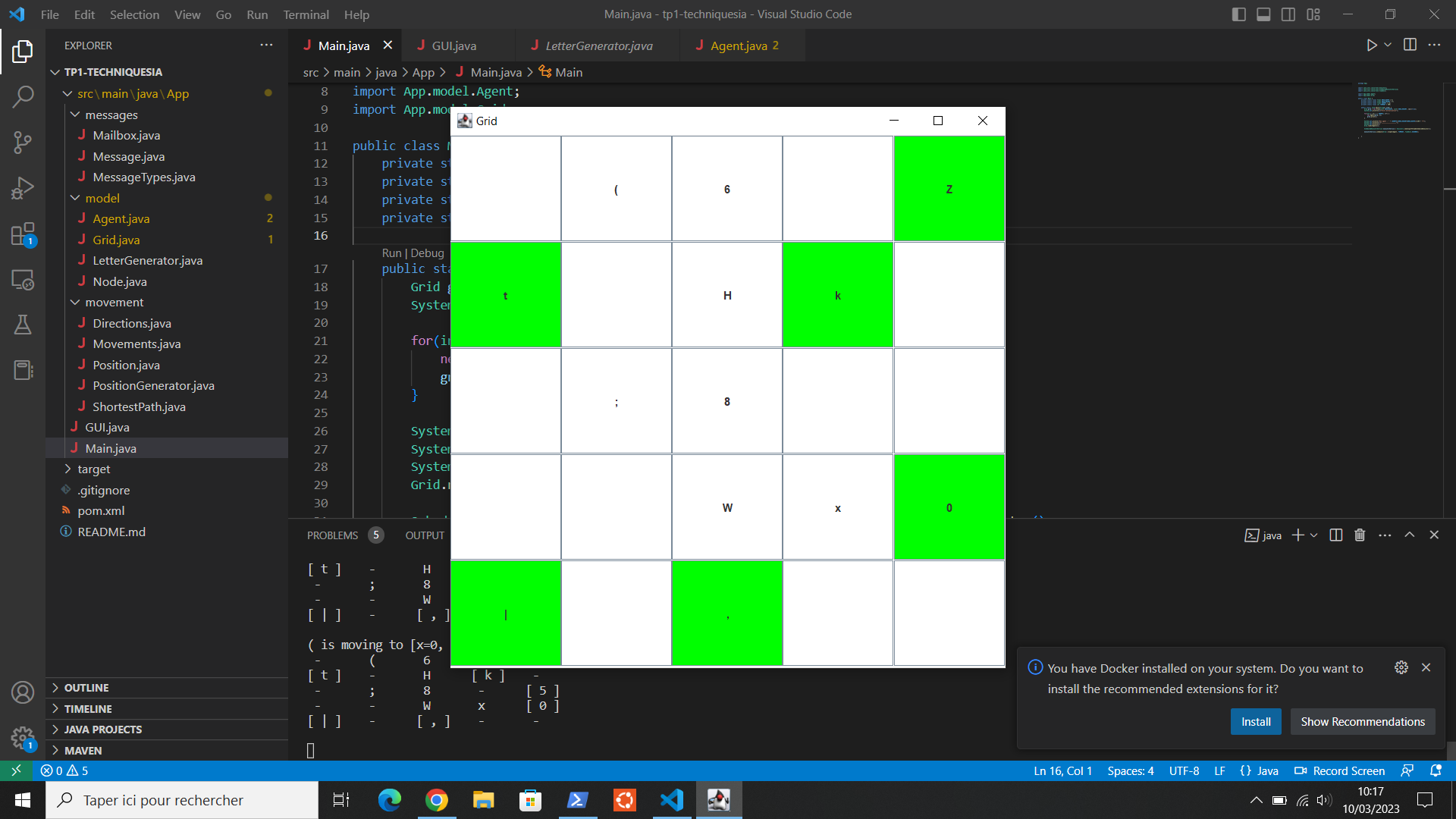
Au début d’une partie, la grille de départ s’initialise avec des paramètres définis : la taille de la grille et le nombre d’éléments que nous appelons **agents**.

Les agents sont générés de manière aléatoire parmi les symboles ASCII en vérifiant qu’il ne peut y avoir deux symboles identiques durant une même partie.

Les agents se trouvent donc dans des cases formant une grille 5x5. La couleur d’une case d’un agent est **blanche** si l’agent n’a pas atteint sa destination finale, lorsque c’est le cas, la case devient **verte**.

Par exemple, ici, les agents “Z”, “t”, “k”, “0”, “l” et “,” ont atteint leur destination finale respective (Image 1).

La partie se termine lorsque tous les agents ont atteint leur destination finale (Image 2)



*Image 1 : Jeu en cours Image 2 : Jeu terminé*

1. **Éléments fonctionnels :**

* Affichage de 3 grilles : la grille où se déroule le jeu, la grille initiale et la grille finale.
* La taille de la grille et le nombre d’agents est indépendant du fonctionnement du jeu, ces paramètres sont donc modifiables sans impact sur le fonctionnement et le jeu se déroulera sans problème aussi bien avec une grille carré (5x5, 3x3…) que rectangulaire (4x3, 5x3, 3x5…).
* Les agents sont générés en choisissant un symbole ASCII de manière aléatoire et deux symboles identiques ne peuvent être générés durant une même partie.
* Lorsque les agents sont choisis, nous générons à la fois une position initiale et une position finale qui est obligatoirement différente de l’initiale.
* Les agents se déplacent bien sur les cases voisines libres et ne sortent pas en dehors de la grille.
* Les agents sont indépendants des autres, ils se déplacent de manière autonome car ils fonctionnent en parallèle, chacun dans un thread différent.
* Les agents communiquent entre eux avec des messages. Plus précisément, un agent envoie un message à un autre si ce dernier lui bloque le chemin, il se déplace alors vers une case adjacente vide pour laisser le chemin libre le temps de laisser passer le premier agent puis il reprend lui-même son propre chemin en recalculant son itinéraire.
* Le programme tourne jusqu'à ce que tous les agents atteignent leur position finale ou jusqu'à ce que le temps prédéfinie maximum soit atteint. Ce temps est lui aussi modifiable.

1. **Éléments non fonctionnels :**

* Le programme tourne en boucle lorsque le nombre d’éléments est trop important par rapport à la taille de la grille (par exemple, à partir de 15 agents sur une grille 5x5).

Pour cela, nous avons tenté d’implémenter une heuristique afin de donner la priorité d’avancement de certains agents par rapport à d’autres grâce à la distance de Manhattan. Plus précisément, nous mesurons la distance qui sépare la position initiale et la destination finale de chaque agent pour donner la priorité d’avancement des threads qui obtenaient la distance la plus courte.

Nous avons eu quelques difficultés à ajouter cette fonctionnalité et surtout à la faire coordonner au fonctionnement des threads.