

TP1 : Jeu du Taquin dans un système Multi-Agents

Florian Geillon, Antoine Billod, Stanislas Bagnol

October 6, 2023

1 Introduction

Le jeu du taquin, également connu sous le nom de puzzle de 15, est un jeu de patience et de réflexion. Le but du jeu est de déplacer des tuiles numérotées sur un plateau en essayant de positionner chacune des tuiles sur son emplacement final. Le plateau est divisé en cases carrées et un nombre de case sont laissées vide pour permettre les déplacements, allant au minimum à une seule case vide.

2 Communication

Pour la communication, nous avons choisi d'utiliser des sockets. Mise en place d'un annuaire afin de gérer la correspondance d'un Id et les ports utilisées. La structure de messages utilisée contient la position à laquelle souhaite aller l'expéditeur du message et qui correspond donc à la position du destinataire du message, la position actuelle de l'expéditeur ainsi que la position où l'expéditeur souhaite aller après être passé sur la case demandée afin que le destinataire ne se déplace pas sur cette case si possible.

3 Stratégies

A chaque tour de boucle, on répète les actions ci dessous tant que la grille n'est toujours pas complète.

3.1 Traitement des messages

On commence par consulter nos messages, qui sont traité les uns après les autres. L'agent commence par vérifier qu'il se trouve toujours à la position demandée dans le message.

Ensuite on regarde si on se trouve dans la configuration 2 cases ont besoin de se diriger dans des directions opposées sur le même axe. Tant que nos positions se sont pas inversées sur cet axe (c'est-à-dire que si par exemple les deux agents devaient se croiser sur l'axe vertical et que l'agent1 était au dessus de l'agent2

alors on boucle tant que l'agent1 n'est pas passé en dessous de l'agent2), j'essaie de me déplacer d'une case de part et d'autre de cet axe en envoyant un message à l'agent situé sur la case si celle-ci n'est pas disponible. Si j'ai pu me déplacer je quitte la fonction.

Ensuite, si le cas précédant n'a pas été rencontré ou qu'aucun déplacement n'a été possible, je tente de me déplacer dans une des 4 directions dont l'ordre est tiré aléatoirement, et on se déplace dans la première direction possible. Si jamais aucun déplacement n'a été possible alors on envoie un message à notre message à notre voisin (hormis celui qui est à la case suivante de l'expéditeur) ayant le moins de voisins.

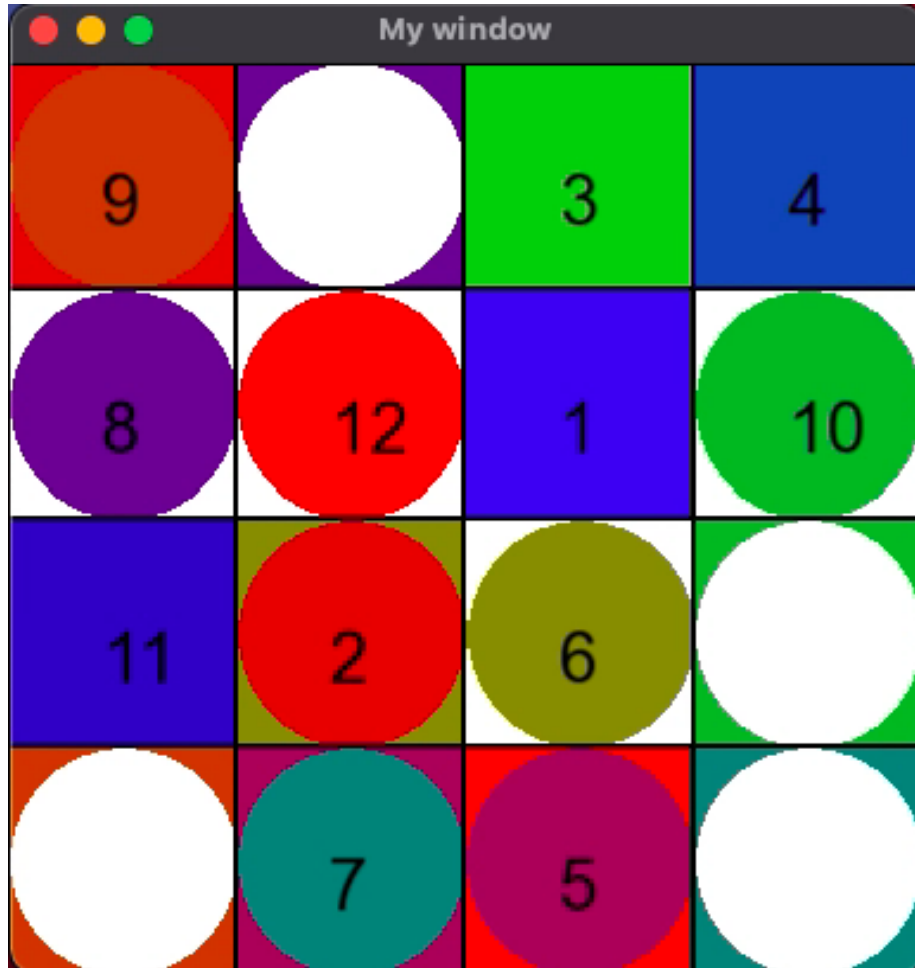
3.2 Calcul de la prochaine case

Pour calculer la prochaine case du plateau qu'une pièce du taquin doit atteindre, on utilise un algorithme A*. Dans cet algorithme on utilise la distance de Manhattan, qui calcule la distance totale entre chaque tuile et sa position cible en comptant le nombre de cases horizontales et verticales à parcourir pour atteindre sa position cible.

3.3 Tentative de déplacement

Si la case où je dois me déplacer est libre, j'y vais et sinon j'envoie un message à l'agent qui est sur la case où je veux aller. J'attends ensuite une secondes et je retente d'aller sur la case si jamais elle s'est libéré suite à mon message.

4 Interface



Pour la visualisation nous avons utilisé la librairie SFML, qui est facilement téléchargeable. Les cercles de couleurs correspondent aux agents et les carrés de couleurs avec un trou blanc au milieu correspondent au case de destination de l'agent ayant la même couleur. Par exemple, dans la capture ci-dessus, l'agent numéro 7 doit se rendre sur la case au coin inférieur droit de la grille.