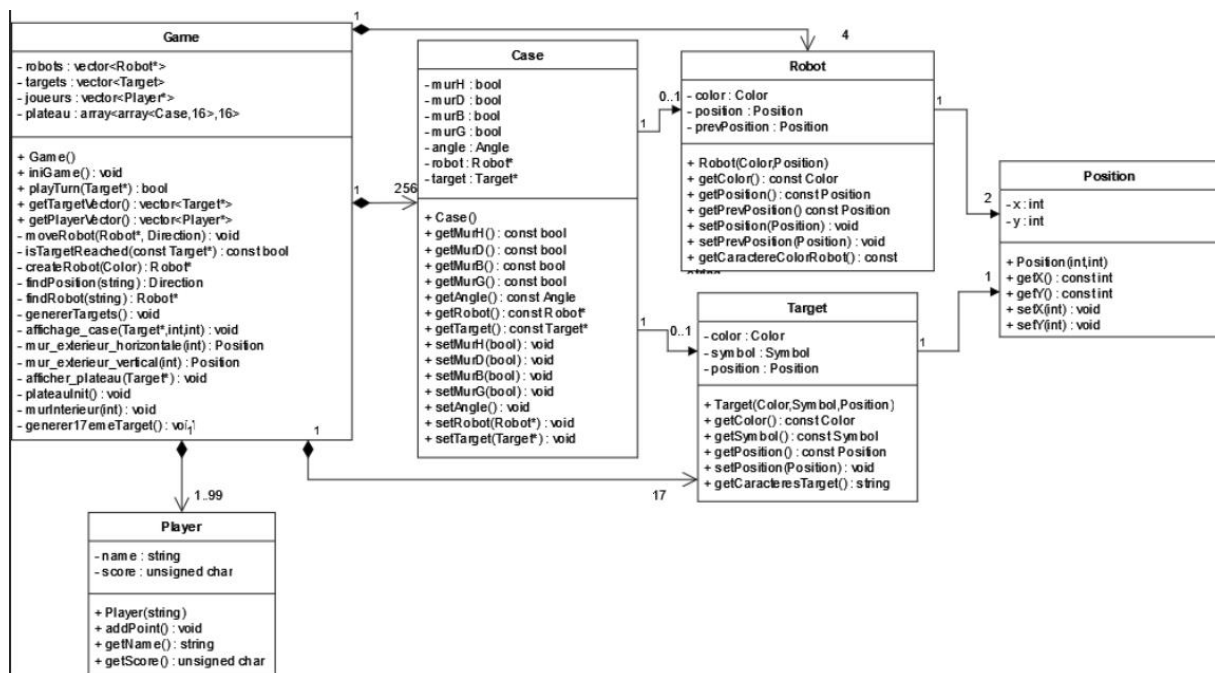


TP – Ricochet Robots

Table des matières

1.	Diagramme de classes	2
2.	Diagramme de cas d'utilisation	4
3.	Description du jeu	5
4.	Illustration du bon fonctionnement	6

1. Diagramme de classes



Pour la conception de notre jeu nous avons implémenté six classes que nous pouvons voir dans le diagramme de classes ci-dessus.

Dans cette conception, la classe "Game" représente le jeu Ricochet Robots lui-même. Elle contient les informations sur le plateau, les robots et les cibles. Les méthodes permettent de jouer, de déplacer un robot, de trouver une solution, d'afficher le plateau et de vérifier une solution proposée. Elle représente également le plateau de jeu.

La classe "Case" représente une case sur le plateau. Elle contient des informations sur les quatre murs de la case ainsi que la cible et le robot si elle en contient un.

Le plateau est un tableau deux dimensions de cases (instances de Case).

La classe Game à accès aux informations sur les cases du plateau, y compris les murs, les robots et les cibles grâce à des accesseurs. Les méthodes permettent d'ajouter des murs, des robots et des cibles, ainsi que d'afficher le plateau. Elle possède un attribut plateau ainsi que des vecteurs d'instances de Robot et de Cible pour représenter les éléments du jeu.

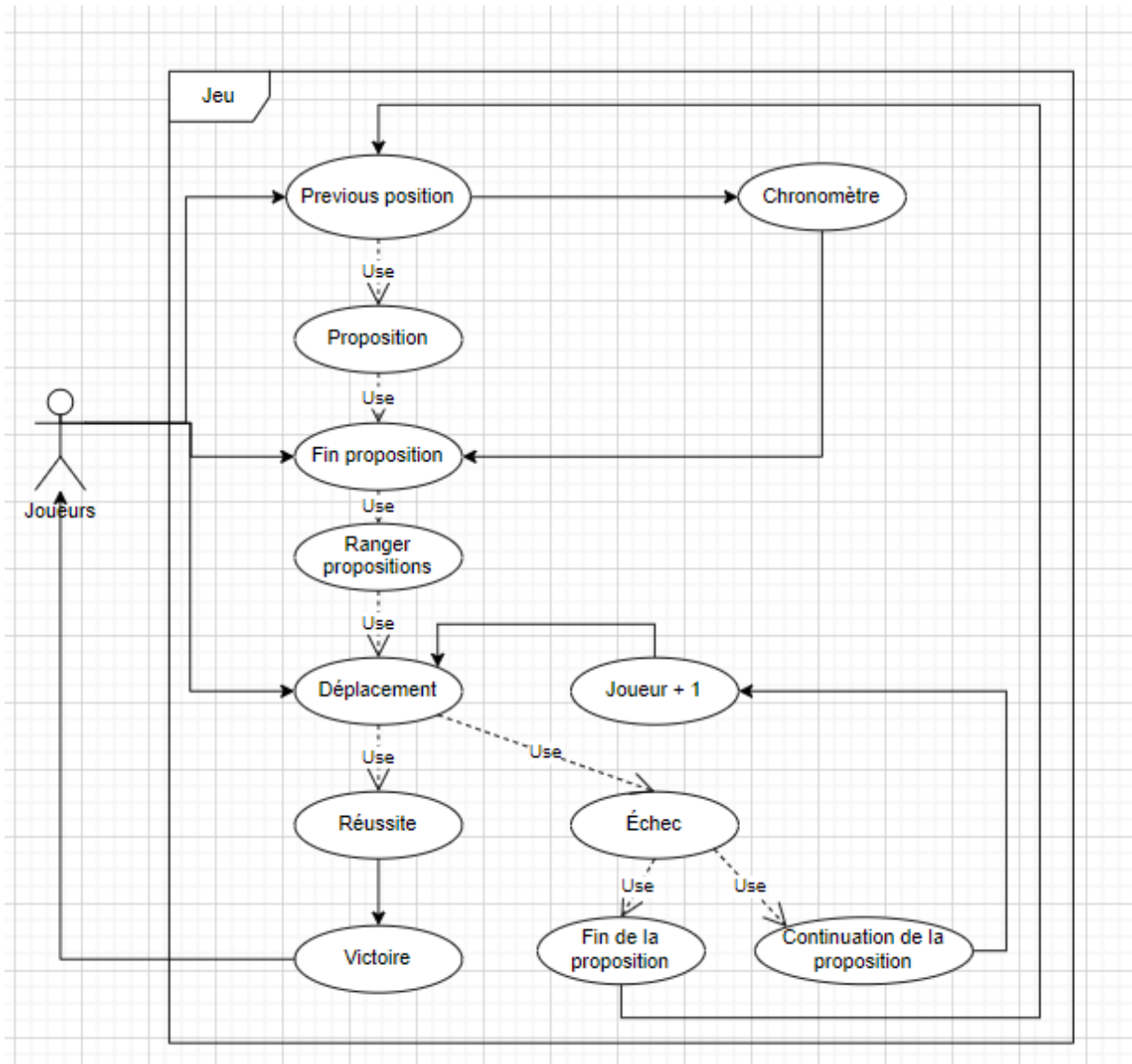
La classe "Robot" représente un robot dans le jeu. Elle contient des informations sur la couleur du robot et sa position sur le plateau.

La classe "Target" représente une cible dans le jeu. Elle contient des informations sur la couleur de la cible, son symbole et sa position sur le plateau.

La classe "Position" représente la position d'un élément. Elle contient des informations sur la position en x et y d'un élément (cible ou robot).

L'implémentation en C++ inclut ces classes, ainsi que les fonctionnalités correspondantes pour jouer au jeu, afficher le plateau, déplacer les robots, trouver des solutions et vérifier les solutions proposées.

2. Diagramme de cas d'utilisation



3. Description du jeu

Tout d'abord, le jeu s'initialise.

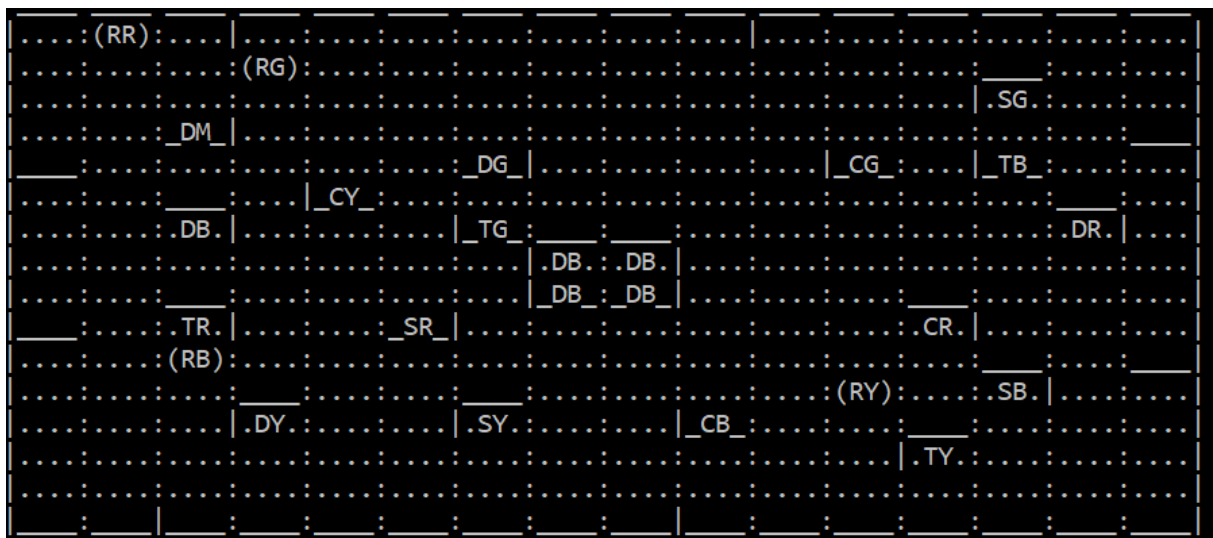
Dans un premier temps nous demandons le nombre de joueur participants.

Par la suite, pour chaque joueur :

- Nous demandons leur nom
- Le nombre de coups qu'ils comptent réaliser

Une fois que tous les joueurs ont fait leur proposition, nous les rangeons dans l'ordre pour que le joueur qui a annoncé faire le moins de coups joue en premier.

A la fin de l'initialisation, nous affichons le plateau avec les murs intérieur et extérieur ainsi que les cibles et les robots générés.



Les robots sont symbolisés par des lettres entre parenthèses : R + première lettre couleur.

Les cibles sont symbolisées par les lettres : première lettre symbole + première lettre couleur.

Les noms des formes et des couleurs sont en anglais (R : rouge, G : vert, Y : jaune, B : bleu, M : multicolore, C : cercle, T : triangle, S : carré, D : diamant)

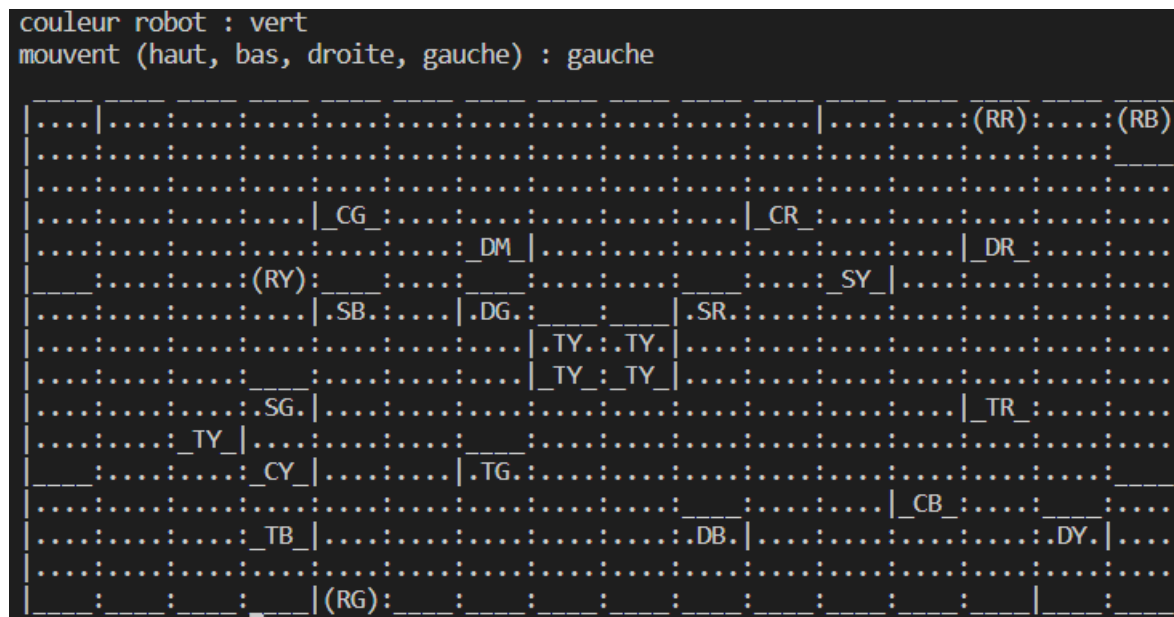
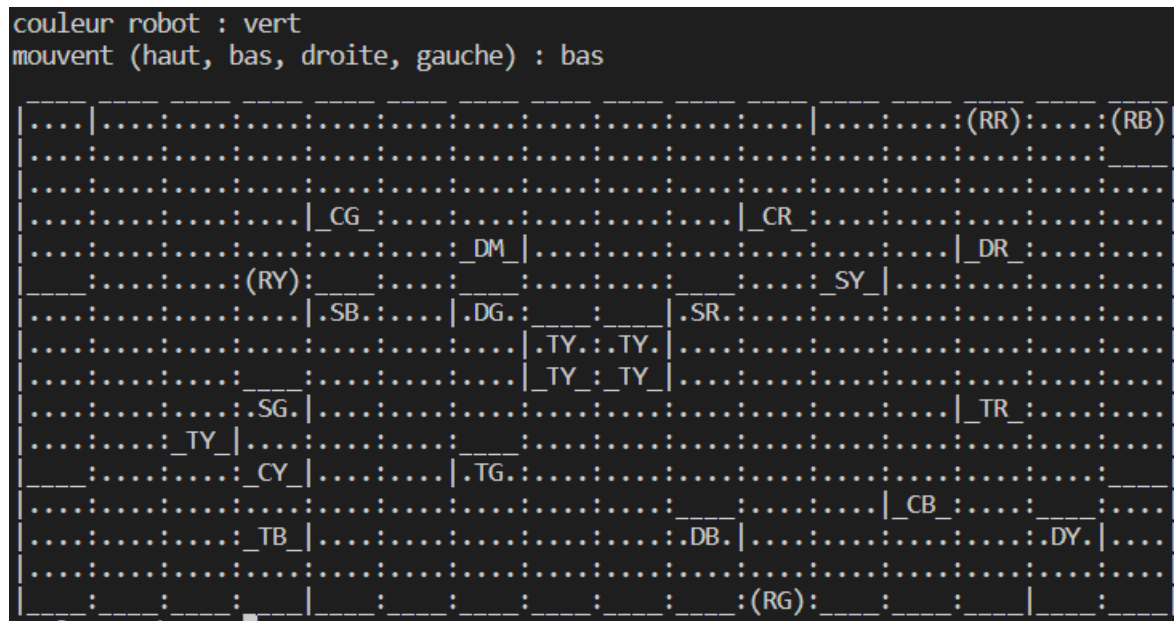
A chaque tour nous affichons le plateau avec le nouveau déplacement des robots.

Lorsqu'un joueur atteint une cible il a une victoire et le jeu se termine. Dans le cas où le joueur n'atteint pas la cible, c'est au joueur suivant de jouer.

Dans la conception de ce jeu, nous n'avons pas mis en place de chronomètre comme il était indiqué dans le sujet car il n'est pas possible de faire tourner un programme en parallèle du programme principal qui fait se dérouler le jeu.

Etant donné que le joueur b a proposé un nombre de coups inférieur au joueur a, c'est le joueur b qui joue.

Sur les images suivantes nous pouvons observer les déplacements du joueur b qui va chercher la cible « TY » avec le robot jaune :



[illegible][illegible]


```
couleur robot : vert
mouvent (haut, bas, droite, gauche) : haut
```

```
|....|(RG):.....:.....:.....:.....:.....|(RR):.....:(RB)|
|.....:.....:.....:.....:.....:.....:.....:.....:.....|
|.....:.....:.....:.....:.....:.....:.....:.....:.....|
|.....:.....:_CG_:.....:.....:.....:.....:_CR_:.....:.....|
|.....:.....:_DM_|.....:.....:.....:.....:_DR_:.....:.....|
|.....:.....:(RY):.....:.....:.....:.....:_SY_|.....:.....|
|.....:.....:_SB_:.....:_DG_:.....:.....:_SR_:.....:.....|
|.....:.....:_TY_:.....:_TY_:.....:.....:.....:.....:.....|
|.....:.....:_TY_:.....:_TY_:.....:.....:.....:.....:.....|
|.....:.....:_SG_|.....:.....:.....:.....:_TR_:.....:.....|
|.....:.....:_TY_|.....:.....:.....:.....:.....:.....:.....|
|.....:.....:_CY_|.....:.....:_TG_:.....:.....:.....:.....|
|.....:.....:.....:.....:.....:.....:.....:.....:_CB_:.....|
|.....:.....:_TB_|.....:.....:.....:....._DB_|.....:....._DY_|
|.....:.....:.....:.....:.....:.....:.....:.....:.....|
|.....:.....:.....:.....:.....:.....:.....:.....:.....|
```

```
couleur robot : vert
mouvent (haut, bas, droite, gauche) : gauche
```

```
|....|(RG):.....:.....:.....:.....:.....|(RR):.....:(RB)|
|.....:.....:.....:.....:.....:.....:.....:.....:.....|
|.....:.....:.....:.....:.....:.....:.....:.....:.....|
|.....:.....:_CG_:.....:.....:.....:.....:_CR_:.....:.....|
|.....:.....:_DM_|.....:.....:.....:.....:_DR_:.....:.....|
|.....:.....:(RY):.....:.....:.....:.....:_SY_|.....:.....|
|.....:.....:_SB_:.....:_DG_:.....:.....:_SR_:.....:.....|
|.....:.....:_TY_:.....:_TY_:.....:.....:.....:.....:.....|
|.....:.....:_TY_:.....:_TY_:.....:.....:.....:.....:.....|
|.....:.....:_SG_|.....:.....:.....:.....:_TR_:.....:.....|
|.....:.....:_TY_|.....:.....:.....:.....:.....:.....:.....|
|.....:.....:_CY_|.....:.....:_TG_:.....:.....:.....:.....|
|.....:.....:.....:.....:.....:.....:.....:.....:_CB_:.....|
|.....:.....:_TB_|.....:.....:.....:....._DB_|.....:....._DY_|
|.....:.....:.....:.....:.....:.....:.....:.....:.....|
|.....:.....:.....:.....:.....:.....:.....:.....:.....|
```

couleur robot : jaune

mouvent (haut, bas, droite, gauche) : haut

[illegible]

```
couleur robot : jaune
```

mouvent (haut, bas, droite, gauche) : gauche

[illegible]

Cependant, lorsqu'un robot atteint une cible la partie s'arrête alors qu'elle devrait passer au tour suivant. Nous n'avons pas résolu ce problème.