

Bases de la programmation orientée objet

– Projet –

Le ‘3 SPOT GAME’ de Edward de Bono

(http://jeuxstrategieer.free.fr/Jeu_des_3_croix_complet.php)

1 Présentation du jeu

Plateau et pièces

Le plateau est constitué de 9 carrés dont 3 sont marqués d’un point comme il est montré dans la figure 1. Ces 3 carrés sont nommés les ‘spots’ du jeu. Chaque joueur (ce jeu se joue à 2) a une pièce colorée (la pièce blanche est neutre) qu’il doit bouger lorsque c’est son tour.

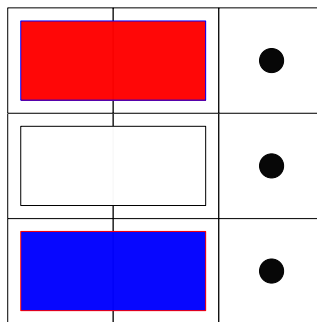


Figure 1: Le plateau, les 3 pièces et les ‘spots’

Objectif

L’objectif est d’atteindre 12 points lorsque l’adversaire en a obtenu au moins 6. En effet, si l’adversaire dispose de moins de 6 points alors celui qui a atteint 12 points est le perdant ! Alors que dans la plupart des jeux, la personne ayant le plus de points est toujours le gagnant, dans celui-ci, le ‘gagnant’ peut être le ‘perdant’.

En conséquence, les joueurs doivent s’adapter entre différentes stratégies:

- Dois-je marquer le plus de points que je peux?
- Dois-je forcer mon adversaire à marquer des points?
- Dois-je éviter de marquer des points?

Ce jeu est très proche de ce que devrait être la justice sociale - allons de l’avant mais ne laissons pas les autres trop en retard par rapport à nous !

Règles

1. La position de départ est indiquée dans la figure 2. Notez que chaque pièce couvre un spot.
2. Chaque joueur choisit la pièce colorée (bleue ou rouge) qu’il va jouer tout le long de la partie. La pièce blanche est neutre.
3. Le jeu commence lorsqu’un joueur déplace sa pièce sur le plateau vers une nouvelle position. Au moins un nouveau carré doit être couvert par la nouvelle position.

4. Après avoir déplacé sa pièce colorée, le joueur doit déplacer la pièce neutre vers une nouvelle position.
5. Après chaque tour, le joueur marque un nombre de points égal au nombre de spots couverts par sa pièce colorée. Les spots couverts par la pièce neutre ne sont pas pris en compte.
6. Le joueur qui atteint 12 points gagne si son adversaire a obtenu 6 points ou plus. Si l'adversaire a strictement moins de 6 points, c'est alors lui qui gagne.

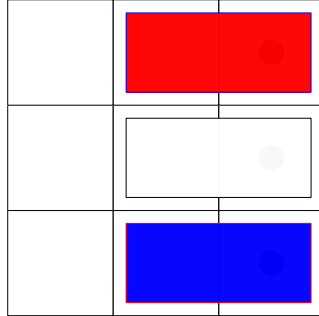


Figure 2: Position de départ du jeu

2 Travail à faire

Vous devez réaliser un programme permettant à 2 joueurs de s'affronter au '3 spot game'. Votre programme devra gérer la partie dans sa totalité.

Avant chaque déplacement, la position courante du plateau devra être affichée et le joueur courant sera prévenu de la pièce qu'il doit déplacer. Votre programme devra aussi comptabiliser les points remportés par les joueurs et détecter automatiquement la fin de partie. Enfin, il affichera la couleur du joueur ayant remporté la partie.

Interface utilisateur

Votre programme affichera le plateau sous la forme suivante:

```

* * * * * * * * * * * * *
*       *       *       *
*       *   R   *   R   *
*       *       *       *
* * * * * * * * * * * * *
*       *       *       *
*       *   W   *   W   *
*       *       *       *
* * * * * * * * * * * * *
*       *       *       *
*       *   B   *   B   *
*       *       *       *
* * * * * * * * * * * * *
* * * * * * * * * * * * *
*       *       *       *
*       *       *       *
*       *   O   *       *
*       *       *       *
* * * * * * * * * * * * *
*       *       *       *
*       *   R   *   W   *   W   *
*       *       *       *
* * * * * * * * * * * * *
*       *       *       *
*       *       *       *
*       *   B   *   B   *
*       *       *       *
* * * * * * * * * * * * *

```

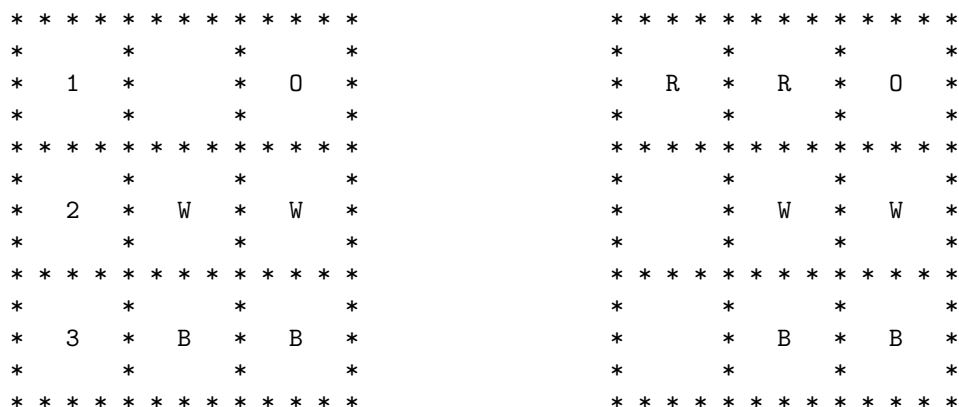
A gauche est donnée la position initiale du jeu et à droite la position après déplacement de la pièce rouge. Notez qu'un caractère 'O' est affiché pour un spot non-recouvert par une pièce. Les autres lettres indiquent la couleur de la pièce occupant le carré ('W' pour 'white', 'R' pour 'red' et 'B' pour 'blue'). Les 3 spots seront toujours les carrés composant la colonne la plus à droite du plateau.

L'objectif est de faire en sorte que l'interface avec les utilisateurs soit le plus simple possible. En particulier, les joueurs doivent pouvoir facilement indiquer les déplacements qu'ils veulent faire.

Lors d'un déplacement, la destination de la pièce peut être complètement spécifiée par une orientation (horizontale ou verticale) et par une coordonnée (celle de la case la plus à gauche si la pièce doit être placée horizontalement ou de la case la plus basse si elle doit l'être verticalement).

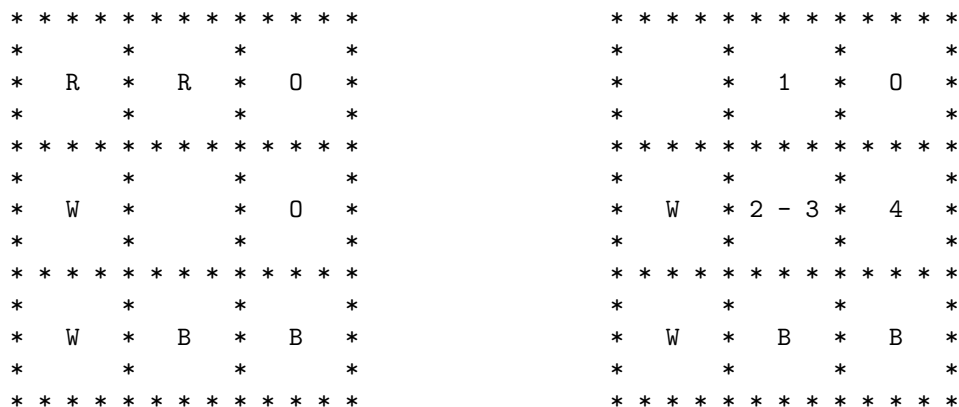
Pour limiter les saisies de la part de l'utilisateur, votre programme numérottera les destinations possibles. Ainsi le choix de l'utilisateur sera restreint à la saisie d'un numéro.

Dans l'exemple suivant, la position de départ est la position initiale du jeu (montrée ci-dessus). Si à partir de cette position, la pièce rouge doit être déplacée, 3 destinations sont possibles (montrées à gauche ci-dessous). Notez que la place initiale de la pièce n'est pas une destination possible et n'a donc pas été numérotée.

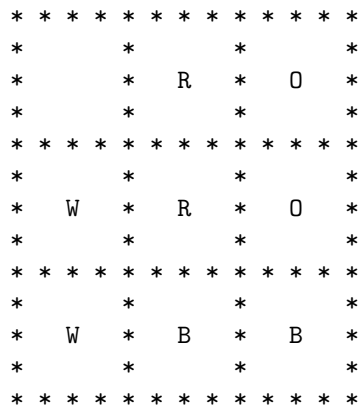


Si le joueur choisit la destination '1' alors cela conduira forcément à la position indiquée sur la droite ci-dessus. En effet, la seule façon de placer la pièce rouge sur cette case est horizontalement. Dans la plupart des cas, il ne sera pas nécessaire de spécifier l'orientation de la pièce car une seule sera possible et c'est donc celle-ci qui sera automatiquement prise. Les destinations '2' et '3' ci-dessus sont similaires, la seule façon d'y placer la pièce rouge est verticalement.

Bien entendu pour certaines positions de jeu, une même case pourra recevoir une pièce placée horizontalement et verticalement. C'est le cas pour l'exemple ci-dessous.



La position avant déplacement est donnée sur la gauche et la pièce devant être déplacée est la rouge. Notez que le carré situé au milieu du plateau peut être la destination de la pièce rouge placée horizontalement ou verticalement. Cette particularité est indiquée par les 2 numéros que contient cette case. Par convention, le premier numéro désigne un placement vertical et le second un placement horizontal. Par conséquence, le choix de la destination '2' conduira à la position suivante.



Qui, quoi et quand?

Votre projet doit être fait en binôme. Les groupes de 3 ne seront pas acceptés. Évitez de faire votre projet tout seul (soit vous êtes très fort et des personnes ont besoin de votre aide, soit vous avez des difficultés et il faut vous faire aider).

Vous devez porter une attention particulière à la rédaction de votre dossier. Sa qualité est déterminante pour l'évaluation de votre travail. Votre dossier doit être un unique document pdf dont la composition est la suivante :

- Une page de garde indiquant le nom et **le groupe** des membres du binôme, l'objet du dossier.
- Une table des matières de l'ensemble du dossier.
- Une brève introduction du projet.
- Le diagramme UML des classes formant vos applications. Seuls les noms des classes et leurs dépendances sont à reporter, il est inutile de préciser leurs attributs et méthodes. Toutefois, vous devez y représenter l'organisation en paquetage.
- Le code Java des tests unitaires de vos classes (en précisant si tous s'exécutent avec succès).
- Le code Java complet de votre projet. L'ordre dans lequel vous présentez vos classes facilite la lecture. Aller des classes élémentaires (celles qui ne dépendent d'aucune autre classe) aux classes plus complexes (en respectant l'ordre de dépendance) est un bon choix.
- Un bilan du projet (les difficultés rencontrées, ce qui est réussi, ce qui peut être amélioré).

Nous vous rappelons que le critère principal de notation est la structuration de votre application. Votre rapport doit mettre en avant cette qualité.

Vous devez rendre votre rapport complet imprimé le **vendredi 9 mars 2018**. De plus, vous déposerez à la même date une archive portant votre nom et contenant l'ensemble de vos fichiers sources (et uniquement cela) dans le puits BPO. Seules les archives au format **jar** ou **zip** seront acceptées.