

Projet informatique : **Joueur de Otrio**

Règles du jeu

Otrio est un jeu de société largement inspiré du célèbre morpion. Cependant, il est plus riche en stratégie. Il se joue de 2 à 4 joueurs. Chaque joueur dispose de 9 cercles de couleur identique et de 3 tailles différentes (petit, moyen et grand) qu'il doit placer sur un plateau de 9 cases, dont chacune comporte 3 emplacements (petit, moyen et grand) (cf. Figure 1).

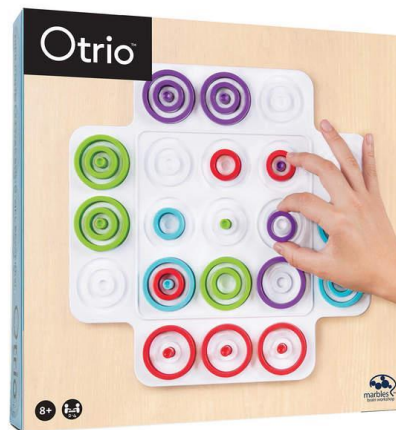


Figure 1 : Le jeu Otrio (source
[https://www.king-jouet.com/jeu-jouet/jeux-societes/jeux-strategie/ref-72046
O-otrio.htm#lasuitedescri](https://www.king-jouet.com/jeu-jouet/jeux-societes/jeux-strategie/ref-72046/O-otrio.htm#lasuitedescri))

Pour gagner, il y a trois possibilités :

- Aligner 3 cercles de même couleur et de même taille horizontalement, verticalement ou diagonalement ;
- Aligner 3 cercles de même couleur et de tailles différentes horizontalement, verticalement ou diagonalement de manière ascendante (du plus petit au plus grand) ou descendante (du plus grand au plus petit) ;
- Mettre 3 cercles de même couleur et de tailles différentes dans la même case du plateau de jeu.

Remarque : Voici un lien vers une vidéo présentant le jeu et les situations de gain :

https://www.youtube.com/watch?v=r_AckXnIBJ4

Travail demandé :

Le projet consiste à programmer ce jeu en Java. Le programme devra permettre de choisir le nombre de joueurs (2 à 4) et le nombre de couleurs par joueur (1 ou 2 si 2 joueurs). Le programme demande le prénom des joueurs et indique qui doit jouer au fur et à mesure du déroulement de la partie. **Dans un premier temps, une version 2 joueurs, 2 couleurs est demandée.** Chaque joueur, à tour de rôle, indique la case dans laquelle il veut jouer et la taille du cercle à placer (petit, moyen ou grand). A chaque coup, le programme devra vérifier la disponibilité de l'emplacement joué et la disponibilité du cercle joué. A chaque coup le programme devra vérifier s'il y a un vainqueur.

Il faudra prévoir la gestion de fin de partie avec la possibilité d'une partie nulle. En cas de victoire de l'un des joueurs, on indiquera qui a gagné. Enfin on offrira la possibilité aux joueurs de recommencer une nouvelle partie.

A vous de choisir la manière d'afficher le plateau de jeu ainsi que les caractères permettant d'afficher les cercles sur le plateau de jeu (il n'est pas demandé de coder une interface graphique). Votre code devra être structuré en sous-programmes.

Extension 1: Humain Vs. IA (noob)

Dans la version deux joueurs, prévoir une version « ordinateur contre humain » où l'ordinateur choisit au hasard un emplacement vide. Vous pouvez commencer par choisir aléatoirement les emplacements pour placer les pièces. Cette solution pourra être améliorée en dotant l'ordinateur d'une intelligence primaire lui permettant de contrer les coups de l'utilisateur.

Extension 2 : Humain Vs. IA (expert)

L'objectif de cette extension est de rendre l'ordinateur « plus intelligent » en élaborant une ou plusieurs stratégies simples plutôt que de parer les coups du joueur. Plusieurs possibilités existent : capitaliser sur des expériences de parties passées ; élaborer des règles de placement de pièces par rapport à des configurations du plateau, etc.

Évaluation :

Le projet est à réaliser **en trinôme (en binôme si pas de possibilité de trinôme)** dans le même groupe de TP. Le projet sera rendu sur moodle avant le 07 janvier 2023 (avant minuit) sous forme de code (les fichiers java utilisés), d'un rapport (format pdf) de 5 pages maximum qui détaillera la méthodologie adoptée et les fonctionnalités développées. La soutenance (max 10 minutes) se déroulera à la séance du TP8 sous forme d'une présentation ppt. Une démonstration du jeu est à prévoir le jour de la soutenance. Un point sur l'avancement du projet sera organisé lors du TD6. L'auto-évaluation finale est à insérer à la fin du rapport (après les 5 pages demandées).

Parmi différents critères seront pris en compte lors de l'évaluation :

- La clarté et la qualité du rapport et du code;
- La qualité de la présentation le jour de la soutenance;
- Le partage des tâches entre trinôme;
- Le niveau d'optimisation du code et du temps de calcul.