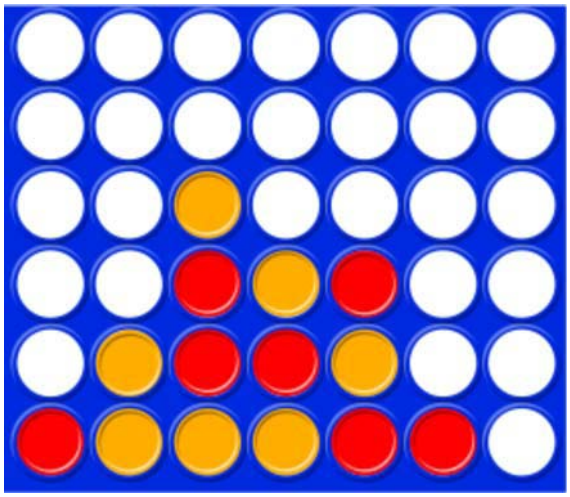
Projet

Puissance 4 (appelé aussi parfois **4 en ligne**) est un jeu de stratégie combinatoire abstrait, commercialisé pour la première fois en 1974 par la Milton Bradley Company, plus connue sous le nom de MB.



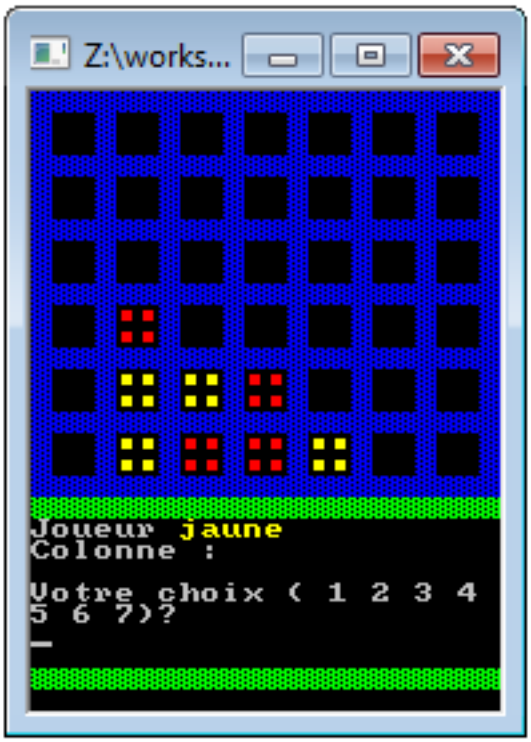
Le but du jeu est d'aligner 4 pions sur une grille comptant 6 rangées et 7 colonnes. Chaque joueur dispose de 21 pions d'une couleur (par convention, en général jaune ou rouge). Tour à tour les deux joueurs placent un pion dans la colonne de leur choix, le pion coulisse alors jusqu'à la position la plus basse possible dans ladite colonne suite à quoi c'est à l'adversaire de jouer. Le vainqueur est le joueur qui réalise le premier un alignement (horizontal, vertical ou diagonal) d'au moins quatre pions de sa couleur. Si alors que toutes les cases de la grille de jeu sont remplies aucun des deux joueurs n'a réalisé un tel alignement, la partie est déclarée nulle.

Le but de ce projet est de réaliser une application qui permet de jouer au Puissance 4. Dans cette application, il devra être possible de faire s'affronter :

- soit deux joueurs humains,
- soit 1 joueur humain et 1 joueur simulé par l'ordinateur,
- soit 2 joueurs simulés par l'ordinateur.

Partie 1. **(il est conseillé de terminer cette partie avant le 10 octobre 2013)**

Dans un premier temps, il vous est proposé de créer **une ou des classes** qui permet de gérer le plateau de jeu et son affichage. Pour l'instant, il ne s'agit pas de gérer une partie en elle-même, mais de préparer tout ce qu'il sera nécessaire pour la gestion du plateau de jeu et l'affichage des différents éléments.
La ou les classes créées doivent donc proposer une interface qui permet, entre autres, de gérer le plateau de jeu et son affichage, de savoir dans quelle colonne il est possible d'ajouter un pion (colonne non pleine), de pouvoir ajouter un pion jaune ou rouge dans une colonne donnée, de déterminer si la partie est terminée (soit parce que 4 pions de la même couleur sont alignés, soit parce que la grille est pleine) et qui a gagné ou si c'est match nul, etc.



Afin de créer un affichage agréable sur la console, vous pouvez utiliser (mais ce n'est pas obligatoire) les sources de la classe `console` qui vous est fournie dans les fichiers "*`console.h`*" et "*`console.cpp`*". Cette classe vous permet de préciser les dimensions de la console (en nombre de caractères), d'afficher un caractère à des coordonnées données, de changer la couleur des caractères, d'effacer tous les caractères de la console, etc. Notons que les caractères de numéro 178 et 254 (en code ASCII) représente de bons candidats pour afficher respectivement les bordures de la grille de jeu et des morceaux des pions. Voici ci-contre à droite un exemple d'affichage (ce n'est qu'une suggestion, toutes les présentations lisibles sont les bienvenues) qu'il est facile d'obtenir.
Attention, cette classe ne fonctionne que sous OS Microsoft Windows. Si vous utilisez un autre OS, vous pouvez chercher des classes qui permettent de faire le même genre de chose.

De même, afin d'interagir de façon conviviale avec un utilisateur, vous pouvez aussi, par exemple, utiliser la fonction `getch()` déclarée dans le fichier d'entête `<conio.h>` qui permet de saisir un caractère sur le clavier sans qu'il soit nécessaire d'utiliser le retour chariot (return). Attention, cette fonction ne fonctionne également que sous OS Microsoft Windows.

Consignes :

- Les classes développées doivent obligatoirement utiliser des tableaux alloués dynamiquement pour gérer l'état du plateau de jeu.
- Définir avec soin l'interface de ces classes de façon à rendre leur utilisation facile.
- Penser que la console sera aussi utilisée pour les dialogues avec l'utilisateur.
- Les classes console et la fonction `getch()` ne sont que des suggestions : vous êtes libres d'utiliser d'autres bibliothèques de votre choix.
- L'exemple de l'affichage proposé ci-dessus n'est aussi qu'une suggestion : vous êtes libres de faire des présentations plus jolies ou plus agréables à utiliser.

Partie 2. **(Cette partie utilise des notions (liées à l'héritage) qui ne seront vues qu'à partir de la semaine du 7 octobre 2013. Vous pouvez néanmoins commencer à la développer.)**

Dans cette deuxième partie, il s'agit de gérer les parties de Puissance 4. Il s'agit tout d'abord de mettre en place un dialogue avec l'utilisateur qui permet de paramétrer une partie : choix du joueur rouge et du joueur jaune, choix du joueur commençant la partie, etc. Pour chacun des joueurs, il devra être possible de préciser si le joueur sera humain ou simulé par le programme. Il s'agit ensuite de gérer la partie elle-même : donner la main successivement aux différents joueurs pour qu'ils annoncent leur coup, gestion des erreurs de saisie (par exemple, la tentative d'ajout dans une colonne pleine), de déterminer si un joueur a gagné (et l'annoncer) ou si la partie est nulle (toutes les colonnes sont pleines et aucun joueur n'a aligné 4 pions), etc.

Consignes :

- Utiliser judicieusement l'héritage en déterminant quelle devrait être l'interface d'une classe joueur et en spécialisant le comportement selon qu'il s'agit d'un joueur humain ou d'un joueur simulé par le programme.
- L'intelligence donnée au (ou aux) joueur(s) simulés par le programme ne sera pas pris en compte dans la notation du projet : vous pouvez simplement créer un joueur qui joue au hasard dans les colonnes où cela est possible. Néanmoins, vous pouvez vous amuser (si vous en avez le temps et l'envie) à créer des programmes-joueurs plus intelligents.

Consignes générales

- Le projet s'effectue par groupe de 3 étudiants (ou 2, **mais pas seul**).
- Les trinômes sont constitués lors de la première semaine dans chaque groupe de TD.
- Le projet est à rendre sous forme d'archive « .zip » contenant :
  - les sources de votre programme (seulement les fichiers .h et .cpp, **pas d'exécutable**) ainsi que les fichiers ou un lien vers les bibliothèques non standards éventuelles que vous utilisez (console, biblio graphique sur autre système).
  - 1 rapport (5 à 10 pages en pdf) : brève présentation du projet, architecture objet du projet en UML, description et explication de chaque classe et de son interface ainsi que des interactions avec les autres objets.
- Le projet est à envoyer par mail avant le 31 octobre 2013 à laetitia.pfaender@gmail.com (pour le groupe 1) ou à claude.renaud@utbm.fr (pour le groupe 2).
- Le fichier rendu s'appellera xxxx_yyyy_zzzz.zip où xxxx, yyyy et zzzz sont les numéros d'étudiant du binôme/trinôme.