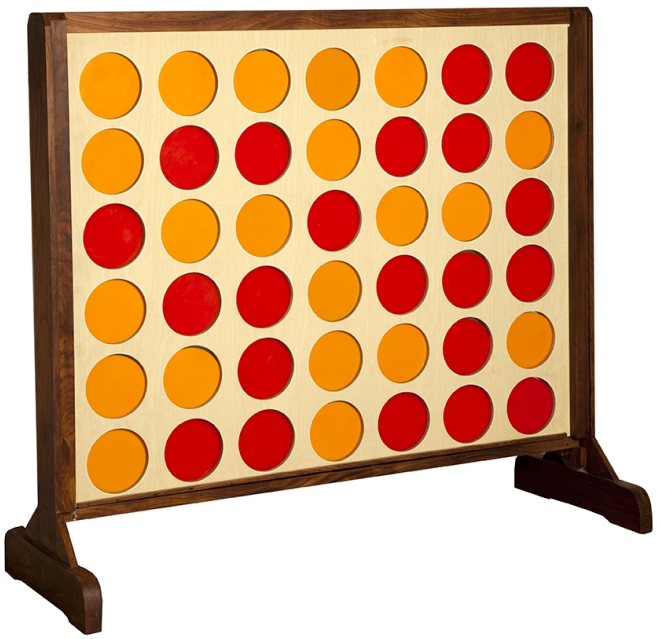
Rapport du développement du puissance4

Par GUARDIA Quentin en L3 d'informatique à l'UPVD
quentin.guardia@etudiant.univ-perp.fr

Table des matières



Source de l'image : okievents.com

Structure

Le fichier « main.c » est le pivot central de l'exécutable « puissance4 ». Il offre trois choix :

- Démarrer une partie. Dans ce cas, la fonction « initialiser » est appelée. Puis la fonction « jouer » s'appelle récursivement tant que la partie n'est pas terminée. Le gagnant ou l'égalité est annoncée grâce à la fonction « fin » puis il est proposé de rejouer. La fonction « jouer » appelle la fonction « verifier » qui, comme son nom l'indique, vérifie si un coup est valide et les fonctions permettant de détecter les alignements diagonaux et droits des pions du joueur actuel.

- Lancer les tests unitaires, en liaison avec le dossier « data_tst ». Ceux-ci permettent de tester la validité des fonctions principales et seront détaillés dans la partie suivante.

- Simuler des parties avec des coups prédéfinis. On contrôle ensuite que les issues soient bien celles attendues. Les simulations sont explicitées un peu plus bas.

Ces deux dernières fonctionnalités utilisent les mêmes fonctions « charger », qui permet de charger une grille depuis un fichier texte et « comparer » qui compare deux grilles.

Tous ces fichiers sont dans le dossier « src », et lors de la compilation leurs fichiers objet associés se situeront dans le dossier « obj ».

Test unitaires

Les tests unitaires assurent le bon déroulement des fonctions « verifier », « initialiser » ainsi que celles testant tous les alignements possibles.

Pour cela, des grilles sont prédéfinies de manière à pouvoir tester ces fonctions. Elles sont chargées à partir des fichiers dans le dossier « data_simul » grâce à la possibilité de lecture de fichiers en C. Ainsi, chaque grille met à l'épreuve les fonctions et pour chaque fonction, si tous les résultats attendus sont correct, il est spécifié en sortie que le test est validé. En cas inverse l'échec est affiché.

On teste les fonctions retournant un booléen en rentrant plusieurs paramètres possibles et en s'assurant que le booléen retourné est toujours correct. Pour la fonction « initialiser », il suffit de comparer la grille affectée avec une grille chargée grâce à une fonction prévue à cet effet.

Toutes les fonctions testées sont appelées par la fonction « jouer » et en ce sens, vérifier leur bon fonctionnement nous permet de tamiser le code à regarder en cas de dysfonctionnement voire d'erreur.

Simulations

Deux fichiers texte sont associés à chaque partie simulée. L'un contient la suite de numéros de colonne à rentrer alternativement et automatiquement par chaque joueur. L'autre contient le résultat sensé être obtenu théoriquement. Dès qu'une simulation est lancée, le fichier contenant les coups à jouer est lu grâce à la fonction standard « fscanf » et la simulation s'arrête lorsqu'un joueur à gagné, lorsqu'il n'est plus possible de jouer ou bien dès qu'il n'y a plus rien à lire en entrée. Puis le tableau final est comparé au tableau théorique. Trois simulations différentes sont mises à disposition, ce qui donne l'avantage de tester des entrées différentes.

Commandes

L'unique commande à exécuter une fois le make est terminé, est « ./puissance4 ». Une fois cette commande lancée, un menu propose de commencer une partie, lancer les tests unitaires ou bien les simulations. Tout est donc compris dans le même exécutable.

La commande « make clean » permet de supprimer les fichiers objet ainsi que l'exécutable.