IMAC
INGÉNIEUR
ESIPE

Programmation et Algorithmique I

Projet

Reversi

Au terme de vos cours et TP d'Algorithmique et Programmation, vous aurez acquis les bases de la programmation C++. Il est désormais temps de mettre en application dans un même programme les différentes notions que vous avez vues. Ce programme permettra à deux utilisateurs de jouer au reversi sur un terminal de commandes.

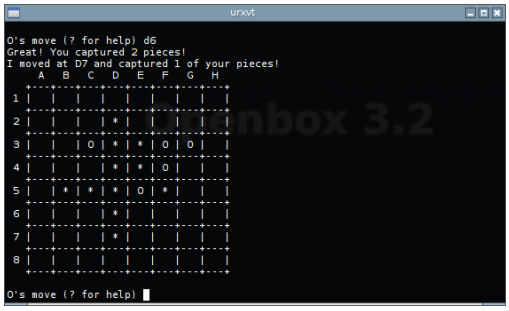


Figure 1 – Boardgames - Reversi

Figure 2 – Inconsolation - minimalist software for Linux

Vous allez donc devoir réaliser un programme en C/C++, le but étant :

- d'appliquer les concepts de base de programmation (structures conditionnelles, boucles, opérations arithmétiques et booléennes)
- de vous familiariser avec le concept de structure
- de vous familiariser avec les concepts d'entrées/sorties (entrée utilisateur/affichage et traitement de fichier)
- d'organiser votre code afin de travailler aisément sur un programme complexe

Évaluateur :
Steeve Vincent : steeve.v91@gmail.com

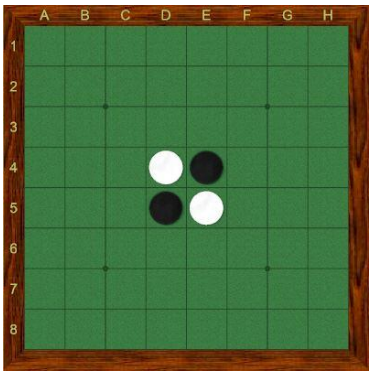
Modalités de rendu

- **Nombre de personnes par projet :** 2
- **Livrable :**
  — Les sources C/C++ ainsi qu'un système de compilation permettant de compiler sur "les machines de l'université" (Linux avec les packages SDL (1 et/ou 2), SDL_image ainsi que make, g++ et cmake)
  — Un rapport au format PDF (maximum 3 pages) décrivant le projet (en particulier les aspects qui diffèrent du sujet initial, inutile de répéter le sujet), votre méthode de travail, des détails techniques si vous souhaitez détailler une fonctionnalité, les difficultés rencontrées et enfin les améliorations possibles.
  — Une vidéo ou un gif présentant succinctement la compilation et l'exécution du programme. Vous n'avez pas besoin de terminer une partie, testez seulement les fonctionnalités principales. Vous pouvez utiliser Screen2Gif pour l'enregistrement.
- **Date de rendu (non définitive) :** Janvier 2021
- **Date de soutenance :** Janvier 2021

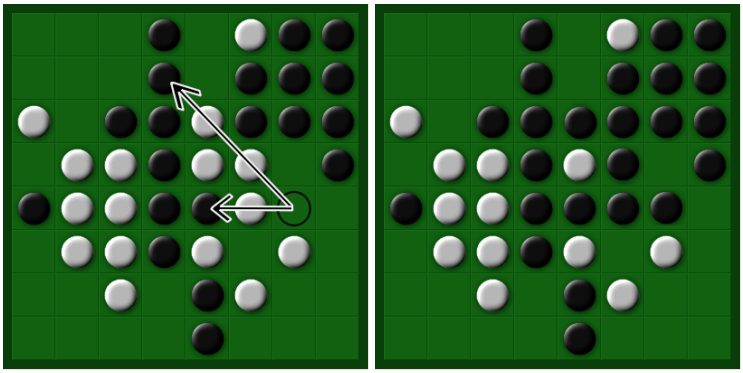
1 Jeu

1.1 Règles

Le reversi oppose deux joueurs qui jouent sur **un plateau de 64 cases (8x8)**. Chaque joueur possède en début de partie deux jetons (respectivement noirs et blancs) placés au centre du plateau comme indiqué ci-dessous.



Chaque joueur va, à son tour, tenter de capturer les jetons de son adversaire, **le but étant d'avoir le plus de jetons de sa couleur lorsque le plateau est rempli ou de prendre tous les jetons de son adversaire.**
Pour prendre les jetons de son adversaire il faut placer un jeton sur une case vide de telle sorte qu'un ou plusieurs jetons adverses se retrouvent entre le nouveau jeton et d'autres de vos jetons : **ces jetons adverses deviennent les vôtres**. Dans l'exemple ci-dessous, le jeton noir qu'on est en train de placer capture le jeton blanc à sa gauche et les deux jetons sur la diagonale gauche supérieure, mais pas le jeton en bas car il n'est pas entre deux de vos jetons.



Vous pouvez encadrer les pions adverses dans **8 directions (gauche, droite, haut, bas et les diagonales)**.
Il n'y a pas de réaction en chaîne, les jetons qui viennent d'être pris n'engendrent pas d'autres prises.
Un placement est valide uniquement s'il prend au moins un jeton adverse. Tant que vous n'avez pas de placement possible, votre adversaire continue de jouer. Si vous avez un seul placement valide, vous devez le jouer, vous ne pouvez pas passer votre tour.

1.2 Fonctionnalités

Vous n'aurez pas besoin de faire une interface graphique, vous allez plutôt simuler le jeu sur un terminal en utilisant des caractères ascii, notamment avec '|' et '_'.

Jalon 0 (afficher le plateau)

- Afficher le plateau de jeu, le numéro (ou la lettre) de chaque ligne et chaque colonne ainsi que les pions de chaque joueur

Jalon 1 (jouer sauf capturer)

- Demander les noms des joueurs
- Afficher le nom du joueur courant et le nombre de jetons qu'il possède
- Demander la case que le joueur veut prendre
- Placer le jeton sur la case ou refuser le coup si la première condition n'est pas remplie (un jeton adverse à côté)
- Alterner les joueurs

Jalon 2 (capturer)

- Quand un jeton est placé, donner tous les jetons adverses qui doivent revenir au joueur courant
- Vérifier la deuxième condition de validité du coup (capture d'au moins un jeton adverse)
- Si le plateau est rempli, mettre fin au jeu et annoncer le vainqueur

Jalon 3 (assistance)

- Afficher à chaque tour toutes les cases jouables par le joueur courant
- Mettre fin au tour si aucun coup n'est possible, et au jeu si aucun joueur ne peut jouer

Fonctionnalités additionnelles

- Implémenter un joueur artificiel
- Implémenter la sauvegarde et le chargement d'une partie
- Utiliser la librairie SDL pour faire une interface graphique

2 Développement

Pour faire ce jeu vous devez respecter les contraintes décrites dans cette partie, tout écart devra être au préalable validé par votre évaluateur et justifié dans le rapport et la soutenance.

- Utiliser les instructions de base de la programmation (**if, for, while**...)
- Faire des entrées et des affichages utilisateur (cin, cout)
- Utiliser des fonctions pour bien segmenter le code
- Séparer votre code sur plusieurs fichiers
- Utiliser des structures :
  — Jeton : contenant une donnée sur la couleur (type au choix) et ses coordonnées
  — Joueur : contenant un nom, une liste chainée de jetons et le nombre de jetons actuels
  — Jeu : contenant deux Joueurs et un tableau à 2 dimensions de pointeurs de Jeton

3 Notation

La note du projet sera prise en compte dans la note finale de la matière Programmation et Algorithmique 1.

Rapport ⋆

Soutenance ⋆⋆

Compilable par l'évaluateur ⋆⋆

Propreté du code ⋆

Fonctionnalités du jeu ⋆ ⋆ ⋆⋆

Instructions **if, for, while**... ⋆⋆

Fonctions ⋆ ⋆ ⋆

Pointeurs ⋆ ⋆ ⋆

Liste chainée ⋆ ⋆ ⋆⋆

Structures ⋆ ⋆ ⋆⋆

Bonus (2) (Soutenance++, Fonctionnalités ++, Corruption, ...)

4 Remarques

— Il est très important que vous réfléchissiez avant de commencer à coder aux principaux modules, algorithmes et aux principales structures de données que vous utiliserez pour votre application. Il faut également que vous vous répartissiez le travail et que vous déterminiez les tâches à réaliser en priorité.

— Vous pouvez utiliser un outil de répartition et suivi de tâches, par exemple Trello, Azendoo ou Notion.

— Ne rédigez pas le rapport à la dernière minute, sinon il sera bâclé et cela se sent toujours.

— N'oubliez pas de tester votre application à chaque spécification implémentée. Il est impensable de tout coder puis de tout vérifier après. Ne visez pas directement votre oeuvre finale, testez les implémentations basiques sur une scène simple.

— Vos chargés de TD et CM sont là pour vous aider. Si vous avez des doutes/des difficultés sur un point, n'attendez pas la soutenance pour nous en parler.