

BUT2 FI R3.04

Projet Jeu de Go

Mikal Ziane

Le projet consiste à développer **et tester** un programme permettant de jouer une partie de go. Les règles de ce jeu ancestral sont détaillées sur <https://jeudego.org/>. Les thèmes du projet concernent notamment : le travail en équipe, la gestion des versions avec git et les thèmes vus en cours et en TP.

1 Travail à faire

Votre programme doit permettre à deux joueurs de s'affronter dans une partie. Un joueur pourra être un humain ou un programme. L'interface avec les utilisateurs se fera via l'utilisation d'un outil comme GoGUI, cf. <https://www.kayufu.com/gogui/>. Il faudra respecter les principales commandes du protocole GTP <https://www.lysator.liu.se/~gunnar/gtp/gtp2-spec-draft2.pdf>. Comme indiqué sur cette page <https://senseis.xmp.net/?GTP>, un bon moyen de se familiariser avec GTP est d'exécuter le logiciel gnugo en mode gtp sans serveur de go et de lancer les commandes à la main.

Qui, quoi et quand?

Votre projet doit être fait par une équipe de 3 ou 4 étudiants du même bi-groupe. Tout ce qui concerne le projet doit être hébergé sur un serveur git tel que github. Les contributions de chacun doivent apparaître, notamment dans les commit ou les gestion des "issues".

Ce qui est important n'est pas de supporter toutes les règles du jeu ni de faire un IA très forte mais de mettre en pratique les thèmes mentionnés ci-dessus.

Un grand nombre de tests sont disponibles sur différentes suites en ligne telle que <https://webdocs.cs.ualberta.ca/games/go>

Vous devez porter une attention particulière à la rédaction de votre dossier. Sa qualité est déterminante pour l'évaluation de votre travail. Sa composition doit être la suivante.

- Une page de garde indiquant le nom et **le groupe** des membres de l'équipe et le titre : "projet Go)".

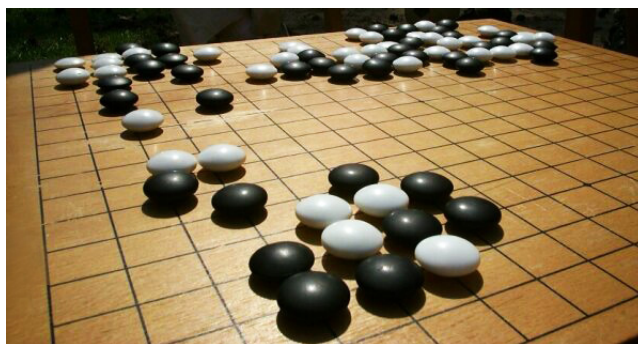


Figure 1: Partie de go en cours
source:https://commons.wikimedia.org/wiki/File:Go_board.jpg

- Une table des matières de l'ensemble du dossier.
- Une brève introduction qui indique ce qui marche et ne marche pas.
- Un diagramme d'architecture. **Ne montrez pas les méthodes ni les attributs.** Pensez aux dépendances.
- Une liste **synthétique**, en français, de ce qui a été testé par des tests unitaires :quelles fonctionnalités de quelles classes.
- Un bilan du projet (les difficultés rencontrées, ce qui est réussi, ce qui peut être amélioré).