Projet Angular

Jeu de go

Objectif du projet

L'objectif est de développer une application web permettant de jouer à une version simplifiée du jeu de Go.

Il n’est pas nécessaire de maitriser complètement le jeu du go pour réaliser l’application car on s’en tiendra à une version simplifiée du jeu pour faciliter la réalisation du projet.

Ce projet adopte une logique de cahier des charges : les objectifs sont fixés, mais la mise en œuvre n’est pas guidée étape par étape.

Vous devrez mobiliser et réutiliser les notions abordées dans les TP précédents pour concevoir votre solution.

Les choix techniques sont libres, mais doivent être maîtrisés et personnels (pas de délégation à une IA ou à d’autres personnes). Ils feront l’oeuvre d’une évaluation orale où il est nécessaire d’argumenter.

Rendu

Le projet est à réaliser de manière individuelle sur Git, avec des push fréquents.

Le code doit être commenté. Si des ressources externes sont utilisées elles doivent indiquées et sourcées en commentaires (et utilisées avec modération).

Jeu de go

Présentation

Le jeu de Go est l'un des plus anciens jeux de stratégie au monde encore pratiqués aujourd'hui.

Le jeu est originaire de Chine bien avant notre ère et s'est répandu en Asie au fil des siècles, ce n'est qu'au XXe siècle qu'il arrivera réellement en Europe.

Aujourd'hui, le Go est pratiqué dans le monde entier, et toujours reconnu pour sa profondeur stratégique malgré des règles simples.

Il est notamment devenu un jalon majeur dans l'histoire de l'intelligence artificielle, avec la victoire du programme AlphaGo contre l'un des meilleurs joueurs professionnels en 2016, presque 20 ans plus tard par rapport aux jeu d'échecs.

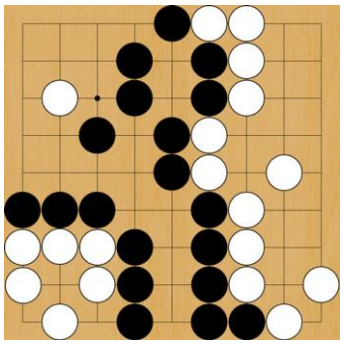
Principes de base

Le Go est un jeu de stratégie qui se joue à deux joueurs sur un plateau quadrillé.

Chaque joueur pose des pierres (noires ou blanches) sur les intersections du plateau, à tour de rôle.

L'objectif est de contrôler le plus de territoire possible tout en capturant les pierres adverses.

La capture de pierres se fait dès qu'elles n'ont plus d'intersections libres autour d'elles (ce qu'on appelle libertés).
Cela est valable pour une seule pierre ou un groupe de pierres de la même couleur.



Exemple d'une partie sur un plateau 9x9

On s’en tiendra à ces règles de base pour le projet, il existe bien sur d’autres règles mais nous les ignorons volontairement pour éviter de complexifier la réalisation.

Juste comprendre les bases suffira pour le projet : plateau, poser des pierres, tour par tour.

Pour les curieux :

- Règles complètes du jeu de go : https://jeudego.org/_php/regleGo.php
- Simulateur de partie à 2 joueurs : https://www.allaboutgo.com/play-go-9.html

Fonctionnalités demandées

Les fonctionnalités et l’approche du jeu seront simplifiées pour permettre une réalisation dans les temps :

- Le plateau de jeu sera au format 9x9.
- Les deux joueurs joueront sur la meme page web à tour de rôle.
- Les pierres seront retirées manuellement par les joueurs et les points seront comptés à ce moment là. On évite un algo de calcul automatique des “libertés”.

Réalisation

La réalisation sera séparées en différentes étapes à réaliser dans l'ordre :

1. Mise en place du plateau de jeu

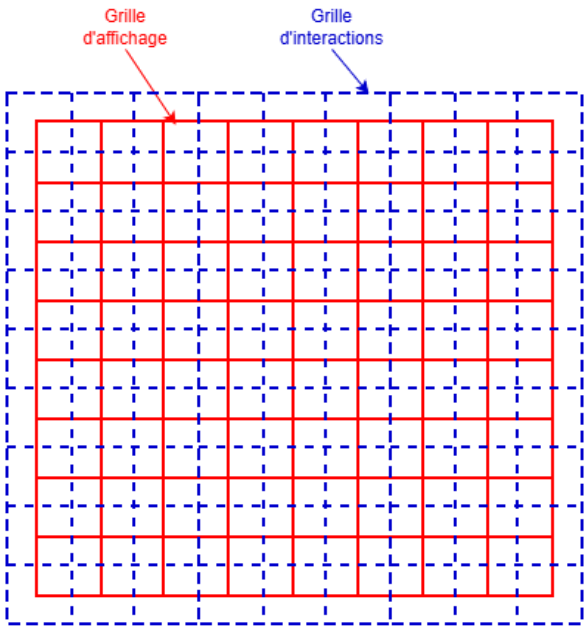
Tâches à réaliser :

- Création du projet Angular et du repo Git
- Création des composants de base : le plateau et les cellules
- Création d'un service qui gèrera les actions de jeu
- Création de l'affichage en mode "nouvelle partie" :
    - Plateau vide créé
    - Joueurs qui jouent tour par tour en posant une pierre chacun

Conseil pour l'affichage du plateau

Pour l'affichage du plateau, jouer sur les intersections peut paraître compliqué, mais une approche simplifiée peut-être utilisée en utilisant deux tables HTML :

- Un tableau de 9x9 : chaque case sera cliquable pour jouer, invisible pour l'utilisateur (grille d'interactions)
- Un tableau de 8x8 : placé derrière le précédent, les bordures affichent les intersections (grille d'affichage)



Gestion des états

Dans notre projet, l'état global du jeu (score des joueurs, tour actuel, plateau de jeu, etc.) sera centralisé dans un service Angular.

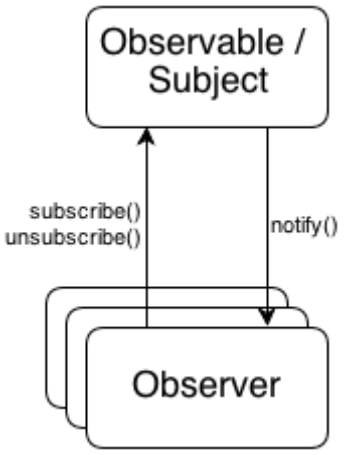
La gestion de ces état du jeu doit s'appuyer sur une approche réactive.

Il est possible d'utiliser soit des Signals (vu dans le précédent TP), soit des Observables avec RxJS.

Les observables

Les observables permettent la gestion des événements, de programmation asynchrone et de gestion de plusieurs valeurs émises au fil du temps.

Les observables définissent un fonction pour publier des valeurs, mais cette fonction n'est exécutée que lorsqu'un consommateur s'abonne à l'observable en appelant la méthode "subscribe" de l'observable. Cela permet aux abonnés de réagir automatiquement aux changements.



Angular utilise principalement la librairie RxJS pour pour les observables.

Guide FR sur les observables : https://angular.fr/services/observable_beginner

Guide EN sur les observables : https://v17.angular.io/guide/observables

2. Développement des actions de jeu

L'objectif est maintenant de réaliser les actions possibles pour chaque joueur pendant son tour :

- Clic gauche : place une pierre de sa couleur sur un emplacement libre
- Clic droit : retire une pierre adverse et augmente son score
- Bouton "Fin de tour" : changement de joueur
- Bouton "Passer" : si les deux joueurs passent consécutivement, la partie est terminée.

3. Développement global de l'application

Le but est maintenant d'enrichir l'application avec des fonctionnalités transverses :

- Sauvegarder les parties en JSON en localStorage (ou au choix)
- Possibilité de charger une partie existante (penser à faire une page qui liste les parties enregistrées
- Finalisation de l'interface avec un affichage élégant et fonctionnel (score actuel, état du jeu, navigation, joueur actif, etc...)

4. Fonctionnalités supplémentaires

Fonctionnalités à ajouter une fois toutes les étapes terminées et testées :

- Proposer le choix de la taille du plateau lors d'une nouvelle partie (19 par défaut).

- Afficher l'historique des coups, avec possibilité de naviguer dans les coups joués lors d'une partie.

- Export/import de parties via fichier JSON

- Proposer un thème sombre

- (avancé) Capture automatique : calcul automatique des libertés, plus besoin du clic droit pour retirer les pierres :
    - Gérer l'interdiction de pose d'un pierre
    - Gérer la capture d'une ou plusieurs pierre après la pose