



# Projet C : jeu de go

Réalisé Par :

AHMED BAAJDDA  
OUSSAMA MAZOUZ  
ALI OUABED  
YASSINE DBAICHI

Encadré Par :

MME MARIAM CHERRABI.

# Plan

- Introduction
- A propos du projet
- Méthodologie
- Les réalisations
- Conclusion

# I-Introduction

- Le jeu de Go est un jeu de stratégie millénaire originaire de Chine il y a plus de 2 500 ans., qui se joue sur un plateau avec des pierres noires et blanches. le Go est considéré comme l'un des jeux les plus complexes au monde en raison du grand nombre de configurations possibles et de la profondeur stratégique qu'il offre

## 2-A propos du projet

- I-Forme du jeu

Le plateau de notre jeu est une grille carrée de 9 lignes sur 9 colonnes. Les joueurs placent des pierres blanches et noires sur les intersections de la grille, en alternance.

Au lieu de l'utilisation des abscisses et des ordonnées des intersections nous avons ainsi un goban contenant des intersections numérotées de 0 et 80. Pour poser une pierre sur une intersection, l'utilisateur n'a besoin que de saisir un seul nombre, plutôt que deux nombres. Une intersection remplie par 0 représente une position vide

Une intersection remplie par 1 représente le blanc (cyan)

Une intersection remplie par 2 représente le noir (vert)



## • 2-Les règles pour jouer:

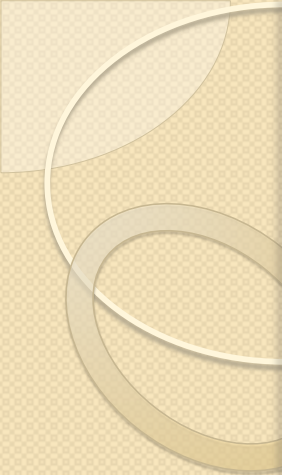
- Les joueurs placent leurs pierres à tour de rôle, en commençant par le joueur noir.
- le joueur doit poser une pierre dans une intersection dans le goban(qui a un indice entre 0 et 80),cette intersection doit être vide et le mouvement doit être légal. cad il ne faut pas qu'il soit un coup de suicide ou un coup de KO.
- Le but du jeu est de contrôler le plus grand territoire possible sur le plateau.
- Le joueur qui contrôle le plus de territoires et qui a capturé le plus de pierres gagne la partie.

### 3-la méthodologie :

- **I-approche de développement du jeu:**
- Nous avons créé tous les fichiers nécessaires au développement du jeu, y compris les fichiers .h et .c. Nous avons commencé par travailler sur l'affichage de la grille et nous nous sommes demandé comment remplir les intersections et changer les tours entre les deux joueurs. Nous avons défini les voisinages de chaque position (haut, bas, gauche et droite), puis nous avons travaillé sur le développement des règles du jeu (capture, suicide, ko). Ensuite, nous avons programmé des fonctions pour le CPU intelligent. Enfin, nous avons ajouté des couleurs et des tableaux de score pour améliorer l'esthétique et l'affichage du jeu.

## 2-Travail en groupe : avantages .

- La collaboration est la pierre angulaire de cette approche. Les membres du groupe travaillent ensemble pour mettre en commun leurs connaissances, leur expérience et leur expertise afin de trouver des solutions aux défis rencontrés. Cette méthode permet également de partager les responsabilités et de répartir les tâches de manière équitable.
- La motivation est également un élément clé de cette méthode de travail. Travailler en groupe peut être très stimulant, car chacun apporte sa contribution et contribue à la réussite du projet. De plus, la dynamique de groupe permet de maintenir un niveau d'engagement élevé tout au long du processus, ce qui est souvent difficile à obtenir lorsque l'on travaille seul.

- 
- Pour arriver aux solutions les plus efficaces et les plus rapidement possibles, les membres du groupe se réunissent chaque jour autour d'une table pour discuter et décomposer le problème en éléments plus simples. Cette méthode permet de mieux comprendre le problème dans son ensemble et de trouver des solutions créatives et innovantes.
  - En fin de compte, travailler en groupe était extrêmement bénéfique pour nous afin de relever des défis complexes et à atteindre des objectifs ambitieux. La collaboration, la motivation et l'esprit d'équipe sont les clés du succès de cette méthode de travail.
  - ***Travail en groupe était un choix judicieux.***





## 4-Presentation de la réalisation :

fonctions	objectif	L'algorithme utilise	Les concepts de C
Check_group_lib()	Donner deux liste: des amis de l intersection i ;et une autre liste de degré de liberté de groupe	Depth first search	Récurtivité Les boucles Les instructions de base
End_game()	Fin de partie	Deux conditions a vérifier (les deux joueurs pass consécutivement)	Les boucles Les instructions de base
Choose_first()	Le joueur a commencer premièrement	aléatoire	random

fonction	objectif	L algorithme utilise	Les concepts de C
pass()	Passer le rôle a l adversaire	Fonction simple	Les instructions de base
Final_score()	Le score final	Addition	Les boucles Les instructions de base
territory()	Les territoires de chaque joueur	Flood fill algorithm	Les boucles Récursivité Les instructions de base
Count_stones()	Les pierres sur le goban de chaque joueur	Parcours par boucle for	Les boucles Les instructions de base

# Les règles de jeu :

fonction	objectif	Algorithm	Les concepts de C
Capture()	Capter tout groupe des pierres qui ont un degré de liberté nul	Parcourir le plateau de jeu, rassembler les pierres reliées entre elles, vérifier si elles sont complètement encerclées par l'adversaire, puis les retirer du plateau ; pour chaque pierre capturée, accorder un point à l'autre joueur.	Récurtivité Les boucles Les instructions de base

**suicide**

**Eviter le coup  
interdit « suicide»**

**1/ Vérifiez les libertés des pierres voisines : si une pierre voisine a une liberté ouverte, le mouvement n'est pas un suicide.**

**2/ Vérifiez si le mouvement crée un nouveau groupe : si le mouvement crée un nouveau groupe avec des libertés, alors il n'est pas un suicide.**

**3/ Vérifiez si le mouvement capture les pierres de l'adversaire : si le mouvement capture une ou plusieurs pierres de l'adversaire, alors ce n'est pas un suicide**

**Récurtivité  
Les boucles  
Les instructions de  
base**

**Ko\_prevent()**

**Eviter le coup inerdit  
KO**

**Ko\_move(): Une  
comparaison ente le  
goban actuel et le  
goban avant précèdent**

**Les boucles  
Les instructions de  
base**

## **AI Fonctions**

**NB : ces fonctions sont utilisé par ordre de priorité**

**Machine\_capture()**

**Fonction dans AI**

**Si un groupe de  
pierres du jouer a un  
seul degré de liberté  
CPU va l'attaquer**

**Récurtivité  
Les boucles  
Les instructions de  
base**

**Save\_machine()**

**Fonction dans AI**

**Si un groupe de  
pierres du CPU a un  
degré de lib = 2 ;le  
CPU va le sauver**

**Récurtivité  
Les boucles  
Les instructions de  
base**

<code>minimize_deg_lib()</code>	Fonction dans AI	Teste si le « save » va minimiser le degré de liberté du groupe ciblé	Récurtivité Les boucles Les instructions de base
<code>surround()</code>	Fonction dans AI	Il détermine le groupe du joueur humain qu'a le moins degré de liberté Ensuite il essaye de l'entourer	Récurtivité Les boucles Les instructions de base


**NB :**

La notion des variables externes globales est très utilisé, principalement pour sauvegarder les données

# Conclusion

- Contrainte du temps
- Contrainte du comprendre les cas spécifiques du jeu
- Contrainte d'implémenter les idées
- Retours d'expérience :
  - au niveau académique :
    - ✓ Apprendre à programmer en C à grande échelle
    - ✓ Apprendre à manipuler des codes de grande taille
    - ✓ Apprendre à gérer son temps et travailler en groupe



- 
- Au niveau personnel :
    - Découvrir un nouveau intéressant jeu de réflexion GO
    - La régularité et la patience
    - La démarche pour dissolver des problèmes délicats



**Merci pour votre attention**