

# ALPHASQUAD

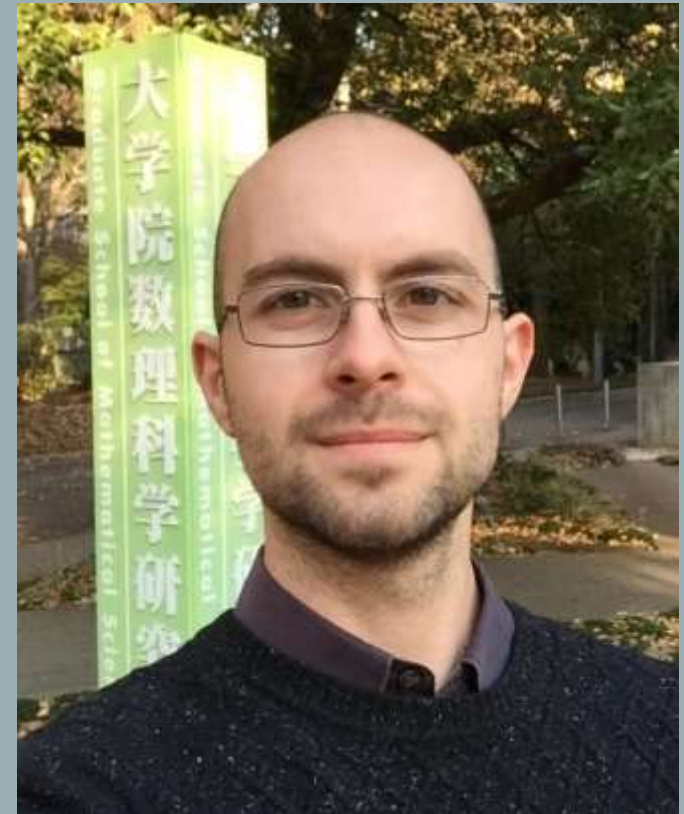
Dalil Chablis

Darko Djordjevic

Youssef Hassanein

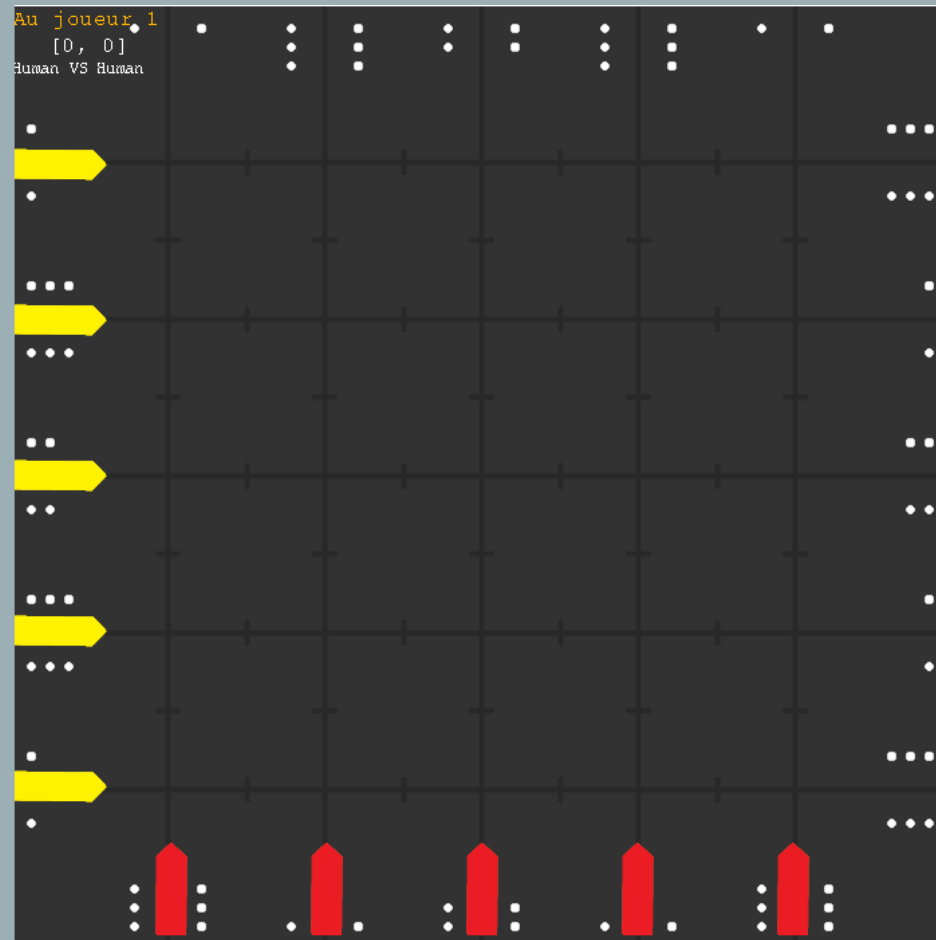


# INTRODUCTION



Adrián Jiménez Pascual

# INTRODUCTION



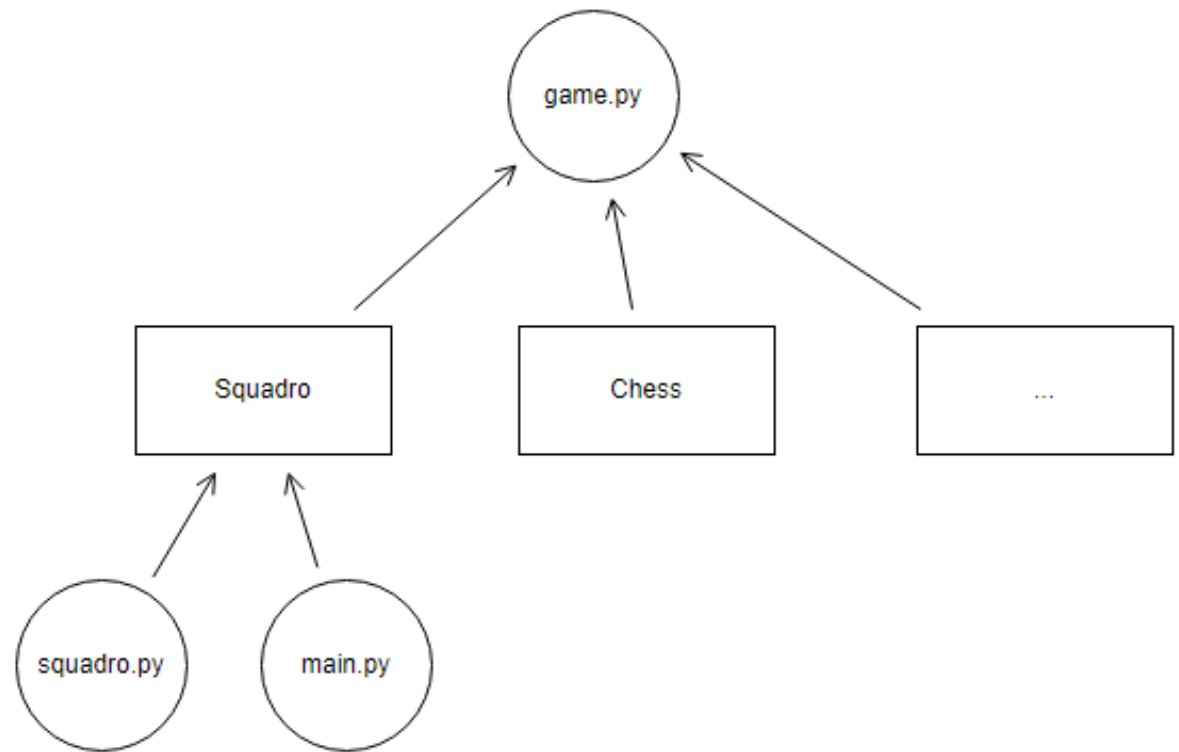
# SOMMAIRE

- Structure de projet
- Implémentation du jeu Squadro et des différents joueurs (humain ou IA)
- Interface graphique moderne (Pygame)
- Joueurs intelligents (Alpha-beta pruning et Monte Carlo Tree Search)
- Mode multiplayer (Sockets)

# STRUCTURE DE PROJET

Structure de projet générique:

- game.py contient les fonctions communes/spécifiques à tous les jeux de plateau



# IMPLEMENTATION DU JEU

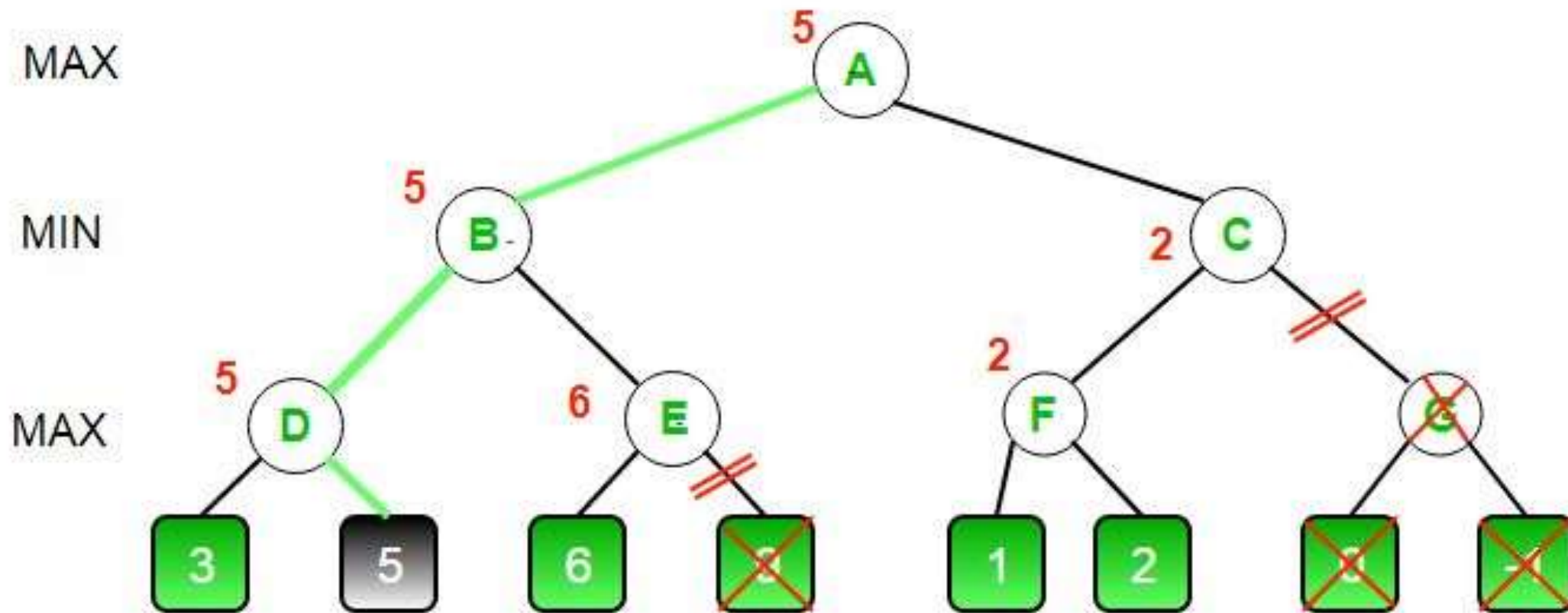
Fonctions indispensables:

- getCoupsValides(jeu)
- saisieCoup(jeu) (spécifique à un joueur)
- joueCoup(jeu)
- finJeu(jeu)
- afficheJeu(jeu) / draw\_board(jeu)

# INTERFACE GRAPHIQUE

- La variable GUI dans game.py → mode console / GUI
- draw\_board(jeu):
  - on blit le background
  - on blit les images
  - on set rect.topleft du pion qui s'est déplacé
- Joueur humain → détection collision rect / click souris

# ALPHA-BETA PRUNING



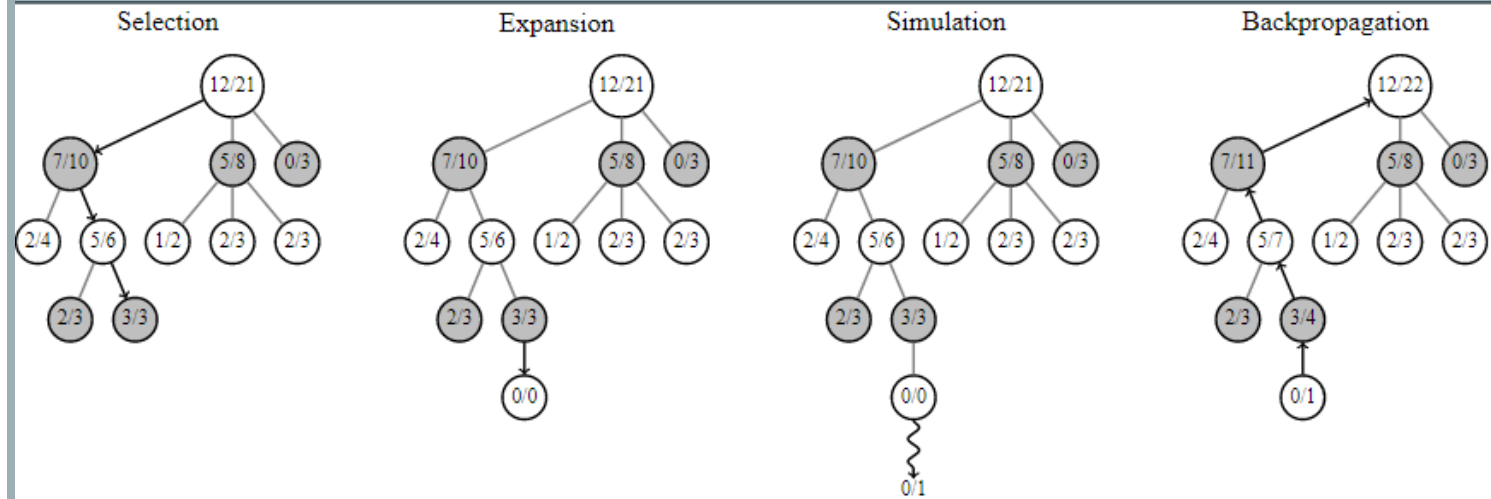
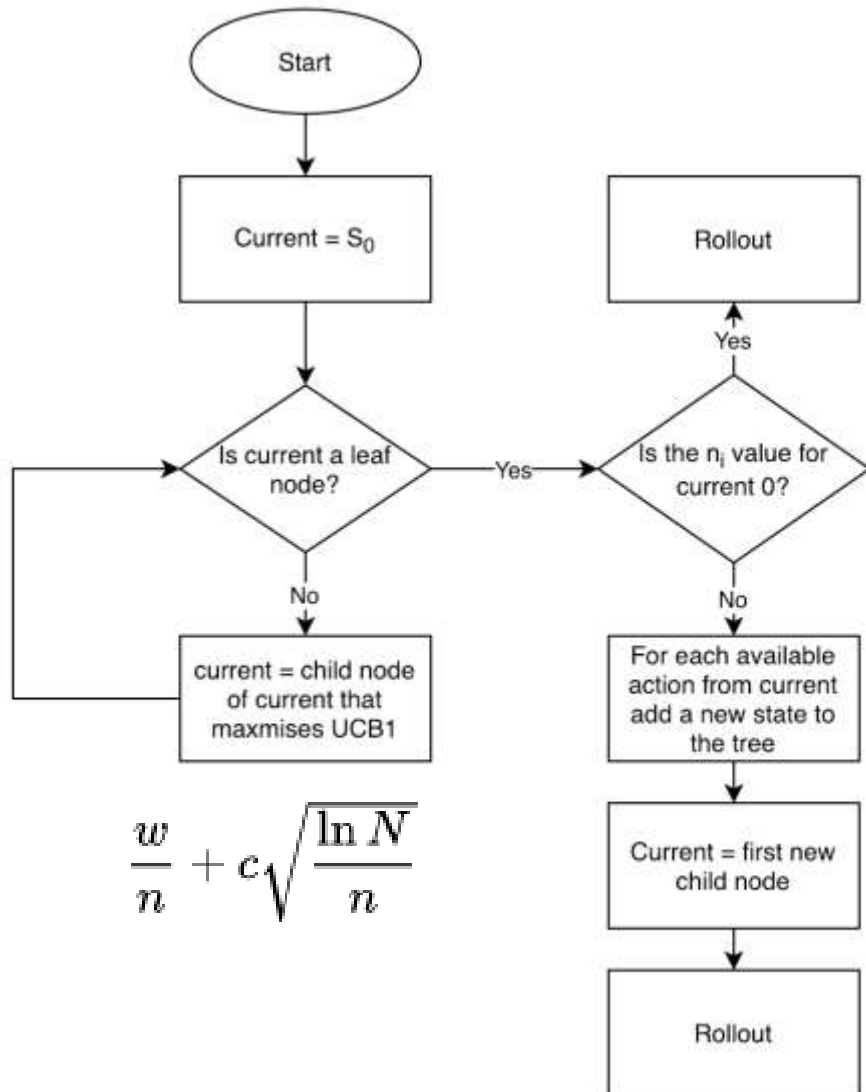
$$O(b^d)$$
$$\Omega(b^{\frac{d}{2}})$$



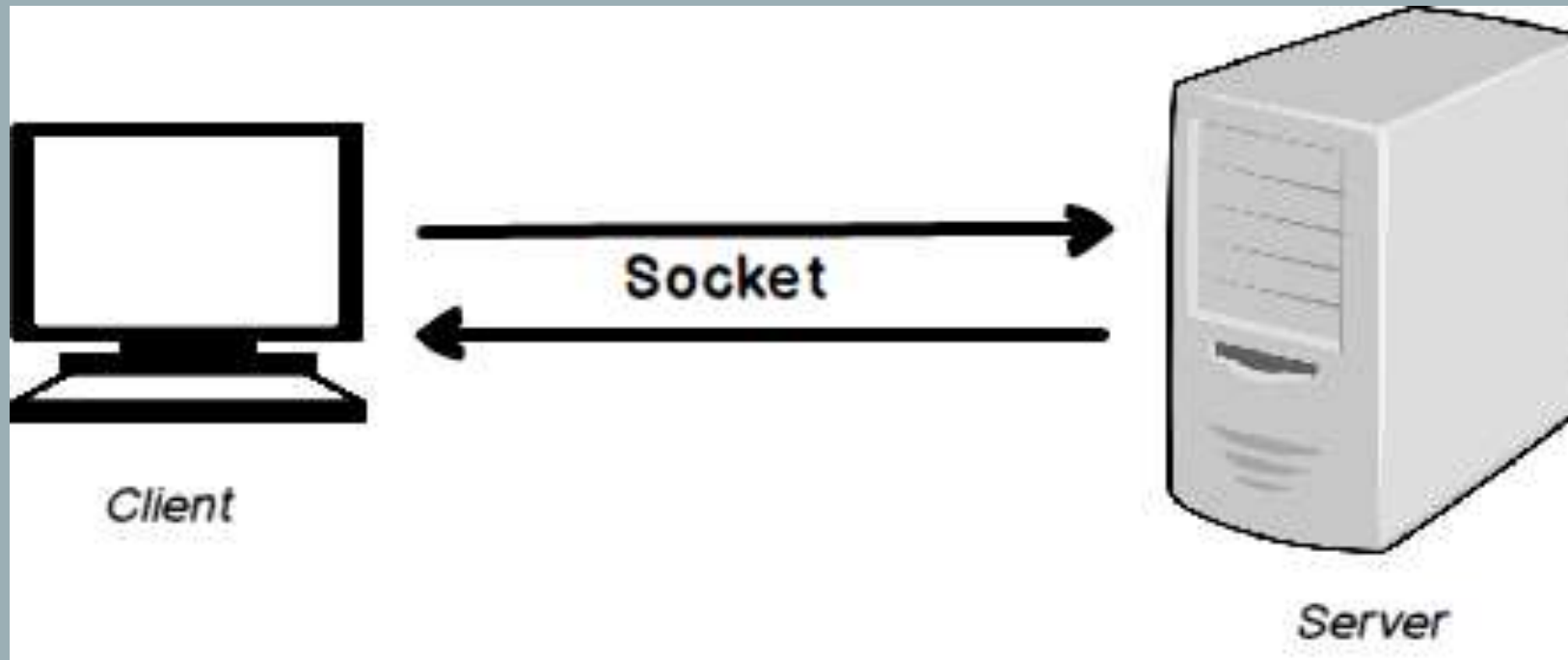
# HEURISTIC

- Fonction d'évaluation → produit scalaire entre un vecteur de poids et un vecteur de fonctions d'évaluations élémentaires
- Nécessité d'avoir un domain knowledge du jeu en question
- Optimisation du vecteur de poids grâce à un algorithme génétique:
  - Initialisation du population avec des vecteurs de poids random (gènes)
  - Fitness function
  - Selection des meilleurs individus
  - Crossovers et mutations

# MCTS



# SOCKETS



DEMONSTRATION