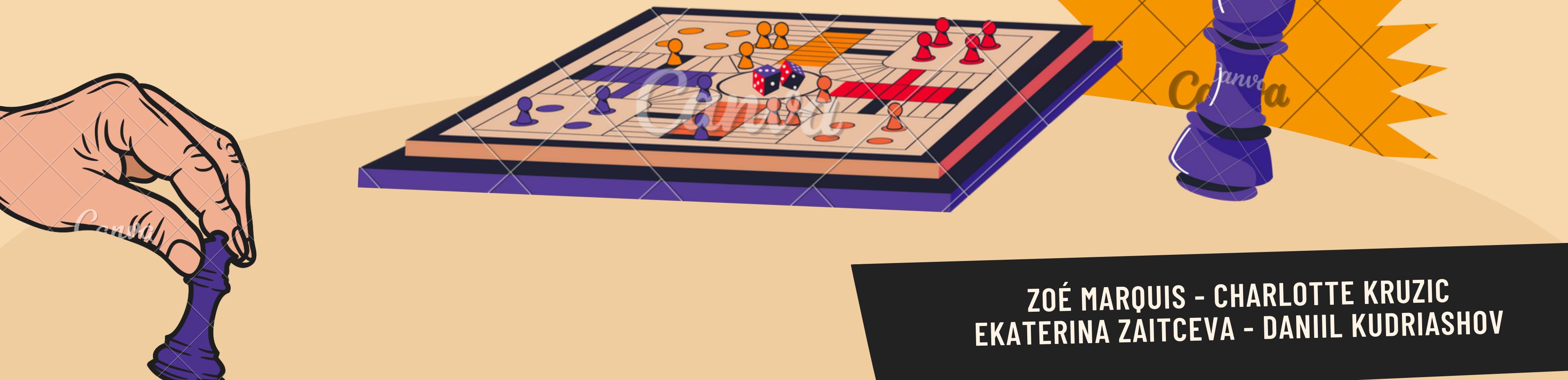
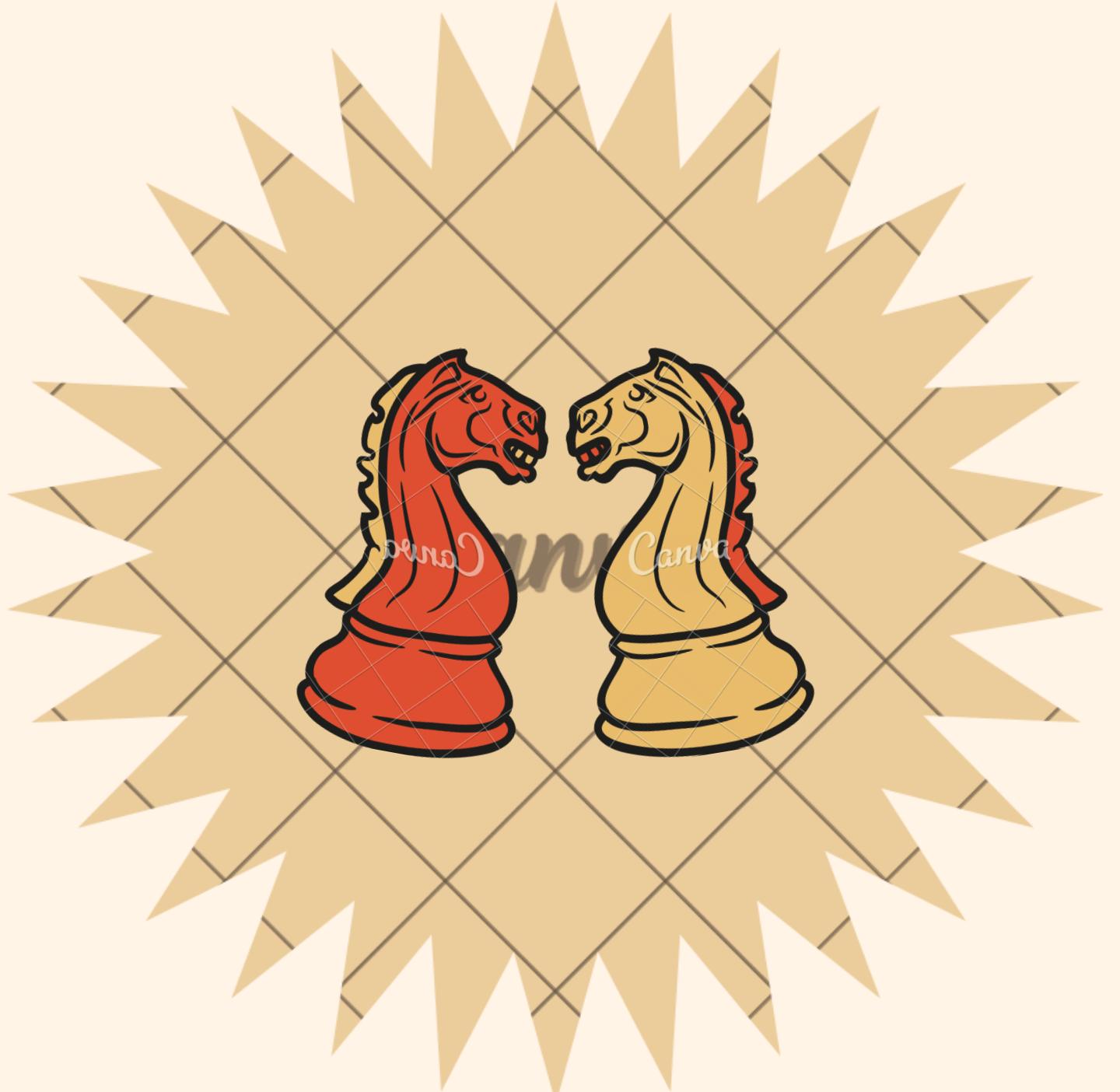


**CRÉATION DE JOUEURS AUTOMATIQUES  
ET NOUVELLES RÈGLES  
POUR JEUX DE PLATEAU  
AVEC APPRENTISSAGE PAR RENFORCEMENT**

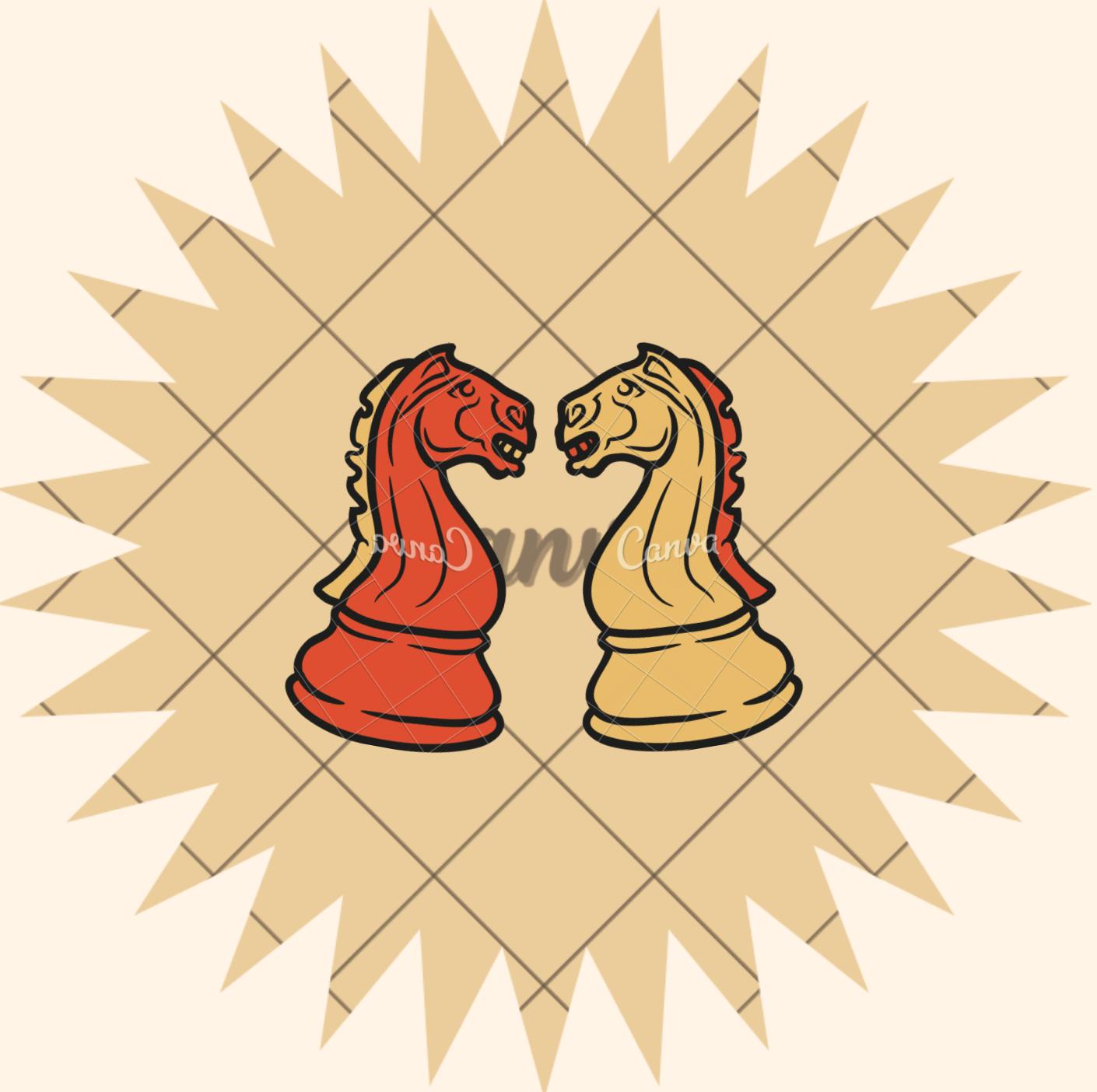


ZOÉ MARQUIS - CHARLOTTE KRUZIC  
EKATERINA ZAITCEVA - DANIIL KUDRIASHOV

# **CONTEXTE & OBJECTIFS**



- 1. Prise de décision dans environnement incertain**
- 2. Planification stratégique**



# **CONTEXTE & OBJECTIFS**

- 1. Entrainer des agents autonomes**
- 2. Explorer des variantes de règles**
- 3. Analyse des impacts sur la jouabilité et l'équilibre du jeu**

# FEUILLE DE ROUTE

- **Modélisation**
- **Entraînement**
- **Personnalisation**
- **Analyse statistique**
- **Jeu**



# **CHANGEMENT DE DIRECTION À LA MOITIÉ DU SEMESTRE**



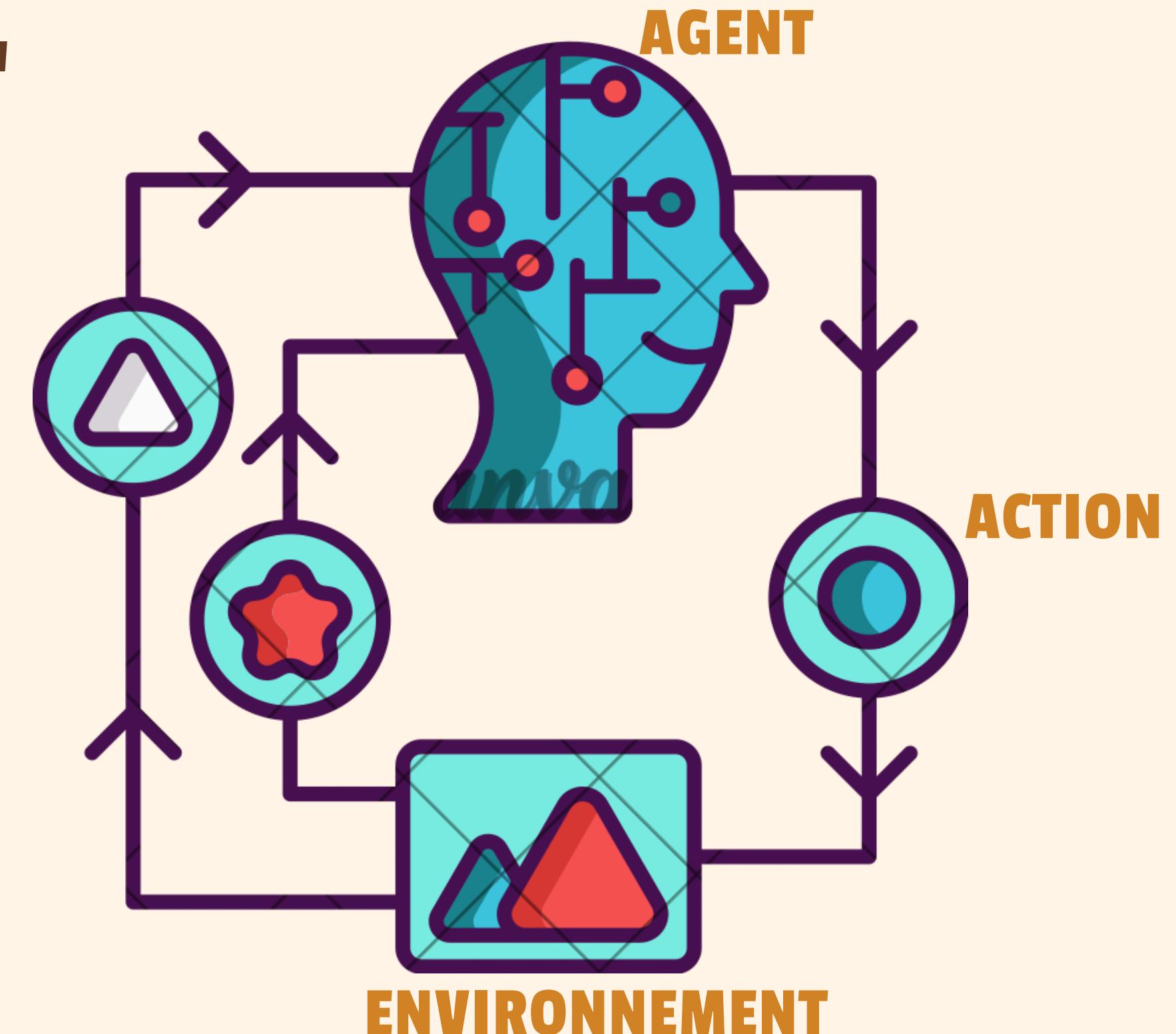
# PETITS CHEVAUX

- Les pions partent de l'écurie pour atteindre l'arrivée.
- Un 6 au dé est nécessaire pour sortir un pion.
- Un pion sur une case occupée renvoie l'adversaire à l'écurie.
- Le premier à amener tous ses pions gagne.



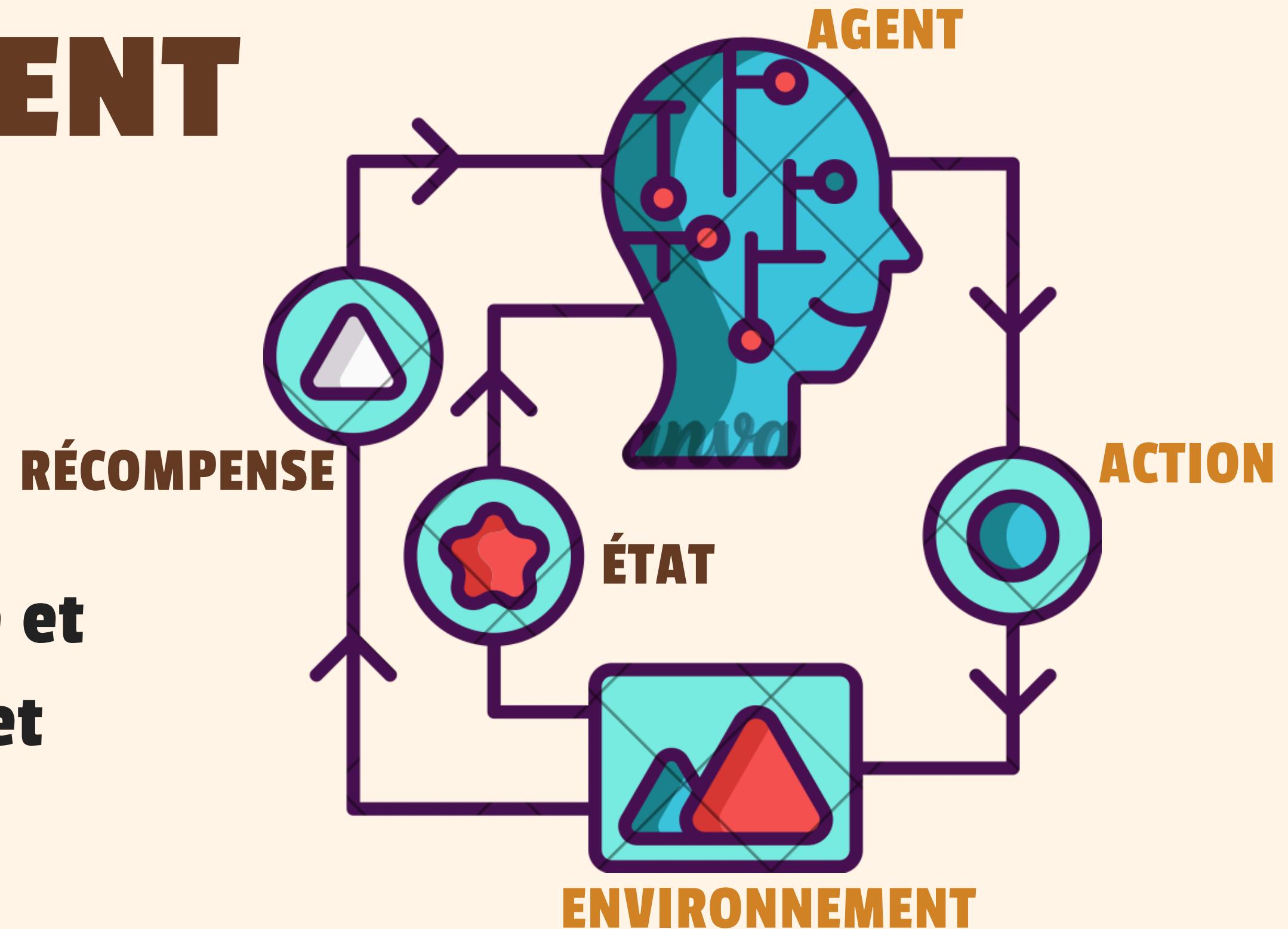
# APPRENTISSAGE PAR RENFORCEMENT

- Agent
- Espace d'actions
- Espace d'observation
- Environnement

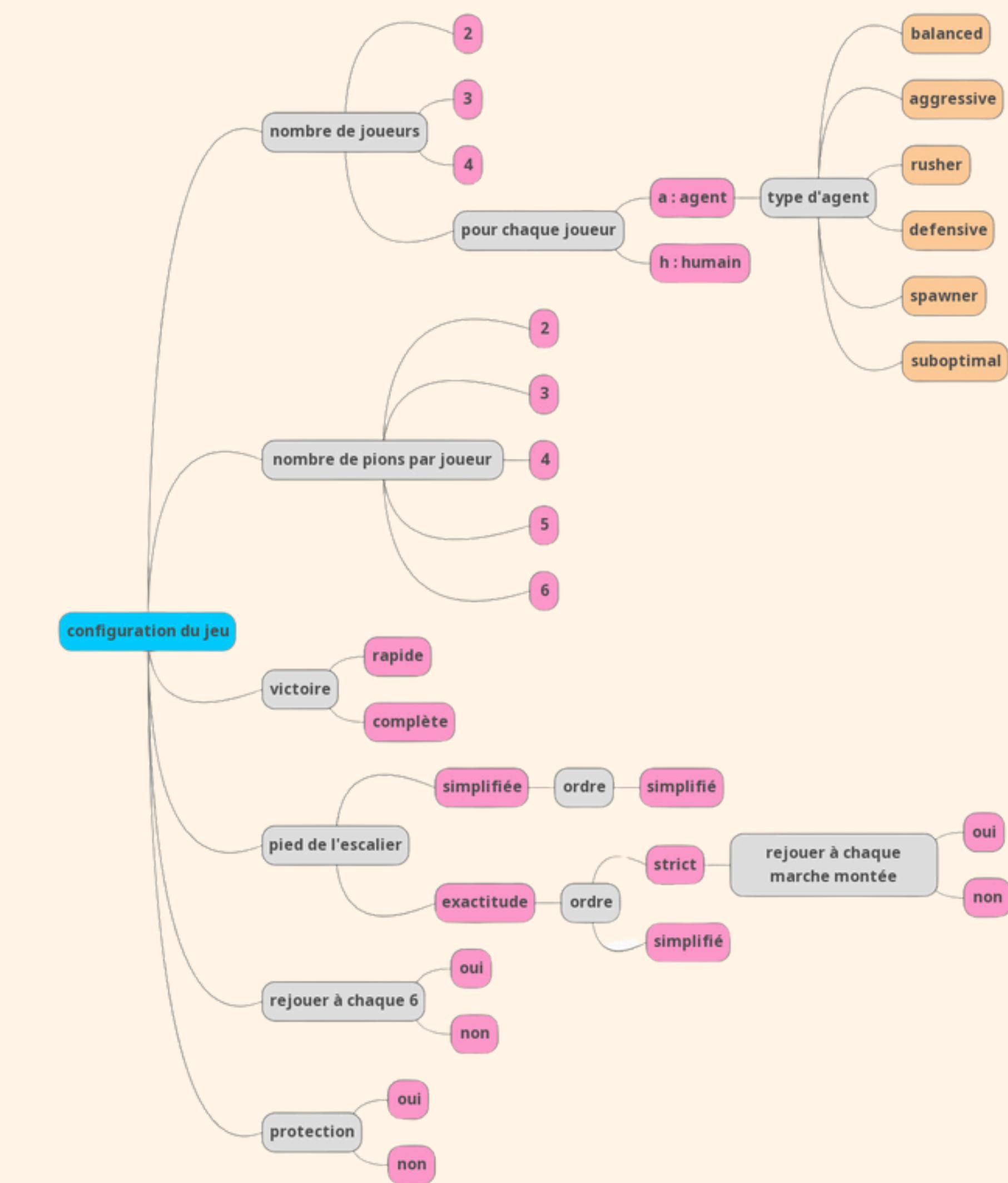


# APPRENTISSAGE PAR RENFORCEMENT

- Observation de l'environnement
- Choix de l'action
- Environnement évolue et retourne observation et récompense



# VARIANTES DES RÈGLES ET DES AGENTS



# DEMONSTRATION

## Développement d'agents autonomes et création de nouvelles règles pour jeux de plateau via l'apprentissage par renforcement

Projet réalisé dans le cadre de l'UE "Projet Master" en Master 2 Sciences des Données et Systèmes Complexes par:

- KRUZIC Charlotte
- MARQUIS Zoé
- KUDRIASHOV Daniil
- ZAITCEVA Ekaterina



The screenshot shows a terminal window with the following interface elements at the top:

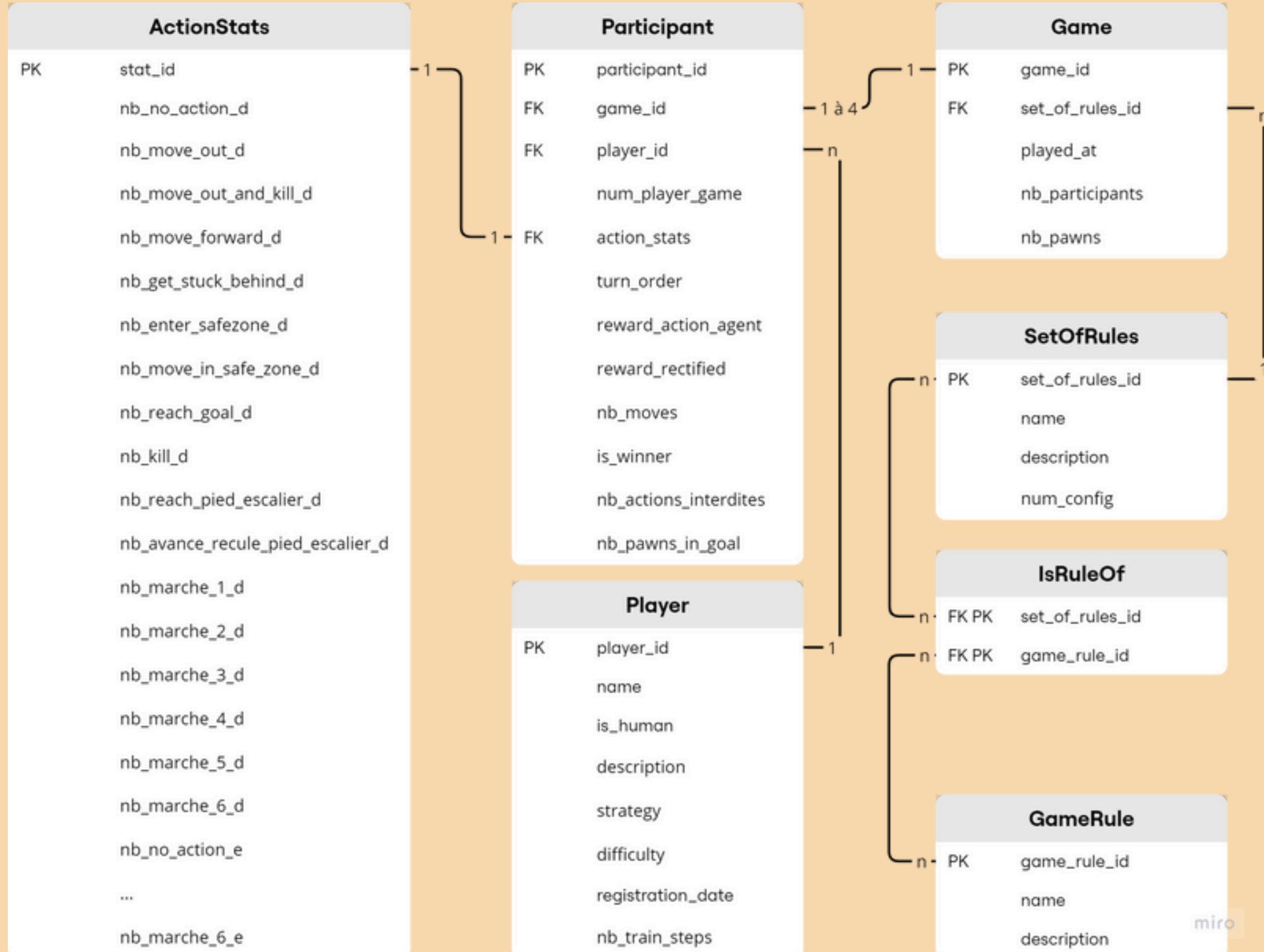
- OUTPUT
- PROBLEMS
- DEBUG CONSOLE
- TERMINAL
- PORTS
- SPELL CHECKER
- COMMENTS

On the right side of the terminal window, there is a tab bar with the text "Python-game" and icons for closing, minimizing, and maximizing the window.

The terminal content is as follows:

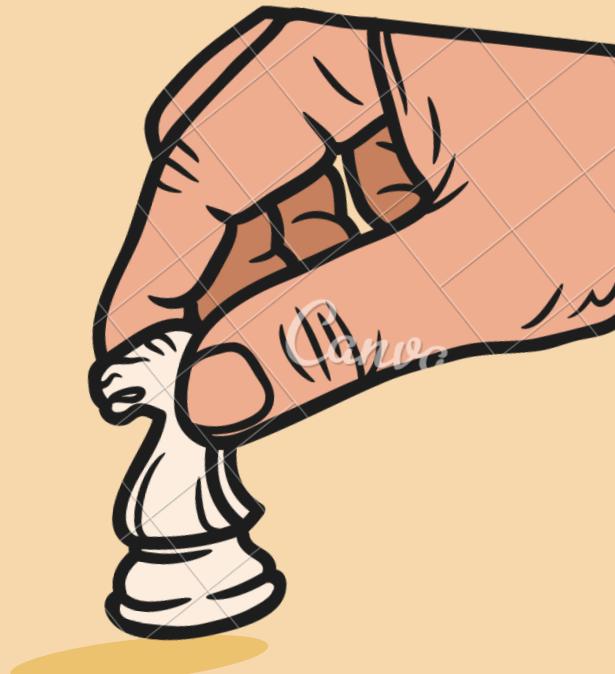
```
°(ludo_env) kotia@debian:~/Uni/S3/Projet_Master/code/kity/game$ python3 play_pygame/play.py
Bienvenue dans le jeu ! Configurons votre partie.
Entrez le nombre de joueurs (entre 2 et 4) : 4
Le joueur 1 est-il humain ou agent ? (h pour humain / a pour agent) : h
```

# GESTION DES DONNÉES

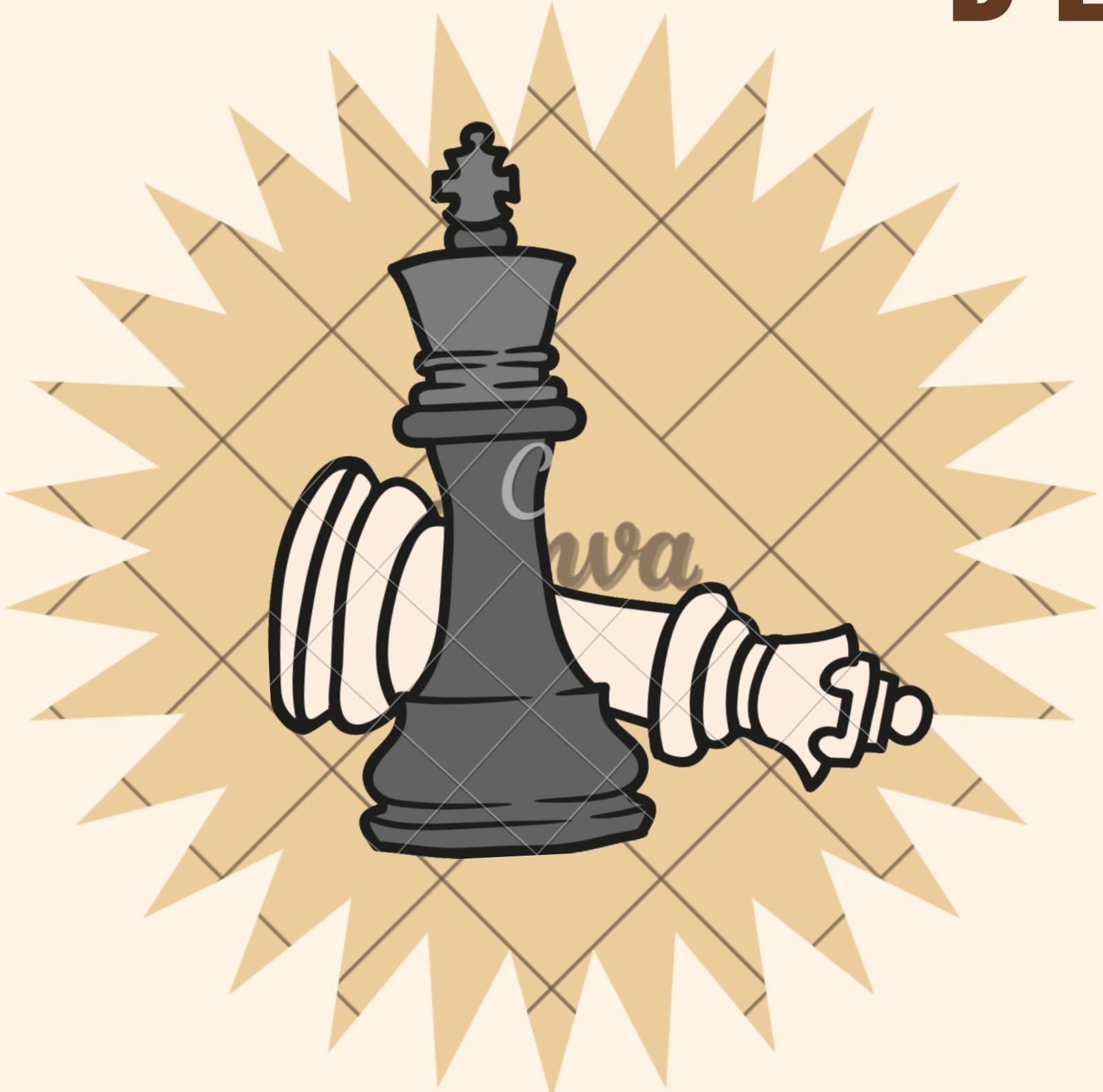


- Objectifs :

- **Collecter**
- **Centraliser**
- **Analyser**

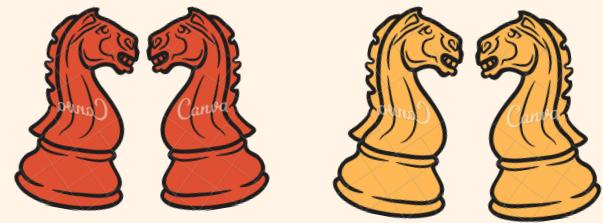


# RÉCOLTE DES DONNÉES D'ENTRAÎNEMENT DES AGENTS

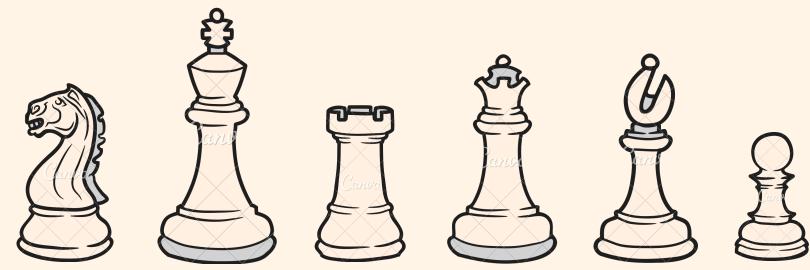


- 1. Configuration des parties**
- 2. Simulation automatique**
- 3. Collecte des données**

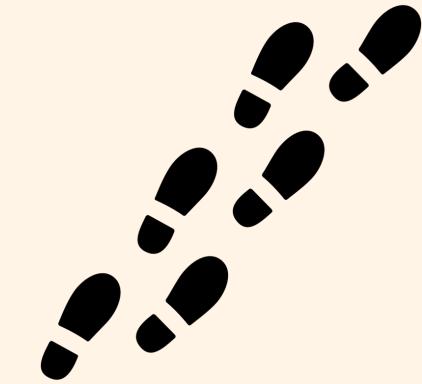
# ANALYSE ENTRAINEMENT



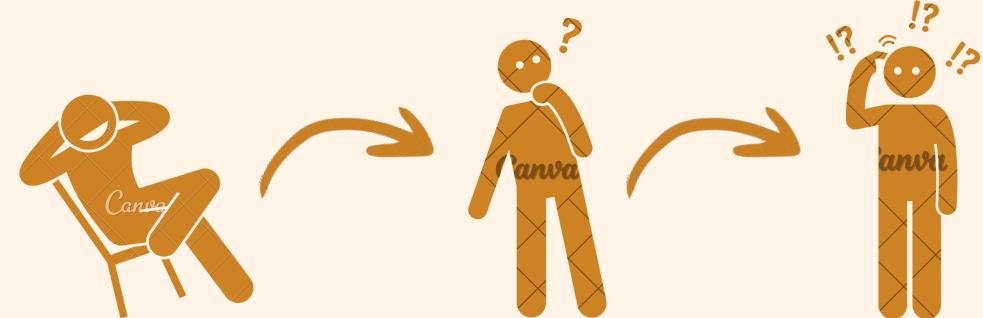
Agents identiques



Tous les types d'agents



Différents pas d'entraînement



Configurations complexité croissante

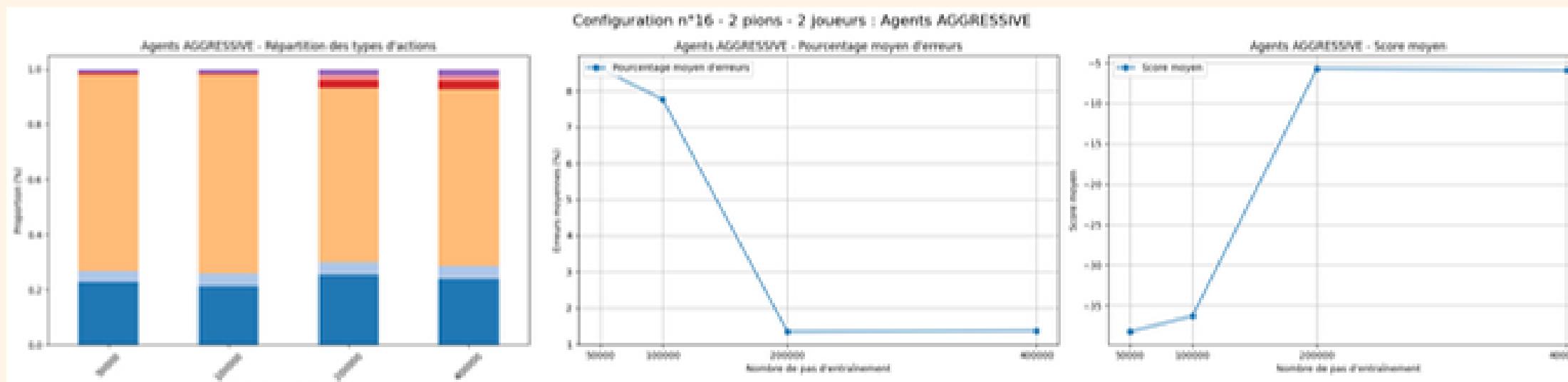


Différents nombres de joueurs et pions

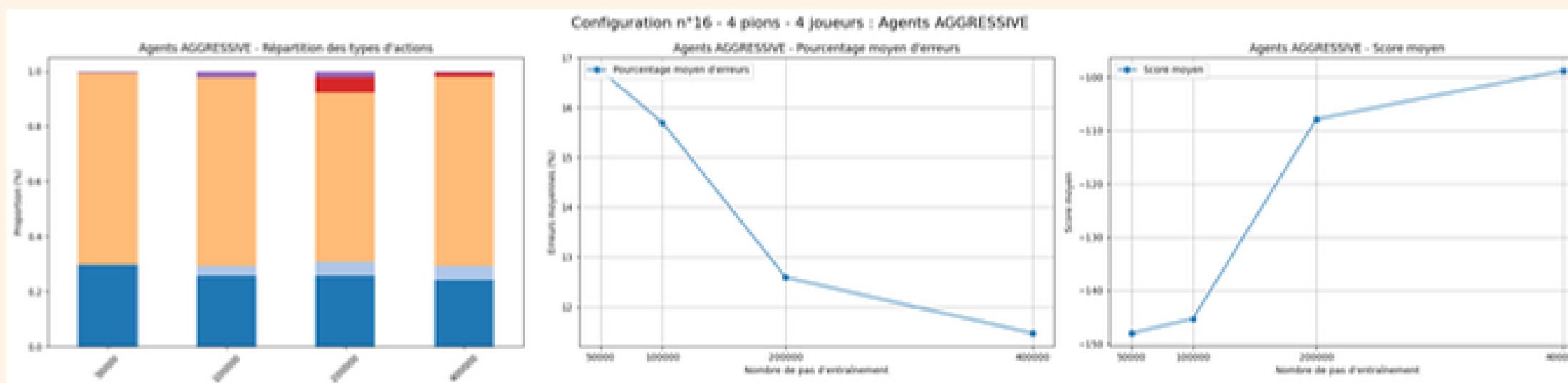
# ANALYSE ENTRAINEMENT

## CONFIGURATIONS SIMPLES

Types d'actions
nb_no_action_d
nb_move_out_d
nb_move_forward_d
nb_enter_safezone_d
nb_move_in_safe_zone_d
nb_reach_goal_d



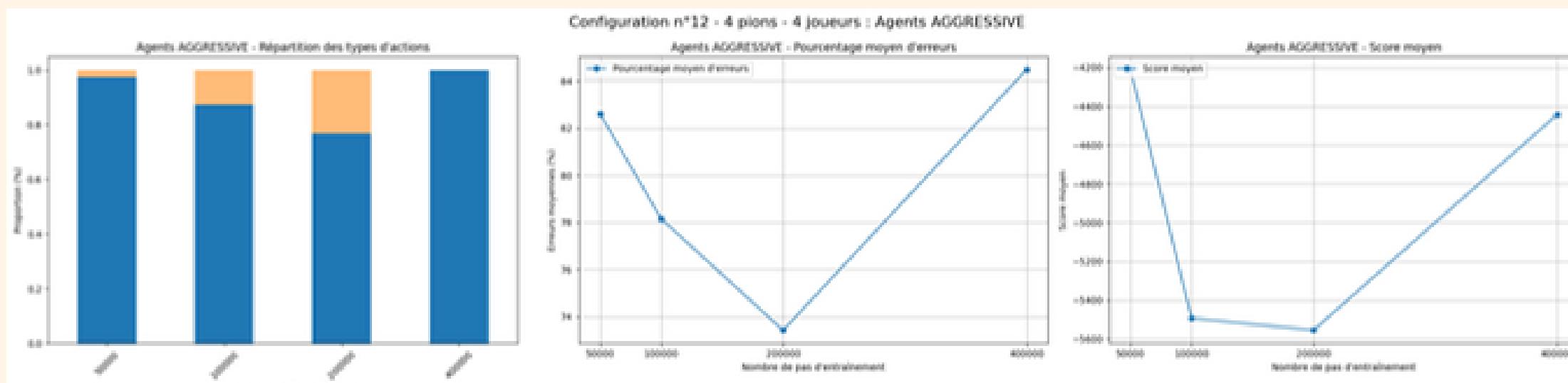
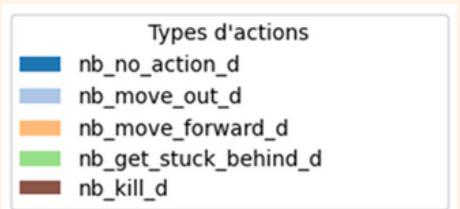
- **Performances claires et stables**
  - erreurs
  - scores
- **Bonne capacité d'adaptation**



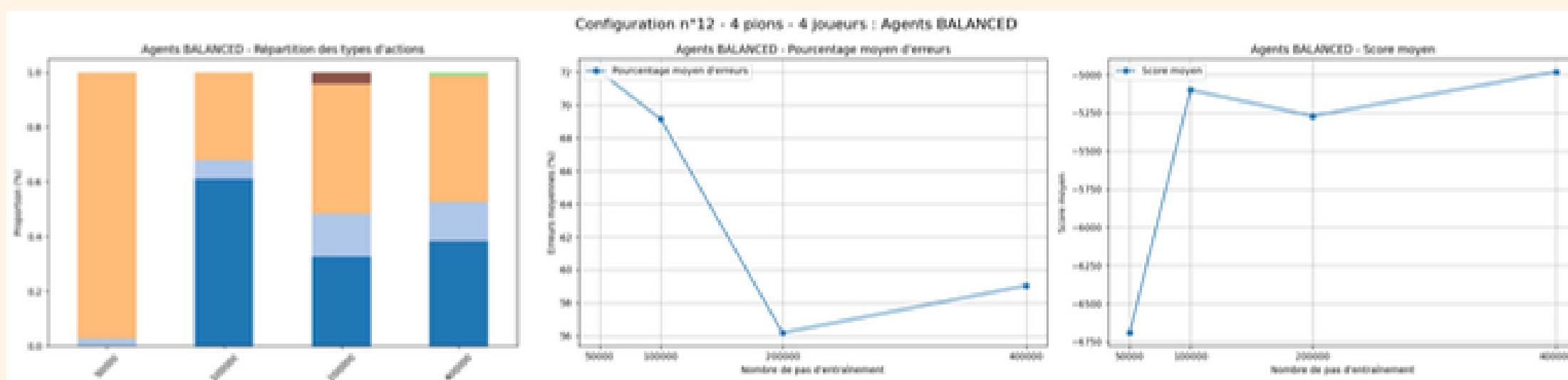
- **Ralentissement de l'entraînement**

# ANALYSE ENTRAINEMENT

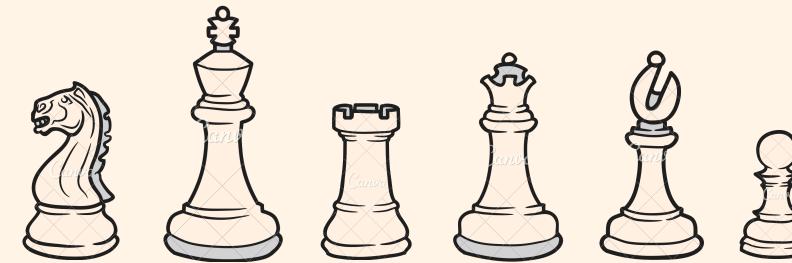
## CONFIGURATIONS COMPLEXES



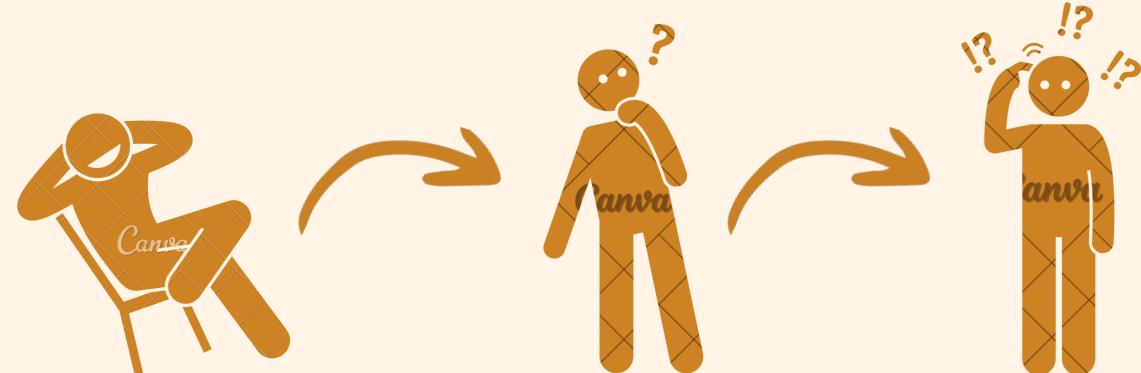
- **Introduction de règles :**
  - **Performances moins stables**
  - **Differences entre agents**



# ANALYSE DES AGENTS ENTRAINÉS



**Performance au sein d'une configuration fixe**



**Métriques de jouabilité**



**Satisfaction des joueurs selon les règles du jeu**

# RÉCOLTE DES DONNÉES POUR L'ANALYSE DES AGENTS

**1. Configuration des matchups: 200 000 pas / 100 parties par configuration**

**2. Types de confrontations**

**a. Parties à 2 joueurs**

**i. Toutes les combinaisons possibles d'agents**

**b. Parties à 4 joueurs**

**i. Format "chacun pour soi" (1v1v1v1)**

**ii. Format 2v2 avec paires d'agents identiques**

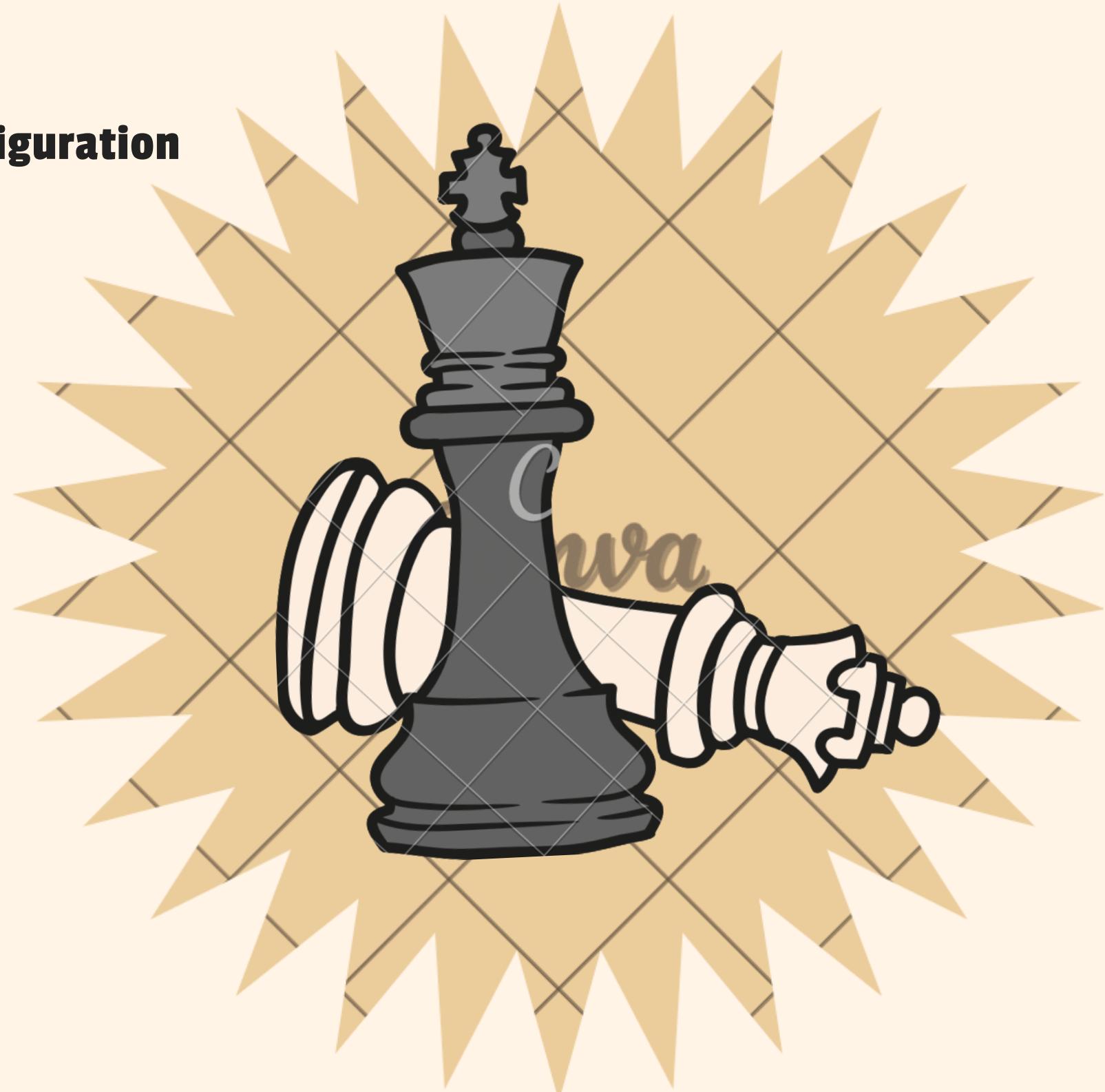
**3. Données collectées**

**a. Taux de victoire**

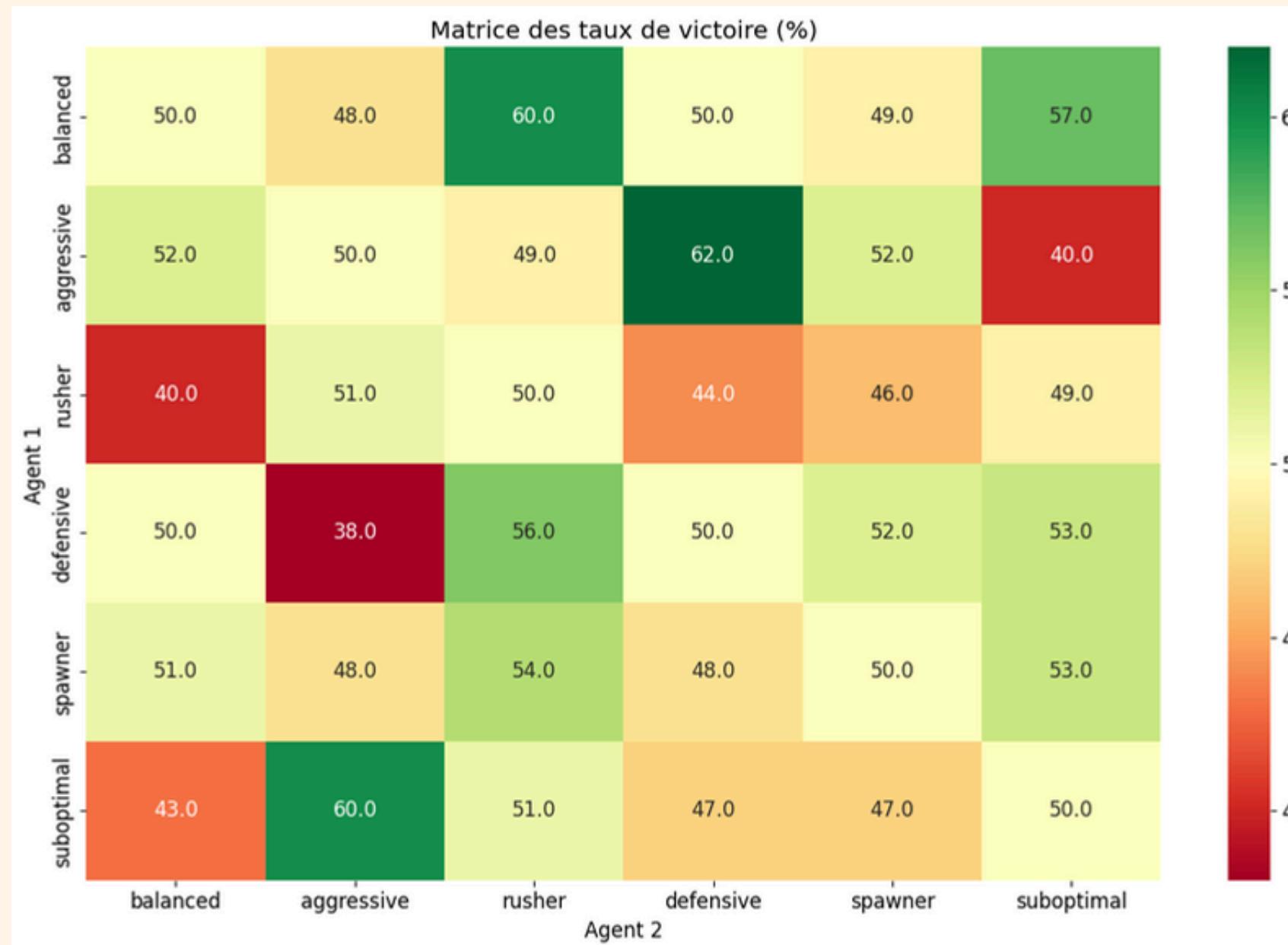
**b. Actions effectuées**

**c. Positions finales**

**d. Métriques de satisfaction**

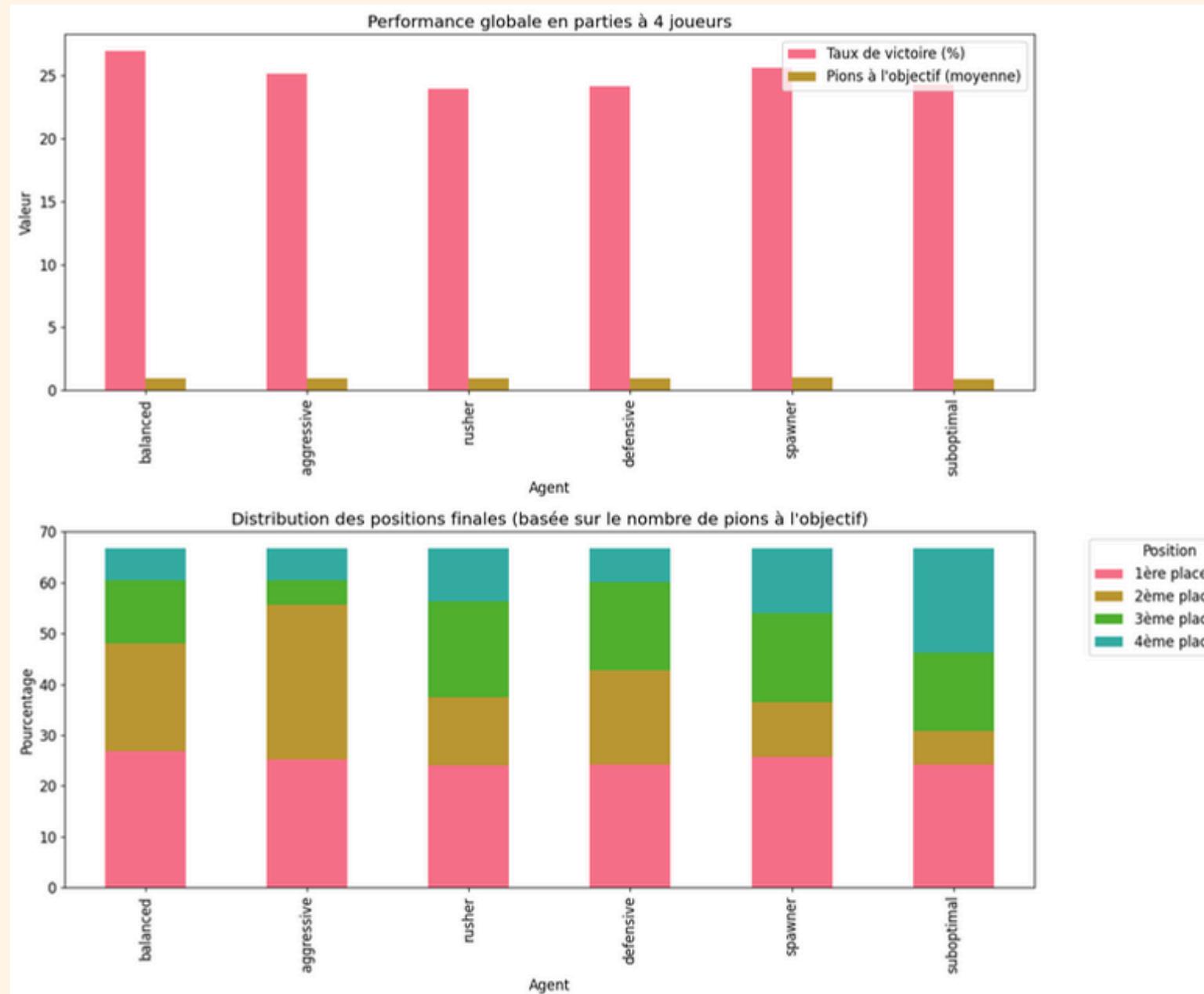


# PERFORMANCE RELATIVE DES AGENTS



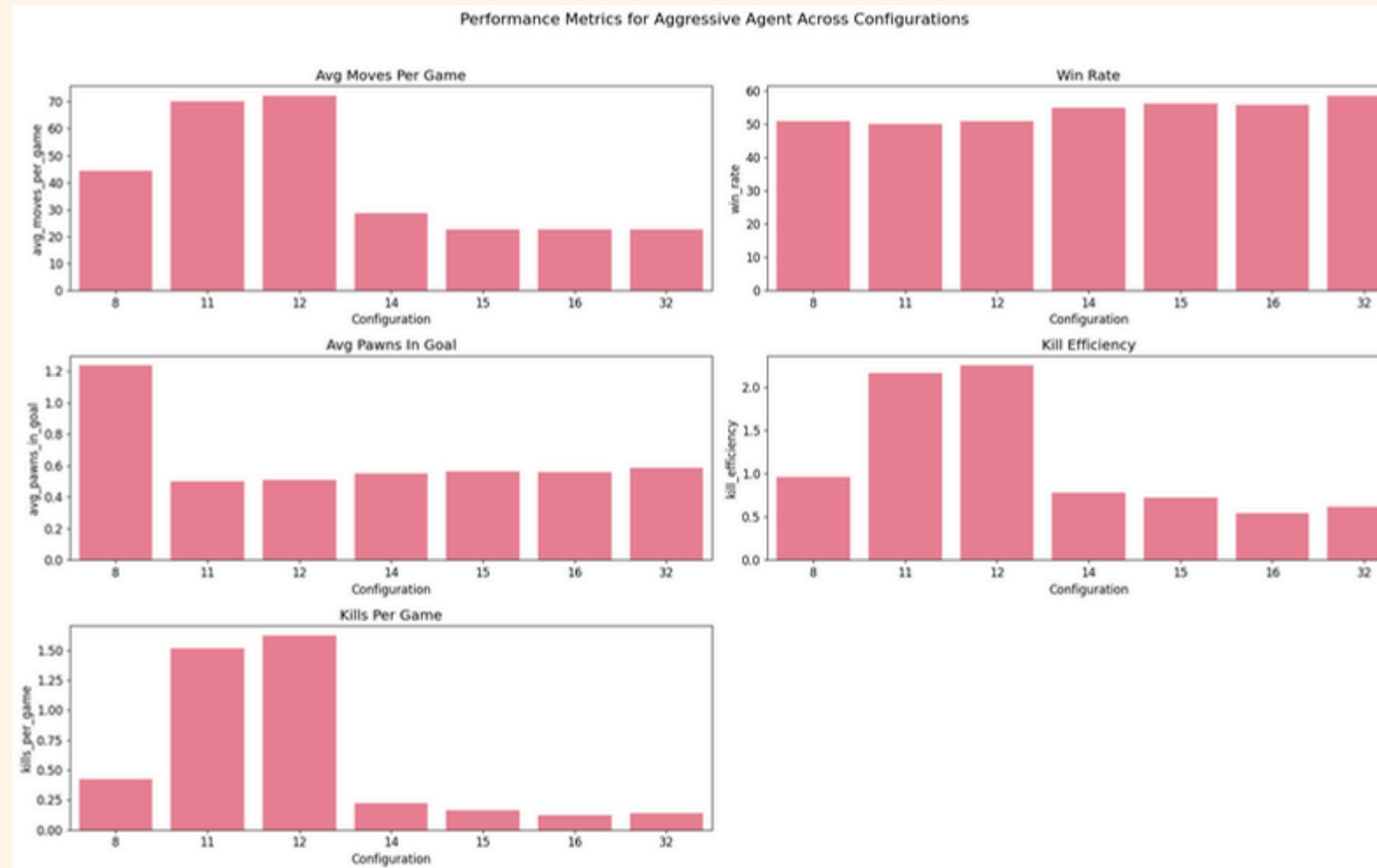
- Différences significatives entre stratégies
- Certaines stratégies sont plus efficaces contre des types spécifiques d'adversaires

# PERFORMANCE RELATIVE DES AGENTS



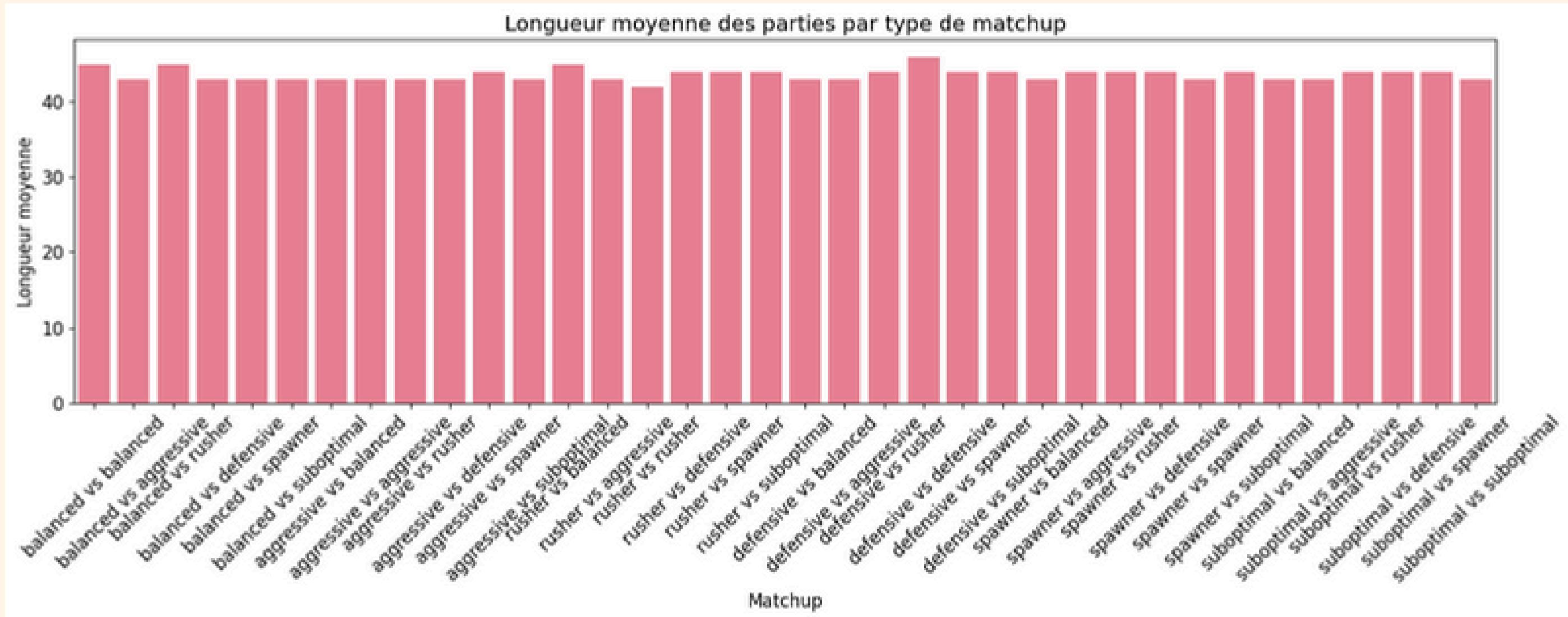
- **Les écarts de performance se réduisent avec plus de joueurs**
- **La complexité des interactions à 4 joueurs tend à équilibrer les stratégies**

# SATISFACTION DES AGENTS



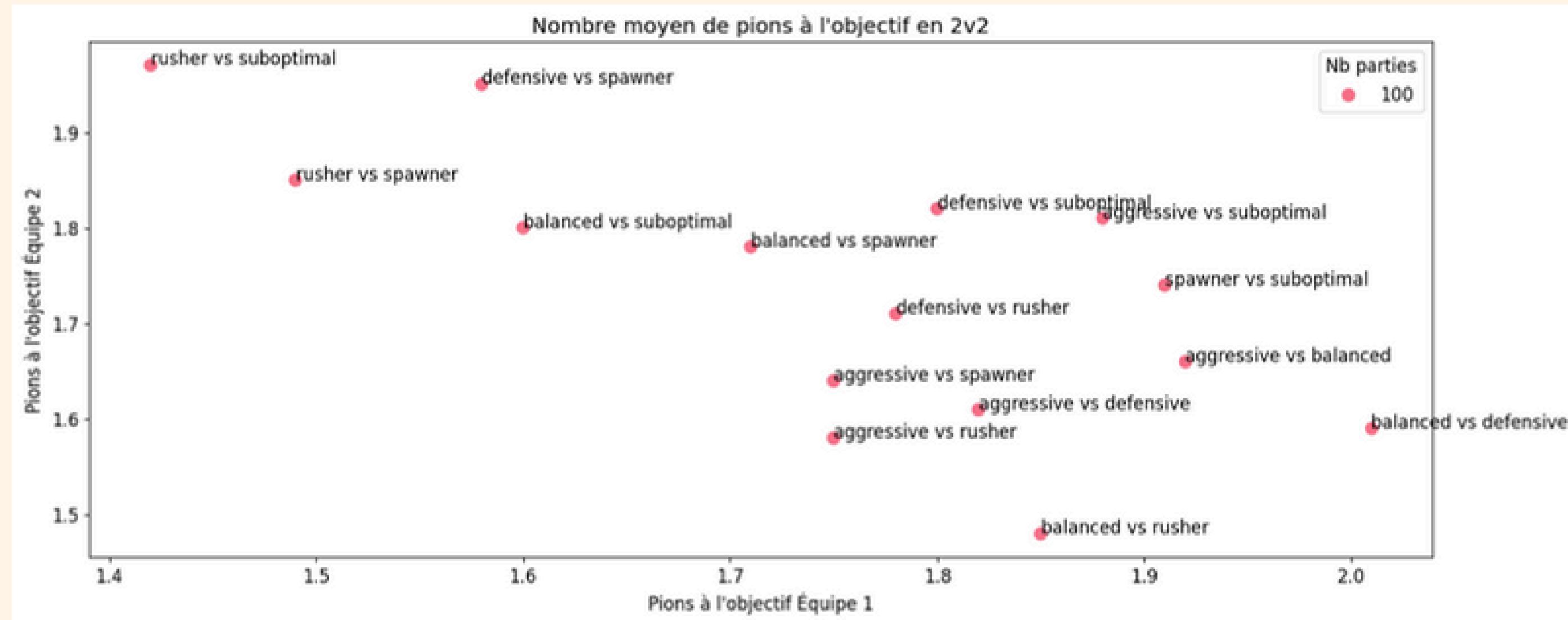
- **Performance ≠ Satisfaction**
- **Meilleur taux de kills ne correspond pas aux meilleures configurations pour gagner**
- **Un agent peut être satisfait sans être le plus efficace**

# MÉTRIQUES DE JOUABILITÉ



- Durée des parties stable quelle que soit la combinaison d'agents
- Suggère un cadre de jeu équilibré en apparence

# MÉTRIQUES DE JOUABILITÉ



- **Disparités importantes dans le nombre de pions atteignant l'objectif**
- **Certaines combinaisons d'agents nettement plus efficaces**
- **Déséquilibres cachés malgré une durée de partie stable**

# LE LABYRINTHE

- Choix initial ?
  - Richesse stratégique
  - Complexité modérée



## Blocages techniques :

- Complexité de l'espace d'actions
- Dynamisme de l'environnement
- Actions interdites

## Tentatives de résolution :

- Réduction de la complexité
- Masquage des actions interdites
- Pénalisation des erreurs

# LEÇONS TIRÉES

- Compréhension des défis liés aux environnements complexes
- Importance du choix des outils et algorithmes
- Visualisation et suivi des performances



# RÉALISATIONS



**CHARLOTTE**

## 1. Labyrinthe

- **Environnement Gymnasium**
- **Entrainement des agents**
- **Recherches sur les blocages**

## 2. Petits Chevaux

- **Gestion de la base de données**
- **Automatisation des parties**
- **Enregistrement automatique des données**
- **Analyse de l'entraînement**

# RÉALISATIONS



ZOÉ

## 1. Labyrinthe

- Amélioration du code existant et de l'interface.
- Mise en place BD.

## 2. Petits Chevaux

- Modélisation du jeu et variantes.
- Mise en place de l'environnement.
- Tests automatiques.

# RÉALISATIONS



**EKATERINA**

## 1. Labyrinthe

- **Ajustement du code existant.**
- **Création d'un thème "océan" et amélioration de l'interface.**
- **Recherche sur les blocages.**

## 2. Petits Chevaux

- **Développement d'une interface utilisateur et d'un système de parties configurables.**
- **Création de supports visuels pour illustrer le jeu.**

# RÉALISATIONS



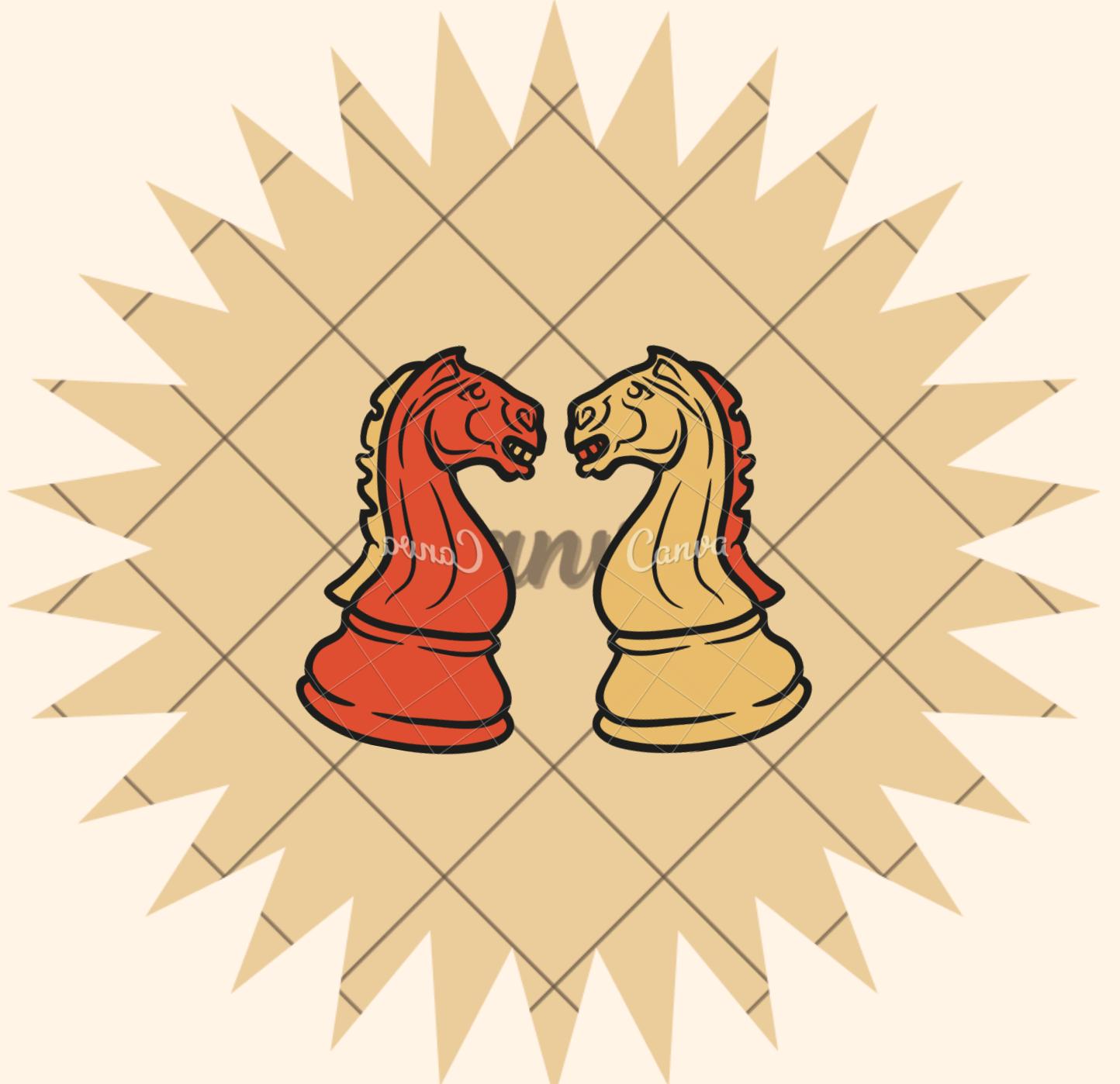
## 1. Labyrinthe

- Exploration des algorithmes d'apprentissage.
- Développement initial des agents RL.
- Tentatives d'optimisation des performances.

## 2. Petits Chevaux

- Crédit des profils d'agents.
- Implémentation des matchups entre agents.
- Analyse statistique des performances agents-règles.

# **CONCLUSION OBJECTIFS ATTEINTS**

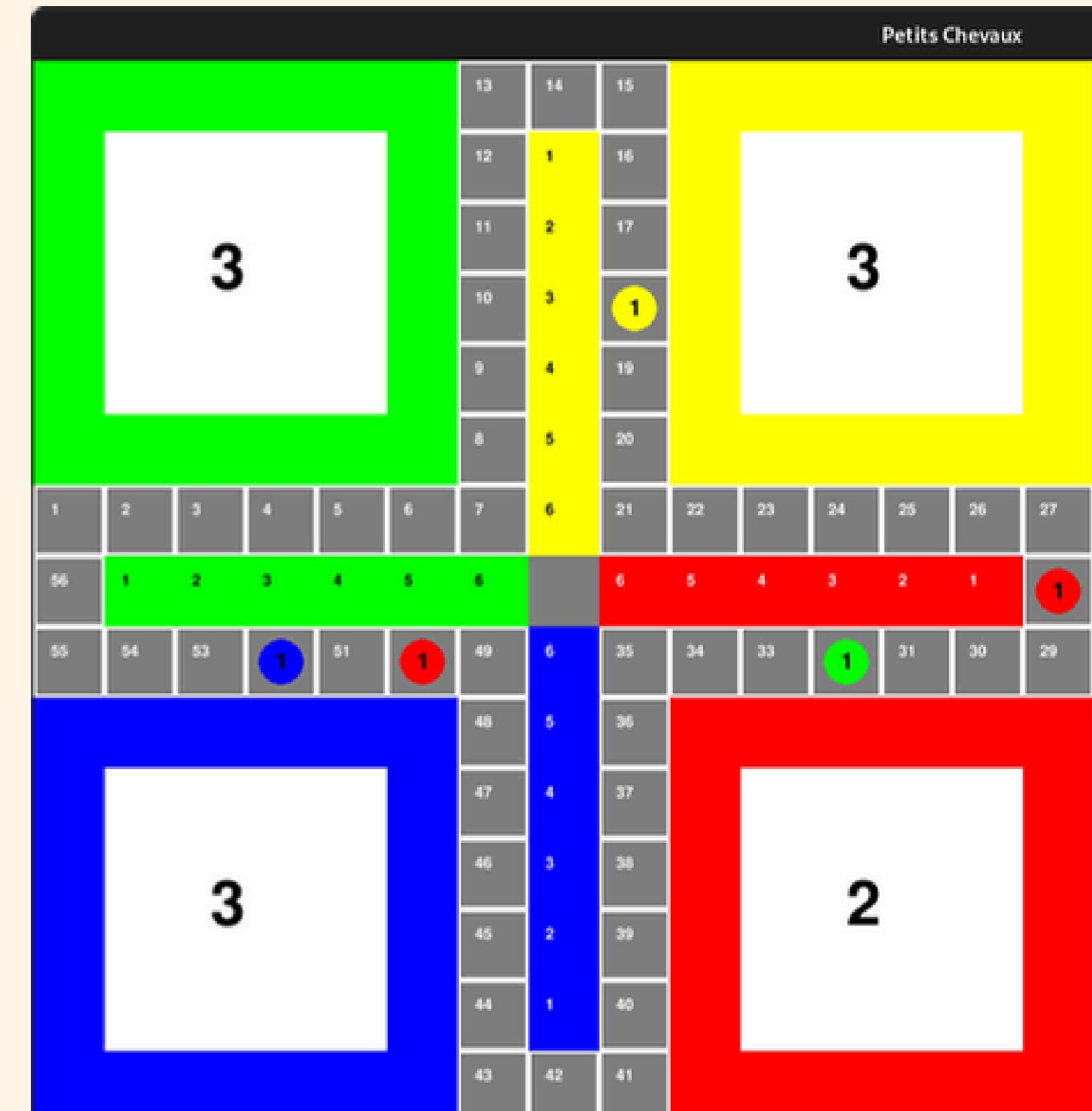


- **Développement d'agents RL avec différentes stratégies adaptées aux variantes du jeu**
- **Interface jouable contre agents entraînés**
- **Analyse statistique approfondie des performances**

# **CONCLUSION**

## **APPLICATIONS PRATIQUES**

- **Outil d'analyse pour concepteurs de jeux**
  - **Test rapide de configurations de règles**
  - **Simulation de différents styles de jeu**



MERCI DE VOTRE ATTENTION

**DES QUESTIONS ?**

