

Étude de quelques algorithmes de joueurs artificiels participants à des jeux de stratégie en temps réel

Dimitri COCHERIL-CRÈVECŒUR
13960

26 mai 2024

Plan

1. Présentation du problème
2. Moteur
3. Stratégies testées

StarCraft

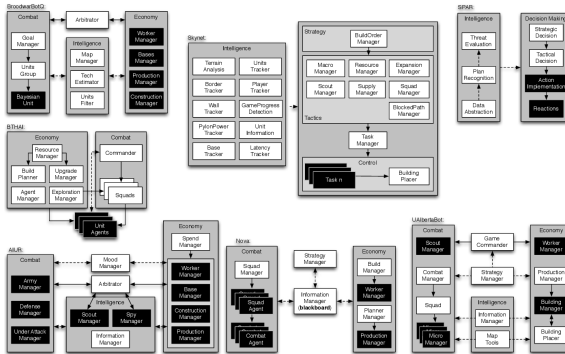
Jeu de stratégie en temps réel :

- ▶ Jeux simultanés
- ▶ Deux joueurs s'affrontent pour le contrôle d'une carte
- ▶ Gestion de ressources : minage, création d'unités, de bases (stratégie)
- ▶ Combat entre unités (tactique)



Modèles existants

- ▶ IA de Google : *AlphaStar*
Apprentissage supervisé puis par renforcement
- ▶ Robots ("bots") :



StarCraft

On se concentre ici sur la partie "combat" :

Deux joueurs disposent d'unités pouvant bouger et attaquer celles adverses

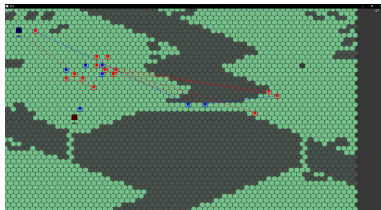
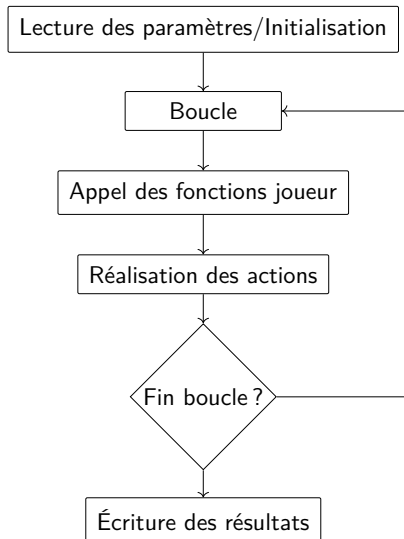
Chaque joueur doit tuer toutes les unités ennemies



Problème

1. Facteur de branchement entre 10^{50} et 10^{200}
2. Durée typique d'une partie : 25 minutes donc
 $25 \times 60 \times 24 = 36000$ états

Développement moteur du jeu



```
void game_class::play() const
{
    for (const auto p : players_)
        /*Pour chaque joueur...*/
        {
            p->moves_get(this, state_);
            /*...on choisit les actions...*/
        }
    state_->moves_make(map_);
    /*...et on effectue les actions...*/
}
```


Stratégie naïve : attaque par puissance

Stratégie naïve : attaquer par distance

Utilisation MCTS

Rassemblement des unités : DBSCAN