

Claim



Dossier de validation IA

Index

1. Introduction

- a. Contexte du projet.....2
- b. Contexte du jeu.....2
- c. Description du jeu Claim.....3

2. Méthodologie de validation IA

- a. Choix des algorithmes.....6

3. Résultat de la validation IA

- a. Performance de l'IA.....7
- b. Comparaison avec les différentes approches.....
- c. Limitations et pistes d'amélioration.....

4. Conclusion

- a. Interprétation des résultats
- b. Réflexions sur les défis rencontrés

1. Introduction

a. Contexte du projet

Notre équipe est composée de 5 membres dynamiques et passionnés, réunis dans le cadre du projet fin de semestre 6 de la licence informatique générale à l'Université de Grenoble Alpes.

Les membres qui ont principalement contribué à l'ia du jeu Claim sont:

- Yosri Oueslati, avec sa solide compréhension des algorithmes d'apprentissage automatique, a dirigé les efforts de conception et d'implémentation de l'intelligence artificielle du jeu.
- Mohamed Amine El Ouechrin, douée pour la communication et la gestion de projet, a coordonné les efforts de l'équipe, assurant une collaboration efficace et une progression harmonieuse du projet.
- Yassine Madhbouh: expert en développement logiciel, a supervisé l'intégration fluide de l'IA dans l'interface utilisateur du jeu, garantissant une expérience utilisateur transparente.

b. Contexte du Jeu.

Le jeu se déroule dans un cadre médiéval où le Roi est mort dans des circonstances mystérieuses. Personne ne sait exactement ce qui s'est passé, mais il a été retrouvé face contre terre dans un tonneau de vin. Maintenant, c'est aux cinq factions du royaume de décider qui sera le nouveau roi.

c. Description du jeu Claim

Mise en place

Pour commencer, mélangez toutes les cartes et placez-les en une pioche face cachée au centre de la table. Distribuez 13 cartes à chaque joueur, que chacun prend en main sans les montrer à l'autre.

Aperçu du jeu

Le jeu se déroule en deux phases distinctes. Dans la première phase, chaque joueur reçoit une main de cartes qu'il utilise pour recruter des partisans. Dans la deuxième phase, ils utilisent les partisans gagnés dans la première phase pour rivaliser et obtenir le soutien des cinq factions du royaume. À la fin du jeu, le joueur qui a la majorité des partisans dans une faction gagne le vote de cette faction. Celui qui remporte le vote d'au moins trois factions gagne la partie.

Phase I : Recruter des partisans

Cette phase consiste en treize plis, un pour chaque carte dans la main de chaque joueur. Le leader pour chaque pli est le joueur qui a remporté le dernier pli (ou dans le cas du premier pli, le plus jeune joueur).

Chaque pli se joue en trois étapes :

Révéler une carte

Retournez la carte du dessus de la pioche au centre de la table et placez-la entre les deux joueurs. Les joueurs vont se disputer cette carte.

Jouer des cartes

1. Le leader joue une carte de sa main. Note : Il peut choisir une carte de n'importe quelle faction, pas nécessairement de la faction pour laquelle les joueurs se disputent.
2. L'autre joueur joue une carte de sa main. Important : Si possible, l'autre joueur doit suivre la faction. C'est-à-dire, s'il a une carte de la faction de la carte jouée par le leader, il doit la jouer (exception : Doppelgänger). Un joueur ne peut jouer une carte d'une autre faction que s'il n'a pas de carte correspondant à la faction jouée par le leader.

Collecter les cartes

1. Le joueur ayant joué la carte de plus grande valeur (0 étant la plus basse, 9 la plus haute) de la faction jouée par le leader remporte la carte au centre. En cas d'égalité, le leader remporte le pli. Note : Si le second joueur a joué une carte d'une autre faction, le leader remporte automatiquement le pli, sauf si un pouvoir de faction (Doppelgänger ou Chevalier) est en jeu.
2. Le gagnant place sa carte remportée face cachée dans son deck de Partisans. Le perdant prend une carte du dessus de la pioche au centre et la place face cachée dans son deck de Partisans. Ce joueur peut regarder la carte, mais ne peut pas la montrer à l'autre joueur. Note : Les cartes dans votre deck de Partisans seront votre main pour la Phase Deux.
3. Si un joueur a joué une carte Mort-vivant, le gagnant du pli les prend et les place face visible dans sa pile de Score. Important : Une carte Mort-vivant gagnée du centre de la table va toujours dans la pile de Partisans du gagnant.
4. Défaussez toutes les cartes jouées restantes. Important : Assurez-vous de garder votre pile de Score face visible et votre deck de Partisans face cachée pour les garder séparés.

Continuez jusqu'à ce que la pioche soit épuisée et que les joueurs n'aient plus de cartes en main. Le jeu passe alors à la Phase Deux.

Phase II: Rassembler des partisans

Les deux joueurs prennent les 13 cartes de Partisans qu'ils ont gagnées pendant la Phase I en main. Maintenant, les joueurs vont jouer 13 plis supplémentaires. Cependant, au lieu de se disputer des cartes particulières au centre, cette fois, les joueurs se disputent les deux cartes jouées à chaque pli.

Chaque pli se joue en deux étapes :

Jouer des cartes

1. Le leader joue une carte de sa main.
2. L'autre joueur joue une carte de sa main. Important : Les règles concernant le suivi de la faction sont toujours en vigueur !

Collecter les cartes

1. Déterminez le gagnant comme dans la Phase Un.
2. Le gagnant du pli place les deux cartes jouées face visible dans sa pile de Score, sauf si un pouvoir de faction (Nains) est en jeu.

Après que tous les joueurs aient joué toutes les cartes de leur main, toutes les factions sont comptabilisées.

Fin du jeu et scoring

Les joueurs comptent combien de cartes de chaque faction ils ont dans leur pile de Score. Celui qui a le plus de cartes d'une faction remporte le vote de cette faction. En cas d'égalité, celui qui a la carte de valeur la plus élevée de cette faction remporte le vote. Le joueur qui remporte le vote d'au moins trois factions gagne la partie.

Pouvoirs spéciaux des factions

Chaque faction a un pouvoir spécial qui affecte le jeu. Ils sont les suivants :

Gobelins

Pas de pouvoir spécial.

Chevaliers

Lorsqu'un Chevalier est joué après un Gobelin, il bat automatiquement le Gobelin, quelle que soit sa valeur. Important : Le joueur doit toujours suivre la faction, si possible.

Morts-vivants

Les cartes Mort-vivants jouées ne sont pas défaussées pendant la Phase Un comme les cartes des autres factions, mais sont ajoutées à la pile de Score du gagnant du pli.

Nains

Dans la Phase Deux, le joueur perdant un pli collecte tous les Nains joués pendant ce pli et les ajoute à sa pile de Score. Le gagnant récupère toujours les autres cartes non-Nains jouées.

Doppelgänger

Cette faction est considérée comme joker. Vous pouvez jouer un Doppelgänger à la place de la faction demandée, même si vous êtes capable de suivre la faction. Lorsqu'il est joué en second, il est considéré comme appartenant à la même faction que la première carte jouée et est considéré comme suivant la faction. Note : Si le leader joue un Doppelgänger, l'autre joueur doit toujours suivre la faction en jouant un Doppelgänger, si possible. Important : Un Doppelgänger ne prend aucun pouvoir spécial de la faction qu'il suit. Par exemple, s'il est joué après un Mort-vivant en Phase Un, il ne sera pas ajouté à la pile de Score du gagnant, ni ne sera pris par le perdant en Phase Deux comme le ferait un Nain.

2. Methodologie IA

a. Choix des algorithmes

Pour la conception de l'intelligence artificielle dans le jeu Claim, nous avons développé trois niveaux de difficulté pour simuler différents niveaux de compétence de l'adversaire. Chaque niveau utilise une stratégie distincte pour jouer les cartes, allant d'une approche simple et aléatoire à une méthode avancée utilisant l'algorithme Minimax. Ces niveaux de difficulté permettent d'offrir aux joueurs une expérience de jeu variée et adaptée à leur niveau de compétence. Voici un aperçu de chacune des IA avant de plonger dans les détails spécifiques de leur fonctionnement :

- **IA FACILE:**

L'IA de niveau facile dans le jeu Claim joue ses cartes de manière complètement aléatoire. Ce niveau d'intelligence artificielle ne suit aucune stratégie spécifique et ne tient pas compte des cartes jouées par l'adversaire ou de la situation actuelle du jeu. L'algorithme derrière cette IA sélectionne simplement une carte au hasard parmi celles disponibles dans sa main pour chaque tour.

Cette approche présente plusieurs avantages :

Simplicité de mise en œuvre : Facile à programmer, elle ne nécessite pas de calculs complexes.

Expérience de jeu détendue : Idéale pour les nouveaux joueurs, elle offre un adversaire prévisible et non intimidant, facilitant l'apprentissage des règles.

Imprévisibilité : L'aléatoire peut introduire des éléments de surprise, rendant le jeu parfois plus intéressant.

Encouragement à l'amélioration : Permet aux débutants d'expérimenter des stratégies sans pression, favorisant une progression douce.

Adaptabilité : Peut être ajustée pour introduire de légères variations, maintenant ainsi l'intérêt des joueurs.

- **IA INTERMÉDIAIRE:**

L'IA de niveau intermédiaire dans le jeu Claim adopte une stratégie plus réfléchie en essayant de toujours jouer la meilleure carte disponible pour remporter chaque pli. Cette approche vise à maximiser les chances de l'IA de gagner les plis, tout en équilibrant les situations où la carte à gagner n'est pas toujours la plus avantageuse. Voici les avantages et les détails de cette stratégie :

Stratégie de maximisation : L'IA tente de jouer la meilleure carte disponible à chaque tour pour remporter le pli, cherchant ainsi à maximiser ses gains à chaque étape du jeu.

Équilibrage des cartes gagnées : Comme les cartes à gagner peuvent varier en valeur et en utilité, cette approche assure que l'IA ne gagne pas toujours les cartes les plus précieuses, équilibrant ainsi les chances entre les joueurs.

Complexité intermédiaire : Cette IA offre un défi plus grand que l'IA facile, en utilisant une logique simple mais efficace pour évaluer et jouer les cartes, sans atteindre la complexité de l'IA difficile.

Amélioration de la compétitivité : Les joueurs doivent ajuster leurs stratégies et anticiper les mouvements de l'IA, rendant le jeu plus dynamique et engageant.

Introduction aux stratégies avancées : Cette IA sert de pont entre une approche aléatoire et des stratégies plus avancées, préparant les joueurs à affronter des adversaires plus compétents.

- **IA DIFFICILE:**

Exécution de l'algorithme

L'algorithme fonctionne en parcourant récursivement toutes les configurations possibles jusqu'à atteindre une profondeur où le jeu est terminé ou une condition d'élagage. En sauvegardant et restaurant l'état du plateau, l'algorithme évite la création de nombreux objets intermédiaires, ce qui économise de la mémoire et améliore les performances.

Minimax

Pour notre IA, nous avons implémenté l'algorithme Minimax. Cet algorithme simule une partie entre deux joueurs en jouant les meilleurs coups pour l'IA, tandis que l'adversaire joue les pires coups pour contrer l'IA. À chaque tour, l'algorithme énumère tous les coups possibles et parcourt chaque branche afin d'explorer celle qui offre la meilleure chance de victoire.

L'algorithme explore toutes les branches, où chaque branche représente une partie avec une configuration différente (l'ordre des cartes jouées par les joueurs est différent). Il se base sur la simulation d'un arbre de toutes les configurations possibles à partir d'un état donné du plateau passé en paramètre. Chaque nœud représente une nouvelle configuration du plateau après qu'un joueur a joué, ce qui crée un arbre de profondeur 26 (chaque joueur ayant 13 cartes). Cette multitude de configurations à explorer rend l'algorithme très gourmand en temps d'exécution, ce qui pose une limitation de puissance de calcul.

Élagage Alpha-Bêta

Pour remédier à ce problème, nous avons utilisé l'élagage alpha-bêta, ce qui a permis d'optimiser l'algorithme et de le rendre beaucoup plus efficace. Grâce à cette technique, certaines branches de l'arbre de jeu sont coupées (élaguées) lorsqu'il est certain qu'elles ne conduiront pas à une meilleure décision que les branches déjà explorées, réduisant ainsi considérablement le nombre de configurations à examiner et améliorant la performance globale de l'algorithme.

Implémentation

L'implémentation détaillée de cet algorithme commence par le cas de base : si le jeu est terminé, la fonction retourne la valeur d'évaluation de l'état du jeu. Avant de simuler les coups, l'état actuel du plateau est sauvegardé. Ensuite, les coups jouables sont générés : si c'est le tour de l'IA, les coups jouables pour l'IA sont générés, et si c'est le tour de l'adversaire, les coups jouables pour l'adversaire sont générés. Pour chaque carte jouable, le plateau est modifié en jouant cette carte. Si nécessaire, le gagnant du tour est déterminé et les cartes sont attribuées en conséquence. Après avoir simulé un

coup, un appel récursif est fait sur le nouvel état du jeu. Après chaque appel récursif, l'état précédent du plateau est restauré. Si c'est le tour de l'IA, Alpha est mis à jour et la meilleure évaluation obtenue est gardée. Si c'est le tour de l'adversaire, Beta est mis à jour et la pire évaluation obtenue est gardée. Si Beta est inférieur ou égal à Alpha, la boucle des coups jouables est interrompue (élagage). L'algorithme Minimax fonctionne en parcourant récursivement toutes les configurations possibles jusqu'à atteindre une profondeur où le jeu est terminé ou une condition d'élagage est rencontrée.

Fonction d'évaluation

La fonction d'évaluation joue un rôle crucial dans la performance de l'algorithme Minimax. Elle permet d'évaluer l'état du plateau à un instant donné pour aider l'algorithme à faire les meilleurs choix possibles. Plus cette fonction est précise, plus l'IA sera performante dans ses décisions.

Nous avons testé deux fonctions d'évaluation différentes :

1. Première fonction d'évaluation : La première fonction renvoyait simplement le score à la fin de la simulation, indiquant si une branche mène à la victoire ou à la défaite en retournant 1 ou 0. Cependant, cette approche s'est révélée insuffisante. Bien qu'elle identifie les branches menant à la victoire, elle ne permet pas de choisir la meilleure branche parmi celles-ci. En effet, il peut y avoir plusieurs branches qui mènent à la victoire, mais elles ne sont pas toutes équivalentes en termes de chances de succès. Par exemple, deux branches peuvent toutes deux aboutir à une victoire, mais l'une peut offrir une probabilité de victoire plus élevée même si l'adversaire change de stratégie. Cette branche serait plus robuste aux mouvements imprévus de l'adversaire, tandis que l'autre pourrait voir ses chances de succès diminuer fortement si l'adversaire joue une carte avantageuse.

Pour améliorer la précision et la robustesse de notre IA, nous devons donc élaborer une fonction d'évaluation plus sophistiquée qui prend en compte non seulement la victoire ou la défaite, mais aussi la probabilité de succès dans différents scénarios. Cette approche permet de sélectionner non seulement les branches gagnantes, mais aussi celles qui maximisent les chances de victoire même face à des mouvements adverses imprévus.

Tests et Statistiques

Pour vérifier le bon fonctionnement de notre implémentation de l'algorithme Minimax, nous avons conçu une série de tests où l'IA joue contre des adversaires de différents niveaux de difficulté. Ces tests avaient pour but de déterminer l'efficacité et la robustesse de notre IA.

Tests de la deuxième phase uniquement

Nous avons d'abord testé notre IA en simulant uniquement la deuxième phase du jeu :

- **Minimax vs. IA Facile**

```
Score MinMax : 955  
Score Facile : 45
```

```
Score MinMax : 941  
Score Facile : 59
```

- **Minimax vs. IA Intermédiaire**

```
Score MinMax : 967  
Score intermediaire : 33
```

```
Score MinMax : 964  
Score intermediaire : 36
```

- **IA Intermédiaire vs. IA Facile**

```
Score intermediaire : 7577  
Score Facile : 2423
```

```
Score intermediaire : 7580  
Score Facile : 2420
```

Ces tests nous ont permis de vérifier si l'algorithme Minimax pouvait prendre de bonnes décisions sans l'influence de la première phase du jeu.

Tests d'une partie complète

Ensuite, nous avons intégré la première phase dans nos tests pour évaluer l'IA dans des conditions de jeu complètes :

- **Minimax vs. IA Facile**

```
Score MinMax : 810  
Score Facile : 190
```

```
Score MinMax : 817  
Score Facile : 183
```

- **Minimax vs. IA Intermédiaire**

```
Score MinMax : 751  
Score intermediaire : 249
```

```
Score MinMax : 743  
Score intermediaire : 257
```

- **IA Intermédiaire vs. IA Facile**

```
Score intermediaire : 765  
Score Facile : 235
```

```
Score intermediaire : 748  
Score Facile : 252
```

Les résultats des tests ont montré que l'IA utilisant l'algorithme Minimax était nettement plus performante que l'IA Facile et légèrement supérieure à l'IA Intermédiaire. En particulier, l'IA Minimax a remporté la majorité des parties lorsqu'elle a joué contre l'IA Facile et a montré une forte résilience même face aux stratégies imprévues de l'IA Intermédiaire.

Tests de vérification de l'algorithme Minimax

Pour nous assurer que l'algorithme Minimax fonctionne correctement, nous avons mené des tests supplémentaires en utilisant des configurations prédéfinies. Nous avons appliqué l'algorithme à des configurations gagnantes à 100 %, des configurations perdantes à 100 %, et vérifié s'il renvoyait les résultats corrects. De plus, nous avons testé l'algorithme avec des configurations étudiées, en jouant contre l'IA pour observer si elle s'adaptait correctement aux choix de cartes imprévus de l'adversaire.

Ces tests ont confirmé que l'algorithme Minimax était capable d'identifier correctement les configurations gagnantes et perdantes, et qu'il s'adaptait bien aux stratégies adverses, assurant des choix optimaux même face à des mouvements inattendus.

Conclusion

La phase de recrutement est une étape cruciale dans le processus global de notre algorithme de jeu de cartes. C'est durant cette phase que les cartes sont sélectionnées et distribuées aux joueurs, déterminant ainsi leurs chances initiales de victoire. Une bonne stratégie de recrutement peut grandement influencer le déroulement du jeu en permettant de constituer une main équilibrée et puissante.

L'algorithme Minimax, bien qu'extrêmement efficace pour optimiser les décisions à chaque tour de jeu, repose sur la qualité des cartes recrutées en phase initiale. Une main bien constituée offre non seulement des options stratégiques diversifiées mais également une meilleure capacité d'adaptation face aux mouvements imprévus de l'adversaire. Ainsi, le succès de l'IA ne dépend pas uniquement de sa capacité à jouer de manière optimale, mais aussi de la qualité des cartes obtenues dès le début du jeu.

Nos tests ont montré que l'IA utilisant l'algorithme Minimax pouvait surmonter de nombreux défis stratégiques, mais la phase de recrutement reste essentielle pour maximiser ses chances de succès. Une bonne combinaison de cartes permet à l'IA de mieux répondre aux stratégies adverses, d'exploiter les faiblesses de l'opposant, et de maintenir une position avantageuse tout au long du jeu.

En somme, pour atteindre une performance optimale, il est crucial de porter une attention particulière à la phase de recrutement. C'est cette étape qui, en grande partie, détermine la chance de gagner et permet à l'algorithme Minimax de pleinement déployer son potentiel stratégique durant la partie.

