

Documentation experimentation

Ce projet contient quatre IA permettant de jouer à Pandémie avec des performances variées.

Evaluation de l'IA

Nous avons utilisé une classe TestGame pour itérer 100 parties avec une IA afin d'évaluer ses performances sur les critères suivants :

- Nombre de Win et Defeats
- Nombre de maladies traitées
- Nombre de défaites par manque de carte joueur
- Nombre de défaites par manque de cube maladie
- Nombre de défaites par outbreaks
- Nombre de défaites par erreur dans la partie

Ainsi, nous avons pu tester l'impact de chaque évolution des IA présentées ci-dessous en tentant de minimiser le nombre de defeats (donc minimiser chaque cause possible de défaite).

Ces tests sont faits en utilisant les paramètres par défaut du game engine.

StupidAi

Cette IA initialement fournie tente de réaliser quatre fois l'action DRIVE vers son premier voisin à chaque tour. Lorsqu'elle doit se défausser de cartes, l'IA choisit de jeter les premières cartes de sa main. Naturellement, elle ne gagne jamais une partie.

Taux de réussite (paramètres par défaut) : 0%

Statistiques sur 100 parties par défaut :

- 0 cured diseases
- win :0 lose:100
- outbreak : 55
- no block : 45
- no card : 0
- no error : 0

NotSoStupidAi

Cette IA, toujours basique, va réaliser systématiquement une séquence de tentative d'application d'une d'action, par ordre d'importance, afin de consommer ses 4 points.

Elle tente d'appliquer chaque action jusqu'à épuisement de ses points ou *UnauthorizedActionException*, qui la mènera à l'application de l'action d'importance moindre suivante.

Cette IA tente les actions dans l'ordre suivant :

1. Découvrir le vaccin d'une maladie encore non découverte.
2. Traiter un maximum de maladies sur sa localisation (jusqu'à épuisement de ses points d'action ou impossibilité).
3. Se déplacer vers un de ses voisins, choisi aléatoirement (jusqu'à épuisement de ses points d'actions).

Si elle doit se défausser de cartes, l'IA se délaïsse des cartes de la couleur la moins présente dans sa main. Cette dernière règle est encore trop aléatoire pour lui permettre de découvrir un vaccin assez régulièrement pour gagner une partie.

Taux de réussite (paramètres par défaut) : 0%

Statistiques sur 100 parties par défaut :

- 11 cured diseases
- outbreak : 71
- no block : 29
- no card : 0
- no error : 0

NotSoStupidAi2

Cette IA est la première à terminer une partie victorieuse. Il s'agit d'une amélioration des règles de la NotSoStupidIA, notamment sur :

- son choix de cartes à défausser lorsque cette situation critique se présente.

Elle se délester de carte concernant une maladie dont le vaccin est déjà découvert et si ce n'est pas possible, de cartes dont la maladie est la moins fréquente dans sa main.

- Son choix de maladie à traiter (sur sa localisation ou ses voisins grâce à un déplacement *DRIVE* préalable)

Cette IA introduit la notion de niveau d'infection d'une ville, afin de prioriser le traitement d'une maladie déjà fortement présente dans une ville accessible et de diminuer le nombre de défaites par outbreak.

- Sa gestion des actions possibles avant de les appliquer

Afin d'éviter de lever des Exception (qui doivent rester exceptionnelles), cette IA teste si les conditions de réalisation d'une action sont réunies. Elle applique ensuite les actions possibles selon leur ordre d'importance.

Taux de réussite (paramètres par défaut) : 2%

Statistiques sur 100 parties par défaut :

- 140 cured diseases
- outbreak : 86
- no block : 12
- no card : 0
- no error : 0

DynamicAI

Cette IA est la plus avancée que nous avons produite. Elle réalise la séquence suivante jusqu'à épuisement de ses 4 points d'actions pour jouer son tour :

1. Si possible, découvrir autant de vaccins que sa main le permet
2. Sinon:
 - a. établir la liste des ses actions autorisées
 - b. Reproduire l'état actuel de la partie dans un FakeBoard (grâce à toutes les informations disponible via l'interface du GameEngine) et appliquer chacune de ses actions possibles.
 - c. Evaluer l'état chaque état intermédiaire ainsi simulé de chaque FakeBoard
 - d. Appliquer la meilleur action

La complexité de cette IA réside dans la fonction d'évaluation (*eval()*) qui combine :

- Le nombre de dangers dans l'état intermédiaire d'un FakeBoard, comparé au nombre de dangers du GameEngine initial. Ces dangers sont estimés selon le nombre de maladies dispersés dans chaque ville (pondéré progressivement avec le nombre de cubes d'une même maladie sur une même ville, c'est-à-dire le risque d'outbreak).
- Les opportunités offertes par la position du player pour traiter sa propre ville ou se déplacer vers des villes à traiter (également pondérées).
- Le nombre de cubes maladie restants.

Taux de réussite (paramètres par défaut) : 37%

Statistiques sur 100 parties par défaut :

- outbreak : 32
- no block : 16
- no card : 15
- no error : 0

Améliorations possibles

Une IA plus avancée pourrait prendre en compte :

- Les probabilités d'apparition des cartes infecteurs (combinées aux probabilités d'apparition des cartes Epidémie), dont la défausse est observable. Ceci pourrait permettre de prioriser différemment les maladies à traiter sur des villes spécifiques.
- Le déroulement de davantage d'actions en simulant 4 actions au sein d'un FakeGame au lieu d'une seule. Ainsi, le tour entier de jeu serait simulé et évalué.
- Le déroulement de davantage d'états intermédiaires (Minimax).

Compte tenu de la faible complexité de nos IA, nous n'avons pas eu à nous soucier du temps de calcul (< 1 seconde). En revanche, dans l'implémentation des deux dernières améliorations présentées ci-dessus, nous aurions probablement à utiliser le multi-threading pour rester sous ce seuil.