



Rapport du TP d'optimisation

APPLICATION DE L'ALGORITHME DE MONTE-CARLO

Théophile Carrasco | US332N | Avril 2024

Objectifs du TP

Ce TP s'est déroulé du mercredi 17 avril au vendredi 19 avril sous la supervision du professeur Matthieu MONTES. Il avait pour but plusieurs objectifs :

- Découvrir et utiliser l'Algorithme de Monte-Carlo dans le domaine du jeu vidéo ;
- Développer cet algorithme dans le but d'optimiser un deck de cartes pour un jeu de cartes à collectionner (JCC) type Magic the Gathering ou Hearthstone ;
- Produire des données afin de pouvoir analyser les résultats obtenus ;

Cela s'est fait en plusieurs temps, tout d'abords il a fallu développer le jeu de cartes et ses règles et permettre à deux bots informatiques de pouvoir faire une partie. Ensuite il a fallu permettre aux bots de pouvoir effectuer un grand nombre de parties en permettant de modifier son deck entre chaque série de parties afin de pouvoir s'améliorer et ainsi d'optimiser son deck. Et finalement, étendre les règles du jeu afin d'en augmenter sa complexité.

REGLES DU JEU INITIALES

Le jeu propose un affrontement entre 2 joueurs en utilisant des cartes. Chacun commence avec son deck de 30 cartes. Le jeu est une succession de tours de jeu dont le but est de réduire les points de vie son adversaire à 0.

Au début de la partie, chaque commence avec :

- 20 points de vie | HP | ;
- 0 points de mana ;
- 4 cartes qu'il pioche de son deck dans sa main

La zone de jeu de chaque joueur se compose de plusieurs parties :

- La pioche qui correspond au deck du joueur ;
- La main qui contient les cartes que peut jouer le joueur ;
- Le plateau | Board | où le joueur pose les cartes qu'il joue

Une carte possède plusieurs caractéristiques telles que :

- L'Attaque | ATK | : les dégâts qu'elle inflige ;
- La Défense | DEF | : les points de vie de la carte ;
- Un Coût | Cost | : ce que doit dépenser le joueur pour pouvoir poser la carte sur son côté. Sa formule est donnée par la formule suivante : $\text{Cost} = \frac{\text{ATK} + \text{DEF}}{2}$;

Un tour de jeu se déroule comme ceci :

1. Le joueur regagne ses points de mana du tour précédent plus un ;
2. Le joueur pioche une carte ;
3. Tant qu'il possède du mana, il peut payer le coût d'une carte et la poser sur le Board ;
4. Pour chaque carte sur son Board, il peut attaquer au choix l'adversaire ou un de ses cartes en jeu.

Optimisation du code

Afin de produire des résultats rapidement, le code a été développé avec la rapidité comme optimum en tête. Ainsi, quand cela est possible, nous réduisons le nombre boucle et de vérification. Il a également été ajouté du multithreading afin de gagner en vitesse. Pour exemple, dans la dernière version du projet pour 2 millions de parties jouées, cela prend entre 6 et 8s.

5.

LES DONNEES RECOLTEES

Les données récoltées à l'issu de chaque simulation :

- Le taux de victoire du deck ;
- Le nombre moyen de tours par partie ;
- La répartition du coût des cartes au début et à la fin de la simulation ;
- Le deck de départ et le deck finalisé ;

Partie 1 : « Mon thé, Carlo ! »

DESCRIPTION

Dans cette première partie, le but était de mettre en place les règles initiales du jeu et de mettre en place un premier algorithme de Monte-Carlo pour optimiser le deck du joueur 1 contre le deck du joueur 2 en changeant une carte depuis la liste exhaustive de cartes.

PROTOCOLE

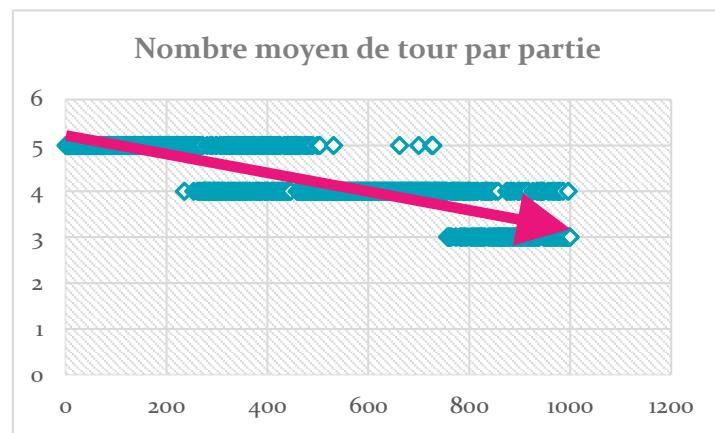
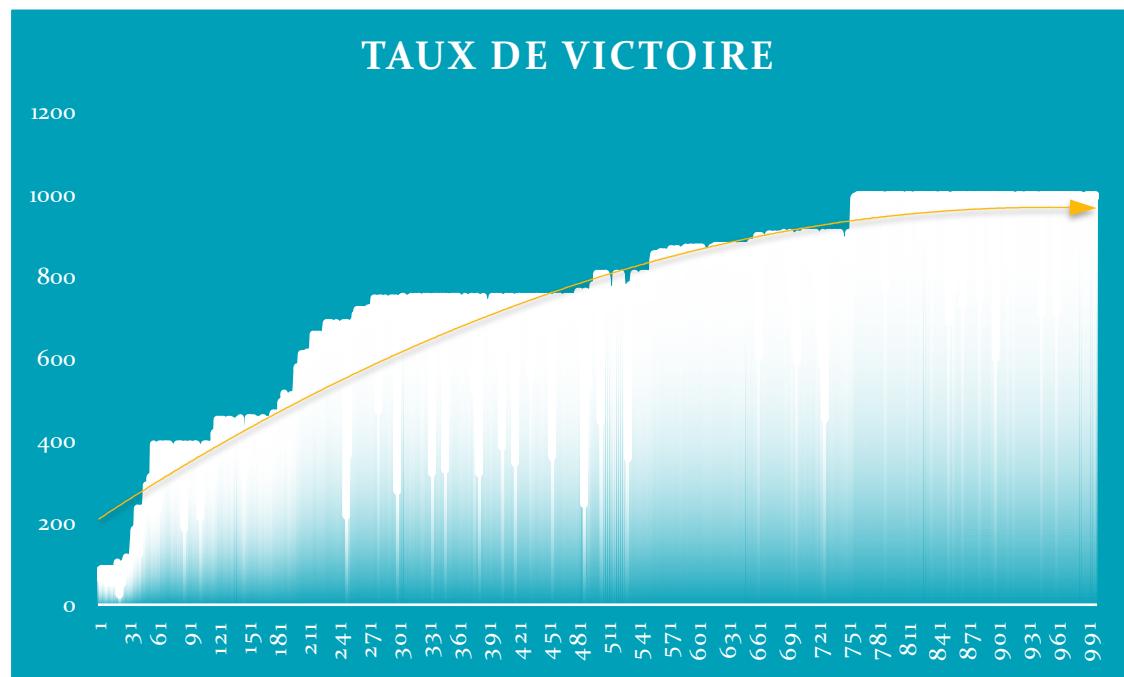
Le protocole consiste à faire 1000 itérations d'affrontement, une itération d'affrontement consiste à changer une carte aléatoirement, puis à faire 1000 parties et de noter le taux de victoire à chaque partie.

Si le taux de victoire à l'itération ‘t’ est supérieur à celui ‘t-1’ cela signifie que le changement de carte est efficace et qu'il doit être conservé.

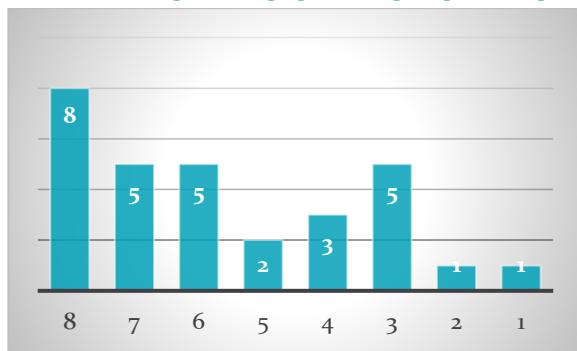
RESULTATS

Les résultats obtenus montrent que le deck s'est amélioré et que les parties sont de plus en plus rapide. Nous passons ainsi d'environ 10% de victoire à quasiment 100% de victoire. Le coût des cartes en moyenne a baissé également.

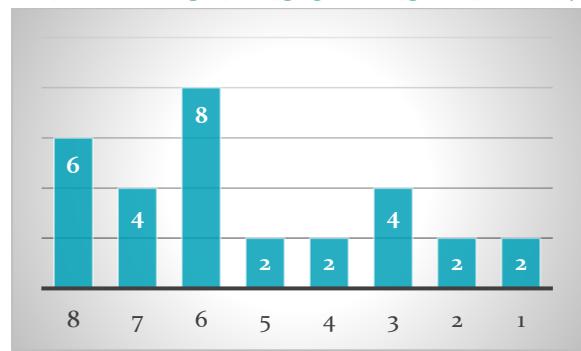
DONNEES



REPARTITION DES CARTES AU DEBUT :



REPARTITION DES CARTES A LA FIN :



DECKS

| ID° | Cost | Attack | Defense |
|-----|------|--------|---------|
| 1 | 1 | 0 | 2 |
| 3 | 2 | 0 | 4 |
| 4 | 3 | 0 | 5 |
| 19 | 3 | 1 | 4 |
| 20 | 3 | 1 | 5 |
| 58 | 3 | 4 | 1 |
| 59 | 3 | 4 | 2 |
| 36 | 4 | 2 | 6 |
| 71 | 4 | 5 | 2 |
| 82 | 4 | 6 | 2 |
| 8 | 5 | 0 | 9 |
| 73 | 5 | 5 | 4 |
| 39 | 6 | 2 | 9 |
| 52 | 6 | 3 | 8 |
| 95 | 6 | 7 | 5 |
| 102 | 6 | 8 | 3 |
| 116 | 6 | 10 | 2 |
| 28 | 7 | 1 | 13 |
| 87 | 7 | 6 | 7 |
| 96 | 7 | 7 | 6 |
| 104 | 7 | 8 | 5 |
| 126 | 7 | 12 | 1 |
| 29 | 8 | 1 | 14 |
| 43 | 8 | 2 | 13 |
| 44 | 8 | 2 | 14 |
| 57 | 8 | 3 | 13 |
| 69 | 8 | 4 | 12 |
| 98 | 8 | 7 | 8 |
| 99 | 8 | 7 | 9 |
| 133 | 8 | 14 | 1 |

| ID° | Cost | Attack | Defense |
|-----|------|--------|---------|
| 1 | 1 | 0 | 2 |
| 16 | 1 | 1 | 1 |
| 32 | 2 | 2 | 2 |
| 45 | 2 | 3 | 1 |
| 46 | 3 | 3 | 2 |
| 47 | 3 | 3 | 3 |
| 58 | 3 | 4 | 1 |
| 70 | 3 | 5 | 1 |
| 49 | 4 | 3 | 5 |
| 82 | 4 | 6 | 2 |
| 51 | 5 | 3 | 7 |
| 84 | 5 | 6 | 4 |
| 11 | 6 | 0 | 12 |
| 25 | 6 | 1 | 10 |
| 39 | 6 | 2 | 9 |
| 86 | 6 | 6 | 6 |
| 94 | 6 | 7 | 4 |
| 109 | 6 | 9 | 2 |
| 115 | 6 | 10 | 1 |
| 115 | 6 | 10 | 1 |
| 13 | 7 | 0 | 14 |
| 42 | 7 | 2 | 12 |
| 127 | 7 | 12 | 2 |
| 127 | 7 | 12 | 2 |
| 14 | 8 | 0 | 15 |
| 15 | 8 | 0 | 16 |
| 90 | 8 | 6 | 10 |
| 132 | 8 | 13 | 3 |
| 134 | 8 | 14 | 2 |
| 135 | 8 | 15 | 1 |

Partie 2 : « La provocation est l'arme des faibles. » - Jean-Paul Sartre

DESCRIPTION

Dans cette deuxième partie, le but était d'étendre les règles mises en place en ajoutant l'attribut Provocation | Taunt | qui impose aux cartes attaquentes d'attaquer la carte défenseure possédant cet attribut.

PROTOCOLE

Le protocole consiste à faire 1000 itérations d'affrontement, une itération d'affrontement consiste à changer une carte aléatoirement, puis à faire 1000 parties et de noter le taux de victoire à chaque partie.

Si le taux de victoire à l'itération 't' est supérieur à celui 't-1' cela signifie que le changement de carte est efficace et qu'il doit être conservé.

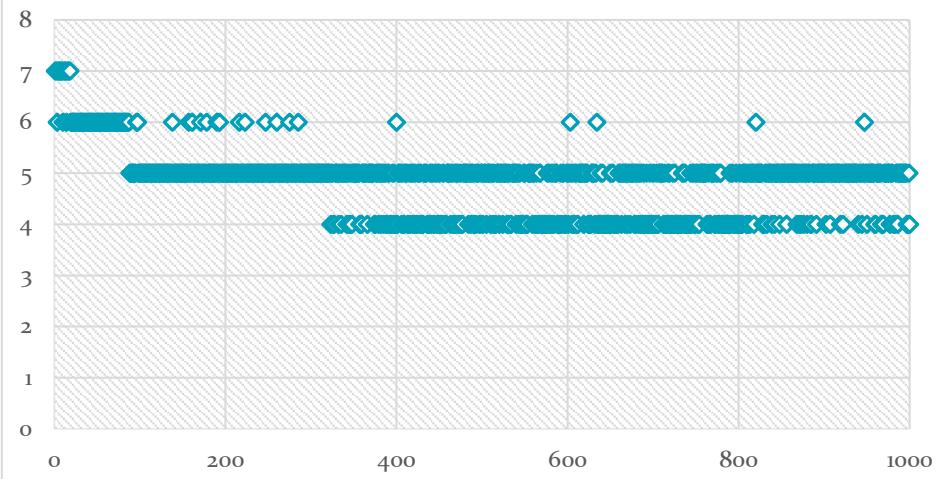
RESULTATS

Les résultats obtenus montrent que le deck s'est améliorer et que les parties sont de plus en plus rapide.

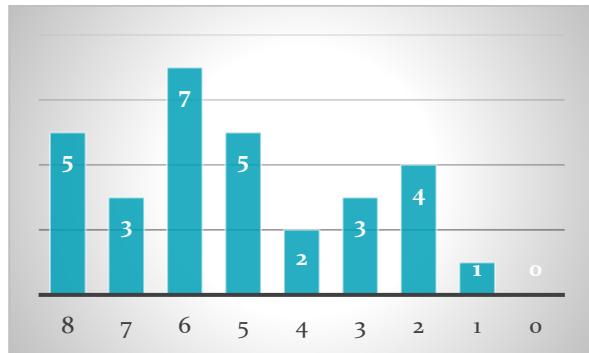
DONNEES



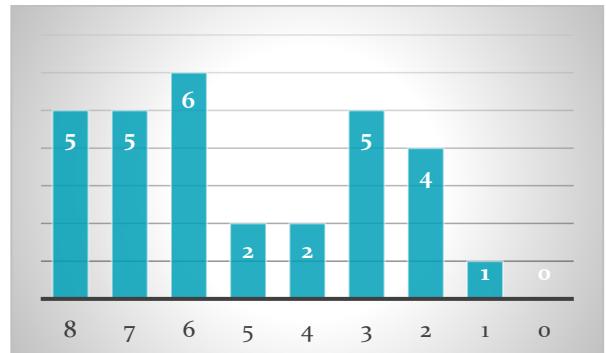
Nombre moyen de tour par partie



REPARTITION DES CARTES AU DEBUT :



REPARTITION DES CARTES A LA FIN :



DECKS

| ID° | Cost | ATK | DEF | Taunt |
|-----|------|-----|-----|-------|
| 16 | 1 | 1 | 1 | 0 |
| 3 | 2 | 0 | 4 | 0 |
| 17 | 2 | 1 | 2 | 0 |
| 31 | 2 | 2 | 1 | 0 |
| 32 | 2 | 2 | 2 | 0 |
| 46 | 3 | 3 | 2 | 0 |
| 58 | 3 | 4 | 1 | 0 |
| 137 | 3 | 0 | 2 | 1 |
| 22 | 4 | 1 | 7 | 0 |
| 71 | 4 | 5 | 2 | 0 |
| 23 | 5 | 1 | 8 | 0 |
| 92 | 5 | 7 | 2 | 0 |
| 142 | 5 | 0 | 7 | 1 |
| 164 | 5 | 2 | 4 | 1 |
| 184 | 5 | 4 | 3 | 1 |
| 26 | 6 | 1 | 11 | 0 |
| 52 | 6 | 3 | 8 | 0 |
| 95 | 6 | 7 | 5 | 0 |
| 103 | 6 | 8 | 4 | 0 |
| 185 | 6 | 4 | 4 | 1 |
| 193 | 6 | 5 | 3 | 1 |
| 207 | 6 | 7 | 2 | 1 |
| 158 | 7 | 1 | 10 | 1 |
| 196 | 7 | 5 | 6 | 1 |
| 209 | 7 | 7 | 4 | 1 |
| 128 | 8 | 12 | 3 | 0 |
| 181 | 8 | 3 | 10 | 1 |
| 216 | 8 | 8 | 5 | 1 |
| 222 | 8 | 10 | 2 | 1 |
| 225 | 8 | 11 | 2 | 1 |

| ID° | Cost | ATK | DEF | Taunt |
|-----|------|-----|-----|-------|
| 16 | 1 | 1 | 1 | 0 |
| 31 | 2 | 2 | 1 | 0 |
| 3 | 2 | 0 | 4 | 0 |
| 45 | 2 | 3 | 1 | 0 |
| 31 | 2 | 2 | 1 | 0 |
| 70 | 3 | 5 | 1 | 0 |
| 19 | 3 | 1 | 4 | 0 |
| 46 | 3 | 3 | 2 | 0 |
| 137 | 3 | 0 | 2 | 1 |
| 59 | 3 | 4 | 2 | 0 |
| 22 | 4 | 1 | 7 | 0 |
| 81 | 4 | 6 | 1 | 0 |
| 92 | 5 | 7 | 2 | 0 |
| 101 | 5 | 8 | 2 | 0 |
| 193 | 6 | 5 | 3 | 1 |
| 40 | 6 | 2 | 10 | 0 |
| 207 | 6 | 7 | 2 | 1 |
| 185 | 6 | 4 | 4 | 1 |
| 95 | 6 | 7 | 5 | 0 |
| 76 | 6 | 5 | 7 | 0 |
| 196 | 7 | 5 | 6 | 1 |
| 130 | 7 | 13 | 1 | 0 |
| 78 | 7 | 5 | 9 | 0 |
| 217 | 7 | 9 | 1 | 1 |
| 187 | 7 | 4 | 6 | 1 |
| 79 | 8 | 5 | 10 | 0 |
| 190 | 8 | 4 | 9 | 1 |
| 128 | 8 | 12 | 3 | 0 |
| 181 | 8 | 3 | 10 | 1 |
| 204 | 8 | 6 | 6 | 1 |

Partie 3 : « Celui qui a tout, convoite tout. » - Proverbe français

DESCRIPTION

Dans dernière partie, le but était d'étendre les règles mises en place en ajoutant les attributs suivants :

- Piétinement | **Trample** | qui permet aux cartes attaquantes de transférer le reste des dégâts qu'elle inflige à la carte défenseure si cette dernière est détruite.
- Distorsion | **Distortion** | qui impose à la carte défenseur ayant **Taunt** d'avoir également **Distortion** pour pouvoir défendre la carte attaquante le possédant.
- Initiative | **First Strike** | qui permet à la carte la possédant d'infliger ses dégâts avant que la carte adverse puisse attaquer.

La Setlist a donc été étendu à 1192 cartes à présents.

PROTOCOLE

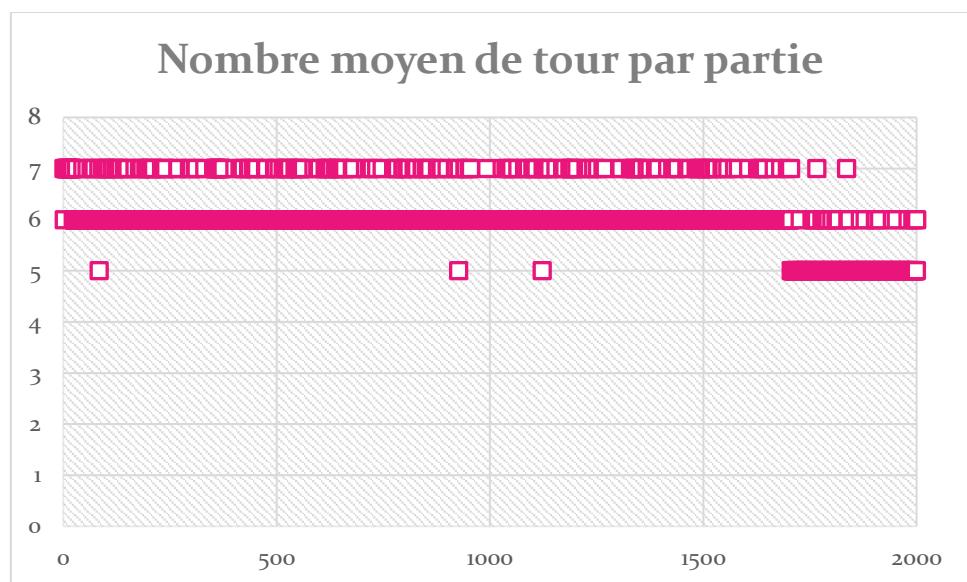
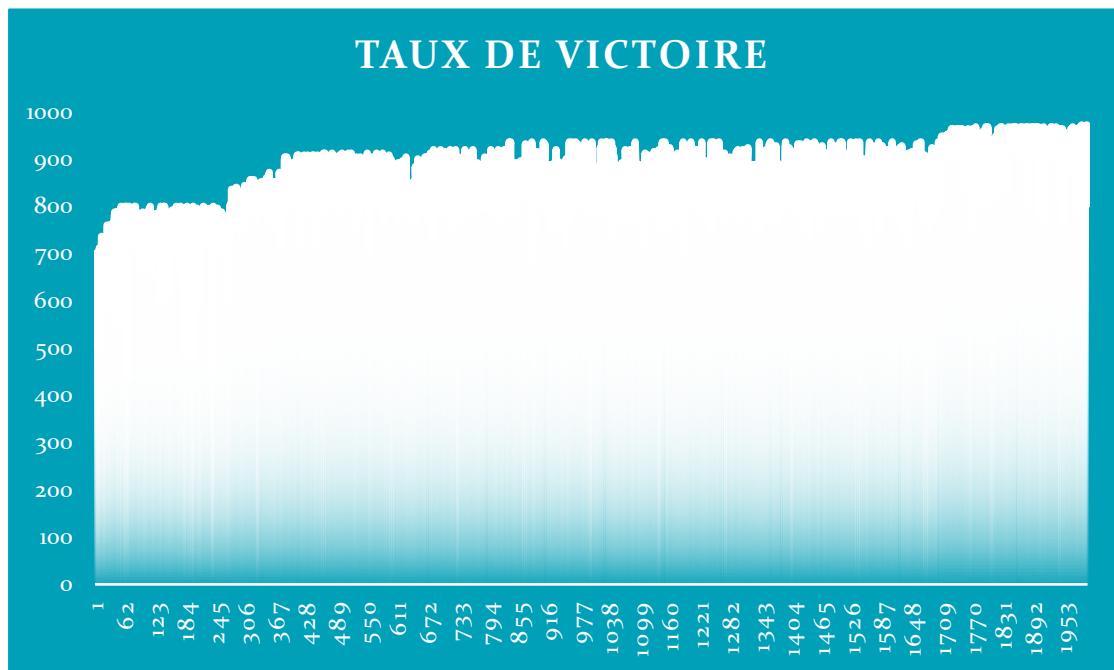
Le protocole consiste à faire 2000 itérations d'affrontement, une itération d'affrontement consiste à changer une carte aléatoirement, puis à faire 1000 parties et de noter le taux de victoire à chaque partie. J'ai décidé d'augmenter le nombre d'itérations étant donné le nombre de cartes conséquent de la Setlist.

Si le taux de victoire à l'itération 't' est supérieur à celui 't-1' cela signifie que le changement de carte est efficace et qu'il doit être conservé.

RESULTATS

Les résultats obtenus montrent que le deck s'est améliorer et que les parties sont de plus en plus rapide. Nous passons ainsi d'environ 10% de victoire à quasiment 100% de victoire.

DONNEES



DECKS

| ID° | Cost | ATK | DEF | Taunt | Trample | Distortion | 1st Strike |
|------|------|-----|-----|-------|---------|------------|------------|
| 16 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 241 | 2 | 1 | 1 | 0 | 0 | 1 | 0 |
| 162 | 4 | 2 | 2 | 1 | 0 | 0 | 0 |
| 438 | 4 | 3 | 2 | 0 | 1 | 0 | 0 |
| 1065 | 5 | 0 | 2 | 1 | 1 | 0 | 1 |
| 92 | 5 | 7 | 2 | 0 | 0 | 0 | 0 |
| 1165 | 6 | 0 | 2 | 1 | 1 | 1 | 1 |
| 1037 | 6 | 5 | 2 | 0 | 1 | 0 | 1 |
| 212 | 6 | 8 | 1 | 1 | 0 | 0 | 0 |
| 1031 | 6 | 4 | 4 | 0 | 1 | 0 | 1 |
| 378 | 6 | 5 | 2 | 1 | 0 | 1 | 0 |
| 870 | 6 | 0 | 8 | 0 | 0 | 1 | 1 |
| 678 | 7 | 4 | 2 | 1 | 1 | 1 | 0 |
| 850 | 7 | 6 | 3 | 1 | 0 | 0 | 1 |
| 78 | 7 | 5 | 9 | 0 | 0 | 0 | 0 |
| 668 | 7 | 2 | 5 | 1 | 1 | 1 | 0 |
| 511 | 7 | 0 | 9 | 1 | 1 | 0 | 0 |
| 263 | 7 | 2 | 10 | 0 | 0 | 1 | 0 |
| 853 | 7 | 7 | 1 | 1 | 0 | 0 | 1 |
| 1174 | 7 | 1 | 4 | 1 | 1 | 1 | 1 |
| 862 | 8 | 10 | 1 | 1 | 0 | 0 | 1 |
| 216 | 8 | 8 | 5 | 1 | 0 | 0 | 0 |
| 895 | 8 | 2 | 10 | 0 | 0 | 1 | 1 |
| 565 | 8 | 8 | 3 | 1 | 1 | 0 | 0 |
| 1153 | 8 | 5 | 5 | 0 | 1 | 1 | 1 |
| 847 | 8 | 5 | 6 | 1 | 0 | 0 | 1 |
| 361 | 8 | 2 | 9 | 1 | 0 | 1 | 0 |
| 496 | 8 | 10 | 4 | 0 | 1 | 0 | 0 |
| 223 | 8 | 10 | 3 | 1 | 0 | 0 | 0 |
| 320 | 8 | 9 | 4 | 0 | 0 | 1 | 0 |

| ID° | Cost | ATK | DEF | Taunt | Trample | Distortion | 1st Strike |
|------|------|-----|-----|-------|---------|------------|------------|
| 16 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 46 | 3 | 3 | 2 | 0 | 0 | 0 | 0 |
| 60 | 4 | 4 | 3 | 0 | 0 | 0 | 0 |
| 1020 | 5 | 3 | 2 | 0 | 1 | 0 | 1 |
| 288 | 5 | 5 | 2 | 0 | 0 | 1 | 0 |
| 467 | 5 | 6 | 1 | 0 | 1 | 0 | 0 |
| 460 | 5 | 5 | 3 | 0 | 1 | 0 | 0 |
| 744 | 5 | 4 | 3 | 0 | 0 | 0 | 1 |
| 346 | 5 | 1 | 4 | 1 | 0 | 1 | 0 |
| 827 | 5 | 3 | 1 | 1 | 0 | 0 | 1 |
| 93 | 5 | 7 | 3 | 0 | 0 | 0 | 0 |
| 383 | 6 | 6 | 1 | 1 | 0 | 1 | 0 |
| 357 | 6 | 2 | 5 | 1 | 0 | 1 | 0 |
| 1037 | 6 | 5 | 2 | 0 | 1 | 0 | 1 |
| 86 | 6 | 6 | 6 | 0 | 0 | 0 | 0 |
| 1022 | 6 | 3 | 4 | 0 | 1 | 0 | 1 |
| 1090 | 7 | 3 | 3 | 1 | 1 | 0 | 1 |
| 464 | 7 | 5 | 7 | 0 | 1 | 0 | 0 |
| 490 | 7 | 9 | 3 | 0 | 1 | 0 | 0 |
| 910 | 7 | 4 | 6 | 0 | 0 | 1 | 1 |
| 633 | 7 | 7 | 2 | 0 | 1 | 1 | 0 |
| 832 | 7 | 3 | 6 | 1 | 0 | 0 | 1 |
| 1191 | 8 | 6 | 1 | 1 | 1 | 1 | 1 |
| 393 | 8 | 8 | 2 | 1 | 0 | 1 | 0 |
| 934 | 8 | 8 | 4 | 0 | 0 | 1 | 1 |
| 382 | 8 | 5 | 6 | 1 | 0 | 1 | 0 |
| 1156 | 8 | 6 | 3 | 0 | 1 | 1 | 1 |
| 328 | 8 | 11 | 3 | 0 | 0 | 1 | 0 |
| 294 | 8 | 5 | 8 | 0 | 0 | 1 | 0 |
| 135 | 8 | 15 | 1 | 0 | 0 | 0 | 0 |

Conclusion

Le projet répond aux objectifs initiaux que nous nous sommes fixés. Nous sommes en mesure de produire plusieurs decks et de les optimiser chacun. Ou encore de pouvoir analyser si des cartes sont trop récurrentes ou au contraire pas assez.

En revanche, plusieurs pistes d'améliorations sont possibles :

- Optimiser la Setlist en elle-même en comptant quand les cartes quittent ou rejoignent le deck afin d'identifier les meilleures et les pires cartes.
- Optimiser le deck contre plusieurs decks différents, eux-mêmes optimisés. Cela pourrait être fait en se servant du multithreading par exemple.

Malheureusement, l'implémentation de ses améliorations n'ont pu se faire faute d'un manque de temps.