
**Université de Caen Normandie**
**UFR des Sciences**
**Département Informatique**

**3ème année de licence d'informatique**

# PolTron: La coalition

## Expérimentations sur coalition d'IA via le jeu Tron


Christopher JACQUIOT, Vincent DE MENEZES,
Alexis MORTELIER, Walid IDOUCHE

Tuteur du projet: Gregory BONNET

Année universitaire : 2018 / 2019

Jury : –
*(si composition du jury connue)*

Soutenu le – mars 2019
*(si date connue)*

# Table des matières

# Table des figures

# Liste des tableaux

# Remerciements

Nous tenons à remercier notre tuteur M. Gregory BONNET, pour la proposition de ce sujet passionnant.

**Résumé**

Dans ce projet mêlant intelligence artificielle, simulation et analyse, nous allons devoir créer un jeu inspiré de Tron sur lequel nous allons faire jouer plusieurs équipes, une coalition et un joueur seul, leur donnant une différence d'intelligence telle que le joueur solo sera le plus intelligent, et nous allons ensuite devoir analyser les résultats de leurs parties pour déterminer les paramètres les plus optimaux pour que cette coalition gagne contre le joueur seul. Pour réaliser cela nous allons avoir recours à diverses technologies pour résoudre les divers problèmes auquels nous allons nous confronter. Parmi ces technologies, le python sera utile pour réaliser rapidement notre modèle et notre interface, le Sqlite avec sa portabilité et sa forte intégration avec la plupart des langages sera primordial pour stocker et manipuler les données résultantes de nos simulations, et le langage d'analyse statistique R sera un grand atout pour aider à raisonner rapidement à partir de ces résultats.

**Abstract**

In this project about artificial intelligence, simulation and analysis, we will have to make an Tron-inspired game on which we will make two teams fight each other, a coalition and an alone player, both having different intelligence levels, the solo player being the most intelligent, and then analyze the results of their games to determine the optimum parameters to make the coalition win against the solo player. To realize this we will need to make good use of diverse technologies to deal with the problems we will face. Amongst thos technologies, Python will be useful to produce efficiently both our model and interface, Sqlite thanks to it's portability and deep integration with most languages will be primordial to store and manipulate the data resulting from our simulations, and the statistical analysis programming language R will be a great asset to help reason quickly from those results.

*Keywords :* **AI analysis simulation Tron**

# Première partie

# Analyse du projet

# 1. Introduction

## 1.1. Objectif général du projet

**Quel est le problème à régler ?** Dans un jeu de Tron dont les règles sont explicitées dans la partie sur le modèle, nous allons faire jouer deux équipes :

— Un joueur seul et intelligent

— Une coalition de joueurs moins intelligents

Le but est d'analyser les meilleurs paramètres pour que notre coalition soit statistiquement la plus efficace contre le joueur seul, si une tendance se dégage de nos simulations.

En d'autres termes, nous allons tenter de répondre à la question :

> Combien faut-il d'idiots pour prendre l'avantage sur un joueur plus intelligent ?

## 1.2. Objectifs à atteindre

**Simulateur** Nous allons devoir permettre à la personne voulant générer des données de paramètrer les intervalles et fréquences d'échantillonnage le plus précisément possible pour permettre de génerer des données plus précises sur certaines conditions si besoin.

Les paramètres inter-dépendants tels que le nombre de joueurs et la taille des cartes doivent être automatiquement régulés pour un fonctionnement entièrement automatisé.

Une interface permettant de suivre la progression de la simulation est aussi très importante pour estimer quand terminent nos simulations.

#### Modèle de jeu

#### IA et son heuristique

**Stockage de masse** Au vu des grandes quantités de données potentielles, une base de donnée bien structurée avec des vues permettant de faciliter l accès aux informations pertinentes pour l'analyse sera primordiale.

**Analyse statistique** Nous allons devoir faciliter la visualisation et le travail sur nos données afin de permettre de se concentrer sur l'analyse plutôt que sur les outils d'analyse. Il sera donc important d'unifier au possible les moyens d'analyse des données et de rédaction d'analyses pour augmenter notre efficacité.

## Deuxième partie

# Cahier des charges

| Fonction de service | Critère/Module | Niveau | Fléxibilité | Contribuant au module |
|---|---|---|---|---|
| FONCTION PRINCIPALE | | | | |
| FP 0 | Exemple d'utilisation | Expliquer comment utiliser le cahier des charges | ~~Chaque étudiants doit comprendre comment utilisé le cahier des charges;~~ ~~Une fois une tache réalisée celle-ci doit être rayé par un code couleur (rouge : fait mais à optimiser/ou pas sûr (souvent le cas)/vert : finis parfaitement;~~ ~~Une fois tous les éléments de la case fléxibilité rayées ("rouge"/"vert") le module et la fonction de service doivent être mis en vert (si le module doit être abandonné pour des raisons x ou y celui-ci doit être mis en rouge);~~ | Vincent |
| FP 1 | Jeu Tron | Création d'un plateau, avec les règles du jeu Tron afin de créer un "jeu" jouable avec plusieurs controleur | Déplacement sur le plateau avec des commandes simples; Laisse un mur à la position d'origine lors d'un déplacement; Déplacement possède sens 4 directions possible; Partie finis lorsqu'il ne reste plus de joueur dans les 2 camps ("bleu"/" rouge"); Un joueur est éliminé lorsqu'il se trouve sur une case de type mur; Type de case ("mur"/" vide"); Extremité du plateau composé de case de type ("mur"); Un tour d'un joueur est composé d'un seul déplacement obligatoire avec 3 directions possible; Dans un cycle de tour chaque joueurs jouent 1 tour; Implémentation d'un compteur de tour; | Alexis |
| FP 2 | Intelligence Artificielle | Choisis un déplacement le moins risqué déterminer par l'analyse du plateau | Implémentation de l'algorithme paranoide; ~~Heuristique~~ : ~~vérification de la zone de contrôle d'un camps sur le plateau~~; Stockage/lecture des états d'un graphe; | Vincent |
| FP 3 | Analyse des expérimentations | Création d'un algorithme permettant d'établir un graphe de chaleur/autre, à partir de plusieurs résultats. | Le résultat d'une simulation sera composé de plusieurs informations ("taille/aire du plateau", "nombre de la coalition", "profondeur de recherche des 2 camps", "moyenne de victoire contre la coalition (en %)", "position de départ"***, "ordre des joueurs"***); "le nombre de tour"); Avec les différentes informations d'une simulation, créer un système permettant de juger une partie selon le taux de victoire + le nombre de tour, afin de déterminer une seule position dans le graphe de chaleur; Faire une moyenne de victoire lorsqu'il existe une configuration identique avec un résultat différent; ***fonctionnalité aléatoire ne doit pas être systématiquement pris en compte pour commencer les premières expériences | Christopher |
| FP4 | Simulateur | Génère l'espace de recherche (paramètres initiaux) et gère l'ajout des résultats de chaque partie à la bdd | Le simulateur doit générer toutes les combinaisons valides et interessantes de paramètres initiaux (M,N, C, Ds, Dc) et lancer un nombre suffisamment grand de parties à ordre de joueur et positions initiales aléatoires avec ces parametres, pour explorer les statistiques de victoires de chacunes de ces combinaisons | Christopher |
| FP 5 | Rapport | Présenter le travail du groupe | Expliquer les modules implémenter; Montrer les résultats obtenus; Expliquer/Faire des hypothèses avec les résultats obtenue; | |
| FONCTION DE CONTRAINTE | | | | |

| | | | | |
|---|---|---|---|---|
| FC 1 | Jeu Tron | L'initialisation du plateau doit être facilement modifiable | Position des joueurs; Taille du plateau; | |
| FC 2 | Intelligence Artificielle | Elasticité des recherches en profondeur | Profondeur des 2 camps facilement configurable; | |
| FC 3 | Intelligence Artificielle | Stockage de plateau | Le Stockage doit avoir plusieurs états, pouvant être identifier à un plateau et fournir une direction qui a déjà été calculé; Il sera composé de l'état + la direction à choisir; | |
| FC 4 | Jeu Tron | Stockage d'une partie | Le Stockage d'une partie doit garder les configurations des parties avec le nombre de parties simulé suivis du taux de réussite de la partie; | |
| FC 5 | Rapport | Rédaction | Le rapport devra être rédigé en LATEX | |
| FC 6 | Module | Autonomie | Chaque modules doivent pouvoir se débrouiller seul sans l'aide d'autre module; Le partage d'information se fera à l'aide du stockage; Permet une diversification dans les langages à adopter; | |
| FC 7 | Logiciel | Langage | Chaque modules doivent avoir un langage qui convient pour sa taches à effectuer; Le langage doit pouvoir être facile à manipuler pour les modules; | |
| FC 8 | Jeu Tron | Taille du plateau | Le plateau peut être rectangulaire | |

## 1.3. Spécifications

### 1.3.1. Spécification des paramètres de simulation

**Attention** Pour pouvoir simuler correctement, il est nécessaire que les paramètres d'entrée donnés au simulateur ne brisent pas les règles suivantes.

$$param > 0 \tag{1.1}$$

$$min_X \leq max_X \tag{1.2}$$

$$0 < Dc < Ds \tag{1.3}$$

### 1.3.2. Contraintes techniques

**Temps imparti réduit** Suite à une annulation de la matière puis à la réouverture de celle ci, le temps imparti pour ce projet as été considérablement amoindri.

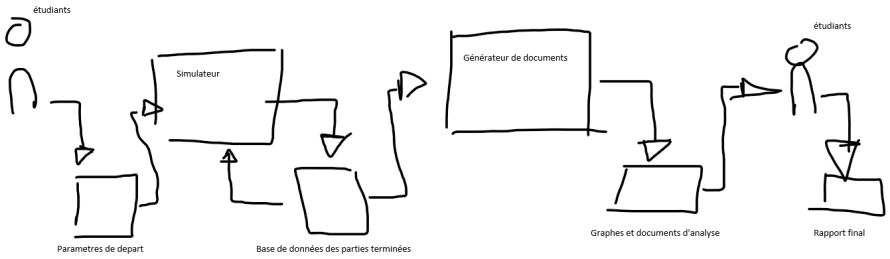
Nous avons environ 5 semaines pour mener ce projet à terme à compter du 31 janvier 2019. Il est donc nécessaire de réduire un maximum les temps de développement pour le mener à bien.

Cela a mené à la nécessité d'évaluer nos options de façon la plus pragmatique possible en termes de coûts en temps d'implémentation.

## 1.4. Choix techniques

### 1.4.1. Schémas de conception

#### Architecture



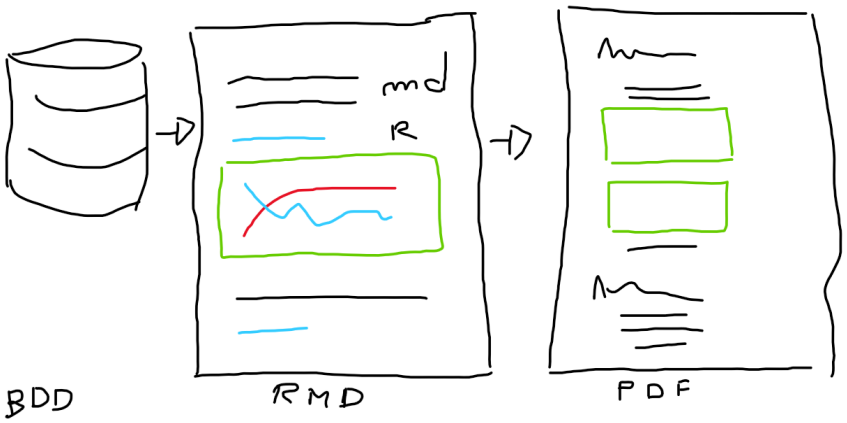

#### Base de données

| game: | player_order: | initial_positions: | Game_results: | Important moments: | Deaths: |
|---|---|---|---|---|---|
| *partie_id | *partie_id | *partie_id | *partie_id | *partie_id | *partie_id |
| M | Ini_Pos | Ini_Pos | Ended_tick | tick | Tick |
| N | Player_Id | Player_Id | Won | Plateau_string | Player_id |
| Ds | Order | Order | | N_walls | |
| Dc | | | | Coalition_deaths | |
| C | | | | | |

**Modèle de jeu**

**Heuristique**

**Analyse**




### 1.4.2. Langages utilisés

**Module simulation :**

- Python pour l'interface commande et les sous modules internes.
- SQL pour la génération et l'interaction avec la bdd

**Module analyse :**

- R pour la génération des graphes et la manipulation des données
- Markdown pour la rédaction du rapport d'analyse

## Troisième partie

# Historique des travaux réalisés

| DATE | PRENOM | MODULE | COMMENTAIRE | TEMPS | RESTE A FAIRE | PROBLEME (FACULTATIF) |
|---|---|---|---|---|---|---|
| 31/01/2019 | Vincent | Mise en place : creation du drive et des fichiers commun pour le déroulement du projet | création d'un drive + planning + cahier des charges | 1H | Accepter les autres membres et configuer le Drive | N/A |
| 31/01/2019 | Christopher | Mise en place et analyse préliminaire | rédaction du récapitulatif du projet + analyse des objectifs du projet + analyse d une optimisation algorithmique pour evalutation de minmax | 2H | analyse du format des données en externe et interne | N/A |
| 31/01/2019 | Christopher | analyse préliminaire | analyse des données à stocker + analyse des données de la simulation + anayse de l'architecture du projet | 2H | analyse des algorithmes internes + analyse des moyens présents + analyse des outils utilisés | N/A |
| 31/01/2019 | Christopher | Préparation | Ajout d'un résumé de mes expériences sur les outils potentiels et de mes potentiels outils déjà prêts | 1H | N/A | N/A |
| 01/02/2019 | Vincent | Cahier des charges | Création du cahier des charges fonctionnel | 3H | Completer les informations manquantes (si présente) | N/A |
| 01/02/2019 | Vincent | Mise en place: config du Drive | Ajout des étudiants et des droits | ~ | N/A | N/A |
| 01/02/2019 | Vincent | Récapitulatif | Section Expériences (Vincent) remplis | 30MIN | N/A | N/A |
| 01/02/2019 | Alexis | Analyse du projet | Lecture de differentes doc sur le jeu, les differentes IA, exemple de code, amelioration possible, infos importante au "n" moment pour avoir un graphe interessant, etc. | 3H | N/A | N/A |
| 01/02/2019 | Walid | Analyse du projet | Analyse des doccuments du jeux , essayer de comprendre le fonctionnement du projet . | 2H | N/A | N/A |
| 02/02/2019 | Christopher | Création du module de bdd | Design et génération automatique de la bdd + interface python d'entrée des données | 2H30 | Test avec de vraies données et potentiellement ajout du stockage du cache pour l'heuristique de l 'IA | Nécessite d'avoir l heuristique de prete avant de pouvoir determiner le format du cache |
| 02/02/2019 | Christopher | Création d'un prototype d'analyseur | Générateur de données aléatoire et rendu des graphes fonctionnels | 6H | Avoir de vraies données pour affiner les affichages les plus interressants + implementer des graphes de comparaison de profondeur de recherche | N/A |
| 02/02/2019 | Vincent | Convertion heuristique | De JAVA à Python | 6H | Rendre l'utilisation de l'heuristique compatible avec la structure du jeu | Reconnaissance des camps bleu et rouge non automatisé |
| 03/02/2019 | Christopher | Amélioration Analyses | Ajout d'une methodologie d'exploration statistique et des fonctions permettant la génération des graphes utiles | 6H | Dupliquer les analyses pour chaque facteur d'entrée des parties et chaque élément d'analyse voulu | N/A |
| 02/02/2019<br>03/02/2019 | Alexis | Conception du jeu en mode joueur contre joueur | Implementation de toute les fonctionnalités et de l'architecture du jeu Tron* | 10h | deplacements | N/A |
| 03/02/2019 | Christopher | Couche de gestion de la simulation | Génére et teste autant de fois que voulu des parties selon toutes les combinaisons de paramètres de départ voulues, gère l'envoi à la bdd et l'affichage du temps restant estimé | 8H | Brancher la simulation sur de vraies parties | En attente du modèle de partie pour le reste |
| 04/02/2019 | Christopher | Interface utilisateur et valeurs par défaut | Permet de modifier les paramètres de départ et d'en savoir plus sur les arguments de recherche | 2H | N/A | N/A |
| 04/02/2019 | Christopher | Début de rédaction du rapport | Retraits de catégories inutiles pour notre projet, ajout de début de contenu sur les outils utilisé, préparé des pages de chapitres potentiels, ai rédigé le résumé et l'abstract du projet | 3H | Rédiger le contenu des chapitres sur ce que j'ai fait; mettre à jour les pdf de cahier des charges quand le projet sera terminé | Il faudra remettre à jour les pdf de cahier et d historique a la fin du projet |
| 04/02/2019 | Christopher | Début de rédaction du rapport | Ajout de schémas et des documents internes (cahier, historique, analyse); Début de rédaction de titres de paragraphes et de sections dans mes chapitres | 3H | Continuer de rédiger | voir au dessus |
| 04/02/2019 | Alexis | Finition des deplacements, et optimisation primaire du code | Voir dans le détails s'il n'y a pas possibilité d'optimisé encore plus | 2H | Implementation ressortant les informations importantes à l'analyse du jeu | N/A |
| 05/02/2019<br>06/02/2019 | Alexis | Changement de pensé du code du jeu, optimisation de celui-ci | Code optimisé à fond, 60 fois plus rapide qu'avant | 4h | FIIIINIIIIIEEEE | N/A |
| 06/02/2019 | Christopher | Compilation Cython | Compilation fonctionnelle du code python via cython et application d'optimisations pour la transcompilation python -> C | 2H | N/A | N/A |
| 06/02/2019 | Christopher | Modélisation du système | Création d'une modélisation basée sur de la physique pour tenter de simuler notre intelligence d une facon differente | 3H | Tweaker le modèle quand on aura accès aux données de l'IA pour tenter de le faire fit le plus possible. | N/A |
| 06/02/2019 | Christopher | Debug de la toolchain Simu -> db -> rapport complet | Debugging complet de l insertion des données dans la bdd et de sa lecture par R | 1H | N/A | N/A |
| 07/02/2019 | Christopher | Optimisation du modèle et des données récoltées | Retrait de la string d'état pour accélérer un maximum la génération de données importantes et éviter les opérations inutiles, gain de temps final sur l'éxecution du modèle: ~25% | 2H | N/A | N/A |
| 07/02/2019 | Christopher | Écriture du chapitre sur la simulation de modèle | Écriture complète et production de schémas illustratifs pour le chapitre de la simulation. | 4H | N/A | Ajout du pdf d'analyses du modèle |
| 07/02/2019 | Christopher | Écriture du chapitre sur le simulateur | Écriture complète et production de schémas illustratifs pour le chapitre du simulateur. | 4H | N/A | N/A |
| 08/02/2019 | Christopher | Ajout de facteurs et amélioration du rendu des analyses | Ajout des facteurs M*N et Ds-Dc à nos analyses de corrélations | 2H | N/A | N/A |
| 08/02/2019 | Christopher | Écriture du chapitre sur l'exploration des données | Écriture complète et production de schémas illustratifs pour le chapitre sur l'exploration statistique. | 3H | N/A | N/A |
| 08/02/2019 | Christopher | Rectifications d'après retours | Ajouté plus de details dans l'abstract concernant le projet lui même, corrigé des fautes oubliées, modifié légèrement quelques phrases grammaticalement incorrectes | 10MIN | N/A | N/A |
| 04/02/2019<br>08/02/2019 | Chris's PC | Génération d'huile de coude (simulation de modèles) | Génération de données résultant d'un total de 2-3 millions de parties simulée via notre modèle physique sur différents sets de données | ~40H | Recommencer pour le modèle avec IA | R.I.P. PC, bientôt tu pourras te reposer x) |
| 08/02/2019 | Christopher | Analyse et rédaction des résultats des données du modèle | Écriture complète et production de schémas illustratifs pour le chapitre d'analyse des résultats du modèle | 4H | voir ci dessus | N/A |
| 09/02/2019 | Vincent | Implémentation Heuristique avec le jeu | Implémentation en objet et non objet d'une heuristique, celle-ci calcule la portée de tout les joueurs, puis attribue les case les plus proches aux équipes. Ajout d'une méthode permettant la vérification des portées de chaques joueurs. | 9H | Faire des tests avec un simulation plus grande (nombre de la coalition, taille du plateau); Légère amélioration des conditions dans la mesure du possible | N/A |
| 10/02/2019 | Christopher | Calcul de différence de vitesse entre simulation et modèle | Calcul et ajout de la différence du nombre de cases à checker sur la meme partie extrême entre le modèle simulé et le modèle d'IA | 2H | N/A | N/A |

**Concernant Walid** Suite à une discussion sur les expériences, et compétences de chacun pour analyser comment mener au mieux ce projet, un accord à été passé avec Walid pour qu'il puisse se familiariser de son coté à Python et aux concepts du projets en tentant d'en réaliser un maximum de son coté.

Afin de ne pas le délaisser non plus, il as été encouragé qu'il pose ses éventuelles questions et s'inspire du code principal pour expérimenter et rattrapper son éventuel retard sur certains concepts.

## 1.5. Outils de programmation

**Alexis :**

**Vincent :**

**Christopher :**

— Pycharm + l'extension Sonar Lint pour programmer en Python
— Rstudio pour programmer le projet en R et étudier le contenu de la base de données

**Walid :**

## 1.6. Bibliothèques utilisées

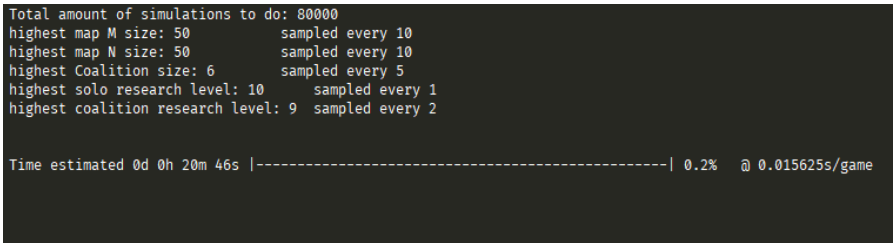
### 1.6.1. Module simulation

— time pour estimer le temps restant avant completion des simulations
— sqlite3 pour l'interfacage avec la bdd sqlite

### 1.6.2. Module analyse

— dplyr pour faciliter la manipulation et la selection par sémantique des données
— ggplot2 pour ses graphes de qualité et facile à configurer
— GGally pour ses outils d'analyse de tables de données complètes
— RSQLite pour l'interfacage avec la bdd sqlite

## 1.7. Interface



# Quatrième partie

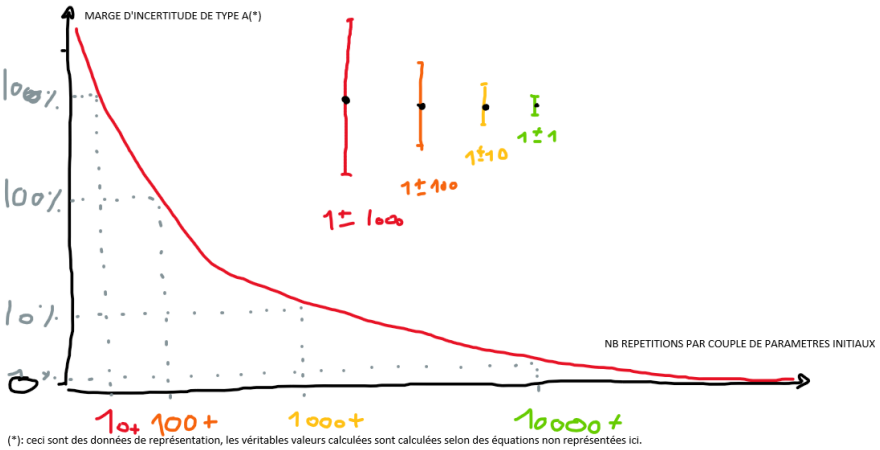
# Réalisation

# 2. Simulateur

## 2.1. Nécessités

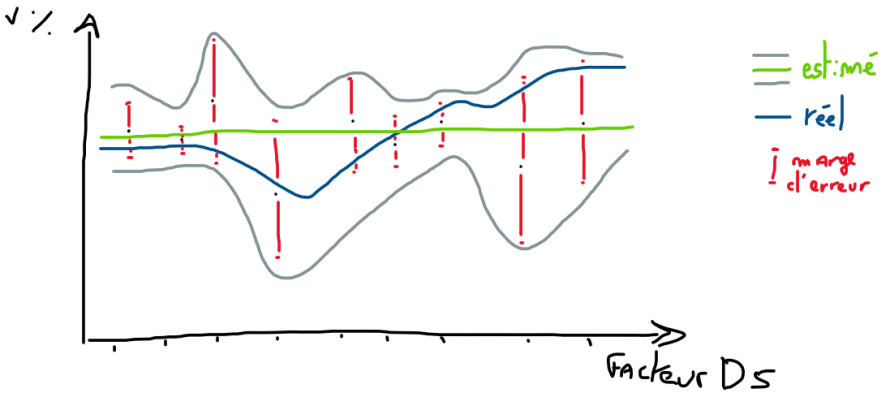
**Tester de façon uniforme notre espace de recherche** Afin de pouvoir avoir des statistiques les moins biaisées possible, il est nécessaire d'uniformiser nos simulations sur notre espace de recherche afin d'éviter la sous-représentation de certains couples de paramètres initiaux.

## 2.2. Problème




**Comment maximiser la précision statistique d'un couple de paramètre unique ?** Pour déterminer le poucentage de victoires d'un certain couple de paramètres initiaux, nous allons avoir besoin de réaliser un certain nombre de simulations.

Cependant, quelques tentatives ne serait potentiellement pas représentatif du pourcentage de victoire, un peu comme 3 lancers de pièces ne font pas 50-50 % de chances d'avoir pile ou face. Il nous faudrais donc répeter notre simulation un maximum de fois pour déterminer l'incertitude statistique de notre mesure. Mais combien de fois ?



**Comment éviter des erreurs d'estimations statistiques ?** Avec des données mal réparties, nous pourrions avoir des soucis d'estimations. Le graphe représente ci-dessus un exemple de mauvaise représentation potentielle, dûe à la différence de marge d'erreur.

Des données avec les memes marges d'erreurs auraient pu potentiellement au moins retrouver le premier creux en essayant de coller un maximum les points. Dans le cas illustré, le calcul pourrais avoir considéré le point haut en incertitude du creux comme une anomalie comparés aux autres points relativement alignés, résultant en une estimation faussée de la forme de nos données.

Évidemment plus de données est toujours mieux pour réduire la marge d'erreur générale de notre estimation, mais comment éviter au moins un maximum cette déformation ?

> Comment pourrions nous maximiser la précision de nos analyses et la lisibilité de nos résultats ?

## 2.3. Approches possibles

**Génération aléatoire et uniforme de paramètres initaux** Nous pourrions tirer parti de l'aléatoire pour générer de façon aléatoire mais uniformément des couples de paramètres initiaux.

Cela aurait le mérite de pouvoir avoir une image globalement représentative de notre phénomène avec de moins en moins de déformations dûes à l'aléatoire à mesure que nous multiplions le nombre de tirage au sort de paramètres.

Le souci avec cette approche est que nous pourrions potentiellement subir les aléas d'un générateur pseudo aléatoire pas réellement uniforme qui pourrait biaiser nos résultats, et que selon notre espace de recherches, il serait necessaire d'avoir beaucoup de tirages au sorts pour s'assurer de la précision sur certaines données.

**Génération complète des points de l'espace de recherche** L'approche inverse serait de générer exactement toutes les combinaisons possibles de paramètres initiaux de notre espace de recherche, et de les répeter un nombre suffisant de fois pour satisfaire le niveau de précision voulu sur chacun de ces points.

La précision de cette approche serait alors directement liée au nombre d'iterations par combinaisons mais aussi potentiellement plus gourmande en simulations que l'approche aléatoire.

## 2.4. Approche utilisée

### Exploration complète d'un espace de recherche voulu

> Nous avons préféré partir sur un simulateur parcourant l'intégralité de notre espace de recherche pour minimiser les biais et aléas d'un générateur aléatoire et ainsi maximiser la précision de nos résultats.

La grande quantité de simulations combinée à la vitesse de calcul du déroulement d'une partie peuvent vite faire durer le processus de génération de données sur plusieurs minutes à plusieurs heures selon l'espace de recherche, mais les données en résultant sont les plus fidèles que nous pourrions avoir en un minimum de temps de génération.

## 2.5. Remarques sur les résultats obtenus

```
Total amount of simulations to do: 657500
highest map M size: 50          sampled every 5
highest map N size: 50          sampled every 5
highest Coalition size: 40      sampled every 5
highest solo research level: 10      sampled every 2
highest coalition research level: 9  sampled every 2


Time estimated 0d 20h 46m 17s |---------------------------------------------| 8.3%     @ 0.124s/game   curr (45,35,21,8,1)    sim#54268
```

**Les performances du modèle de simulation sont critiques** La grande quantité de simulations nécessaire pour évaluer un espace de recherche à 5 dimensions sur de petits intervalles à une précision convenable rendent le temps d'execution des simulations cruciales pour générer nos données en un temps raisonnable.

```
   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.000    0.000    0.204    0.204 <string>:1(<module>)
        2    0.000    0.000    0.000    0.000 abc.py:137(__instancecheck__)
178122/6896    0.132    0.000    0.177    0.000 model.py:116(propagate_case_search)
      719    0.000    0.000    0.000    0.000 model.py:131(has_ended)
```

Notre langage de départ étant python, nous avons optimisé notre vitesse d'exécution à l'aide du trancompilateur Cython qui permet de génerer du code C à partir de code source python.

Pour accélerer encore plus nous avons tiré parti de la capacité de Python à intégrer du typage statique via les annotations pour indiquer à Cython les types des variables et le laisser optimiser encore plus profondément les algorithmes C utilisés, en plus d'avoir des indications plus complètes et lisibles pour la documentation en bonus.

```
   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.085    0.085    0.085    0.085 <string>:1(<module>)
        2    0.000    0.000    0.000    0.000 abc.py:137(__instancecheck__)
       41    0.000    0.000    0.000    0.000 random.py:224(_randbelow)
        1    0.000    0.000    0.000    0.000 random.py:264(shuffle)
        1    0.000    0.000    0.000    0.000 random.py:286(sample)
```

**Les données sont bel et bien réparties de façon uniforme** Grâce à cette approche, nous pouvons bel et bien voir l'uniformité de nos tests sur les paramètres initiaux, la taille maximale de la coalition étant considérée variable selon la taille du plateau, il est cependant normal de voir une densité plus forte de tests plus M et N grandissent, conformément à la taille supérieure de l'intervalles de valeurs C à tester sur ces dimensions d'arène.

Mais même cette augmentation de densité est uniforme. Nous pouvons retrouver ce genre d'informations sur les graphes de densités de nos analyses.

## 2.6. Pistes d'amélioration

**Simulations en parallèle** Nous avons tenté de faire simuler de multiples simulations en parallèle pour pouvoir profiter des multiples coeurs de nos sytèmes de calculs, mais notre Cython as malheureusement souffert de l'overhead python de la librairie multiprocessing et l'éxecution s'est révélée plus lente que sans.

Une implémentation du multiprocessing directement en C ou via une libraire pyhton déjà optimisée pour Cython devrait permettre d'accélérer grandement les calculs de simulation en parallèlisant la charge de calcul sur autant de coeurs que possibles.

# 3. Modèle de jeu

## 3.1. Nécessités

**but général de l'element** Genre ouais on as besoin d'un moyen de calculer efficacement et rapidement des données de simulation de bonbon qui se font manger pour le projet.

## 3.2. Problème

**probleme 1** blabla bla vitesse. besoin rapidité mais consomme truc genre memoire.

**probleme 2** blabla besoin d economiser de la memoire ou un truc mais contraintes de vitesse sinon.

> PROBLEMATIQUE QUI COMBINE ET RESUME LES PROBLEMES SUR COMMENT BIEN ALLIER LES DEUX DANS NOTRE CAS.

## 3.3. Approches possibles

**Approche 1** résumé de ce que c'est, pour et contre, difference avec les autres solutions

**Approche 1** résumé de ce que c'est, pour et contre, difference avec les autres solutions

## 3.4. Approche utilisée

**Approche finalement choisie** Les avantages qui ont fait prendre cette décision résumés et plus de details sur les bons cotés de cette décision

> RESUME DE LA DECISION PRISE

## 3.5. Remarques sur les résultats obtenus

**probleme X** C'etait pas tip top au final comme solution, on as eu un probleme avec X, ce n'etait pas le meilleur des choix et avons rencontré des difficultés que l'on explique rapidement ici.

## 3.6. Pistes d'amélioration

**meilleures approches**

**potentielle optimisation**

**potentiel polissage**

# 4. Intelligence Artificielle - Heuristique

## 4.1. Nécessités

**but général de l'element** Genre ouais on as besoin d'un moyen de calculer efficacement et rapidement des données de simulation de bonbon qui se font manger pour le projet.

## 4.2. Problème

**probleme 1** blabla bla vitesse. besoin rapidité mais consomme truc genre memoire.

**probleme 2** blabla besoin d economiser de la memoire ou un truc mais contraintes de vitesse sinon.

> PROBLEMATIQUE QUI COMBINE ET RESUME LES PROBLEMES SUR COMMENT BIEN ALLIER LES DEUX DANS NOTRE CAS.

## 4.3. Approches possibles

**Approche 1** résumé de ce que c'est, pour et contre, difference avec les autres solutions

**Approche 1** résumé de ce que c'est, pour et contre, difference avec les autres solutions

## 4.4. Approche utilisée

**Approche finalement choisie** Les avantages qui ont fait prendre cette décision résumés et plus de details sur les bons cotés de cette décision

> RESUME DE LA DECISION PRISE

## 4.5. Remarques sur les résultats obtenus

**probleme X** C'etait pas tip top au final comme solution, on as eu un probleme avec X, ce n'etait pas le meilleur des choix et avons rencontré des difficultés que l'on explique rapidement ici.

## 4.6. Pistes d'amélioration

**meilleures approches**

**potentielle optimisation**

**potentiel polissage**

# 5. Intelligence Artificielle - Optimisations

## 5.1. Nécessités

**but général de l'element** Genre ouais on as besoin d'un moyen de calculer efficacement et rapidement des données de simulation de bonbon qui se font manger pour le projet.

## 5.2. Problème

**probleme 1** blabla bla vitesse. besoin rapidité mais consomme truc genre memoire.

**probleme 2** blabla besoin d economiser de la memoire ou un truc mais contraintes de vitesse sinon.

> PROBLEMATIQUE QUI COMBINE ET RESUME LES PROBLEMES SUR COMMENT BIEN ALLIER LES DEUX DANS NOTRE CAS.

## 5.3. Approches possibles

**Approche 1** résumé de ce que c'est, pour et contre, difference avec les autres solutions

**Approche 1** résumé de ce que c'est, pour et contre, difference avec les autres solutions

## 5.4. Approche utilisée

**Approche finalement choisie** Les avantages qui ont fait prendre cette décision résumés et plus de details sur les bons cotés de cette décision

> RESUME DE LA DECISION PRISE

## 5.5. Remarques sur les résultats obtenus

**probleme X** C'etait pas tip top au final comme solution, on as eu un probleme avec X, ce n'etait pas le meilleur des choix et avons rencontré des difficultés que l'on explique rapidement ici.

## 5.6. Pistes d'amélioration

**meilleures approches**

**potentielle optimisation**

**potentiel polissage**

# 6. Analyse - Exploration

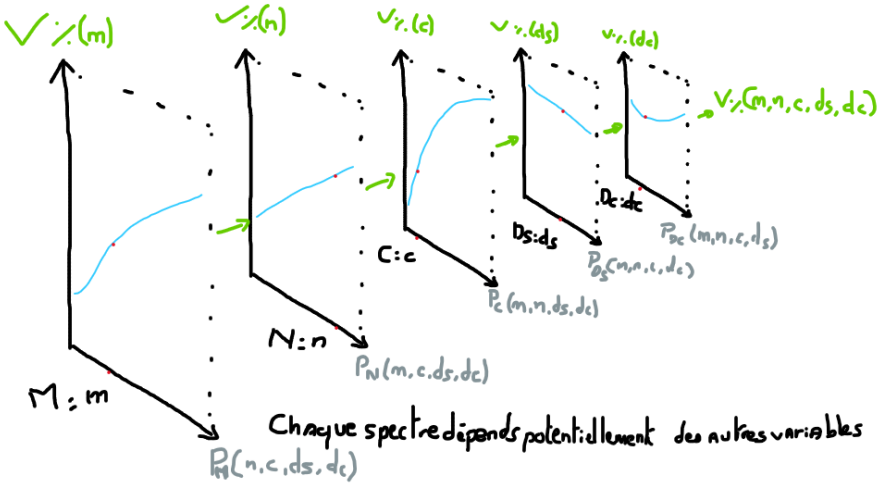
## 6.1. Nécessités

**Determiner les facteurs d'une victoire** L'objectif final de ce projet est de déterminer les meilleurs paramètres initiaux permettant de maximiser le taux de victoires de la coalition.

Cela étant dit, notre objectif pour y parvenir est d'utiliser l'outil de l'analyse statistique, mais sur les données d'un demi-million de parties potentielles avec une demi douzaine de facteurs différents, par où commencer ?

## 6.2. Problème

**Comment déterminer les bonnes correlations ?** Faire des statistiques à partir de l'intégralité de nos données permet de déterminer des tendances générales, mais sur notre cas où nous avons 5 dimensions de données indépendantes, et donc potentiellement des variations de ces correlations à chaque modification infime de n'importe quel facteur, comment pourrions nous analyser relativement efficacement l'évolution de ces tendances pour tenter de déterminer de potentielles correlations cachées entre plusieurs variables ?



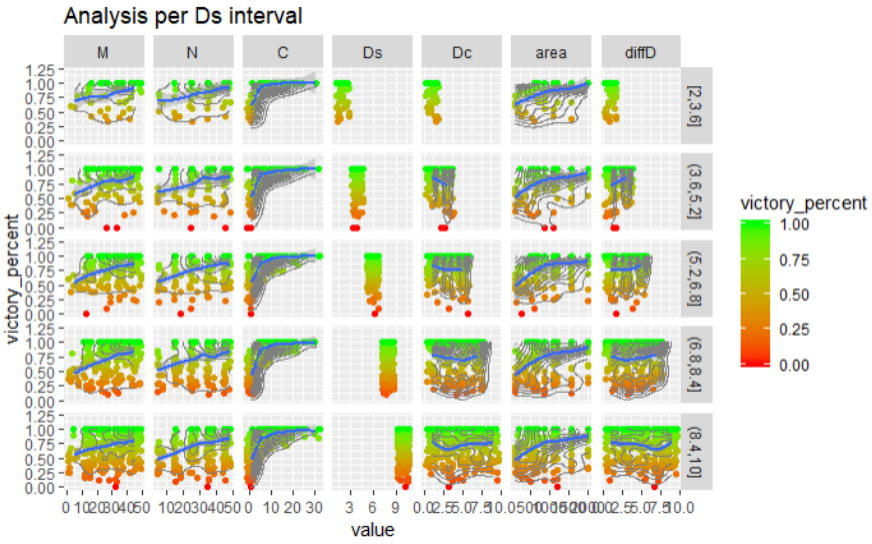

> Comment pourrions nous analyser nos données pour pouvoir y déceler des informations de la façon la plus efficace et complète possible sur autant d'axes ?

## 6.3. Approche utilisée

**Analyse par tranches** La quantité inconnue de données que nous avons pour chaque variable, construire une matrice à 5 dimensions serait prohibitif pour nos moyens actuels, autant en espace mémoire qu'en temps de calcul, d'analyse et de génération. C'est pour

cette raison que nous avons opté pour une simple analyse par intervalles de données présentes.

> Analyser les tendances de victoire en fixant une variable ou plusieurs variables à la fois et en scindant nos données en 5 intervalles de tailles équivalentes nous permet d'analyser la progression des spectres entre de grandes variations des variables en questions et d'avoir une idée générale des relations entre variables.




## 6.4. Remarques sur les résultats obtenus

**Les données sont parlantes** Voir progresser les intervalles de données disponibles en fonction des intervalles de chaque variable et les voir se chevaucher petit à petit permet vraiment d'avoir une meilleure idée de ce que représente chaque intervalle dans la totalité des données présentes. De plus, la comparaison aisée entre les différents spectres à différents intervalles montrent bel et bien si les spectres changent beaucoup ou non selon tel ou tel intervalle d'une variable et permet de déterminer l'influence de cette variable sur ces spectres ou non.

## 6.5. Pistes d'amélioration

**Génération d'un profil 5D voire n-D de probabilités !** Comme dit plus haut, à partir de ce genre de données il paraitraît tout naturel de tenter de modéliser un spectre 5D de probabilités permettant de déterminer automatiquement la probabilité de victoire d'un couple de paramètres initiaux arbitraires.

Cela pourrait d'ailleurs être un sujet bien chargé très intéressant à implémenter et à travailler qui pourrait permettre de s'intéresser à des sujets peu communs comme les tenseurs et l'interpolation !

## Cinquième partie

# Analyse des données générées

# 7. Analyse des données de l'IA

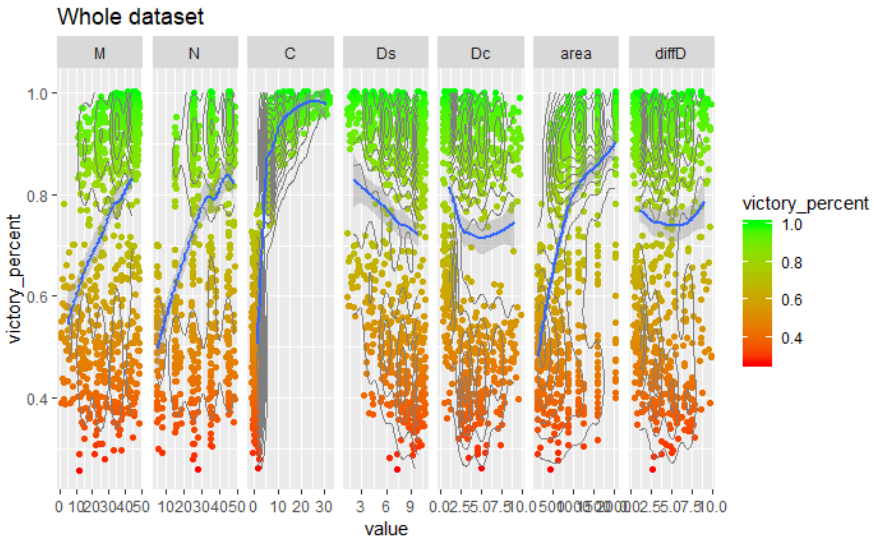
# 8. Analyse des données du modèle

**Paramètres initiaux** Cette simulation as été réalisée avec 100 réitérations pour chaque combinaison possible avec les paramètres par défaut suivants.

```
Total amount of simulations to do: 85500
highest map M size: 50          sampled every 10
highest map N size: 50          sampled every 10
highest Coalition size: 32      sampled every 5
highest solo research level: 10     sampled every 2
highest coalition research level: 9  sampled every 2

Time estimated 0d 0h 0m 0s |████████████████████████████████████████| 100.0%  @ 0.173s/game  curr (25,25,6,8,7)    sim#85500
```

## 8.1. Analyse d'ensemble

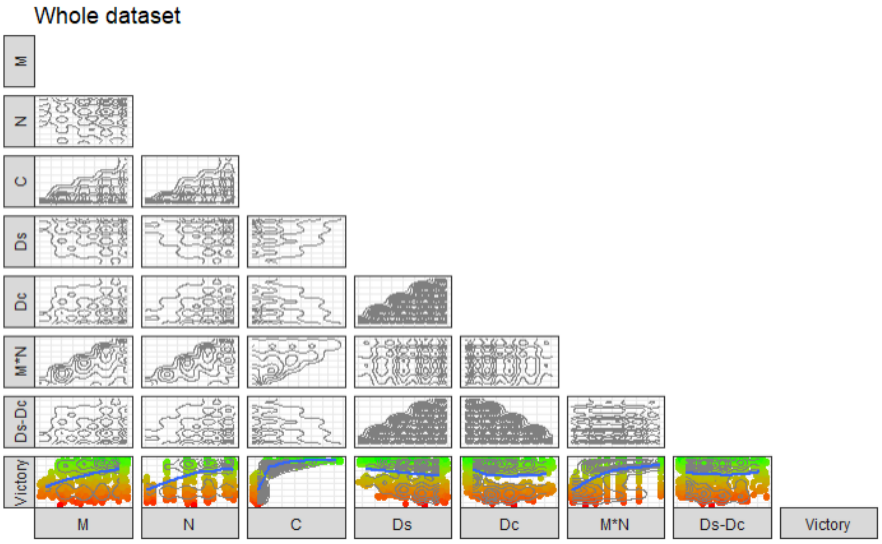



**Quelques facteurs sortent du lot** À commencer par le facteur C qui est le seul avec une fonction qui tends vers 1 si rapidement, les mesures d'espace M, N et M*N semblent avoir aussi une corrélation positive avec le pourcentage de victoire.

La tendance de Dc à réduire le pourcentage de victoire avant un certain point est aussi étonnante, on pourrais penser que plus de prévoyance de la part des joueurs en groupe puissent les avantager, mais on dirais qu'il y a un certain équilibre entre prévoyance et hasard qu'il leur faut conserver pour être efficace. On retrouve la même tendance étonnante sur la courbe de différence d'intelligence, où réduire la différence au minimum comme de la maximiser semble porter les meme résultats, tandis qu'une différence entre les deux semble être le pire choix possible.

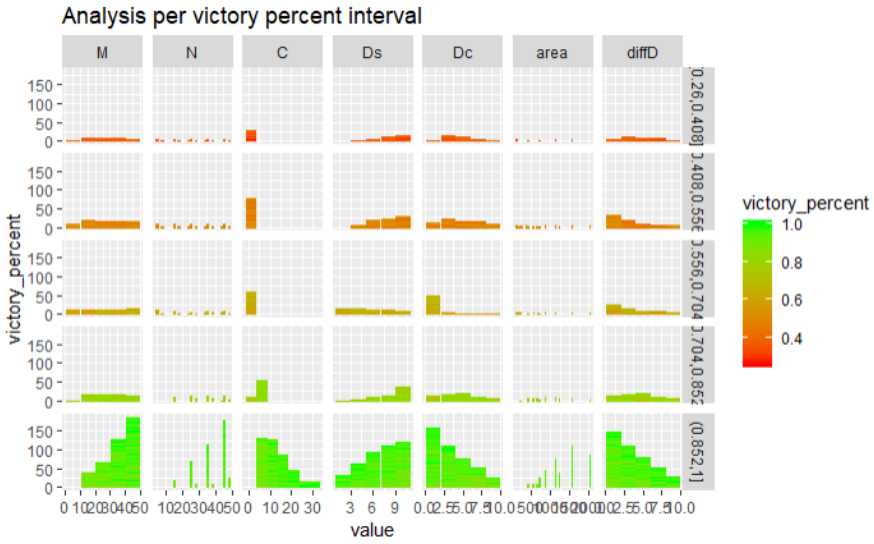
Le facteur ds semble coller à nos attentes en revanche, plus le joueur solo est intelligent, plus les chances de gagner semblent diminuer en général.

Ceci étant dit, les courbes de densité semblent mettre en évidence une certaine division sur tous les autres graphes que le C, une partie basse éparse et large, et une partie haute dense et étroitement proche du 1.




## 8.2. Analyses détaillées

### 8.2.1. Répartitions des pourcentages de victoires

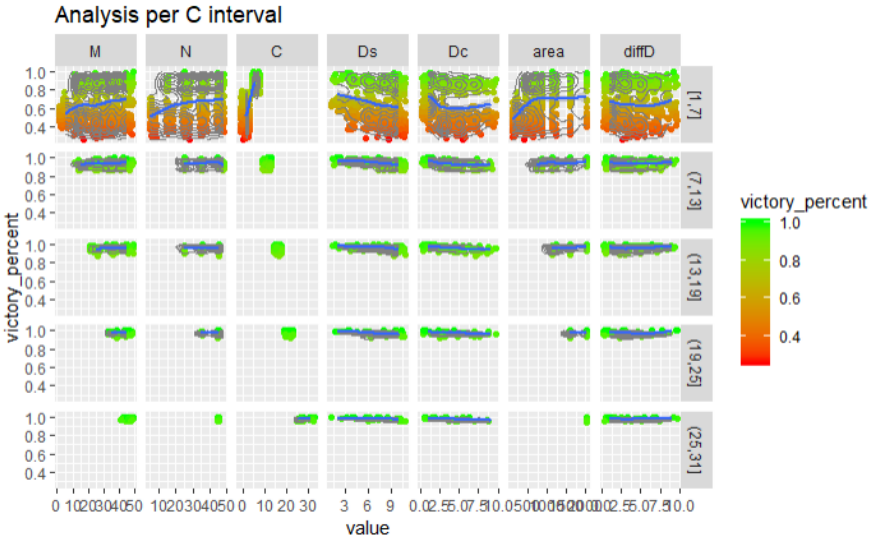



**Une différence clairement marquée** Les axes en Y étant fixes entre les différents intervalles de victoires, nous pouvons bel et bien constater sans déformation plusieurs ensembles de données.

Les parties à 80-% de chances de victoire ont une répartition relativement homogème sur les differents spectres excepté sur le facteur C, qui indique très clairement que toutes les valeurs de petit C sont en deçà de 80%.

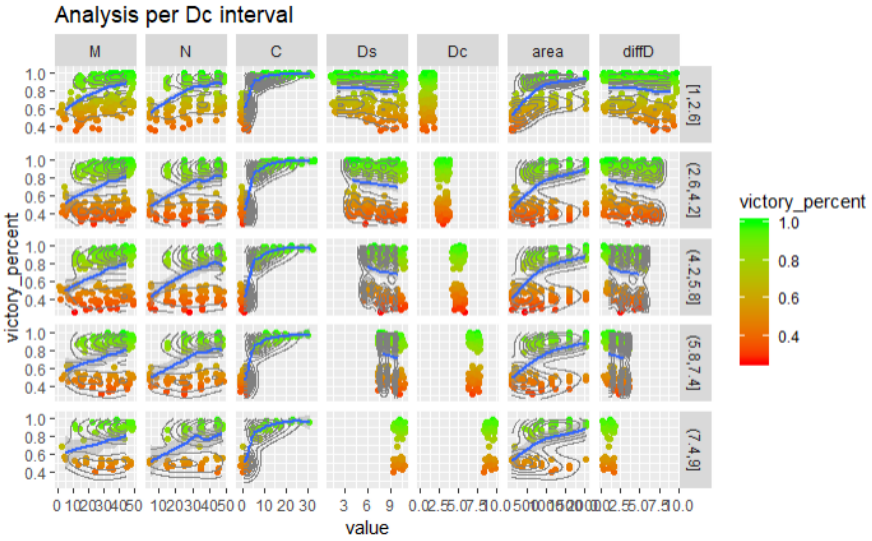
Cette majorité se fait aussi remarquer sur tous les autres facteurs, meme si les groupes de données y sont marqués de manière moins évidente. La partie basse des résultats est bel et bien éparse dès les 20%, et la quantité de combinaisons de données à 0 voire 60% est preque comparable à celle de la partie haute. On assiste donc bel et bien à deux groupes de données où la majorité des points les plus mauvais sont liés aux faibles valeurs de C, en opposition à la majorité de points où C n'est plus minime.

### 8.2.2. Découpage du spectre selon des intervalles de C




**C est définitivement un facteur MAJEUR** Cette représentation montre bien l'écrasante influence de C sur le pourcentage de victoire, aussitôt une valeur (minime en plus) atteinte, les chances de victoires enregistrées sont quasiment assurées. Autrement nous retrouvons les fonctions du départ mais plus proche de la neutralité qu'au-dessus. Voir parfois les fonctions semblent stagner à ces faibles valeurs de C.

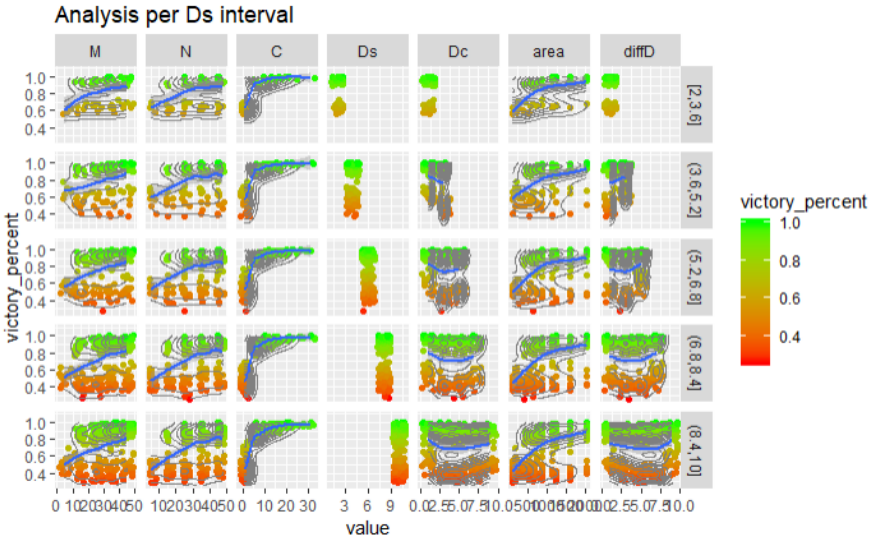
### 8.2.3. Découpage du spectre selon des intervalles de Dc




**Aucun impact réel décelé** Dc ne semble vraiment pas influer sur les différents spectres, par conséquent, Dc ne semble pas avoir d'impact tout court sur notre distribution de probabilité.

Ceci dit, dans ces graphes les deux clusters de points sont particulièrement visibles.

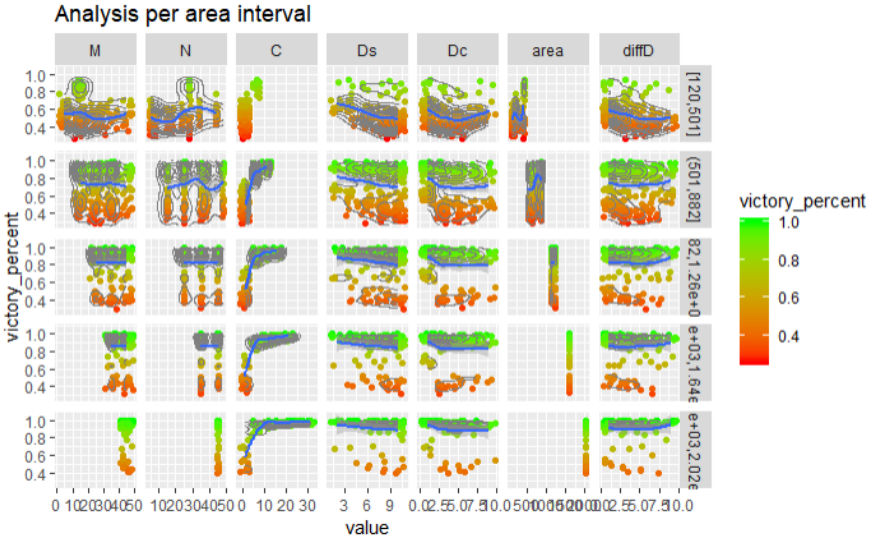
### 8.2.4. Découpage du spectre selon des intervalles de Ds




**Aucun impact réel décelé non plus** Ds ne semble vraiment pas influer sur les différents spectres non plus, par conséquent, Ds ne semble pas avoir d'impact tout court sur notre distribution de probabilité.

Ceci dit, nous avons encore une fois un clivage visible entre deux groupes de données.

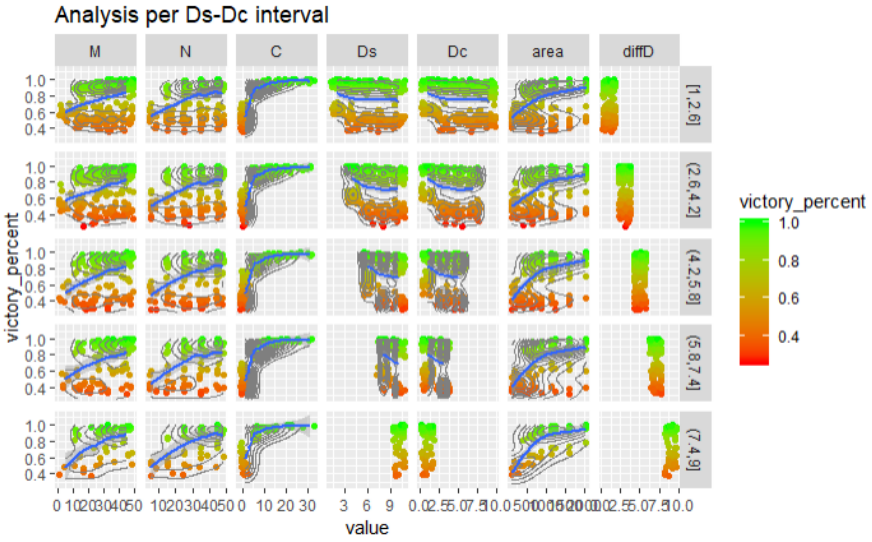
### 8.2.5. Découpage du spectre selon des intervalles de l'aire M*N




**Une tendance positive !** Plus la zone d'aire augmente, plus on peut voir les points et les fonctions se décaler vers la victoire. Sur M on peut même voir petit à petit les points les plus bas disparaitre au fur et à mesure, comme pour Ds et Dc.

> Cela peut s'expliquer par la correlation suivante : Plus l'aire est grande, plus la taille de la coalition peut l'être aussi, résultant naturellement vers de meilleures chances de victoire. Car plus de joueurs implique aussi plus de contrôle global du plateau.

### 8.2.6. Découpage du spectre selon des intervalles de la différence de niveau



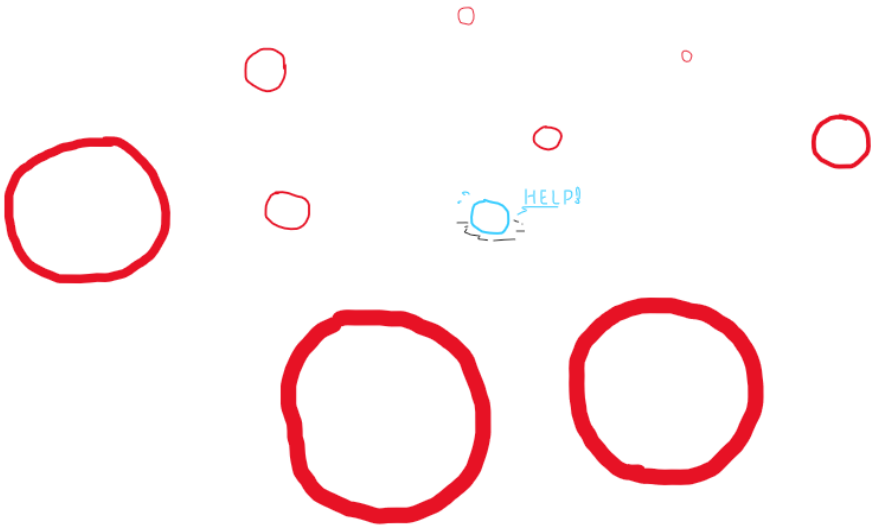

**Toujours aucun impact réel décelé** La différence d'intelligence ne semble pas non plus influer particulièrement sur les chances de victoires des autres variables.

### 8.2.7. Conclusions d'analyse

**Le contrôle de la carte est le facteur le plus important d'après notre modèle** La seule véritable corrélation que nous ayons pu déceler ici entre variables et pourcentages de victoire est en fonction du nombre de joueurs présents sur la carte.

Si nous y réfléchissons bien, cela fait même sens !

> Même avec la meilleure stratégie possible, un joueur seul ne peut pas empêcher des dizaines de joueurs de jouer contre lui à la fois et venant de toutes directions.



## Sixième partie

# Problèmes, tests et expérimentations

# 9. Problèmes rencontrés

## 9.1. Problème majeur 1

## 9.2. Problème majeur 2

## 9.3. Problème mineur 1

## 9.4. Problème mineur 2

# 10. Expérimentations

## 10.1. Expérimentation 1

## 10.2. Expérimentation 2

# 11. Conclusion

**Résumé des objectifs au résultat final**

**Résumé du résultat final comparé aux résultats escomptés**

**Résumé des liens avec les connaissances et compétences universitatires**

**Résumé sur l'enrichissement personnel**

**Résumé difficultés rencontrées**

**Résumé perspectives envisagées, appréciation perso, poursuite..**

# Septième partie

# Annexes

# 12. Analyse - Simulation

## 12.1. Nécessités

**Tenter de modéliser un système à priori intelligent** Les calculs d'heuristique sont chers, et il est parfois utile pour optimiser la prédiction ou la detection d'anomalies de tenter de trouver un modèle mathématique qui décris bien l'évolution de notre système.

Mais si les calculs d'heuristiques sont chers, ils sont malheureusement au premiers abord nécessaires, pour pouvoir observer des phénomènes dans leur globalité avant de tout analyser. Pour tenter de trouver un modèle il va donc falloir beaucoup de données pour trouver de potentielles corrélations reccurentes.

## 12.2. Problème

**À quel point telles données sont elles pertinentes ?** Afin de pouvoir affiner notre modèle et le faire ressembler un maximum au comportement de notre véritable phénomène, il est important de déterminer l'importance de chaque facteur dans son comportement.

Comment déterminer par la suite l'équivalence de cette importance dans notre autre visualisation du problème ? Sont-elles comparables ?

**Quels types de modèles pouvont nous créer ?** Il existe une infinité de fonctions possibles, correspondant à notre profil recherché sur notre intervalle de données. Si notre modèle doit être capable d'extrapoler, l'intuition et les analogies au monde physique peuvent-ils être suffisants ?

La réponse à cette question est évidemment complexe. Mais aussi évidente : On ne peut pas prédire avec exactitude quelque chose sur laquelle nous n'avons aucune donnée. Par conséquent, il va falloir partir du principe que les données suivantes ressembleront à une tendance connue qui corresponds déjà aux données présentes.

Mais comment déterminer laquelle ?

> Comment pourrions nous déterminer un type de modèle et ses paramètres pour représenter le même comportement que notre phénomène connu ?

## 12.3. Approches possibles

**Mathématiques pures** Avec des mathématiques pures, et à partir de suffisamment de données d'entrées, nous devrions pouvoir construire un spectre de probabilités de victoire en fonction des paramètres initiaux.

Cela devrait permettre une prédiction des plus fidèles de notre comportement sur l'intervalle connu, mais l'extrapolation sur un spectre de probabilités nous semble être d'un très haut niveau de mathématique que nous n'avons malheureusement pas dans notre équipe.

La prédiction en données connues serait donc instantanée mais l'extrapolation impossible.

**Modèle simulé inspiré par la physique** Avec de la physique il est possible de tenter de raisonner différemment, et de calculer potentiellement plus rapidement le meme genre de résultats que le calcul complet de notre phénomène de départ.

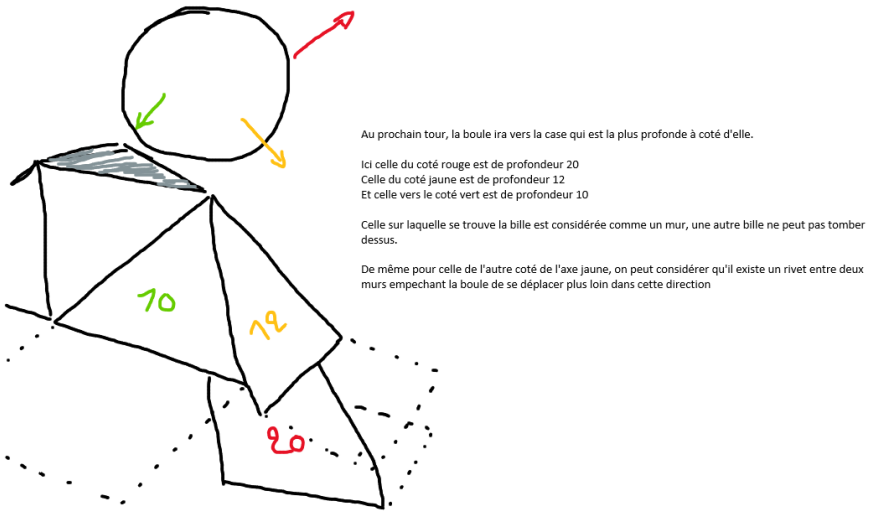
De plus, cela permettrais aussi potentiellement de mettre en équation le comportement des acteurs de notre phénomène, ici les IA, et de potentiellement trouver un modèle simplifié et fonctionnel pour leur heuristique.

Ici les équations pourraient permettre de l'extrapolation relativement aisément, mais la précision de notre modèle va nécessiter un lourd travail de réglages pour trouver l'équilibre entre l'influence des paramètres dans le modèle des IA et celle sur les paramètres dans le modèle simulé.

Un exemple parlant est la traduction de l'influence de la profondeur de recherche des IA vers le modèle physique, où il n'y aura pas d'IA.

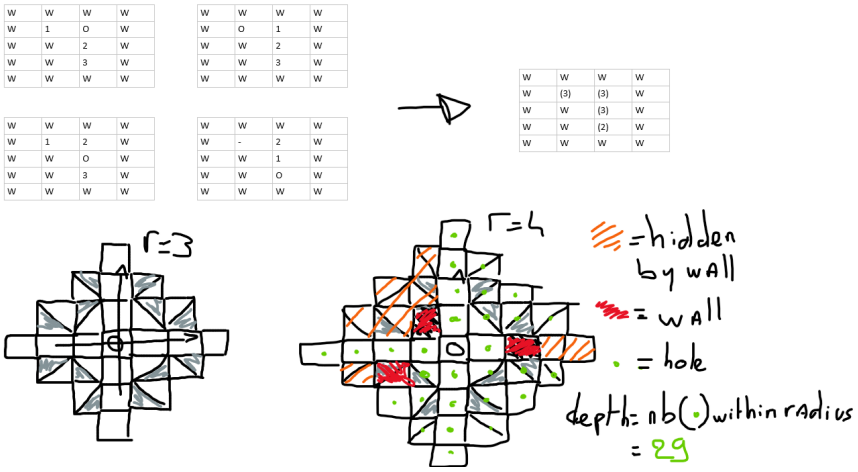
## 12.4. Approche utilisée

**Modèle physique** Le temps et l'expérience limitée des membres du groupe dans le domaine de la simulation nous ont forcés à partir pour le modèle inspiré de la physique.



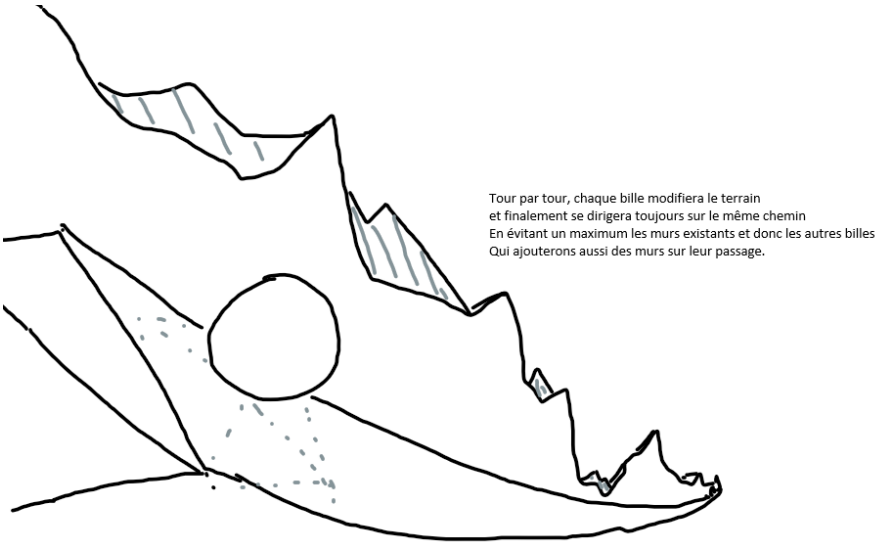

Nous avons donc pris les joueurs pour des billes (littéralement), et les considérons désormais en roulis perpétuel vers la pente la plus accentuée qui s'offre à eux.

À la suite de leur roulis (changement de case), un mur se crée à leur position actuelle, les forcant à continuer de rouler au tour suivant.




Ces murs sont de matériau friable, comme du sable, et une fois posés, remplissent les trous environnants d'une légère quantité de sable sans jamais les boucher, réduisant ainsi leur pente.

Une bille ne peut plus rouler si elle est entourée de murs, autrement dit, si elle n'as plus de pente sur laquelle rouler. Elle est alors retirée du jeu et considérée hors jeu.




Ce système devrais favoriser le contrôle de la carte de façon naturelle, car les billes seront donc naturellement inclines à se diriger vers les endroits les plus profonds, c.à.d. ceux ayant le plus d espace libre, et continueront toujours de rouler tant qu'elles n'auront pas atteint de cul de sac.

Pour tenter de donner une dimension d'équipe, nous avons attribués une légère force attirant la coalition vers la position du joueur pour départager deux cases de même profondeur.

Un barycentre de force pourrait être nécessaire pour calculer la force inverse pour le joueur seul, et cela nous as semblé potentiellement trop coûteux en temps de calculs supplémentaires pour rendre le modèle potentiellement viable. L'idée reste à tester.

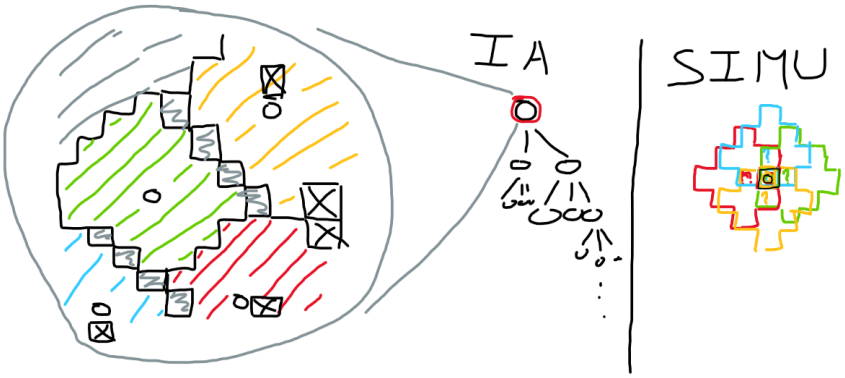
## 12.5. Remarques sur les résultats obtenus

**Un semblant de stratégie !** Nous avons pu constater que malgré l'absence de prédiction de la part des billes, celles ci semblaient avoir parfois des semblants de stratégies, par exemple, nous avons pu surprendre le joueur solo à coincer un adversaire dans un couloir de deux cases puis lui faire une queue de poisson en sortie !

Même sans prédiction, nous arrivons donc à recalculer des mouvements potentiellement stratégiques, il y a donc espoir de pouvoir affiner notre modèle plus encore.

**Un optimisation de temps de calcul efficace !** Là où notre IA est forcée de calculer la zone de controle d'un joueur jusque $3^C * Ds$ fois dans le pire des cas (tous les joueurs peuvent aller dans 3 directions à chaque tour, sur Ds tours), sur l'intégralité de la carte importante pour chaque joueur simulé à chaques simulation, notre système se contente d'un rayon de recherche de profondeur qui calcule en une fois l'intérêt d'une case et de ses alentours, et ce sur les quatre cases adjacentes à la bille.

La calcul de la zone ne calcule aucune case inutile et donc s'arrête soit en cas de rayon atteint, soit en cas d'obstacle atteint dans sa propagation de comptage, ce calcul peut donc etre de complexité O(1) en cas d'encerclement et sa moyenne réduit à mesure que le nombre de murs augmente.




A contrario, la recherche pour notre IA nécessite de calculer et propager autant de zones que de joueurs, incluant donc une plus grande surface à propager pour le calcul des zones d'un seul joueur, lors d'une unique étape de simulation.

## 12.6. Ordre de grandeur de la différence de vitesse de calcul

| M | N | C | Ds | Dc |
|---|---|---|---|---|
| 50 | 50 | 36 | 10 | 9 |

FIGURE 12.1. – Paramètres initiaux

**Des paramètres initiaux extrêmes** Cette partie est la partie la plus extrême de notre échantillon de données sortant de notre simulation modélisée, en sachant que celles ci ont toutes été répétées 100 fois pour avoir une certaine précision. Nous allons ici tenter d'estimer à grand renfort d'approximations la quantité de cases touchées par nos calculs pour la décision de nos joueurs afin de pouvoir comparer la différence d'efficacité entre nos deux modèles.

**Les constantes** Nous déterminons d'abord les constantes qui nous serviront à simplifier nos équations :

— $D$ représente la profondeur moyenne de recherche des joueurs

— $t_{max}$ représente le nombre maximum de tours pouvant être joués si personne ne meurt tout le long de la partie.

$$D = \frac{C * Dc + 1 * Ds}{C + 1} \tag{12.1}$$
$$\approx 9 \tag{12.2}$$
$$t_{max} = \frac{M * N}{C + 1} \tag{12.3}$$
$$\approx 68 \tag{12.4}$$

### 12.6.1. Modèle IA

$$nbCases(t, M, N, C) = M * N - (C + 1) * t \tag{12.5}$$
$$nbCases(t) = 2500 - 37 * t \tag{12.6}$$
$$nbCases/tour/joueur(t, D) = \int_t^{t+D} nbCases(x)dx \tag{12.7}$$
$$= [2500 * t - 37 * \frac{t^2}{2}]_t^{t+D} \tag{12.8}$$
$$= 2500(t + D - t) - 37 * \frac{(t + D - t)^2}{2} \tag{12.9}$$
$$nbCases/tour/joueur(D) = 2500 * D - \frac{37}{2} * D^2 \tag{12.10}$$
$$totalCases(t_{max}, C, D) = (C + 1) * t_{max} * nbCases/tour/joueur(D) \tag{12.11}$$
$$= (C + 1) * t_{max} * (2500 * D - \frac{37}{2} * D^2) \tag{12.12}$$
$$= 37 * 68 * (2500 * 9 - \frac{37}{2} * 9^2) \tag{12.13}$$
$$= 2516 * (22500 - 18.5 * 9^2) \tag{12.14}$$
$$= 2516 * (22500 - 1498.5) \tag{12.15}$$
$$= 2516 * 21001.5 \tag{12.16}$$
$$totalCases(t_{max}, C, D) \approx 52839774 \tag{12.17}$$

### 12.6.2. Modèle simulé

$$nbCasesInRadius(D) = \int_0^D 4 * x dx \tag{12.18}$$

$$= [4\frac{x^2}{2}]_0^D \tag{12.19}$$

$$nbCasesInRadius(D) = 2D^2 \tag{12.20}$$

$$totalNbMurs(t, C) = t * (C + 1) \tag{12.21}$$

$$ratioMur/case(t, M, N, C) = \frac{totalNbMurs}{M * N} \tag{12.22}$$

$$ratioMur/case(t, M, N, C) = \frac{t * (C + 1)}{M * N} \tag{12.23}$$

$$nbMurInRadius(t, M, N, C, D) = ratioMur/case(t, M, N, C) * nbCasesInRadius(D) \tag{12.24}$$

$$nbMurInRadius(t, M, N, C, D) = \frac{t * (C + 1)}{M * N} * 2D^2 \tag{12.25}$$

$$casesLibreInRadius(t, M, N, C, D) = nbCasesInRadius(D) - nbMurInRadius(t, M, N, C, D) \tag{12.26}$$

$$casesLibreInRadius(t, M, N, C, D) = 2D^2 - \frac{t * (C + 1)}{M * N} * 2D^2 \tag{12.27}$$

$$casesLibreInRadius(t, M, N, C, D) = 2D^2 * (1 - \frac{t * (C + 1)}{M * N}) \tag{12.28}$$

$$nbCases/player/turn(t, M, N, C, D) = 3 * casesLibreInRadius(t, M, N, C, D) \tag{12.29}$$

$$= 3 * 2D^2 * (1 - \frac{t * (C + 1)}{M * N}) \tag{12.30}$$

$$nbCases/player/turn(t, M, N, C, D) = 6D^2 * (1 - \frac{t * (C + 1)}{M * N}) \tag{12.31}$$

$$nbCases/turn(t, M, N, C, D) = (C + 1) * nbCases/player/turn(t, M, N, C, D) \tag{12.32}$$

$$nbCases/turn(t, M, N, C, D) = (C + 1) * 6D^2 * (1 - \frac{t * (C + 1)}{M * N}) \tag{12.33}$$

$$totalCases(t_{max}, M, N, C, D) = \int_0^{t_{max}} nbCases/turn(t, M, N, C, D)dt \tag{12.34}$$

$$= (C + 1) * 6D^2 * \int_0^{t_{max}} 1 - \frac{t * (C + 1)}{M * N} dt \tag{12.35}$$

$$= (C + 1) * 6D^2 * [t - \frac{(C + 1)}{M * N} * \frac{t^2}{2}]_0^{t_{max}} \tag{12.36}$$

$$= (C + 1) * 6D^2 * (t_{max} - \frac{(C + 1) * t_{max}^2}{2 * M * N}) \tag{12.37}$$

$$= 37 * 486 * (68 - \frac{37 * 68^2}{5000}) \tag{12.38}$$

$$= 17982 * (68 - 34) \tag{12.39}$$

$$totalCases(t_{max}, M, N, C, D) = 611388 \tag{12.40}$$

### 12.6.3. Comparaison des deux modèles

**Nous pouvons calculer notre speedup** Grâce aux résultats ci dessus, nous pouvons calculer le speed up entre nos deux modèles.

$$\frac{totalCases_{IA}}{totalCases_{simu}} = \frac{52839774}{611388} \tag{12.41}$$

$$\frac{totalCases_{IA}}{totalCases_{simu}} \approx 86 \tag{12.42}$$

> Pour ce cas extrême, notre simulation teste 86x moins de cases que notre IA de base. C'est énorme.

## 12.7. Pistes d'amélioration

**Un meilleur réglage des facteurs du modèle** Pour le moment nous avons considéré les facteurs de notre modèle relativement linéairement à partir des paramètres initiaux mais l'IA peut potentiellement réagir à d'autres facteurs que nous pourrions peut etre quantifier pour améliorer notre modèle.