

Projet Intelligence Artificielle

IA pour « Pokémon Stadium 2[Rental] »

Année universitaire 2024-2025

Table des matières

1	Introduction	2
2	Présentation complète du système de combat Pokémon	2
2.1	Structures générales	2
2.2	Les attributs des Pokémon	2
2.3	Les attaques des Pokémon	4
2.4	Déroulement d'un combat	5
3	Complexité du jeu.	6
3.1	Complexité Générale	6
3.2	Rolls d'attaques	6
3.3	Facteurs de Branchement	6
4	Choix du format et Implications.	7
4.1	[[Gen 2] Stadium OU Rental]	7
4.2	Complexité réelle dans notre projet	7
4.3	Conséquences sur l'IA et la stratégie	8
5	Choix pour l'implémentation du jeu	8
5.1	Pokemon Stadium 2	8
5.2	Simulateur : Pokemon Showdown	8
6	Implémentation de l'IA	8
6.1	Pokemon damage calculator	8
6.2	Pokemon Showdown	9
6.3	Mewtwo Intelligence	9
6.4	L'algorithme MinMax	10
6.5	L'algorithme MinMaxalpha	10
6.6	Les 3 Niveaux d'IA	10
7	Tests des IA	11
7.1	Performances comparées des intelligences artificielles	11
7.2	Bilan stratégique global et impact des équipes	12
8	Sources	13

1 Introduction

Dans le cadre de ce projet d'Intelligence Artificielle, j'ai choisi de travailler sur **Pokémon**, et spécifiquement **Pokémon Stadium 2**, un jeu au tour par tour à actions simultanées où deux joueurs s'affrontent via des créatures nommées Pokémon. Ce choix s'explique par la richesse stratégique du système de combat Pokémon, qui conjugue mathématiques, probabilités, et décisions séquentielles.

L'objectif était d'implémenter des intelligences artificielles capables de simuler ces combats de manière crédible, efficace et évolutive. Avant de présenter nos IA, il est essentiel de comprendre en profondeur la complexité du système des combats qu'offre Pokémon, notamment ses mécaniques statistiques et stratégiques, ainsi que les contraintes particulières du format [Gen 2] Stadium OU Rental qui fut notre terrain d'expérimentation.

2 Présentation complète du système de combat Pokémon

2.1 Structures générales

Les combats Pokémon impliquent des Pokémon avec des particularités singulières qui vont influencer sur leurs attributs, force et faiblesses. Le but d'un combat est de constituer une équipe de jusqu'à 6 Pokémon et de pouvoir vaincre une équipe de jusqu'à 6 Pokémon.

2.2 Les attributs des Pokémon

Beaucoup de facteurs entrent en jeu dans un combat de Pokémon. Rentrons dans les attributs qui constituent un Pokémon.

- **Base stats** : Ce sont les statistiques de base uniques à chaque Pokémon. Il en existe 6, elles sont influencées et accentuées avec le niveau jusqu'au niveau 100.
 - PV (HP), Représente le nombre de point de vie.
 - Attaque (Atk), Détermine les dégâts infligés par les attaques physiques.
 - Défense (Def), Représente les dégâts reçus par les attaques physiques.
 - Attaque Spéciale (SpA), Détermine les dégâts infligés par les attaques spéciales.
 - Défense Spéciale (SpD), Détermine les dégâts reçus infligés par les attaques spéciales.
 - Vitesse (Spe), Représente la vitesse.
- **IVs (Individual Values)** : Points 0 à 31 propres à chaque statistique aléatoires à la naissance (ou capture en jeu), et qui influent directement les stats jusqu'au niveau 100.
- **EVs (Effort Values)** : 510 points qui peuvent être répartis entre les différentes stats, 0 à 252 par stat, ces points sont gagnés à chaque combat dépendant du Pokémon vaincu ou sont influençables via des objets.
- **Nature** : Modifie deux statistiques : l'une est augmentée de 10%, l'autre diminuée de 10%. Certaines natures sont dites neutres, lorsqu'elles augmentent et réduisent la même statistique.
- **Objets** : ajoutent des effets passifs (ex : multiplier une stat, rendre un move prioritaire, régénérer la vie)

- **Types** : Type d'un Pokémon, il en existe 18, ils peuvent donner un boost aux attaques de même type que celui du Pokémon(STAB)[+50%], influencer si une attaque va faire plus de dégâts(Feu vs Plante)[+200%] moins de dégâts (Plante Vs Acier)[] Un Pokémon peut avoir 1 à 2 type.

Pokémon Type Chart
created by pokemondb.net
Applies to all games since Pokémon X&Y (2013)

0 No effect (0%)
 ½ Not very effective (50%)
 Normal (100%)
 2 Super-effective (200%)

DEFENSE → ATTACK ↴	NOR	FIR	WAT	ELE	GRA	ICE	FIG	POI	GRO	FLY	PSY	BUG	ROC	GHO	DRA	DAR	STE	FAI
NORMAL													½	0			½	
FIRE		½	½		2	2						2	½		½		2	
WATER		2	½		½				2				2		½			
ELECTRIC			2	½	½				0	2					½			
GRASS		½	2		½			½	2	½		½	2		½		½	
ICE		½	½		2	½			2	2					2		½	
FIGHTING	2					2		½		½	½	½	2	0		2	2	½
POISON					2			½	½				½	½			0	2
GROUND		2		2	½			2		0		½	2				2	
FLYING				½	2		2					2	½				½	
PSYCHIC							2	2			½					0	½	
BUG		½			2		½	½		½	2			½	2	½	½	
ROCK		2				2	½		½	2		2					½	
GHOST	0										2			2		½		
DRAGON															2		½	0
DARK							½				2			2		½		½
STEEL		½	½	½		2							2				½	2
FAIRY		½					2	½							2	2	½	

Image du Type Chart.

- **Niveaux** : Allant de 1 à 100, influence les statistiques, certaines attaques et permet à certains Pokémon de changer de forme, impliquant une modification de Base State(Evolution)
- **Gender** : Genre du Pokémon. Il peut être Male, femelle, ou asexué
- **Bonheur** : Statistique allant de 0 à 255.
- **Talents** : Introduits à partir de la génération 3, les talents (ou "abilities") sont des effets passifs propres à chaque espèce de Pokémon. Ils peuvent affecter le déroulement du combat (ex. : immunité à un type, régénération, changement de météo) ou agir en dehors des combats. Un Pokémon peut avoir un ou plusieurs talents possibles, mais un seul est actif à la fois.

2.3 Les attaques des Pokémon

Chaque Pokémon peut connaître jusqu'à 4 attaques à la fois. Une attaque peut être de type physique, spéciale ou statut, et constitue l'outil principal d'un Pokémon pour interagir en combat. Les attaques se distinguent par plusieurs attributs qui influencent leur efficacité et leur usage stratégique.

- **Type** : Chaque attaque possède un type (ex. : Feu, Eau, Électrik, etc.). Il détermine son efficacité contre les types adverses, selon les affinités élémentaires.
- **Catégorie** : Elle peut être physique (basée sur l'Attaque), spéciale (basée sur l'Attaque Spéciale) ou statut (effets divers sans dégâts directs).
- **Puissance** : Valeur numérique exprimant le potentiel de dégâts d'une attaque //(ex. : Tonnerre a une puissance de 90).
- **Précision** : Probabilité qu'une attaque touche sa cible (exprimée en pourcentage, généralement entre 30% et 100%).
- **Priorité** : Niveau qui détermine l'ordre d'exécution avant même la comparaison de vitesses (ex. : Vive-Attaque a une priorité +1).
- **Points de Pouvoir (PP)** : Nombre d'utilisations disponibles pour une attaque. Lorsqu'un move n'a plus de PP, il ne peut plus être utilisé.
- **Effets secondaires** : Certains moves peuvent infliger des statuts (paralysie, brûlure), baisser des statistiques adverses, ou produire des effets spéciaux (météo, piègeage...).
- **Coup critique** : Probabilité qu'un coup inflige environ 1,5 fois plus de dégâts que prévu. Certaines attaques ou situations augmentent cette probabilité.
- **Cible** : Spécifie sur qui l'attaque agit : un adversaire, soi-même, ou tous les Pokémon (ex. : Hurlement affecte l'adversaire, Danse Pluie agit globalement).
- **Contact physique** : Certaines attaques impliquent un contact direct, ce qui peut activer certains effets.
- **Nombre de coups** : Certaines attaques frappent plusieurs fois en un seul tour (ex. : Furie peut frapper entre 2 et 5 fois).
- **Charge/Rechargement** : Certaines attaques nécessitent un tour de préparation (ex. : Lance-Soleil) ou de recharge (ex. : Ultralaser).
- **Modification de terrain** : Quelques attaques changent les conditions du combat (ex. : Danse Pluie modifie la météo, ce qui influence d'autres attaques).
- **Blocage ou piègeage** : Certaines attaques empêchent l'adversaire de fuir ou de changer de Pokémon (ex. : Ligotage, Requiem).
- **Effet de soin** : Certaines attaques restaurent les PV du lanceur ou d'un allié (ex. : Vole-Vie, Repos).
- **OHKO** : Attaque qui tue en un coup avec une précision variable.

Pour résumer, un Pokémon est défini par l'ensemble de ses attributs : statistiques, type, niveau, nature, attaques, et autres paramètres spécifiques. L'image suivante illustre l'ensemble des informations composant une instance complète d'un Pokémon dans notre simulateur :

The screenshot displays a Pokémon simulator interface for Rayquaza-Mega. The top section includes a nickname field (Rayquaza), a copy button, and buttons for import/export, move, and delete. Below this, the 'Details' tab shows the Pokémon's level (100), gender (—), happiness (255), shiny status (Yes), and HP type (Dark). The Tera Type is Dragon, and the ability is Delta Stream. The item is Choice Band. The moves listed are Dragon Dance, Dragon Ascent, V-create, and Dragon Claw. The stats section shows HP, Atk (252+), Def, SpA (—), SpD (4), and Spe (252). The EVs section shows a guessed spread of 252 Atk / 4 SpD / 252 Spe / (+Atk, -SpA) with a Smogon analysis link. The base stats are HP 105, Attack 180, Defense 100, Sp. Atk. 180, Sp. Def. 100, and Speed 115. The EVs are 252+ for Atk, 4 for SpD, and 252 for Spe. The IVs are 31 for all stats. The nature is Adamant (+Atk, -SpA). The remaining EVs are 0.

Base	EVs	IVs	Total
HP 105		31	351
Attack 180	252+	31	504
Defense 100		31	236
Sp. Atk. 180	—	31	356
Sp. Def. 100	4	31	237
Speed 115	252	31	329

Photo : Vue détaillée d'un Pokémon avec l'ensemble de ses caractéristiques.

2.4 Déroulement d'un combat

Au début d'un combat les joueurs choisissent avec quels pokémons ils vont commencer(Lead).

Suite à cela, chaque tour, les joueurs choisissent une action en simultané :

- Soit une des 4 attaques disponibles de son Pokémon actif,
- Soit un changement vers l'un des autres Pokémon vivants de son équipe.

Pour gagner il faut mettre KO tous les Pokémon adverse.

Remarque : Il existe plusieurs modes de combat, comme le double, le VGC, le multi et d'autres.

Ici il sera question du mode combat Solo, donc 1v1 avec 6v6 Pokémon uniquement.

3 Complexité du jeu.

3.1 Complexité Générale

Plusieurs facteurs influent sur la complexité du système de combat du jeu.

- **Nombre d'attaques disponible (Gen 9)** : 934
- **Nombre d'attaques connu en moyenne par Pokémons** : $74 \Rightarrow 744 \approx 1.15 \times 10^6$
- **Nombre de Pokémons disponibles (Gen 9)** : 1025
- **Combinaisons d'équipes de 6(Pokémons uniques)** $1025^6 \approx 1.49 \times 10^{15}$
- **Répartitions d'IV Possible** : 32 valeurs $(0-31) * 6 = 32^6$
- **Répartitions d'EV Possible** : 64^6
- **Nombre de nature possible** : 25
- **Nombre d'objets tenu possible** : 337
- **Talent possibles** : 3

En combinant ces facteurs, on peut estimer le nombre de **sets distincts possibles pour un seul Pokémon** comme :

$$744 \times 32^6 \times 64^6 \times 25 \times 337 \times 3 \approx (1.15 \times 10^6) \times (1.07 \times 10^9) \times (6.87 \times 10^{10}) \times 25 \times 337 \times 3 \approx 7.0 \times 10^{27}$$

Ainsi, le nombre de **combinaisons d'équipes complètes de 6 Pokémon** (sans redondance, avec sets complets) est de l'ordre de :

$$(7.0 \times 10^{27})^6 \approx 1.17 \times 10^{166}$$

Cette estimation donne un ordre de grandeur du nombre de stratégies théoriquement possibles, ce qui rend toute approche exhaustive complètement irréaliste.

3.2 Rolls d'attaques

Chaque attaque a un roll, une même attaque d'un même Pokémon sur le même autre peut donner plusieurs résultats possible, 39 valeurs. Et avec une possibilité de Rater(Miss) et faire un coup critique.

Finalement on a 39×2 soit 78 branches.

3.3 Facteurs de Branchement

Par tour nous avons par joueurs 4 actions possibles, les attaques du Pokémon actif, ainsi que 5 switch possible.

Cela nous donne $9 \times 9 = 81$ combinaisons possibles par tour.

En ajoutant les rolls 81×78 on arrive à **6318** branchements finaux.

On estime un match durer en moyenne 20 tours, on a donc 6318^{20} soit 1.28×10^{70}

Cette complexité rend toute exploration exhaustive impossible. Cela justifie l'usage d'algorithmes de recherche partielle (type Minimax à profondeur bornée, avec élagage α - β) et de fonctions d'évaluation heuristique.

4 Choix du format et Implications.

Pour cet exercice un choix bien précis a été fait, en effet Pokémon existe sur 9 générations et aujourd'hui, **34 jeux principaux** ont été publiés dans la série canonique (incluant les versions doubles et remakes).

Pour avoir un espace de travail raisonnable le choix du format fut la **[[Gen 2] Stadium OU Rental]**

4.1 [[Gen 2] Stadium OU Rental]

Nous allons passer de 2025 à 2001. Pokémon Stadium 2 est un jeu dérivé de Pokémon Or/Argent/Crystal, donc la 2eme génération. Ce jeu en particulier contient tous les Pokémon de cette génération ainsi que des pokémons d'emprunt, ce qui nous intéresse ici pour passer car on a un mode de jeu avec des sets fixes. Et un pool de pokémon réduit.

Pourquoi ne pas avoir choisi la 1ère génération pour ce projet ?

Bien que nous ayons un nombre de pokémon réduit[151] en première génération[Rouge/Bleu/Jaune] et d'un jeu Pokémon Stadium 1 avec le même système d'emprunt, on a surtout un nombre de bug hallucinant, ce qui rend la simulation très pauvre et qui se reposerait sur de l'exploitation de bug, de mécaniques qui ne fonctionnent pas forcément bien, un équilibrage plus faible et une diversité plus faible autant en terme de pokémon, de types et d'attaques.

Que change la [Gen 2] Stadium OU Rental] et que cela implique-t-il ?

- On passe de 1025 Pokémon a 251, 239 même avec la banlist.
(Certains pokémon trop forts sont banni pour garder un équilibrage.)
- Des movesets fixes, chaque pokémon a 4 attaques unique et connus.
<https://www.serebii.net/stadium2/rental.shtml>
- Pas d'objets tenus.
- Niveaux et Statistique d'EV et d'IV Fixes(Maximales).
- Natures inexistantes
- Talents inexistantes
- 17 types au lieu de 18.
- Nombre d'attaques réduites
- Nombre d'attaques disponible : 251

4.2 Complexité réelle dans notre projet

- **Nombre d'équipes possibles** : $2396 \approx 2.26 \times 10^{11}$
- **Actions par tour** :
 - 4 attaques + 0 à 5 switchs disponibles \Rightarrow entre 4 et 9 actions par joueur
 - **Nombre total d'interactions par tour** : jusqu'à $9 \times 9 = 81$ combinaisons possibles
- **Nombre total d'interactions possibles à profondeur 2** : $\approx 81^2 = 6561$
- **Nombre total d'interactions possibles à profondeur 3** : $\approx 81^3 = 531441$
- **Nombre total de branches sur 20 tours** : 1.22×10^{38} (On était a 1.28×10^{70})

4.3 Conséquences sur l'IA et la stratégie

Pokémon est un jeu non déterministe à information non parfaite et à somme nulle. En choisissant ce format le jeu a été réduit et permet d'être à information parfaite.

On a réduit le nombre de pokémon de 1025 à 251, on a réduit le nombre d'attaques de 934 à 251 et rendu le nombre d'attaque fixe par pokémon.

Aussi le tier et le format Rental, donc les attaques en plus d'être fixes, sont connus de tous.

L'espace de recherche s'en voit réduit, les states sont fixes et notre arbre à une profondeur 2 atteignable et imaginable. Avec quelques optimisations la profondeur 3 fut atteinte.

5 Choix pour l'implémentation du jeu

5.1 Pokemon Stadium 2

Le choix d'utiliser le vrai jeu fut envisagé mais plusieurs problèmes ont fait surface. Il nécessite soit le vrai jeu et donc engendre des coûts (Jeu+Console+Television), ou bien un émulateur.

L'émulateur du jeu aussi pose énormément de problème car pour interpréter les actions de l'adversaire il aurait fallu un OCR puissant et constant ou une lecture dans la RAM de l'émulateur.

Le but de ce projet étant de se focaliser sur l'implémentation des IA et non sur le "Hacking" d'un émulateur et/ou jeu.

Le choix fut donc porté sur un simulateur de combat.

5.2 Simulateur : Pokemon Showdown

Pokemon Showdown est un simulateur en ligne de combats pokémon. Il est possible d'affronter en ligne des joueurs dans plusieurs tiers et génération différentes.

Son Code Source est disponible en ligne, il est donc possible d'exploiter ce simulateur. Il faudra cependant noter que notre tier n'est pas exactement présent comme on le voudrait mais il y a les bases. On va donc pouvoir utiliser ce simulateur comme interface à la fois pour affronter des joueurs mais surtout affronter nos IA dans un espace restreint.

6 Implémentation de l'IA

6.1 Pokemon damage calculator

Pour calculer les dégâts nous avons une interface damage-calc qui est une bibliothèque Javascript/Typescript.

Grâce à ça on va pouvoir calculer les dégâts d'attaques sur un pokémon et donc pouvoir wrapper une majorité de calcul. On sera capable d'avoir les calculs spécifiques à la génération 2 (la nôtre) et pouvoir éviter quelque erreur car on aura les mêmes calculs et fonction que notre simulateur (Pokemon Showdown).

Les calculs doivent prendre en compte des facteurs comme les statistiques, les types, les attaques, l'état des pokémons, leurs boosts ou malus et la gestion des rolls.

6.2 Pokemon Showdown

Le code source Pokemon Showdown est entièrement en typescript. Cela pose problème mais après quelques réglages des bons fichiers le tier a pu être implémenté

- formats.ts fut modifié pour ajouter le tier et adapter les règles à celles du jeu précisément.
- rulesets.ts pour ajouter des règles comme l'interdiction des items.
- learnsets.ts pour faire en sorte que les pokémons ne puissent utiliser que les moves qu'ils sont censé avoir dans le jeu en équipe d'emprunt, autrement il y aurait beaucoup de choix possibles.
- Le simulateur a été adapté pour tourner sur le navigateur et se battre simplement.

6.3 Mewtwo Intelligence

Le module principal de ce projet est poke-env, une librairie Python qui expose une API haut-niveau permettant de connecter à un serveur Pokémon Showdown, de simuler des combats, d'envoyer des actions (attaques, switches), et de récupérer l'état du jeu en temps réel pour l'analyse et la prise de décision.

Grace à poke-env on va pouvoir communiquer avec le server, qui est en typescript/javascript, via du code python.

Pour récupérer les sets des pokémon il a fallu utiliser le site de serebii, qui a tous les move sets, convertir le html en liste et extraire les learnsets, il a servi pour modifier le code de pokemon Showdown mais cela nous sert aussi pour nos simulation et pour préparer les teams.

On évalue de l'impact potentiel de chaque action possible, notamment via des appels au module `DamageCalcSMOGON.py`, qui permet d'estimer les dégâts de chaque attaque grâce à l'API officielle `@smogon/damage-calc`.

Sur cette base, Mewtwo Intelligence explore les actions disponibles à l'aide d'un algorithme `Minimax` limité en profondeur (généralement 2 ou 3), qui simule les combinaisons d'actions possibles des deux joueurs. Pour chaque couple d'actions (attaque vs switch, etc.), l'IA simule l'état futur et évalue l'avantage stratégique via une heuristique basée principalement sur la différence de PV totaux.

Pour rendre les simulations fiables, chaque état est cloné et modifié temporairement, sans affecter le déroulement réel du combat, via l'objet `Attestates`. La logique de simulation intègre les priorités d'attaque, les vitesses, et les boosts de statistiques, tout en ignorant volontairement les éléments aléatoires (critiques, miss, variation des dégâts) pour garantir un comportement purement déterministe.

Après avoir choisi l'IA à affronter il suffit d'être connecté et d'accepter la demande de challenge.

Il est possible de faire affronter 2 IA, et de regarder leur match.

6.4 L'algorithme MinMax

On simule tous les tours possible pour notre arbre minmax, et puis selon la profondeur on simule les tours suivants. On arrive donc à :

- 81 tours simulé pour une profondeur de 1.
- 6561 tours simulé pour une profondeur de 2.
- 531441 tours simulé pour une profondeur de 3.

Il faut noter que notre arbre n'est pas un minmax classique car les 2 joueurs jouent en même temps.

L'heuristique est de choisir l'action qui nous avantage le plus contre l'action qui va le plus avantager l'adversaire. En somme on se prépare au pire pour minimiser les dégâts reçu et maximiser les dégâts qu'on peut faire en retour.

Plus on va avancer en profondeur plus on va pouvoir prévoir les coups possibles et optimaux. Ça va permettre à l'IA de possiblement faire un coup a priori mauvais ou sous optimal mais qui paye au tour d'après.

6.5 L'algorithme MinMaxalphabeta

Cet algorithme est une variation de Minmax, on va varier le nombre de branches qu'on va explorer en les triant, on part du principe que les pires branches en profondeur n ne seront jamais utiles donc on va les ignorer. Ça nous permet de limiter les calculs, réduire le temps de calcul et même pouvoir varier la profondeur plus loin que 3 en limitant les branches traités.

6.6 Les 3 Niveaux d'IA

Snell_Lv1 — IA heuristique simpliste Cette version ne simule pas d'état futur : elle se base uniquement sur l'état actuel et regarde en profondeur 1 l'action maximisant l'avantage immédiat (dégâts infligés estimés).

Son temps de calcul est de 10 secondes en moyenne.

Snell_Lv2 — IA Minimax profondeur 2 Cette IA explore toutes les combinaisons d'actions possibles algorithme MinMaxalphabeta sur deux tours et avec un rétention de 80% des meilleures branches. Elle choisit l'action menant au meilleur état anticipé selon une heuristique basée sur la différence totale de points de vie. Elle commence à réagir aux menaces adverses et peut éviter des décisions naïves.

Son temps de calcul est de 20 secondes en moyenne.

Snell_Lv3 — IA Minimax profondeur 3 pondérée Cette IA explore toutes les combinaisons d'actions possibles algorithme MinMaxalphabeta sur une profondeur de 3 qui est possible en utilisant une rétention de 25% des meilleures branchent. Cette IA adopte des comportements plus intelligents, comme sacrifier un Pokémon faiblement utile pour sécuriser un avantage stratégique. Elle a une vision a plus long terme du combat et ignore les coups les plus sous optimaux.

Son temps de calcul est de 30 à 40 secondes en moyenne.

Remarque : Plus il y a de pokémon KO, plus le temps de calcul réduit.

7 Tests des IA

7.1 Performances comparées des intelligences artificielles

Pokémon bot	Pokémon adv	Total vivants	Actions bot	Actions adv	Nœuds profondeur 1	Profondeur 2	Profondeur 3
1	1	2	4	4	16	256	4096
1	2	3	4	5	20	400	8000
2	2	4	5	5	25	625	15625
2	3	5	5	6	30	900	27000
3	3	6	6	6	36	1296	46656
3	4	7	6	7	42	1764	74088
4	4	8	7	7	49	2401	117649
4	5	9	7	8	56	3136	175616
5	5	10	8	8	64	4096	262144
5	6	11	8	9	72	5184	373248
6	6	12	9	9	81	6561	531441

Catégorie	Nœuds depth 1 (k²)	Nœuds depth 2 (k⁴)	Cumul depth 1
1	1	1	1
2	4	16	5
3	9	81	14
4	16	256	30
5	25	625	55
6	36	1296	91
7	49	2401	140
8	64	4096	204
9	81	6561	285

Evolution du nombre de nœuds par rapport au nombre de pokémon.

Les différentes versions de notre IA (Snell_Lv1, Snell_Lv2, Snell_Lv3) ont été confrontées à un total de 509 matchs. Chaque niveau met en œuvre une stratégie distincte, allant d'une prise de décision naïve à une planification stratégique par arbre MinMax de profondeur variable.

- **Snell_Lv1** : 237 victoires (**46.6%**), durée moyenne de victoire : **196s**, en **39.2 tours**.
- **Snell_Lv2** : 208 victoires (**40.9%**), durée moyenne : **237s**, en **53.2 tours**.
- **Snell_Lv3** : 63 victoires (**12.4%**), durée moyenne : **494s**, en **38.6 tours**.

On constate que l'IA de niveau 1 (Snell_Lv1), qui utilise une stratégie simple à un tour d'anticipation, dépasse largement les niveaux 2 et 3 en termes de taux de victoire. Cela s'explique par une exécution rapide de ses actions et un arbitrage efficace sans surcoût de calcul.

À l'inverse, Snell_Lv3, malgré une anticipation stratégique poussée (profondeur 3, alpha-bêta), montre une **forte dégradation des performances**, tant en efficacité qu'en temps d'exécution. La stratégie se révèle trop lourde computationnellement, induisant une sur-analyse qui ne se traduit pas par de meilleurs résultats.

Le temps moyen des matchs est ainsi **2.5 fois plus long** pour Snell_Lv3 que pour Snell_Lv1, sans impact positif sur les performances. Ce constat souligne que dans ce format particulier, une profondeur d'analyse élevée n'est **pas corrélée à un taux de victoire supérieur**.

7.2 Bilan stratégique global et impact des équipes

L'évaluation des performances globales confirme que le facteur **équipe** joue un rôle majeur dans les résultats des IA.

Voici les cinq meilleures équipes du tournoi, classées par taux de réussite :

Équipe	Taux de victoire	Tours moyens	Durée moyenne (s)
team2.txt	15.7%	40.4	288
team4.txt	9.8%	42.7	207
team1.txt	8.6%	52.8	304
team5.txt	7.9%	51.3	195
team3.txt	7.5%	43.3	233

Certaines compositions se révèlent donc naturellement plus robustes, grâce à une meilleure synergie des types, des priorités d'attaque ou des courbes de stats. D'autres équipes, notamment issues de génération aléatoire, n'ont remporté aucune victoire.

Conclusion du projet :

Ce travail nous a permis de valider empiriquement plusieurs hypothèses :

- Une stratégie **simple** mais efficace, combinée à un traitement rapide du log, suffit à surpasser des modèles plus complexes.
- L'impact de la **profondeur** d'anticipation est limité si elle n'est pas accompagnée d'un gain qualitatif dans la sélection des branches.
- La **composition d'équipe** influence fortement le déroulement du match, indépendamment du niveau d'IA.

Ainsi, Snell_Lv1 s'impose comme la meilleure stratégie sur notre corpus, tant en robustesse qu'en efficacité computationnelle.

8 Sources

Les sources utilisées pour ce projet proviennent à la fois de documents officiels, de bases de données communautaires, de vidéos explicatives et de forums spécialisés. Voici la liste complète classée par thématique :

Données et sets Pokémon

- <https://www.serebii.net/stadium2/l100rental.shtml>
- <https://www.serebii.net/stadium2/rental.shtml>
- <https://pokemondb.net/type>
- https://bulbapedia.bulbagarden.net/wiki/Pok  mon_Stadium_2
- https://en.wikipedia.org/wiki/Pok  mon_Stadium_2
- https://bulbapedia.bulbagarden.net/wiki/Rule_variants
- https://bulbapedia.bulbagarden.net/wiki/List_of_battle_glitches_in_Generation_I

Simulateur et outils techniques

- <https://github.com/smogon/pokemon-showdown>
- <https://pokemonshowdown.com>
- <https://github.com/smogon/damage-calc>
- <https://poke-env.readthedocs.io/en/latest/>

Articles et discussions stratégiques

- <https://www.smogon.com/forums/threads/pokemon-stadium-2-rental-metagame.3729642>
- <https://gamerant.com/pokemon-stadium-2-rental-pokemon-tier-list>
- https://www.reddit.com/r/pokemon/comments/1b9s4cw/pokemon_stadium_2_best_rentals

Ressources mathématiques et combinatoires

- https://www.reddit.com/r/pokemon/comments/85lhb2/how_many_competitive_pokemon_teams_are_possible/
- <https://www.smogon.com/forums/threads/how-many-possible-teams-are-there-in-pokemon.3648997/>
- <https://math.stackexchange.com/questions/2825678/how-many-ways-can-the-same-team>

Algorithmes et théorie de l'IA

- https://fr.wikipedia.org/wiki/  lagage_alpha-b  ta
- <https://www.youtube.com/watch?v=C1KpQc9cWmM>
- <https://www.youtube.com/watch?v=rhvj7CmTRkg>
- <https://www.youtube.com/watch?v=l-hh51ncgDI&t=311s>