

Projet NAO et IA

Vous choisirez un des sujets suivants.

Pour votre projet, vous êtes libre d'utiliser Prolog, les réseaux neuronaux ou la méta-heuristique de votre choix (algorithme génétique, colonie de fourmis, recuit...). A vous de choisir le meilleur outil et de l'adapter.

La présentation graphique de la résolution et des résultats est attendue.

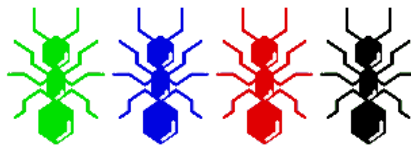
Les projets sont à rendre pour la dernière séance. Les éléments à rendre sont les suivants :

- Planification du projet (diagramme de Gantt par exemple avec un poker planning par exemple en début de projet).
- Choisir la méthode la plus appropriée pour le traitement.
- Un rapport présentant cette méthode dans le contexte choisi, en PDF
- La structure de données utilisée
- Le code commenté devra être présent sur Gitlab
- Un ou des jeux de test
- Réaliser une interface graphique utile pour l'apprentissage du réseau et son exploitation future.

Pour chaque jeu de test, des données de performance sont attendues.

Projet NAO et Intelligence Artificielle

Echelle de difficulté du plus facile au plus compliqué



Si le projet est noté d'une seule fourmi, un groupe de 2 personnes est possible.

Si le projet a été noté de deux fourmis, alors il est destiné à un groupe de 3 à 4 personnes.

Lors du déroulement de ce projet, l'algorithme et le langage vous appartiennent. Vous serez donc libres d'utiliser Prolog, les réseaux neuronaux ou la méta-heuristique de votre choix (algorithme génétique, colonie de fourmis, recuit...).

Attention à votre choix et à sa justification quant au meilleur outil. Un mauvais choix pourrait mener à une trop longue période d'adaptation.

Pour votre programme, la présentation graphique de la résolution et des résultats est attendue.

Les projets sont à rendre pour l'oral de la dernière séance. Les éléments à rendre sont les suivants :

- Planification du projet (diagramme de Gantt par exemple avec un poker planning par exemple en début de projet).
- Un rapport **en PDF** présentant
 - La méthode la plus appropriée pour le traitement dans le contexte choisi,
 - La structure de données utilisée,
 - Un ou des jeux de test
 - Tout élément propre à la compréhension de votre méthode
- Le code commenté **en français** devra être présent sur Gitlab / Github (attention au partage avec l'enseignant qui doit durer dans le temps)
- Une interface graphique utile pour l'application et son exploitation future.

Pour chaque jeu de test, des données de performance sont attendues.

De plus, chaque projet peut être amené à préciser ses propres rendus spécifiques.

1. Logimages



Vers 1994, un certain type de casse-tête a été très populaire en Angleterre avant de venir sur nos côtes. Ces casse-tête sont des énigmes en provenance du Japon et sont régulièrement publiés dans les journaux. Il suffit d'utiliser votre logique et votre habileté pour remplir la grille et révéler une image ou un diagramme. En tant que programmeur Prolog, vous seriez dans une excellente position: vous avez votre ordinateur pour faire le travail! Sauf qu'ici Prolog ne sera pas tout seul.

Le puzzle se déroule comme ceci:. Essentiellement, chaque ligne et chaque colonne d'une image bitmap rectangulaire est annoté avec les longueurs respectives de ses chaînes distinctes de cellules occupées. La personne qui résout le casse-tête doit remplir le bitmap donné que ces longueurs.

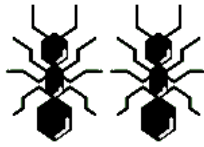
Problème	:	Solution:
_ _ _ _ _ _ _ 3		_ X X X _ _ _ _ 3
_ _ _ _ _ _ _ 2 1		X X _ _ X _ _ _ _ 2 1
_ _ _ _ _ _ _ 3 2		_ X X X _ _ X X _ 3 2
_ _ _ _ _ _ _ 2 2		_ _ X X _ _ X X _ 2 2
_ _ _ _ _ _ _ 6		_ _ X X X X X X _ 6
_ _ _ _ _ _ _ 1 5		X _ X X X X X _ _ 1 5
_ _ _ _ _ _ _ 6		X X X X X X _ _ 6
_ _ _ _ _ _ _ 1		_ _ _ _ X _ _ _ _ 1
_ _ _ _ _ _ _ 2		_ _ _ X X _ _ _ _ 2
1 3 1 7 5 3 4 3		1 3 1 7 5 3 4 3
2 1 5 1		2 1 5 1

Pour l'exemple ci-dessus, le problème peut être énoncé comme deux listes $[[3], [2,1], [3,2], [2,2], [6], [1,5], [6], [1], [2]]$ et $[[1,2], [3,1], [1,5], [7,1], [5], [3], [4], [3]]$ qui donnent les "solides" longueurs des lignes et des colonnes, de haut en bas et de gauche à droite, respectivement. Les énigmes publiées sont plus grandes que cet exemple, par exemple, 25 x 20, et ont apparemment toujours des solutions uniques.

Le prolog ne sera pas seul langage utilisé sur ce projet. Il vous faudra faire une comparaison avec une autre solution.

Données, sources	Résultat attendu
Un fichier CSV définissant la grille	Un fichier CSV avec la grille résolue

2. Des algorithmes bien artistiques



La musique est régie par des règles d'harmonie. La génération d'un morceau par l'ordinateur en utilisant ces règles d'harmonie est donc possible.

Pour apprendre les règles, vous pouvez vous baser sur le site suivant par exemple.

<http://michelbaron.phpnet.us/harmonie.htm>

Vous pourrez vous contenter des règles de quinte dans un premier temps puis augmenter le nombre de règles.

Pour jouer la musique, le lien <https://wiki.python.org/moin/PythonInMusic> peut être utile.

Une construction en temps réel est attendue. Pensée par bloc de notes ou note à note, selon la méthode choisie, le résultat devra être enregistrable si possible et écoutable en direct à minima.

Données, sources	Résultat attendu
Rien ou une compilation d'extraits musicaux	Un fichier musical et une sortie audio en direct

3. Light Ricochet Robot™



Ricochet Robots (Rasende Roboter pour la première édition en allemand) est un jeu de société créé par Alex Randolph et illustré par Franz Vohwinkel, édité en 1999 par Hans im Glück / Tilsit. Le jeu est composé d'un plateau, de tuiles représentant chacune une des cases du plateau, et de pions appelés « robots ». Pour la partie, un objectif de couleur est tiré au sort. Le but du jeu est d'amener le robot de la couleur correspondante sur l'objectif en un minimum de coups.

Sur le plateau les robots se déplacent en ligne droite, et le plus loin possible avant de rencontrer un obstacle. Les joueurs peuvent utiliser les quatre robots comme ils le souhaitent.

Une fois mis en mouvement, le robot ne peut s'arrêter que lorsqu'il rencontre un obstacle. On peut alors le faire repartir dans une autre direction ou choisir un autre robot. Les obstacles peuvent être :

- les bords du plateau
- les murs symbolisés sur le plateau
- un autre robot

Chaque déplacement de robot compte pour un mouvement, quel que soit le nombre de cases parcourues.

Après avoir tiré au sort un labyrinthe et la position des objectifs, placez les 4 robots de couleur aléatoirement.

Puis en boucle :

- tirez un objectif jusqu'à ce que tous aient été pris
- amenez le robot de la couleur de l'objectif sur ce dernier

Une carte simplifiée pourra être utilisée

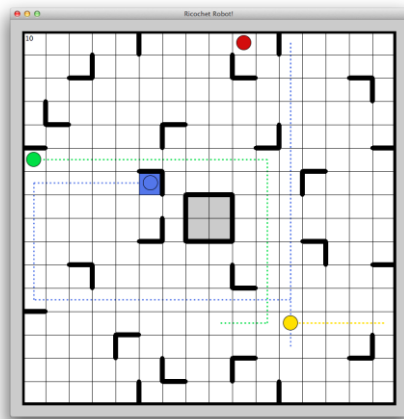


Figure 1. Le robot bleu a atteint l'objectif bleu

Données, sources	Résultat attendu
Un fichier CSV définissant la grille	Un fichier donnant la suite des ordres à donner aux robots

4. Employer votre temps



Toute ressemblance entre cette situation et une situation réelle est involontaire. Ou pas.

Le problème d'un emploi du temps revêt une très grande importance pratique. Dans notre cas, nous étudierons la version plus particulière des job dating :

- Une liste des plages horaires disponibles $\{T1, \dots, Tm\}$
- Une liste d'étudiants $\{J1, \dots, Jj\}$
- Une liste d'entreprises $\{E1, \dots, Ei\}$
- La liste des offres proposées par chaque entreprise sous la forme $\{(E1, O1), \dots, (Et, On)\}$ avec Et une entreprise et On une offre.
- La liste des étudiants devant voir quelle entreprise $\{(J1, E1), \dots, (Jn, Ek)\}$

Le problème consiste à trouver un emploi du temps satisfaisant aux contraintes spécifiques suivantes:

- Il faut utiliser le moins de créneaux possibles
- On privilégiera les étudiants ayant le plus d'entretien de façon à les faire passer en premier
- On gardera un écart d'un créneau entre les entretiens des étudiants
- Il ne faut pas de doublon bien évidemment
- Les entreprises ne doivent pas avoir de trou dans leur créneaux

On attend en sortie la liste des créneaux avec les entreprises et les étudiants mais aussi pour chaque entreprise la liste de ses entretiens avec les créneaux et enfin pour chaque étudiants la liste de ses entretiens.

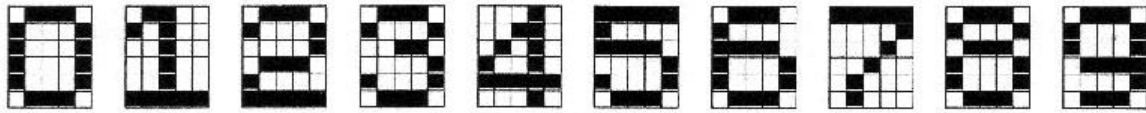
Données, sources	Résultat attendu
Un fichier CSV définissant les listes	Un fichier TXT avec la liste de créneaux, un pour lister les rendez-vous de chaque entreprise et un pour liste les rendez-vous de chaque étudiant

5. Reconnaissance de caractère



On se propose de réaliser un réseau de neurones de reconnaissance de caractères (on s'intéressera dans un premier temps aux chiffres de 0 à 9).

Après une phase d'apprentissage, le réseau devra être en mesure de reconnaître des données bruitées.



Exemple de codage de caractères

A partir de ce point de départ, vous réaliserez les améliorations suivantes :

- Extension à la reconnaissance de lettres.
- Possibilité de reconnaître votre propre écriture manuscrite (sous forme de texte)
- Réalisation d'un prototype de machine de tri automatique.

Vous apporterez toutes autres améliorations qui vous semblent utiles.

Données, sources	Résultat attendu
Un dossier de fichiers image de codes postaux et villes à trier	Un fichier TXT avec l'interprétation des textes et la décision à appliquer à chacun (département et ville de destination)

6. Le Labyrinthe



La théorie de Donald Hebb fut testée en 1951 par Marvin Minsky et Dean Edmond qui réalisèrent un « cerveau artificiel » à l'Université de Harvard. La machine, appelée Snark, était composée de trois cents tubes à vide, quarante résistances variables servant de coefficients synaptiques associés à un système électromécanique pour leur réglage. Malgré de nombreux dysfonctionnement, la machine simula le comportement d'un rat recherchant de la nourriture dans un labyrinthe.

Le principe de ce projet va être de reproduire les capacités d'apprentissage de Snark et de se servir du résultat pour déplacer HAL, un robot virtuel, au sein d'un labyrinthe.

Pour des raisons pratiques, nous limiterons la taille du labyrinthe à une matrice (ex. 8x8 mais ce n'est qu'un exemple). Chaque position peut contenir un obstacle, l'objectif ou être un espace vide.

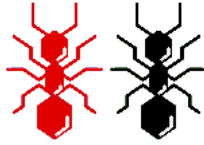
Le but va être d'employer un algorithme d'apprentissage pour arriver à rejoindre l'objectif puis de guider Hal selon les instructions ainsi calculées et se saisir de l'objectif.

1. Dans un premier temps, il vous est demandé de choisir un algorithme à appliquer au problème. Sont attendus dans votre rapport, un cours exposé des principes de fonctionnement mais aussi les avantages et inconvénients, l'algorithme...
2. Dans un second temps, vous implémenterez le problème en python.

Vous comparerez le résultat avec l'implémentation de l'algorithme de la « main droite ».

Données, sources	Résultat attendu
Un fichier CSV définissant le labyrinthe	Un fichier donnant la suite des ordres à donner à Hal et le nombre de mouvements ainsi obtenus par Hal avec une comparaison avec l'algorithme de la main droite

7. Les petits chevaux



Le but est d'apprendre à NAO à jouer aux petits chevaux. En lançant le dé ou en le lisant, suivant plusieurs chevaux, calculant la meilleure tactique, gérant les déplacements des adversaires...

L'utilisation d'un terrain à cases numérotées facilitera la représentation du problème.

Il n'est demandé dans ce projet que de gérer l'intelligence du jeu pas les déplacements de NAO.

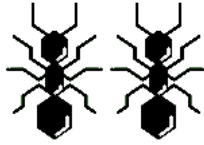
L'interaction avec NAO pourra passer par un casque micro pour minimiser les perturbations environnementales.

Le but va être d'employer un algorithme d'intelligence artificielle pour arriver à remplir votre objectif puis de guider NAO selon les instructions ainsi calculées.

1. Dans un premier temps, il vous est demandé de choisir un algorithme à appliquer au problème. Sont attendus dans votre rapport, un cours exposé des principes de fonctionnement mais aussi les avantages et inconvénients, l'algorithme...
2. Dans un second temps, vous implémenterez le problème en python.
3. Enfin, vous utiliserez le SDK de NAO pour gérer vos échanges avec lui et remplir votre objectif.

Données, sources	Résultat attendu
Un casque micro pour capter les commandes vocales	La suite de réponses données par NAO.

8. Le poker



Il s'agit d'apprendre à jouer au poker à NAO. La reconnaissance des cartes (peut-être un jeu aux dimensions exagérées <http://www.latoutmaitre.com/?c32>) serait une première étape.

Pour aider dans la résolution, vous pourrez vous appuyer sur le jeu de données suivant : <http://archive.ics.uci.edu/ml/datasets/Poker+Hand>

Il n'est demandé dans ce projet que de gérer l'intelligence du jeu pas les déplacements de NAO.

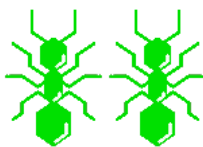
L'interaction avec NAO pourra passer par un casque micro pour minimiser les perturbations environnementales.

Le but va être d'employer un algorithme d'intelligence artificielle pour arriver à remplir votre objectif puis de guider NAO selon les instructions ainsi.

1. Dans un premier temps, il vous est demandé de choisir un algorithme à appliquer au problème. Sont attendus dans votre rapport, un cours exposé des principes de fonctionnement mais aussi les avantages et inconvénients, l'algorithme...
2. Dans un second temps, vous implémenterez le problème en python.
3. Enfin, vous utiliserez le SDK de NAO pour gérer vos échanges oraux (ceux du poker) avec lui et remplir votre objectif.

Données, sources	Résultat attendu
Un casque micro pour capter les commandes vocales	La suite de réponses données par NAO.

9. Protocole STP



Le protocole du Spanning Tree a été créé en 1985 par Radia Perlman. Les réseaux commutés de type Ethernet doivent avoir un chemin unique entre deux points, cela s'appelle une topologie sans boucle. En effet, la présence de boucle génère des tempêtes de diffusion qui paralysent le réseau : tous les liens sont saturés de trames de diffusion qui tournent en rond dans les boucles et les tables d'apprentissage des switchs deviennent instables.

Si l'on représente le réseau par un graphe aux nœuds fortement interconnectés et où les nœuds représentent des switchs, vous avez pour tâche de recréer un protocole équivalent.

Pour vous placer dans les mêmes conditions que le STP, vous partirez juste après l'élection du commutateur racine.

- Le but premier est de créer une IA donnant une topologie sans boucle, avec les liens les plus courts.
- Dans un second temps, la topologie, et donc l'IA, doivent pouvoir réagir si la commande d'extinction (ou d'allumage) d'un nœud survient.

Données, sources	Résultat attendu
Un fichier CSV permettant la création du réseau et précisant les identifiants de chaque nœud. Le plus petit sera le commutateur racine.	Une représentation du graphe sans boucle à l'instant t. Il doit être possible de modifier le graphe en faisant tomber un nœud à tout instant.

10. Pile TCP/IP



Attention théorie des réseaux inside !

En utilisant un sniffeur et un trafic non sécurisé, vous injecterez l'ensemble (données + entêtes) de la pile TCP/IP de la couche application à la couche IP. Ces entêtes ont été obtenus par passage dans toutes les couches du modèle OSI et des règles établies selon l'homme et les données. L'objectif est de retrouver des règles propre à la machine mais valables pour le transfert de ces même données.

Vous aurez donc en entrées : des données qui correspondent à un message complet (encapsulation complète), avec plusieurs qualités (HTTP, TCP, IP, ...). Cela constituera votre jeu de données.

Après apprentissage, votre application doit pouvoir :

Dans un premier temps, lire n'importe quel message qui lui ait donné

Dans un second temps, écrire ses propres messages complets via l'« encapsulation » réalisée par l'IA (soit un datagramme IP valide).



Données, sources	Résultat attendu
<ul style="list-style-type: none">• Un fichier TXT ou, mieux, une écoute directe de la couche réseau.• Un message de la couche application	<ul style="list-style-type: none">• Une traduction des données en données utilisables par la couche application• Un datagramme IP valide

11. Routage à haute densité



Attention théorie des protocoles réseaux inside !

Il s'agit pour ce projet de trouver un remplaçant au BGP. Le BGP actuellement entre les plus grandes AS maintient des centaines de milliers de lignes. Contrairement aux protocoles de routage interne, BGP n'utilise pas de métrique classique mais fonde les décisions de routage sur les chemins parcourus, les attributs des préfixes et un ensemble de règles de sélection définies par l'administrateur de l'AS. On le qualifie de protocole à vecteur de chemins.

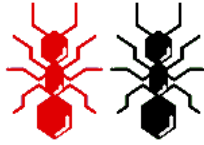
Il s'agit ici de proposer une IA permettant de traiter ces milliers de lignes pour améliorer le protocole. En effet, le BGP comporte de nombreux désavantages qui mènent à sa non-optimisation : vitesse de convergence, oscillations, MOAS, équilibrage de la charge.

En partant des annonces réalisées par les AS (et donc de métriques multiples), il est sûrement possible de définir une IA créant, de façon réactive, le meilleur réseau possible. Le projet ne s'attardera pas sur la propagation de la solution obtenue.

- Le but premier est de créer une IA donnant une topologie avec les liens les plus courts et tenant compte des métriques diverses de BGP.
- Dans un second temps, la topologie, et donc l'IA, doivent pouvoir réagir si la commande d'extinction (ou d'allumage) d'un nœud survient.

Données, sources	Résultat attendu
Un fichier CSV avec les lignes d'une table de routage BGP permettant la création du réseau.	Une représentation du graphe du réseau à l'instant t. Il doit être possible de modifier le graphe en faisant tomber un nœud à tout instant.

12. Génération de livre/article



En vous basant sur le corpus du site xyz, vous réaliserez une IA capable d'écrire des articles voire des livres justes tant au niveau de la forme que du fond.

Pour avoir une idée du résultat pour un livre, le lien suivant a été obtenu avec une IA :

<http://botnik.org/content/harry-potter.html>

Le site Kaggle propose plusieurs sets d'articles (en anglais ou en français) qui peuvent servir de base d'apprentissage. Pour les bases d'apprentissage de livres, les epub d'auteurs aux droits libérés ne manquent pas.

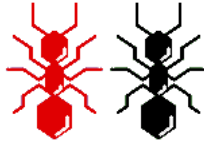
<https://www.kaggle.com/saxinou/articles>

<https://www.kaggle.com/asad1m9a9h6mood/news-articles>

- Dans un premier temps, vous vous concentrerez sur la rédaction d'article d'au minimum une page de texte.
- Dans un second temps, vous étendrez votre concept pour pouvoir générer un livre de 50 000 mots (un peu comme le NaNoWriMo).

Données, sources	Résultat attendu
Une base d'apprentissage en français centrée sur un style ou un auteur	Un texte en français respectant la grammaire, la conjugaison et la sémantique.

13. EPSI Car



En partant d'une maquette motorisée sommaire, vous réaliserez une voiture autonome capable d'aller d'un point A à un point B connus d'elle en évitant les obstacles.

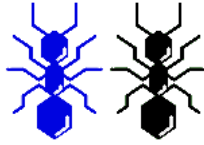
L'IA de votre véhicule devra donc connaître la géographie d'un lieu, être en capacité de savoir où il se trouve par lui-même, de comprendre quel lieu est visé par le déplacement puis le réaliser.

Pour être une vraie EPSI Car, l'ajout et la reconnaissance d'une signalisation de base sera nécessaire : panneaux stop, sens interdit, lignes blanches.

- Le premier objectif va être de trouver le moyen de donner un sens géographique à votre véhicule.
- Ensuite, il s'agira de lui donner la capacité de se rendre d'un lieu à un autre en évitant les obstacles
- Enfin, il devra respecter le code de la route
- Un gros bonus serait l'interaction avec le véhicule par commande vocale

Données, sources	Résultat attendu
Une destination	Déplacement du véhicule en évitant les obstacles et en respectant le code de la route jusqu'au point d'arrivée

14. Lève-toi et marche



Le principe de ce projet consiste à réaliser un prototype de robot à 4, 6 ou 8 pattes et à réaliser l'apprentissage de la marche. A l'extrême, il devra être capable de résister à une contrainte opposée, voire à un coup.

Il y a deux possibilités :

Soit le prototype sera réalisé avec les Raspberry et/ou Arduino et des moteurs liés à des éléments imprimés en 3D. Les moteurs pourront alors être soit en rotation libre soit avec un nombre de positions déterminé à l'avance.

Soit le tout sera simulé à l'aide d'un moteur physique (Newton Physics, PhysX, ...). Dans ce second cas, le nombre de pattes sera un paramètre.

Le processus d'apprentissage devra se centrer sur l'évaluation de la réussite dans la marche.

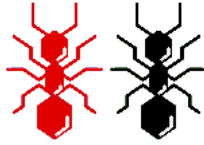
La première étape consistera à obtenir de chaque membre la capacité de bouger sous l'impulsion de muscles/moteurs.

La seconde étape sera d'établir la mesure de la progression et l'enchaînement pour une évolution.

L'étape ultime serait de perturber la marche pour voir comment le modèle s'en sort

Données, sources	Résultat attendu
Une modèle d' « animal »	Un « animal » marchant sans problème même sous la contrainte

15. IDS



Le concept d'IDS est un concept maintenant bien ancré dans le monde du firewalling. La plupart se basent sur la détection de pattern pour réaliser l'identification de ce qui est normal et de ce qui est anormal.

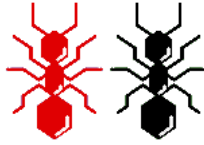
L'idée derrière ce projet est de créer un IDS en partant de l'écoute du réseau (d'un hôte et/ou d'un réseau). Votre IA devra être capable de détecter un comportement anormal, gérer la masse d'information, d'apprendre d'elle et de faire des recommandations. Vous n'êtes pas obligé d'aller jusqu'à un IPS, mais cette ouverture serait un ajout des plus intéressants.

Le choix NIDS ou HIDS déterminera vos sondes et donc l'alimentation de votre IA. Il sera donc à considérer en priorité.

- Une première étape consisterait à nourrir une IA avec des trames réseaux normales et anormales
- Dans un second temps, l'IA se verrait dotée de sondes pour interagir avec le réseau
- Dans un dernier temps, l'IA sera placée sur un réseau qui sera attaqué

Données, sources	Résultat attendu
Des écoutes réseaux	Une décision de l'IDS avec, si possible, une probabilité (ex : Attaque par injections SQL – Probabilité 67%)

16. #important



Actuellement, les tendances sur les réseaux sociaux sont des indicateurs importants et intéressants pour nombre d'entreprises.

En se consacrant tout d'abord à une poignée d'entre eux, vous chercherez à détecter les aspirations du moment.

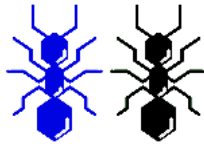
Cette détection passe par une étape de collecte puis par une phase d'interprétation. Il s'agit ici de travailler sur les tendances au sens large et non sur de simples #mots. La notion sémantique aura donc son importance.

Dans un second temps, et en vous basant sur la masse d'informations récoltées, vous devriez pouvoir développer votre IA vers l'origine de la tendance et, de là, votre outil devrait être capable de proposer des moyens d'influencer ces tendances.

- La première étape sera de construire une ou des sondes pour récupérer l'ensemble des tendances sur les réseaux sociaux visés. Des informations quantitatives et, si possible, de date, seront idéales à conserver.
- L'application devra être construite de telle manière à rapprocher des tendances apparentées (ex: #IA et #AI)
- Il s'agira ensuite d'essayer de déterminer, pour une tendance donnée par l'utilisateur, son origine. A l'extrême il sera intéressant de pouvoir représenter son évolution.
- A partir des éléments précédents, l'application devrait alors proposer des éléments pour influencer la tendance en question

Données, sources	Résultat attendu
Un ou plusieurs réseaux sociaux	Une ou des propositions d'action sur les tendances sélectionnées

17. Attrapez les tous



Pour représenter l'adaptation d'une équipe ou de produits à une tâche, nous nous proposons d'utiliser les combats pokemons.

La septième génération de Pokemon a amené le score à 802 pokemons. Les possibilités commencent à être importantes quand on en vient à la stratégie avec des équipes à 6 pokemons. Le tout devient explosif si l'on ajoute les combinaisons d'attaques/défenses et les objets.

Justement, sans vous soucier du principe des tiers pokemons, essayez de composer la meilleure équipe possible. Vous devrez forcément poser des contraintes pour cela qui resteront à votre discrétion.

Dans un second temps, vous vous intéresserez au tiers « Use them all », mis au point récemment, en tentant à nouveau de créer la meilleure équipe.

Il vous reviendra de créer la méthode d'évaluation des équipes. Les essais grandeur nature sur Showdown sont à minimiser pendant le temps de projet.

Pour les données, vous pourrez vous baser sur Kaggle :

<https://www.kaggle.com/mylesoneill/pokemon-sun-and-moon-gen-7-stats>.

D'autres sets existent sur le site.

- Dans un premier temps, à l'aide des datasets qui vous sembleront les plus intéressants, construisez votre dataset
- Dans un second temps, vous utiliserez ce dataset pour créer la meilleure équipe sans tenir compte d'autre chose que des règles de combat
- Enfin, vous appliquerez les règles du tiers « Use them all » avant de régénérer une ou des équipes optimisées

Données, sources	Résultat attendu
Des datasets de données définissant des pokemons et un tiers	Une équipe optimisée pour être la meilleure dans le set défini

18. Onitama



Sculpté dans les rochers des montagnes embrumés du vieux Japon réside l'Autel d'Onitama. Les maîtres d'écoles d'arts martiaux effectuent le voyage vers l'Onitama avec leurs disciples les plus prometteurs pour rencontrer dans ses murs sacrés les autres maîtres et disciples afin de leur prouver leurs supériorités au cours d'une bataille.

Sorti en septembre 2017, Onitama est un jeu de plateau japonais créé par Shimpei Sato assez simple permettant à deux joueurs de s'affronter sur un échiquier 5×5. Sur la première ligne devant lui, chacun pose son pion maître au centre (sur l'arche de temple) et 2 pions élèves de chaque côté du roi. Parmi les 16 cartes de mouvement, on en donne 2 à chaque joueur et une cinquième est placée à côté du tapis. Le reste des cartes ne sera pas utilisé pour cette partie.

Ce projet vous propose de créer une interface permettant soit de jouer contre une IA soit de regarder jouer deux IA. L'une des IA se basera sur la théorie des jeux et l'algorithme MinMax. L'autre IA utilisera une autre méthode pour calculer ses coups.

Les règles et cartes sont disponibles sur les liens suivants :

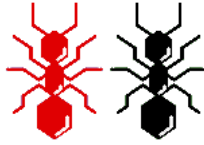
http://www.igiari.com/Onitama_regles.pdf

https://galeriatrolli.pl/29634-big_default/onitama-senseis-path.jpg

- Dans un premier temps, vous implémenterez l'IA du MinMax pour le jeu d'Onitama
- Dans un second temps, vous permettrez le jeu entre IA ou avec un humain
- Puis vous entraînerez une nouvelle IA contre le MinMax

Données, sources	Résultat attendu
Les joueurs (IA ou Humains)	Une partie de jeu d'Onitama entre les joueurs

19. MétéoEPSI



La prédiction est un domaine très utilisé en IA. Pour mettre cela en pratique, et parce que c'est l'un des domaines les plus controversés pour ses modèles, nous nous intéresserons à la météo. Il vous est donc proposé ici de battre des professionnels de la météo sur leur terrain.

Le projet se basera sur les données (données textuelles et/ou images satellites) passées que vous obtiendrez avec via les sites suivants :

https://donneespubliques.meteofrance.fr/?fond=produit&id_produit=109&id_rubrique=28

<http://www.infoclimat.fr/modeles-meteorologiques.html#forceTerm=78>

Ou plus complet :

<https://www.eumetsat.int/website/home/Data/DataDelivery/EUMETSATDataCentre/OrderingClient/index.html>

Avec ces images, vous réaliserez un apprentissage qui vous amènera à défier la prédiction réalisée par les sites de prédiction météorologique comme

<http://www.meteociel.com/modeles/gfs/precipitations/3h.htm>

- Dans un premier temps, vous mettrez en forme vos données (textes et/ou images)
- Puis, vous réaliserez un apprentissage à partir de données passées
- Enfin, à partir de données récentes, vous réaliserez une prédiction pour la météo à 3h, 6h, 9h, ...
Avant de comparer avec la prédiction de meteociel.com

Données, sources	Résultat attendu
Un ou des datasets obtenus à partir de sites de météo	Une prédiction en image de la météo par tranche de 3h