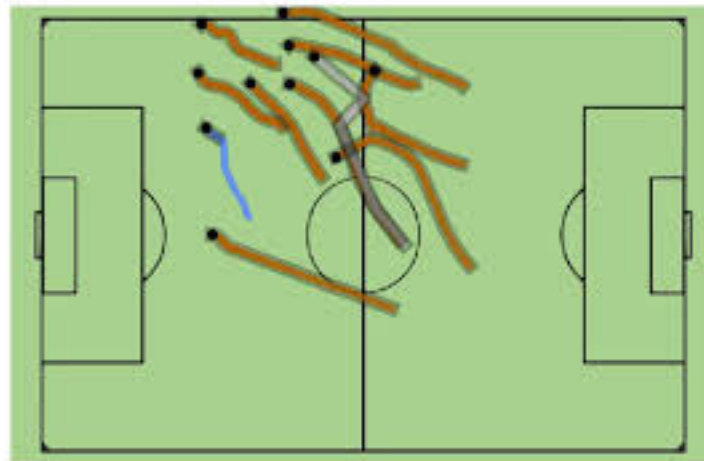


PROJET

Football multi-agents



Abalo Lena
Ait Sidi Hammou Achraf
Chollet Tom

Professeur : Mme Douae Ahmadoun

1 Description du contexte choisi

Nous avons fait le choix d'implémenter un projet de système multi-agents qui nous intéresse, mais qui a aussi déjà été appréhendé par DeepMind : il s'agit de la **simulation d'un match de football** par la programmation multi-agents. La simulation que nous avons mise en place oppose deux équipes d'agents, et la partie s'arrête lorsqu'un but est marqué par l'une des équipes, qui est donc la gagnante de la partie.

Ainsi, l'enjeu de ce projet est de procurer aux agents des informations pertinentes leur permettant d'effectuer des actions propres à notre contexte. Dans la suite de ce rapport, nous présenterons l'environnement ainsi que les agents mis en place et leurs caractéristiques.

2 Environnement mis en place

Dans le cadre de notre simulation, nous avons réalisé un environnement possédant les caractéristiques suivantes :

■ Accessible

Tous les joueurs ont accès entièrement aux informations liées aux autres joueurs ainsi qu'à l'environnement qui les entoure.

■ Déterministe

Les actions des agents ont un effet déterministe sur l'environnement.

■ Non épisodique

Après chaque action d'un agent, l'état du système n'est plus le même. L'on ne peut donc considérer les actions comme des épisodes indépendants.

■ Statique

Les changements de l'environnement ne sont liés qu'aux actions des agents.

■ Discret

L'ensemble des états possibles d'un agent donné est réduit à la surface du terrain de football, et cet ensemble est dénombrable.

3 Agents implémentés

Tous les agents implémentés dans le cadre de notre projet sont des agents **coopératifs** car ils possèdent des fonctions d'utilité et réalisent des actions maximisant cette fonction d'utilité.

Cependant, ils possèdent quelques caractéristiques des agents **réactifs** tels :

- La modularité : Les agents réalisent une tâche précise par un comportement précis.
- La simplicité : La perception d'un évènement entraîne une action.

De plus, les actions des agents sont organisées selon une architecture réactive de type **Subsomption** (par niveaux).

Trois types d'agents ont été mis en place, ayant chacun des attributs différents et des comportements différents.

3.1 Les agents Player

Ces agents représentent les joueurs de chaque équipe. Les joueurs jouent l'attaque ou la défense en fonction de la situation dans laquelle ils se situent. Leurs attributs sont les suivants :

- La position grâce à ses coordonnées x et y
- La vitesse
- L'équipe à laquelle le joueur appartient
- La zone d'interception, représentant l'intervalle dans lequel il peut intercepter le ballon.
- Une distance de visibilité, ainsi les autres joueurs hors de celle-ci ne sont pas pris en compte lors du choix des actions.

Les actions et déplacements de chaque joueur sont réalisées dans l'objectif de maximiser sa fonction d'utilité, et d'interagir avec ses coéquipiers possédant l'utilité la plus élevée. Un joueur fait la passe à un coéquipier si celui-ci a une utilité plus élevée que la sienne.

Les actions prises par les joueurs suivent un ordre de priorité en fonction de niveaux. Ainsi, les comportements de niveaux bas correspondent aux actions de priorité haute. Pour notre problème, dans le cas où l'on a le ballon, on définit les niveaux suivants :

- Niveau 0 : Effectuer un tir vers les buts lorsque l'on est prêt de ceux-ci
- Niveau 1 : Faire la passe lorsqu'on est marqué par un adversaire
- Niveau 2 : Se déplacer

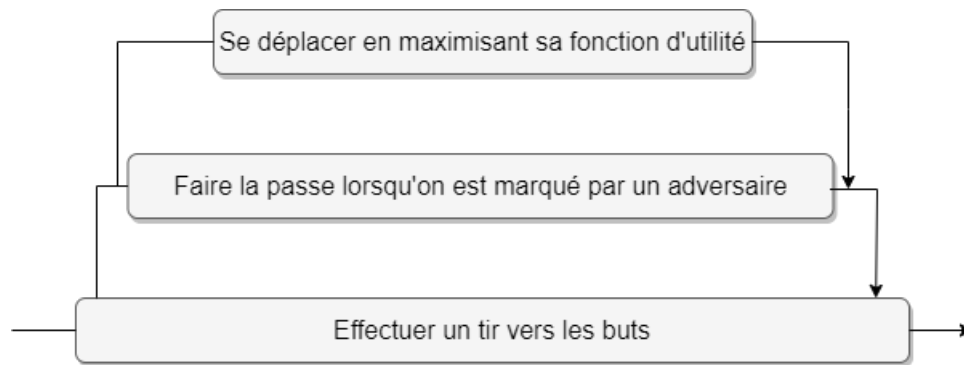


FIGURE 1 – Architecture par niveaux lorsqu'un joueur a le ballon

Dans le cas où personne ne possède le ballon, les niveaux sont les suivants :

- Niveau 0 : Récupérer le ballon
- Niveau 1 : Se déplacer

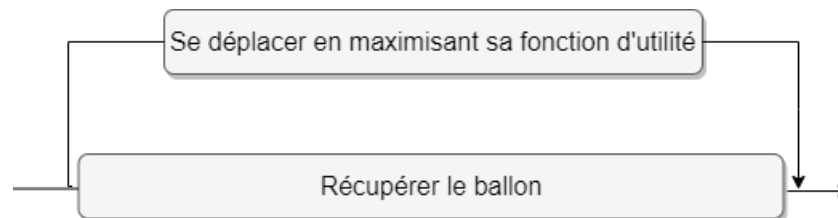


FIGURE 2 – Architecture par niveaux lorsqu'un joueur a le ballon

Lorsqu'un autre joueur possède le ballon, le joueur se déplace juste dans le but de maximiser sa fonction d'utilité, et peut intercepter le ballon avec une certaine probabilité.

Nous avons implémenté deux fonctions d'utilité différentes :

- Une première assez simple, qui repose sur plusieurs faits : les attaquants doivent essayer de s'éloigner des défenseurs et se rapprocher du but adverse, et les défenseurs doivent essayer de rester proches des attaquants mais aussi de leur but.
- Une seconde plus complexe faisant intervenir plusieurs facteurs dans son calcul : **En attaque**, l'objectif est de se rapprocher de coéquipiers ayant une distance élevée des défenseurs, se rapprocher des buts adverses, et ne pas rester trop près de ses coéquipiers. **En défense**, l'objectif est plutôt de se rapprocher du ballon s'il est dans sa zone de visibilité, se rapprocher des attaquants pour les marquer, ne pas s'éloigner de ses buts et ne pas rester trop près de ses coéquipiers. Chacun de ces facteurs est pondéré à l'aide d'un coefficient, et les valeurs de ces coefficients ont un impact sur les comportements des agents et le déroulement des différents matchs.

3.2 L'agent Ball

Cet agent représente le ballon de football. Ses attributs sont les suivants :

- Sa position grâce à ses coordonnées x et y
- Sa vitesse
- Son angle de déplacement
- Un booléen indiquant si un joueur possède la balle ou non

Tous les déplacements de la balle sont conditionnés par le joueur qui la possède. Lorsqu'un joueur réalise un tir, la balle se déplace avec l'angle de ce joueur mais avec sa propre vitesse ; cette vitesse décroît au fur et à mesure que la balle avance, réduite de moitié à chaque tour.

3.3 Les agents GoalKeeper

Ces agents représentent les gardiens des buts. Leurs attributs sont les suivants :

- La position grâce à ses coordonnées x et y
- La vitesse
- La zone d'interception, représentant l'intervalle dans lequel il peut intercepter le ballon.
- Un compteur de tours permettant au gardien venant d'intercepter la balle d'attendre quelques tours avant de relancer le jeu.

Les actions et déplacements du gardien sont réalisées dans l'objectif de maximiser sa fonction d'utilité, et d'interagir avec ses coéquipiers possédant l'utilité la plus élevée. Le gardien agit également selon sa probabilité d'intercepter la balle, calculée au tour i pour le tour $i + 1$.

4 Simulations et résultats

Les simulations ont été réalisées pour les deux types de fonction d'utilité mis en place.

4.1 Cas de la fonction d'utilité simple

On peut observer les résultats suivants :

Match 1

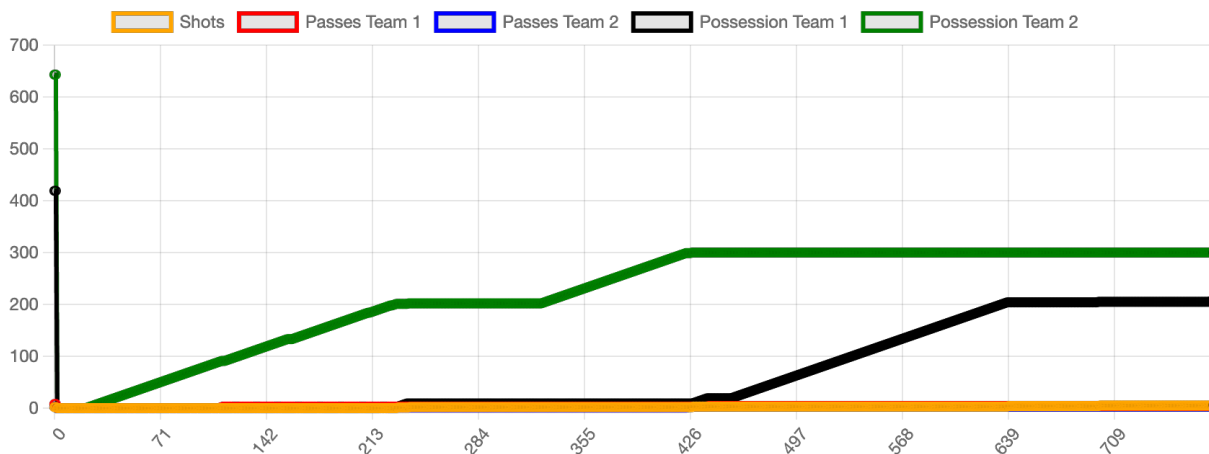


FIGURE 3 – Utilité simple - Match 1

Equipe	Possessions	Passes
1	206	4
2	300	3

Le gagnant de ce match a été l'équipe 1.

Match 2

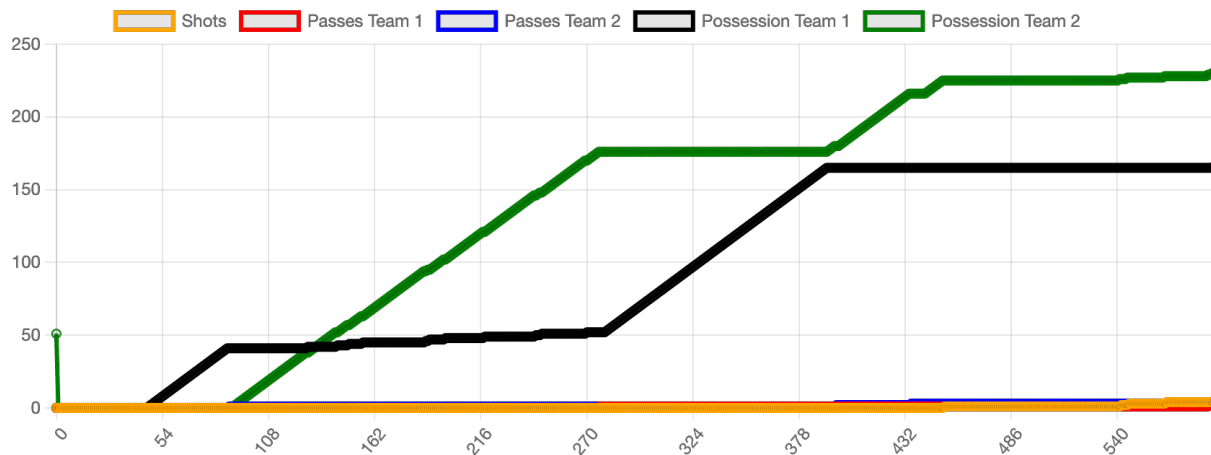


FIGURE 4 – Utilité simple - Match 2

Equipe	Possessions	Passes
1	166	2
2	230	4

Le gagnant de ce match a été l'équipe 2.

L'on remarque que lorsque les agents utilisent cette fonction d'utilité, le nombre de passes est très faible, et les temps de possession assez long, comportement n'étant pas très représentatif d'un vrai match de football.

Cela peut s'expliquer par la simplicité de la fonction d'utilité, qui ne prend pas en compte la distance à la balle ou le fait que les attaquants doivent s'écarter de leur coéquipier pour recevoir une passe. Ainsi, les fonctions d'utilité des coéquipiers ne sont pas assez évolutives, et le joueur la possédant la garde pendant trop de temps.

4.2 Cas de la fonction d'utilité élaborée

On peut observer les résultats suivants :

Match 1

Equipe	Possessions	Passes
1	123	70
2	217	82

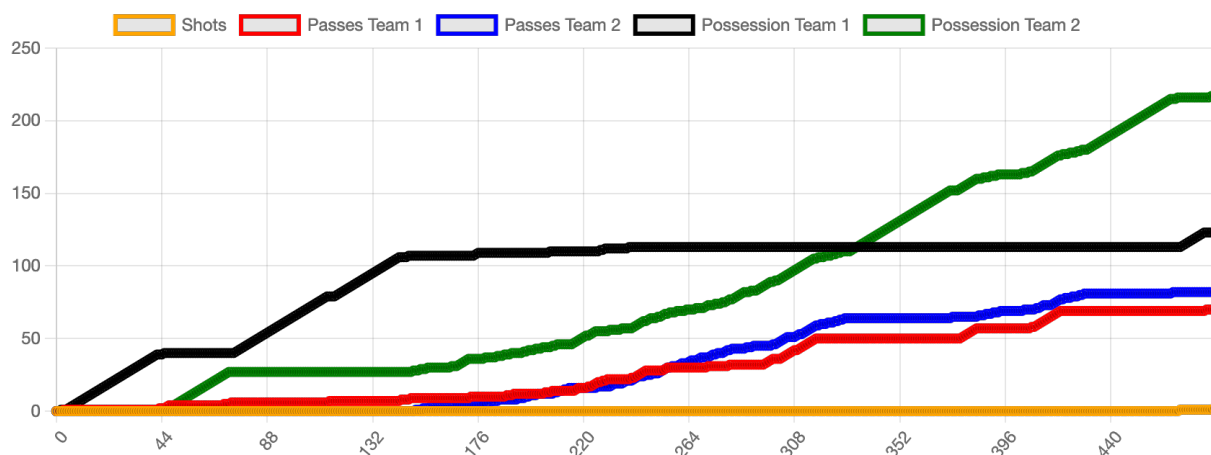


FIGURE 5 – Utilité élaborée - Match 1

Le gagnant de ce match a été l'équipe 2.

Match 2

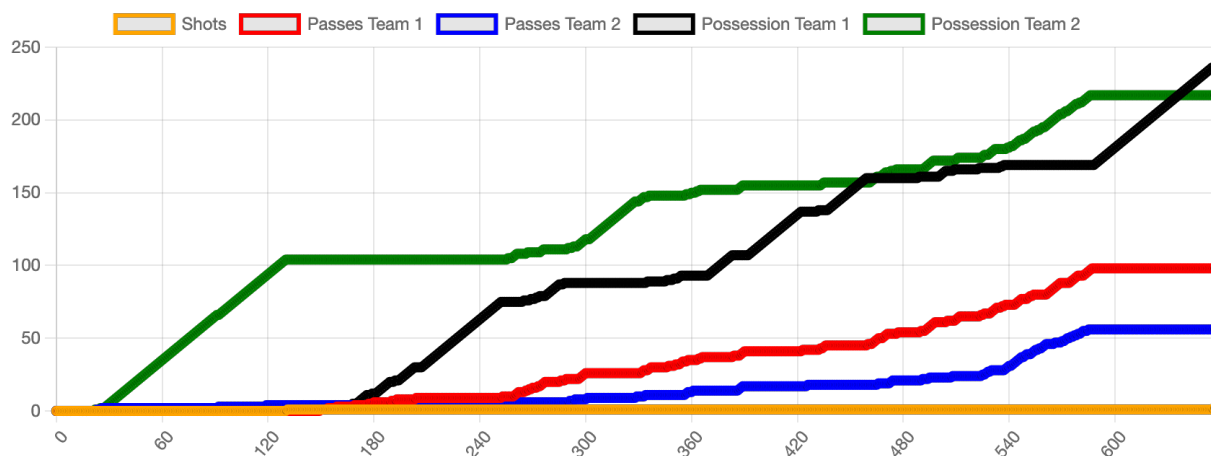


FIGURE 6 – Utilité élaborée - Match 2

Equipe	Possessions	Passes
1	236	98
2	217	56

Le gagnant de ce match a été l'équipe 1.

L'on remarque que lorsque les agents utilisent cette fonction d'utilité, le nombre de passes est plus élevé, et les matchs sont plus dynamiques. On observe des comportements plus représentatifs d'un vrai match de football. On remarque que de façon générale, l'équipe ayant le temps de possession le plus élevé gagne le match. On peut donc déduire qu'il est nécessaire et important de prendre en compte différents facteurs dans le calcul de la fonction d'utilité, afin de reproduire au mieux les raisonnements et comportements des joueurs de football.

Cependant, les valeurs des poids des différents facteurs de la fonction d'utilité ont un impact important sur le déroulement des matchs. Par exemple, en réduisant le poids du facteur de la distance au ballon en défense (les défenseurs perdent des points s'ils sont trop éloignés du ballon), on observe les résultats suivants :

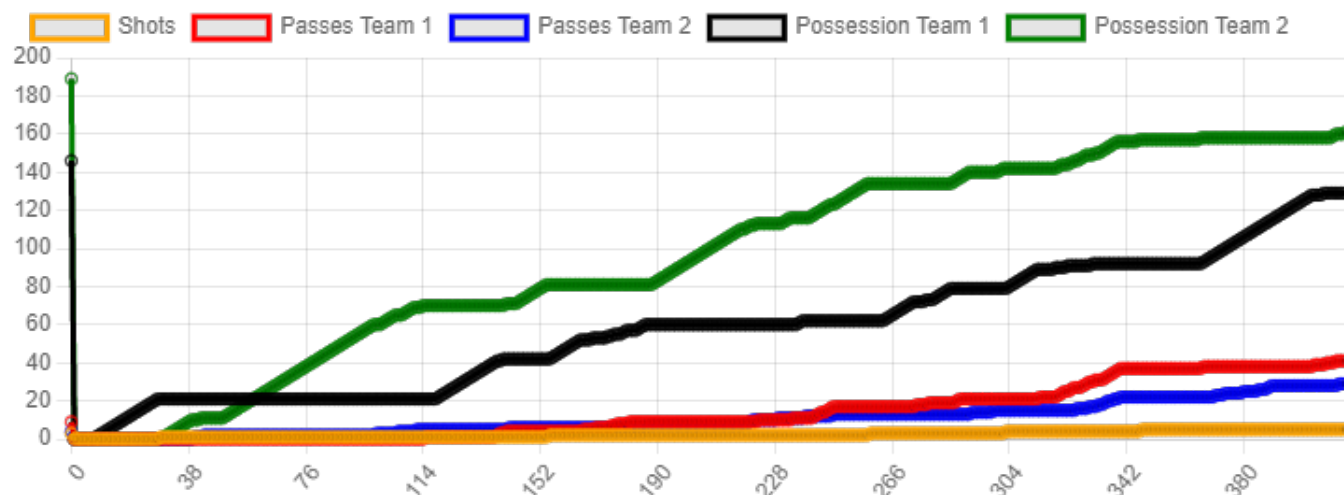


FIGURE 7 – Utilité élaborée - Facteur distance au ballon réduit

Equipe	Possessions	Passes
1	129	41
2	161	29

Comme nous pouvons le voir, la réduction de l'importance de la distance au ballon pour les défenseurs entraîne la réduction du nombre de passes observées durant le match. En effet, les défenseurs essayant moins de se rapprocher du ballon, ils marquent moins les attaquants adverses qui se retrouvent donc moins obligés de faire la passe à d'autres coéquipiers.

Une autre illustration de l'importance de la valeur des poids des facteurs est l'apparition de comportements émergents lorsqu'un facteur est mal pondéré. Par exemple, en essayant de réduire l'importance de la distance du joueur à ses coéquipiers (le joueur ne doit pas rester trop près de ses coéquipiers), on obtient le comportement suivant :

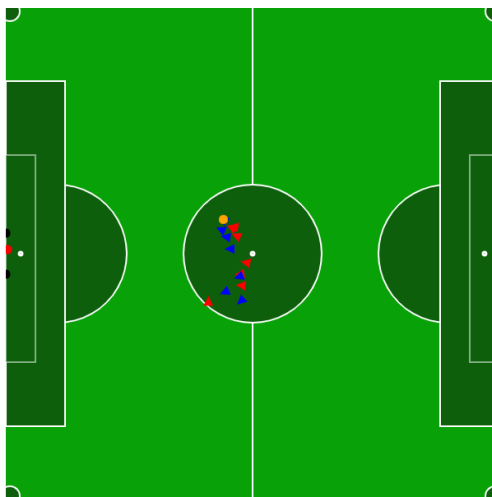


FIGURE 8 – Utilité élaborée - Facteur distance aux coéquipiers réduit

L'on remarque que les joueurs se retrouvent entassés dans un endroit du terrain, comportement très peu représentatif d'un vrai match de football. De plus, ce type de situation entraîne des moments de latence du joueur, car les valeurs d'utilité de ses coéquipiers se retrouvent à peu près toutes égales, et il est donc difficile de décider à quel coéquipier faire la passe.

5 Consignes pour l'exécution

Chaque type d'agent ainsi que le terrain de football ont été implémentés dans des fichiers Python différents, tous appelés ensuite dans le main. Pour exécuter le projet il faut donc télécharger le dossier puis exécuter le fichier `main.py`.

Une fois l'interface lancée, plusieurs paramètres peuvent être changés à l'aide des sliders : le nombre de joueurs, la vitesse des joueurs, ainsi que les poids des différents facteurs de calcul de l'utilité des joueurs.

6 Contribution des membres

Nous avons opté pour un travail en équipe en nous réunissant plusieurs fois pour réfléchir à nos idées et les implémenter. Chaque membre a donc participé à toutes les mailles du projet, de l'implémentation des comportements des agents à la réalisation du rapport en passant par l'implémentation graphique du terrain.

7 Conclusion

Nous avons pu simuler un match de football avec un système à multi agents réactifs. Malgré l'absence d'état interne et de communication directe entre les agents, nous avons observé des comportements émergents tels que des passes en retraits, plutôt que d'avancer seul ou tirer directement. Nous avons également observé une grande diversité de scénarios possibles.

Mais nous avons pu voir les limites l'architecture réactive, notamment la complexité et le temps nécessaire pour programmer toutes les règles. Dans le temps qui nous était imparti, nous avons fait le choix d'abstraire toutes les règles de bas niveaux avec une fonction d'utilité à maximiser. Même si cette méthode paraît à priori plus simple, le choix des fonctions et variables reste arbitraire et difficile à déterminer.

La seconde limite de l'approche réactive est l'absence de mémoire et donc de planification, or le football est un jeu de stratégie en équipe qui fait souvent appel à des actions en plusieurs étapes, et entre plusieurs joueurs. Par exemple, le "une-deux" consiste à faire une passe puis se dégager des défenseurs pour redemander le ballon. Nous avons pu néanmoins observer ce genre d'actions sous la forme de comportement émergent car elles maximisent naturellement l'utilité des joueurs.

En solution à ces limites, nous pourrions opter pour deux solutions.

La première serait de remplacer notre fonction d'utilité par une structure hiérarchique et plus de règles. Par exemple, "ne pas tirer si le joueur se trouve trop en dehors de l'axe" serait une sous-tâche du groupe "tirer". Cette nouvelle structure nous aiderait à implémenter plus de règles et obtenir des agents plus performants.

La seconde approche, celle de l'équipe de recherche de Google [Kur+19], est d'utiliser un modèle de Reinforcement Learning pour apprendre la fonction d'utilité optimale. Il est néanmoins très difficile d'appliquer ce genre de méthode à des problèmes multi agents. En effet, l'environnement présenté dans le papier utilise un modèle à un agent unique pour le porteur du ballon et un algorithme rule-based pour les autres joueurs, avec un comportement simple (suivre le sens du jeu). En revanche, Kurach et al. ont observés que l'entraînement d'un modèle multi-agent, même si plus long, offrait de meilleures performances.

Références

[Kur+19] Karol KURACH et al. *Google Research Football : A Novel Reinforcement Learning Environment*. 2019. DOI : 10.48550/ARXIV.1907.11180. URL : <https://arxiv.org/abs/1907.11180>.