



COMPTE-RENDU

Soccer Simulator

Tang Fabien, Delgado Thomas



Sommaire

Introduction.....	2
Stratégies.....	2
1v1	2
2v2	2
4v4	3
Améliorations	5
Améliorations générales.....	5
Optimisation de paramètres (Observer)	6
Apprentissage supervisé (Arbre de décision).....	7
Conclusion	7

Introduction

Lors de ce projet, nous avons appris à coder une intelligence artificielle (IA) tout en essayant de l'améliorer grâce à l'apprentissage statistique. Ainsi, après avoir étudié le fonctionnement du moteur d'un jeu de foot lors des premières semaines, notre objectif fût de coder l'IA des joueurs.

Stratégies

Afin de coder des stratégies intéressantes, nous avons défini dans un premier temps une classe « Comportement » censé imiter le comportement d'un joueur de foot de façon simplifié. Notre classe comportement est ainsi constitué de :

- une stratégie offensive,
- une stratégie défensive et
- un sélecteur (de stratégie).

Le sélecteur analyse en permanence l'état du jeu et renvoi un booléen en fonction de la position du ballon (quarts, tiers, milieu de terrain...). Le booléen renvoyé permet de déterminer la stratégie à adopter (offensive ou défensive).

1v1

Pour le 1v1, nous avons d'abord codé une stratégie du type « fonceur » c'est-à-dire un code ordonnant au joueur de foncer vers le ballon puis de shooter vers le but adverse. Au fur et à mesure des séances, nous l'avons amélioré en ordonnant au joueur de « dribbler » sur les 3 quarts de terrain (=réaliser des shoots pas très puissants vers le but adverse) puis de shooter.

2v2

Notre 2v2 est composé d'un gardien et d'un attaquant :

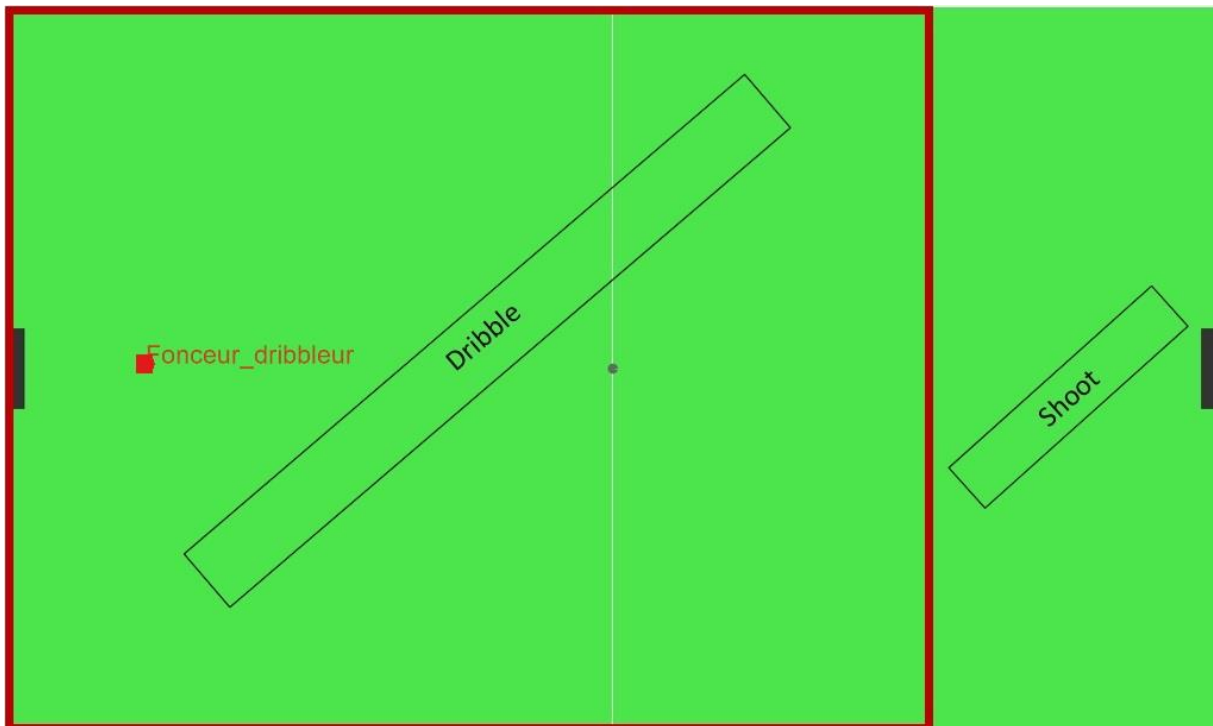
- Le gardien reste devant son but quand le ballon n'est pas dans son quart de terrain (=zone du gardien) et shoot le ballon vers le coéquipier le plus proche quand il en a la possibilité.
- L'attaquant reprend le même principe du fonceur 1v1 à l'exception qu'il se dirige vers le milieu de terrain (plus précisément en $(\text{GAME_WIDTH}/2, 3*\text{GAME_HEIGHT}/4)$) quand le ballon se trouve dans la zone du gardien.

4v4

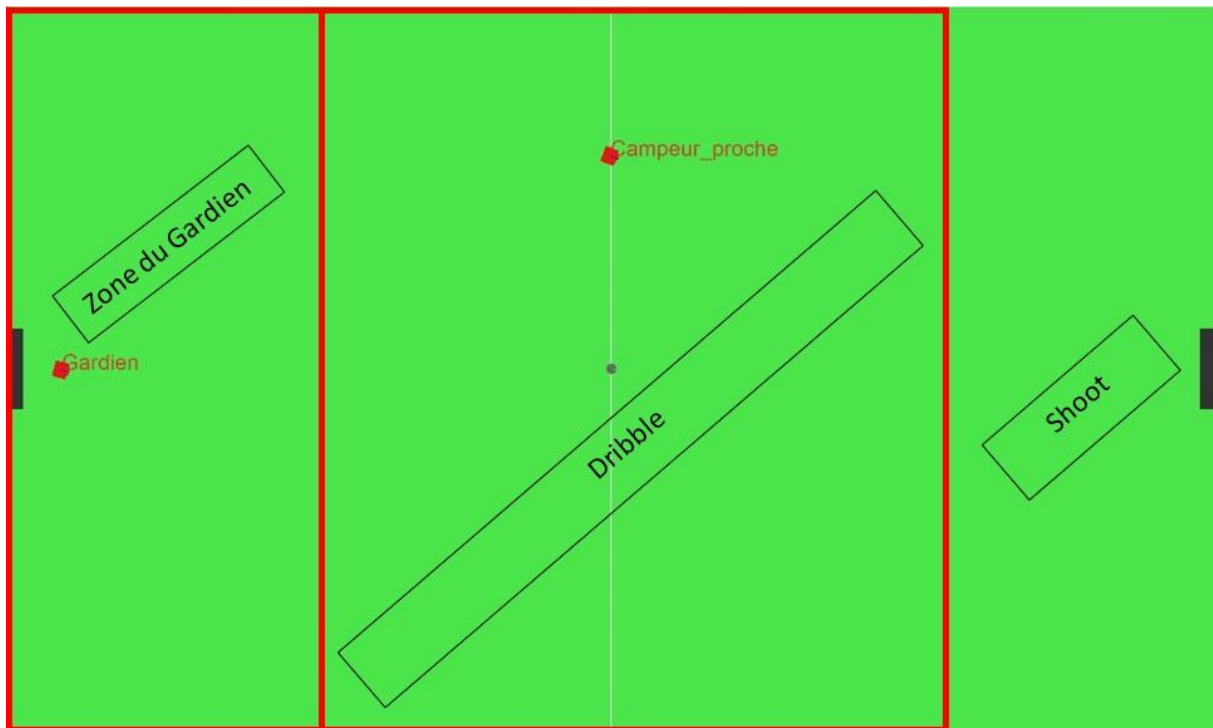
Le 4v4 est composé d'un gardien, d'un passeur et de deux attaquants :

- Le gardien est identique que celui du 2v2
- Le premier attaquant est identique que celui du 2v2
- Le second attaquant est identique au premier à l'exception qu'il se dirige en $(\text{GAME_WIDTH}/2, \text{GAME_HEIGHT}/4)$
- Le passeur force vers le ballon et ne shoot que vers le coéquipier le plus proche.

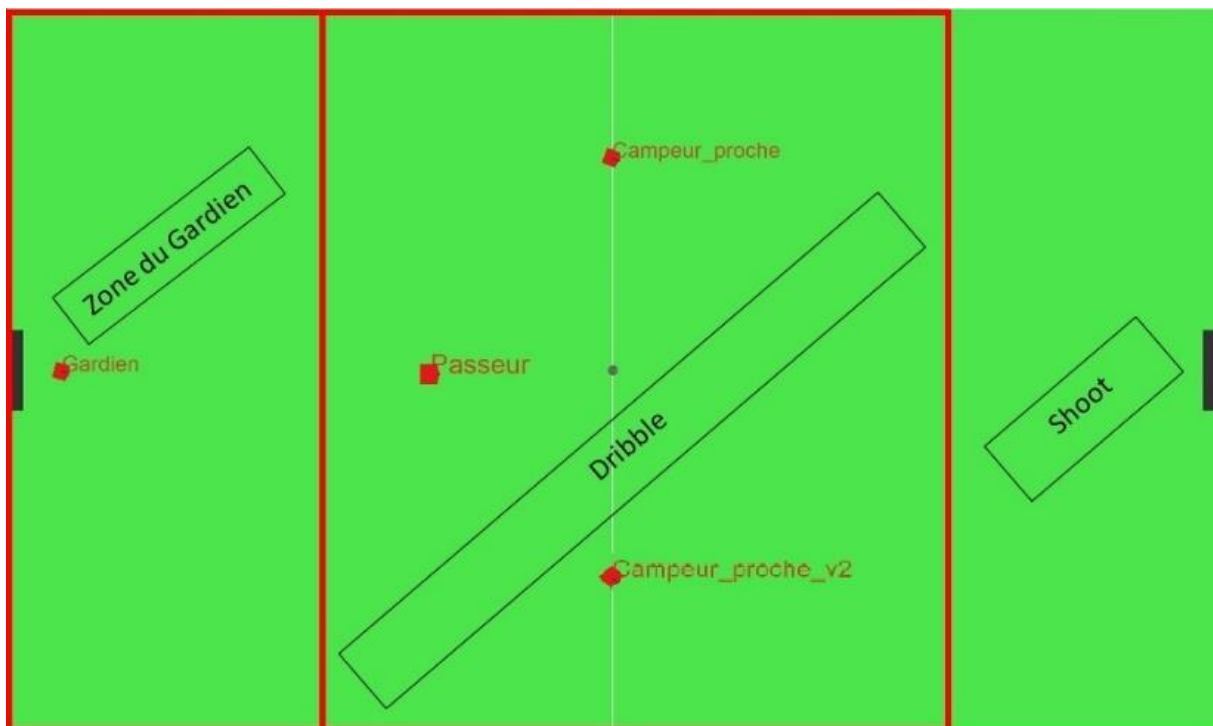
1v1



2v2



4v4



Améliorations

Une fois nos stratégies faites, nous avons dû les améliorer et les optimiser.

Améliorations générales

Dans un premier temps, nous avons fait en sorte que nos joueurs se dirigent vers le vecteur vitesse du ballon (et non vers le ballon) puis nous avons codé certaines fonctions afin d'améliorer certaines de leurs actions :

- Peut_shooter() vérifiant la distance entre le joueur et la balle et renvoyant un booléen indiquant si on peut shooter ou non,
- Aller_courir_marcher() permettant de courir vers le ballon et de ralentir un peu avant de shooter pour être plus précis.

Nous avons modifié notre stratégie 1v1 en instaurant un temps d'attente :

- Au début de la partie tant que le ballon est au milieu pour récupérer le ballon plus facilement après que l'adversaire ait shooté dedans.
- Au moment où notre joueur va commencer à dribbler.
➔ Le joueur dribble quand l'adversaire se trouve à une certaine distance de lui afin de pouvoir le dépasser et d'aller marquer plus facilement.
(voir schéma)



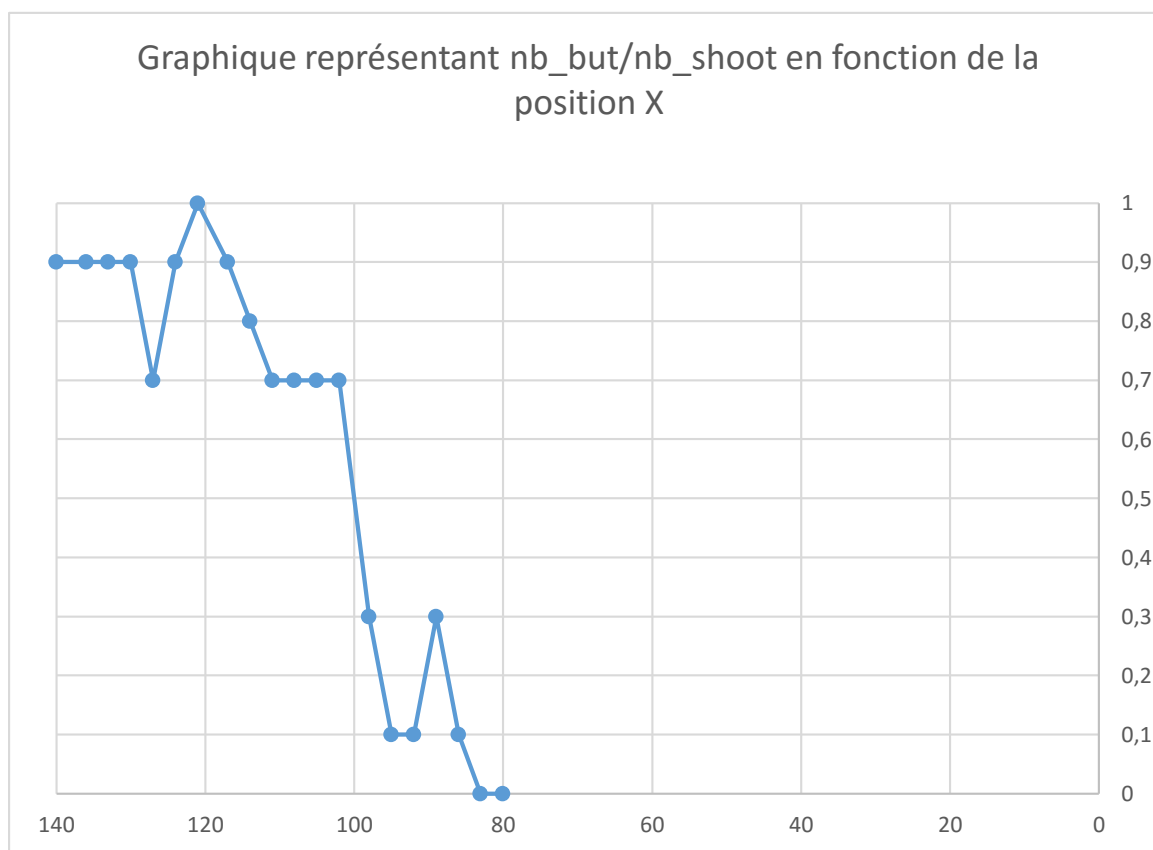
Pour notre 2v2, nous avons amélioré notre gardien en l'orientant par rapport au ballon (il se place entre le ballon et le milieu de son but) et en le faisant sortir quand le ballon se trouve dans le camp adverse.

Néanmoins le fait de faire sortir le gardien n'améliore pas les choses (classement -6 places) nous en avons donc conclu que cela n'était pas utile.

Optimisation de paramètres (Observer)

Au niveau optimisation, nous avons déterminé la zone de shoot optimal des attaquants en faisant tourner plusieurs simulations et modifiant la distance entre la balle et le but adverse. Le critère d'évaluation était donc le nombre de buts marqués sur le nombre de tirs réalisés.

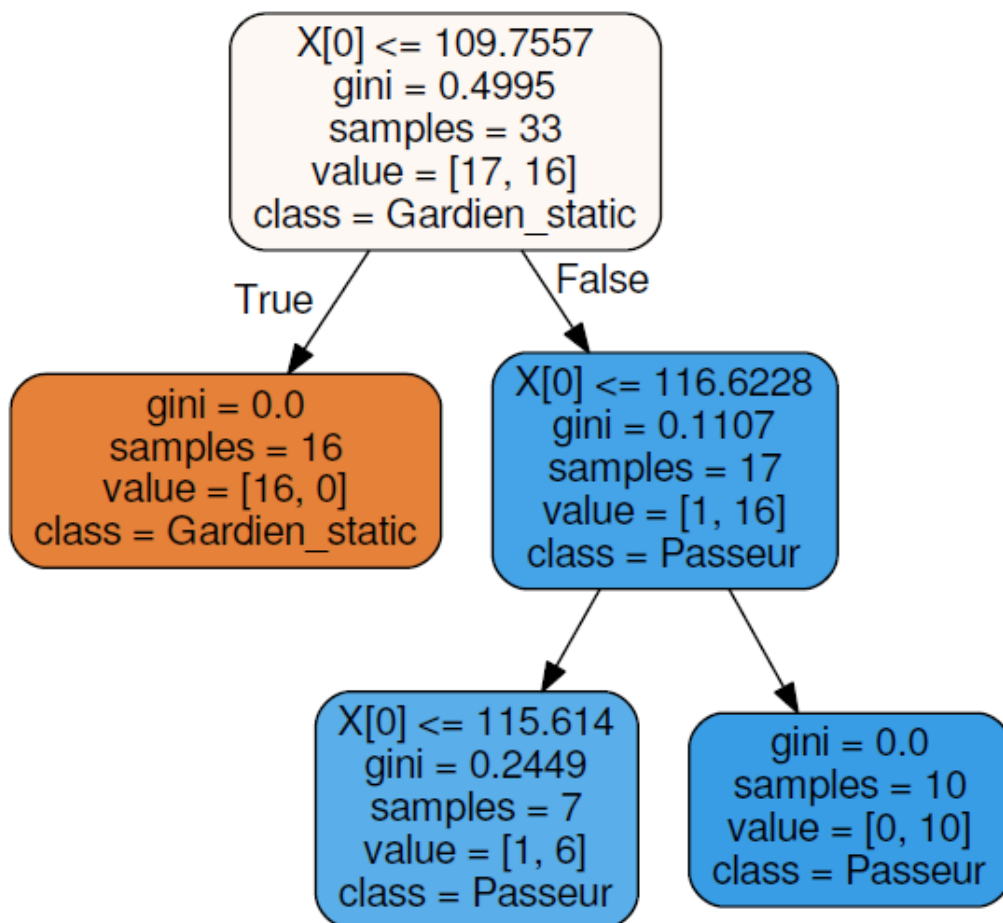
Position en X	140	136	133	130	127	124	121	117	114	111	108	105	102	98	95	92	89	86	83	80
nb_but/nb_shoot	0,9	0,9	0,9	0,9	0,7	0,9	1	0,9	0,8	0,7	0,7	0,7	0,7	0,3	0,1	0,1	0,3	0,1	0	0



D'après les résultats, nous en avons conclu que la zone de shoot optimal s'étale de X = 114 à X = 140.

Apprentissage supervisé (Arbre de décision)

Enfin, nous avons utilisé l'apprentissage supervisé pour redéfinir la zone du gardien manuellement :



D'après l'arbre, si le ballon se trouve en 150-109 (en x), le joueur adopte la stratégie passeur (c'est-à-dire qu'il sort de son but pour foncer vers la balle et faire une passe) sinon il reste gardien.

Conclusion

En conclusion, ce projet nous a permis d'apprendre de manière ludique à coder, améliorer et optimiser des IA imitant des joueurs de foot. Les stratégies peuvent toujours être améliorées en passant par exemple plus de stratégies dans la classe comportement ou en changeant/rajoutant d'autre critère de sélection de stratégie comme la position du joueur adverse, le score etc...