Intelligence artificielle

Programme d'intelligence artificielle de Robocup Laval : une présentation technique.

Par Frédérick Giasson (fred [at] fgiasson.com)

Le programme d'intelligence artificielle du projet de robotique Robocup Laval, est basé sur des principes, des algorithmes et des méthodes amplement étudiés dans la littérature. La ligne directrice de développement de ce programme est le système BDI (Belief-Desire-intention) (Wooldridge 2000).

1- Le concept

Le ***Belief*** correspond à l'information que l'agent (dans la situation qui nous occupe présentement, un robot joueur de soccer) sait à propos du monde, de l'environnement dans lequel il évolue aux temps passé, présent et futur (Wooldridge 2000). Nous ferons référence à ce concept en tant qu'état du jeu. Ce que le joueur sait ou plus précisément, ce qu'il croit savoir sur le monde dans lequel il évoluait et évolue toujours. Nous faisons référence à ce que le joueur croit savoir puisqu'il est loin d'être sûr que ses perceptions et son interprétation soient efficaces à 100%.

Le **Desire** représente l'état dans lequel l'agent voudrait se retrouver dans un monde idéal. (Wooldridge 2000). Dans le présent cas, le but que le joueur se fixe.

Le *belief* d'un joueur, c'est l'état dans lequel il est à un temps présent (pour nous le temps présent c'est le *frame* de vision en cours d'analyse). Il faut bien préciser qu'il n'est peut être pas dans le même état que nous le percevons, nous humains. Mais plutôt dans l'état qu'il se perçoit lui, ainsi qu'avec les aberrations que ses sens peuvent lui donner par rapport au « monde réel » tel que perçu par nous. À l'aide de ce *belief*, l'agent va se donner des buts à atteindre pour accomplir des tâches dans le monde où il évolue. Ce sont ces buts « idéalistes » qu'il tentera d'atteindre tout au long de son évolution.

L'***intention*** représente les engagements que l'agent a pris pour atteindre son *desire*. (Wooldridge 2000). Nous ferons référence à ce concept en t'en qu'opération effectuée par le joueur. Ce sont les opérations pensées et exécutées par les agents pour atteindre leurs buts fixés préalablement pour arriver à l'état du monde dans lequel il voulait évoluer initialement.

Voici un exemple possible qui pourrait survenir lors une partie :

« Un de nos attaquant reçoit la balle d'une certaine façon. À partir de ce moment, il perçoit que son monde a changé. Il perçoit qu'il a possession de la balle, qu'il a des alliés potentiels qui peuvent l'aider à sa tâche et qu'il a des ennemis qui vont tout faire pour le faire échouer. À partir de ce moment, notre joueur s'est fait une représentation de son monde, selon les informations qu'il a eu (il est possible que les informations ne soient pas véridiques et qu'il ne soit en fait aucunement en possession de la balle.). Par la suite, selon sa situation, il va se donner un but à atteindre. Ici, son but évident est d'aller compter un point dans le but adverse. Il fera tout ce qui est en son pouvoir pour atteindre ce but. Il pourra accomplir différentes actions pour l'atteindre le plus efficacement. Ce sont ces *intentions* qui feront évoluer le jeu. Cependant, comme son but n'est qu'un *belief*, il est fort possible que ne l'atteigne pas. Par exemple, si un joueur adverse prend possession de sa balle, alors il ne pensera qu'à la récupérer et non de pas continuer à aller compter avec une balle qu'il ne possède pas. »

Ce petit exemple démontre bien le concept BDF.

2- Les agents

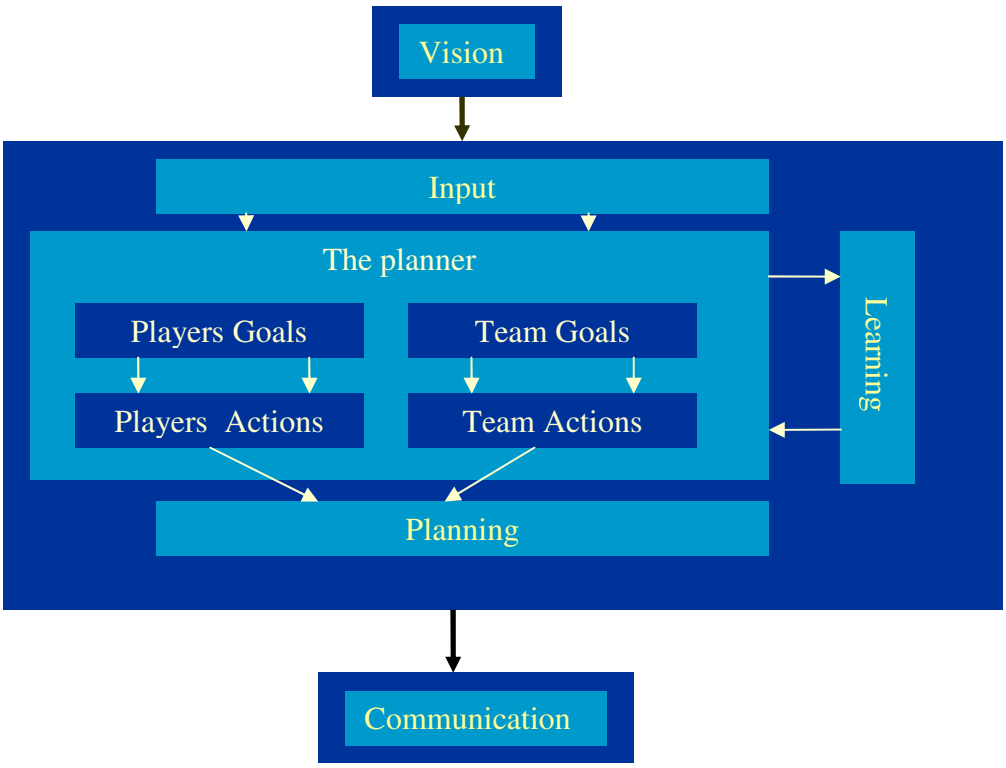
Comme le veut notre concept décrit ci-haut, notre système fonctionne par l'interaction des agents entre eux. Dans notre système, il y a deux types d'agents qui évoluent. Premièrement, il y a les agents joueurs, qui comprennent le gardien de but, les défenseurs et les attaquants. Deuxièmement, il y a l'agent entraîneur.

L'agent joueur est celui qui fera les actions dans le monde réel. Chaque robot est en fait un agent. Ces agents ont une pleine autonomie. Ce sont eux en définitive, qui décident de l'action qu'ils feront pour atteindre des buts propres à chacun d'eux. Chaque agent évolue différemment dans le monde et par le fait même ils sont chacun dans un état différent. Ils possèdent également leur but personnel par rapport au jeu. Chacun décide donc de son sort, et prend la décision qu'il croit la meilleure pour atteindre son but selon la situation dans laquelle il est. Bien que le niveau d'individualité des agents soit élevé, cela n'empêche pas une très forte coopération entre les agents et malgré qu'ils aient leurs propres buts et stratégies, ils sont également enclin à mettre à terme les buts et stratégies de l'équipe.

Les buts et les stratégies de l'équipe sont pensés et suggérés par notre deuxième type d'agent, l'agent entraîneur. Cependant, il ne faut pas perdre de vue qu'ils seront mis en œuvre par les agents joueurs. Chaque agent évolue différemment dans le jeu, cependant, ils n'ont pas toujours une vision globale du jeu et ne savent pas toujours où sont leurs coéquipiers puisque la communication inter-agents est plutôt faible. C'est alors que l'agent entraîneur vient nous aider à prendre des décisions qui tiennent compte de l'ensemble du jeu, des joueurs adverses et des joueurs de notre propre équipe. Cet agent ne fait que suggérer des actions que les joueurs pourraient exécuter. Il leur suggèrera une action, qu'ils pourront exécuter pour atteindre un but d'équipe. Cependant, ce sont ces derniers qui décideront s'ils prennent en compte ou non, dépendant de l'utilité, dans le jeux courant, de cette action qui a été suggérée par l'entraîneur.

3- Le système

Voici, ci-dessous un schéma décrivant l'interaction des grands modules qui composent notre système d'intelligence artificielle.



3.1- Module d'Input

Ce module reçoit et gère toutes les informations en provenance du système de vision. Par la suite, le module d'Input redirige les informations au *module de Décision* seulement s'il est prêt à les recevoir. Dans le cas où le *module de Décision* n'est pas prêt à les recevoir, les informations seront sauvegardées au cas où il aurait le temps de les analyser avant le prochain envoie de données.

3.2- Module de Décision

La fonction de ce module est de formuler des requêtes d'actions potentielles provenant de chaque joueur ainsi que de l'équipe afin de les acheminer au *module de Planning*. C'est lorsque ces décisions seront prises que les stratégies de chaque joueur et de l'équipe apparaîtront lors du match. Lorsque que l'équipe et les joueurs auront fait leur choix potentiel d'actions à exécuter, ils communiqueront leurs requêtes au *module de Planning*.

3.3- Module de Planning

La fonction du *module de Planning* est de déterminer par un autre processus de décision quelle action chacun des joueurs doit exécuter ainsi que faire l'élaboration d'un plan d'action(s) pour arriver à atteindre leur but. Quand les décisions finales seront prises et que les plans d'actions seront terminés, le *module de Planning* enverra ses requêtes au *module d'Output.*

3.4- Module d'Output

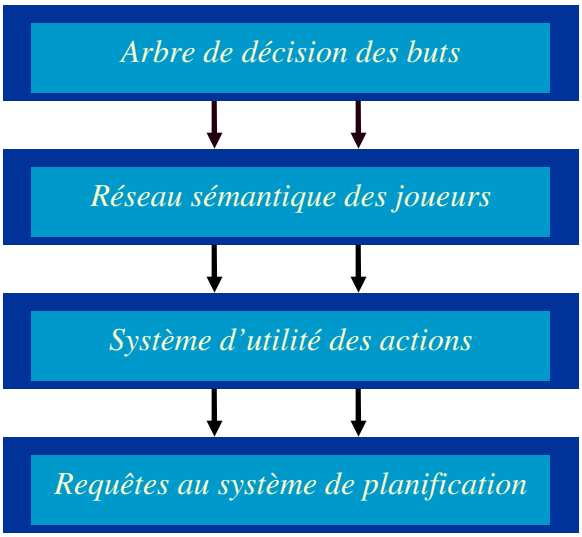
La fonction du *module d'Output* est d'envoyer les requêtes provenant du *module de Planning* et de les rediriger vers le programme de communication qui les enverra par la suite aux robots.

3.5- Module de learning

La fonction *du module Learning* est d'apprendre des informations sur les performances et les méthodes de jeu de l'équipe adverse. Lorsque de nouvelles données seront apprises, par exemple la vitesse moyenne de mouvements des robots adverses, nous pourrons prendre en compte ces nouvelles informations dans nos algorithmes de décisions. Pour le moment, ce module n'est pas implanté dans le programme d'intelligence artificielle et ne le sera pas pour la compétition de l'été 2004.

4- *Prise de décision des joueurs*

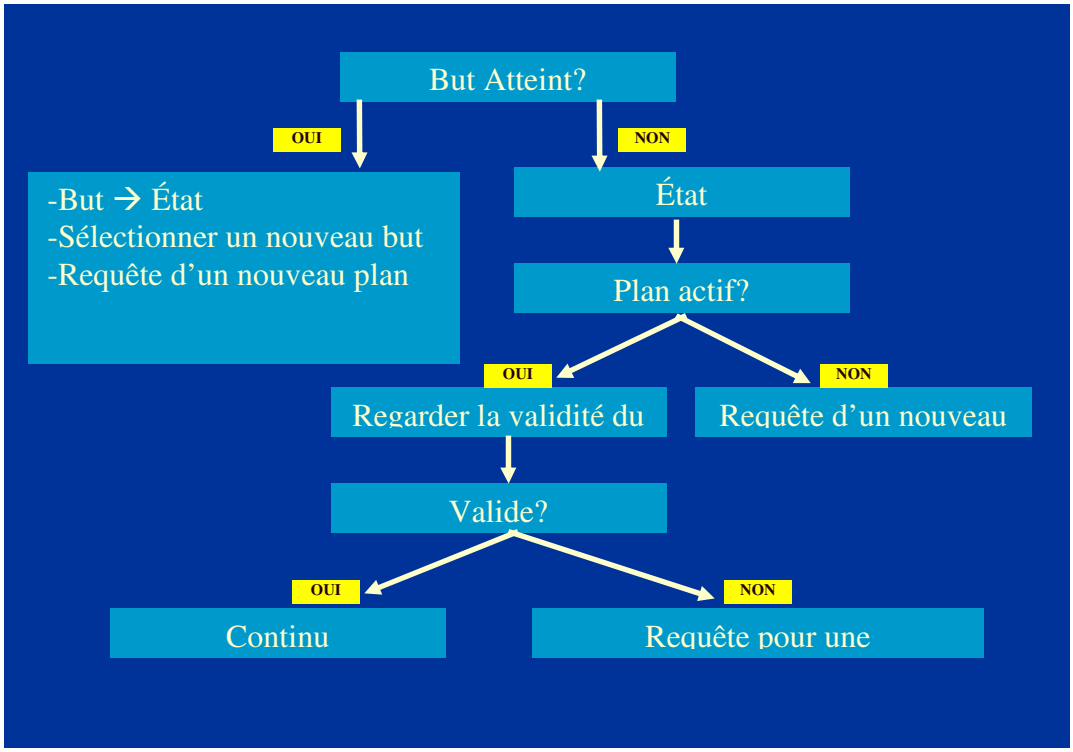
Nous allons maintenant explorer la structure et toutes les instances qui composent notre système de prise de décision. Regardons premièrement ce schéma :



4.1- *L'arbre de décision des buts*

L'arbre de prise de décision des buts est fort simple. Il est simplement utilisé pour savoir si un but est atteint, si un plan d'action est actif, si la validité du plan d'action est toujours valide. Si le but courrant d'un joueur est atteint, il en trouvera un autre dans le *Réseau Sémantique des Joueurs* et par le fait même, changera l'état du joueur. Si un plan d'action est terminé ou invalide, il fera une requête *au module de Planning* pour en avoir un nouveau ou pour refaire celui qui est invalide. Cet arbre est consulté à chaque nouvelle unité de temps. Dans notre système, une unité de temps représente une *frame* en provenance du système de vision.

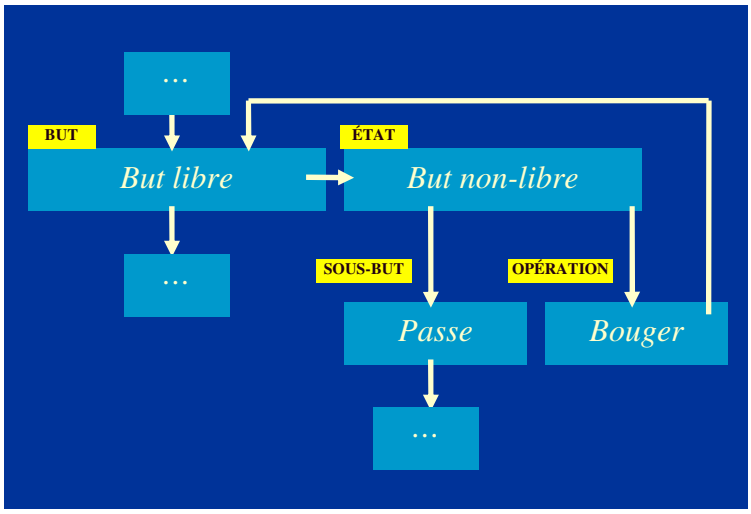
Voici un exemple d'une partie d'un arbre de prise de décision des buts :



4.2- Réseau sémantique des joueurs

Le réseau sémantique des joueur est également fort simple. Le but dudit réseau est de mettre en relation les buts, états et opérations de chacun des types de joueurs pour aider à la prise de décision.

Voici un exemple d'une branche du réseau :



4.3- Système d'utilité des actions

Le système d'utilité des actions est utilisé, comme son nom l'indique, pour avoir une valeur numérique de l'utilité d'une action dans une situation donnée. Ce système n'est pas seulement présent dans la prise de décision des joueurs, mais également dans celle de l'équipe. Il ne faut pas se tromper, nous ne parlons pas d'une fonction d'utilité tel que décrite dans (RUSSEL,NORVIG 1995) qui consiste à assigner un nombre qui représente la souhaitabilité d'un état futur, mais bien un nombre qui représente l'utilité d'une action dans un état présent.

Par exemple, si un de nos joueurs doit aller prendre la balle à un des joueurs adverse, il aura plusieurs manières d'y arriver. En fait, il y aura plusieurs actions qu'il pourra exécuter pour aller prendre cette balle. Nous devons savoir laquelle des ces actions est la plus utile dans une situation donnée. C'est alors que ce système vient nous aider pour la prise d'une telle décision. Il tiendra compte de l'état courant d'un de nos joueurs, il regardera les actions qu'il est apte à exécuter, il calculera l'utilité de cette action selon ces paramètres, nous retournera une valeur numérique qui nous aidera à trouver qu'elle action est la plus utile.

4.4- Requêtes au système de planification

Finalement, lorsque les meilleures actions à exécuter dans une situation donnée sont sélectionnées pour chacun des joueurs, nous allons faire la suggestion par le biais d'une requête au *module de Planning*. Il ne faut pas oublier, comme nous l'avons dit ci-haut, que les décisions prises par les joueurs et l'entraîneur ne sont que des suggestions, c'est le *module de Planning* qui prendra la décision finale en analysant les requêtes qui lui sont parvenues.

5- Prise de décision de l'entraîneur

Contrairement aux agents joueurs, l'agent entraîneur n'utilise pas de réseaux sémantiques pour aider à la prise de décision. Contrairement à eux, il utilise un système de cartes de jeux.

« Les cartes de jeux » est un système de décision s'appuyant sur l'utilisation des ensembles flous comprenant les zones de jeu, les joueurs adverses et la balle. L'intersection de ces ensembles avec des cartes de jeux préalablement construites nous aidera à prendre des décisions éclairées en prenant en compte l'état global du jeu.

5.1- Structure des cartes de jeux

Une carte est la représentation d'un terrain de jeu que nous séparons en 12 zones virtuelles. Chacune de ces zones est composée d'états possibles dans lesquelles cette zone pourrait éventuellement se retrouver. Chacun de ces états est également composé de plusieurs buts/opérations que nous pourrons possiblement exécuter selon l'état du jeu au moment que cet état est sélectionné pour cette zone. Notre système est composé d'une base de données de quelques 20 ou 30 cartes de jeux différentes qui tentent de rencontrer la plupart des situations dans lesquelles nos joueurs pourraient se retrouver. C'est à travers ces multiples cartes de jeux que nos stratégies d'équipe se développent et ressortent sur le terrain de jeu.

5.2- Fonctionnement d'une carte de jeux

À chaque fois que nous recevons une *frame* du système de vision, nous l'analysons. Donc, nous la séparons en 12 zones et nous calculons un certain potentiel de contingentement pour chacune des zones. Ce potentiel est calculer selon la position des joueurs adverses ainsi que de la balle. Il y a une propagation d'une zone à une autre d'un certain potentiel dégressif qui en définitive, nous indiquera le potentiel de contingentement de chacune des zones. C'est grâce à ces potentiels différents entre les joueurs adverses et la balle que nous pouvons focusser les actions de notre équipe sur un ou l'autre. Nous pouvons les faire davantage couvrir les joueurs ennemis ou alors les faire courir après la balle en ne changeant que la valeur du potentiel de propagation d'une zone à l'autre.

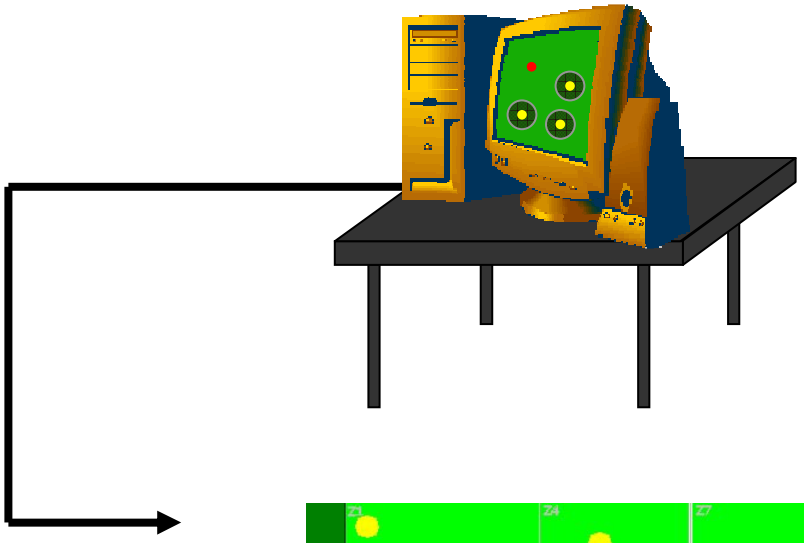
Par la suite, nous allons considérer chacun de ces potentiels comme un élément d'un ensemble. Nous allons créer deux ensembles distincts, un ensemble A qui sera l'ensemble des potentiels des 12 zones de notre jeu courant, et les ensemble $B_1 \ldots B_n$ qui sont l'ensemble des potentiels des 12 zones de chacune de nos cartes de jeux de notre base de donnée. Chacun de ces ensembles sont ordonnés et chacun de leurs éléments sont compris dans l'intervalle [0,1], 0 étant l'absence de contingentement, et 1 étant un maximum de contingentement.

Finalement, nous allons calculer le degré d'appartenance entre l'ensemble A et chacun des ensemble de $B_1 \ldots B_n$ en faisant l'intersection entre chacun de ces ensembles. Par la suite, nous avons 3 différentes formules pour calculer le degré d'appartenance entre chacun des ensemble après avoir effectué l'intersection desdits ensembles.

C'est lorsque nous avons découvert la meilleure carte de jeux à prendre en compte dans la situation de jeu courante que nous faisons la sélection des états/buts/opérations de cette carte, tel que discuté ci-haut. Finalement, nous retirer 4 différentes actions que nous allons formuler en requêtes au *module de Planification*. Il ne faut toujours pas perdre de vu que ce ne sont que des suggestions et que les joueurs ne sont pas obligés d'en tenir compte s'ils croient qu'ils ont une action plus utile à faire dans cette situation courante.

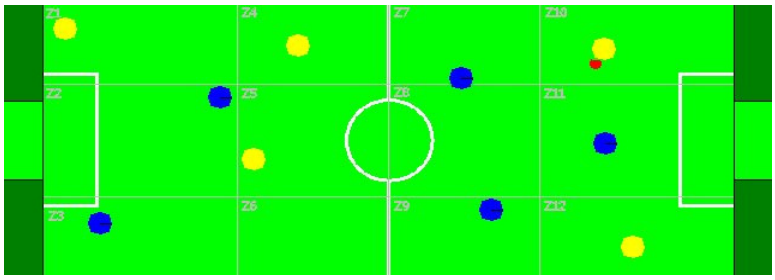
5.3- Démonstration du système

<u>Étape #1</u>

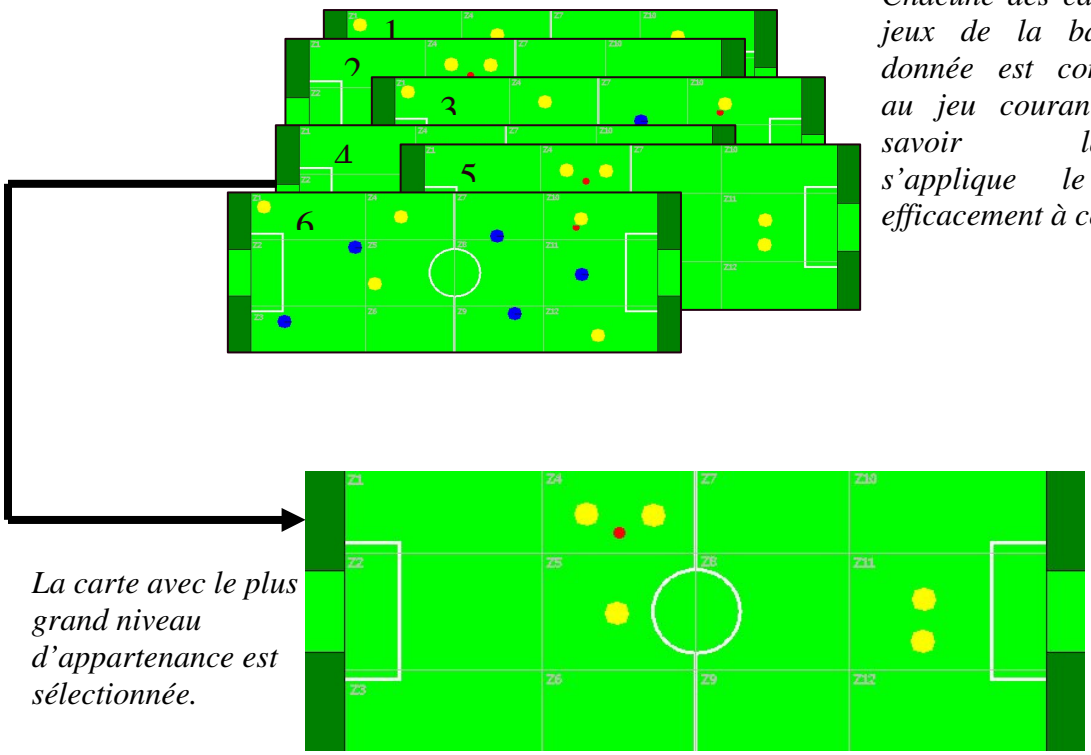


Le système de vision envoie un frame au programme d'IA pour fin d'analyse.

La frame est analysée par le programme d'IA.



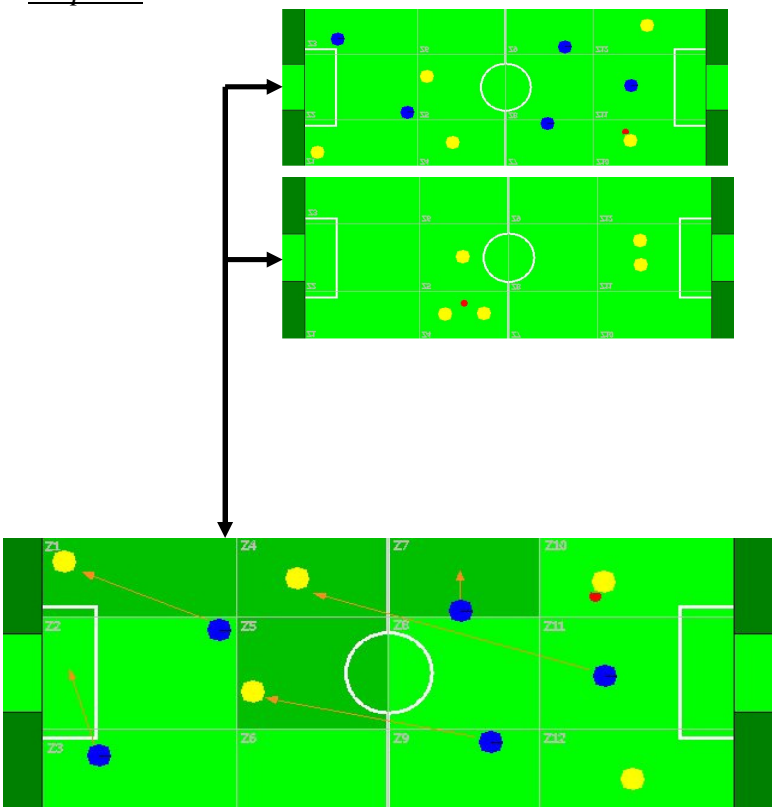
Étape #2



Chacune des cartes de jeux de la base de donnée est comparée au jeu courant pour savoir laquelle s'applique le plus efficacement à celui-ci.

La carte avec le plus grand niveau d'appartenance est sélectionnée.

Étape #3



Niveau d'appartenance de chacune des zones.

Sélection des 4 zones avec le plus haut potentiel d'appartenance entre la carte sélectionnée et le jeu courant.

6- Conclusion

Comme nous l'avons vu, le programme d'intelligence artificielle est basé sur le concept BDI. Chacun de nos robots sont des agents indépendants et autonomes prenant leurs propres décisions. Une certaine concurrence existe entre les agents qui composent notre système. Cette concurrence est visible lorsque nous regardons les requêtes qui sont parvenues au *module de Planification* et que nous faisons un choix en regardant l'utilité des actions soumises. C'est le fait de performer cette action qui crée une certaine concurrence entre les agents de notre système. Aucune négociation n'est mise en place pour gérer cette concurrence. Cependant nous pourrons éventuellement inclure ce phénomène dans une version future pour étudier le comportement de telles situations de concurrence dans le cadre d'une partie de soccer entre les robots d'une même équipe aillant pour but de vaincre une équipe adverse.