







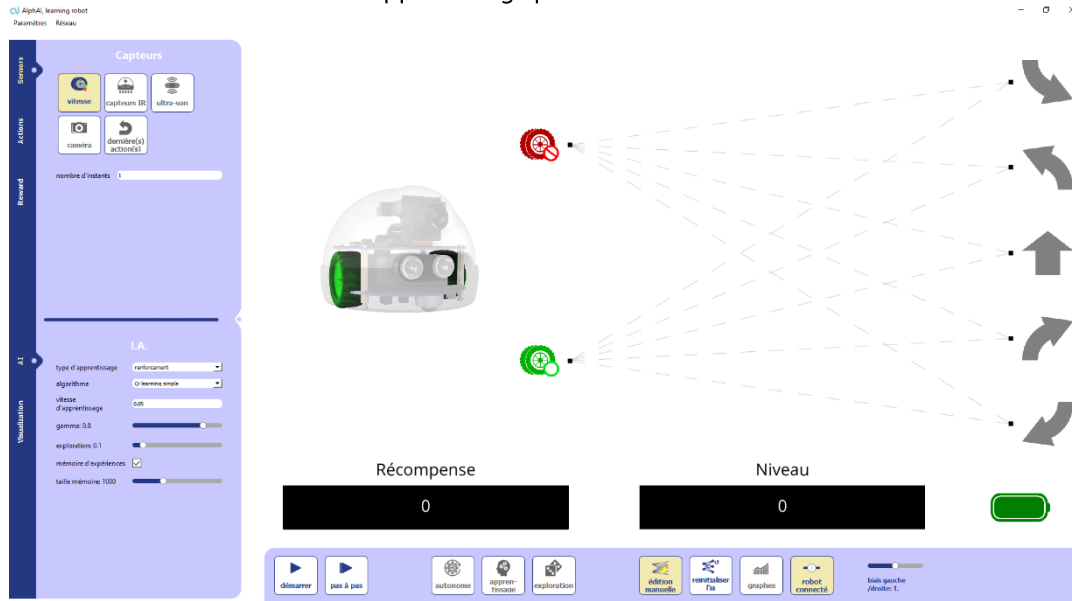
ATELIER « APPRENTISSAGE PAR RENFORCEMENT »

EDITION MANUELLE D'UN RESEAU DE NEURONES SIMPLE

Nous allons voir plus en détail comment fonctionne les connections dans le réseau de neurone du robot avec un cas simple.

Chargez la configuration « Edition manuelle d'un réseau de neurones simple ». Ou pour sélectionner vous-mêmes les paramètres, lancez le programme puis :

- Dans onglet IA :
 - Sélectionnez q-learning simple
- Dans l'onglet Capteurs, sélectionnez :
 -  « bloqué/en mouvement »
- Dans l'onglet Récompense :
 - choisissez « grande vitesse, punition à l'arrêt » 
- Dans l'onglet du bas :
 - cochez le mode « édition manuelle simple » 
 - cliquez sur « réinitialise le réseau » 
 - décochez apprentissage  et exploration 



Les entrées possibles sont :

- Le robot n'est pas bloqué
- Le robot est bloqué

En sortie, 5 actions sont possibles : reculer à droite, tourner à gauche, aller tout droit, tourner à droite, reculer à gauche.

Ici il n'y a pas d'apprentissage, c'est à **vous de jouer** en créant les liens adéquats entre ces entrées (les capteurs) et ces sorties (les actions) pour générer un comportement cohérent du robot. Si vous créez un lien entre un capteur et une action, cela signifiera que dès que le capteur est activé, le robot effectuera l'action qui y est relié.



Tracez les connexions que vous pensez être bonnes, puis démarrez **démarrer** le robot. Vous pouvez également modifier les connexions une fois le robot démarré.

Objectif : Mettre en place les connexions qui permettent au robot d'obtenir le plus de récompense et donc d'atteindre le niveau le plus élevé possible. **Indice :** le robot reçoit des récompenses lorsqu'il va en avant.

Variante : pour ajouter un peu de complexité, on peut prendre en compte les ultra-sons. Dans l'onglet



capteur rejoutez absence/présence d'obstacle

Notions apprises :

Pour que le robot ait un comportement, il faut créer des connections entre ces capteurs et les différentes actions qu'il peut faire.

APPRENTISSAGE PAR RENFORCEMENT - NAVIGATION BLOQUE/ EN MOUVEMENT


Précédemment nous avons-nous même créé le réseau de connexions pour que le robot agisse comme on le voulait. A présent nous allons voir qu'on peut utiliser l'intelligence artificielle pour que le robot parvienne de lui-même a construire ce même réseau sans intervention humaine. Pour cela on va se servir de l'apprentissage par renforcement.

Le principe de l'apprentissage par renforcement est d'encourager le robot à faire certaines actions en lui donnant des récompenses ou des punitions. Le robot modifiera alors son comportement de façon à maximiser son niveau, qui est la moyenne de ses récompenses sur les 2 dernière minutes.

Ici, plus le robot ira vite, plus on lui donnera de récompense, mais s'il s'arrête, on lui donnera une punition. Pour maximiser son niveau le robot aura intérêt à aller tout droit le plus souvent possible et à s'arrêter le moins souvent possible.

Chargez la configuration « apprentissage par renforcement – navigation bloqué/ en mouvement».

Ou pour sélectionner vous-mêmes les paramètres, lancez le programme puis :

- Dans l'onglet IA :
 - Sélectionnez Q-learning simple
- Dans l'onglet Capteurs, sélectionnez :
 -  « bloqué/en mouvement »
- Dans l'onglet du bas :



- décochez exploration


- Dans l'onglet Récompense :



- choisissez « grande vitesse, punition à l'arrêt »

Contrairement à la dernière fois, les capteurs ne sont pas reliés à une seule action mais à toute en même temps. Chaque connexion possède un poids différent qui correspond à la récompense que le robot s'attend à recevoir en utilisant cette connexion. Pour l'instant les poids sont initialisés aléatoirement.




Démarrez le robot. Vous pouvez aussi le lancer pas à pas  pour mieux comprendre ce qui se passe à chaque étape.

Initialement le robot n'est pas coincé, pour lui la meilleure action, celle qui lui apporterait le plus de récompense (celle avec le poids le plus fort) est de reculer, donc il choisit cette action. Mais comme il ne reçoit pas de récompense, il diminue le poids de cette arrête, puis il recommence. Comme il ne reçoit toujours pas de récompense, il continue de la diminuer, jusqu'à ce que ce poids devienne inférieur à celui d'une autre arrête. A ce moment, le robot pense que l'action qui lui donne le plus de récompense est tourner à gauche. Il tourne donc à gauche, obtient une récompense supérieure à sa prédiction, donc augmente le poids de cette arrête jusqu'à ce qu'elle se stabilise à 30. A ce moment, le robot pense que tourner à gauche est la meilleure action, et comme cette action le fait tourner sans jamais s'arrêter, il continuera à la faire indéfiniment.

Le robot se met donc à tourner à gauche sans s'arrêter. Or ce n'est pas sa meilleure action : il aurait un meilleur niveau en allant tout droit. Mais comme le robot n'a jamais essayé d'aller tout droit, il ne peut pas le savoir. Il faut donc forcer le robot à faire des essais.



Activez la case exploration  dans l'onglet du bas.

Lorsque l'exploration est activée, le robot ne choisira pas forcément l'arrête de poids le plus grand, mais choisira de temps en temps une autre direction. Lorsque le robot finira par choisir au hasard l'action aller tout droit, il recevra une forte récompense et augmentera le poids de cette arrête. Au bout d'un moment le poids de l'arrête dépassera celui de tourner à gauche, et le robot se mettra à aller tout droit au lieu de tourner en rond.

Au bout d'un moment, le robot a le comportement voulu (aller tout droit le plus souvent possible, reculer dès qu'il est coincé), mais cette fois-ci il y est parvenu entièrement par lui-même, sans qu'on ait eu besoin de lui indiquer quoi faire.

Notions apprises :

Pour permettre au robot d'apprendre sans être supervisé par des humains, on lui donne plus ou moins récompense à chaque fois qu'il effectue une action. En fonction de ce qu'il recevra, **le robot adaptera se choix de façon à maximiser ses récompenses.**

Les récompenses en question sont simplement une valeur numérique, négatives s'il s'agit d'une punition, que le robot cherche à maximiser.

Il faut aussi **permettre au robot d'explorer** et de ne pas toujours choisir l'action qui lui semble la meilleure, car cela lui permettra de découvrir de nouvelles actions qui pourront s'avérer encore meilleure.

APPRENTISSAGE PAR RENFORCEMENT - NAVIGATION AVEC CAMERA



Le principe est le même que précédemment, mais cette fois on utilise la caméra du robot. Les entrées sont donc très nombreuses (tous les pixels de la caméra et leur composition de rouge, bleu et vert) et on a besoin d'ajouter des couches de neurones pour bien analyser l'image ; on se retrouve donc avec un réseau très complexe, qu'on ne pourra pas faire à la main ou essayer de comprendre étape par étape.

Chargez la configuration « apprentissage par renforcement – navigation avec camera ».


Ou pour sélectionner vous-mêmes les paramètres, lancez le programme puis :

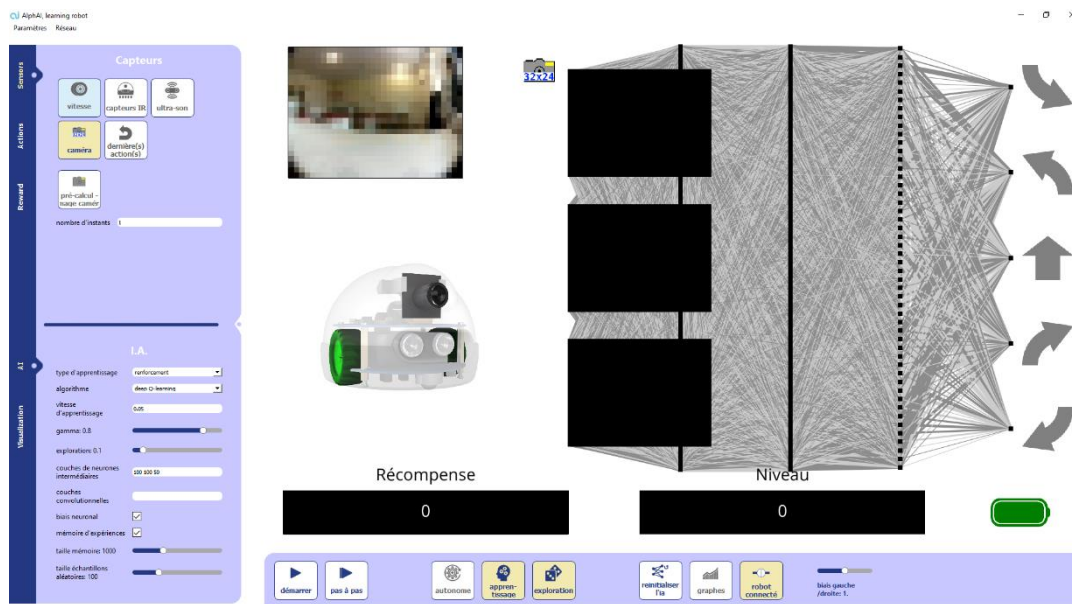
- Dans l'onglet IA :
 - sélectionnez « deep Q-learning »
 - dans « couches de neurones intermédiaires » mettez « 100 100 50 »
 - cochez mémoire d'expérience

- Dans l'onglet Capteurs, sélectionnez :

-  « bloqué/en mouvement »
-  « caméra » (par exemple 32x24)

- Dans l'onglet récompense :

- choisissez « grande vitesse, punition à l'arrêt » 



Démarrez le robot. Au bout d'un certain temps (environ 10 minutes) le robot finira par parcourir toute l'arène en anticipant les murs.



Notions apprises :


L'intérêt de l'intelligence artificielle est de pouvoir **traiter des réseaux gigantesques** et trouver les bonnes connexions qui permettront au robot d'avoir le comportement voulu, alors qu'il serait impossible d'essayer de le faire à la main ou de les intuer.

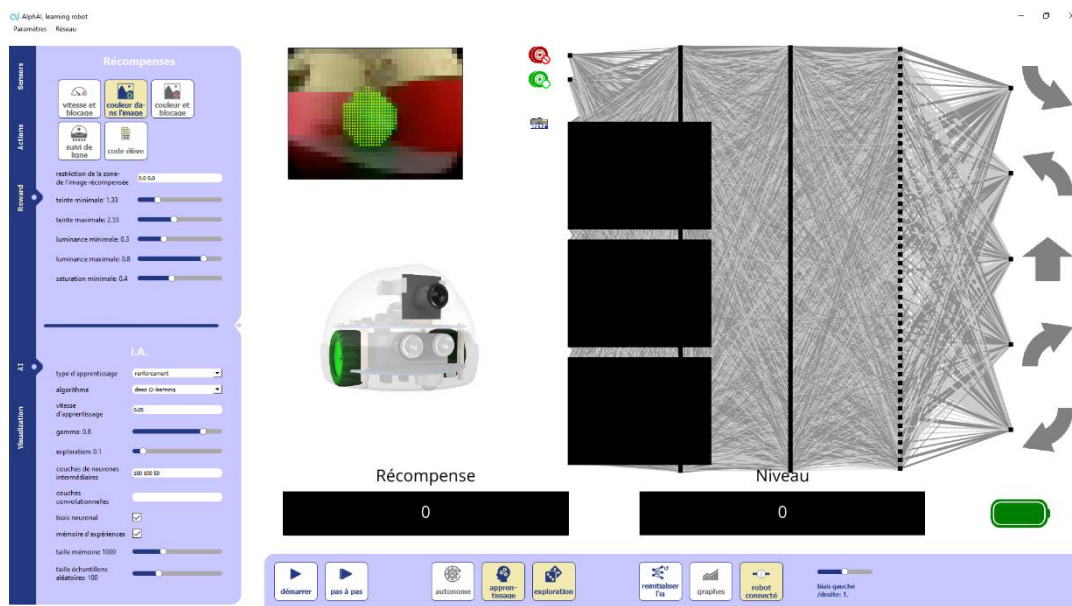
SUIVI D'UN BALLON

Voici un autre exemple d'apprentissage par renforcement. En changeant le type de récompense et en plaçant un nouvel élément dans l'arène, on peut réussir à faire adopter un comportement complètement différent au robot.

Chargez la configuration « suivi d'un ballon (vert) ». Ou pour sélectionner vous-mêmes les paramètres, lancez le programme puis :

- Dans l'onglet IA,
 - sélectionnez « deep Q-learning » ;
 - dans « couches de neurones intermédiaires » mettez « 100 100 50 »,
 - cochez mémoire d'expérience
- Dans l'onglet Capteurs, sélectionnez
 -  « bloqué/en mouvement »
 -  « caméra » (par exemple 32x24)
- Dans l'onglet Récompenses :

- choisissez « couleur dans l'image » 



-Placez un ballon de couleur verte dans l'arène (vous pouvez utiliser une autre couleur, mais dans ce cas il vous faudra modifier les paramètres de détection de couleur dans l'onglet Récompenses).

-Vérifiez que le réglage de la couleur du ballon est correct en plaçant le ballon devant le robot ; des points devraient apparaître sur l'image de la camera au niveau du ballon.

-Dans l'onglet récompense, variez la teinte et la luminosité pour faire correspondre la couleur à celle du ballon



-Démarez le robot. Au bout d'un moment (environ 20 minutes), le robot a se mettra à poursuivre le ballon.

Le robot essaye de maximiser le nombre de pixel de la couleur du ballon sur sa caméra : il va donc apprendre à s'approcher du ballon le plus près possible, mais comme en s'en approchant il tape dedans, il se met de nouveau à s'en rapprocher ainsi de suite.

Notions apprises :

Lors de l'apprentissage par renforcement, pour donner un comportement précis au robot, **il faut bien réfléchir à la récompense** ; de façon à ce qu'elle récompense le robot uniquement lorsqu'il a le comportement souhaité.