

Rapport de projet

-

IA & Cognition : IA Développementale

Travail de : ABDRABO Khaled, BRIGNONE Jean

Agent 1 : L'agent qui n'aimait pas s'ennuyer

Le but de cet agent était d'apprendre à prédire le résultat de ses actions en fonction de son environnement ainsi que de choisir une action différente quand ses prédictions sont correctes depuis trop longtemps, nous avons choisi un cycle de 4, comme préconisé dans le sujet. Nous avons décidé qu'au départ, par défaut, l'agent choisirait toujours l'action 0, car il est ignorant sur les différentes sorties de chacune des actions. Il va ensuite vérifier que sa prédiction est exacte, ou non.

Concernant l'ennui, nous avons ajouté un compteur d'actions qui, une fois arrivé à 4, forcerait l'agent à changer d'action pour ne pas s'ennuyer. Ce même compteur reprend la valeur de 0 si la prédiction faite par l'agent est fausse. Sur les captures d'écran ci-dessous, vous pourrez voir un compteur de bonnes prédictions. Dès qu'il atteint 4 ou que l'agent se trompe, retour à 0.

Voici la sortie pour l'exécution dans l'environnement 1 :

```
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 1
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 2
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 3
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 4
Action: 1, Anticipation: 0, Outcome: 1, Satisfaction: (anticipation: False, valence: 1), nombre de bonnes predictions: 1
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 4
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 1
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 2
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 3
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 4
Action: 1, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 1
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 4
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 1
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 2
```

Et ici la sortie de l'exécution dans l'environnement 2 :

```
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 0, Anticipation: 0, Outcome: 1, Satisfaction: (anticipation: False, valence: 1), nombre de bonnes predictions: 1
Action: 0, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 0, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 0, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 0, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 4
Action: 1, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 1
Action: 1, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 2
Action: 1, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 3
Action: 1, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 4
Action: 0, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 0, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 0, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 0, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 4
Action: 1, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 1
Action: 1, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 2
Action: 1, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 3
Action: 1, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 4
Action: 0, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 0, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
```

Agent 2 : L'agent qui préférerait les interactions positives

Le but de l'agent 2 est de choisir les interactions qui ont une valeur hédoniste positive, sauf s'il s'ennuie. Dans ce cas là il va préférer faire une action différente malgré le fait que cela puisse mener à une interaction négative.

Cet agent va donc commencer par l'action 0, comme nous l'avons expliqué dans la partie 1, et va continuer avec cette action là si la valeur de l'interaction est positive, sinon, il changera. Après avoir trouvé une action à valeur positive, il continuera à faire cette action jusqu'à ce qu'il s'ennuie, auquel cas il changera.

Voici la sortie pour l'exécution dans l'environnement 1, pour les valences $[-1, 1]$, $[-1, 1]$:

```
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 1
Action: 1, Anticipation: 0, Outcome: 1, Satisfaction: (anticipation: False, valence: 1), nombre de bonnes predictions: 2
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 0
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
```

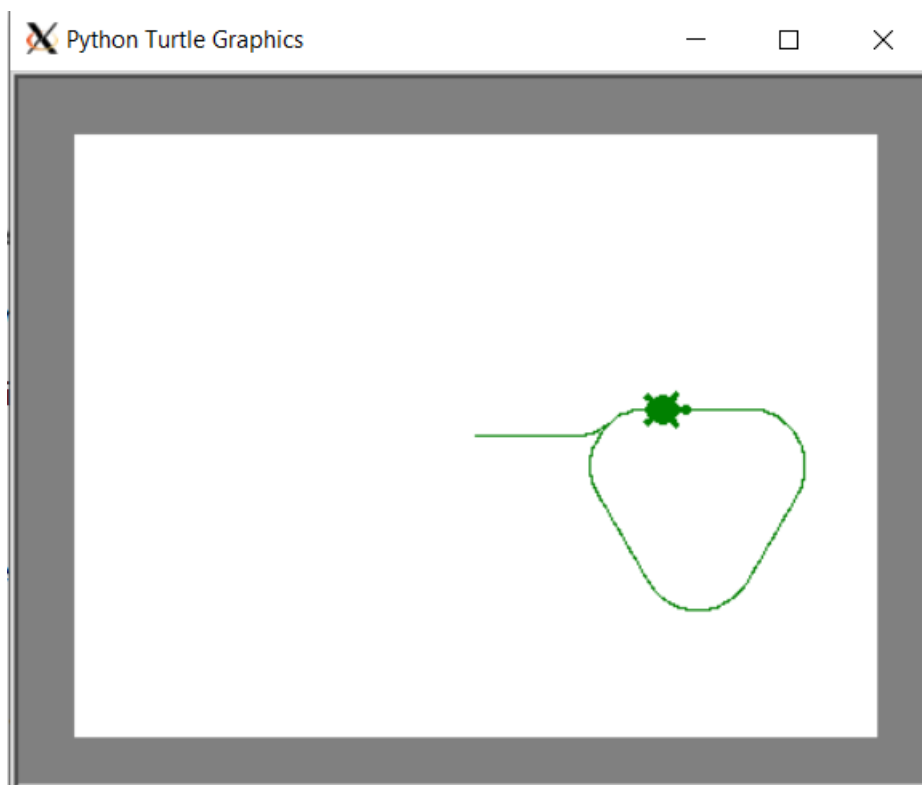
Voici la sortie pour l'exécution dans l'environnement 1 pour les valences $[1, -1]$, $[-1, -1]$:

```
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 0
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 1, Anticipation: 0, Outcome: 1, Satisfaction: (anticipation: False, valence: -1), nombre de bonnes predictions: 0
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 0
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
```


Agent 3 : L'agent qui pilotait une tortue

Comme expliqué dans le sujet, l'agent 3 est une spécialisation de l'agent 2, exécuté dans l'environnement Turtepy_enacter.py. Nous avons donc ajouté une action supplémentaire à l'agent. Tout comme pour l'agent 2, notre agent 3 a pour but de rechercher des valeurs positives, sauf ennui. Nous avons donc essayé différentes valeurs pour les valences.

Voici la sortie pour l'exécution pour les valences $[-1, 1]$, $[-1, 1]$, $[1, -1]$:



Ainsi que la trace de l'exécution :

```
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 1
Action: 1, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 2
Action: 2, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 2, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 2, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 2, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 2, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 2, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 2, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 2, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 2, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 2, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 2, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 2, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 2, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
```


Dans ce cas-là, la tortue ne se cogne pas, mais dans nos autres tests (valences $[[1, -1], [-1, 1], [1, -1]]$), il arrive à la tortue de se cogner contre les murs, la valence de cette action devient alors négative.

Dans notre cas ici, l'ennui empêche la tortue de choisir l'action 2 quatre fois de suite, c'est pour cela qu'elles sont entrecoupées d'actions 0, puis il revient à l'action 2, dont il sait que la valence est positive.

Agent 4 : L'agent qui adaptait ses actions au contexte

Cet agent là ne connaît pas l'ennui, il peut donc effectuer la même action plusieurs fois de suite, tant que la valence n'en devient pas négative, sans problèmes. Cet agent commence par l'action 0, comme expliqué précédemment, et va créer petit à petit des couples d'interactions afin de pouvoir, dans le futur, anticiper. A chaque fois qu'il fait 2 actions, il enregistre l'interaction et il remplit petit à petit son dictionnaire d'interactions et c'est grâce à cela qu'il pourra anticiper.

Prenons par exemple l'exécution de l'agent 4 dans l'environnement 3 qui, on le rappelle, retourne une valeur positive si les actions se succèdent (valences : $[[1, -1], [1, -1], [1, -1]]$).

```

Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 0, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 1
Action: 1, Anticipation: 0, Outcome: 1, Satisfaction: (anticipation: False, valence: 1), nombre de bonnes predictions: 2
Action: 1, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 0
Action: 1, Anticipation: 0, Outcome: 0, Satisfaction: (anticipation: True, valence: -1), nombre de bonnes predictions: 1
Action: 2, Anticipation: None, Outcome: 1, Satisfaction: (anticipation: False, valence: 1), nombre de bonnes predictions: 2
Action: 2, Anticipation: None, Outcome: 0, Satisfaction: (anticipation: False, valence: -1), nombre de bonnes predictions: 0
Action: 0, Anticipation: None, Outcome: 1, Satisfaction: (anticipation: False, valence: 1), nombre de bonnes predictions: 0
Action: 0, Anticipation: None, Outcome: 0, Satisfaction: (anticipation: False, valence: -1), nombre de bonnes predictions: 0
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 0
Action: 2, Anticipation: None, Outcome: 1, Satisfaction: (anticipation: False, valence: 1), nombre de bonnes predictions: 1
Action: 0, Anticipation: None, Outcome: 1, Satisfaction: (anticipation: False, valence: 1), nombre de bonnes predictions: 0
Action: 1, Anticipation: None, Outcome: 1, Satisfaction: (anticipation: False, valence: 1), nombre de bonnes predictions: 0
Action: 2, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 0
Action: 0, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 1
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 2
Action: 2, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 3
Action: 0, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 4
Action: 1, Anticipation: 1, Outcome: 1, Satisfaction: (anticipation: True, valence: 1), nombre de bonnes predictions: 5

```

L'agent commence par essayer jusqu'à ce qu'il se rende compte que lorsqu'il enchaîne les actions 0, 1 puis 2, les valences sont positives, il continue donc de s'exécuter de la sorte jusqu'à la fin.

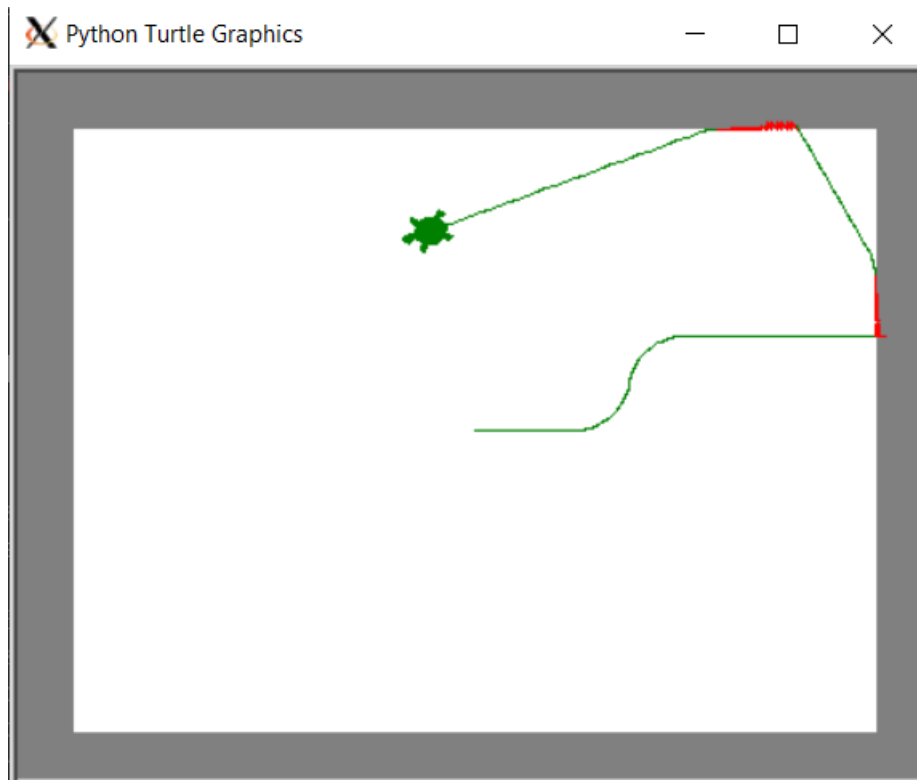
Il nous a ensuite été demandé d'exécuter cet agent dans l'environnement TurtlePy que nous avons utilisé pour l'agent 3. Le but est de vérifier qu'il peut anticiper le fait de se cogner lorsqu'il est déjà face à un mur.

Nous avons effectué différents tests avec différentes tables de valences :

Pour $[[-1, 1], [-1, 1], [1, -1]]$, l'agent se rend compte qu'exécuter l'action 2 en boucle lui apportera des valeurs positives, il ne fait alors que tourner en rond jusqu'à la fin de l'exécution.

Pour une table de valences $[[1, -1], [1, -1], [1, -1]]$, c'est à peu près pareil, sauf que c'est l'action 1 qu'il fait en boucle, en tournant en rond jusqu'à la fin de l'exécution.

Dans le cas d'une table $[[1, -1], [-1, 1], [-1, -1]]$, l'agent va se cogner, mais va changer de direction, se cogner encore. Si l'on étend la durée de l'exécution, on peut imaginer qu'il va sans arrêt changer de direction après avoir passé un certain temps à marcher contre un mur.



Conclusion

La première chose que l'on remarque en implémentant ces agents c'est que l'on finit très souvent par tomber sur des boucles infinies. Cela nous rappelle que ces agents sont très limités et que coder un comportement complexe prend bien plus de temps et de travail que cela. Il faudrait arriver à leur faire construire une représentation de leur environnement peut-être, pour qu'ils se rappellent la position des murs et peut-être prédire la position de ceux qu'il n'a pas encore rencontrés, si l'environnement suit une certaine logique.

L'implémentation de ces agents était une bonne introduction, de notre point de vue, à l'apprentissage simple de ces agents, à petite échelle.