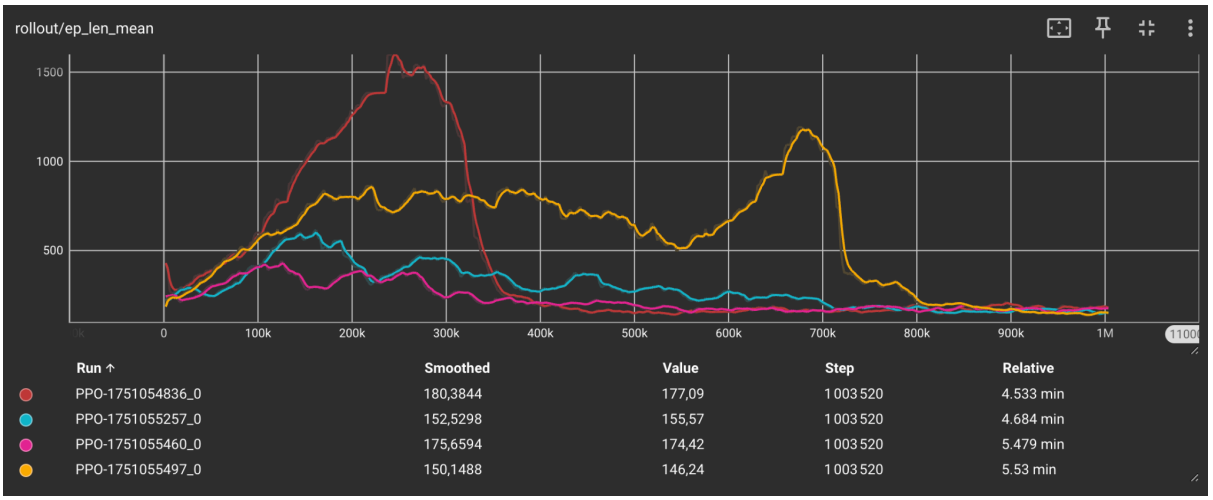
Actuellement j'essaie de donner à l'IA envie de courir vers la pomme, parce que c'est ennuyeux à suivre si elle prends son temps (spoiler : mauvaise idée). Pour ça j'utilise la méthode du punissement en cas de lenteur. Le problème c'est qu'elle préfère crever vite pour ne pas souffir.

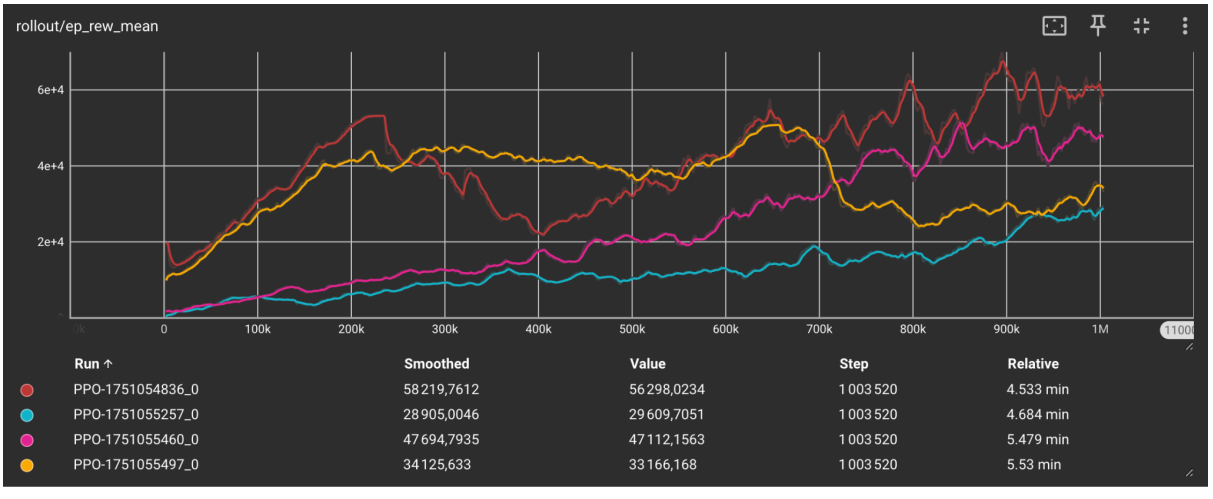
C'est bon, en ajustant la punition, l'IA est devenue très forte.

J'ai testé de lancer des entraînements sur 2 départ très légèrement différents : deux partait avec +50 par défaut (rouge et jaune), deux autres avec + 5 (bleu et rose). Au début, les +50 prennent leur temps. Par exemple, à 40 000 itérations, ils tournent encore autour du pot, tandis que les +5 sont déjà rapides, mais peurent plus vite. Étonnament, c'est une +50 qui remporte cette course, grâce à un excellent taux de récompenses après 80 000 itérations.



Run	Smoothed	Value	Step	Relative
PPO-1751054836_0	180,3844	177,09	1 003 520	4.533 min
PPO-1751055257_0	152,5298	155,57	1 003 520	4.684 min
PPO-1751055460_0	175,6594	174,42	1 003 520	5.479 min
PPO-1751055497_0	150,1488	146,24	1 003 520	5.53 min

Durée de vie par itération.



Run	Smoothed	Value	Step	Relative
PPO-1751054836_0	58 219,7612	56 298,0234	1 003 520	4.533 min
PPO-1751055257_0	28 905,0046	29 609,7051	1 003 520	4.684 min
PPO-1751055460_0	47 694,7935	47 112,1563	1 003 520	5.479 min
PPO-1751055497_0	34 125,633	33 166,168	1 003 520	5.53 min

Récompenses par itération.

Avoir un reward négatif dès le départ est très dérangeant pour l'IA. Le modèle qui partait de -50 se suicidait le plus tôt possible. Le modèle qui partait de 0 n'a jamais eu l'occasion

d'apprendre à manger la pomme. Soit la mort n'est pas assez punitive, soit la punition de lenteur est trop forte.

Les 4 modèles du dernier graphique connaissent une croissance fulgurante jusqu'à atteindre le palier de récompenses 6k, parfois dès 1M d'itérations. Ils sont alors particulièrement rapides pour atteindre la pomme. C'est autour de ce palier que l'IA commence à avoir une queue trop longue, et meure souvent en lui rentrant dedans. Ces 4 premiers modèles n'apprendront jamais comment bien vivre avec leur queue. Autour des 4M d'itérations, elles ont toutes connues une chute de récompenses dont elles ne se relèveront jamais vraiment. La peur de manger sa propre queue semble freiner.

Une autre approche a donc été envisagée : démarrer avec une longue queue, pour que l'IA apprenne directement à ne pas se rentrer dedans. Cependant l'IA n'arrivait pas à apprendre à manger la pomme.

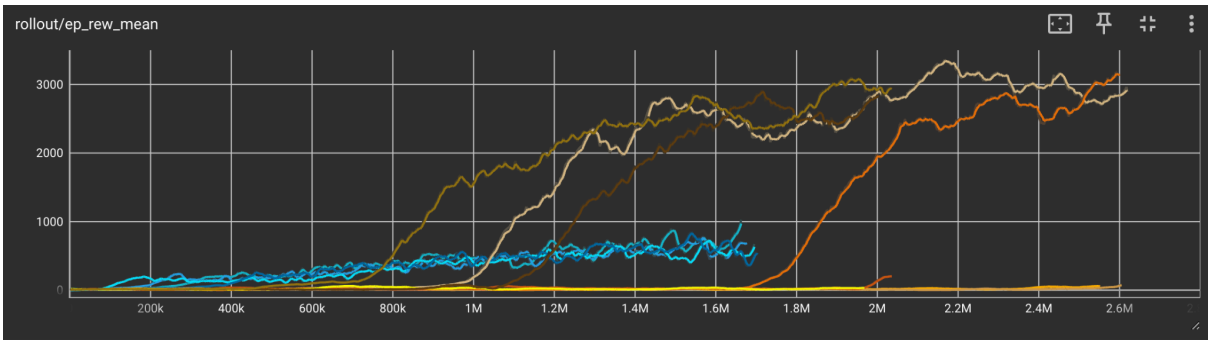
Il a donc été décidé de partir d'un des 4 premiers modèles, à un état jeune mais déjà excellent pour courir vers les pommes. Un branche de ce modèle a ensuite été entraînée en démarrant avec une longue queue, une forte punission en cas de mort, une forte récompense en cas de pomme et une plus faible punission en cas de lenteur. Ce nouveau modèle a vite appris à vivre avec sa queue et a pu progresssivement s'améliorer pendant 5M itérations supplémentaires, jusqu'au modèle PPO-1751056370-9M. Ce modèle obtient en moyenne 9.455 pommes par épisode. Des améliorations mineures, à base de départ avec une queue plus longue, n'auront jamais réussi à détrôner ce record.

Un modèle, démarrant avec une queue de longueur 12, était prometteur. Cependant, arrivé à une moyenne de 8 p/ep, il a stagné puis régressé. On dirait qu'il n'arrive pas à avoir conscience de sa queue. Les autres modèles ont appris un déplacement dans 3 directions pour être sûr de ne pas se rentrer dedans, ce qui empêche de faire demi-tour.

Essayons autre chose : commençons par un modèle surtout entraîné pour ne pas mourir, donc récompensé pour le temps qu'il reste en vie. Il gagnera aussi des rewards en fonction du nombre de pommes qu'il a mangé. Plus tard, on lui apprendra la vitesse. Après plusieurs essais, on obtient le même problème des 3 directions : les modèles décident de ne jamais aller à gauche pour ne pas rentrer dans leur queue.

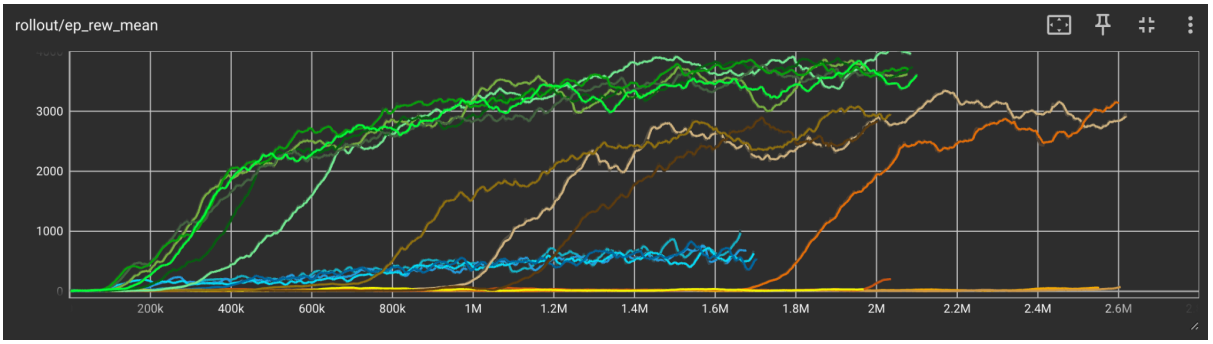
Après beaucoup (trop) d'ajustements et de changements visant à apprendre au serpent où se trouve sa queue, plusieurs lessons on été tirées :
- il ne sert à rien de forcer un modèle à être récompensé rapidement, puisqu'au fur et à mesure des entraînements, il désire obtenir plus de reward dans le même temps imparti, et cherche donc la vitesse. Les récompenses et punitions basées sur la vitesse ne font que ralentir le processus et complexifier l'observation.
- démarrer avec une longue queue n'aide pas, le problème ne vient pas de là.
- observer des nombres compris entre 0 et 1 semble plus approprié.
- donner une récompense au moment où le serpent mange la pomme donne de bien meilleurs résultats que donner une récompense en fonction du nombre de pommes mangées. Ne me demandez pas pourquoi, je n'en sais rien.

Tentons est un nouveau concept pour aider l'ia à ne pas mourir. L'idée est de simuler sa vision avec des distances d'obstacle, comme un radar, plutôt que de lui dire où elle se trouve sur la carte. Sur graphique ci-dessous, on remarque que les modèles mettent du temps avant de comprendre subitement cette nouvelle donnée. Une fois comprise, ils deviennent redoutables et surpassent tous les modèles précédents. Grâce à ce changement, l'IA est enfin arrivée à la presque optimisation parfaite de sa course. N'oublions pas que pour l'instant, le jeu se déroule sans agrandissement de la queue.



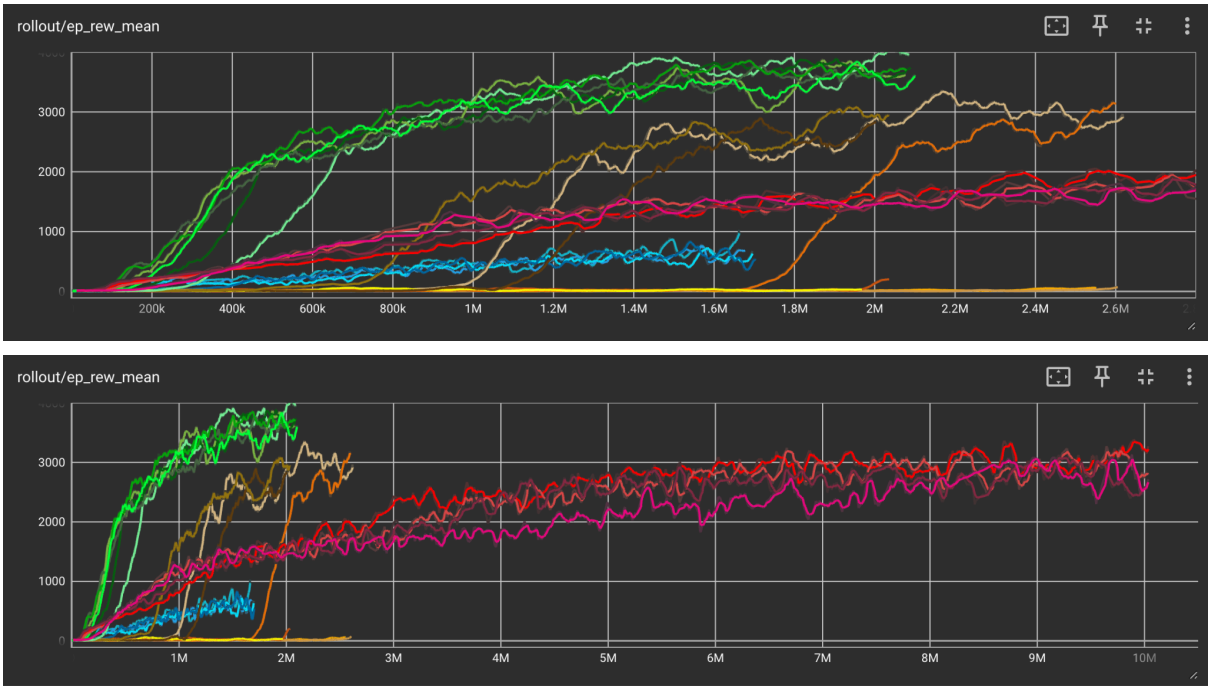
En bleu : l'observation ne concerne que apple_position et snake_head.
En jaune et dégradés : le prochain obstacle dans chaque direction est ajouté à l'observation.

Devant le succès de l'approche "radar", il est tentant de reproduire ce concept avec la pomme. Plutôt que de donner des positions, donnons à observer la distance entre le serpent et la pomme. Ici encore, le modèle comprends beaucoup mieux cette donnée. Qui plus est, peut-être parce qu'elle ressemble aux radars d'obstacles, il comprends désormais très vite les obstacles.



En vert : apple_position et snake_head sont remplacés par apple_delta.

Essayons maintenant en implémentant la règle officielle du jeu Snake, cad en faisant pousser la queue lorsque le serpent mange une pomme. Notons que l'approche radar permet d'identifier la queue sans avoir à l'ajouter à l'observation.



En rouge : même environnement que vert, mais la queue grandit à chaque pomme mangée.

Bingo !!!! Les modèles ont appris plus lentement que la génération précédente, mais en constante amélioration jusqu'au 10M itérations. Si les récompenses sont plus basses, c'est parce que l'agrandissement de la queue rend la jeu plus difficile. L'entraînement aurait pu être continué, mais les résultats obtenus sont satisfaisants. Le modèle 232559-9.9M obtient ainsi une moyenne record de 34.634 p/ep en situation réelle.