SnA.I.ke

Présentation générale

Afin de pouvoir démontrer le pouvoir du deep reinforcement learning, nous voulons créer un jeu de Snake doté d'une intelligence artificielle de type DRL. Durant une simulation, nous voulons pouvoir afficher les informations pertinentes provenant de l'intelligence artificielle. Nous aimerions aussi pouvoir jouer à ce même jeu par nous-même, afin de pouvoir se divertir durant nos temps libres. Nous voulons enregistrer le highscore du joueur, ainsi que celui de l'IA. Finalement, nous voulons pouvoir visualiser les statistiques des simulations à l'aide de graphiques.

Présentation détaillée

Premièrement, nous voulons un menu principal à plusieurs options. La première sera de jouer une partie régulière, donc contrôlée par le joueur. La deuxième sera la simulation de DRL. La troisième option, et non la moindre, sera de pouvoir visualiser les graphiques de données. Le nom du programme sera aussi affiché sur le dessus du menu. Le menu principal devrait avoir un certain style visuel.

Pour le jeu en tant que tel, nous voulons un Snake classique, donc un serpent qui doit manger une pomme. Manger une pomme fera en sorte que le corps du serpent devient plus long. Si à n'importe quel moment dans la partie la tête du serpent touche soit à un mur ou à son propre corps, la partie se termine. Le bût, naturellement, est de devenir le plus long possible. Le joueur pourra se déplacer sur 4 directions, vers le haut, le bas, la droite et la gauche en utilisant les flèches du clavier. Avant de commencer la partie, le joueur aura plusieurs options offertes à lui, la vitesse du serpent, la couleur du serpent, la couleur de la tête, la couleur de la pomme et la taille de la planche de jeu. Nous voulons avoir 2 tailles possible, 20x20 et 40x40. Si le joueur quitte le programme, on veut garder les dernières couleurs utilisées. Quand le joueur perd, afficher un message de fin de jeu ainsi que son

score actuel. Si le joueur bat son meilleur score, l'enregistrer dans la base de données et afficher un message pour l'avertir.

Pour la simulation, nous voulons encore avoir le choix de couleur et de la grosseur de planche, mais la vitesse n'est pas nécessaire ici. Un bouton sera utilisé pour commencer et mettre sur pause la simulation en tout temps, ainsi qu'un bouton pour la réinitialiser. Cette fois ci, nous voulons faire rouler le jeu 4 fois, afin de pouvoirs enregistrer des résultats plus rapidement. Ceci veut donc dire qu'il faut faire l'affichage de ces 4 parties aussi. Nous voulons pouvoir voir les informations pertinentes des parties, tel que le nombre de pommes mangées, ainsi que les informations de l'I.A, comme son pointage de renforcement ainsi que le nombre d'épisodes. Pour ce faire, cliquer sur une des 4 parties affichera l'information de celle-ci. À chaque fin d'épisode, nous allons enregistrer dans la base de données le nombre de pas, le score, le nombre de points de renforcement et la durée de l'épisode. Finalement, si l'intelligence artificielle bat son meilleur score, l'enregistrer dans la base de données.

Pour les affichages graphiques, nous voulons pouvoir visualiser les informations enregistrées dans notre base de données. Il doit y avoir 4 choix de graphiques, le score par épisode, le nombre de points de renforcement par épisode, la durée par épisode et le nombre de pas fait par secondes dans chaque épisodes.

Contraintes

Étant le clou du spectacle, il est primordial que l'intelligence artificielle soit fluide. Nous ne voulons aucuns ralentissements du programme lors d'une simulation, il devrait rouler aussi bien que si on fait une partie normale. Il est important de comprendre que les fonctionnalités de jeu et d'intelligence artificielle sont les priorités et passe naturellement avant les composantes stylistiques (les couleurs et le style du menu). Nous voulons que le tout soit codé en Python. Pour la base de données, nous voulons utiliser PostgresSQL.

Plateformes ciblées

Nous désirons que le programme soit une application de desktop.