



Henallux

Informatique de Gestion - Bloc 3

Projet IA : implémenter une intelligence artificielle dans un jeu

Auteur:

DE BUCK Henry
SOMME Aurélien

À l'attention de:

Mme Smal
Mme Sboui

Un devoir soumis pour le cours de :

IG 309

*Projet : création d'une intelligence artificielle
Programmation Web et programmation fonctionnelle*

Décembre 17, 2022

Table des matières

1	<i>Introduction</i>	2
2	<i>La structure du projet</i>	2
3	<i>La base de données</i>	2
4	<i>L'intelligence artificielle</i>	3
4.1	<i>Son fonctionnement</i>	3
4.1.1	La Q function	3
4.1.2	Notre IA dans le programme	3
5	<i>Analyse de l'IA</i>	4
5.1	<i>Le déroulement de son entraînement</i>	4
6	<i>Fonctionnalités supplémentaires développées</i>	4

Table des figures

Figure 1	Schéma de base de données	2
----------	---------------------------	---

1 Introduction

Dans le cadre de notre troisième année d'informatique de gestion en option data, nous avons remis un projet dans nos cours de création d'intelligence artificielle et de programmation web et fonctionnelle.

Ce projet a pour but dans un premier temps :

De faire tourner sur un serveur web un jeu fonctionnant en python. Dans ce but nous utiliserons Flask, qui fera tourner pour nous le serveur et la base de données à l'aide de Flask SQLAlchemy.

Et dans un second temps :

Y incorporer une intelligence artificielle qui apprendra à jouer sur ce jeu qui aura une taille de plateau variable. Pour ça nous utiliserons la Q-function.

2 La structure du projet

Dans le dossier principal du projet se trouve le fichier python run permettant de démarrer le serveur, le fichier config, le readme.txt (expliquant comment le projet fonctionne), la base de données app.db et un dossier game_app.

Le dossier game_app contient un dossier static (contenant les fichiers js/css/svg), un fichier python init, un fichier view qui gèrera toutes les requêtes entre le serveur et le web, un fichier models contenant tous les models SQL de la DB (database), un fichier ai gérant l'intelligence artificiel et un fichier business qui s'occupera de la gestion d'une partie (en orienté objet).

3 La base de données

Nous retrouvons ici le modèle de notre base de données qui reprend chaque table. Pour chacune d'elles, nous avons leurs attributs et types.

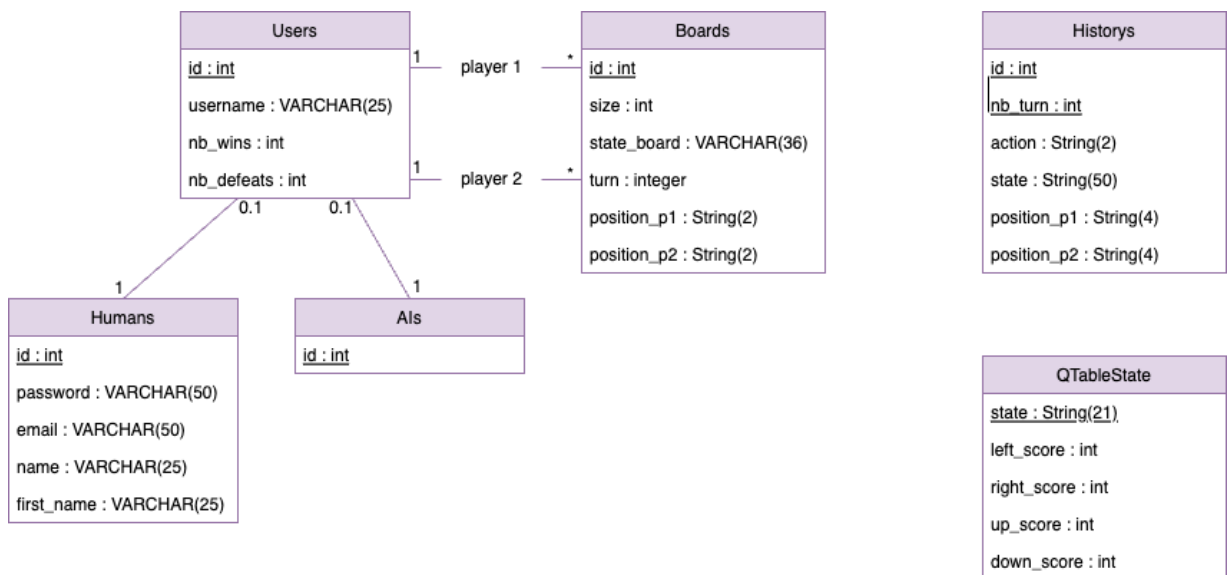


Figure 1 Schéma de base de données

Nous avons donc dans notre modèle :

Des utilisateurs contenant leurs pseudonymes ainsi que leur nombre de parties gagnées ou perdues. Un utilisateur sera soit une IA soit un être humain. Dans le cas d'un humain, on retiendra alors les informations de connexion ainsi que son nom et prénom.

Il est important de noter que l'on retient dans les utilisateurs le nombre de victoires et de défaites. Il s'agit là de redondance de données, il est tout à fait possible de retrouver chaque partie dans lequel à jouer un utilisateur et calculer le nombre de cases qu'il possédait pour savoir s'il a gagné ou perdu. Nous préférons le stocker pour éviter d'occuper notre serveur à faire de longues requêtes SQL. D'autant plus que pour épargner notre base de données, nous comptons effacer par cycle les parties finies.

Nos parties sont enregistrées sous forme de plateaux. Chacun contient les deux utilisateurs jouant la partie ainsi que l'état de la partie actuelle (actualisé à chaque mouvement). Notre partie contiendra alors aussi la taille du plateau et le nombre de tours.

Pour ces parties nous gardons chaque tour en mémoire dans la base de données sous forme d'un historique. Cet historique permet notamment de rembobiner une partie ainsi que pour calculer nos récompenses dans la Q fonction.

Enfin, on enregistre les états de la Q fonction pour permettre à notre IA de jouer.

4 L'intelligence artificielle

4.1 Son fonctionnement

L'IA utilise la Q fonction pour apprendre au fil des parties. À partir du deuxième coup de l'IA, cette dernière va mettre à jour la QTable (stocké en DB) à l'aide de la formule de la Q fonction. Pour mettre cela à jour, il faut un état d'une partie ainsi que l'action et l'état suivant obtenu (au prochain tour de l'IA). C'est donc pour cela que la QTable de peut pas être mise à jour au premier tour de l'IA.

La Q fonction permet de connaître l'espérance de gain d'une action selon un état. Dans notre cas, l'espérance/valeur de gagner la partie en allant depuis un état vers le haut/bas/droite/gauche. Une QTable aura donc 4 valeurs différentes pour les 4 directions. L'IA choisira donc la direction ou la valeur sera la plus haute.

4.1.1 La Q fonction

$$Q[s, a] = (1 - \alpha)Q[s, a] + \alpha(r + \gamma \max_{a'} Q[s', a'])$$

4.1.2 Notre IA dans le programme

Notre IA choisit la meilleure direction à chaque tour basé sur la QTable.

5 Analyse de l'IA

5.1 Le déroulement de son entraînement

Au départ nous avons entraîné l'IA sur un epsilon favorisant l'exploration étant donné que les QTables étaient vides. Le learning rate est élevé au début permettant de rapidement obtenir une base pour l'IA. Le gamma n'est pas très élevé au début étant donné qu'on manque d'information pour mettre à jour la QTable.

Ensuite nous avons réduit l'epsilon au fur et à mesure afin que l'IA explore et en même temps se base sur ce qu'elle a appris. Nous avons réduit le learning rate et augmenter le nombre de parties jouées afin d'avoir un entraînement plus précis. Le gamma est également plus haut afin de donner du poids à l'état suivant.

6 Fonctionnalités supplémentaires développées

Nous avons réalisé un tableau settings qui permet de choisir notre token, couleur dans la partie pour les deux joueurs ainsi que la taille du tableau.

Nous avons aussi un splendide mode nuit qui ne fonctionne pas avec le cash.