

## Compte rendu technique : L'Intelligence Artificielle dans Space Shooter

Ce document explique en détail comment on a mis de l'IA dans notre projet. (suite à la demande d'intégrer un algorithme de recherche (A\*) et un Perceptron). Voici comment on a fait.

### 1. L'Algorithme de Recherche (Le "A\* Simplifié")

Pour l'algorithme de recherche, on a décidé de l'utiliser pour le tir spécial du Boss : le "MegaLaser".

Le but était que le laser trouve son chemin vers le joueur tout seul, comme un missile à tête chercheuse.

#### **Comment ça fonctionne?**

Au lieu de faire un chemin compliqué avec des nœuds (comme un vrai A\* avec des murs), on a fait une version adaptée à l'espace (où c'est vide).

Le laser calcule à chaque fois "l'heuristique", c'est-à-dire la distance qui le sépare du vaisseau.

On utilise la formule de Pythagore pour ça :  $\text{distance} = \text{racine}((x_2 - x_1)^2 + (y_2 - y_1)^2)$ .

Si le joueur est plus à droite que le laser, le laser va à droite.

Si le joueur est plus à gauche, le laser va à gauche.

#### **Optimisation :**

Pour pas que ce soit impossible à esquiver (sinon c'est de la triche), on a ajouté une règle : si le laser est trop proche du joueur (moins de 100 pixels), il arrête de "réfléchir" et continue tout droit. Ça laisse une chance au joueur de bouger au dernier moment.

### 2. Le Perceptron (Apprentissage Renforcé)

Pour le Perceptron, on voulait que le Boss ait une sorte de cerveau. Qu'il ne bouge pas juste au hasard.

On a créé un fichier perceptron.py avec une classe qui imite un neurone biologique (enfin en version mathématique simple).

#### **Les Entrées (Ce que le boss "voit") :**

Le cerveau du boss reçoit 3 informations :

1. Sa vie est basse ? (Plus il a mal, plus le chiffre est grand)
2. Il est aligné avec nous ? (Est-ce qu'il est en face ou sur le côté)
3. Il est proche ? (Est-ce qu'on est collé à lui)

#### **La Décision (La Sortie) :**

Le perceptron fait une somme de tout ça avec des "poids" (l'importance de chaque info).

Si le résultat est positif (1), le Boss passe en mode rapide. Il va super vite pour nous foncer dedans ou esquiver.

Si le résultat est négatif (0), le Boss reste en mode lent. Il prend son temps pour viser.

**L'Apprentissage (Mémoire Génétique) :**

C'est la partie la plus intéressante dans l'ajout du perceptron pour rendre le jeu jouissif. Le boss n'apprend pas pendant le combat (c'est trop court), mais il apprend de ses ancêtres.

Dans le fichier main.py, on sauvegarde le cerveau du boss quand il meurt.

Pour le boss de la vague d'après, on reprend le même cerveau mais on augmente un peu le poids de la "douleur".

Ça veut dire que le boss suivant sera encore plus agressif quand il perdra de la vie.

C'est comme une évolution génétique accélérée.