

## Conception d'une intelligence artificielle capable de jouer de manière autonome et performante au jeu Snake

L'intelligence artificielle, en constante expansion, offre de nombreuses applications polyvalentes, y compris dans l'ingénierie. Mon intérêt pour l'informatique m'a poussés à explorer les algorithmes génétiques et les réseaux neuronaux dans le contexte ludique du jeu Snake. Je souhaite comprendre les fondements théoriques de cette technologie et son application pratique.

L'engouement persistant pour le jeu Snake lui confère un caractère ludique emblématique. En concevant une IA performante et autonome pour résoudre ce jeu d'arcade des années 80, nous établissons un lien clair avec le thème de l'année tout en explorant les fondements théoriques des réseaux neuronaux et des algorithmes génétiques.

### Positionnement thématique (ÉTAPE 1) :

- *INFORMATIQUE (Informatique Théorique)*
- *INFORMATIQUE (Informatique pratique)*
- *MATHEMATIQUES (Géométrie)*

### Mots-clés (ÉTAPE 1) :

#### Mots-clés (en français) Mots-clés (en anglais)

<i>Intelligence artificielle</i>	<i>Artificial Intelligence</i>
<i>Algorithme génétique</i>	<i>Genetic Algorithm</i>
<i>Réseaux neuronaux</i>	<i>Neural Networks</i>
<i>Crossing-over</i>	<i>Crossover</i>
<i>Snake</i>	<i>Snake</i>

### Bibliographie commentée

Le jeu Snake est né en 1976 sur la borne d'arcade Blockade fabriquée par l'entreprise Gremlin. Il est popularisé sur les téléphones Nokia en 1997 [1].

Le principe du jeu Snake est de contrôler un serpent qui se déplace en mangeant des pommes pour grandir. L'objectif est de manger le plus de pommes et le jeu prend fin quand le serpent se heurte à un mur ou se mord la queue.

L'idée ici est de concevoir et d'entraîner une IA pour jouer de manière autonome et efficace à ce jeu en se basant sur un algorithme génétique appliqué à un réseau de neurones. En 1943, Warren McCulloch et Walter Pitts ont révolutionné notre compréhension des neurones en les comparant à des éléments de logique booléenne [2]. Leur idée novatrice était de considérer qu'un neurone s'active à partir d'un seuil et que son activation ou désactivation impacte les décisions prises par le réseau neuronal complet. Cette approche a permis des avancées majeures dans le domaine de l'intelligence artificielle.

Les réseaux de neurones artificiels sont constitués de couches nodales, contenant une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie.

On s'intéresse ici à l'application de cette théorie au jeu Snake. À chaque itération du jeu, le serpent décide de se déplacer dans une des quatre directions (haut, bas, gauche, droite) sans revenir en arrière. Cette direction est obtenue comme sortie d'un réseau de neurones comportant deux couches cachées. Les entrées du réseau de neurones sont les distances en nombre de cases de la tête du serpent : à la pomme, à un élément de son corps et à la limite de la grille et ce dans les huit directions de vision de la tête du serpent.

La notion de poids a été introduite par Frank Rosenblatt en 1958 [2] pour accorder plus ou moins d'importance à certains neurones. McCulloch et Pitts ont proposé de rendre le réseau neuronal plus performant en optimisant ces poids.

Ici nous proposons de conduire cette optimisation à l'aide d'algorithmes génétiques [3, 4] pour permettre au serpent de manger le plus de pommes possible.

Cette approche développée par John Holland en 1975 [5] a pour objectif d'imiter l'évolution naturelle, en faisant évoluer une population au lieu d'étudier un seul individu.

À partir d'une population initiale, les serpents les plus performants sont sélectionnés grâce à une fonction de fitness représentant leur score au jeu et sont croisés, créant ainsi des enfants dont le réseau neuronal est composé d'une partie du réseau neuronal de chaque parent. De plus, à chaque génération, des mutations sont introduites de manière aléatoire sur le réseau neuronal des enfants afin d'engendrer

de possibles améliorations.

Le choix de la fonction de fitness est déterminant dans l'efficacité de l'algorithme génétique. Plusieurs

stratégies de sélection, de croisement et de mutation peuvent aussi être mises en œuvre et influencent

la performance de l'entraînement du réseau de neurones [6].

Il est légitime de se demander comment formaliser ces différents concepts et de se poser la question

de la convergence de l'entraînement. Une approche consiste à étudier la représentation en graphe des réseaux de neurones, et à considérer les schémas d'Holland pour saisir comment les opérateurs génétiques agissent sur la population de serpents [5, 7, 4].

## Problématique retenue

Notre sujet vise à répondre aux questions suivantes: comment entraîner une intelligence artificielle pour être compétitive au jeu Snake à l'aide d'algorithmes génétiques et réseaux de neurones? Quelles optimisations et paramètres sont à prendre en compte pour obtenir un bon score? Quels sont les fondements théoriques de ces technologies?

## Objectifs du TIPE du candidat

Notre objectif est de développer une intelligence artificielle autonome et performante pour le jeu Snake.

Pour cela, nous commençons par coder un jeu fonctionnel en Python avec la bibliothèque PyGame.

Ensuite, nous créons un réseau neuronal en codant les fonctions de vision du serpent, l'activation des neurones et leur réponse pour prendre des décisions de direction.

Nous nous concentrons ensuite sur la mise en place de fonctions permettant d'appliquer des opérateurs génétiques tels que la sélection, le croisement et les mutations pour entraîner ces réseaux neuronaux.

Enfin, nous approfondissons les aspects théoriques de ces technologies.

## Références bibliographiques (ÉTAPE 1)

- [1] K. SMITH : the golden age arcade historian "the ultimate (so-far) history of gremlin industries part 2" : [https://allincolorforaquarter.blogspot.com/2015/09/the-ultimate-so-far-history-of-gremlin\\_25.html](https://allincolorforaquarter.blogspot.com/2015/09/the-ultimate-so-far-history-of-gremlin_25.html)
- [2] IBM : Qu'est-ce qu'un réseau de neurones ? : <https://www.ibm.com/fr-fr/topics/neural-networks>
- [3] R. POLI, W. LANGDON, N. MCPHEE, ET J. KOZA : A Field Guide to Genetic Programming : ISBN: 978-1-4092-0073-4

- [4] D. WHITLEY : “A genetic algorithm tutorial,” Statistics and computing, vol. 4, pp. 65–85 : [https://www.researchgate.net/publication/2425017\\_A\\_Genetic\\_Algorithm\\_Tutorial](https://www.researchgate.net/publication/2425017_A_Genetic_Algorithm_Tutorial)
- [5] J. HOLLAND : Adaptation in Natural and Artificial Systems : An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence : ISBN: 9780262581110
- [6] J.-F. YEH, P.-H. SU, S.-H. HUANG, ET T.-C. CHIANG, : snake game ai : Movement rating functions and evolutionary algorithm-based optimization : [https://www.researchgate.net/publication/309390260\\_Snake\\_game\\_AI\\_Movement\\_rating\\_functions\\_and\\_evolutionary\\_algorithm-based\\_optimization](https://www.researchgate.net/publication/309390260_Snake_game_AI_Movement_rating_functions_and_evolutionary_algorithm-based_optimization)
- [7] L. DAVIS : Handbook of Genetic Algorithms : [http://www0.cs.ucl.ac.uk/staff/W.Langdon/ftp/papers/poli08\\_fieldguide.pdf](http://www0.cs.ucl.ac.uk/staff/W.Langdon/ftp/papers/poli08_fieldguide.pdf)