

Smart Dash

Rapport fin de Semestre



El Bouroumi Anas
Grossman Jeremy
Perez Ioann

Tuteur : Guénaél Cabanes

Tables des matières

- Introduction
- Analyse
- Réalisation
- Planning du déroulement du projet
- Répartition du travail
- Présentation d'un élément original
- Objectifs pour la fin du projet
- Conclusion

Introduction

Le présent document a pour objectif de rendre compte de l'avancement du projet Geometry Dash AI. Nous aborderons les différentes phases du projet, depuis sa conception jusqu'aux résultats attendus.

Notre projet consiste à développer une intelligence artificielle capable de jouer au jeu Geometry Dash. Geometry Dash est un jeu de plateforme exigeant où un cube se déplace automatiquement vers la droite, et le joueur doit sauter au bon moment pour éviter les obstacles. L'objectif ultime est de permettre à l'intelligence artificielle de naviguer avec succès à travers des niveaux qu'elle n'a jamais rencontrés auparavant, démontrant ainsi sa capacité à réagir à un nouveau terrain.

Pour ce faire, nous allons créer une version épurée de Geometry Dash permettant de calculer les positions du joueur et du terrain sans recourir à un affichage graphique.

Cela facilitera un entraînement plus rapide de l'intelligence artificielle. Les fonctionnalités clés incluent le calcul dynamique des positions, un système d'entraînement efficace pour une adaptation rapide, et la garantie de l'adaptabilité de l'IA à de nouveaux terrains. La méthodologie comprend le développement d'un moteur de jeu simplifié, l'implémentation d'algorithmes d'apprentissage, la création de scénarios d'entraînement diversifiés, et l'optimisation des performances pour un apprentissage rapide. Les résultats attendus sont un moteur de jeu simplifié, une IA entraînée capable de jouer à des niveaux inconnus sans affichage graphique, et des performances optimisées pour un entraînement efficace.

Analyse

Ce projet complexe peut être décomposé en plusieurs sections distinctes, chacune jouant un rôle crucial dans le succès global de l'initiative.

Moteur de Jeu

Le développement du moteur de jeu représente la première étape cruciale de notre projet. Cette phase englobe la conception et l'implémentation d'un moteur de jeu simplifié pour Geometry Dash. L'objectif est de créer une plateforme fonctionnelle avec laquelle l'IA pourra interagir et évoluer.

Les principaux éléments à prendre en compte dans cette section comprennent le calcul dynamique des positions et la gestion des collisions par rapport au bloc et au pique des terrains..

Nous avons donc créé une version très simplifiée du jeu, où le joueur se déplace bloc par bloc à chaque intervalle de temps.

Diagramme de classe du début :

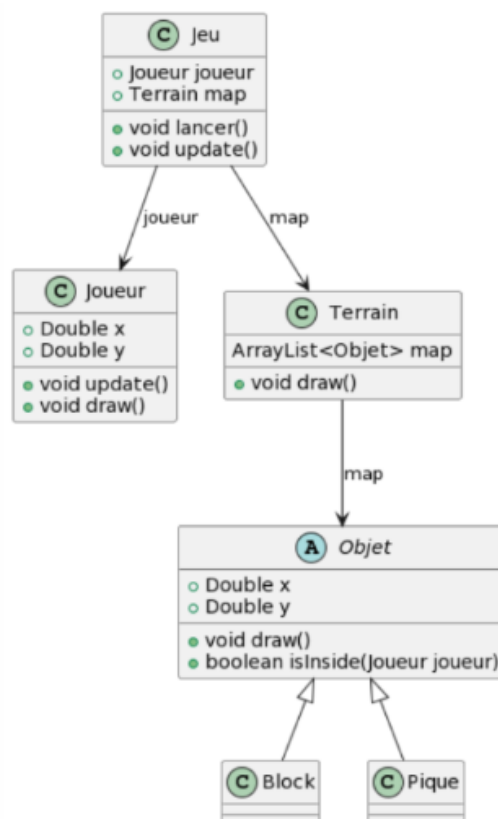


Diagramme de classe final :



IA et Apprentissage

La partie IA et apprentissage est le cœur même de notre projet. Nous devons concevoir un système d'intelligence artificielle capable d'apprendre à jouer à Geometry Dash. Cela implique le développement de réseaux neuronaux modulaires, l'implémentation d'algorithmes d'apprentissage par évolution, et la création de scénarios d'entraînement variés. L'objectif est de permettre à l'IA de prendre des décisions intelligentes basées sur l'environnement de jeu.

Nous avons également mis en place des fabriques afin de simplifier la création d'instance de différents type de réseaux.

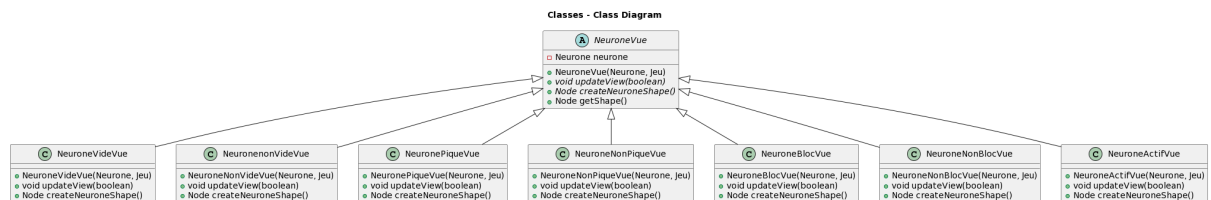
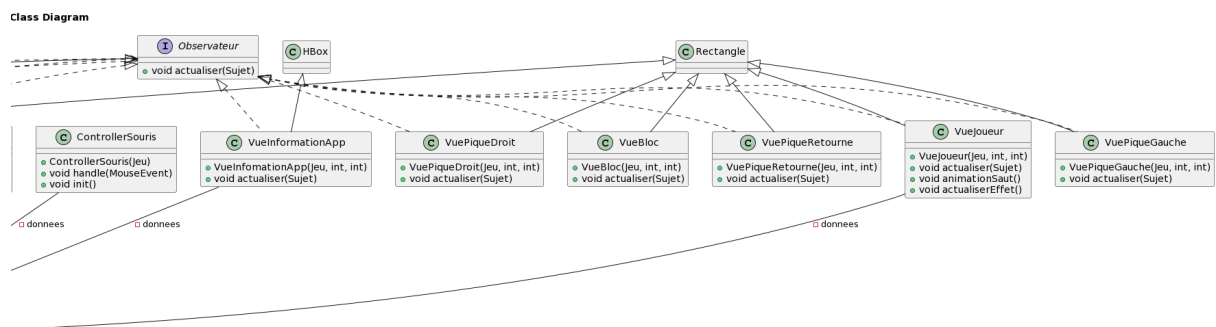
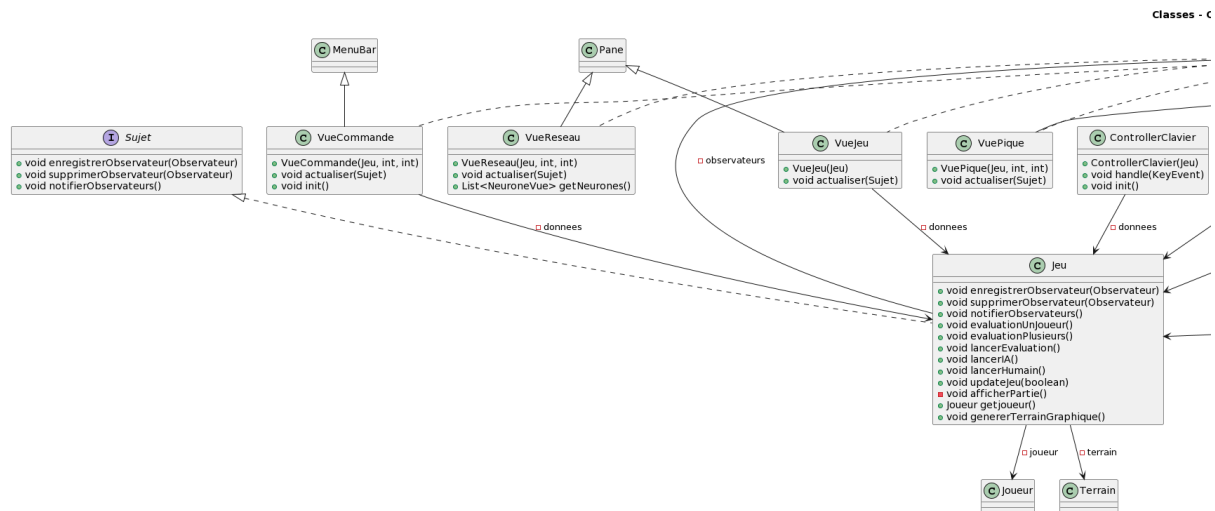
Diagramme de classe final de l'IA:



Moteur Graphique

Bien que notre objectif principal soit de développer une IA sans besoin d'une interface graphique, une section consacrée au moteur graphique est essentielle. Cela peut inclure la création d'un moteur graphique simplifié pour permettre un affichage visuel, même s'il n'est pas utilisé par l'IA. Cette partie peut également aborder des aspects liés à la convivialité et à la visualisation des résultats de l'IA.

Diagramme de classe final du moteur graphique :



Chacune de ces sections contribue de manière significative à la réalisation de notre objectif global, et leur développement doit être soigneusement planifié et coordonné pour assurer le succès du projet.

Evolution par rapport à l'étude préalable :

Création de l'IA	Création du moteur jeu	Création moteur graphique
Création de neurone simples ✓	Création de niveau ✓	Représentation des éléments 2D ✓
Réseau de neurones simple ✓	Saut et déplacement ✓	Animation du personnage ✓
Algorithme d'apprentissage ✓	Collisions ✓	Interface utilisateur ✓
Enregistrement de l'apprentissage ✓		
Simulation en parallèle ✓		

Réalisation

La phase de réalisation de notre projet a nécessité une approche méthodique et collaborative, impliquant des efforts significatifs dans divers domaines. Cette section détaille l'architecture logicielle adoptée, les tests de validation effectués, ainsi que les défis rencontrés tout au long du processus.

Architecture Logicielle

Moteur de Jeu

Pour la conception du moteur de jeu, nous avons divisé les responsabilités entre différents composants, facilitant ainsi la gestion du mouvement du joueur, le calcul des collisions et la gestion de l'environnement et des terrains.

IA et Apprentissage

L'architecture logicielle de la partie IA est basée sur des réseaux neuronaux modulaires.

Chaque module correspond à une fonction spécifique, comme la prise de décision sur le moment du saut. Cette approche modulaire facilite l'extension du système pour gérer des situations plus complexes. Nous avons de plus implémenté des algorithmes d'apprentissage par évolution (plus précisément NEAT), permettant à l'IA d'ajuster ses comportements en fonction des résultats obtenus lors de scénarios d'entraînement.

Moteur Graphique

Etant donnée que cette partie n'était pas la priorité , nous avons développé un moteur graphique minimaliste pour des besoins de validation et de débogage. Ce moteur affiche visuellement l'évolution du joueur virtuel et permet de vérifier la cohérence entre les décisions prises par l'IA et le comportement observé.

Tests de Validation

La phase de tests a été cruciale pour garantir le bon fonctionnement de notre solution et ainsi éviter de se retrouver avec un jeu qui ne fonctionne pas correctement où des neurones défectueux lors de l'apprentissage.

Nous avons effectué des tests de validation à plusieurs niveaux :

Unitaires : Chaque module du moteur de jeu et de l'IA a été soumis à des tests unitaires pour vérifier son bon fonctionnement individuel.

Intégration : Nous avons évalué l'interaction entre les différents composants du système, garantissant une communication fonctionnelle.

Scénarios d'Entraînement : Des tests ont été réalisés avec des scénarios d'entraînement variés pour évaluer la capacité d'adaptation de l'IA à des situations diverses.

Moteur Graphique : Les fonctionnalités du moteur graphique ont été testées pour s'assurer de leur conformité aux attentes.

Difficultés Rencontrées

Durant ce projet nous avons pu faire face à quelques difficultés. Parmi les difficultés les plus notables, nous avons rencontré :

- Complexité de l'Apprentissage : La conception d'un système d'apprentissage capable de s'adapter efficacement à des environnements complexes a posé des défis, nécessitant des itérations fréquentes. En effet il a fallu choisir judicieusement des méthodes de mutation et de croisement optimal afin d'avoir un taux d'apprentissage si trop faible ni trop important et ainsi donc obtenir la convergence attendue.
- Complexité du réseau de neurones initial : Il a également fallu choisir une structure permettant à l'IA de pouvoir apprendre correctement, sans quoi l'IA convergent beaucoup trop tôt. Nous devons également faire attention à ce que la structure ne soit pas trop importante non plus afin d'éviter le surapprentissage de l'IA sur les niveaux d'apprentissage. Cela en vient à

modifier le nombre de modules, le nombre de neurones par modules, les positions initiales..

- Nous avons également eu quelques difficultés plus générales en ce qui concerne les effets de bords des objets java étant données qu'on définissait les réseaux de neurones des joueurs à chaque génération.

En dépit de ces défis, notre approche collaborative et notre engagement envers des tests rigoureux ont permis de surmonter ces obstacles et d'atteindre des résultats significatifs dans la réalisation de notre projet.

Planning de déroulement du projet

Ce projet s'est articulé autour de plusieurs axes clés : la création du moteur de jeu, l'élaboration des structures de l'Intelligence Artificielle (IA), et l'implémentation de l'apprentissage par renforcement.

Pour ce projet, nous avons adopté les principes de la méthode Agile, en procédant par itérations. Nous allons donc décrire le projet itération par itération.

Itération 1 :

Nous avons entamé la première itération par le développement du moteur de jeu. Notre version initiale, bien que simplifiée, permettait de se déplacer bloc par bloc, intégrant uniquement des blocs et des pics comme obstacles. Parallèlement, nous avons travaillé sur la mise en place des structures fondamentales de l'IA et des modules neuronaux.

Itération 2 :

Durant la deuxième itération, notre focus s'est porté sur l'apprentissage. Nous avons employé un algorithme d'apprentissage évolutif, connu sous le nom d'algorithme NEAT.

Cette première implémentation avait des aspects fixes, comme la position des neurones. Nous avons également créé des terrains d'entraînement et une méthode permettant de combiner ces terrains pour en générer de nouveaux, aléatoires, à partir des existants.

L'enregistrement de l'apprentissage a été implémenté, permettant la sérialisation des données dans des fichiers texte pour une récupération ultérieure.

Les statistiques compilées dans un document PDF résumant l'évolution de l'apprentissage, génération après génération, et présentent une courbe d'évolution de la population ainsi que les performances des dix meilleurs individus.

Itération 3 :

Pour la troisième itération, nous nous sommes principalement concentrés sur l'amélioration de l'apprentissage, suite à des résultats peu convaincants obtenus lors de la deuxième itération. Nous avons corrigé les problèmes liés aux effets de bord et à la gestion de la mémoire.

Nous avons ensuite envisagé l'introduction d'une nouvelle forme de mutation capable de modifier aléatoirement la position des neurones, avec une probabilité définie pour chaque neurone.

Nous avons ensuite adopté une approche permettant à un joueur de s'exercer sur plusieurs terrains, sélectionnant celui avec le meilleur score moyen.

Le système de dé-sérialisation mis en place facilite la récupération des réseaux de neurones à partir des fichiers enregistrés, une méthode de parallélisation a également été développée pour permettre l'exécution simultanée de plusieurs sessions d'apprentissage.

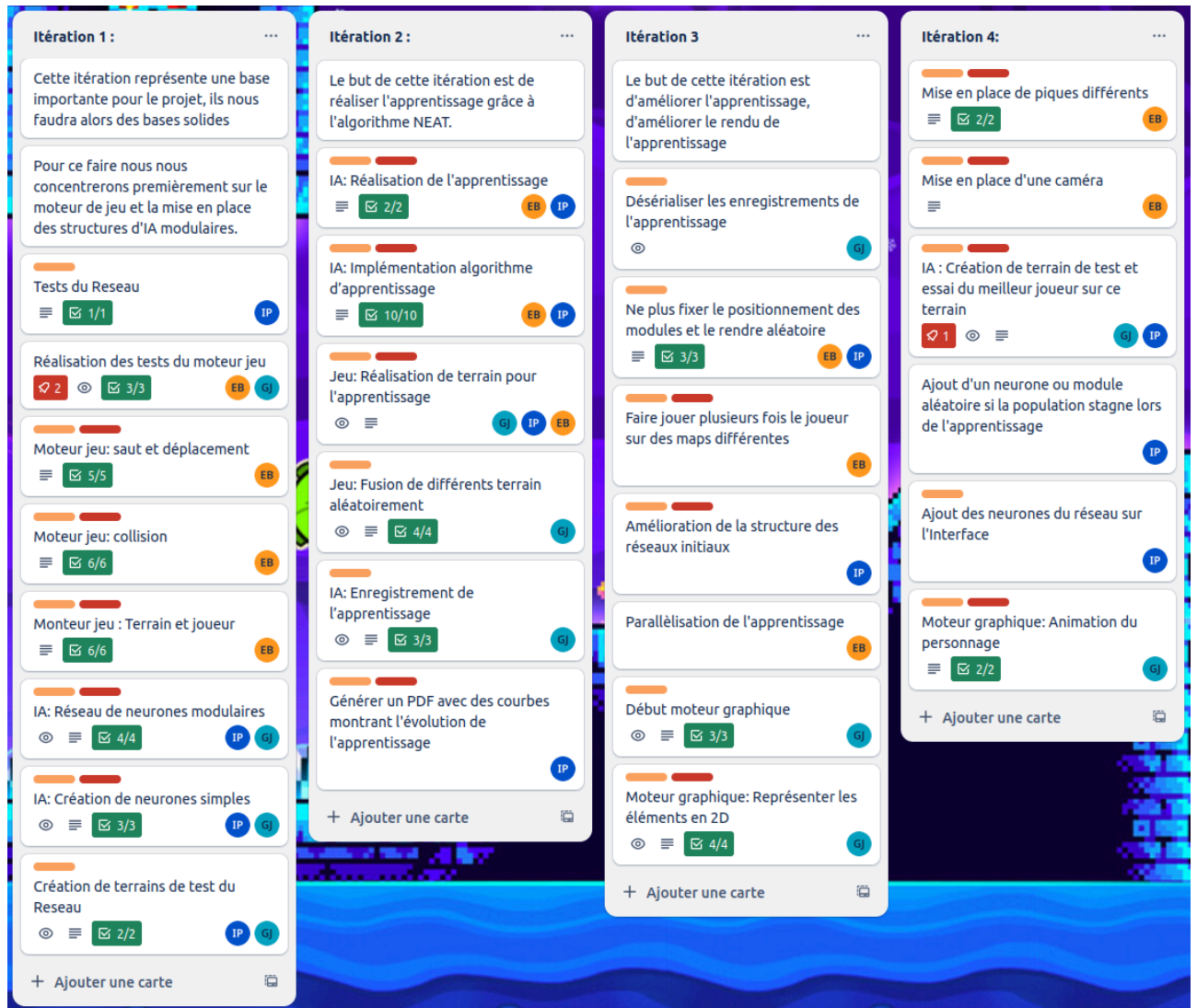
Nous avons fini par commencer à intégrer les patrons de conception Modèle-Vue-Contrôleur (MVC) et Observateur pour le moteur graphique.

Itération 4 :

Dans la quatrième itération, l'accent a été mis sur l'aspect graphique du projet. Nous avons introduit des animations pour le personnage, ajouté un effet de lueur autour de celui-ci et modifié l'orientation des pics en fonction de leur placement.

Une interface utilisateur présentant des informations sur le personnage et le réseau de neurones a été mise en place, ainsi qu'une vue spécifique pour l'affichage des neurones.

Trello configuré selon les principes de la méthode Agile :



Répartition du travail

Nous avons organisé le travail de façon équitable, en permettant à chacun de choisir les tâches qui lui plaisaient le plus. Lorsque plusieurs personnes souhaitaient travailler sur la même tâche, elle était partagée pour que chacun puisse apporter sa contribution.

Itération 1 :

Dans la première itération, Anas a pris en charge le développement du moteur de jeu, pendant que Ioann et Jeremy se concentraient sur la mise en place de la structure des intelligences artificielles (IA).

Itération 2 :

Pour la deuxième itération, les trois étudiants se sont concentrés sur la création de terrains, avec pour objectif que chaque étudiant développe cinq terrains différents pour l'entraînement de l'IA.

Ensuite, Anas et Ioann ont travaillé ensemble sur l'implémentation de l'apprentissage, en s'assurant de suivre l'étude préalable et le pseudo-code établi. Simultanément, Jeremy s'est focalisé sur la création d'une méthode permettant de fusionner différents terrains et de générer de nouveaux terrains aléatoires pour l'entraînement.

Ioann a conclu cette itération en produisant un PDF qui documentait les statistiques de l'apprentissage par rapport à chaque génération.

Itération 3 :

Pour la troisième itération, Anas et Ioann ont concentré leurs efforts sur l'amélioration de l'apprentissage, Ioann travaillant sur la mutation des positions et Anas sur la répétition des sessions de jeu pour un même joueur. Jeremy, de son côté, s'est occupé de la sérialisation et de l'intégration des patrons de conception pour le moteur graphique.

Itération 4 :

Lors de la quatrième itération, tous les étudiants se sont focalisés sur l'aspect graphique de l'application. Anas a amélioré l'affichage en ajoutant des effets visuels sur le personnage et en orientant correctement les obstacles. Jeremy a pris en charge l'animation du joueur.

et a introduit une vue "Informations" pour afficher des détails sur le joueur, son réseau de neurones, etc. Enfin, Ioann a mis en œuvre l'affichage graphique des réseaux de neurones, qui s'affiche à l'activation d'un module spécifique.

Présentation d'un élément original

Anas :

Pour ce projet Geometry Dash, je ressens une grande fierté en constatant que notre IA apprend de manière satisfaisante. Il est impressionnant de voir l'IA jouer à un niveau élevé à la fin de son processus d'apprentissage, montrant une amélioration génération après génération. Tout cela est rendu possible grâce à l'algorithme d'apprentissage par évolution NEAT, qui s'inspire de l'évolution biologique. Je trouve cet algorithme particulièrement logique, et les résultats obtenus sont véritablement impressionnants.

Ioann:

Dans ce projet Geometry Dash, je trouve que notre choix de réseaux de neurones est original. En effet, contrairement à ce qu'on pourrait retrouver dans un réseau plus traditionnel, nous avons utilisé des réseaux de neurones modulaires pour lesquels nous avons créé notre propre structure, avec des types de neurones propres à chaque type de cases présentes dans notre jeu. Puis le fait de voir qu'avec une bonne configuration des réseaux l'IA peut par elle-même réussir tout type de niveaux jusqu'à dépasser la capacité intellectuelle de l'humain sur ce jeu a été satisfaisant.

Jérémy:

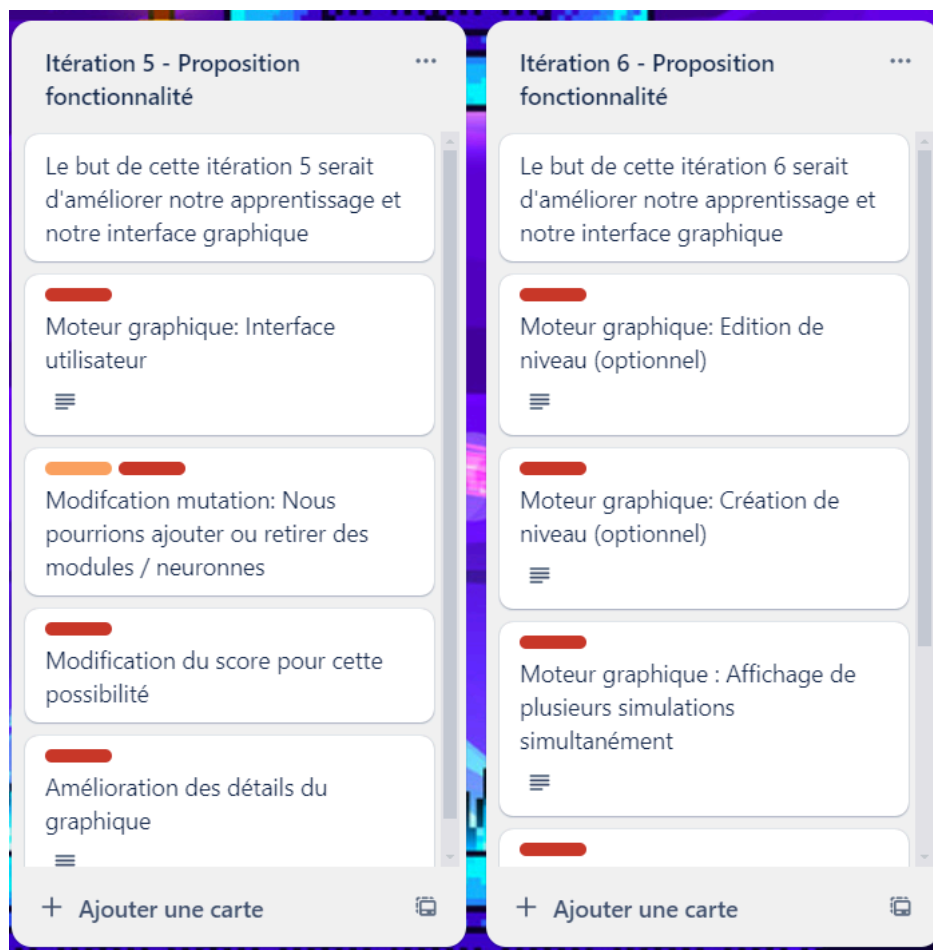
Dans le cadre de notre projet sur Geometry Dash, je trouve que l'aspect le plus gratifiant a été l'évolution remarquable de notre intelligence artificielle tout au long du processus. En élaborant notre propre IA, nous avons choisi d'adopter l'algorithme NEAT pour guider notre progression. À chaque itération, nous avons pu constater une nette amélioration de notre IA, jusqu'à obtenir un résultat final qui a dépassé nos attentes. Ce développement continu de notre IA a constitué un élément central et significatif de notre projet, et je tenais à souligner son importance dans notre réussite globale.

Objectifs finaux

Nos objectifs finaux pour la conclusion du projet consistent à améliorer notre système d'apprentissage, visant à doter notre intelligence artificielle de la capacité de compléter un nombre maximal de niveaux, y compris ceux auxquels elle n'a jamais été confrontée auparavant.

Par la suite, nous envisageons d'améliorer notre moteur graphique afin de rendre notre travail plus valorisant. Dans cette optique, nous souhaitons intégrer une interface utilisateur qui simplifierait l'utilisation de notre application.

Trello montrant les fonctionnalités prévues pour les itérations 5 et 6 :



Conclusion

Pour conclure, nous avons achevé les principales fonctionnalités de notre projet, ce qui signifie que nous avons atteint notre objectif initial tel que défini dans l'étude préalable. Nous avons créé une intelligence artificielle capable de jouer et d'apprendre à s'améliorer dans une version simplifiée du célèbre jeu Geometry Dash.

Bien que nous ayons réalisé une grande partie du projet, celui-ci n'est pas encore totalement achevé. Il est donc nécessaire de continuer à perfectionner l'apprentissage de nos IA pour identifier la structure la plus optimale, et de tenter de développer une IA capable de s'adapter à des environnements inconnus.

Les aspects graphiques du projet pourront également être améliorés lors des prochaines itérations.