

KALANA

Sujet AM4:
Animation physique et optimisation
de la morphologie de créatures
animées

Encadrant : Alexandre Meyer

Printemps 2021-2022
Université Claude-Bernard Lyon1

p1905392
Ahmed ATANANE
p1907353
Anna-Ludvina Cayuela
p1812594
Laura Nesta

Le projet

Le sujet :

L'objectif du projet est d'établir un algorithme génétique pour permettre à une créature d'évoluer et de survivre dans un environnement.

L'environnement :

Après avoir réfléchi séparément à une idée pour mettre en forme notre projet, nous avons décidé de faire évoluer notre créature dans l'environnement suivant: deux falaises séparées par un précipice. Le but pour la créature étant d'atteindre la falaise opposée à la falaise de départ.

Notre créature devra donc atteindre sa cible en volant et sera soumise à la gravité et à la pression de l'air.

Les deux falaises sont des Mesh Collider et une zone symbolise la zone que nos créatures doivent atteindre pour pouvoir être considérées pour la génération d'une nouvelle population.

La créature:

Pour ce qui est de la créature, nous avons créé un Prefab. Un Prefab est un gameObject que nous pouvons configurer facilement. Cela a été particulièrement utile lors de la génération de plusieurs créatures. En effet, le Prefab nous permet de créer de nouvelles instances de celui-ci dans la scène.

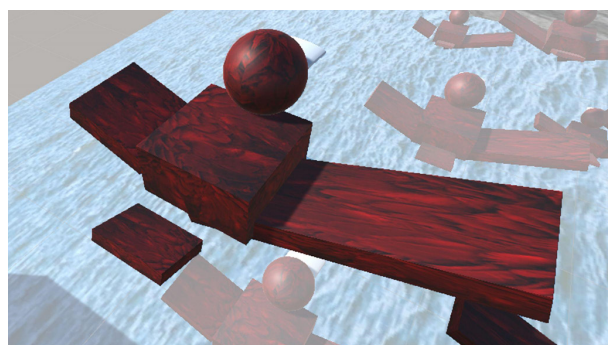
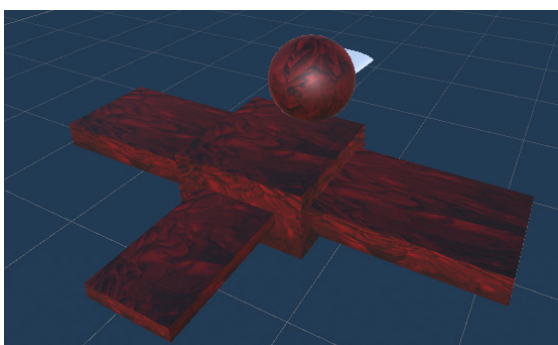
Notre Prefab Créature est ainsi composé d'un corps, d'une tête, de deux ailes et d'une queue. Chaque composant de notre oiseau possède un script qui lui est associé.

Par exemple, pour les ailes, dans le script Aile.cs un booléen indique s'il s'agit ou non d'une aile gauche et applique en conséquence un mouvement à l'aile.

L'oiseau est donc soumis à la gravité. Les battements des ailes se font latéralement avec de faibles mouvements en avant. La prise d'altitude se fait en appliquant une force à la créature lorsque l'aile est en position basse via un vecteur force $v(\text{speed}, \text{verticalSpeed}, \text{horizontalSpeed})$ ou $v(\text{speed}, 0, \text{horizontalSpeed})$ (selon la position de l'aile).

Les battements de la queue, eux, se font seulement en avant mais la prise d'altitude s'effectue selon le même principe que celui des ailes mais suivant un vecteur différent: $v(\text{horizontalSpeed}, 0, 0)$ ou $v(\text{horizontalSpeed}, \text{verticalSpeed}, 0)$

Aussi, la taille des composants influe sur la force. En effet, on multiplie les vecteurs forces, par la taille du composant en question.

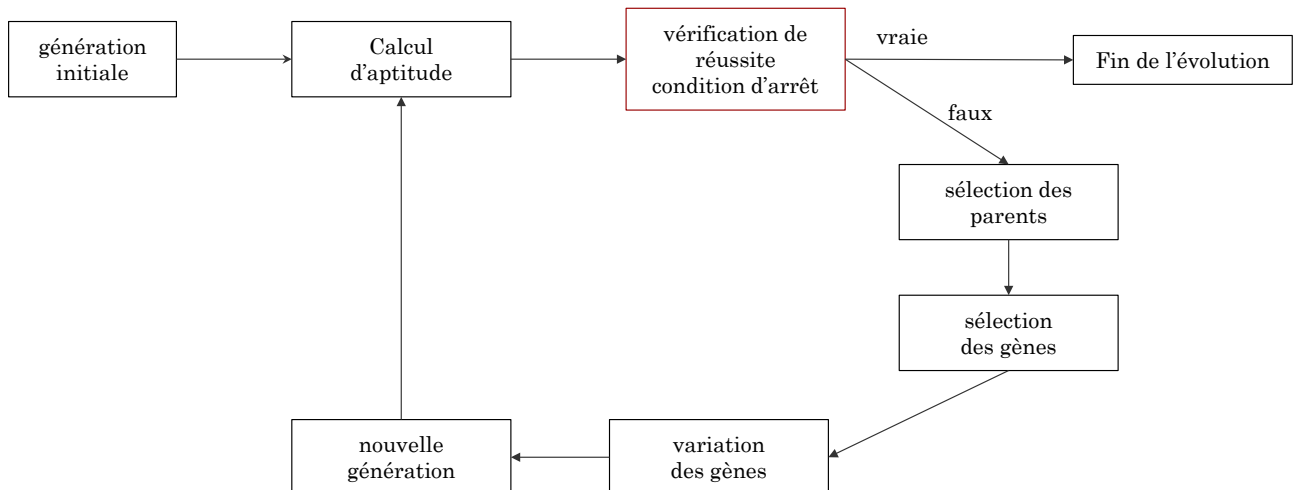


Le projet

L'algorithme génétique :

L'objectif de l'algorithme génétique est de permettre à une population de créature d'évoluer. Il existe différents types d'algorithme génétique mais le principe général reste le même.

Nous avons choisi de créer notre propre algorithme génétique représenté par le schéma suivant.

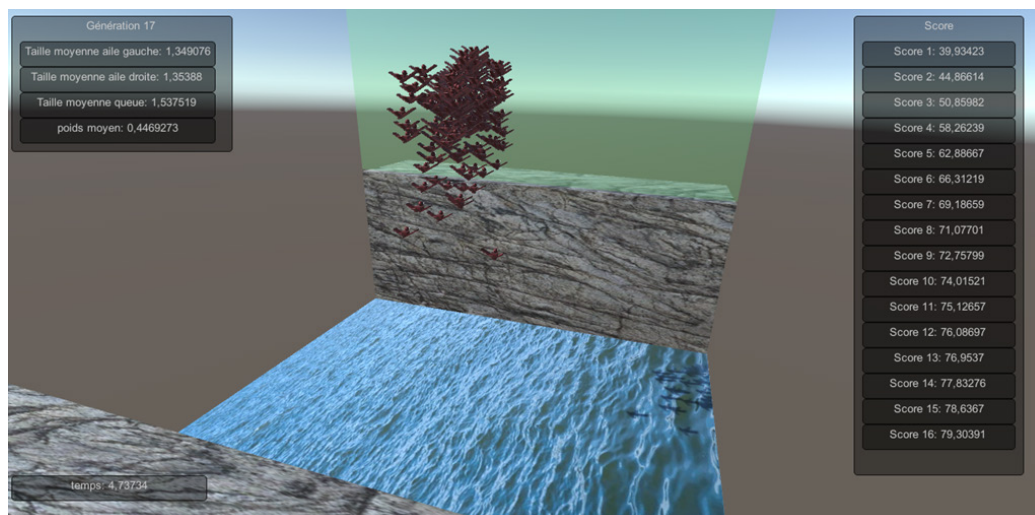


L'algorithme génère une première génération de créatures avec des gènes complètement aléatoire. A la fin de la simulation, chaque créature se voit attribuer un score. Les 60 meilleures créatures sont sélectionnées pour représenter les parents.

Contrairement à la plupart des algorithmes génétiques qui sélectionne 2 individus et mélange leurs gènes pour en créer une nouvelle, nous choisissons les gènes aléatoirement parmi l'ensemble de la population parent.

Une fois cette première sélection effectuée, une variation jusqu'à 40% est appliquée sur l'ensemble des gènes. La nouvelle génération est ainsi créée.

Ce processus est réitéré tant que la moyenne des scores d'une génération n'a pas atteint 80/100.



Le projet

Les résultats :

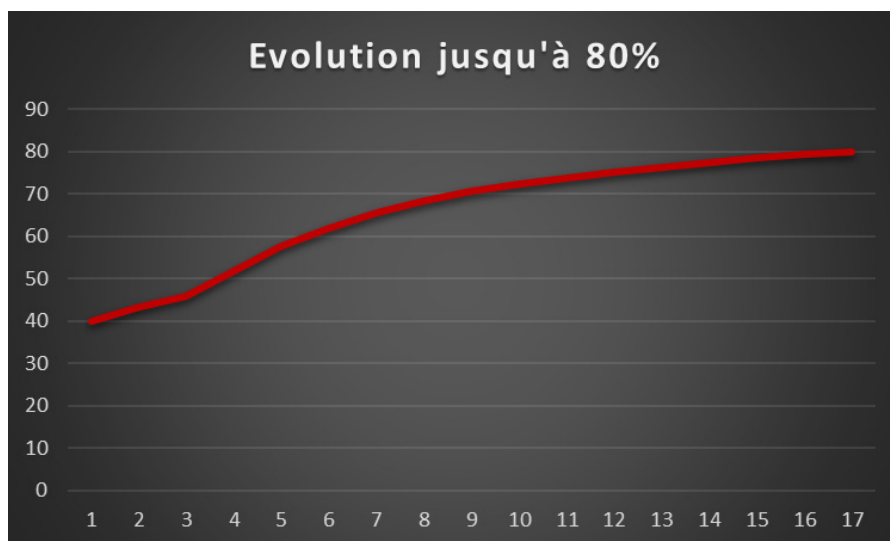
Nous avons élaboré notre algorithme génétique appliqué aux oiseaux principalement sur les scripts Génération.cs permettant de générer une population aléatoire d'un certain nombre de créatures, et Évolution.cs permettant de lancer des générations avec des populations plus performantes à chaque fois.

Lorsque nous avons terminé, en exécutant notre algorithme, nous obtenons une première génération avec une famille d'oiseaux plus nombreux à tomber dans le précipice et moins nombreux à atteindre la zone verte qui est l'objectif avec un score initial en dessous de la moyenne.

Après une génération, l'ancienne population d'oiseaux est détruite, les oiseaux qui ont gardé les gènes les plus efficaces ont tendance à rejoindre la zone verte et chaque nouvelle famille de créatures générée conserve de plus en plus d'oiseaux performants et moins d'oiseaux qui tombent à l'eau.

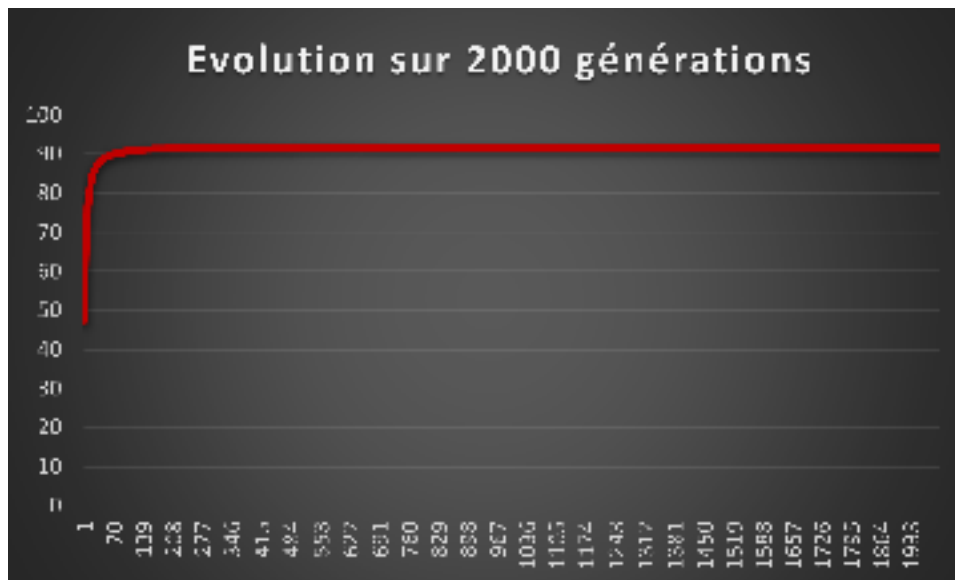
A chaque génération lancée, sont affichées sur la fenêtre graphique les moyennes des différents gènes de la population de créatures ainsi que le score de la génération correspondant à la moyenne des gènes.

A la dernière génération lancée, tous les oiseaux sont aptes à atteindre la zone verte et on affiche au premier plan la créature la plus performante avec les meilleurs gènes. De plus, les scores de toutes les générations lancées sont affichés.



Lorsqu'on étudie la technique d'optimisation de notre algorithme avec un graphe représentatif d'un nombre considérable de générations, on constate une hausse rapide du score lors des premières générations, ensuite une stabilisation du score. Par conséquent, notre algorithme sur plusieurs générations tendra à converger vers des gènes très similaires en raison d'un risque de mutation plus élevé (ce risque de mutation est extrêmement élevé par rapport au taux de mutation naturel).

Avec notre exemple simple, nous avons un score maximal atteignant 91/100, et une stabilisation à partir de la 40e génération. Nous constatons en outre qu'après 2000 générations, nous trouvons une solution extrêmement proche de cela (à moins de 4 chiffres significatifs). L'avancement de l'algorithme génétique peut être vu ci-dessous :



L'organisation

Tout d'abord, nous tenons à remercier M. Meyer pour son investissement dans notre projet. Ses conseils pour notre projet ont été d'une grande aide.

Organisation:

En ce qui concerne l'organisation de notre groupe pour ce projet, nous étions en synchronisation dans les différentes tâches à effectuer. Nous travaillions en présentiel les lundi matin de 8h à 13h dans les salles du quai 43 avec la présence du professeur pour un suivi des versions de notre projet ainsi que des conseils pour son avancement.

Communication:

Hormis les séances en présentiel, on communiquait également sur un serveur Discord par écrit et par outil vocal.

Outils de développement:

Pour le partage de notre code, on utilisait principalement la plateforme GitHub. Chacun travaillait sur une branche différente du programme, ce qui était plus efficace et plus productif. On ajoutait à la branche principale "main" les éléments du code qui fonctionnaient et qu'on pouvait garder pour le rendu final.

Pour l'écriture du code, on utilisait plus Visual Studio Code qui possède une interface meilleure et est simple d'utilisation de par sa proximité avec GitHub.

Pour le moteur de jeu, on a installé Unity Hub nous permettant de configurer Unity3D en faisant en sorte d'avoir une version similaire pour tous les trois.

Prise de décisions:

Il n'y a pas de leader au sein du groupe. Toutes les prises de décisions ont été faites à l'unanimité avec l'accord de tous les membres du groupe.

Néanmoins, Laura a supervisé l'organisation de l'avancée globale du projet et les différentes étapes à effectuer.

Les difficultés

Prise en main de Unity:

Tout d'abord, la compréhension et la prise en main de Unity nous a pris du temps. En effet, le grand nombre de méthodes disponibles et permettant de faire un grand nombre de choses nous ont un peu dépassés.

Projets non aboutis:

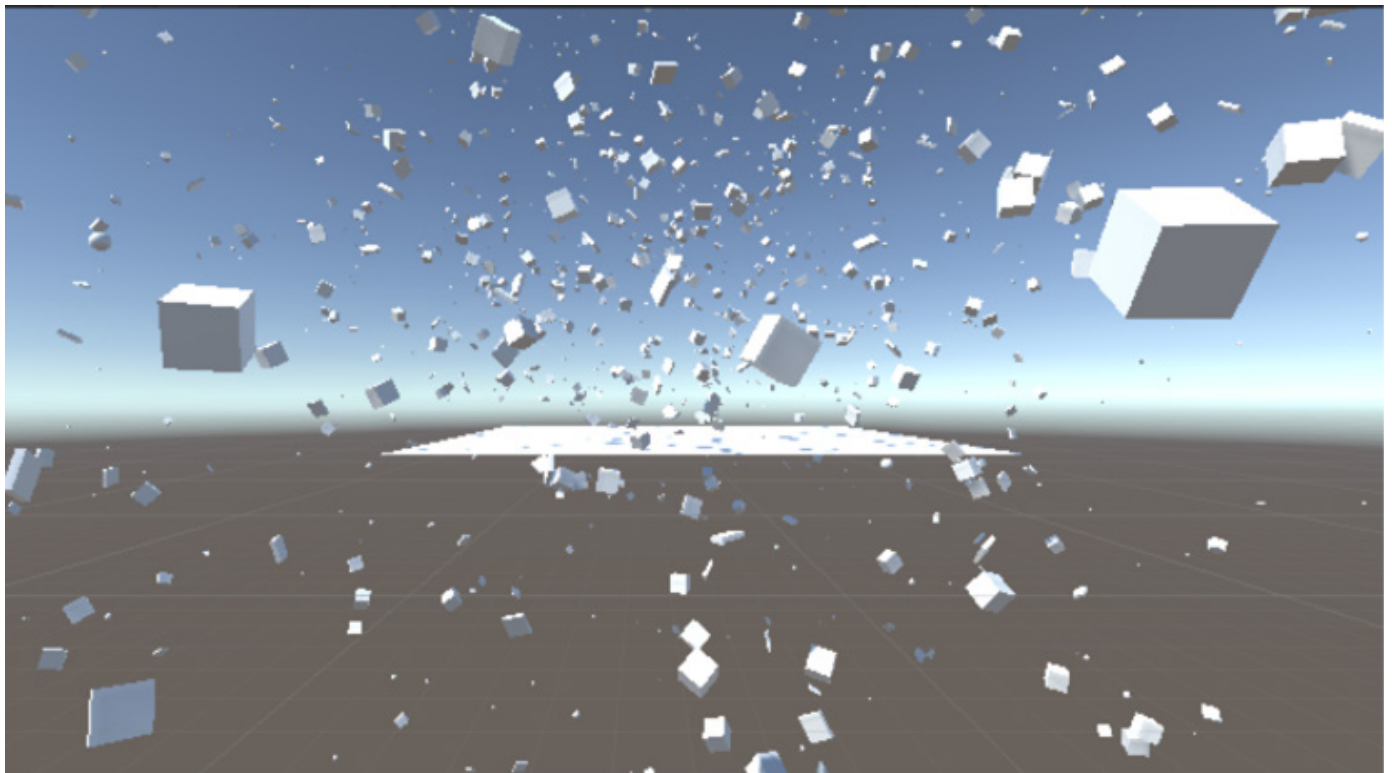
Aussi, nous avons pensé à mettre en place des obstacles entre les deux falaises pour rendre plus difficile l'accès à la zone cible aux oiseaux.

Pour ce qui est de la créature, notre première idée avait été de la construire directement dans la scène. De plus, les ailes et la queue étaient reliées au corps de notre créature via des character joints (articulation à rotule qui permet de limiter l'articulation sur chaque axe).

On ajoutait ensuite des forces à chaque aile et queue, ce qui faisait voler notre créature. Cependant, cette façon de faire s'est avérée trop compliquée à mettre en place et nous a posé problème lors de l'instanciation de plusieurs créatures dans la scène.

Nous avons aussi trouvé le Prefab d'un bel oiseau sur internet. Nous avons toutefois choisi de privilégier une créature simple et fonctionnelle avant de nous attaquer au fonctionnement de ce Prefab, que nous avons ensuite complètement abandonné.

Nous avons alors opté pour un Prefab plus simple et plus maniable.



Exemple de résultats infructueux

Bilan

Le projet:

Malgré les difficultés, nous avons réussi à présenter un projet fini.

L'équipe:

Nous avons réussi à créer une bonne cohésion d'équipe. Bien que nous ayons parfois manqué en efficacité, nous avons réussi à répartir le travail et faire en sorte que chaque membre du groupe puisse prendre une part significative au projet et participer efficacement à son élaboration.

Les pistes d'améliorations:

Certaines étapes du projet on mises plus de temps que prévu à être réaliser. Autrement, nous aurions pu apporter d'autres améliorations au projet.

En plus des améliorations que nous n'avons pas ajouté au projet car elles n'étaient pas fonctionnelles, nous aurions pu améliorer notre créature, par exemple en lui implémentant une intelligence lui permettant de prendre en compte son environnement.

Le cadre:

Bien que notre groupe était physiquement présent à toutes les séances de travail, nous avons sans doute manqué un peu de communication pour présenter l'avancer de notre projet à notre encadrant.

Néanmoins, sa disponibilité nous a tout de même permis de nous débloquent quand nous n'arrivions plus à avancer et d'aboutir à un projet fonctionnel.

