



PROJET D'INFORMATIQUE

Simulation de foule avec transmission d'émotions

D.ino GAME

Rapport de fin de projet

Lucie LEOPOLD, Julie PERE, Clément VIVIER

2020 - 2021

SOMMAIRE :

I) Introduction

II) Modélisation

- A. Mouvement de foule
- B. Contagion
- C. Gameplay

III) Développement

- A. Détails du travail réalisé
- B. Design
- C. Points clefs de la simulation

IV) Test

- A. Comportement de foule
- B. Echelle d'émotion
- C. Attirance pour les bus

V) Synthèse

- A. Conclusion
- B. Améliorations

VI) Bibliographie

I) Introduction

Les simulations en informatique sont très souvent utilisées pour représenter des situations ou pour mettre en avant des événements. Elles permettent ainsi de représenter les multitudes de cas possibles d'une situation voire même de prédire certains cas de figure. Dans le cadre de notre projet, nous nous sommes intéressés aux simulations de mouvement de foule. Elles consistent à reproduire numériquement le déplacement et les interactions d'un grand nombre d'individus. On peut retrouver ce type de simulation dans les jeux-vidéos ou bien dans les analyses prédictives de taux de reproductions d'un virus dans une foule, ou encore de transmission d'un sentiment comme la panique, par exemple.

L'objectif de notre projet est de créer une simulation qui modélise les comportements de dinosaures au sein d'un troupeau. Cela revient à étudier la mise en place et la transmission d'une émotion spécifique d'un dinosaure à un autre, puis de faire évoluer leurs comportements en fonction de cela. Notre simulation va donc reposer sur plusieurs paramètres : la simulation d'un comportement de foule, la transmission d'émotions suivant des cas de figures spécifiques puis la modification du comportement de ces individus en fonction de leurs émotions.

Cette simulation est aussi une manière pour notre groupe de rendre une blague qui nous est chère, réalité.

Pour réaliser notre simulation, nous avons utilisé le moteur de jeux GODOT. C'est un logiciel qui permet la conception de jeux en 2D ou en 3D. Il utilise un langage de programmation semblable au Python, appelé le GDscript. Enfin, tout jeu créé sous GODOT appartient entièrement au développeur. C'est donc une façon plutôt accessible de réaliser une simulation.

II) Modélisation

A. Mouvement de foule

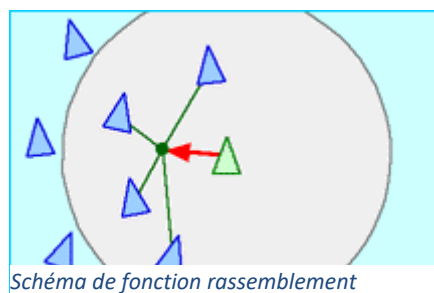
Les mouvements de foule peuvent être définis de plusieurs façons. On peut établir que le mouvement est provoqué par la masse d'individus, par des groupes d'individus ou bien encore par les individus eux-mêmes. Pour notre simulation, nous allons utiliser la dernière approche, soit celle qui considère chaque individu comme une entité à part entière.

L'algorithme des BOIDS est un des algorithmes représentant la foule de cette façon. A l'origine, Craig REYNOLDS a voulu développer une simulation permettant de modéliser le vol d'un essaim d'oiseau. Cette modélisation fonctionne suivant trois règles que nous avons appliquées dans notre projet.

Chaque agent de notre simulation possède sa propre vitesse, position et direction tout en appartenant à un groupe que l'on nomme **essaim** (swarm en anglais). Trois règles définissent ensuite le comportement de chacun de nos individus, ce qui au final crée un mouvement de foule.

Ces trois règles sont : le **rassemblement** (gather), l'**éloignement** (separation) et l'**alignement** (alignment).

La règle de rassemblement fait que chaque agent va s'orienter vers un point équidistant de tous les autres agents dans un rayon spécifique. Ainsi, les agents vont avoir tendance à se rapprocher les uns des autres pour garder une sorte de cohésion.



La règle de rassemblement est compensée par celle de l'éloignement. Avec cette règle, chaque agent va avoir tendance à rester à la même distance de tous les autres agents autour de lui dans un périmètre spécifique.

Les règles de rassemblement et de d'éloignement combinées permettent à notre essaim d'individus de rester groupés tout en gardant une distance raisonnable les uns des autres.

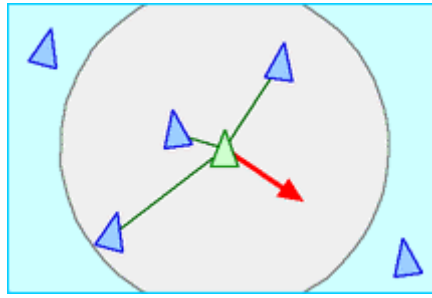


Schéma de fonction éloignement

Enfin, pour que notre essaim se déplace comme il le ferait dans la nature, chaque agent doit suivre la même direction que ses voisins. De ce fait, chaque agent va adapter sa direction à celle de ses voisins dans une zone donnée.

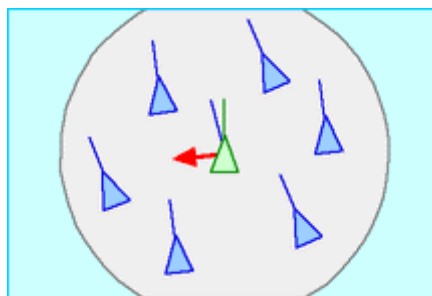


Schéma de fonction alignement

Toutes ces règles sont définies par des fonctions dans notre simulations. Ces fonctions renvoient des vecteurs qui correspondent à une direction, multiplié par un coefficient. Les coefficients permettent de définir l'intensité d'application de la règle.

B. Contagion

Les contagions peuvent être représentées par des modèles mathématiques comme celui du S.I.R par exemple. Ce modèle est dit à compartiments, ce qui signifie que l'on divise la population étudiée en plusieurs catégories : les personnes saines (**susceptible**), les personnes infectées (**infected**) et les personnes guéries (**recovered**).

Avec ce type de modèle, on imagine que tout part d'un patient dit « zéro », porteur de la maladie. Cet agent infecté va évoluer dans une foule et transmettre la maladie aux personnes saines, plus ou moins susceptibles d'être contaminées. Dès lors, le nombre d'infectés va augmenter et le nombre de susceptibles va diminuer au sein de la population jusqu'à une certaine période, où les individus malades deviennent guéris. Plus le temps s'écoulera, plus une immunité collective sera visible grâce au nombre croissant d'individus guéris et au nombre décroissant d'individus infectés.

Ce modèle permet de décrire de façon hypothétique l'évolution d'une épidémie. Hypothétique puisque d'autres facteurs sont à prendre en compte dans ce type d'étude, comme la contagiosité d'un virus ou encore son mode de transmission.

Les émotions peuvent être considérées comme des virus imperceptibles. En effet, les humains ont tendance à ressentir les émotions autour d'eux et à les reproduire. Il peut y avoir des transmissions d'émotions avec des « individus zéros » ; comme par exemple quand une personne triste arrive dans un groupe, elle peut faire baisser le moral du reste des individus du groupe. Il peut aussi y avoir des transmissions plus situationnelles. Par exemple, dans un environnement exigu comme un bus, s'il y a beaucoup de monde, les personnes dans le bus vont presque toutes commencer à ressentir un sentiment de stress ou de panique monter dû à une forte proximité. Ressentir ou non ce stress dépend du nombre de personne dans le bus ainsi que de l'état émotionnel de chacun.

Pour notre projet, nous nous sommes inspirés du modèle SIR et avons imaginé des critères d'apparition d'émotion pour les dinosaures dans leur troupeau. Chaque dinosaure possède une variable d'émotion qui définit s'ils ont peur, s'ils sont en colère ou s'ils sont dans leur état normal. Celle-ci est initialisée aléatoirement entre 5 et 15 au début de la simulation. Ils sont donc tous dans un état neutre au départ. Puis au cours de la simulation, leur émotion va évoluer en fonction du nombre de voisin qu'ils ont. Le schéma ci-dessous explique les cas de figure qui font fluctuer leur émotion.

Echelle d'émotion des dinosaures

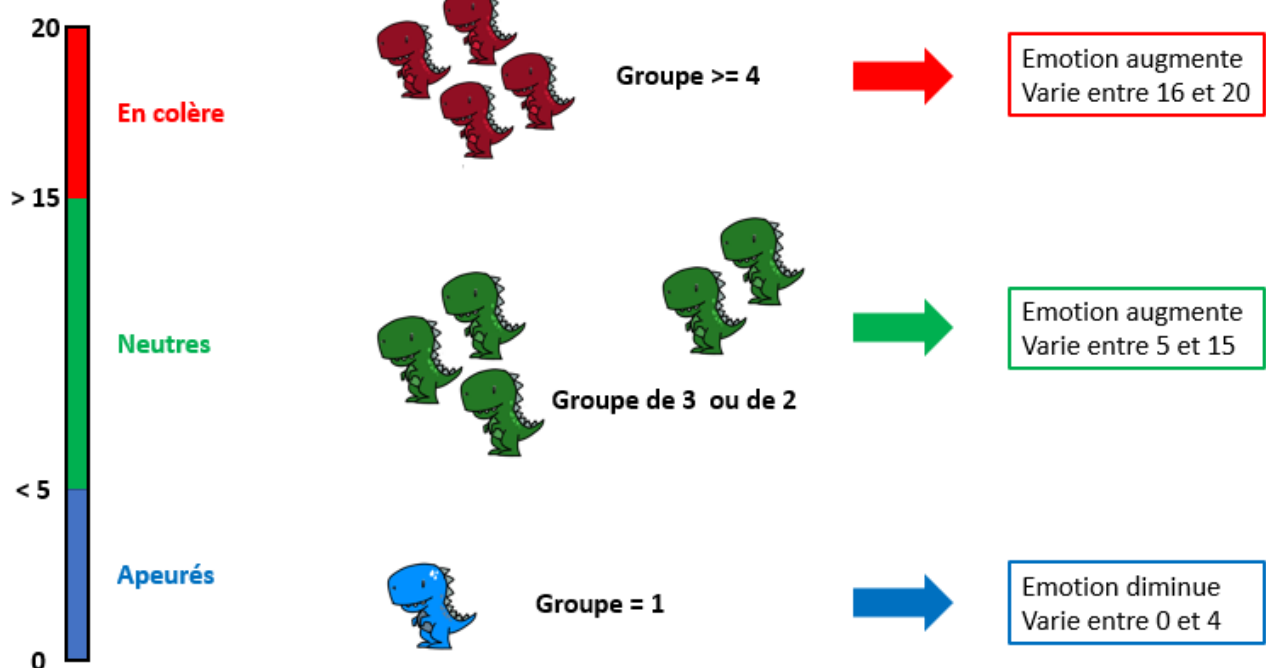


Schéma de la variation d'émotion des dinosaures en fonction de la taille du groupe

On notera que les couleurs sur le schéma correspondent aux couleurs utilisées dans la simulation.

Il existe deux cas de figure spécifiques qui modifient aussi l'émotion des dinosaures : lorsqu'ils sont trois ou lorsqu'ils sont deux.

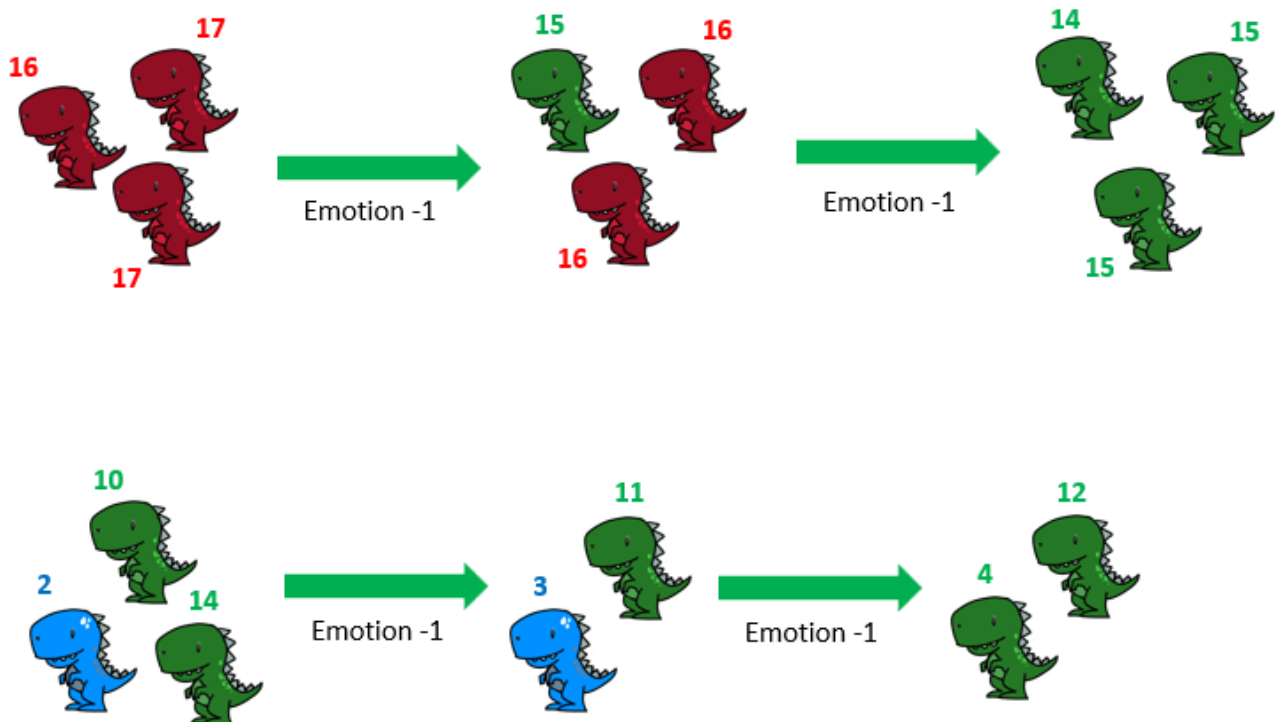


Schéma des cas de figure faisant évoluer l'émotion des dinosaures

C. Gameplay

Notre simulation est un jeu à « zéro joueur ». Cela signifie que tous les personnages interagissent entre eux sans aide extérieur. Il est plus question d'étudier le comportement aléatoire des dinosaures en fonction de critères que de jouer à un jeu à proprement parler.

La simulation démarre toute seule. Des dinosaures et des bus évoluent sur le plan d'une ville. Ils appartiennent respectivement à leur essaim et par conséquent ne sont pas régis par les mêmes paramètres de mouvement de foule. Les bus sont seulement autorisés à se déplacer dans les rues de la carte tandis que les dinosaures peuvent évoluer partout dans la fenêtre. De plus, les dinosaures ont une échelle d'émotion qui évolue au cours du temps et en fonction de la taille des groupes de dinosaures.

Lorsque les dinosaures sont neutres, ils ont un comportement de foule par défaut. Lorsqu'ils deviennent en colère, ils ont tendance à rester proches les uns des autres. Ils sont

aussi attirés par les bus. Leur attirance pour les bus l’emporte sur leur volonté de se déplacer comme une foule. En étant en colère, ils ont la capacité de renverser les bus. Ils chercheront à renverser chaque bus qui roule sur la carte jusqu’à que tous les bus soient cassés. Lorsqu’ils sont apeurés, ils ont tendance à vouloir se rapprocher le plus vite possible du groupe de dinosaures le plus proche, afin de ne pas rester seul.

Enfin, la simulation s’arrête quand tous les autobus sont cassés. Un menu de fin s’affiche à l’écran, proposant au « joueur » de relancer la simulation ou bien de la quitter.

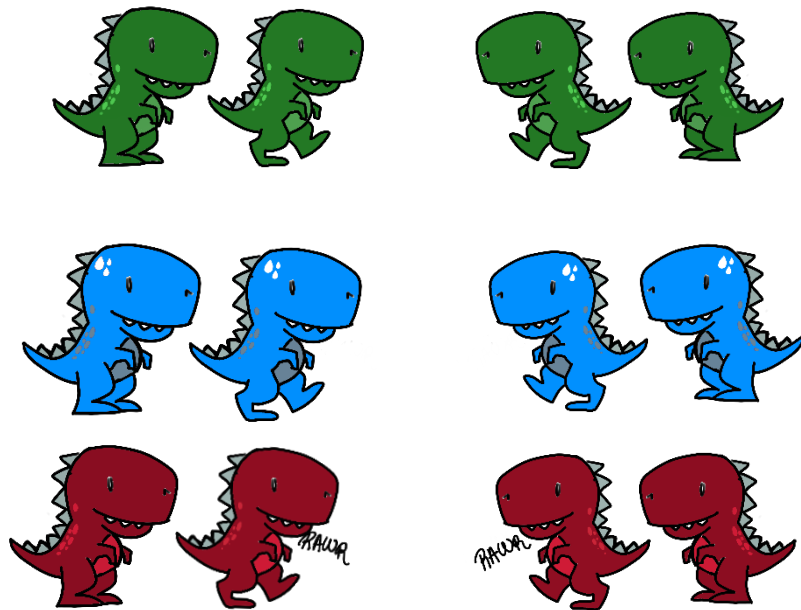
III) Développement

A. Détail du travail réalisé

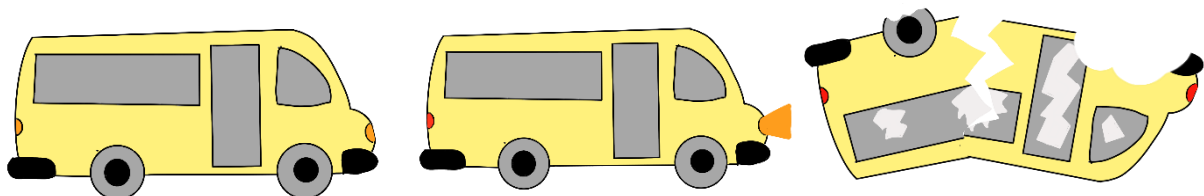
Tâches	Détails	Personne
Conception du jeu	<ul style="list-style-type: none"> - Dinosaures : méchants - Echelle d’émotion pour les dinosaures - Peur, colère, neutre - Bus ne se déplaçant que dans les rues d’une carte - Dinosaures peuvent renverser le bus si en colère - Bus se cassent quand ils sont renversés - Dinosaures attirés par les bus 	Equipe
Design du jeux	<ul style="list-style-type: none"> - Dessin de la carte - Dessin des dinosaures - Dessin des bus 	Clément Julie
Déplacement de foule	<ul style="list-style-type: none"> - Fonction gather - Fonction separate - Fonction alignment 	Equipe
Contagion	<ul style="list-style-type: none"> - Echelle d’émotion - Variation émotion - Compter les dinosaures proches 	Lucie Julie
Bus	<ul style="list-style-type: none"> - Se déplacer dans les rues - Collisions avec les murs 	Clément
Dinosaures	<ul style="list-style-type: none"> - Renverser les bus - Attirés par les bus - Modification de comportement 	Lucie
Rapport et soutenance	<ul style="list-style-type: none"> - Rédaction rapport - Faire le PowerPoint de la présentation 	Equipe

B. Design

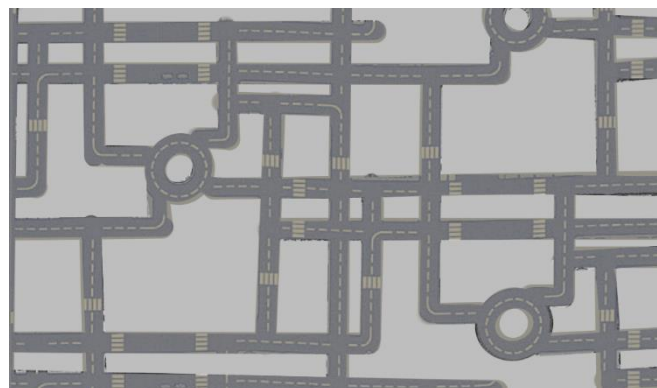
Dans Godot, les designs et les animations des personnages se font avec des Sprites. Afin de donner une impression de mouvement à nos dinosaures nous avons dessiné quatre types de dinosaures permettant de les faire marcher vers la droite ou vers la gauche. Ils sont déclinés en 3 couleurs pour représenter leur état émotionnel.



Design des différents types de dinosaures



Design des autobus

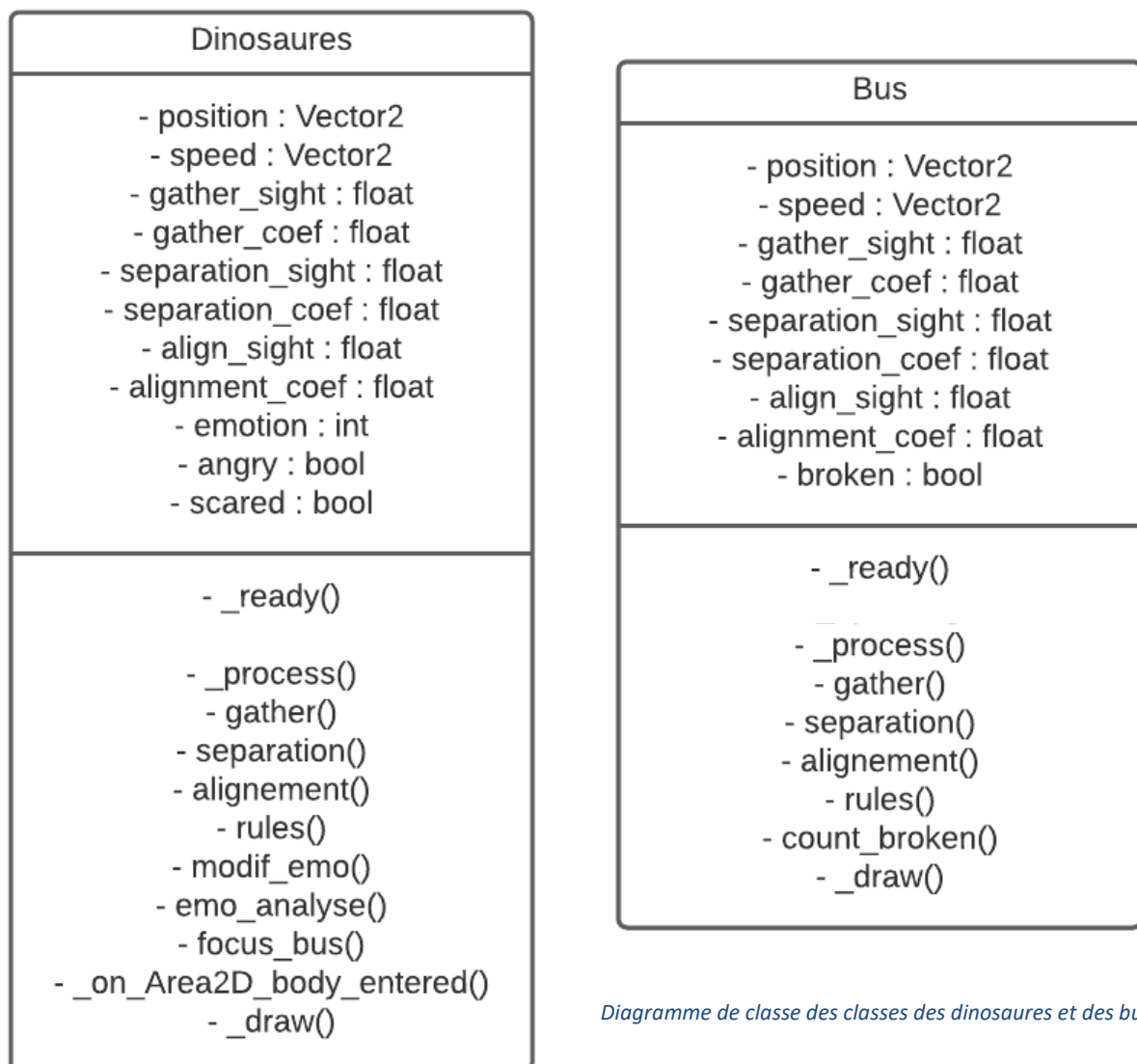


Design de la carte

C. Points clefs de la simulation

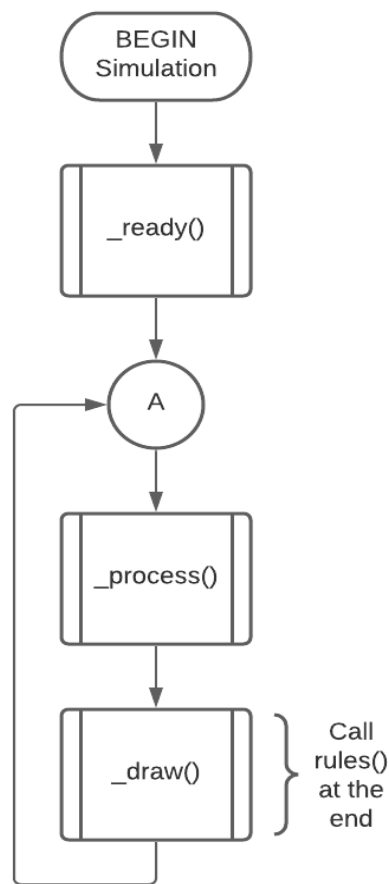
Ce paragraphe a pour but d'énoncer les points importants du développement du projet afin que n'importe qui puisse le reprendre et l'améliorer facilement.

Nous utilisons la programmation orientée objet pour notre projet. En effet, une simulation de foule fait intervenir plusieurs fois des objets de mêmes natures. Nos deux classes d'objet ici sont les dinosaures et les bus.



Dans Godot, relier les différentes scènes se fait grâce à : *export (PackedScene)*. Dans notre cas, cela permet de lier les scènes de nos objets à notre scène principale appelée main. On retrouve dans le script main tout ce qui permet de faire fonctionner le projet soit l'instanciation de nos bus et de nos dinosaures ainsi que l'appel de toutes leurs méthodes via la fonction `rules()`.

L'appel des fonctions dans un script Godot se fait dans un ordre précis afin que chaque élément soit créé puis fonctionne comme il faut tout au long de la simulation.



Logigramme du fonctionnement de la simulation

La fonction `_ready()`, qui est une fonction `built_in`, permet d'initialiser tous les paramètres de nos objets et de les instancier.

La fonction `process()` est appelée à chaque nouvelle image de la simulation après un certain temps écoulé entre deux image nommé `delta`. Elle permet donc de gérer tout ce qui est en rapport avec le déplacement, la vitesse ou le comportement des agents.

La fonction `draw()`, dans le cadre des dinosaures et des bus permet d'afficher leurs sprites.

La boucle de notre simulation se termine par l'appel de la fonction `rules()` contenant toutes les méthodes propres aux dinosaures et aux bus. Son appel se fait dans la fonction `process()` du script `main`.

La fonction `rules()` contient donc toutes les méthodes des dinosaures et toutes celles des bus. Cela permet de n'appeler qu'une seule fonction pour exécuter toutes ces méthodes, dans le fichier `main`.

Les bus sont les seuls objets capables d'interagir avec des éléments de la carte, ici les bâtiments. Pour cela, il faut faire attention au type de nos bus ainsi qu'à leur mode de déplacement. Nos bus sont donc des « Kinematic bodies ». Il faut aussi qu'ils soient constamment présents dans la fenêtre afin de pouvoir rentrer en collisions avec les bâtiments ou d'autres bus. Les fonctions « `move_and_collide` » et « `move_and_slide` » permettent cela. Chaque bus et bâtiment possèdent donc une « collision shape » afin que le logiciel puisse savoir quand deux objets entrent en contact.

Nos dinosaures quant à eux sont des « Area 2D ». Leur mode de déplacement s'apparente plutôt à la façon dont on fait des dessins animés ; c'est-à-dire que leur dessin change à chaque écoulement de temps `delta` à leur nouvelle position. Afin de pouvoir détecter qu'un dinosaure entre en collision avec un bus, nous utilisons les signaux de Godot. Les signaux permettent de donner des indications aux objets de la simulation dans des cas

de figures spécifiques. Dans notre cas, nous détectons dès qu'un corps ici un bus entre dans la zone d'un dinosaure à l'aide de la fonction `built_in « _on_Area2D_body_entered »`. Lorsque cette entrée est détectée, la variable de définition de l'état du bus passe à cassé.

La partie des dinosaures attirés par les bus lorsqu'ils sont en colère reprend le même principe que la fonction rassemblement. En effet, nous avons fait en sorte de trouver la position du bus le plus proche d'un dinosaure rouge puis nous orientons le dinosaure dans cette direction. Cette direction est en fait un vecteur entre le dinosaure et le bus. Nous ajoutons ce vecteur à la vitesse. La vitesse dans Godot est elle aussi un vecteur. Puisqu'un vecteur est orienté, il peut servir de direction. Afin de faire en sorte que nos dinosaures aient plus tendance à être attirés par les bus que par la volonté de se déplacer comme une foule, nous avons multiplié le vecteur direction vers un bus par un coefficient plus grand que les coefficients de rassemblement, de séparation et d'alignement.

IV) Tests

A. Comportement de foule

Notre simulation de comportement de foule est étudiable tout d'abord via nos deux types d'agent. Comme dit précédemment, les bus et les dinosaures n'appartiennent pas au même essaim ; ils évoluent suivant leurs propres règles de mouvement de foule. Ces règles sont définies à la fois par des coefficients mais aussi par l'importance du périmètre autour d'un agent permettant d'appliquer une des règles.

On constate bien que les dinosaures et les bus n'ont pas les mêmes façons de se déplacer. Les dinosaures forment bien une sorte de troupeau tandis que les bus ont plus tendance à évoluer indépendamment les uns des autres. Cela reflète assez bien ce qui pourrait se passer dans la réalité. De plus, les bus possèdent ce type de comportement afin de pouvoir se déplacer plus facilement dans les rues de la carte.

Nous pouvons vérifier l'évolution du comportement de notre foule de façon un peu plus poussée grâce aux variations d'émotion des dinosaures. Nous avons défini un nouveau comportement au sein du troupeau en fonction du sentiment de peur ou de colère. Une fonction d'analyse émotionnelle, permet de définir dans quel état sont les dinosaures et de modifier leur comportement en conséquence. Ainsi les dinosaures rouges, en colère, se transmettent leur colère les uns aux autres et alimentent ce sentiment en restant groupés. Plus les dinosaures rouges restent groupés, plus ils sont en colère longtemps et plus ils cherchent à rester au contact d'autres dinosaures en colère. On remarque donc au sein de notre troupeau de dinosaures un mouvement de foule indépendant, propre à la colère. Cette situation est aussi perceptible avec les dinosaures apeurés. Une fois un dinosaure devenu bleu, car excentré des autres, on remarque qu'il s'oriente le plus vite possible vers un groupe de dinosaures proches de lui. Finalement, nos dinosaures, n'ont pas un mais 3 mouvements de foule différents adaptés à leurs sentiments.

B. Echelle d'émotion

Afin de représenter une variation d'émotion au cours du temps, nous avons choisi de définir une échelle d'émotion variant de 0 à 20. Les valeurs 0 et 20 sont des extremums, l'émotion des dinosaures ne peut ni aller en-dessous de 0 ni au-dessus de 20. Cela nous permet d'être certain de toujours travailler avec une échelle fixe et de connaître les paliers et les écarts de cette échelle. Dès lors, nous avons contraint notre fonction de modification du niveau d'émotion à ne fonctionner que pour un niveau d'émotion entre 0 et 20. Cela signifie que toutes valeurs d'émotions allant en-dessous ou au-dessus de ces 2 valeurs seront mises par défaut à 0 ou à 20.

Comme expliqué précédemment, la modulation des émotions des dinosaures se fait en fonction de la taille du groupe. Il fallait penser à prendre en compte tous les cas de figures permettant de représenter au mieux notre transmission de sentiments. Il fallait que les niveaux d'émotions puissent varier dans certaines configurations uniquement. Pour ce faire, nous avons dû compter le nombre de dinosaures dans une zone définie. Puis nous avons testé la valeur de ce nombre pour établir les zones à forte densité d'agent et celle à faible densité. Il fallait aussi tester leur nombre en fonction du type d'émotion déjà ressenti par les dinosaures. En effet il faut faire diminuer l'émotion des dinosaures rouges quand ils se retrouvent moins nombreux tandis qu'il faut faire augmenter celle des dinosaures bleus dans cette même situation. Une fois toutes ces conditions bien vérifiées et testées on a bien une répartition des émotions des dinosaures en fonction de leur proximité.

C. Attirance pour les bus

L'attirance pour un bus se caractérise par un vecteur vitesse qui indique la direction du bus le plus proche. Pour réaliser cela, il fallait tester quels bus étaient cassés et lesquels ne l'étaient pas. De plus les dinosaures ne peuvent être attirés que par les bus non-cassés, il fallait donc supprimer de la liste des « proies » possibles, les bus qui venaient d'être renverser par les dinosaures.

Une fois la liste de bus susceptibles d'être attaqués définie, on teste lequel est le plus proche d'un dinosaure rouge et on oriente le dinosaure dans sa direction jusqu'à qu'ils entrent en contact. Afin que les dinosaures rouges restent groupés, nous orientons tous les autres dinosaures en colère autour vers le bus le sélectionné.

Le problème principal a été de rendre cette attirance visible dans la simulation. Pour cela il fallait que l'attirance soit plus forte que l'effet de foule sans pour autant l'annuler. Pour arriver au résultat de la simulation, nous avons testé visuellement ce que la simulation donnait pour différents coefficients d'alignement, de rassemblent, d'attirance.

V) Synthèse

A. Conclusion

Notre projet comprend différentes parties et différents schémas de fonctionnement qui peuvent s'appliquer à plusieurs objets. En effet, nos dinosaures constituent à eux seuls, plusieurs foules différentes et indépendantes du fait de leurs émotions. Notre simulation montre bien qu'il y a une interaction à la fois entre les membres d'un même groupe comme les dinosaures ainsi qu'entre les membres de différents groupes comme entre les dinosaures et les bus. Elle permet aussi d'illustrer le rapport qu'il peut y avoir entre la densité d'individus et l'augmentation d'un paramètre comme les émotions dans notre cas. Nous sommes donc finalement arrivés au bout de ce semestre à une simulation de foule cohérente et à une représentation graphique plutôt compatible avec notre situation de départ.

B. Amélioration

Notre simulation n'est cependant qu'un point de départ ; de nombreuses autres fonctionnalités pourraient être apportées. Par exemple un troisième agent pourrait venir réparer les bus. On pourrait imaginer transformer notre simulation en jeu dans ce cas-là, en rendant ce troisième agent jouable par quelqu'un. Le but serait alors de réparer tous les bus sans se faire toucher par des dinosaures rouges. D'autres émotions pourraient aussi venir complexifier la simulation. Modifier le comportement des dinosaures en fonction d'éléments sur la carte serait une encore une autre possibilité. Enfin, bien sûr, nous pourrions chercher à optimiser la simulation et à réduire le nombre de lignes de code autant que possible.

VI) Bibliographie

- Wong, Timm. "Boids." [Stanford.edu](https://cs.stanford.edu/people/eroberts/courses/soco/projects/2008-09/modeling-natural-systems/boids.html), Sept. 2008,
cs.stanford.edu/people/eroberts/courses/soco/projects/2008-09/modeling-natural-systems/boids.html.
- Contributeurs aux projets Wikimedia. "Type de Modèle Mathématique En Épidémiologie." [Wikipedia.org](https://fr.wikipedia.org/wiki/Mod%C3%A8les_compartimentaux_en_%C3%A9pid%C3%A9miologie), WikimediaFoundation, Inc., Mar. 2010,
fr.wikipedia.org/wiki/Mod%C3%A8les_compartimentaux_en_%C3%A9pid%C3%A9miologie.
- Füllsack, Manfred. "SIR - a Model for Epidemiology." [Uni-Graz.at](https://uni-graz.at/systems-sciences.uni-graz.at/etextbook/sw2/sir.html), systems-sciences.uni-graz.at/etextbook/sw2/sir.html.
- "Contagion Émotionnelle : Comment Transmettons-Nous Nos Émotions Aux Autres ? - Nos Pensées." *Nos Pensées*, 16 Aug. 2017, nospensees.fr/contagion-emotionnelle-transmettons-nos-emotions-aux-autres/.
- Linietsky, Juan, et al. "Godot Docs." *Godot Engine Documentation*, 2014,
docs.godotengine.org/en/stable/.
- Shake Shake Go. "Shake Shake Go - Dinosaur [OFFICIAL VIDEO]." *YouTube*, 23 Feb. 2018,
www.youtube.com/watch?v=vV7DrNCoRro&ab_channel=ShakeShakeGo. Accessed 17 June 2021.s