

# Particules émergentes

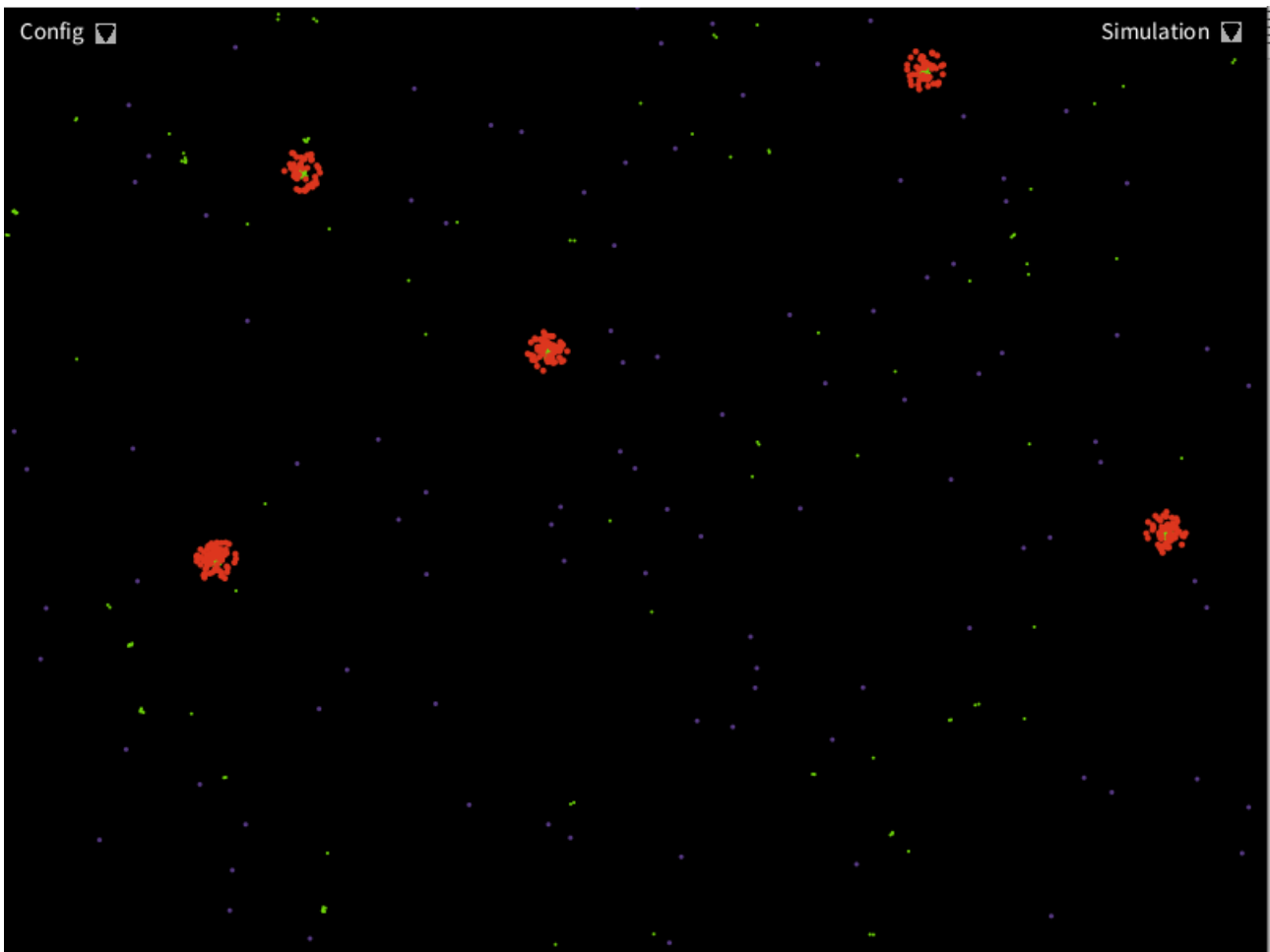
**1-Introduction**

**2-Principe de fonctionnement**

**3-Code**

**5-Résultats**

**6-Références, sources, inspirations**



# Introduction

Un système de particules qui présente des comportements émergents est un système où un certain nombre de particules interagissent entre elles selon certaines règles simples et où l'ensemble des particules adoptent des comportement complexes qui n'avait pas été définis ou anticipés au départ. Mon projet a pour but de donner un moyen d'explorer facilement un grand nombres de ces systèmes avec des paramètres différents et de permettre à l'utilisateur d'interagir directement avec le systeme.

Je commencerais par présenter les lois théoriques du système que j'ai mis au point ainsi que les possibilités d'interaction de l'utilisateur, puis je montrerais la manière dont j'ai mis ces principes en code, les techniques et paradigmes utilisés. Ensuite je montrerais les différents résultats obtenus et j'exposerais les différentes structures émergentes que mon système génère. Enfin, je présenterais les différentes inspirations et références que j'ai utilisées pour construire ce projet, ainsi que mes sources.

## Principe de fonctionnement

Les systèmes de particules que j'ai implémentés est composées d'un certain nombres d'espèces de particules qui possèdent chacune des caractéristiques différentes ce qui fait que chaque particule interagira avec les autres en fonction des caractéristiques de son espèces.

Chaque particules possède un vecteur représentant sa position, un représentant sa vitesse et peut subir un accélération. Chaque particule possède également une couleur et une taille mais qui n'influent pas sur son comportement.

Chaque particule est en interaction avec les autres de la manière suivantes :

- *Chaque particule trop proche (en deçà d'un certain rayon) repousse la particule avec une certaine force*
- *Toutes les autre particules suffisamment proche, en deçà d'un autre rayon, attirent la particule avec une force dépendante des paramètres et inversement proportionnelle au carré de la distance entre les deux particules*
- *Les particules dépassant le cadre de la simulation ont leur vecteur vitesse renversé de manière à toujours pointer vers la simulation, de plus, les particules subissent un accélération vers le centre de la simulation.*

La force donnée à l'attraction ou la répulsion d'une particule par une autre dépend des paramètres des deux particules.

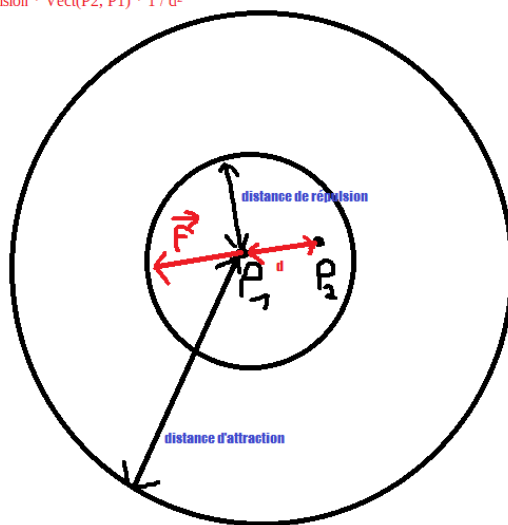
Chaque particule possède donc les caractéristiques suivantes : une couleur, une taille, une distance de vision pour voir et être attiré par les autres particules, une force avec laquelle les autres particules l'attirent, une distance de répulsion, une force de répulsion et enfin un vitesse maximum afin d'éviter que le système s'emballe.

Un des gros point du projet est aussi l'interaction du système avec l'utilisateur. L'utilisateur à ainsi la possibilité d'utiliser sa souris pour attirer ou repousser les particules proches du curseur. L'utilisateur peut aussi mettre en pause la simulation et la relancer avec des paramètres aléatoires pour chaque espèce de particules.

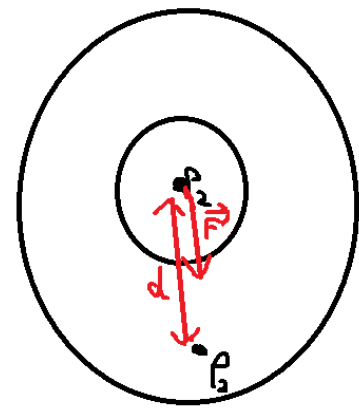
Il peut également sauvegarder les paramètres de la simulation et charger des paramètres sauvegardés précédemment.

Pour le côté graphique, j'ai également ajouté la possibilité d'ajouter de la transparence pour que les particules laissent une trace derrière elles.

$$F = \text{force Répulsion} * \text{Vect}(P_2, P_1) * 1 / d^2$$



$$F = \text{force Attraction} * \text{Vect}(P_1, P_2) * 1 / d^2$$



## Code

J'ai choisi d'utiliser le paradigme Orienté Objet pour ce projet, même si j'ai utilisé quelques techniques hybrides pour la gestion des interactions avec l'utilisateur. Considérant la taille du projet, j'ai également utilisé de la modularité avec quatre fichiers de code différents.

Chaque particule est modélisée par un objet qui contient toute ses informations et on peut interagir avec la particule via ses méthodes. Les paramètres d'une particule sont aussi contenus dans une objet dont les méthodes permettent notamment de choisir aléatoirement les paramètres en utilisant des lois gaussiennes que j'ai paramétrées de manière à obtenir facilement des simulations intéressantes.

Le système de particules est aussi un objet qui contient notamment une liste de toute les particules classées par espèces. Les méthodes du système permettent notamment de facilement réinitialiser le système avec des nouveaux paramètres aléatoires.

Le système possède aussi et surtout un méthode update() permettant de faire avancer la simulation d'un pas et une méthode display() qui permet d'afficher toute les particules, ces deux méthodes sont appelées dans le draw().

L'autre gros morceaux du code, et celui qui m'a pris le plus de temps, est celui qui gère l'interaction avec l'utilisateur. Pour faire la GUI, j'ai utilisé la librairie G4P pour Processing qui permet une utilisation relativement simple des différents éléments (boutons, checkboxes, ...) et de la gestion des événements.

J'ai utilisé la librairie globalement comme attendue avec des objets représentant les différents éléments de la GUI stockés dans d'autres objets plus grands (Config et ParamSimulation) et des méthodes/fonctions dans le fichier de code principal qui serve de gestionnaire d'événement.

Certains éléments nécessitant aussi des images spécifiques, j'ai créé quelques textures (dans assets/icones) pour remplir cette tâche.

## Résultat

Le projet fini nous permet bien d'explorer les différents systèmes et de voir les structures récurrentes qui peuvent apparaître, j'en fait une liste non exhaustive ici et j'ai déjà enregistré un certain nombre de presets dans l'application qui permettent de voir certains comportements intéressants.

Je commence par dire que certains comportements parmi les plus intéressants nécessitent une action de l'utilisateur pour être vus, soit attirer toutes les particules très proches les unes des autres, soit les repousser aux limites de la simulation avec la souris.

Parmi les formes récurrentes, je peux donc citer les « noyaux », un ensemble de particules qui s'attirent fortement entre elles sans trop se repousser et qui forme une masse mouvante qui peut parfois être très mobile. L'autre structure la plus courante est la « cellule », c'est un ensemble de particules à la fois attirées et repoussées par un noyau et qui reste relativement stable autour de celui-ci. Parfois les cellules peuvent avoir plusieurs couches de particules autour de leur noyau et parfois la couche extérieure agit comme une cage pour le noyau. Les cellules sont souvent assez fixes dans l'espace et stables, elles ont souvent un comportement de les unes avec les autres lorsqu'elles se rapprochent suffisamment.

Un autre motif récurrent est ce qu'on pourrait appeler des « particules libres » qui ont tendance à rester relativement isolées des autres particules. Certaines sont très rapides et peuvent parfois ressembler à des boîds, d'autres sont assez lentes et constituent des formations de particules qui se répartissent équitablement dans l'espace.

Et il y a encore bien d'autres structures à observer.

## Références, sources, inspirations

Le livre numérique « [The Nature of Code](#) » qui m'a apporté beaucoup de réponses, d'inspiration et de techniques, notamment pour la simulation des systèmes de particules :

<https://natureofcode.com/>

Différentes vidéos sur YouTube qui abordent le sujet des systèmes de particules émergents et de leur simulation :

[Artificial Particle Life - Simulation & Code par Brainxyz](#)

[How Particle Life emerges from simplicity par Tom Mohr](#)

La librairie G4P que j'ai utilisée pour la partie interactions avec l'utilisateur.

<http://www.lagers.org.uk/g4p/>