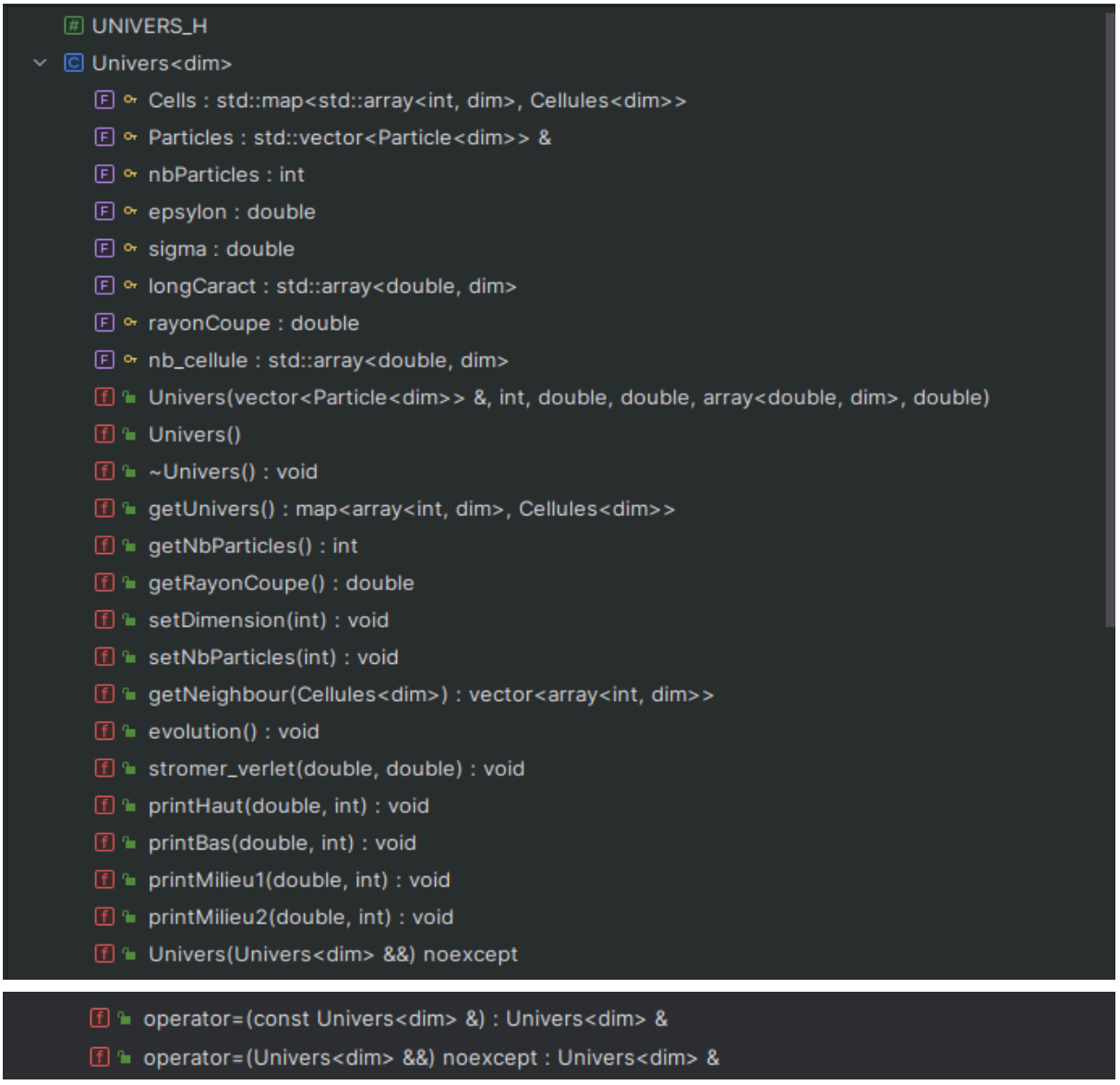
Rapport Projet C++ : Zakaria El Kazdam & Riham Faraj

Contexte:

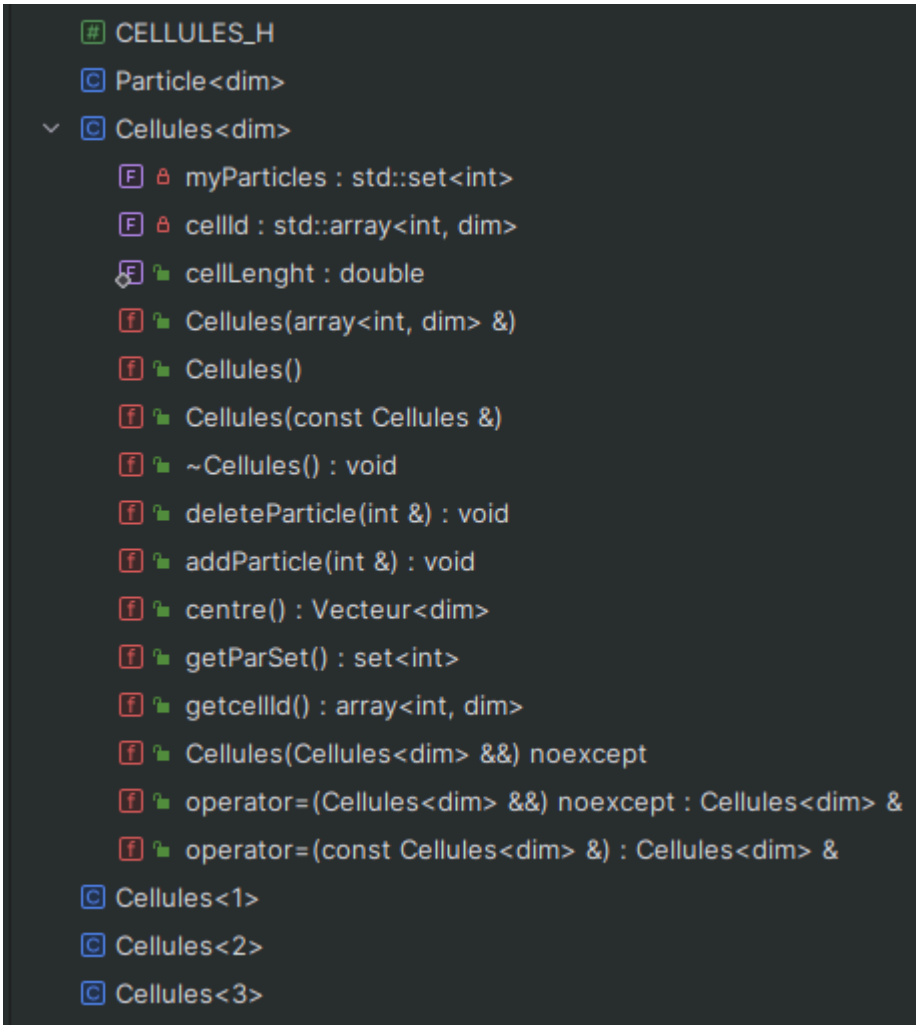
Le projet concerne la simulation de l'évolution de particules dans un univers divisé en plusieurs cellules. Chaque cellule contient des particules qui interagissent entre elles et se déplacent dans l'univers en se limitant par les longueurs caractéristiques . Le système permet à un utilisateur de définir les paramètres initiaux de la simulation, notamment la dimension de l'univers (dans l'ensemble {1,2,3}), le nombre de particules, leurs configurations initiales (position, vitesse, forces subies), le pas de temps (dt), et le temps final de la simulation (t_end).
Une fois les paramètres définis et la simulation lancée, le système génère des fichiers VTK pour chaque étape de la simulation (ces fichiers contiennent la position et la vitesse ainsi que la masse de chaque particules à chaque dt), permettant à Paraview, un outil de visualisation scientifique, de représenter graphiquement l'évolution des particules dans l'espace et le temps.

Univers:



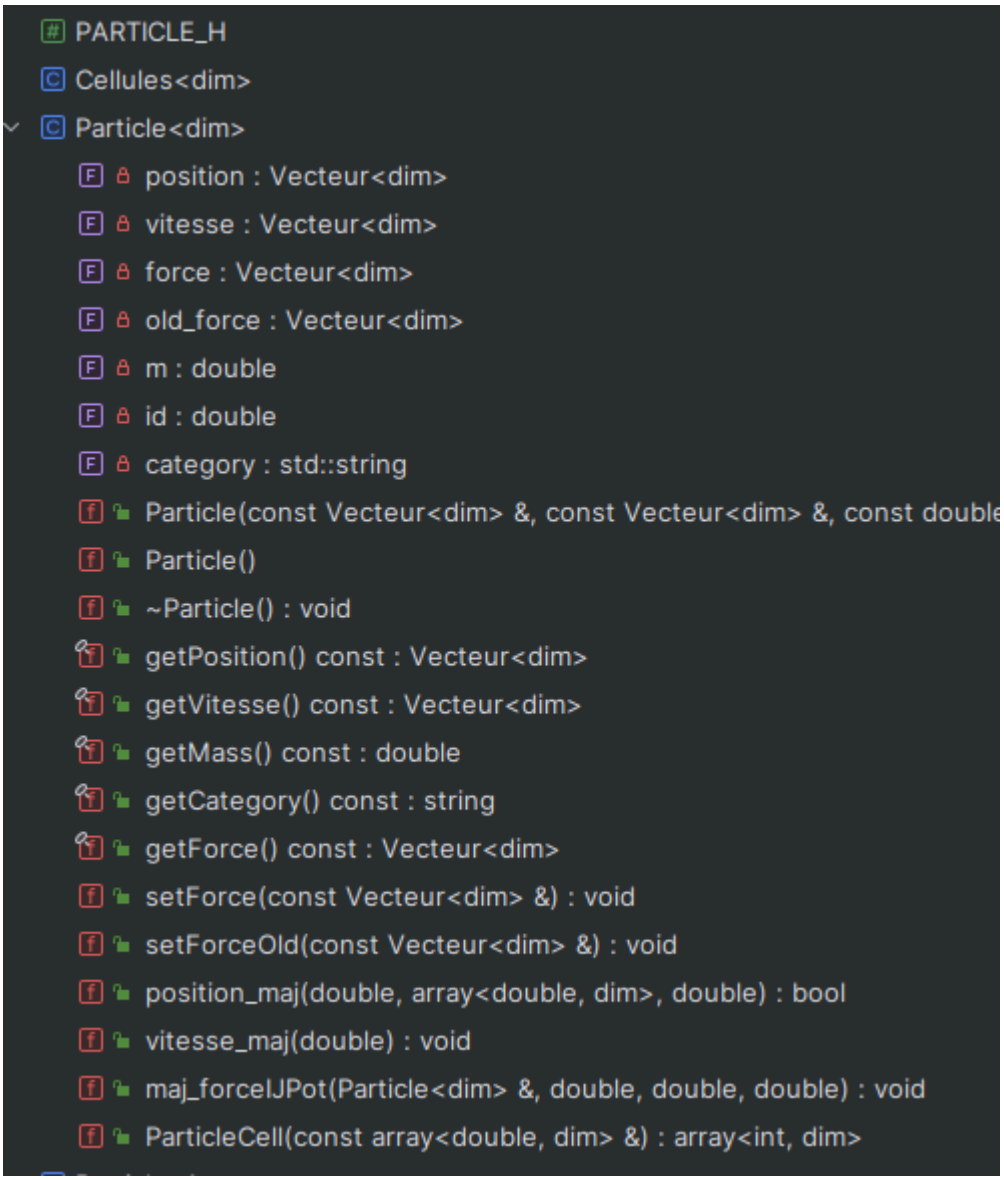
L’univers contient tout d’abord ses propriétés comme attributs tel que epsylon, sigma , rayonCoupe ainsi que les longueurs caractéristiques.
On a choisi d’utiliser Univers comme template pour qu’il soit adapté à n’importe quelle dimension afin d’éviter de charger le code par trop de if ou switch pour chaque méthode.
L’univers contient aussi une map de cellules dont les clefs sont des arrays, ces derniers identifient les cellules et leur tailles dépend de la dimension de l’Univers bien sûr.

Cellules:



Comme univers a un nombre de dimension , les cellules qui composent l’univers possèdent eux aussi la même nombre de dimensions , d'où le premier intérêt à utiliser une template .
La cellule possède comme attribut les particules qu’elle contient , en gardant que leur index dans la collection de particules utilisés pour construire l’univers au lieu des particules elle mêmes.
De plus on a choisit un set pour pour faciliter l’ajout ou la suppression d’une particule lorsque cette dernière arrive ou qui la cellule vers une autre qui est voisine , cette opération se fait O(log(n)).

Particle :



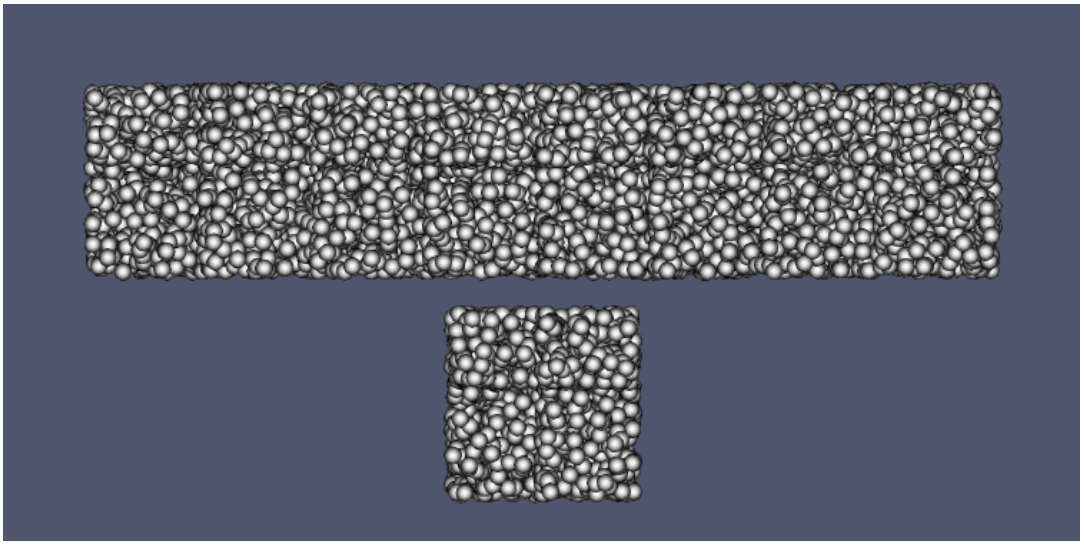
Chaque particule contient tous ses informations comme attributes sous forme d'objet de classe Vecteur , qui possède la même nombre de dimension que l'univers , cette classe est aussi une template !

Vecteur:

Vecteur est la classe de base de tout le projet , elle possède seul attribut un std::array qui contient les coordonnées de n'importe quel vecteur dans l'univers.
Plusieurs opérateur ont été ajouté pour faciliter la manipulation des vecteur tel que + ,* , -, …

- VECTEUR_H
- Vecteur<dim>
  - value : std::array<double, dim>
  - Vecteur(array<double, dim>)
  - Vecteur(const Vecteur<dim> &)
  - Vecteur()
  - ~Vecteur() : void
  - getValue() const : array<double, dim>
  - setValue(array<double, dim>) : void
  - dist(Vecteur<dim>) : double
  - sum(Vecteur<dim>) : Vecteur<dim>
  - diff(Vecteur<dim>) : Vecteur<dim>
  - multCoeff(double) : Vecteur<dim>
  - print() : void
  - operator=(const Vecteur<dim> &) : Vecteur<dim> &
  - Vecteur(Vecteur<dim> &&) noexcept
  - operator=(Vecteur<dim> &&) noexcept : Vecteur<dim> &
  - operator+(const Vecteur<dim> &) const : Vecteur<dim>
  - operator-(const Vecteur<dim> &) const : Vecteur<dim>
  - operator==(const Vecteur<dim> &) const : bool
  - operator*(double) const : Vecteur<dim>
- Vecteur<1>
- Vecteur<2>
- Vecteur<3>

Visualisation :



En guise de manque de temps , on a choisit que les particules apparaissent de l'autre sens de l'univers lorsqu'elles franchissent l'une des paroies (de manière random dans la première cellule de l'autre côté , l'univers est circulaire !)

ACVL

Diagramme des cas d'utilisation

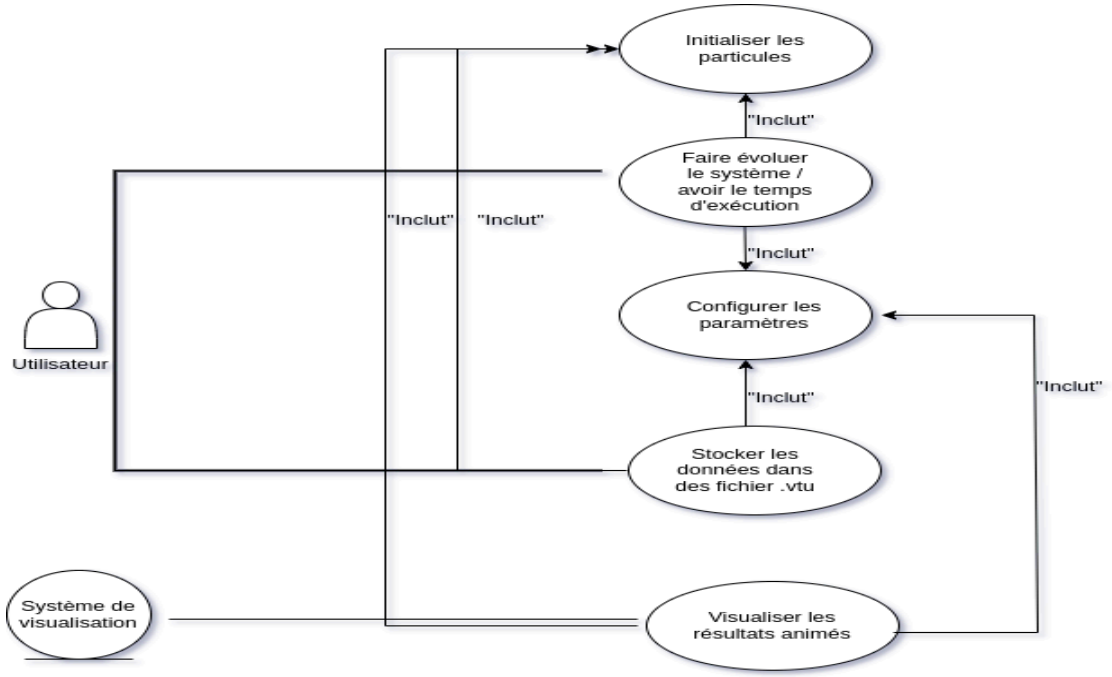
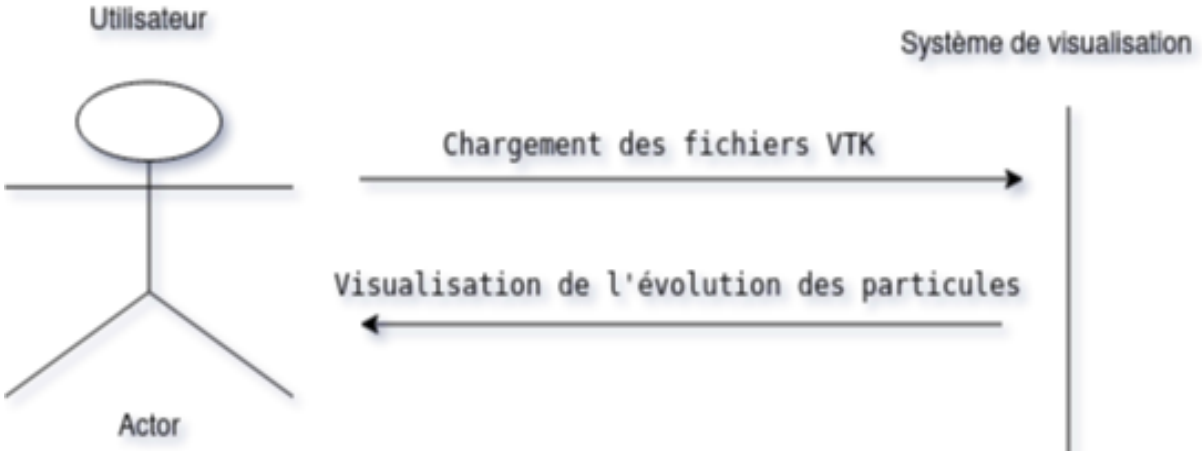


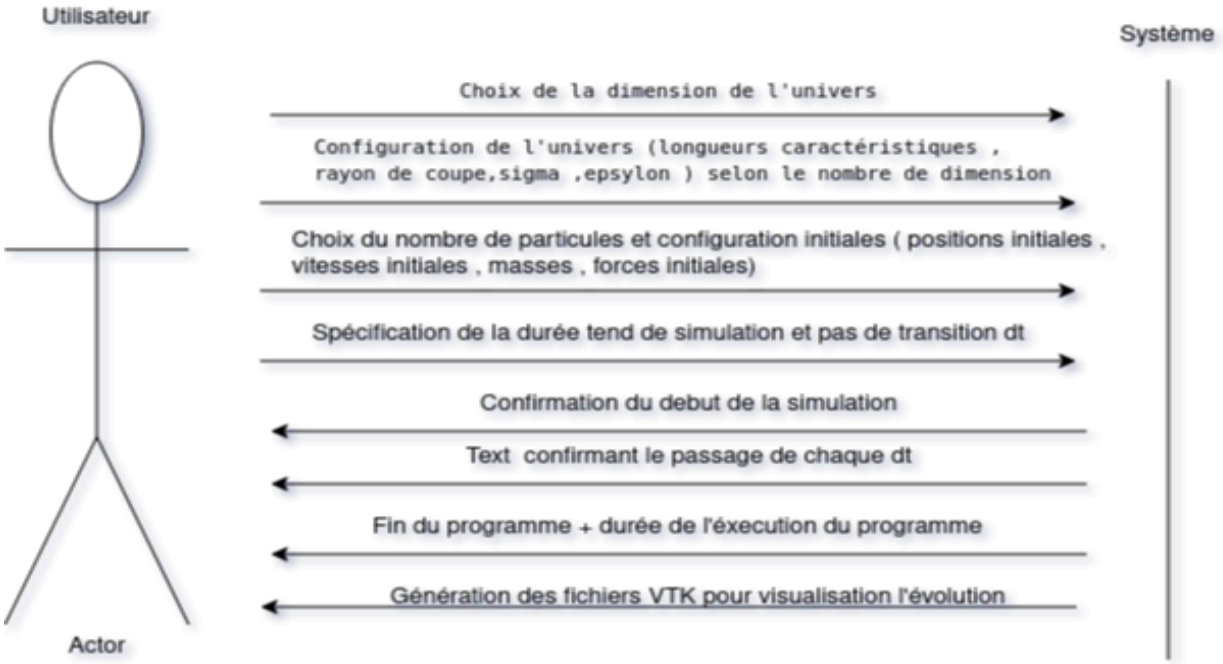
Diagramme de séquence:

Visualisation de l'évolution des particules dans l'espace et temps



Pré-Conditions :
-Bonne configuations des fichiers VTK

Configuration de la simulation



Pré-conditions :
- Le nombre de dimension doit appartenir à {1,2,3}
-Les caractérisques des particules et l'univers doivent être en cohérence avec la dimension de l'univers
- il faut que dt < tend

Diagramme de Transitions:

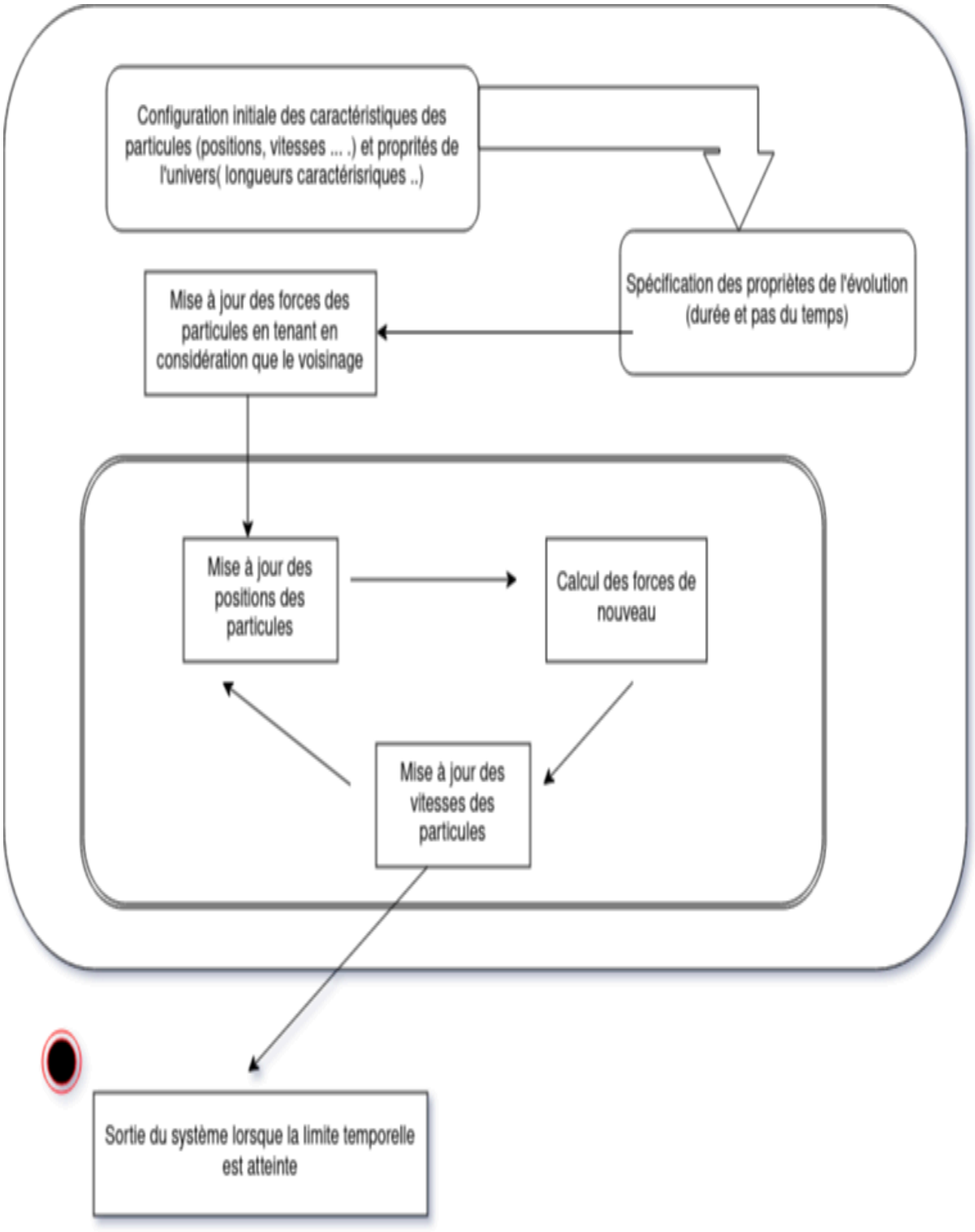


Diagramme de classes d'analyse: