Seveyrat Camille **L2 MIASHS**
Boccaccio Melissa **S3**

Compétence en Informatique

Niveau Expert Projet
Implémentation d'un Système Multi-Agents

Sommaire

1. Présentation du projet:

Pour ce projet nous avons décidé de faire notre système multi-agent sur la création des étoiles. Ce système multi-agent ne représente pas exactement comment les étoiles se créent dû à la complicité de la chose. Tout d'abord, nous allons expliquer comment les étoiles se forment dans la réalité puis faire le lien avec nos modifications et nos simplifications pour créer un programme le plus ressemblant.

Cette explication est importante car lors de la création du projet, la première chose que nous avons fait est de bien comprendre le phénomène d'évolution de la formation des étoiles grâce à de nombreuses recherches, cela facilitera aussi la compréhension.

Explication:

La nébuleuse (nuage de gaz et de poussière) va se fragmenter dû à la contraction gravitationnelle en plusieurs cœurs protostellaires. La phase protostellaire est un stade précoce dans le processus de formation d'une étoile. Il s'agit d'une région dense, qui, en accumulant de la matière, forme une proto-étoile, celle-ci étant principalement constituée d'Hydrogène et d'Hélium. Lorsque la proto-étoile atteint une certaine température et densité dans son noyau, la fusion d'atomes d'Hydrogène va créer un atome d'Hélium : c'est la fusion nucléaire. Pendant cette fusion, il y a une augmentation de la pression et une perte de masse ce qui crée une énergie thermique, et qui amène à que l'étoile s'allume.

Création des étoiles pour le projet:

Le panel représente une partie de nébuleuse. Dans le panel, au commencement on y retrouvera des atomes d'Hélium et des atomes d'Hydrogène, environ 25% d'atomes d'Hélium et 75% d'atomes d'Hydrogène. On initialisera sûrement aussi un certain nombre de protoétoiles directement dans le panel, au commencement, mais nous n'en sommes pas certaines à présent.

Les atomes d'Hydrogène et d'Hélium seront attirés entre eux grâce à la force d'attraction gravitationnelle que l'on aura configurée. Les atomes se rejoignant, un nuage dense se crée pour former, à un certain nombre d'atomes accumulés, une protoétoile. Nous n'avons pas créé d'agent "coeur protostellaire" pour simplifier notre système multi-agent. Nous pourrons potentiellement utiliser la formule de la masse critique de Jeans, masse à partir de laquelle commence l'effondrement de ce nuage dense pour former une proto-étoile.

Suite à cela, la protoétoile va attirer d'autres atomes toujours grâce à la force d'attraction gravitationnelle, et atteint un certain stade, la fusion nucléaire se produira. Pour cela, nous pouvons utiliser une formule reliant la température à la masse et la densité de la proto-étoile. La température atteindra un seuil critique, à partir duquel commencera la fusion. Nous gardons tout de même en option la création d'une simplification de ces phénomènes complexes, et donc de ces formules.

Au fur et à mesure du processus, nous allons générer des nouveaux atomes d'Hélium et d'Hydrogène pour que le système multi-agent fonctionne de manière constante. Par conséquent, il faudra donc un cycle de vie pour les étoiles : nous pouvons soit faire mourir deux étoiles lorsqu'elles se rencontrent (ont la même position), soit attribuer un cycle de vie à chaque étoile.

On utilise seulement 5 agents (dont un alien qui sera là pour le pathfinding) pour ce projet, il y en a beaucoup plus en réalité, mais ce sont les agents les plus importants pour réaliser et comprendre comment la création d'étoile se fait.

Grâce au système multi-agent, on va pouvoir voir comment se réalise la création de clusters. Nous testerons également plusieurs vitesses de simulation pour déterminer la plus performante. De plus, ne pouvant pas représenter le nombre d'atomes réellement présents dans une nébuleuse, nous leur attribuerons nous-même les masses et les volumes de nos agents.

Ce système sert donc à comprendre comment la formation d'étoiles se fait, comprendre quels sont les agents dans cette longue équation, quelles sont les relations (mathématiques) entre eux, tout cela de façon simplifiée !

2. Agents:

Listage des agents

Dans ce projet on retrouve 5 agents tous dotés d'un nom, d'une position (x et y), d'une vitesse, d'une accélération et d'une masse. Tous nos agents sont attirés entre eux par la force d'attraction gravitationnelle.

L'alien: L'alien ne nous sert que pour le pathfinding, seule sa position change au cours de la simulation.

L'Hélium: L'hélium a une vitesse, une accélération et une position qui changent constamment au cours de la simulation. Seule sa masse sera constante. Dans le panel, il se déplace donc grâce à la force d'attraction gravitationnelle par rapport à tous les autres agents. Il est représenté par un point de couleur cyan. Nous allons en générer au début de façon aléatoire (on verra le nombre au fur et à mesure pour que le système soit le plus représentatif et qu'il n'y ait pas une majoration de certains agents…), puis en régénérer suite à la création d'étoiles pour créer un cycle.

L'Hydrogène: L'hydrogène a en majorité les mêmes caractéristiques que l'Hélium. Il est lui représenté par un point de couleur bleu.

La protoétoile: La position de l'étoile, sa vitesse et son accélération ne seront pas constantes. En effet, la proto-étoile gagne de la masse grâce aux atomes qui s'ajoutent à elle. Son mouvement n'est aussi influencé que par la force d'attraction gravitationnelle. C'est la phase intermédiaire entre les atomes et l'étoile. Elles sont représentées en rouge dans le panel.

L'étoile: Les étoiles ne sont pas générées dès le début dans le panel, elles arriveront suite à la fusion nucléaire. La position de l'étoile, sa vitesse et son accélération ne seront pas constantes, tout comme la proto-étoile. Son mouvement n'est aussi influencé que par la force d'attraction gravitationnelle. Elle est représentée par un point jaune deux fois plus gros que la représentation graphique des atomes.

Formules mathématiques :

La première formule que nous avons utilisé dans notre programme est la force d'attraction gravitationnelle :

$$F = G \cdot \frac{m_1 \cdot m_2}{r^2}$$

- F est la force d'attraction gravitationnelle entre les deux objets (en newtons, N).
- G est la **constante gravitationnelle**, qui vaut environ $6{,}674 \times 10^{-11}\ N \cdot m^2.kg^{-2}$
- m_1 et m_2 sont les masses des deux objets (en kilogrammes, kg).
- r est la distance entre les centres des deux objets (en mètres, m).

Après cela, nous avons déterminé l'accélération de chaque particule :

$$a_1 = G \cdot \frac{m_2}{r^2}$$

- a_1 est la particule de masse m_1, qui interagit avec a_2, qui est la particule de masse m_2

Par la suite, nous avons utilisé les équations de la vitesse et de la position correspondantes :

$$v(t) = v_0 + a \cdot t$$

- v(t) est la vitesse de la particule à un instant t (en m/s).
- v_0 est la vitesse initiale de la particule (en m/s).
- a est l'accélération de la particule (en m/s^2).
- t est le temps écoulé (en secondes, s).

$$x(t) = x_0 + v_0 \cdot t + \frac{1}{2} a \cdot t^2$$

- x(t) est la position de la particule à un instant t (en m/s).
- v_0 est la vitesse initiale de la particule (en m/s).
- a est l'accélération de la particule (en m/s^2).
- t est le temps écoulé (en secondes, s).

Ensuite, pour déterminer le moment où le nuage dense s'effondrera sous sa propre gravité pour former une proto-étoile, nous utiliserons la formule de la masse de Jeans. Cela correspond à la masse critique à partir de laquelle le nuage commence à s'effondrer :

$$M_J = \frac{5k_BT}{G\mu m_H} \cdot \left(\frac{3}{4\pi\rho}\right)^{1/2}$$

- M_J est la masse de Jeans (en kg)
- μ est la masse moléculaire du gaz (environ 2.3)
- k_B est la constante de Boltzmann (1,38.10^{-23} J/K)
- T est la température d'un nuage de gaz (en K) (entre 10 K et 20 K)
- G est la constante gravitationnelle (6,674×10^{-11} $N \cdot m^2.kg^{-2}$)
- ρ est la densité (en kg/m^3)

Cela donne :

$$M_J = \frac{5 \times (1.38 \times 10^{-23}) \times T}{(6.674 \times 10^{-11}) \times (2.3 \times 1.67 \times 10^{-27})} \cdot \left(\frac{3}{4\pi\rho}\right)^{1/2}$$

Formules de la densité et du volume (nous considérons que tous nos objets sont sphériques)

$$\rho = \frac{M}{V} \qquad V = \frac{4}{3}\pi R^3$$

- ρ est la densité (en kg/m^3)
- M est la masse de la région concernée (en kg)
- V est le volume de cette région en m^3
- R est le rayon (en m)

Pour finir, voici une formule reliant la température à la densité (lorsque la température atteint 107 K, la fusion nucléaire s'opère)

$$T = \frac{G \cdot M \cdot \rho}{k_B \cdot R}$$

- T est la température de la proto-étoile (en K)
- G est la constante gravitationnelle (6,674×10^{-11} $N \cdot m^2.kg^{-2}$)
- M est la masse de la proto-étoile (en kg)
- ρ est la densité (en kg/m^3)
- k_B est la constante de Boltzmann (1,38.10^{-23} J/K)
- R est le rayon (en m)

3. Présentation du projet:

Le programme

Le programme est en java avec l'utilisation de l'IDE Intellij, on utilise A* pour le pathfinding de l'alien, on utilise des threads et swing pour le panel.

Java est un langage de programmation orienté objet, c'est-à-dire qu'il est un modèle de programmation qui repose sur le concept de classes et d'objets. Pour ce projet on à plusieurs classes: des classes contenant les agents, des classes où les formules sont définies, des classes définissant les agents et leurs position, la classe contenant le panel, une autre pour le pathfinding et une pour la fonction main. Toutes ces classes sont dans un package: `package fr.um3.projet;`

Classe Agents:
On instancie, dans un constructeur, un nom et une position. On y retrouve également une fonction run() et stop(). Les agents ont hérité de cette classe.

Position:
On crée une position x et une autre y, car le panel est en 2D.

Les Agents:
Tous les agents sont définis par un nom, une position, une vitesse, une accélération et une masse.

Mass :
La masse est définie par un nombre de type double.

Velocity :
Tout comme la position, la vitesse est définie par un x et un y.

Accélération :
La classe accélération est également définie par un x et un y, et elle possède une méthode qui calcule l'accélération d'un agent par rapport à une liste d'autres agents.

Le panel:
Pour l'instant, on a généré tous les agents au commencement, on changera cela dans les semaines qui suivent pour les générer successivement et recréer la formation des étoiles. On utilise une méthode generate() qu'on instancie dans le constructeur du panel. Pour la création du panel il faut importer plusieurs éléments tels que java.swing.*, java.util.ArrayList;...

```
for (int i = 0; i < 50; i++) {
   int x = (int) (Math.random() * 800); // Génère des positions
aléatoires pour X
   int y = (int) (Math.random() * 650); // Génère des positions
aléatoires pour Y
   Position position = new Position(x, y); // Crée une nouvelle
position
   hydrogens.add(new Hydrogen(position))
```

Pour la représentation graphique on crée juste des ronds de couleur différentes pour chaque agent, voici un exemple.

```
@Override
protected void paintComponent(Graphics g) {
   super.paintComponent(g);

   g.setColor(Color.CYAN); //création d'un rond
   for (Hydrogen hydrogen : hydrogens) {
       int x = hydrogen.getPosition().getX();
       int y = hydrogen.getPosition().getY();

       g.fillOval(x, y, 10, 10);
   }
```

Dans le panel c'est également là qu'on retrouve une partie du pathfinding pour qu'il se déplace dans le panel avec les autres agents et non dans une autre interface graphique. On initialise un timer, une position de départ et d'arrivée pour l'alien. On utilise des threads pour cela.

Le pathfinding (recherche de chemin):

On choisit de prendre un alien qui se déplace dans un environnement en deux dimensions, le panel. L'objectif est de trouver un chemin optimal entre un point de départ et un point d'arrivée tout en évitant des obstacles (étoiles et proto-étoiles). Le programme utilise l'algorithme A* pour cette recherche de chemin. Ce chemin est calculé en tenant compte de la position de départ, de la position cible, et des obstacles. Il fonctionne en explorant les voisins d'un nœud donné et en choisissant le chemin avec le coût total (g + h) le plus faible. L'algorithme continue jusqu'à ce qu'il trouve un chemin vers la cible ou qu'il n'y ait plus de nœuds à explorer.

Composantes principales de A* :

Node : Représente chaque position explorée lors de la recherche du chemin, en conservant des informations telles que le coût du chemin (g), l'estimation vers la cible (h) et la somme de ces deux valeurs (f).
g(n) : Coût pour aller du nœud de départ jusqu'au nœud n.
h(n) : Heuristique, estimation du coût pour aller de n jusqu'à la cible.
f(n) : Somme des deux valeurs g(n) + h(n), qui guide l'algorithme dans son exploration.

Le processus de pathfinding démarre en exécutant la méthode findPath(). Cette méthode commence avec le point de départ qui est ajouté à la liste ouverte (openList) qui contient les nœuds à explorer. Une liste fermée (closedList) est utilisée pour suivre les nœuds déjà explorés. Pour chaque nœud exploré, on examine les positions voisines (haut, bas, gauche, droite) et on calcule le coût du chemin vers chaque voisin (g). Si le voisin n'est pas un obstacle et qu'il n'a pas déjà été exploré, il est ajouté à la liste des nœuds à explorer. À chaque nouvelle position, la méthode willCollide() est appelée pour vérifier si la position est bloquée par une étoile ou une proto-étoile. Si c'est le cas, cette position est ignorée. Une fois que le chemin est trouvé, il est affiché graphiquement dans le Panel grâce à repaint(), qui redessine l'alien se déplaçant pas à pas vers la cible. Si aucun chemin n'est trouvé, un message est affiché.

Main:

Pour l'instant dans la classe main on a initialisé les threads.

4. Repartition du projet:

Camille:

2ème semaine:
- choix du projet

3ème semaine:
- recherche sur la création des étoiles
- commencement du code en créant des classes pour chaque agent
- diagramme de gantt

4ème semaine:
- recherche des formules mathématiques

5ème semaine:
- commencement de la création du panel avec swing
- Agents représentés graphiquement

6ème semaine:
- premières interactions entre les agents (force gravitationnelle dans le code)

7ème semaine:
- 1er compte rendu

8ème semaine:
- aide au pathfinding
- transformation des atomes en protoétoile

Melissa:

2ème semaine:
- choix du projet

3ème semaine:
- agent et descriptif bien complet
- compréhension de la création d'étoile

4ème semaine:
- commencement du code des classes
- recherche et utilisations des threads

En Java, un thread est une unité d'exécution qui permet d'exécuter du code de manière concurrente. En d'autres termes, les threads permettent d'exécuter plusieurs tâches en parallèle dans un programme. Ils sont essentiels pour améliorer les performances dans certaines situations, notamment lorsque des tâches lourdes ou indépendantes doivent être exécutées simultanément.

5ème semaine:
- répondre à la question de pourquoi les threads s'exécute en "random"

L'ordre d'exécution des threads dans un programme Java peut sembler aléatoire (random) à cause de la façon dont le système d'exploitation et la JVM gèrent l'ordonnancement des threads. Les threads peuvent être exécutés en fonction de leur état, de la charge du système et de l'algorithme d'ordonnancement utilisé. Cela permet une exécution concurrente, où plusieurs tâches peuvent progresser en parallèle, mais sans garantie d'ordre d'exécution.
- commencement du pathfinding

6ème semaine:

- pathfinding

7ème semaine:

- 1er compte rendu

8ème semaine:

- pathfinding
- approfondissement des threads dans le code