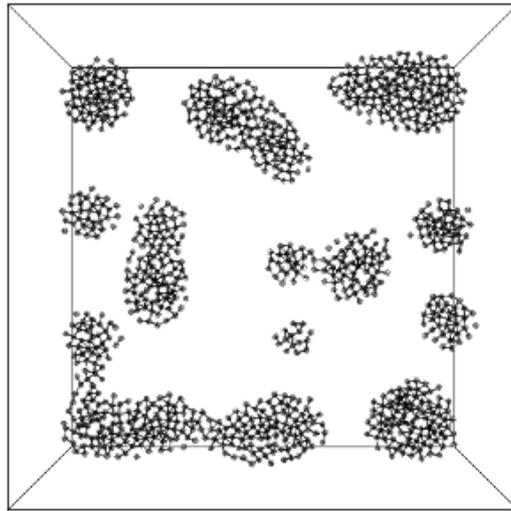


Caractéristiques techniques du projet Iquisme



Nicolas LIAUTAUD, Lucas CURCI

Table des matières

1	Organisation	3
1.1	Conventions et logiciels	3
1.2	Structure de fichiers	4
1.3	Étapes de développement	4
2	Modélisation du code	5
2.1	Description des classes	5
2.1.1	Sketch	5
2.1.2	World	5
2.1.3	Scenario	5
2.1.4	Particle	5
2.1.5	Behaviour	6
2.1.6	Brain	6
2.1.7	Memory	6
2.2	Modélisation	6
3	Algorithmes	7
3.1	Système d'animation	7
3.1.1	Rassemblement	7
3.1.2	Imitation	7
3.1.3	Répulsion	8
3.1.4	Résultante et viscosité	8
3.2	Champ de force du monde	8
3.3	Flou de profondeur	9
4	Prototype	10
4.1	Navigation	10
4.1.1	Manipulation de la caméra	11
4.1.2	Fenêtre de contrôles	11

1 Organisation

1.1 Conventions et logiciels

Sont décrits ici les conventions et logiciels retenus pour les travaux de pré-développement et de développement. Il a été choisi de travailler entièrement au moyen de logiciels et de langages de programmation *open-source*, sous environnement GNU/Linux¹.

Langue de développement : Anglais

Langage de programmation : Java²

Bibliothèques notables : Processing³, OpenGL⁴

Format : application/java-archive (.jar)

Système d'exploitation : multiplateformes (Java Virtual Machine⁵)

Conventions : conventions d'écriture Sun⁶

Documentation : JavaDoc⁷

Interface graphique : javax swing⁸

Gestion de projet : Planner⁹

Modélisation du code : Umbrello¹⁰

Environnement de développement : Eclipse¹¹

Gestion de versions : Subversion¹²

1. <http://ubuntu.com>
2. <http://java.com>
3. <http://processing.org>
4. <http://opengl.org>
5. <http://java.sun.com/docs/books/jvms>
6. <http://java.sun.com/docs/codeconv/CodeConventions.pdf>
7. <http://download.oracle.com/javase/1.5.0/docs/tooldocs/solaris/javadoc.html>
8. <http://download.oracle.com/javase/1.4.2/docs/api/javax/swing/package-summary.html>
9. <http://live.gnome.org/planner>
10. <http://uml.sourceforge.net>
11. <http://eclipse.org>
12. <http://subversion.apache.org>

1.2 Structure de fichiers

Cette section décrit l'arborescence de fichiers utilisée pour le développement du projet. Seuls les répertoires du dépôt subversion sont indiqués ici.

utils : rassemble les documents facilitant le travail de développement.

utils/uml : réunit les fichiers relatifs à la modélisation du code.

utils/scripts : contient les scripts spécialement développés pour faciliter la réalisation du projet.

utils/docs : regroupe des documents de référence pour le développement ainsi que les descriptifs du projet dans son ensemble.

utils/cal : réunit les fichiers relatifs au calendrier de développement.

trunk : ce répertoire de travail contient le programme en cours d'écriture.

trunk/src : sources du programme.

trunk/libs : bibliothèques nécessaires au fonctionnement du programme.

tags : réunit les différentes versions stables du programme (*releases*).

report : contient les journaux de bord et comptes-rendus de développement.

1.3 Étapes de développement

Le calendrier de développement est découpé en différentes étapes de travail correspondant au déploiement progressif de nouvelles fonctionnalités. Chaque étape est suivie d'une phase de recherche et suppression des erreurs ou des incohérences amenant à une version partielle mais fonctionnelle du programme.

1. Monde et particules
 - (a) Monde, particules et système d'animation
 - (b) Interactions sociales et comportements
 - (c) Mémoire et analyse
 - (d) Réflexion et réseaux neuronaux
2. Audio
 - (a) Flux de données
 - (b) Analyse du son
 - (c) Effets sur la simulation
3. Interface graphique
 - (a) Gestion de la simulation
 - (b) Gestion des comportements
 - (c) Gestion du scénario

2 Modélisation du code

Le programme, à l'image du langage Java, est orienté objet afin de respecter facilement la structuration du développement. Cette section décrit les classes importantes du programme et présente leur modélisation selon le standard UML.

2.1 Description des classes

Cette partie synthétise le fonctionnement et l'utilité des classes principales, ainsi que leurs relations. D'autres classes gravitent autour de cette structure principale (types de données, classes purement utilitaires, systèmes en projet, etc...) et ne sont pas citées ici.

2.1.1 Sketch

Le sketch est la classe principale, contenant tout le programme. Elle crée le monde, le scénario et l'interface utilisateur, règle les paramètres d'exécution de Processing puis affiche et met à jour la simulation à chaque instant.

2.1.2 World

Le monde est un objet régissant des instances de Particles, et est représenté par un parallélépipède rectangle contenant physiquement les dites particules. Cette classe simule un environnement physique avec sa taille, ses forces, sa viscosité...

2.1.3 Scenario

Le scénario est un objet régissant Behavior, donc le comportement général des particules, au fil du temps. Il contient un ensemble de tableaux relatifs aux actes de la composition musicale.

2.1.4 Particle

Une particule est un objet possédant un comportement (Behavior) et un cerveau (Brain), et définie par une position, une vitesse, une distance de contact, de perception, etc... ainsi qu'un ensemble de caractéristiques (agressivité, stress, énergie...) représentant son état. Toutes les particules suivent un comportement général défini par le scénario, puis chacune répond aux situations ponctuelles selon le résultat de l'analyse faite par Brain.

2.1.5 Behaviour

Le comportement des particules devant changer au fil du scénario, il a été décidé de définir ces différents comportements en dehors des particules elles-même, en tant qu'implémentation d'une interface Behaviour. Il est ainsi possible de créer n'importe quel nombre de comportements, et de les utiliser de manière similaire.

2.1.6 Brain

Le cerveau est un objet qui analyse des situations actuelles et passées et, en conjonction avec le comportement et l'état d'une particule, prend des décisions. Il contient une instance de Memory.

2.1.7 Memory

La mémoire est un objet stockant les situations importantes dans lesquelles se trouve la particule, les réactions de la particule face à ces situations et les résultats de ces décisions. L'analyse de ces données est effectuée par Brain.

2.2 Modélisation

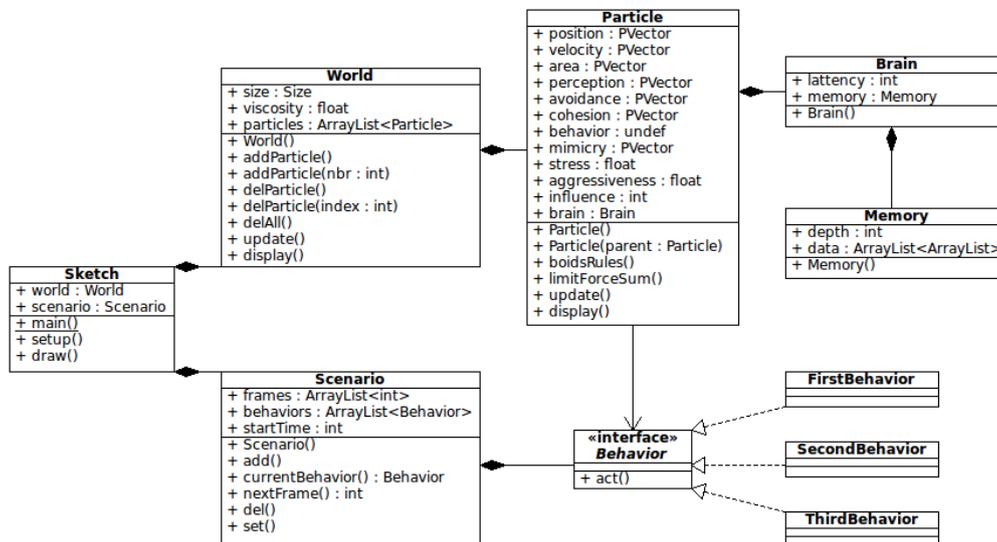


FIGURE 1 – Diagramme des classes principales

Ce diagramme ne mentionne pas tous les attributs et les méthodes des classes (getters/setters, méthodes purement utilitaires, etc...) et toutes les classes (types de données, en projet, etc...) afin de faciliter la vue d'ensemble de la structure principale.

3 Algorithmes

Cette section décrit quelques-uns des algorithmes et systèmes utilisés.

3.1 Système d'animation

L'animation générale des particules vis-à-vis des autres est définie par trois règles pouvant être modulées selon le scénario ou en réponse aux situations locales.

3.1.1 Rassemblement

Chaque particule essaie de se rapprocher du centre du groupe de particules perçues.

Le centre du groupe de N particules perçues de positions \vec{p}_n est :

$$\vec{b} = \frac{S_n}{N}; S_n = \sum_{i=0}^N \vec{p}_n$$

La force \vec{f}_1 dirigeant une particule de position \vec{p} vers ce barycentre est le vecteur allant de \vec{p} vers \vec{b} , et modulé par une variable de cohésion c :

$$\vec{f}_1 = (\vec{b} - \vec{p}) * c$$

3.1.2 Imitation

Chaque particule essaie d'aller dans la direction moyenne des particules perçues.

La moyenne des directions de N particules perçues de vitesse \vec{v}_n est :

$$\vec{d} = \frac{S_n}{N}; S_n = \sum_{i=0}^N \vec{v}_n$$

La force \vec{f}_2 transformant la vitesse \vec{v} d'une particule en \vec{d} est le vecteur allant de \vec{v} vers \vec{d} , modulé par une variable de mimétisme m :

$$\vec{f}_2 = (\vec{d} - \vec{v}) * m$$

3.1.3 Répulsion

Chaque particule essaie de s'éloigner des particules trop proches.

La force \vec{r} éloignant une particule de positions \vec{p}_1 d'une autre de position \vec{p}_2 est un vecteur dirigé de \vec{p}_2 vers \vec{p}_1 et d'intensité relative à la distance entre les deux particules d :

$$\vec{r} = \frac{\vec{p}_2 - \vec{p}_1}{\|\vec{p}_2 - \vec{p}_1\|} * d^2$$

La normalisation du résultat permet de n'en conserver que la direction et le sens, et de faire varier sa longueur selon la distance. Ainsi plus deux particules sont proches, plus la force sera forte. L'utilisation de d^2 amène à un regroupement des particules en amas nerveux (type amas cellulaire), une utilisation de d seul amène à un regroupement moins systématique et plus fluide (type banc de poisson).

La force \vec{f}_3 éloignant une particule de toutes les N particules touchées est une moyenne des forces \vec{r} modulée par une variable d'évitement a :

$$\vec{f}_3 = \frac{S_n}{N} * a; S_n = \sum_{i=0}^N \vec{r}_n$$

3.1.4 Résultante et viscosité

La réaction de chaque particule vis-à-vis de ces forces est une force résultante \vec{F} (pouvant être modulée) ajoutée à sa vitesse actuelle \vec{v} :

$$\vec{v} = \vec{v} + f_1 + f_2 + f_3$$

La vitesse finale de la particule est alors affectée par la viscosité du monde μ , avant d'être ajoutée à sa position \vec{p} .

$$\vec{p} = \vec{p} + (\vec{v} * \mu)$$

3.2 Champ de force du monde

Pour éviter que les particules ne sortent du monde ce dernier leur applique une force vers l'intérieur de plus en plus importante à mesure qu'elles s'approchent des bords ou en sortent. La force appliquée lorsqu'une particule est à l'intérieur du monde tend fortement vers zéro et n'a donc aucune influence, mais augmente plus ou moins rapidement lorsqu'elle atteint les bords de la zone donnée.

La formule utilisée pour définir la force que le monde applique à la direction d'une particule de position \vec{p} et de vitesse \vec{v} est :

$$\vec{v} = \vec{v} - \left(\frac{\vec{p}}{e}\right)^c$$

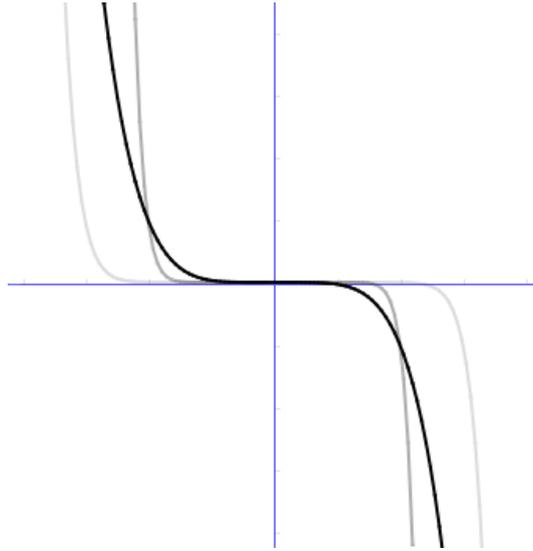


FIGURE 2 – Courbe utilisée pour le champ de force du monde. Il est possible d'en modifier la courbure au moyen de c (cf. courbe gris-foncé) et l'écartement vis-à-vis de l'origine au moyen de e (cf. courbe gris-clair).

La valeur e est définie selon la taille du monde, le champ de force s'applique ainsi à l'approche des bords de ce dernier.

La valeur c permet de définir la vitesse du changement de direction de la particule : une faible valeur affecte lentement sa direction, une valeur importante la modifie rapidement à l'image d'un rebond sur une frontière physique.

3.3 Flou de profondeur

Le flou de profondeur est simulé en modifiant l'épaisseur et l'opacité des particules selon leur distance au plan focal. Plus une particule est éloignée du plan focal, plus son opacité diminue et plus son épaisseur augmente, donnant une impression de flou. A l'inverse, plus elle est proche du plan focal plus son épaisseur diminue et plus elle s'opacifie, donnant une impression de netteté.

Sachant la position \vec{p} d'une particule, la position \vec{c} de la caméra et la position \vec{t} du point visé par la caméra, la normale \vec{n} du plan focal de la caméra est :

$$\vec{n} = \vec{t} - \vec{c}$$

La distance de la caméra au point visé est :

$$z = \sqrt{(c_x - t_x)^2 + (c_y - t_y)^2 + (c_z - t_z)^2}$$

Et la distance de la particule au plan focal est :

$$d = \frac{|n_x * p_x + n_y * p_y + n_z * p_z + z|}{\sqrt{n_x^2 + n_y^2 + n_z^2}}$$

L'intensité du flou de profondeur est la proportion de cette distance d par rapport à la distance z de la caméra au point visé. Cette manipulation permet de rendre l'effet de la focale (zoom) sur la plage de netteté :

$$dof = \frac{d * 100}{z}$$

L'épaisseur de la particule est alors directement tirée de ce résultat, son opacité est quant à elle contrainte dans l'intervalle [50, 255] afin d'en tirer une valeur alpha valable sans jamais être entièrement transparente.

4 Prototype

Le prototype est une application de présentation du projet développée avec les mêmes langages, logiciels et structure de classes préalablement définis visant à mettre en pratique et illustrer certains principes, algorithmes et recherches esthétiques. Il n'est en aucun cas une représentation définitive de l'aspect visuel et fonctionnel de l'application. Les fonctionnements sociaux, énergétiques, d'analyse ou de réflexion ne sont ici que très légèrement approchés, "simulés" afin de donner à voir un ersatz de leur résultat par des moyens détournés, ou ne sont pas implémentés.

4.1 Navigation

Le prototype est divisé en deux parties : une fenêtre de visualisation et une fenêtre de contrôles. La première permet d'observer la simulation depuis l'angle voulu, la seconde permet d'en modifier les paramètres afin d'en approcher les divers aspects.

4.1.1 Manipulation de la caméra

Lors du lancement de l'application, la caméra est située au-dessus du monde, représenté par un parallépipède rectangle gris-foncé. Elle peut être manipulée au moyen de la souris :

Tourner autour du point observé : clic gauche.

Zoomer : clic droit ou défilement molette.

Déplacer la caméra : clic droit et gauche ou clic molette.

Replacer la caméra à son état initial : double clic.

4.1.2 Fenêtre de contrôles

La première partie de la fenêtre permet de visualiser le nombre d'images affichées par seconde, de choisir un réglage pré-enregistré et de relancer la simulation. Le premier réglage de la liste (*mitosis*) illustre la mitose, où chaque particule se divise régulièrement. Le deuxième (*soup*) simule une anarchie, une soupe dans laquelle les particules ne se stabilisent pas. Dans le troisième (*groups*) les particules se regroupent en amas et dans le quatrième (*pyramids*) s'organisent en structures sociales.

World settings Permet de régler un certain nombre de paramètres du monde. Il est nécessaire de valider avec la touche *Entrée* ou relancer la simulation au moyen du bouton *Reset* pour que les changements soient effectifs. Une modification du nombre de particules relance nécessairement la simulation.

Start number : nombre de particules au lancement de la simulation.

Max number : nombre maximum de particules pouvant naître.

Viscosity : facteur de viscosité du monde.

Field slope : forme de la courbe du champ de force du monde. Avec une grande valeur la barrière est nette et franche, avec une faible valeur elle est étendue et progressive.

Particles settings Rassemble un certain nombre de paramètres des particules. Il est nécessaire de valider avec la touche *Entrée* pour que les changements soient effectifs.

Area : distance de contact.

Perception : distance de perception.

Cohesion : facteur de cohésion.

Avoidance : facteur d'évitement.

Mimicry : facteur d'imitation.

Velocity limit : vitesse maximale.

Simulation options Concentre diverses options permettant d'activer ou non certaines procédures ou de définir certains comportements de la simulation.

Mitosis : activer la mitose, division cellulaire.

Birth on plane : les particules naissent sur un plan. Dans le cas contraire les particules naissent dans l'espace 3D.

Rand birth position : les particules naissent n'importe où dans l'espace. Dans le cas contraire elles naissent au centre du monde (début de la simulation) ou à l'emplacement de leur parent (mitose).

Manage energy : activer la gestion de l'énergie. Les particules puisent dans leur énergie pour se déplacer, énergie qui se remplit lorsqu'elles sont immobiles.

Repulsion on d^2 : utiliser la répulsion selon la distance au carré. Dans le cas contraire utilise une répulsion selon la distance.

Pyramidal hierarchy : active la structuration pyramidale des groupes sociaux.

Particles color Permet de choisir les couleurs des particules, afin d'observer plus facilement certains fonctionnements.

default : couleur par défaut.

groups : A chaque groupe social est attribuée une couleur différente et les particules sont colorisées selon leur groupe d'appartenance, ce qui permet d'observer la distribution et l'agrégation des différents groupes.

energy : les particules sont colorisées selon leur énergie, de rouge (beaucoup) à blanc (peu), ce qui permet d'en observer les flux.

Particles model Permet de choisir le type d'affichage des particules.

dots : un simple affichage de points, le plus vélocé à calculer.

ellipses : affichage de pseudo-sphères composées de trois ellipses entrecroisées.

cells : affiche des disques transparents (étudié pour une utilisation conjointe avec le flou de profondeur).

boxes : affiche des cubes.

- Display options** Propose des options d'affichage supplémentaires.
- Depth of field** : activer le flou de profondeur.
- Particles links** : activer l'affichage des liens de contact reliant les particules entre elles.
- Cursor** : afficher ou cacher le curseur.