

Simulation d'un écosystème

Sommaire

1	Description du problème initial	1
2	Description des évolutions souhaitées	2
3	Interface Homme Machine	4
4	Analyse des résultats de simulation	4
5	Livrables & déroulement	5

Introduction

L'objectif de ce BE est double. Il s'agit dans un premier temps d'étudier un problème et de formaliser une réponse à ce problème au travers de l'utilisation de *design patterns* appropriés. Dans un second temps, le travail consistera à proposer une implémentation de ces patterns pour aboutir à un logiciel fonctionnel.

Le cadre de l'exercice est un problème de *refactoring* : partant d'une architecture simple (et du code correspondant), on souhaite *faire évoluer le modèle objet* pour répondre proprement aux spécifications décrites ci-dessous puis les mettre en œuvre.

Ainsi, à partir du besoin exprimé dans ce document, dans une première partie de l'exercice, on réalisera une **analyse objet du problème** qui débouchera au minimum, sur un diagramme de classes UML. L'objectif de l'exercice est de s'appuyer explicitement sur des design patterns existants pour répondre à ce problème spécifique. Un ou plusieurs diagrammes de séquence complèteront très avantageusement cette description statique.

La seconde partie de l'exercice consistera à **comprendre, compiler et exécuter le code fourni**, avant de le modifier et/ou le faire évoluer pour *implémenter le nouveau diagramme objet*. En effet, le code fourni fonctionne mais ne correspond qu'à la description du problème initial.

1 Description du problème initial

On s'intéresse ici au problème de la *simulation d'un écosystème*¹.

La simulation porte sur un système composé de bestioles. A chaque instant, chaque bestiole se trouve en une position particulière et se déplace avec une certaine vitesse dans une direction donnée. C'est d'ailleurs la seule action que les bestioles sont capables de réaliser pour l'instant. Quelle que soit la représentation graphique d'une bestiole, l'occupation spatiale (ou taille) de chaque bestiole est modélisée par une ellipse dont l'axe principal est orienté selon le déplacement de la bestiole. A la création d'une bestiole, sa position, sa taille, sa vitesse et sa direction sont initialisées aléatoirement.

Les bestioles sont placées dans un aquarium rempli d'une matière (par exemple de l'eau) qui constitue leur milieu de vie. Ainsi l'aquarium contient ce milieu de vie, qui lui-même contient les bestioles. C'est

1. Exercice inspiré des supports de M. Philippe Renevier-Gonin, disponibles en ligne à l'adresse <http://deptinfo.unice.fr/~renevier/>, dernier accès le 03/01/2012.

d'ailleurs au travers de ce milieu que les bestioles peuvent faire tout ce qu'elles ont à faire (c'est à dire bouger pour l'instant !) au cours du temps. Les bestioles perçoivent également leur environnement, c'est-à-dire le milieu de vie et notamment ses limites, mais aussi les bestioles voisines. On considère que les bestioles rebondissent sans dégât en cas de collision avec les frontières du milieu de vie.

D'un point de vue technique, on considérera que l'aquarium aura la charge de coordonner la simulation au cours du temps.

2 Description des évolutions souhaitées

2.1 Naissance des bestioles

On souhaite centraliser la création des bestioles tout en prévoyant la possibilité de générer par la suite de nouvelles espèces² de bestioles.

La création de la population initiale de bestioles se fait au travers de configurations particulières (potentiellement plusieurs) qui spécifient le pourcentage de bestioles possédant un comportement dynamique particulier (par exemple, 20% de bestioles **grégaire**s et 80% de bestioles **peureuses**). On rappelle que même si le modèle de simulation permet de gérer différentes configurations de population, une instance de simulation disposera d'une unique ressource de création de bestioles, partagée entre les différentes entités impliquées dans la simulation.

Par ailleurs, en cours de simulation, de nouvelles bestioles peuvent naître spontanément (aléatoirement) avec un **taux de naissance prédéfini**, mais conformément à la **répartition** induite par la configuration initiale.

Les bestioles peuvent également naître sous l'action d'un **événement extérieur**.

2.2 Mort des bestioles

Chaque bestiole a un **age limite**, déterminé aléatoirement à la naissance. Lorsque la bestiole atteint cet age au cours de la simulation, elle meurt et disparaît du milieu de vie.

A tout moment, deux ou plusieurs bestioles peuvent entrer en collision. Pour chaque bestiole impliquée, soit elle rebondit et **repart** dans la direction opposée à sa direction de déplacement d'avant la collision, soit elle **meurt** avec une certaine probabilité.

Enfin, une bestiole peut être achevée sur un **événement extérieur**.

2.3 Clonage d'une bestiole

Les bestioles sont des créatures qui disposent de la capacité à augmenter spontanément leur population par auto-clonage (une bestiole se divise parfois en deux bestioles totalement identiques). Cet événement n'est pas systématique, mais a une certaine **probabilité** d'arriver à tout instant de la vie des bestioles.

2.4 Capteurs & Accessoires

Afin de pouvoir observer leur environnement et/ou interagir avec celui-ci, les bestioles peuvent être munis de **capteurs** (yeux, oreilles) et d'**accessoires** supplémentaires (nageoires, carapace et camouflage). Le fonctionnement de chacun de ces capteurs ou accessoires est expliqué ci-dessous.

Attention, chaque bestiole peut être équipée de **zéro à plusieurs** capteurs différents et de zéro ou plusieurs accessoires différents. Ainsi, dans une simulation, des bestioles avec des yeux et un camouflage peuvent **côtoyer** des bestioles avec des yeux et des oreilles mais pas d'accessoire et d'autres bestioles qui par exemple, ne possèdent ni capteur, ni accessoire, ...

2. La notion d'espèces n'a rien à voir avec la notion de comportement dynamique qui sera définie ensuite

2.4.1 Description des capteurs

Les capteurs permettent aux bestioles qui en sont dotés, de percevoir les autres bestioles de la simulation en cours avec des restrictions propres à chaque capteur, comme illustré en Figure 1 :

- **Yeux** : Les yeux permettent à la bestiole de voir en face d'elle avec un champ angulaire de vision $\alpha \in [\alpha^{\min}; \alpha^{\max}]$ sur une distance $\delta_y \in [\delta_y^{\min}; \delta_y^{\max}]$. La possibilité pour la bestiole équipée de ces yeux, de détecter la présence d'une autre bestiole dans son champ de vision, est modélisée par la capacité de détection $\gamma_y \in [\gamma_y^{\min}; \gamma_y^{\max}]$ avec $0 \leq \gamma_y^{\min} \leq \gamma_y^{\max} \leq 1$.
- **Oreilles** : Les oreilles permettent à la bestiole d'entendre autour d'elle sur une plage de distance $\delta_o \in [\delta_o^{\min}; \delta_o^{\max}]$. Le champ de perception dans ce cas peut être considéré comme constant et égal à $[0; 2\pi]$. La possibilité pour la bestiole équipée de ces oreilles, de détecter la présence d'une autre bestiole, est aussi modélisée par la capacité de détection $\gamma_o \in [\gamma_o^{\min}; \gamma_o^{\max}]$ avec $0 \leq \gamma_o^{\min} \leq \gamma_o^{\max} \leq 1$.

Plus la capacité de détection γ d'un capteur est proche de zéro, moins ce capteur sera performant pour détecter les autres bestioles (jusqu'à ne rien détecter quand la valeur est égale à 0). A l'inverse, lorsque la capacité de détection γ est égale à 1, le capteur est efficace en détection à 100%.

Les valeurs définissant les limites de perception et de détection seront paramétrables à travers la configuration initiale de l'aquarium. Par conséquent, toutes les bestioles d'une simulation partagent ces mêmes valeurs *limite*. En revanche, les capteurs de chaque bestiole ont des caractéristiques propres.



Figure 1 – Exemples de capteurs pouvant équiper une bestiole (avec leurs caractéristiques géométriques).

2.4.2 Description des accessoires

Les accessoires permettent aux bestioles qui en sont dotés, d'affecter les valeurs de certains paramètres de leurs actions ou

- **Nageoires** : Les nageoires permettent à la bestiole équipée, de se déplacer plus vite en appliquant un coefficient multiplicateur $\nu \in [1; \nu^{\max}]$ à la vitesse de la bestiole.
- **Carapace** : Une bestiole équipée d'une carapace est plus résistante lors d'une collision avec une autre bestiole. Ainsi la probabilité qu'elle meure est réduite d'un coefficient $\omega \in [1; \omega^{\max}]$. En revanche, la présence de cette carapace réduit sa vitesse de déplacement d'un facteur $\eta \in [1; \eta^{\max}]$.
- **Camouflage** : Une bestiole disposant d'un camouflage sera plus difficile à détecter par les autres bestioles. Cette capacité de camouflage est modélisée par $\psi \in [\psi^{\min}; \psi^{\max}]$ avec $0 \leq \psi^{\min} \leq \psi^{\max} \leq 1$. Quand ψ est égale à 0, la bestiole n'est pas du tout camouflée (c'est aussi le cas de toute bestiole sans l'accessoire *camouflage*) et est donc totalement détectable par les autres bestioles. Plus la valeur de ψ s'approche de 1, plus le camouflage est efficace, jusqu'à atteindre le camouflage parfait (bestiole indétectable) pour $\psi = 1$.

Les valeurs définissant les limites des coefficients et des capacités seront paramétrables à travers la configuration initiale de l'aquarium.

2.4.3 Règles de détection

Pour l'instant, les règles de détection sont définies de manière statique ; il n'y a pas de probabilité de détection à estimer à chaque pas de la simulation. Ainsi la bestiole A détecte la bestiole B si au moins une des conditions suivantes est réalisée :

- A entend B et la capacité de détection de A (avec ses oreilles) est plus élevée que la capacité de camouflage de B ($\gamma_o(A) > \psi(B)$),
- A voit B et la capacité de détection de A (avec ses yeux) est plus élevée que la capacité de camouflage de B ($\gamma_y(A) > \psi(B)$).

On notera que les règles de détection entre deux bestioles ne sont pas symétriques.

2.5 Comportement dynamique des bestioles

Chaque bestiole possède dès sa naissance, un comportement particulier parmi une liste de comportements prédéfinis :

- comportement **grégaire** : une bestiole dite *grégaire* ajuste sa direction sur la direction moyenne des bestioles environnantes.
- comportement **peureuse** : lorsque le nombre de bestioles autour d'une bestiole dite *peureuse* devient trop important, celle-ci fuit à vive allure dans la direction opposée avant de reprendre sa vitesse de croisière.
- comportement **kamikaze** : une bestiole dite *kamikaze* est attirée par la bestiole la plus proche et tente de l'attaquer en cherchant à provoquer une collision.
- comportement **prévoyante** : une bestiole dite *prévoyante* estime les trajectoires des bestioles autour d'elle et ajuste sa trajectoire pour éviter d'éventuelles collisions.
- comportement **à personnalités multiples** : une bestiole dite *à personnalités multiples* adopte successivement d'autres comportements choisis aléatoirement au cours du temps. Néanmoins, elle reste une bestiole à personnalités multiples.

Dans la définition des comportements d'une bestiole, les expressions *bestioles environnantes*, *bestioles qui sont autour* correspondent à l'ensemble des bestioles détectées par cette bestiole.

Attention, le comportement des bestioles évolue de manière dynamique c'est-à-dire qu'une bestiole donnée pourra changer de comportement, suite à l'occurrence d'un événement extérieur.

3 Interface Homme Machine

L'archive associée à ce sujet, contient une réalisation de la version initiale de la simulation. Cette réalisation propose une interface graphique homme-machine qui exploite la bibliothèque CImg, afin de présenter visuellement l'état de l'aquarium à chaque instant de la simulation. Une implémentation des nouvelles fonctionnalités reposant sur cette même interface homme-machine est suffisante.

En revanche, il est demandé que la visualisation de l'aquarium permette de différencier les différents types de bestioles (c'est-à-dire selon leur comportement) ainsi que d'identifier l'ensemble des capteurs et accessoires supporté par chaque bestiole.

4 Analyse des résultats de simulation

La simulation doit permettre d'observer l'évolution de la population initiale de bestioles pendant toute la durée de celle-ci.

Cette simulation doit pouvoir fournir à chaque instant (ou pas de simulation), un état précis de la population de bestioles contenues dans l'aquarium (avec leurs caractéristiques).

L'état en fin de simulation fera office de résultat ou bilan final de la simulation.

Néanmoins, il est aussi envisagé de pouvoir produire une analyse fine du déroulement de la simulation en étudiant les états produits pour chaque pas de la simulation. L'ensemble de ces états pourra ainsi alimenter différents algorithmes d'analyse s'intéressant à des critères particuliers et/ou à des configurations initiales spécifiques.

5 Livrables & déroulement

5.1 Livrables

L'analyse du besoin exprimée doit conduire à une modélisation de la solution envisagée au travers de design patterns. L'utilisation de design patterns **appropriés** pour modéliser certains aspects statiques ou dynamiques de la simulation est demandée. La justification (textuelle) du choix de ces design patterns fait partie de cette demande.

Par ailleurs, la modélisation envisagée doit être formalisée au travers de diagrammes UML (diagramme des cas d'utilisation, diagramme de classes et diagrammes de séquences) accompagnée de textes explicatifs.

L'implémentation doit être réalisée en C++ et peut s'appuyer sur le code déjà fournie. Elle doit évidemment mettre en œuvre la modélisation envisagée.

Une présentation / démonstration viendra clôturer le projet.

5.2 Suggestion d'organisation

Il peut être judicieux de regrouper l'ensemble des fonctionnalités demandées, en 3 ou 4 groupes et de prévoir :

- une phase de conception globale qui identifie le périmètre (et les interfaces) de chaque groupe,
- pour chaque groupe (en parallèle éventuellement) une phase de conception détaillée suivie d'une phase de développement puis d'une phase de test pour valider la conformité du travail effectué par le groupe vis-à-vis du plan global,
- une phase d'intégration rassemblant les réalisations de chaque groupe pour produire l'application finale
- une phase de tests de cette application montrant et quantifiant l'adéquation de cette application avec les attentes exprimées dans le sujet.

L'approche proposée permet de produire les différents livrables attendus (et de valoriser des résultats mêmes partiels) ; néanmoins, l'organisation effective du projet reste à la discrétion de l'équipe.