



U.E. 5.8 : « B.E. Simulation »

CI - 2019



ENSTA Bretagne
2 rue F. Verny
29806 Brest Cedex 9, France

MATAR Jalal jalal.matar@ensta-bretagne.org
TERSEUR Julie julie.terseur@ensta-bretagne.org

Sommaire

Introduction.....	3
1. Analyse conceptuelle du problème.....	4
1.1. La zone océanique.....	4
1.2. Les artefacts et la boîte noire.....	5
1.3. Les drones.....	5
1.4. Le bateau <i>Millenium</i>	6
1.5. Le scénario.....	7
2. Conception de la simulation.....	8
2.1. Moteur utilisé.....	8
2.2. Modèle d'une entité.....	8
2.3. Retour sur les hypothèses adoptées et comportements voulus	8

Introduction

La présente étude est commandée par M. Bobba Fett, qui souhaite localiser une boîte noire dans une zone de type océanique contenant par ailleurs d'autres artefacts. En particulier, le client souhaite que la mission soit la plus rapide possible, et que son coût reste dans un budget limité.

Aussi, la société UnderWaterExplorer se propose de livrer au client une simulation de la mission. La simulation repose sur le largage de drones sous-marins par le bateau *Millenium* ; drones qui devront parcourir et analyser la zone à la recherche de la boîte noire.

Afin de minimiser les coûts et la durée de la mission, on va s'intéresser à optimiser le nombre de drones déployés (dans la limite de 5 drones) et au mouvement qu'ils devront adopter.

Le présent rapport commence par une analyse conceptuelle du problème décrivant les différents acteurs et leurs interactions dans un scénario, puis s'intéresse à la conception de la simulation, c'est-à-dire aux solutions retenues en termes de code.

Un fichier Excel fourni en annexe présente un exemple de présentation des exigences (fonctionnelles, contraintes...) ainsi que leur ordre de priorité.

1. Analyse conceptuelle du problème

Le système étudié est le bateau *Millenium* avec les drones qu'il déploie. Tous sont en interaction avec les artefacts, la boîte noire et la zone océanique.

Les entités de simulation identifiées sont donc :

- Le bateau *Millenium*
- Les drones déployés
- Les artefacts
- La boîte noire
- L'océan.

Il apparaît au vu des comportements respectifs des artefacts et de la boîte noire que ceux-ci sont très similaires ; aussi, pour alléger le code et proposer une simplification, on considérera la boîte noire comme étant un type particulier d'artefact (au niveau de la couleur et de la forme).

Par ailleurs, comme le bateau déploie les drones, ceux-ci sont des entités autonomes, mais temporaires, dont le cycle de vie dépend de celui du bateau.

On peut rajouter aux entités précédentes une entité Scénario ; en effet, on conserve et adapte le code fourni en correction de TD, pour lequel le Scénario crée la zone, ses artefacts et le bateau.

On caractérise les entités ainsi identifiées par leur type, leurs variables d'état, leurs paramètres techniques et données d'initialisation, leurs événements et leurs comportements.

1.1. La zone océanique

Entité	Zone océanique
Type d'entité	Permanente
Variables d'état	/
Variables statistiques	/
Paramètres techniques et données d'initialisation	Dimensions (size, épaisseur)
Évènements	/
Comportement	/

Le cahier des charges stipule que les dimensions de la zone sont de 20 * 20 km en surface, pour 3 km de profondeur.

Dans un premier temps l'océan reste « statique », on n'inclut donc pas d'évènements, de nouvelle création de bouées, etc. Plus tard, il pourrait être intéressant de simuler des vagues pour avoir un cadre plus réaliste.

1.2. Les artefacts et la boîte noire

Entité	Artefacts
Type d'entité	Permanente
Variables d'état	isDetected
Variables statistiques	/
Paramètres techniques et données d'initialisation	Nom Type Couleur Position Vitesse Accélération Vitesse Rotation Accélération Rotation
Évènements	/
Comportement	/

L'état isDetected est initialisé à false (aucun artefact connu), et sera passé à true par un évènement du drone lors de la première détection de l'artefact par celui-ci.

Les artefacts sont initialisés selon la loi suivante :

- 60% de sphères rouges, 30% de cylindres jaunes, 10% de boîtes vertes.
- Leur position est déterminée au moment où le scénario se lance, de manière aléatoire, et reste fixe par la suite (donc pas de vitesse ni de fait d'accélération)
- Un artefact supplémentaire est créé, de type « boîte noire ». Il n'est pas pris en compte dans les pourcentages précédents.

1.3. Les drones

Entité	Drones
Type d'entité	Temporaire (mais autonome)
Variables d'état	Identifying batteryDied
Variables statistiques	/
Paramètres techniques et données d'initialisation	Nom DetectionRadius Position BatteryLife Vitesse Accélération Vitesse Rotation Accélération Rotation
Évènements	detectArtifact BatteryLife
Comportement	RechercheBoiteNoire

On initialise le drone avec un nom, une position (là où il est largué par le bateau), une portée de sonar, et un pourcentage de batterie.

Comme une batterie faible/épuisée arrête le drone, on a un état `batteryDied` qui va caractériser la capacité du drone à continuer sa mission. Son autre variable d'état est `identifying`, qui caractérise l'état du drone en train d'identifier un objet détecté, par opposition à la détection préalablement effectuée lors du déplacement en surface.

Les événements associés aux états susdits sont `BatteryLife`, qui décrémente le pourcentage de batterie restant par unité de temps, et `detectArtifact`, correspondant à l'émission sonar par le drone et la récupération des positions d'artefacts (mais pas leurs types) détectés.

Le drone suit le comportement `RechercheBoiteNoire` défini dans le cahier des charges :

- Le drone est d'abord largué en surface. Il se déplace selon un mouvement d'« accordéon » à raison de 4m/s. Toutes les minutes, il scanne son environnement et fait passer les nouvelles bouées en « détectées » (boolean `isDetected`).
 - S'il détecte une ou plusieurs nouvelles bouées, il plonge à raison de 1m/s pendant 10 mètres, puis se dirige vers sa cible en ligne droite. Il la contourne (pour la scanner) et remonte à une vitesse de 2m/s. On répète cette opération pour toutes les nouvelles bouées détectées.
 - Quand il a photographié toutes les bouées connues, le drone remonte une « dernière » fois et reprend son mouvement d'accordéon.
 - Si le drone rencontre la boîte noire, il s'arrête et envoie l'information au bateau. Il attend ensuite l'arrivée du bateau (pendant que les autres drones retournent au navire, informés par le bateau).
- Si la batterie du drone vient à descendre en-dessous de 5%, il s'arrête et envoie l'information au bateau, dans le but que celui-ci vienne lui fournir une batterie neuve.

1.4. Le bateau *Millenium*

Entité	Bateau
Type d'entité	Permanente
Variables d'état	<code>clichesAIdentifier</code>
Variables statistiques	/
Paramètres techniques et données d'initialisation	Nom Position Vitesse Accélération Vitesse Rotation Accélération Rotation
Évènements	<code>DroneArrival</code>
Comportement	<code>IdentificationCliche</code>

Le bateau est initialisé au centre de la zone océanique. Il peut se déplacer librement, à raison de 8m/s.

Quand il s'initialise, le bateau crée les drones avec l'évènement `droneArrival` et les stocke dans la liste drones.

Le bateau *Millenium* dispose d'une liste de clichés à identifier (quand ce n'est pas le drone qui identifie l'artefact lui-même) qu'il traite au fur et à mesure selon le comportement `IdentificationCliche` :

- Le bateau reçoit un cliché d'un drone. Le cliché est mis dans la liste des clichés à identifier.
- L'équipe identifie un cliché à la fois (on ne peut paralléliser cette tâche).
 - Pour les artefacts sphériques ou cylindriques, l'identification n'échoue jamais.
 - Si l'artefact est une boîte, le taux de réussite de l'identification n'est que de 90%.
 - Si l'identification de la boîte a échoué, le drone doit refaire une prise de clichés. Cet ordre lui est transmis par l'équipage.
 - L'analyse du deuxième cliché n'a pas d'erreur.
 - Si l'artefact est la boîte noire cherchée, le bateau envoie le message aux drones de revenir vers lui, et va à la position de la boîte noire.
- Si pendant ce process le navire reçoit une information de batterie morte, il va à la position du drone en détresse et change la batterie (prend de 20 à 40 minutes).

1.5.Le scénario

Entité	Scénario
Type d'entité	Permanente
Variables d'état	/
Variables statistiques	/
Paramètres techniques et données d'initialisation	Nom Children
Évènements	boueeArrival NavireArrival OceanArrival
Comportement	/

Le scénario crée la zone océanique, les artefacts (de manière aléatoire), et le navire. Il ne crée pas les drones ; cette création s'effectue dans le navire.

2. Conception de la simulation

2.1.Moteur utilisé

On réutilise le moteur graphique fourni en TD. Une seule modification a été apportée, quant à la vitesse de zoom/dézoom, pour faciliter la vue dans une grande zone. (mais le moteur de base reste parfaitement utilisable)

2.2.Modèle d'une entité

Une entité est constituée, en termes de code, de deux packages.

Le premier package est le package « EntityName », qui comprend :

- Une classe Entity, héritant de SimEntity. Elle gère les événements de l'entité.
- Une classe EntityInit, héritant de SimInitParamters. Elle sert à initialiser les variables de classe/instance de l'entité
- Une classe EntityFeature, héritant de SimFeatures, qui répertorie les autres variables utiles à l'utilisation de l'entité, sans avoir besoin de les initialiser.

Le deuxième package est le package « EntityNameRepresentation3D », constitué d'une classe éponyme et de son interface. Cette classe sert à définir la représentation graphique de l'entité, pour laquelle on utilise les formes prédéfinies, qu'on peut translater/tourner etc.

2.3.Retour sur les hypothèses adoptées et comportements voulus

Drone

Le drone possède deux événements. DetectArtifact() est lancé dans la méthode afterActivate() et se répète toutes les minutes. Cela sert à ne pas mélanger la partie détection de la partie mouvement ; ces deux actions pouvant alors s'effectuer en parallèle.

DetectArtifact() fait appel à un SimObjectRequest qu'on initialise dans le constructeur. Cet objet va permettre, à terme, de renvoyer les bouées nouvellement détectées, et de faire passer en état « isDetected » celles qu'on ne connaissait pas précédemment. Pour chacune des bouées ainsi obtenues, le drone récupère sa position et l'affiche dans le log. Si la bouée détectée est la boîte noire recherchée, un event est posté d'arrêt, mais cette fonction n'est présentement pas fonctionnelle (le drone reste « bloqué » dans le mouvement qu'il était en train d'effectuer).

Le comportement voulu pour DetectArtifact, à terme, est que si un artefact est nouvellement détecté, le drone enregistre sa position courante en surface, plonge et effectue le scan, puis remonte et reprend le mouvement « accordéon » en surface depuis la position mémorisée. Cela n'est présentement pas implémenté.

DetectArtifact doit se répéter toutes les minutes, d'après le cahier des charges ; cependant, pour avoir plus de données lors de la simulation, on a fait le choix de le faire s'effectuer toutes les 15 secondes (mais cela est aisément modifiable en LogicalDuration.ofMinutes(1)).

Enfin, le dernier comportement voulu de DetectArtefact est l'envoi de la position de la boîte noire au Bateau lorsque celle-ci est détectée ; en effet, comme les drones sont des « enfants » du navire (créés pendant son AfterActivate), on devrait pouvoir facilement établir un canal de communication entre eux. Cela n'a pas été réalisé.

BatteryLife() est l'autre évènement du drone ; chaque minute, la batterie se décrémente de 0.5 unités par minutes (modèle affine). On pourrait envisager un autre modèle mathématique. Les batteries sont initialisées à 100 unités, soit 3h20mn (réglable). La batterie lance une alerte quand elle atteint 5%, présentement dans le log mais elle pourrait utiliser le canal de communication avec le bateau susmentionné.

Zone

La taille de la zone est présentement fixée à $200 * 200 * 30$ unités (réduction d'un facteur 100) pour pouvoir apprécier la simulation plus rapidement ; en effet, avec les « vraies » valeurs, et compte tenu de la vitesse de déplacement des drones, la simulation prend beaucoup plus de temps et est plus difficilement visualisable. Revenir aux vraies valeurs n'est que l'affaire d'un facteur multiplicatif.

Un drone ayant une portée sonar de 5 km (soit 10km de diamètre, soit la moitié de la zone), il apparaît qu'une « période spatiale » du drone suffirait à détecter toute la zone. On envoie deux drones dans des directions opposées pour réduire les contraintes de temps ; des drones supplémentaires peuvent être envisagés si l'on constate que malgré la couverture de zone, la mission s'effectue trop lentement. Chaque drone couvre donc une moitié de zone ; leur comportement est symétrique.

Le calcul des trajectoires de drones a été réalisé afin que le drone passe par le centre du cercle inscrit lui permettant de détecter dans les coins. Son rayon étant de 5km, on effectue un théorème de Pythagore pour récupérer l'abscisse et l'ordonnée du point en question ; on prendra un point situé à (3,4) m du coin. (puisque $3^2 + 4^2 = 5^2$)

L'EntitySequencer du drone (en surface) ne considère pas de points cibles, mais une trajectoire prédéfinie selon ces calculs :

- Un mouvement rectiligne de 6km
- Un demi-tour de diamètre 7km
- Un mouvement rectiligne de 12 km.

Pour éviter la duplication de code (ne pas avoir une classe séquenceur par drone), et comme les drones ont des trajectoires symétriques par rapport au centre, on leur attribue le même type de séquenceur, à une variable « sens » (int valant 1 ou -1) près. Ce sens va déterminer leur trajectoire.

Sur le schéma ci-dessous, les courbes vertes et oranges représentent la trajectoire des deux drones, ainsi qu'un exemple de leur portée radar leur permettant d'atteindre les coins de la zone ; le point noir représentant le bateau au centre de la zone.

