

UnderWaterExplorer

# Simulation

UV 5.8

Yanis BERKOUK – Huseyin KIZIL  
30/01/2019

## Tables des matières

1.	Cahier des charges .....	2
1.1	Description du problème .....	2
1.1.1	Le besoin du client .....	2
1.1.2	La société UnderWaterExplorer .....	2
1.1.3	Le système caché à détecter .....	2
1.1.4	La zone sous marine.....	2
1.1.5	Le système de de détection de UnderWaterExplorer.....	2
1.2	Résultats attendus .....	5
2.	Objectifs de l'étude .....	6
3.	Analyse du problème .....	6
4.	Modélisation du système .....	8
4.1	Entité Bateau/équipage .....	8
4.2	Entité Artefact .....	10
4.3	Drone .....	11
4.4	Scénario .....	13
5.	Implémentation du problème .....	14
5.1	Création de l'environnement et d'une entité bateau .....	14
5.1.1	La zone sous-marine .....	14
5.1.2	Les Artefacts .....	15
5.1.3	La Cible .....	15
5.1.4	Le Bateau.....	15
5.2	Création de l'entité drone.....	17
5.3	Simulation .....	17
6.	Analyse des résultats.....	21
7.	Axes d'amélioration de la modélisation .....	25

# **1. CAHIER DES CHARGES**

## **1.1 Description du problème**

### **1.1.1 Le besoin du client**

M.Boba Fett recherche un dispositif caché sous-marin dans une zone qu'il a préalablement identifié. Il a entendu parler de votre société UnderWaterExplorer. Elle est célèbre pour sa capacité à monter des expéditions sous-marines afin d'identifier des formes pertinentes sous l'eau. Elle a tout un savoir-faire notamment dans l'optimisation des missions sous-marines. M.Boba Fett souhaite que le travail soit exécuté le plus rapidement possible une fois sur zone. Mais son budget est limité.

### **1.1.2 La société UnderWaterExplorer**

La société UnderWaterExplorer dispose d'un navire principal permettant d'atteindre n'importe quelle mer du globe. Equipé d'une flotte de drones sous-marins, l'équipage du bateau peut programmer chacun des drones qui, via un sonar, vont analyser les fonds marins.

Vous êtes le spécialiste « mission » de UnderWaterExplorer.

L'insistance de M.Boba Fett sur la nécessité de rester le moins longtemps sur zone, vous pousse à mettre en place un outil d'aide à la planification de mission permettant notamment de dimensionner :

- le nombre de drones à envoyer, le stock de batteries à emporter
- les trajectoires à emprunter par chaque drone

### **1.1.3 Le système caché à détecter**

Le système caché à détecter est estimé pouvant avoir une taille variant de 2m à 5m il est de couleur noire et de forme cubique. Il peut se trouver n'importe où au niveau de la zone identifiée par M.Fett mais au moins à 1km de profondeur.

### **1.1.4 La zone sous-marine**

La zone à couvrir est un carré de 20 km par 20 km. La profondeur atteint 3km. Le fond marin présente de nombreux artefacts de type variés et pouvant flotter à différentes hauteurs par rapport au fond. 60% des artefacts sont des sphères rouges, 30% sont des cylindres jaunes, 10% sont des boîtes cubiques vertes.

On estime qu'il y a entre 30 et 40 artefacts dans la zone indiquée.

### **1.1.5 Le système de de détection de UnderWaterExplorer**

#### **1.1.5.1 Fonctionnement général du système**

Une fois sur zone, l'équipage du bateau principal « Le Millenium » lance les drones selon la programmation prévue. Chaque drone dispose alors d'une autonomie complète mais limitée dans le temps lié à la consommation des batteries.

Le drone reste en surface tant que le sonar ne détecte rien. S'il détecte une forme, il prévient l'équipage du bateau, puis il se rend à la position détectée en plongeant jusqu'à une certaine distance de la cible. Grâce à ses systèmes de projecteur et des caméras HD performants, il peut enregistrer ce qu'il voit. Dès la fin de l'enregistrement effectué, il remonte en surface à la position d'où il a plongé et envoie les images capturées au bateau principal par voie hertzienne.

Sur le bateau, une équipe identifie l'objet. Quand l'objet détecté est autre que parallélépipédique, l'identification est réussie à 100%. Sinon 90% du temps, l'objet est parfaitement reconnu. En cas contraire, l'équipage peut redemander au drone ayant envoyé le cliché de recommencer sa plongée. L'identification est sûre à 100% à l'issue de la deuxième plongée.

Lorsque le drone n'a plus de batteries, il se laisse remonter en surface et envoie un message au bateau principal sur sa position et l'absence d'énergie.

Une opération de récupération est alors engagée. Les batteries sont remplacées par l'équipage et le drone peut repartir. L'opération de remplacement de batterie nécessite au moins 20 minutes et peut atteindre 40 minutes. Il n'y a pas de lien entre la performance à un moment ou à un autre.

Le drone peut aussi retourner au bateau de lui-même selon la planification. Une fois en surface, il connaît la position du bateau principal qui envoie régulièrement sa position aux drones.

Une fois le cube noir trouvé, l'ensemble des drones retourne au bateau Millenium. Si des drones sous-marins sont en train de scanner une cible sous l'eau au moment où le cube noir est trouvé, ils finissent leur action avant de remonter puis de retourner au Millenium.

### 1.1.5.2 Le drone sous-marin

Il est doté d'un sonar détectant des cibles jusqu'à 5 km. Il réalise des scans omnidirectionnels toutes les minutes. Après la première détection de cible, le sonar continue de réaliser des échos afin de toujours suivre sa cible. Pour rappel le son se déplace à 1500m/s dans l'eau.

Le drone plonge sous l'eau par des ballast. Il reste en position horizontale pendant la phase de plongée. Il plonge à raison de 1m par secondes.

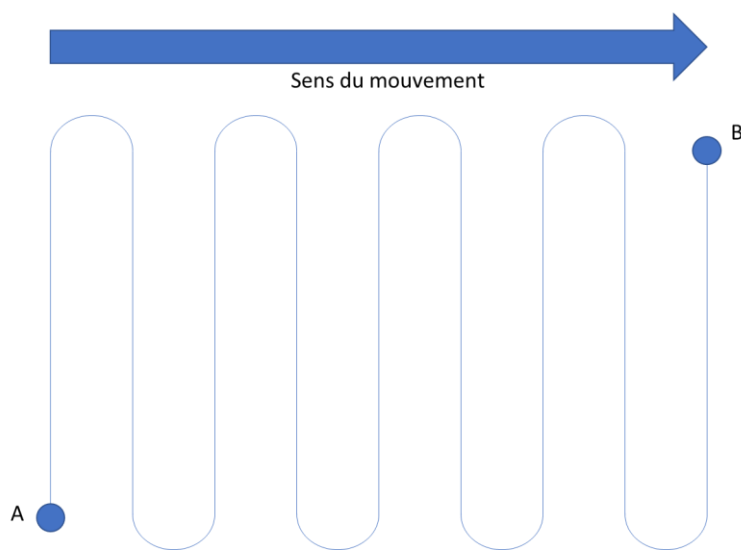
Une fois 10m sous l'eau, il se déplace directement en ligne droite vers sa cible. Sous l'eau le drone se déplace à raison de 3 m/s.

Une fois à 2 m du bord de la cible il effectue le tour à 360 ° de la cible en la scannant.

Une fois le scan réalisé, le drone vide ses ballasts rapidement et remonte en surface en position horizontale. Il remonte à raison de 2m par secondes.

L'envoi des clichés au bateau principal prend entre 3 et 5 minutes. Cela dépend de la qualité du signal, et de l'état de la batterie.

En surface le drone se déplace à raison de 4 m/s. Pour aller d'un point A à un point B, il se déplacerait selon un mouvement comme ci-dessous :



L'amplitude des mouvements est à déterminer pour avoir une couverture optimale.

Il est programmé pour réaliser le processus montée, descente comme expliqué ici pour chaque nouvelle détection. Les détections déjà traitées ne sont pas à analyser de nouveau.

La batterie offre une autonomie variable entre 1h et 3h. Cela semble assez aléatoire. On considère que même épuisée la batterie permet d'envoyer des messages au bateau principal.

### **1.1.5.3 Le bateau Millenium**

Le Millenium se déplace en surface à raison de 8 m par secondes. Il peut embarquer jusqu'à 5 drones. L'équipe d'analyse des clichés prend 10 minutes à analyser les clichés des drones. Ils ne peuvent paralléliser l'analyse des clichés.

De ce fait si l'équipe reçoit plusieurs clichés issus de drones, ils seront traités l'un après l'autre.

### **1.1.5.4 Cout de la mission**

Le cout de la mission présente un cout fixe indépendant du nombre de drones. Le cout d'emploi d'un drone représente 30% du cout de la mission fixe. Il est multiplié proportionnellement au nombre de drones.

## **1.2 Résultats attendus**

M. Boba Fett veut plusieurs devis figurant : le cout de la prestation et le temps mis pour analyser l'ensemble d'une zone selon le nombre de drones utilisés.

## 2. OBJECTIFS DE L'ETUDE

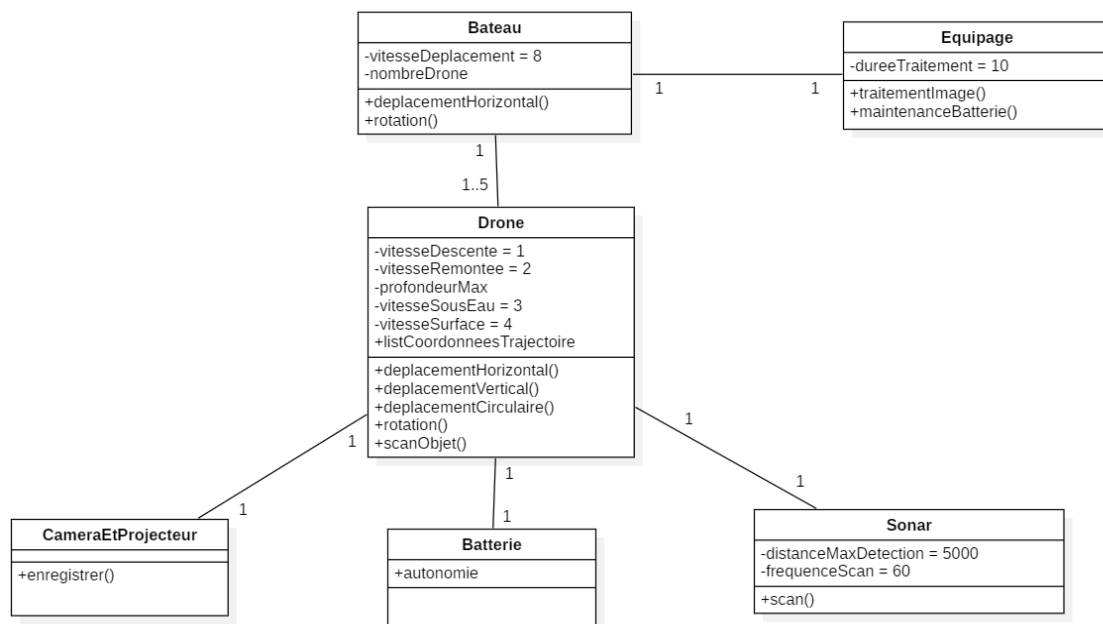
La société Underwater recherche un dispositif sous-marin caché sur une zone qui a été prédéfinie en amont. Souhaitant que la mission se réalise le plus rapidement possible une fois sur zone, la société Underwater dispose néanmoins d'un budget limité.

Pour ce faire, le client dispose d'un bateau ainsi que de 5 drones sous-marins équipés de sonar pour scanner le fond marin (§1.1.5.3).

Ainsi, l'étude suivante a pour objectif de simuler des missions d'analyses sous-marines et d'identifier, selon le critère de la durée et du coût, la mission optimale.

## 3. ANALYSE DU PROBLEME

Le système étudié est le bateau (et son équipage) ainsi que ses drones.



Notre système est constitué du bateau Millenium ainsi que d'un ou plusieurs drones se trouvant à son bord.

Le drone recherche depuis la surface, lorsqu'une cible est détectée, le drone descend à une vitesse de 1m/s pour atteindre une profondeur de 10m. Il se déplace ensuite avec une vitesse de 3m/s à la recherche de cible. Chaque drone est équipé d'un sonar en charge de scanner le fond (avec une fréquence d'1 minute) pour trouver l'objet cible ainsi que d'une caméra et d'un projecteur pour enregistrer dès qu'un objet est détecté.

Une fois l'enregistrement terminée, l'appareil remonte à la surface à une vitesse de 2 m/s pour envoyer ces données à l'équipage, (le temps d'envoi varie aléatoirement entre 3 et 5 minutes) qui traitera celles-ci en 10 minutes (§1.1.5.2).

Les systèmes environnant en interaction avec le système étudié sont essentiellement constitués des artefacts. L'environnement de la modélisation est une zone sous-marine d'un volume de 1200km<sup>3</sup>, représenté ci-dessous.

Cette zone sous-marine contient un nombre aléatoire d'artefacts compris entre 30 et 40.

60% des artefacts sont des sphères rouges, 30% sont des cylindres jaunes, 10% sont des boîtes cubiques vertes. Les artefacts se trouvent à différentes positions dans la zone et suivent une répartition aléatoire au sein de la zone de recherche. (§1.1.4)

L'élément à identifier est un cube de couleur noir ayant une longueur comprise entre 2 et 5 mètres.  
On déterminera la longueur du cube de manière aléatoire

Les entités de simulation identifiées à la lecture du sujet sont :

- Le bateau et l'équipage (que l'on appellera bateau)
- Les drones
- Les artefacts (inclus la cible)

On fera l'hypothèse que les cycles de vie des drones sont inclus dans celui du bateau. C'est le bateau qui coordonne l'opération des drones.

Les artefacts sont des entités autonomes permanentes.

On décide que c'est le scénario (qui représente le fil de l'histoire) qui crée les drones au cours du temps. Etant donné qu'il a besoin d'un événement temporel pour le faire il sera une entité de simulation également. La répartition dans l'espace des artefacts dans la zone d'exploration est effectuée aléatoirement.

C'est le drone qui porte la notion d'événement d'identifier dans l'artefact.

Nous allons dans les paragraphes qui suivent analyser chaque entité et identifier puis caractériser :

- leurs variables d'états.
- leurs propriétés d'initialisation et caractéristiques
- les événements temporels, et d'états
- les comportements et hypothèses de modélisation



## 4. MODELISATION DU SYSTEME

### 4.1 Entité Bateau/équipage

<b>Entité</b>	Bateau + équipage
<b>Type d'Entité</b>	Permanente
<b>Variables d'états</b>	nombreDrone positionBateauX positionBateauY listEnregistrement
<b>Variables statistiques/de scrutation</b>	tailleListEnregistrement
<b>Paramètres techniques et Données d'initialisation</b>	Nom vitesseDeplacement dureeTraitement
<b>événements</b>	deploiementDrone maintenanceBatterie deplacement traitementEnregistrement observationListEnregistrement finDeMission debutDeMission
<b>Comportement</b>	ecouteEnregistrement
<b>Processus Stochastique</b>	Aucun

On considérera dans cette étude que le bateau et son équipage forme une seule entité. Il s'agit d'une entité permanente.

- Cette entité est caractérisée par les variables d'état suivantes (§1.1.5.3) :
- Le nombre de drones contenu à son bord
- La position du bateau (suivant ses coordonnées dans le plan)
- La liste des enregistrements en attente d'être traitée par l'équipage

On suppose que la longueur de la liste des enregistrements en attente d'être traitées est aléatoire. Les caractéristiques du bateau sont :

- Son nom
- Sa vitesse de déplacement en surface de 8m/s
- La durée de traitement des enregistrements par l'équipage : 10 min par enregistrements reçue.

Les événements temporels identifiées pour le bateau sont :

- `debutMission` : cela correspond à l'initialisation de la mission. Le bateau démarre son chronomètre, attribue les coordonnées des points de trajectoire à chaque drone et engendre le déploiement (`deploiementDrone`).
- `deploiementDrone` : cet événement correspond à la création d'un nouveau drone. (§1.1.5.3)
- `maintenanceBatterie` : cet événement correspond au remplacement de la batterie par l'équipage. Nous supposons dans cette étude que la durée de remplacement, qui est comprise entre 20 et 40 min, suit une loi exponentielle de paramètre  $1/(30 \times 60)$ . Pour estimer le paramètre nous avons choisis le centre de l'intervalle [20 ; 40] minutes qui vaut 30 minutes. Nous l'avons ensuite converti en seconde. (§1.1.5.1)
- `deplacement` : Nous supposons que le bateau ne se déplace que pour récupérer les drones pour changer leurs batteries ou pour les récupérer lorsque la cible à identifier est retrouvée. On suppose que le bateau connaît en permanence la position des drones déployés.
- `traitementEnregistrement` : cet événement correspond au traitement des images envoyées par les drones. Cette durée est connue et vaut 10 min par enregistrement reçu. (§1.1.5.3)
- `observationListEnregistrement` : cet événement permet à l'équipage d'observer l'état de la file d'attente des enregistrements à traiter, compte tenu que les données ne sont pas traitées en série (§1.1.5.3)

Enfin le bateau écoute en permanence les drones et les données qu'ils envoient. Autrement dit le bateau est constamment disposé à recevoir les données transmises par les drones (enregistrements, position, autonomie).

Hypothèse:

- Puisque la cible est cubique et que lorsqu'un cube est détecté, son identification est sûre à 100%, un drone n'ira donc jamais inspecter 2 fois un même artefact.

## 4.2 Entité Artefact

Entité	Artefact
Type d'Entité	Autonome temporaire
Variables d'états	etatArtefact
Variables statistiques	
Paramètres techniques et Données d'initialisation	Nom Type positionArtefactX positionArtefactY positionArtefactZ
Événements	Aucun
Comportement	Pas de comportements déterministes identifiés
Processus Stochastique	Aucun

La variable d'état `etatArtefact` détermine l'état de l'artefact à tout instant :

- Artefact détecté par le drone (Détecté)
- Artefact non détecté par le drone (nonDétecté)

Il y a 4 types d'artefacts (§1.1.4) :

- Type 0 : le cube noire (= cible)
- Type 1: les sphères rouges
- Type 2: les cylindres jaunes
- Type 3: les cubes verts

Les paramètres techniques d'un artefact sont :

- Son nom
- Son type, avec une probabilité de 6/10 d'être de type 1, une probabilité de 3/10 d'être de type 2 et une probabilité de 1/10 d'être de type 3 (§1.1.5.1)
- Sa position initiale (`positionArtefactX`, `positionArtefactY`, `positionArtefactZ`) qui sera fournie par le scénario

Les hypothèses prises lors de la modélisation de l'artefact sont :

- Un artefact détecté mais qui ne correspond à la cible identifiée par le client est retiré de la simulation car les drones vont l'ignorer, c'est pour cela qu'ils sont considérés comme des entités temporaires (§1.1.5.1).

### 4.3 Drone

Dans cette étude nous considérons que les drones et ses composants (caméra et radar) ne forment qu'une seule entité.

Les drones constituent des entités permanentes dans notre scénario.

(§ 1.1.5.3)

Les variables d'états d'un drone sont :

- ses variables de position : positionDroneX, positionDroneY et positionDroneZ
- son activité : ce qu'il est actuellement en train de faire (scanner le fond, analyser un artefact, envoyer des données)
- l'autonomie de sa batterie

Les variables caractéristiques des drones sont les suivantes :

- Le nom du drone
- La position initiale qui correspond à la position à partir de laquelle il commence à scanner le fond marin.
- Les vitesses caractéristiques du drone dans les différentes phases de descente, de remontée, d'avance à la surface et d'avance sous l'eau.
- La liste des coordonnées des points de trajectoire : cette liste sera générée en fonction du nombre drone simulé.
- La surface balayée permet à l'équipage de connaître la surface de zone qui reste à explorer
- Enfin le nombre d'artefact identifié par un drone qui permet à l'équipage d'estimer le nombre manquant d'artefact à analyser.

<b>Entité</b>	Drone
<b>Type d'Entité</b>	Permanent
<b>Variables d'états</b>	positionDroneX positionDroneY positionDroneZ activiteDrone autonomieBatterie
<b>Variables statistiques/de scrutation</b>	dureeAutonomie
<b>Paramètres techniques et Données d'initialisation</b>	Nom positionInitial vitesseDescente vitesseRemonte vitesseSousMarine vitesseSurface listCoordonneesTrajectoire surfaceBalayee nombreArtefactIdentifie
<b>Événements</b>	eteindre() identifierArtefact() enregistrer() envoyerEnregistrement()
<b>Comportement</b>	scanner() seDeplacer()
<b>Processus Stochastique</b>	Aucun

Les événements temporels caractérisant l'évolution du drone au cours de la simulation sont les suivant:

- Une fois que le drone a scanné un artefact, il commence le processus d'identification en s'immergeant de 10 mètres sous l'eau via les ballasts. Il se dirige ensuite vers la cible en ligne droite. Enfin, lorsqu'il est à une distance déterminée (2m) il analyse l'objet (enregistrer). Une fois qu'il a fini cela, il remonte à la surface et envoie l'enregistrement (envoyerEnregistrement()) (§1.1.5.2).
- L'événement enregistrer correspond à l'allumage des projecteurs et à au lancement de la caméra. On définit l'enregistrement comme une prise de photos en rafale. On considère dans cette étude que la longueur des enregistrements fournie aux équipes est identique indépendamment du drone ou de l'artefact.

- L'envoi de l'enregistrement suit une loi normale de paramètre  $(4 \times 60, \frac{1}{4})$ . Nous modélisons l'envoi de l'enregistrement par une telle loi car il s'agit d'un phénomène qui dépend de plusieurs paramètres indépendants (état de charge, qualité du signal) (§1.1.5.2).
- Si la batterie est déchargée, le drone s'éteint mais continue d'émettre les données permettant au bateau et à l'équipage de le localiser, de le récupérer et de changer sa batterie. Sachant que la batterie se décharge entre 1h et 3h (et donc le drone s'éteint sur le même intervalle) nous modéliserons la durée d'activité du drone par une loi exponentielle de paramètre égale à l'inverse de la durée de vie moyenne d'une batterie qui vaut 2h, qui correspond en fait à la fréquence de changement de batterie (§1.1.5.2).

Hypothèses:

- Lorsqu'un drone envoie des données, il ne fait rien d'autre (pas de déplacement ni de scan).
- Toutes les batteries sont identiques et présentent les mêmes caractéristiques.
- La durée d'envoi des enregistrements par les drones est considérée identique quel que soit le drone.
- Le drone ne détecte qu'un artefact à la fois et l'artefact identifié par le drone correspond au plus proche. Ainsi un drone ne rencontrera jamais sur sa trajectoire un autre artefact que celui ciblé initialement.

## 4.4 Scénario

On fait le choix de modéliser les coordonnées de position des artefacts à leurs création en suivant une loi uniforme pour chaque coordonnée (X,Y,Z) puisqu'un artefact a autant de chance d'être à une position qu'à une autre (§1.1.4).

Ainsi listPositionArtefacts va contenir un triplet de réels correspondant à la position de chaque artefact que le scénario va lui fournir à sa création :

- positionArtefactX suit une loi uniforme sur l'intervalle [0,20000]
- positionArtefactY suit une loi uniforme sur l'intervalle [0,20000]
- positionArtefactZ suit une loi uniforme sur l'intervalle [0,3000]

Les paramètres d'initialisation d'un scénario sont les suivants :

- Le nom du scénario
- Le nombre de drone de la simulation en cours
- Les points de la trajectoire que les drones doivent suivre. Ces trajectoires dépendent du nombre de drones utilisé pour l'exploration.

Les événements du scénario sont les suivants :

- debutMission : On démarre ainsi la mission\*
- finMission : La mission se termine
- creationArtefacts : les artefacts seront créés au lancement de la simulation. Leur répartition dans l'espace sera déterminée, autrement dit, les coordonnées de position de chaque artefact et de la cible à identifier suivent une loi uniforme comme définie plus haut.

Entité	rechercheCibleSousMarineScenario
Variable d'états	Permanente
Variables d'états	Aucune
Variables statistiques/ de scrutation	Position de la cible
Paramètres techniques/ d'initialisation	Nom nombreDrones trajectoireDrones listPositionArtefacts
Événements déterministes	debutMission finMission creationArtefacts
Comportements déterministes	Aucun
Paramètres stochastiques	Processus de creationArtefacts

## 5. IMPLEMENTATION DU PROBLEME

### 5.1 Création de l'environnement et d'une entité bateau

L'environnement (artefacts, cible, océan) est créé par La classe *ScenarioEnvironmentCreation* du package *kizil\_berkoug.BE*. Ce package contient les classes ci-dessous.

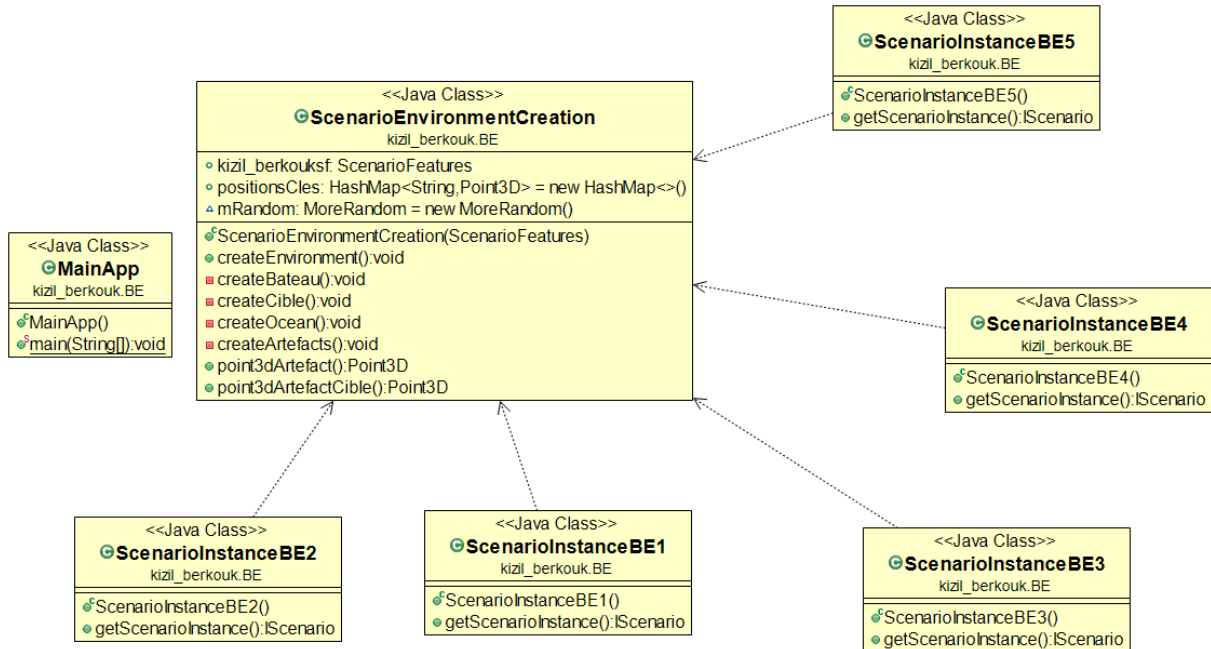
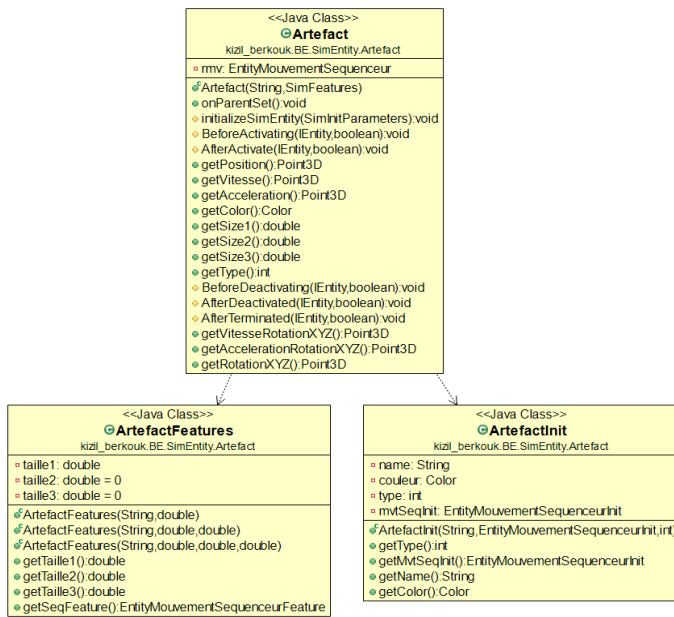


Figure 1 : Création de l'environnement

#### 5.1.1 La zone sous-marine

La classe *ScenarioEnvironmentCreation* du package *kizil\_berkoug.BE* permet de créer la zone sous-marine (un volume de dimension 20x20x3 km) avec la méthode *createOcean()*. Les dimensions de la zone de recherche peuvent être modifiée dans la classe *EntityOcean3DRepresentation* du package *kizil\_berkoug.BE.SimEntity.Ocean.3DRepresentation*.

## 5.1.2 Les Artefacts



La classe *ScenarioEnvironmentCreation* du package *kizil\_berkoug.BE* permet de générer des artefacts (méthode *createArtefacts()*) répartis aléatoirement, au sein de la zone de recherche, selon une loi de probabilité uniforme appliqué à chaque coordonnée de l'artefact (méthode *point3dArtefact()*). Pour la modélisation, nous avons créé trois types d'artefacts comme explicitée dans le cahier des charges (§1.1.4). La proportion de chaque type d'artefacts a été respecté et le nombre d'artefact est généré aléatoirement entre 30 et 40 (§1.1.4).

Figure 2 : Entité Artefact

## 5.1.3 La Cible

La cible est également créée dans cette même classe par la méthode *createCible()*. La cible étant cubique la longueur du côté est générée aléatoirement et varie entre 2 et 5 mètres (§1.1.3). De même sa position est aléatoire et chaque coordonnée de la cible suit une loi de probabilité uniforme sur un intervalle de longueur 20km pour les abscisses et les ordonnées et un intervalle de [-3, -1] km pour la profondeur. La génération de la position de la cible est faite par la méthode *point3dArtefactCible()*.

## 5.1.4 Le Bateau

Enfin le bateau est le dernier élément généré dans cette classe par la méthode *createBateau()*. Il est placé au centre de la zone et à la capacité de se déplacer. Néanmoins nous avons, comme convenu dans le cahier des charges, décider de le laisser statique au centre de la zone.



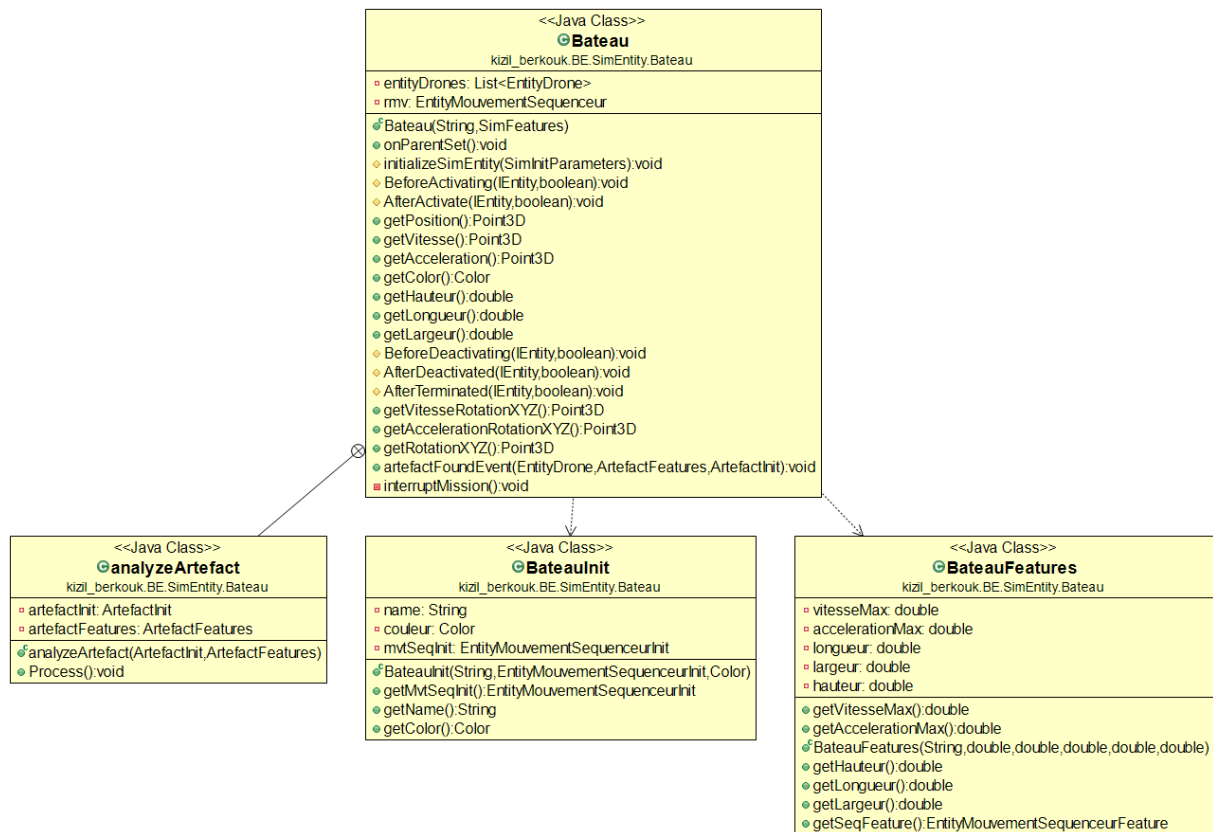


Figure 3 : Entité Bateau

## 5.2 Création de l'entité drone

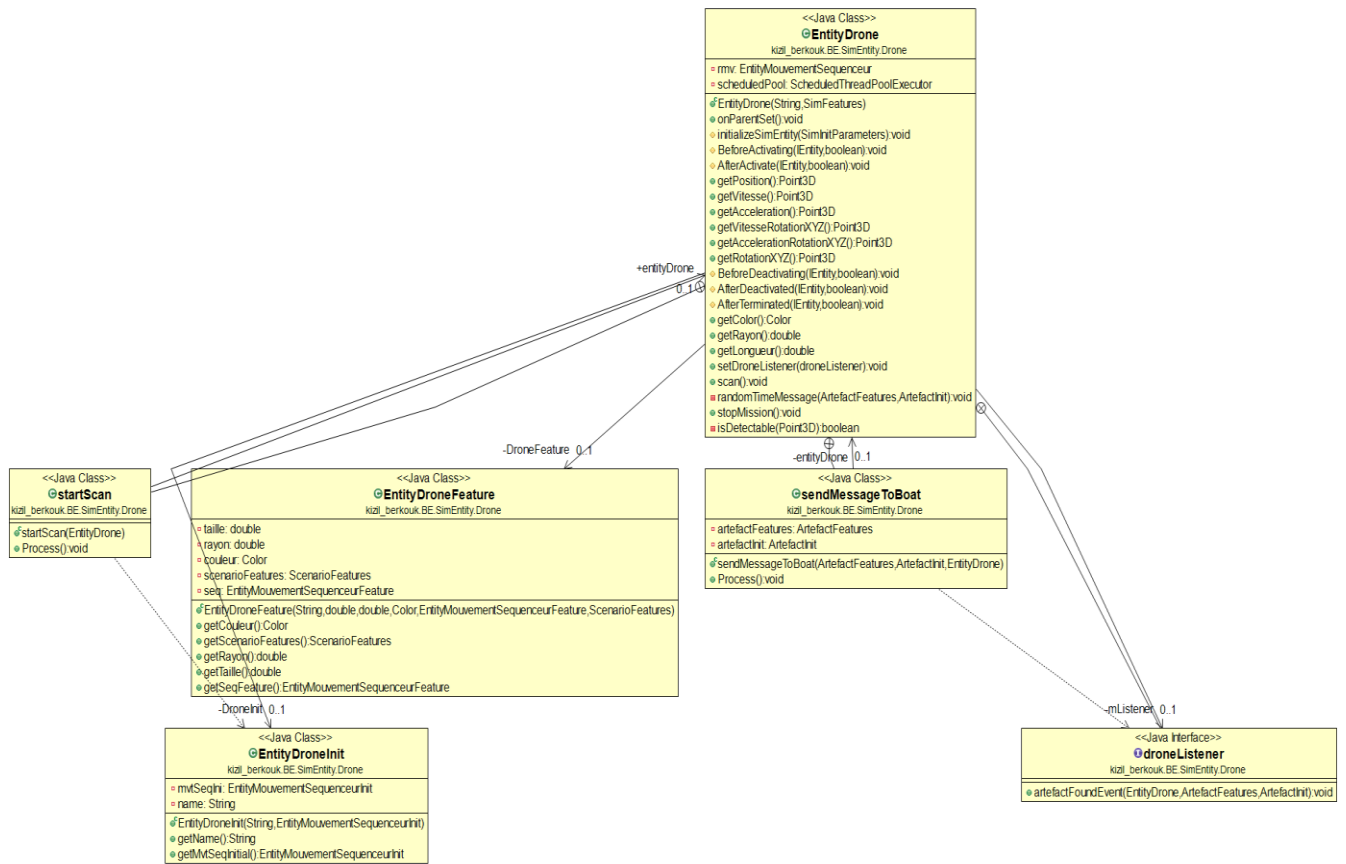


Figure 4 : Entité Drone

Les drones sont créés en utilisant le package **SimEntityDrone**

Les scénarii sont générés de la manière suivante :

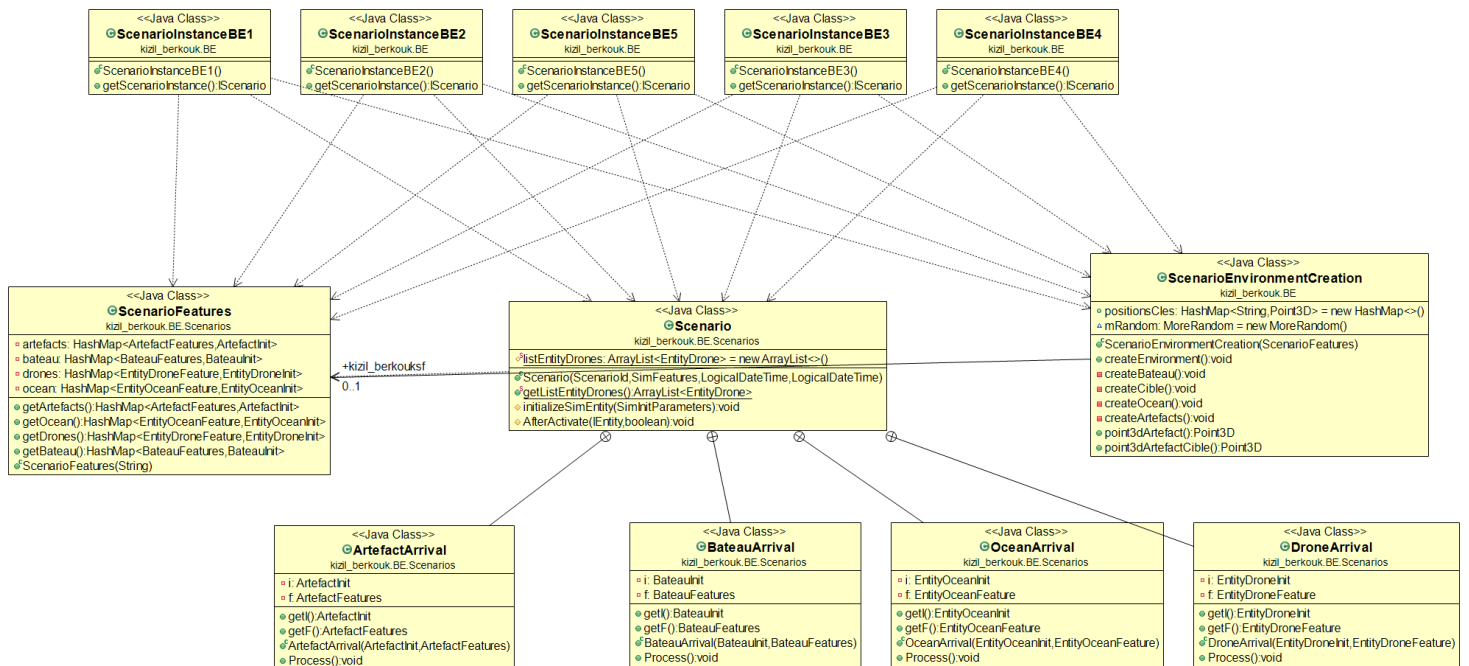


Figure 5: Gestion des scénarii

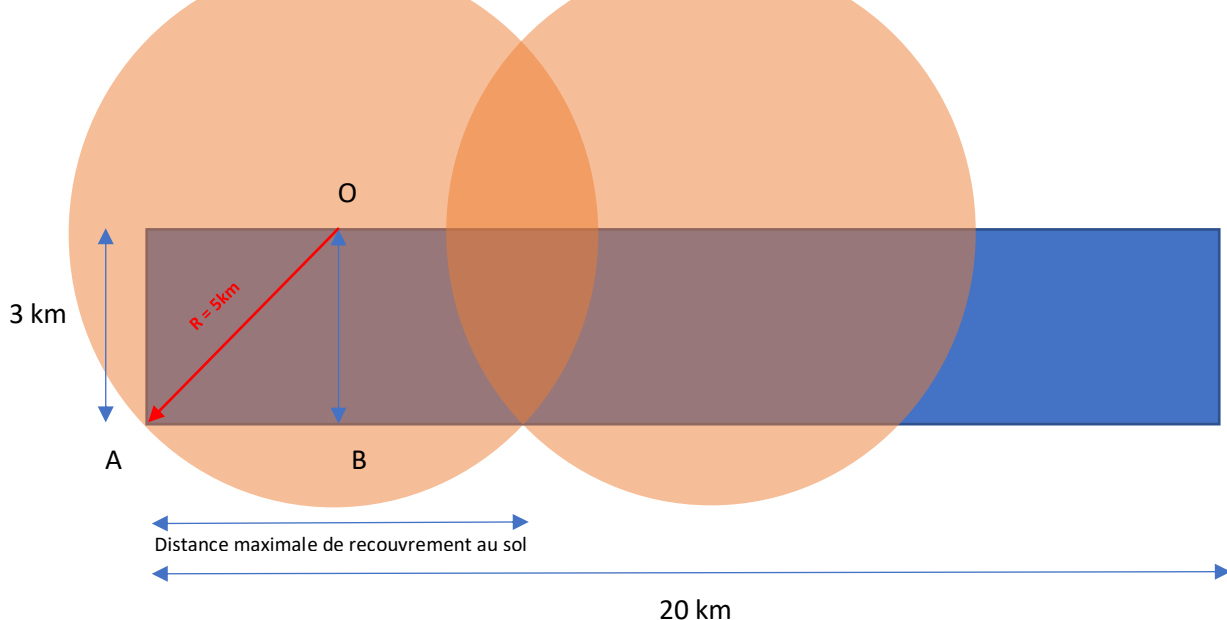


Figure 1 : Dispositif de détection radar pour une portée de 5 km

Sachant que la portée du radar vaut 5 km, nous avons modélisé l'espace visible par le radar, par une boule de rayon 5km et de centre la position du drone. Nous pouvons modéliser la figure ci-dessus.

La figure 1 représente une coupe de la zone de recherche dans le plan (X, Z), le problème est identique dans le plan (Y, Z). Pour pouvoir déterminer le rayon de la trajectoire courbée il faut connaître la « distance maximal de recouvrement au sol » compte tenu de la portée du radar de 5km.

Un théorème de Pythagore dans le triangle AOB nous permet d'estimer la « distance maximal de recouvrement au sol » qui vaut 4 km. Dans notre simulation les drones réalisent de phases rectilignes ou linéaires en surface à une vitesse constante de 4 m/s. Les mouvements des drones sont créés par rapport au nombre de drones par les classes :

- *EntityMouvementSeigneurKizilBerkouk1*
- *EntityMouvementSeigneurKizilBerkouk2*
- *EntityMouvementSeigneurKizilBerkouk3*
- *EntityMouvementSeigneurKizilBerkouk4*
- *EntityMouvementSeigneurKizilBerkouk5*

Ces classes contenues dans le package **kizil\_berkouk.BE.SimEntity.MouvementSeigneur** contiennent selon le nombre de drones choisis pour la simulation le séquençage de ce que va suivre chaque drone impliqué dans la simulation.

Ainsi la trajectoire des drones est la suivante :

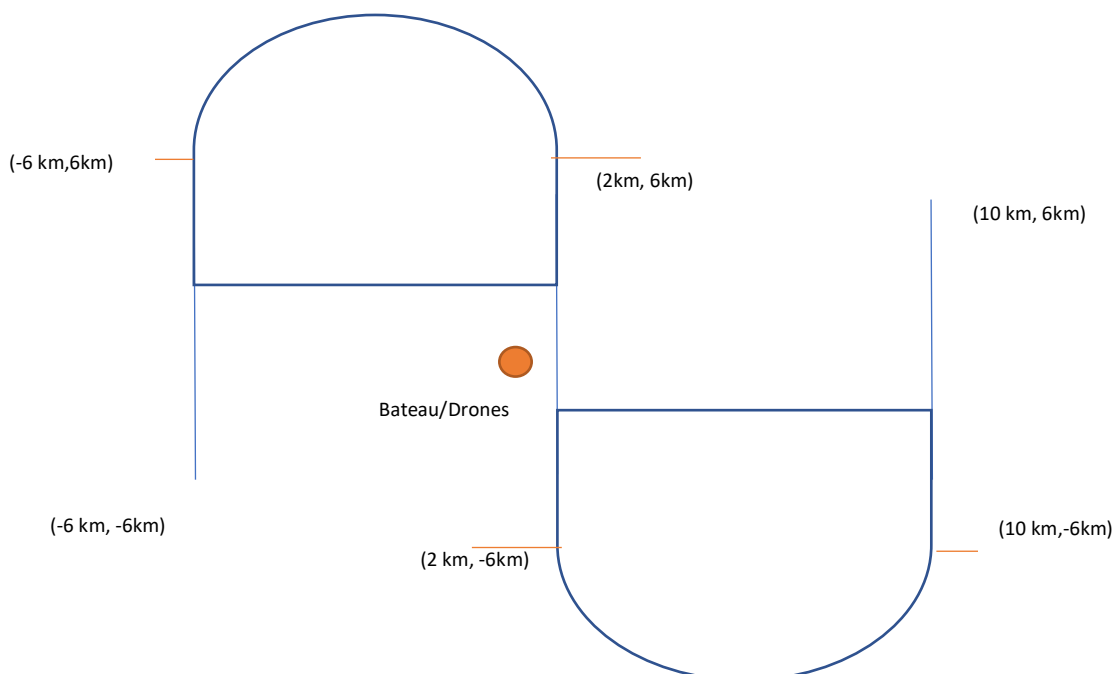


Figure 2 : Vue du dessus de la trajectoire des drones

Les déplacements élémentaires des drones sont :

- Des phases linéaires implémentées par la classe *RectilinearMover*
- Des phases circulaires implémentées par la classe *CircularMover*

D'après nos calculs nous avons estimé que la trajectoire idéale pour pouvoir couvrir toute la zone est celle illustrée en figure 2. Ainsi dans notre simulation tous les drones réalisent tout ou partie de cette trajectoire. Une distance totale de 61,12 km.

Dans la simulation à un drone, le drone quitte le bateau en direction du point (-6km, -6km) et réalise toute la trajectoire.

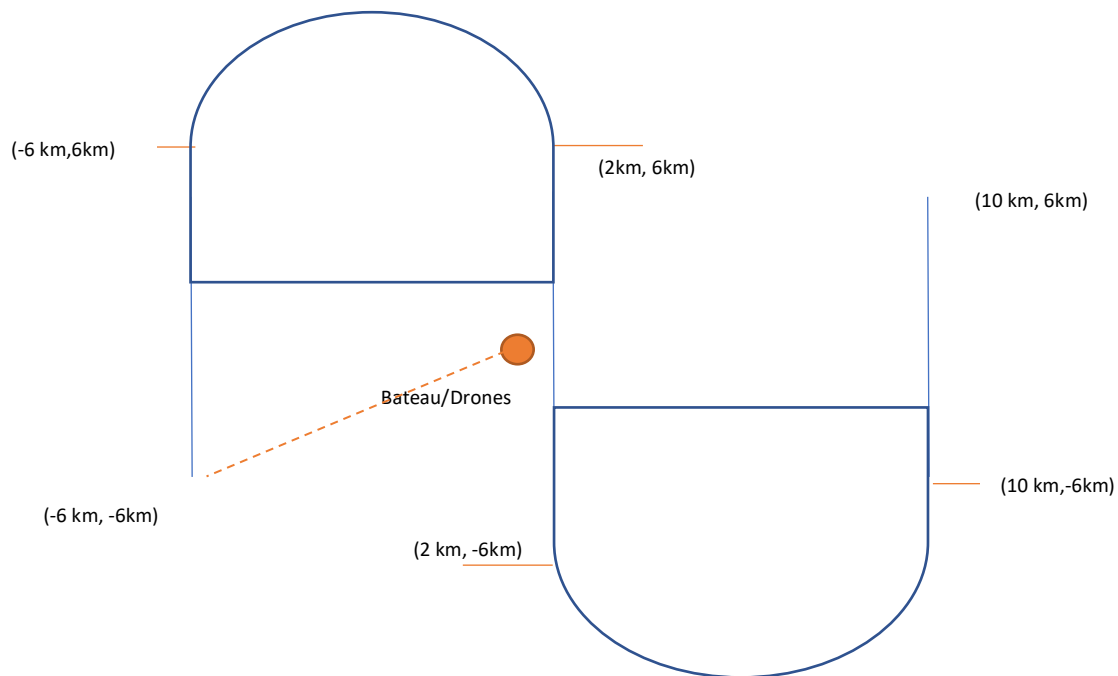


Figure 2 : Vue du dessus de la trajectoire du drone 1 scénario à 1 drone

Pour les simulations impliquant 3, 4 et 5 drones nous avons divisé la distance totale à parcourir par le nombre de drones en prenant en compte le retard de 10 min entre les départs de deux drones successifs.

## 5.3 Simulation

Tout d'abord, via « `scenarii_settings` », l'utilisateur choisit le nombre de drones pour la simulation et inscrit, s'il le veut, le germe initial (« `germeInitial : x` ») ; s'il n'est pas saisi, il sera généré de manière aléatoire. En fonction du nombre de drones souhaités (2 par défaut) le programme va chercher le scénario adéquat.

La capacité du drone à scanner a été implémenté de la manière suivante. Tout d'abord via des Post récurifs, le drone va exécuter la fonction « `scan` » toutes les minutes logiques. Pour détecter les bateaux autour de lui dans un rayon de 5 km, le drone va parcourir la liste des artefactas via « `ScenarioFeatures` ». Lorsque le drone a détecté et enregistré un artefact, il envoie cet enregistrement dans un délai de 3 à 5 minutes logique au bateau. La mise en place des transferts des messages a été effectué de deux manières différentes.

Premièrement, la communication entre le bateau et les drones se fait via la mise en place d'une variable d'instance « `entityDrones` » sous forme de List qui contient l'ensemble des drones. Cela représente le fait que le bateau connaît et « contrôle » les drones. Ensuite via une fonction basique, on envoie notre message.

Pour envoyer un message d'un drone au bateau, cela se fait via un listener qui permet au bateau d'écouter les messages provenant du drone (via une interface dans la classe `EntityDrone` qu'on implémente dans la classe `Bateau`).

Le bateau prendra enfin entre 20 et 40 minutes à analyser les données. Les éléments traités ne seront jamais traités 2 fois puisque les artefacts ne pourront pas les détecter 2 fois (ils sont retirés de la liste des artefacts visibles à chaque fois qu'ils sont détectés).

En outre 2 queues ont été mises en place dans le but d'éviter l'exécution concurrente de tâches. La première se situe au niveau de la classe `Bateau` qui sert à traiter les enregistrements reçus dans l'ordre. (Remarque : des mutex (via « `synchronized` ») ont été mis en place au niveau de cette queue pour ne pas perdre d'enregistrement). Une deuxième se situe au niveau des envoies d'enregistrement du drone au bateau. En effet, puisque nous n'avons pas réussi à mettre en place des interruptions et donc la plongée du drone, le scan continue lorsqu'un artefact est détecté bien que 30 minutes aient été ajoutées (Voir paragraphe 7).

## 6. ANALYSE DES RESULTATS

Pour proposer le nombre optimal de drone à envoyer permettant à la fois d'assurer un rapport temps/coût optimal. Nous avons réalisé par scénario 10 simulations et relevé le drone à l'origine de la découverte pour chaque simulation ainsi que le temps mis par l'équipe pour identifier la cible. Sur les 10 simulations nous avons donc calculé le temps moyen de découverte par scénario ainsi que l'occurrence de découverte par un drone.

Ces données se présentent de la manière suivante pour le scénario à un drone :

### SIMULATION 1 DRONE

Bilan	Drone découverte	Temps de Découverte
	Drone 1	14:18:00
	Drone 1	00:57:00
	Drone 1	13:10:00
	Drone 1	17:04:00
	Drone 1	11:57:00
	Drone 1	13:11:00
	Drone 1	12:05:00
	Drone 1	15:53:00
	Drone 1	13:09:00
	Drone 1	20:03:00
Bilan Moyen	DRONE 1 : 10/10	13:10:42

Ces données se présentent de la manière suivante pour le scénario à deux drones :

### SIMULATION 2 DRONES

Bilan	Drone découverte	Temps de Découverte
	Drone 1	16:20:00
	Drone 2	02:29:01
	Drone 2	07:58:01
	Drone 1	04:33:00
	Drone 1	06:45:00
	Drone 2	10:58:00
	Drone 2	08:42:00
	Drone 1	16:50:00
	Drone 2	10:29:00
	Drone 1	13:48:00
Bilan Moyen	DRONE 1 : 5/10 - DRONE 2 : 5/10	09:53:12

Ces données se présentent de la manière suivante pour le scénario à trois drones :

#### SIMULATION 3 DRONES

Bilan	Drone découverte	Temps de Découverte
	Drone 3	05:16:00
	Drone 1	09:34:01
	Drone 2	17:43:01
	Drone 1	14:47:00
	Drone 3	03:07:00
	Drone 3	18:28:00
	Drone 3	18:58:00
	Drone 2	18:08:00
	Drone 2	09:15:00
	Drone 3	09:23:00
Bilan Moyen	DRONE 1 : 2/10 - DRONE 2 : 3/10 - DRONE 3 : 5/10	12:27:54

Ces données se présentent de la manière suivante pour le scénario à quatre drones :

#### SIMULATION 4 DRONES

Bilan	Drone découverte	Temps de Découverte
	Drone 3	05:16:00
	Drone 1	12:18:01
	Drone 1	18:21:01
	Drone 2	13:42:00
	Drone 4	13:36:00
	Drone 1	06:42:00
	Drone 2	16:42:00
	Drone 4	06:38:00
	Drone 2	03:21:00
	Drone 2	14:51:00
Bilan Moyen	DRONE1:3/10 - DRONE2:4/10 - DRONE3:1/10 - DRONE 4 : 2/10	11:08:42

Ces données se présentent de la manière suivante pour le scénario à cinq drones :

#### SIMULATION 5 DRONES

Bilan	Drone découverte	Temps Découverte	de
	Drone 1	06:52:00	
	Drone 2	08:12:01	
	Drone 4	05:15:01	
	Drone 3	03:28:00	
	Drone 2	14:46:00	
	Drone 1	06:42:00	
	Drone 1	10:42:00	
	Drone 5	13:56:00	
	Drone 2	21:21:00	
	Drone 1	08:06:00	
Bilan Moyen	DRONE1:4/10 DRONE2:3/10 DRONE3:1/10 DRONE4:1/10 DRONE5:1/10 09:56:00		



On en déduit ainsi que :

- Le nombre de drone optimal pour un temps de mission minimale est de 2 drones
- Dans ce cas le durée de la mission moyenne serait de 9 heures et 52 minutes
- Compte tenu du nombre de drone le coût de la mission serait de 1,6 fois le coût de la mission

Tableau 1 : Devis

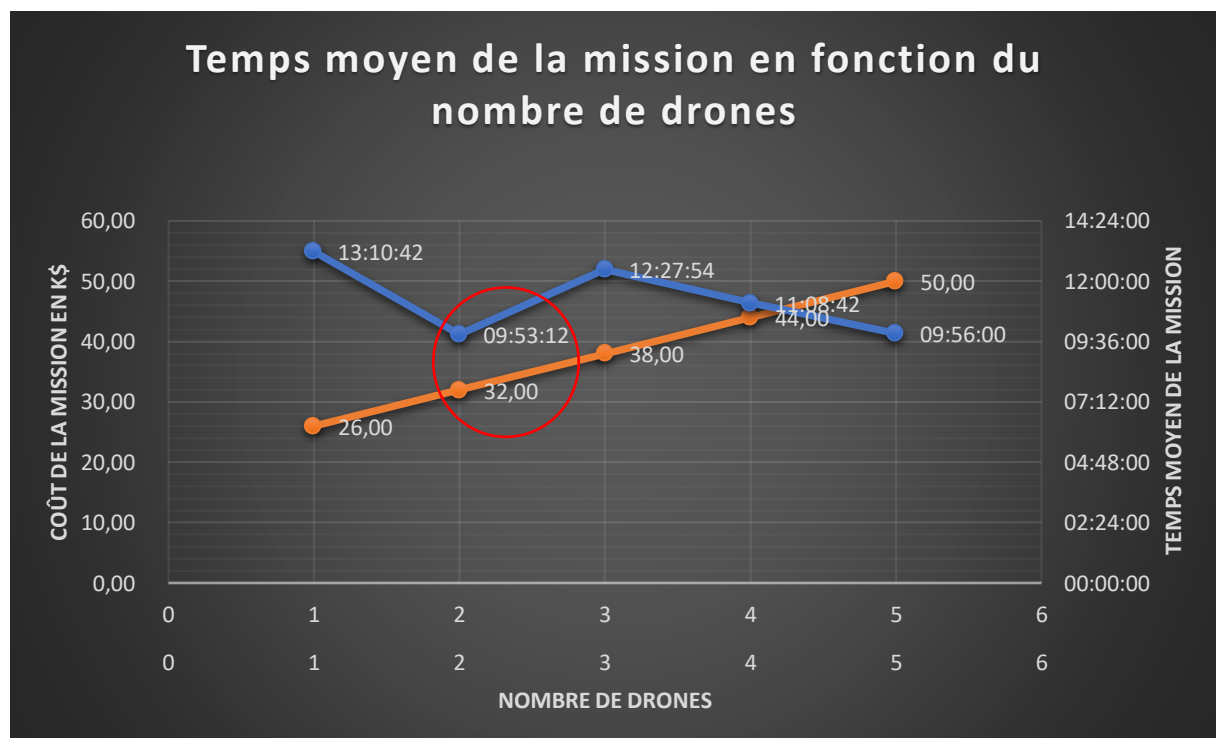
Pour parvenir à fournir un devis chiffré nous estimerons que le coût fixe de la mission est de 20k\$.

Ainsi selon notre simulation, le coût de la mission idéale s'élève à 32k\$ soit 60% plus élevée que le coût fixe, pour une durée de mission de 9 heures et 52 minutes avec 2 drones.

Nombres de Drones	Temps Moyen	Coût de la mission(k\$)
1	13:10:42	26,00
2	09:53:12	32,00
3	12:27:54	38,00
4	11:08:42	44,00
5	09:56:00	50,00

Il semble inutile d'utiliser 5 drones en effet le coût de la mission serait plus élevée de 150% et pour une durée de mission de 9 heures et 56 minutes.

Le graphe ci-dessous montre que le rapport cout/durée de la mission est optimal permet de synthétiser la situation.



## 7. AXES DE PROGRESSION DE LA MODELISATION

Notre simulation met en jeu des drones avec une autonomie illimitée et qui ne se déplace qu'en surface. Ce qui sous-entend qu'ils arrivent à détecter des artefacts depuis la surface de l'eau.

- Dans nos modélisations le déplacement des drones ne se fait qu'en surface, et le drone identifiant un artefact ou la cible le fait directement depuis la surface. Nous n'avons pas pu créer les événements *Interrupt()* permettant d'arrêter le drone peu importe la phase de séquençage dans laquelle il est engagée. (Le code essayé est disponible en commentaire)
- Par manque de temps, nous n'avons pas pu implémenter l'autonomie de la batterie du drone. En effet, si cet événement *Interrupt()* avait été créé nous aurions pu, interrompre le mouvement du drone après un temps d'exploration compris entre 1h et 3h pour une durée comprise entre 20 et 40 min (§)(temps de changements de la batterie du drone par le bateau).
- 

Bien que nous n'ayions pas eu le temps de simuler les phases de déplacements sous-marines du drone et par souci d'authenticité de simulation nous avons ajouté au temps de traitements des images une durée de 30 minutes qui représente le temps moyen passé sous l'eau lorsqu'un artefact est identifié. Ainsi le temps de durée de la mission est plus authentique qui prend en compte les durées des phases sous-marines (phase de descente verticale, puis la phase de placement sous l'eau la phase de rotation autour de l'artefact identifiée, et la phase de remontée (§1.1.5.1)).

De plus, il aurait été nécessaire de simuler la mission un plus grand nombre de fois pour que les conclusions de la partie (§6) soient plus réelles. En effet, 10 simulations par scénario sont trop peu, il aurait au moins été nécessaire de simuler une centaine de fois par scénario pour avoir des conclusions exploitables.

Enfin, il est à noter que nous avons travaillé de manière collaborative via Git et Github. Vous pourrez trouver le projet également à ce lien : <https://github.com/huseyin39/UnderWaterSimulation>.