

Surveillance de plage à l'aide d'un essaim de drones autonomes

Hoffmann Alexis - Ribeiro-Pinto Adrien

28 mars 2022

Résumé

L'utilisation d'essaims de drones autonomes est un sujet au coeur de toutes les conversations depuis quelques années lorsqu'il s'agit de remplir une mission complexe, difficile, hors de portée et récurrente. De plus en plus de systèmes autonomes se sont introduit dans notre quotidien et c'est dans l'optique d'automatiser la surveillance de zones de baignade que nous avons traité le sujet suivant.

L'utilisation d'essaims de drones autonomes au bord des zones de baignade permet aux secouristes de réagir plus vite et de façon plus précise que lors des missions de sauvetage à l'heure actuelle. En effet là où une intervention nécessite la confirmation visuelle des maîtres nageurs en cas de noyade, cette confirmation peut prendre quelques secondes dans le meilleur des cas voir quelques minutes.

Avec un essaim de drones, le délai entre le moment où une personne susceptible de se noyer et l'intervention du maître nageur est minime. En effet, les drones ayant un point de vue d'ensemble de la zone de baignade peuvent prévenir les maîtres nageurs se situant au poste de secours (avec un retour caméra en direct) et le maîtres nageurs s'occupant de surveiller la baignade.

En portant une application permettant le positionnement et la gestion de(s) essaim(s) de drones sur smartphone ou tablette, nous facilitons l'utilisation de l'essaim pour les maîtres nageurs et de ce fait le sauvetage des personnes en danger.

Table des matières

0.1	Introduction	3
1	Présentation du projet	4
1.1	Introduction	4
1.2	Étude de l'existant	4
1.2.1	Helper Drone	4
1.2.2	Reconnaissance de noyade en piscine - AngelEye	4
1.2.3	Ripper Corp	5
1.3	Modèle de développement	5
1.4	Planning prévisionnel	5
1.5	Conclusion	6
2	Spécification des besoins	7
2.1	Introduction	7
2.2	User stories	7
2.2.1	Environnement de la simulation	7
2.2.2	Déploiement de l'essaim sur une zone	7
2.2.3	Adapter la disposition des drones à la configuration de l'essaim	8
2.2.4	Adaptation de l'essaim en cas de perte d'un drone	8
2.2.5	Visuel de la caméra des drones sur la zone de surveillance	8
2.2.6	Cycle jour/nuit	8
2.2.7	Station de contrôle dirigée par un humain	8
2.3	Cas d'utilisation	8
2.4	Conclusion	9
3	Conception de l'application	10
3.1	Introduction	10
3.2	Diagramme de l'application utilisateur	10
3.3	Architecture	10
3.3.1	Environnement	11
3.3.2	Actionnaires	11
3.3.3	L'essaim	11
3.3.4	UI	12
3.4	Gantt Effectif	13
3.5	Conclusion	13
4	Réalisation de l'application	14
4.1	Introduction	14
4.2	Choix logiciels argumentés	14
4.2.1	Caméras	14
4.2.2	Immersion 3ème personne	14
4.3	Réalisation	15
4.3.1	Captures	15
4.3.2	Algorithmes/méthodes/données	21
4.3.3	Résultats	21
4.4	Tests	21

4.4.1	23 Tests de l'éditeur	21
4.4.2	52 Tests du mode jeu	22
4.5	Déroulé du Projet comparaison gantt prévisionnel / effectif	22
4.5.1	Les problèmes	22
4.5.2	Les difficultés	22
4.5.3	Les possibilités écartées	22
4.5.4	Approche du drone vers un personne se noyant	23
4.6	Conclusion (Bilan/Perspectives)	23

Table des figures

1.1	Planning prévisionnel	5
3.1	Diagramme des commandes possibles par l'utilisateur	10
3.2	Architecture de l'essaim et des scripts du drone	11
3.3	États de l'essaim	12
3.4	Gantt Effectif	13
4.1	Menu de l'application	15
4.2	Centre de secours	15
4.3	Point de vue de la caméra du drone n°2	16
4.4	Point de vue de la caméra du poste de secours	16
4.5	Crash du drone n°3	17
4.6	Essaim de drones en cours de repositionnement	17
4.7	Positionnement en ligne	18
4.8	Plan en survol	18
4.9	Position autour de la zone	19
4.10	Positionnement en Arc	19
4.11	Menu latéral	20
4.12	Situation de noyade	20
4.13	Essaim en attente de déploiement	21
14	Architecture des tests avec les fichiers assembleurs spécifiques	24

0.1 Introduction

Chaque année le nombre de personnes victimes de noyades accidentelles, survenues le long des côtes du territoire français, représente environ 1000 décès soit environ 25% des noyades recensées. En 2021, entre le 1er Juin et le 31 Août, d'après une enquête recensant le nombre de noyades en France, sur environ 2700 noyades accidentelles, environ 630 ont été suivies d'un décès(23%).

Dans une optique d'améliorer les moyens de surveillance des sites de baignade, nous nous sommes demandés comment nous pouvions réduire à l'aide de systèmes autonomes le nombre de noyades accidentelles en France.

La solution vers laquelle nous nous sommes tournés est l'utilisation d'essaims de drones supervisés par les maîtres nageurs et le personnel de la sécurité civile.

Nous nous démarquons des autres solutions similaires qui utilisent le déploiement d'un drone par une reconnaissance et le support aérien des sites de baignade avec essaims de drones autonomes. En effet, l'utilisation d'essaims de drones autonomes permet de surveiller une zone plus large tout en ayant un appui visuel permanent sur la situation.

Ce rapport détaillera les différentes phases par lesquelles nous sommes passés par afin d'aboutir à une solution fiable et satisfaisante. Pour cela le rapport définit le travail que nous avons effectué, il est composé de quatre parties.

Dans un premier temps nous allons vous présenter le projet, quels sont les projets similaires et notre méthode de développement. Ensuite, dans un deuxième temps, nous vous parlerons des spécifications des besoins de notre projet et nous détaillerons les cas d'utilisation que nous avons développés. Puis nous détaillerons les étapes de la conception de l'application

Chapitre 1

Présentation du projet

1.1 Introduction

Dans ce chapitre, nous allons présenter tout d'abord l'étude de l'existant des travaux qui ont été réalisé sur la surveillance de baignades pour prévenir les noyades. Puis nous présenterons le modèle de développement que nous avons suivi pour le développement de notre projet.

1.2 Étude de l'existant

Bien que le domaine de la reconnaissance et la détection assistée par algorithmes de réseaux de neurones soit un sujet de plus en plus populaire de nos jours, son utilisation pour le secourisme en milieu aquatique reste très faible.

Nous allons vous présenter dans cette partie, trois projets qui nous ont le plus marqué dans le domaine du secourisme en cas de noyades.

1.2.1 Helper Drone

Premièrement, le projet "Helper Drone" est un projet français créé par une équipe constituée de Fabien Farge, David Gavend et d'Anthony Gavend dans les Landes. Ils ont monté ce projet afin de réduire le nombre de morts par noyade le long des côtes chaque année. Leur idée était de ne pas intervenir depuis la mer mais plutôt depuis les airs au moyen d'un drone pour assister le sauvetage.

Le drone "Helper" peut embarquer avec lui une bouée de sauvetage gonflable, un kit de secours et d'autres kits utilisables dans d'autres cas de figures plus spécifiques. Le drone dispose d'une balise GPS afin de se repérer et de communiquer sa position au poste de contrôle à proximité. Les missions de sauvetage qui lui sont attribuées sont réglables depuis une application mobile dont le secouriste dispose.

Le "Helper" a déjà été utilisé pour des interventions en bord de mer et se retrouve aussi utilisé sur des complexes pétroliers. Les conditions aux larges des côtes rendant les interventions de secours plus difficile qu'en bord de mer, son utilisation est d'autant plus appréciée.

1.2.2 Reconnaissance de noyade en piscine - AngelEye

Deuxièmement, le projet "AngelEye" est un projet d'assistance à la surveillance des baignades dans les piscines. Le problème que rencontrent les maîtres nageurs lorsqu'il s'agit de secourir une personne de la noyade est celui d'un manque d'informations.

Une victime de noyade se trouvant sous l'eau peut parfois ne pas être visible par le maître nageur se trouvant à quelques mètres de celle-ci. Voilà pourquoi l'utilisation de plusieurs caméras aquatiques ou terrestres permettent de réduire le risque de pertes d'informations et ainsi d'augmenter le nombre de personnes secourues d'une noyade.

C'est dans le but de fournir plus d'informations aux maîtres nageurs surveillant les piscines que le projet "AngelEye" a vu le jour. "AngelEye" est un algorithme basé sur un réseau de neurones. Il traite

les images de dizaines de caméras positionnées dans les piscines et informe le maître nageur en cas de noyade détectée.

1.2.3 Ripper Corp

Finalement, le Ripper Corp est une organisation australienne qui a mis au point un drone capable d'accomplir différentes missions telles que l'apport de bouées de sauvetage en cas de noyade en bord de mer. Mais aussi des missions à longue distance pour une reconnaissance et une surveillance de sites dangereux.

A défaut d'être complètement autonome, le drone de "Ripper Cost" nécessite un pilotage à distance, pour autant il n'en perd pas moins les fonctions essentielles à son succès qui sont un apport visuel et une reconnaissance de cas de noyades.

Ayant déjà fait ses preuves depuis quelques années, la société "Ripper Cost" a voulu élargir l'horizon de ses missions en utilisant le modèle de drone déjà établi pour les missions de sauvetage en mer. L'une de ses dernières missions publiques a été la reconnaissance et le sauvetage de koalas afin de les protéger, que ce soit de maladies, prédateurs ou de feux de forêts qui touchent l'Australie chaque année.

L'utilisation de différentes caméras telle que la caméra infrarouge permet de repérer les koalas dans les arbres mais elle peut aussi permettre une reconnaissance de nuit en mer. De façon générale, la caméra standard permet de réaliser la plupart des missions.

De futures missions sont en voie de développement afin de gérer de multiples drones simultanément, le point de vue de l'essaim n'est pas écarté mais cela reste encore à être confirmé.

1.3 Modèle de développement

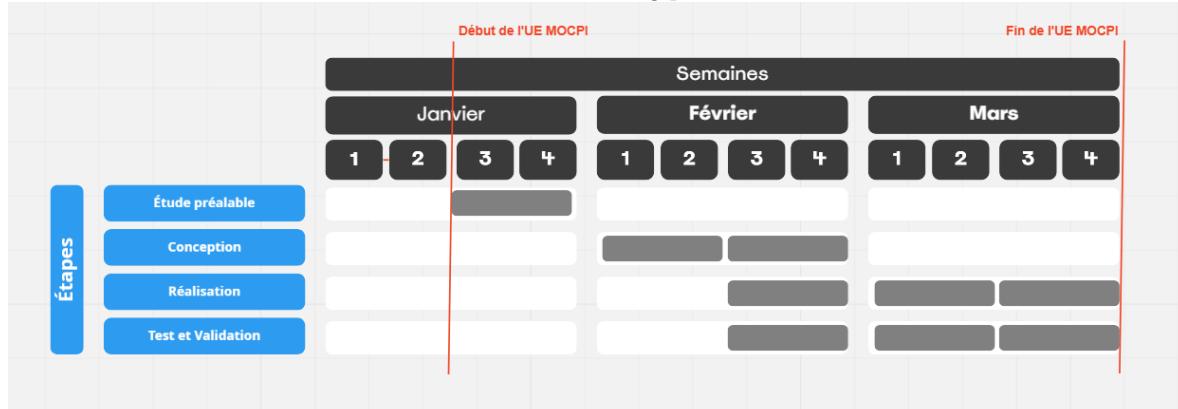
Afin de procéder au développement de notre application de surveillance de baignades par des essaims de drones, nous avons décidé d'utiliser le modèle "Scrum" qui nous paraissait plus adapté à sa réalisation.

En effet, étant donné que nous étions deux pour la réalisation du projet, nous avons préféré utiliser une méthode où la responsabilité de la planification et de la réalisation des différentes tâches, soit la responsabilité de l'équipe. De plus, au vu des multiples changements potentiels et des objectifs qui pouvaient être écartés au cours du projet, le modèle Scrum correspondait plus à nos attentes que le cycle en V par exemple. Nous avons aussi écarté le modèle "Extreme Programming" vu en cours afin d'éviter de changer de casquette (client, chef de projet, développeur, etc) car comme dit précédemment nous n'étions pas assez pour pouvoir l'utiliser.

1.4 Planning prévisionnel

Le planning prévisionnel qui suit est le planning que nous avons essayé de respecter lors de la réalisation du projet. Il indique la répartition et le déroulement de la réalisation du projet.

FIGURE 1.1 – Planning prévisionnel



1.5 Conclusion

Nous avons, à travers ce chapitre, présenté le projet ainsi que son développement. Nous avons analysé les différentes solutions existantes qui ont vu le jour depuis une dizaine d'années dans le domaine de la surveillance de noyades afin d'établir un plan prévisionnel des tâches réalisables.

Ainsi, nous nous sommes positionnés parmi les projets existants internationalement, ceci va nous permettre d'aborder la prochaine étape sur la spécification des besoins.

Chapitre 2

Spécification des besoins

2.1 Introduction

Dans ce chapitre, nous allons aborder les différents besoins de l'application de surveillance de baignades avec essaims de drones. Étant donné que nous avons conçu cette application en utilisant le modèle "Scrum" comme vu dans le chapitre précédent, nous allons vous représenter les besoins de l'application sous formes de "User Stories". Et pour finir nous aborderons les cas d'utilisation de l'application en situation réelle.

2.2 User stories

Voici les "User Stories" que nous avons réalisé en partant de la plus importante en allant à la moins importante au vu du fonctionnement et de l'objectif de surveillance de la baignade.

2.2.1 Environnement de la simulation

Pour cette "User Story", nous devions modéliser une plage au relief simple, une plateforme en mer permettant le stockage et le déploiement des drones et pour finir une mer plate.

Nous avons modélisé une plage statique lors de notre première itération et l'objectif, à terme, était de générer un environnement de façon procédurale. Voilà pourquoi nous avons modélisé un menu principal dans lequel nous avions pour projet de choisir la configuration de la plage à générer.

2.2.2 Déploiement de l'essaim sur une zone

Comme son nom l'indique, l'objectif de cette "User Story" est de déployer un essaim de drones le long d'une zone. Devant l'ampleur de cette user story, nous avons du la découper en de nombreuses tâches plus spécifiques afin de réaliser cet objectif.

En effet, pour parler d'essaim, il faut d'abord qu'il y ait un drone. Il fallait accomplir la tâche "Modélisation d'un drone (physique, modèle 3D)". Pour cela, nous avons fait la modélisation d'un drone (dji mini 2) au niveau visuel en modélisant son apparence physique mais aussi au niveau de son interaction physique en prenant en compte la physique du drone (ses mouvements dans l'espace).

Une autre tâche cruciale est la suivante : "Script de services d'un drone sur sa manipulation". Cette tâche donne pour objectif au drone "Va à telle position", il fallait donc gérer la position relative du drone dans l'espace afin de remplir la mission.

Une fois la gestion du drone implémentée, nous avons pu commencer à parler d'essaim avec cette autre tâche : "Modélisation d'un essaim de drone décollant". Pour remplir cet objectif nous avons donné un "état" au drone afin de gérer les différents cas de figure dans lesquels l'essaim pouvait être configuré.

Une fois les différentes tâches et le squelette du script de l'essaim en place, les autres user stories écrites en début de projet, se sont donc déroulées de manière intuitives.

2.2.3 Adapter la disposition des drones à la configuration de l'essaim

Cette "User Story" s'occupe de gérer la disposition de l'essaim suivant le souhait de l'utilisateur. Dans l'application, nous avons implémenté deux dispositions, une en ligne et une en arc de cercle. L'objectif final est d'ajouter une disposition permettant d'adapter le positionnement de l'essaim suivant la disposition de la plage (en bord de mer, près d'un barrage, etc).

2.2.4 Adaptation de l'essaim en cas de perte d'un drone

L'essaim, lors de la surveillance de la baignade, doit dans ce cas de figure s'adapter à la zone à couvrir lors du crash d'un drone. C'est à dire que les drones restant se partagent la zone à couvrir en se replaçant dans l'espace le long de la zone de baignade.

2.2.5 Visuel de la caméra des drones sur la zone de surveillance

Afin d'avoir un appui visuel et une reconnaissance de potentielles noyades, nous avons modélisé des caméras sur les drones qui s'occupent de la surveillance. De plus, une fonctionnalité importante de l'application de surveillance est d'avoir un retour visuel par le maître nageur sur la plage ainsi que dans le poste de secours. Nous avons donc un poste de contrôle avec une multitude d'écrans ayant un retour visuel sur chaque drone, ainsi que la possibilité à partir de l'application du maître nageur de changer de point de vue sur le drone voulu.

2.2.6 Cycle jour/nuit

Afin de gérer le déploiement de l'essaim suivant des horaires de baignades réglementées par les maîtres nageurs, nous avons implémenté un cycle jour/nuit. Cette tâche avait été pensée afin d'empêcher le déploiement de l'essaim de nuit (car manque de visibilité) mais la possibilité de rajouter une détection nocturne n'avait pas complètement été écarté.

2.2.7 Station de contrôle dirigée par un humain

Un des objectifs que nous avons abordé sous plusieurs angles est celui de la gestion des missions des drones.

Nous avons voulu permettre aux maîtres nageurs situés plus en retrait au poste de secours de superviser et d'apporter leur aide si nécessaire au maître nageur se trouvant sur la plage lors de la surveillance de la baignade.

Pour cela nous avons créé un poste de secours avec différents écrans représentant le retour vidéo des caméras des différents drones opérationnels de l'essaim. Nous avons aussi implémenté un menu afin de permettre la navigation entre les différents points de vue positionnés dans la scène (caméras fixes ou dynamique, cf. drones).

2.3 Cas d'utilisation

Le cas d'utilisation dont nous avions pensé est le suivant :

Les maîtres nageurs en arrivant sur la plage le matin à l'ouverture de la baignade déplient l'essaim de drone autour de la zone de baignade qu'ils ont défini au préalable. L'essaim va se positionner aux coordonnées de la zone à surveiller dans la disposition définie ultérieurement. Lors d'une potentielle noyade, le drone ou le poste de secours alerte le maître nageur s'il n'a pas remarqué la noyade. Il décide s'il y a besoin d'intervenir ou pas en passant sur le visuel de la caméra du drone ayant repéré la noyade.

Une fois l'identification faite, il peut intervenir ou faire intervenir une personne de son équipe pour le sauvetage de la victime.

À la fin de la période de surveillance de la baignade, le maître nageur ordonne à l'essaim de stopper toutes les missions qu'il réalisait et lui ordonne de retourner à la base (bateau pour recharge).

2.4 Conclusion

Nous avons vu dans ce chapitre la liste des tâches ("User Stories") que nous avons mis en oeuvre dans notre projet. Puis nous avons vu un cas d'utilisation de l'application de surveillance de noyade. Notez que cette application peut être portée sur d'autres situations similaires de surveillance (en foret, au bord d'un lac, dans un parc d'attraction, etc).

Maintenant que ces objectifs ont été vu, nous allons voir la conception de l'application et comment nous avons réalisé ces objectifs.

Chapitre 3

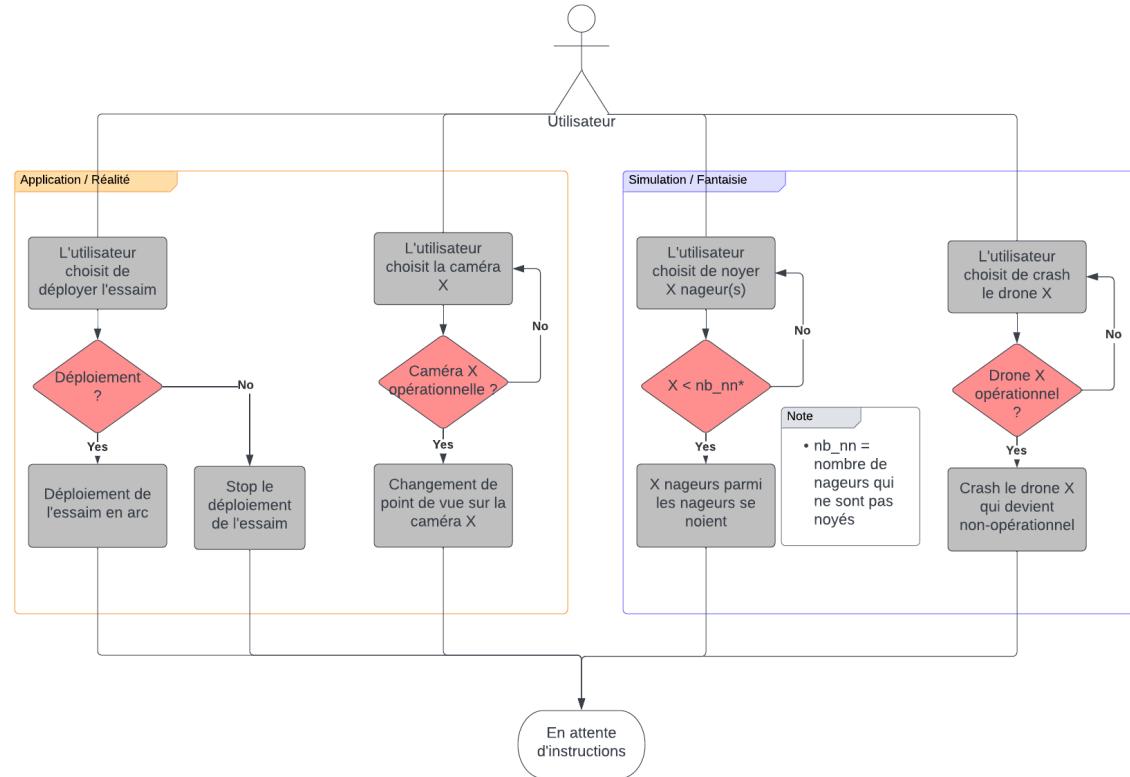
Conception de l'application

3.1 Introduction

Dans ce chapitre, nous allons aborder les méthodes de conception que nous avons mis en oeuvre pour la réalisation de notre projet.

3.2 Diagramme de l'application utilisateur

FIGURE 3.1 – Diagramme des commandes possibles par l'utilisateur



3.3 Architecture

Dans cette partie nous traiterons de l'architecture que nous avons mis en place dans le projet pour réaliser la simulation de surveillance de baignade.

3.3.1 Environnement

Pour ce qui est de l'environnement dans lequel la simulation prend place, nous avons choisi une plage en bord de mer puisque c'est l'utilisation qui nous paraissait la plus adaptée à la mission d'assistance de surveillance de baignades. De plus, Bordeaux étant géographiquement proche de l'océan, cela nous paraissait plus adapté à une audience sur la protection des zones de baignade.

3.3.2 Actionnaires

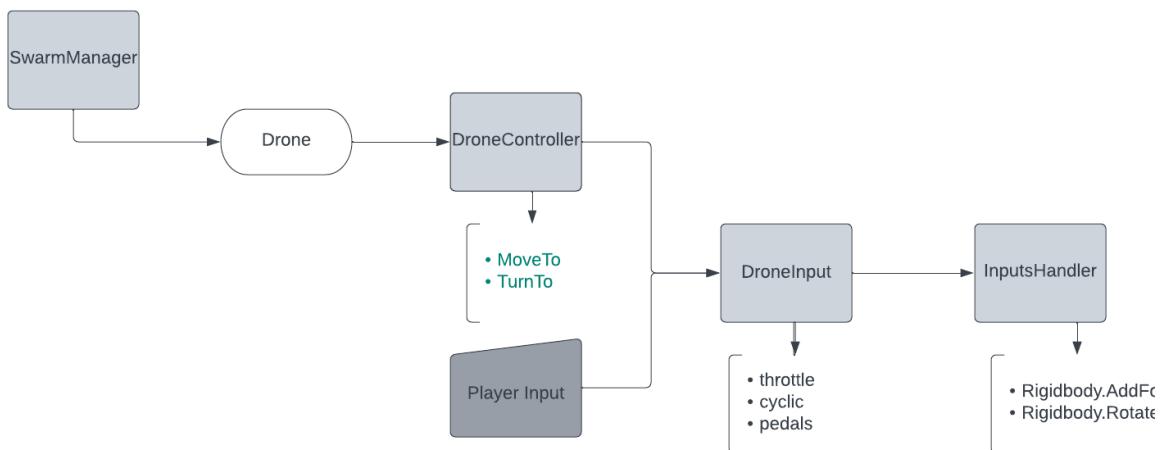
Dans la partie des actionnaires, nous avons regroupé tous les objets de la scène qui agissent sur l'environnement de simulation. Nous y retrouvons donc le maître nageur qui est dirigé par le joueur dans la simulation, mais aussi le bateau qui s'occupe de gérer les différents essaims de drones.

Pour ce qui est du maître nageur, un script écoutant les messages envoyés par le clavier du joueur permet à celui-ci de se déplacer dans l'espace afin d'avoir une immersion 3D de la simulation.

3.3.3 L'essaim

Dans cette partie, nous parlerons de l'architecture des scripts associés aux drones, dans leur contrôle et dans la gestion de l'essaim avec ses différents états.

FIGURE 3.2 – Architecture de l'essaim et des scripts du drone



Le drone

Pour ce qui est de la physique du drone. On a sur le prefab du Drone, un composant nommé **PlayerInput** qui prend en paramètre une carte d'action de la part d'un utilisateur.

Le script "DroneInputs" est celui qui est chargé de prendre en charge les valeurs données et de les associer à des variables d'accélérateur (throttle), de pédaux (pedals) et de cycle (cyclic).

Le script "InputHandler" récupère ces valeurs de "DroneInputs" et les applique dans ses opérations d'ajout de force et de rotation au "Rigidbody".

Le script "DroneController" vient donc quand à lui proposer une manipulation en terme de position à atteindre au drone en modifiant directement les variables de "DroneInputs". **MoveTo** associe une position à la variable position à atteindre. L'Update du script cherche à toujours atteindre cette dernière.

Le gestionnaire de l'essaim

Concernant maintenant l'essaim et ses états. Ils sont huit en anglais :

1. SpawningDrones
2. Standby

3. TakeOff
4. OnTheWayIn
5. Monitoring
6. OnTheWayBack
7. Landing
8. Repositioning

Lorsque la simulation commence, le "SwarmManager" débute avec l'état "SpawningDrones". Lors que la première trame, cinq drones sont donc instanciés sur la plateforme.

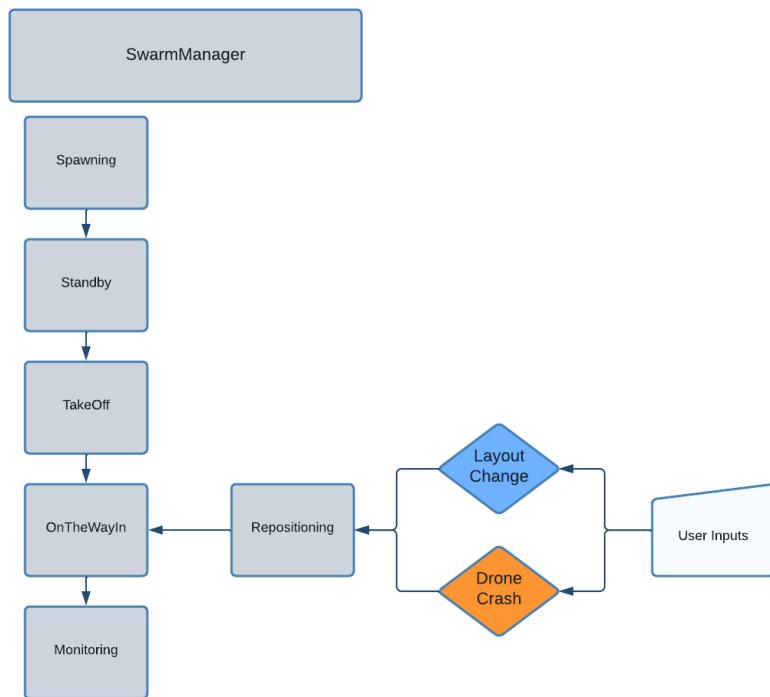
Une fois les apparitions faites, l'essaim se place en veille ("Standby") en attente d'activation de la part du maître nageur. Puis, une fois activé, l'essaim décolle ("TakeOff"). Il calcule la position à atteindre et ordonne aux drones d'aller à celle-ci, et tout cela dans la première trame de l'état "OnTheWayIn".

Une fois en position, les drones se stabilisent et se placent dans l'état de surveillance ("Monitoring"). L'essaim vient alors ordonner dans la première trame de cet état, de placer la caméra en direction des nageurs.

Maintenant en cas de changement de disposition ou de perte de drone, l'essaim se place dans l'état de repositionnement ("Repositioning"). Une vérification des drones toujours réactifs est faite et les drones ne répondant pas à ce contrôle sont retirés de la liste des drones.

Les états de retour et d'atterrissement ("OnTheWayBack" et "Landing) n'ont pas été implémentés par manque de temps.

FIGURE 3.3 – États de l'essaim



3.3.4 UI

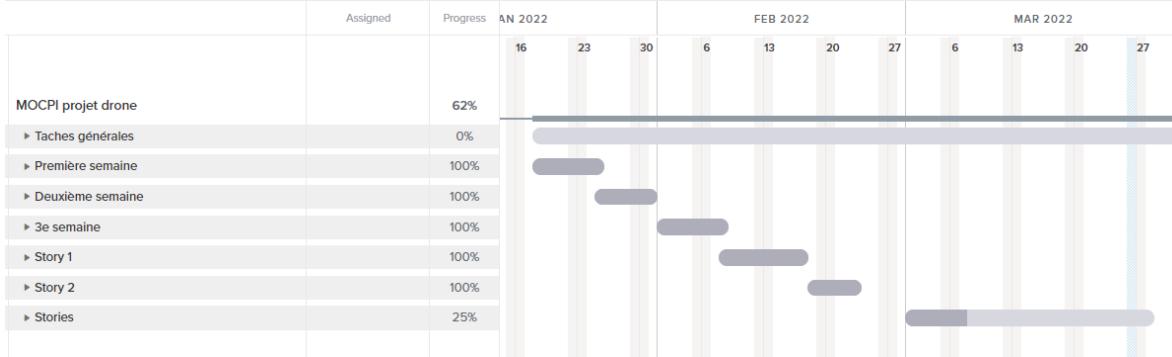
Dans la partie UI de la scène, nous avons divisé cette section en 2 parties. La première étant le menu de pause de la scène qui comme son nom l'indique met en pause la scène. Elle affiche un bouton pour continuer, un pour revenir au menu et un dernier pour quitter l'application.

Dans la deuxième partie, appelée "SideMenu", nous avons le menu permettant de contrôler l'essaim. La partie supérieure permet de contrôler le déploiement ou le retrait de la mission de surveillance de la zone de baignade au moyen d'un bouton "Start" et d'un bouton "Stop". La partie centrale permet

de spécifier le nombre de nageurs dans la zone de baignade que l'on veut noyer (dans un but purement de démonstration bien entendu), le "slider" permet de choisir ce nombre et la détection de personnes se noyant s'affiche au dessus. La partie inférieure du menu permet de contrôler la caméra des différents drones ainsi que la possibilité de crash le drone choisi (de même que pour les noyades ce n'est qu'à titre de démonstration). Un indicateur visuel permet de différencier les drones opérationnels de ceux qui se sont crashés ou ne sont plus opérationnels.

3.4 Gantt Effectif

FIGURE 3.4 – Gantt Effectif



3.5 Conclusion

Ce chapitre a été consacré à différentes étapes de la conception détaillée de l'application de surveillance de baignade. En partant du diagramme de l'application utilisateur, regroupant les différentes possibilités données au maître nageur sur le déploiement de l'essaim. En passant par l'architecture de l'application (essaim, drone, etc) pour finalement traiter la partie interface utilisateur.

Le prochain chapitre contiendra des points techniques, que nous avons trouvé important de noter, à la réalisation de l'application ainsi que les difficultés que nous avons rencontré.

Chapitre 4

Réalisation de l'application

4.1 Introduction

Dans ce chapitre, nous traiterons des choix logiciels que nous avons fait pour la réalisation de l'application. Nous verrons les objectifs que nous avons réussi à réaliser ainsi que les points que nous avons écarté par manque de temps ou par problèmes de conception. Nous aborderons les tests réalisés et les résultats de notre implémentation.

4.2 Choix logiciels argumentés

Nous avons utilisé Unity afin de réaliser ce projet car Unity est un logiciel de conception 3D nous permettant de réaliser la simulation du déploiement de drones le long d'une plage. En effet n'ayant pas la chance d'avoir plusieurs drones à notre disposition afin de pouvoir tester le déploiement et la surveillance en situation réelle, nous avons opté pour une simulation logicielle 3D.

4.2.1 Caméras

Étant donné que les drones de nos jours possèdent tous une multitude de caméras que ce soit pour faire de la reconnaissance vidéo ou pour se repérer dans l'espace. Nous avons modélisé des drones possédant une caméra pour le retour vidéo et une "cinemachinevirtualcamera" permettant de faire des changements de points de vue entre les différents drones. L'utilisation des "cinemachinevirtualcamera" est utilisée pour faire des changements de points de vue dans les scènes, comme par exemple pour les montages de scènes d'action dans les films. Il y a une caméra à proprement parlé dans la scène qui se déplace entre différents points de vue dans la scène.

4.2.2 Immersion 3ème personne

Pour le choix de l'immersion en situation réelle, nous avons fait le choix de modéliser un maître nageur contrôlé par le joueur. Le but étant de permettre au "joueur" de contrôler l'application de déploiement des essaims de drones. De ce fait, Unity nous a permis de modéliser une interface utilisateur dans la scène permettant au maître nageur de contrôler le déploiement de l'essaim ainsi que les différents points de vue des drones.

4.3 Réalisation

4.3.1 Captures

FIGURE 4.1 – Menu de l'application

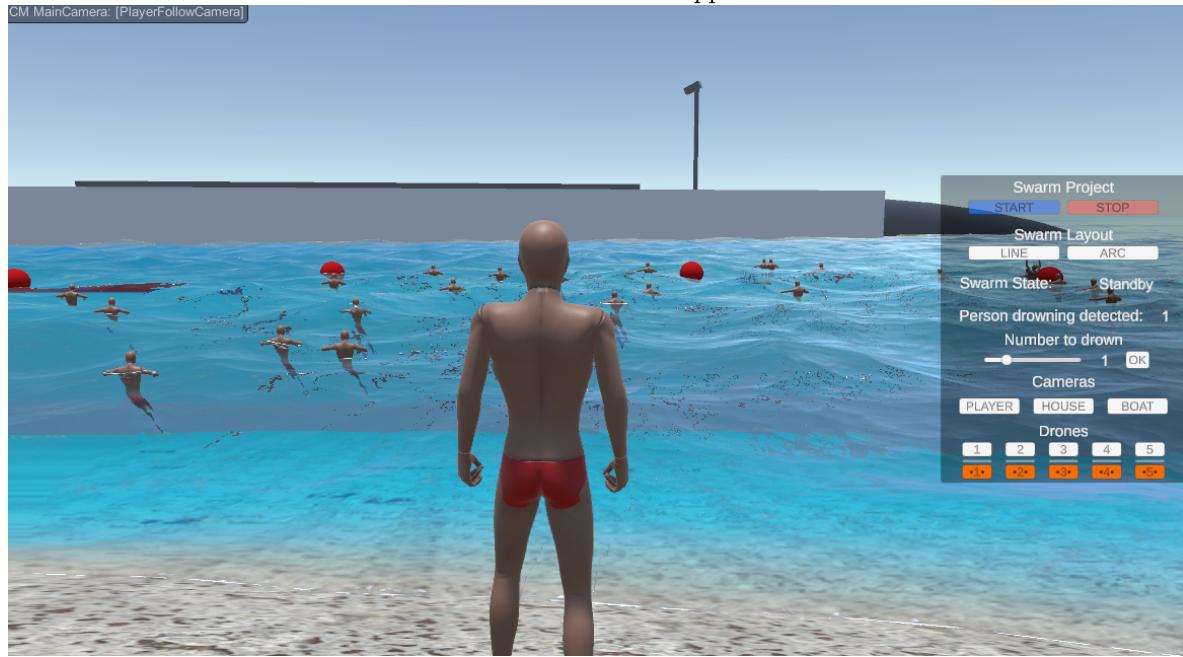


FIGURE 4.2 – Centre de secours

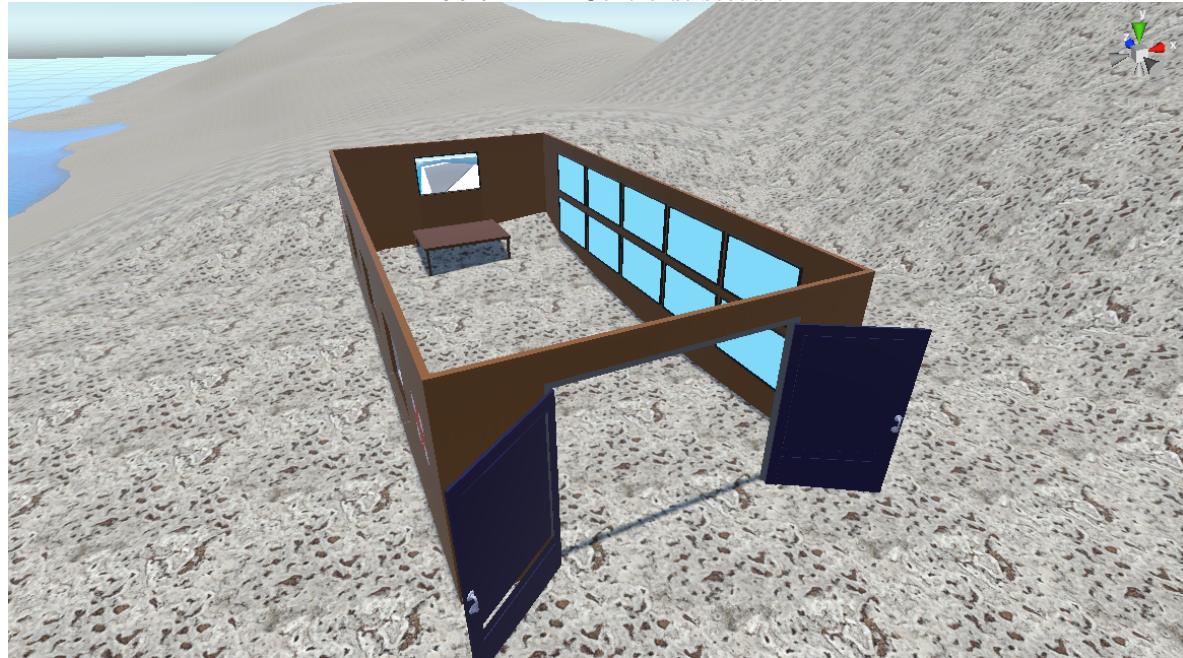


FIGURE 4.3 – Point de vue de la caméra du drone n°2



FIGURE 4.4 – Point de vue de la caméra du poste de secours

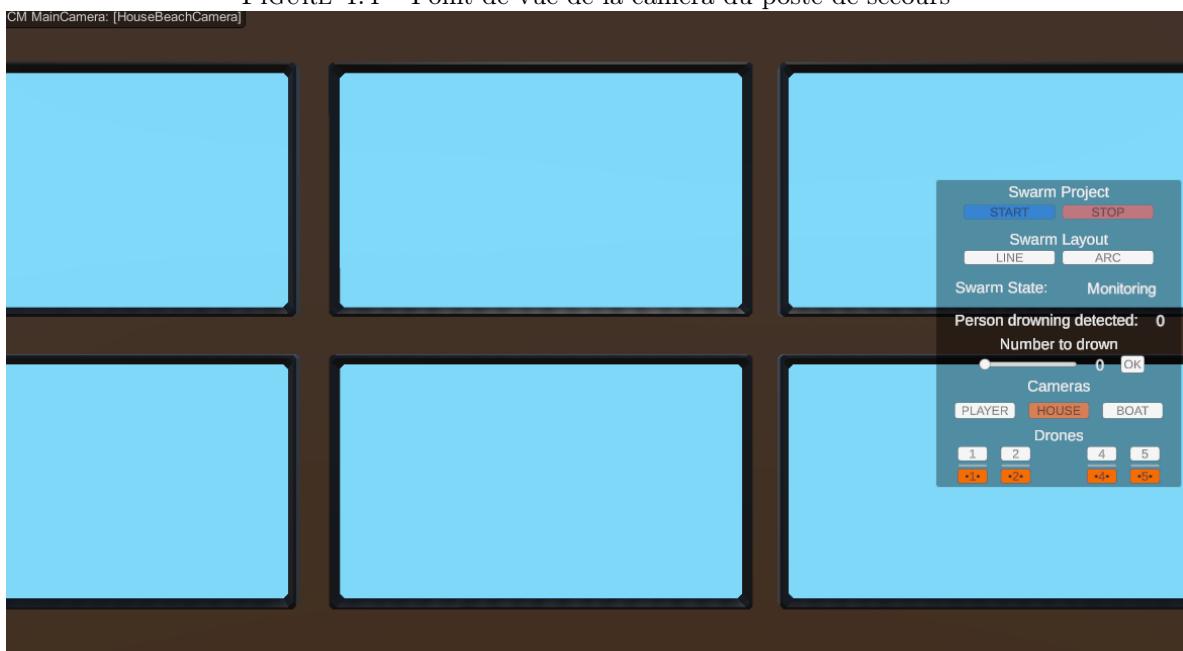


FIGURE 4.5 – Crash du drone n°3

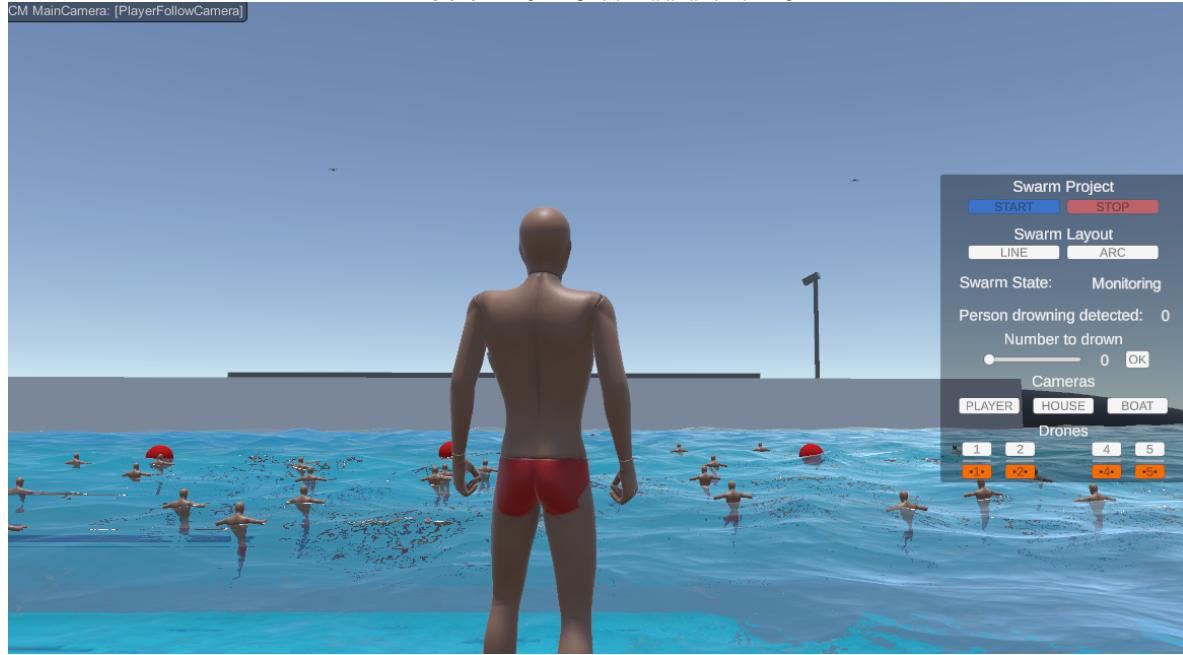


FIGURE 4.6 – Essaim de drones en cours de repositionnement

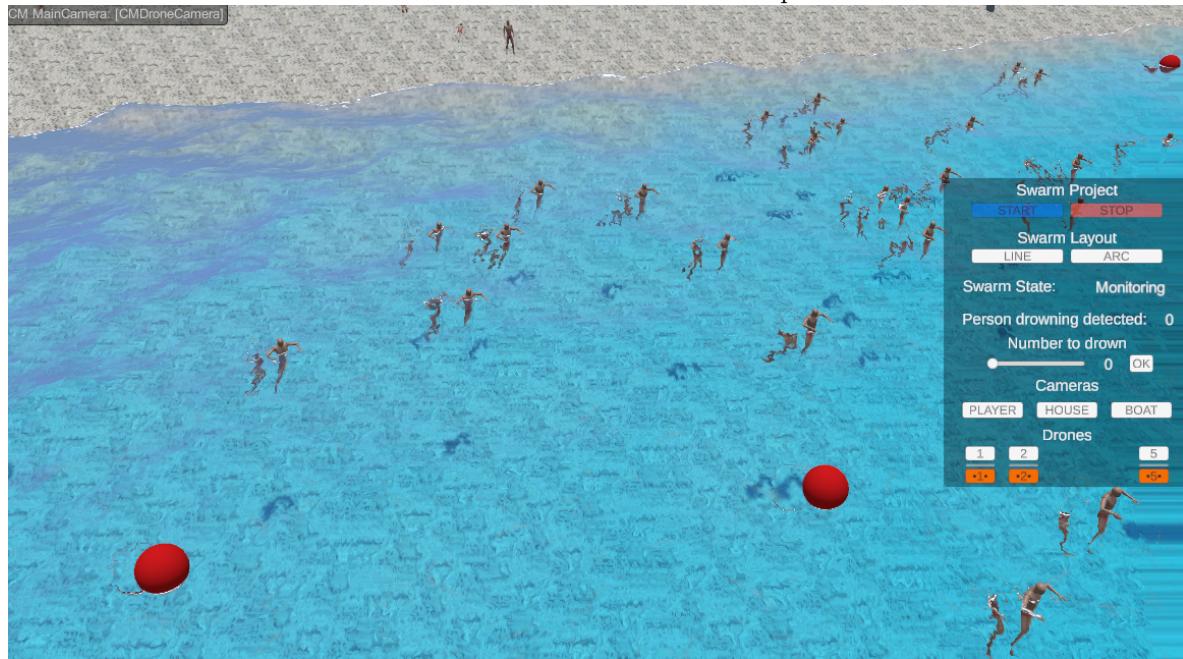


FIGURE 4.7 – Positionnement en ligne

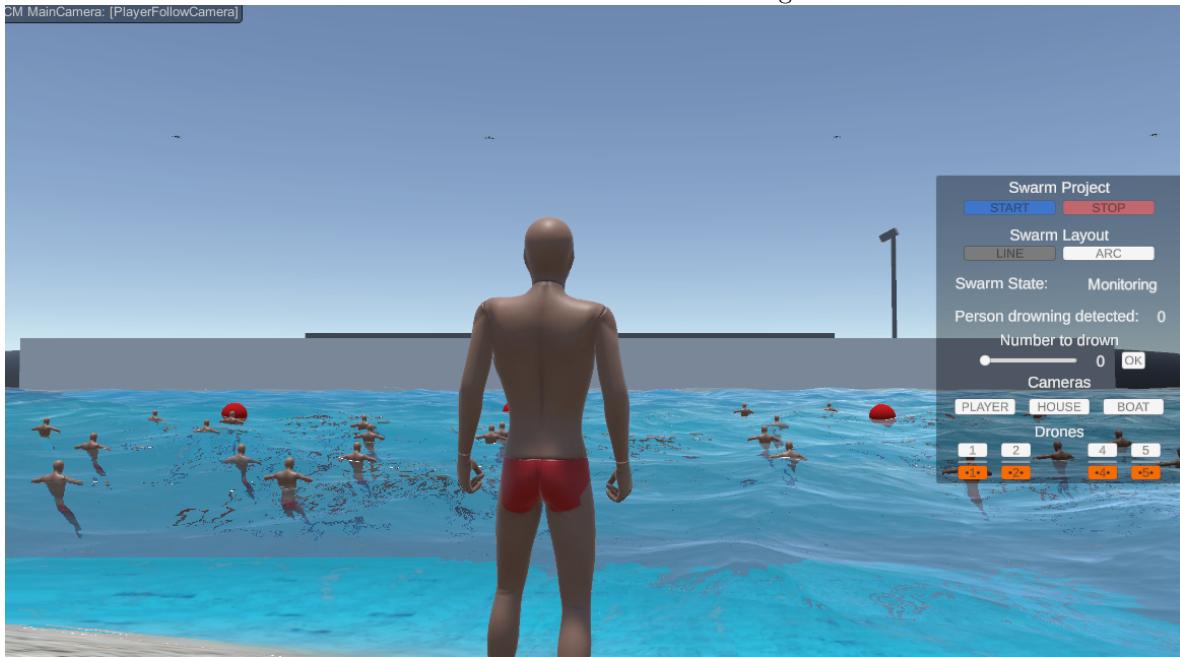


FIGURE 4.8 – Plan en survol

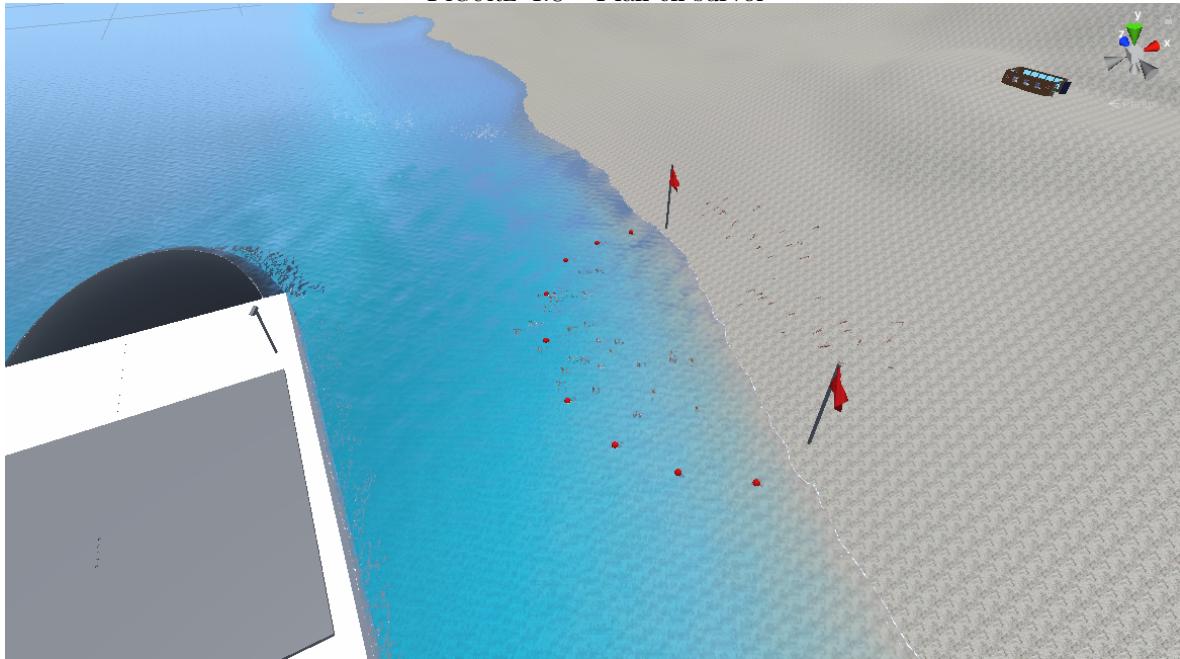


FIGURE 4.9 – Position autour de la zone

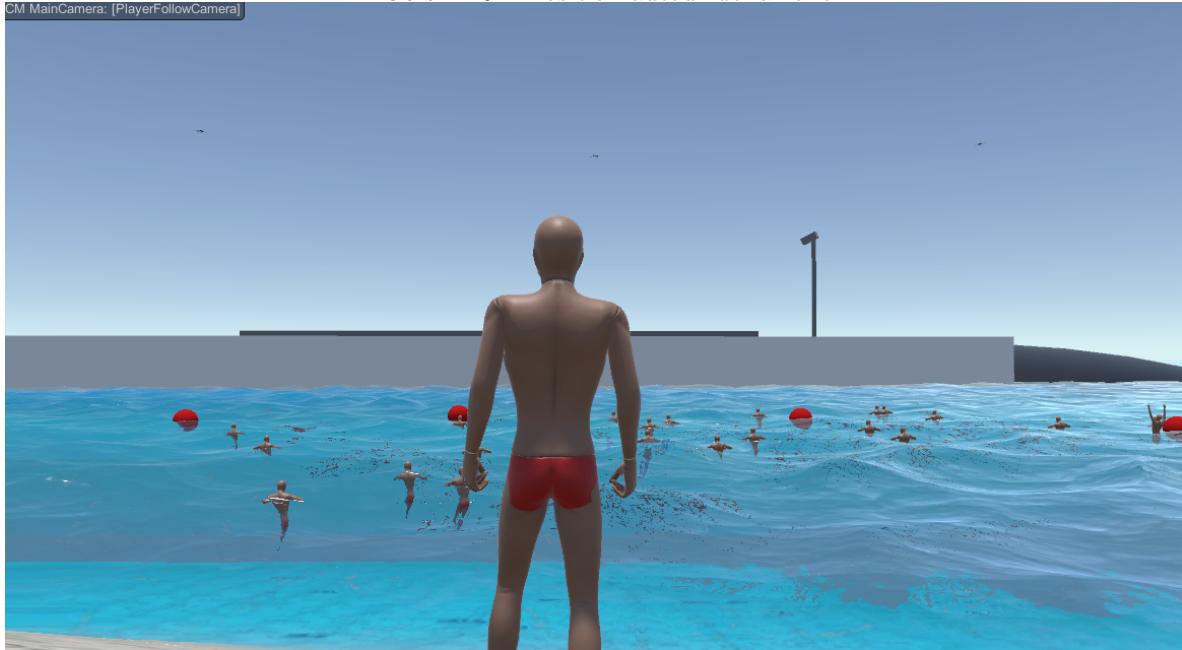


FIGURE 4.10 – Positionnement en Arc

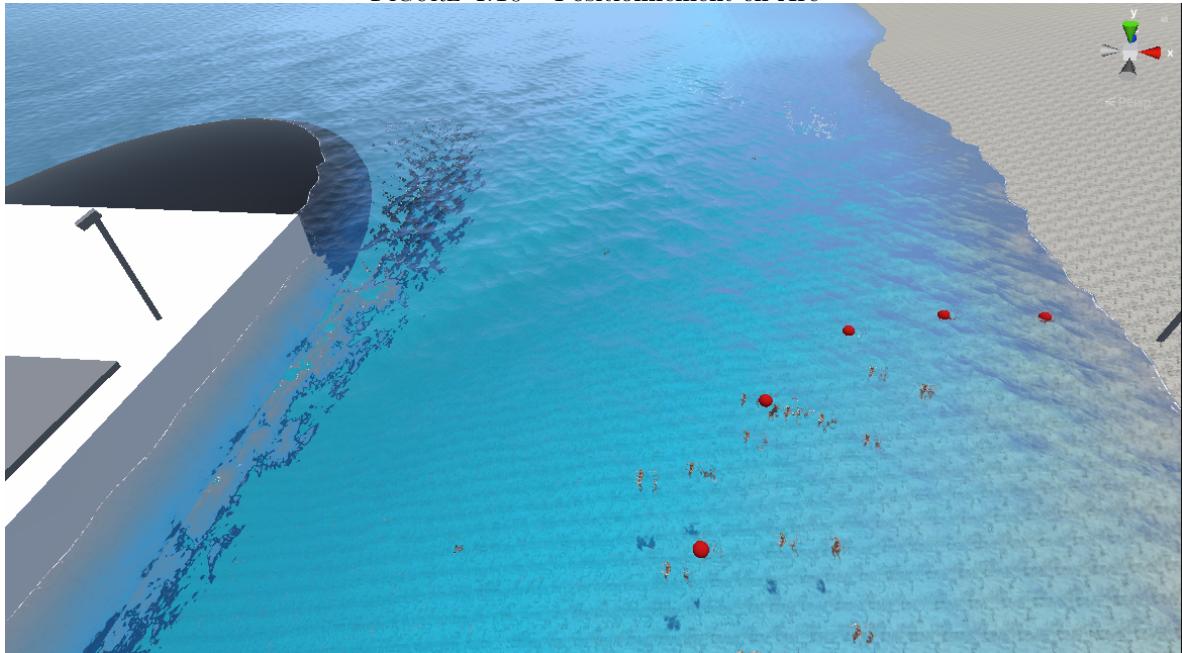


FIGURE 4.11 – Menu latéral

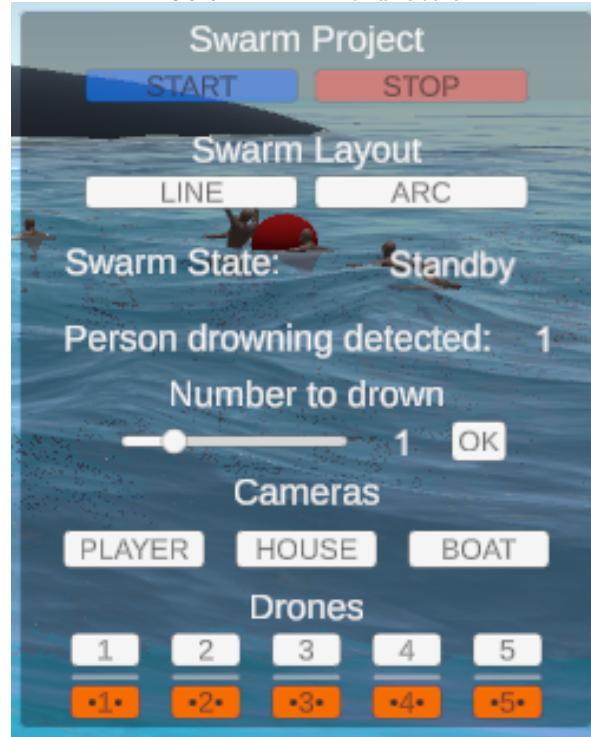


FIGURE 4.12 – Situation de noyade

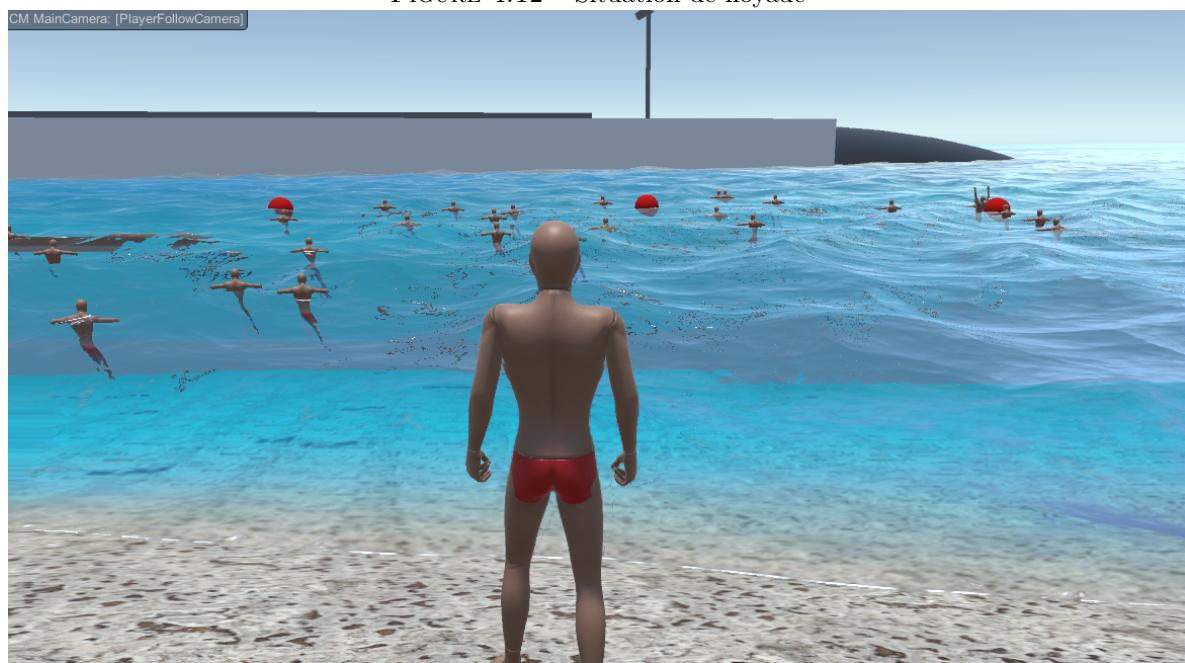
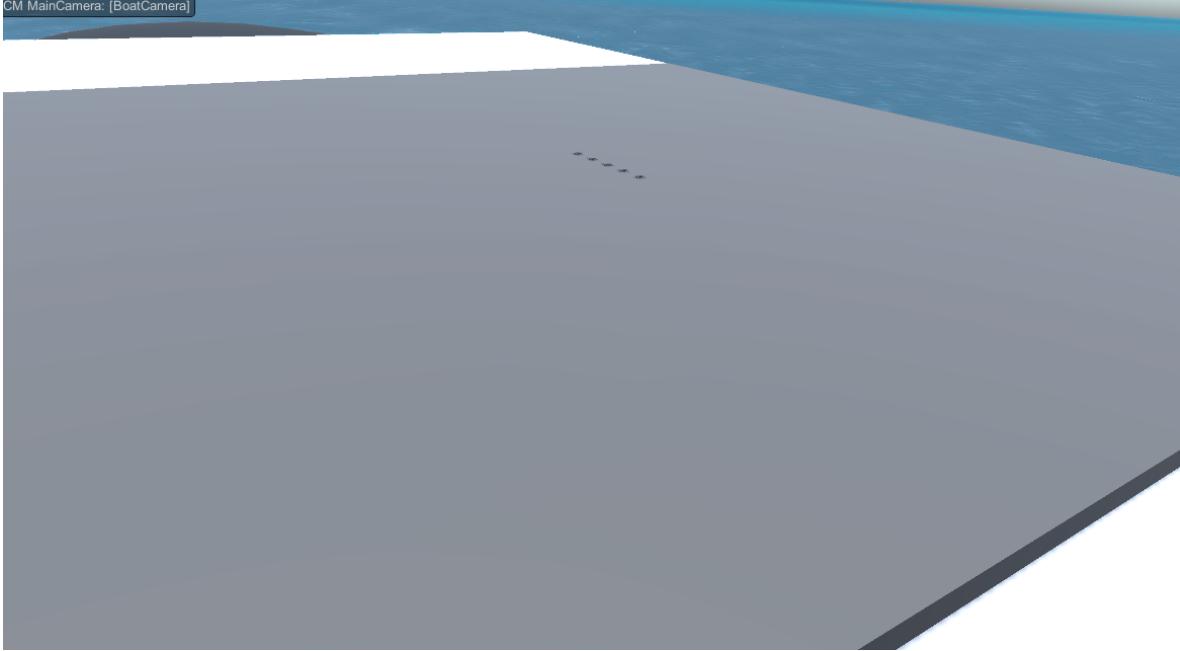


FIGURE 4.13 – Essaim en attente de déploiement



4.3.2 Algorithmes/méthodes/données

Pour le calcul de disposition en ligne, il y a les nageurs qui font office de point cible à atteindre pour un drone de l'essaim. Pour P la position global des nageurs, p_c la position cible, N le nombre total de drone dans l'essaim, r le rang du drone dans l'essaim, d la distance et la hauteur de sécurité en la position à atteindre et les nageurs et L la taille de longueur de la zone.

$$p_c = \begin{pmatrix} P_x - d_x \\ P_y + d_y \\ P_z - \frac{L}{2} + L \frac{r}{N+1} \end{pmatrix}$$

Quand au calcul de disposition en arc, les variables restent les mêmes mais le calcul est différent :

$$p_c = \begin{pmatrix} P_x + d_x \sin \alpha \\ P_y + d_y \\ P_z + \frac{L}{2} \cos \alpha \end{pmatrix}, \alpha = \pi + \frac{r\pi}{N+1}$$

4.3.3 Résultats

4.4 Tests

Au début du projet, il fallait mettre en place une véritable architecture interne à Unity. Pour tester le calcul d'une méthode, on peut comprendre l'approche d'un test classique du test unitaire d'un calcul par exemple.

Mais pour tester qu'un drone ait parcouru un chemin de A vers B en 2 secondes, Unity propose un framework de tests, permettant de lancer le test, attendre 5 trames et tester un comportement en particulier. Il y a donc 2 types de tests. Ceux de l'éditeur et ceux qui sont dans le mode de jeu tel que le jeu prendrait place sur une version finale. Pour chaque type, on a un dossier qui va contenir un fichier avec des paramètres pour l'assembleur et les fichiers de tests concernés. Voir en annexe cette architecture.

4.4.1 23 Tests de l'éditeur

Concernant l'approche de TDD, la pratique consistant à tester petit à petit s'est avérée très efficace. Dans ce type de tests, il y a eu la classe du contrôleur du drone avec notamment ceux de changement

de position à atteindre. Mais encore ceux de vérification qu'un position se trouve bien être dans un rayon de la position à atteindre. Les améliorations possibles de cette classe serait dans la méthode "GetDirection" pour pouvoir construire un modèle géométrique inverse avec la jacobienne par exemple.

Des tests ont commencé à être mis en place pour un script sur la plateforme mais dans la version du rendu du projet, il ne sera pas utilisé.

4.4.2 52 Tests du mode jeu

Les tests ont surtout porté sur la manipulation du drone et des différents aspects des états de l'essaim. Sur les tests du drone, on retrouve ceux sur le script des inputs du drone avec notamment les vérifications sur les valeurs des variables d'accélérateur, de pédaliers et de cycle avec les comportements attendus du drone.

Un exemple maintenant plus parlant serait celui du test du contrôleur du drone dans ce mode. On peut imaginer un drone qui se trouve à une position de base. On lui dit d'aller à une nouvelle position. On attend 2 secondes et on vérifie si la position attendue est la même que celle du drone.

On peut réaliser ce test sur de nombreuses positions à atteindre tout autour du drone.

Maintenant vis-à-vis de l'essaim, le développement en TDD a tout de même, il faut le reconnaître, permis de passer à côté de nombreux bogues. Si on prend le cas des tests sur le crash d'un drone, il faut vérifier que l'essaim passe dans l'état de repositionnement et que dès qu'il est dans l'état de surveillance, chaque drone doit être à leur nouvelle place calculé par rapport à leur rang dans l'essaim.

Un test aussi avancé en terme de vérification si il est bien écrit, permet de coder et de réussir à avoir un comportement qui fonctionne même sans regarder le visuel de cette opération.

Dans l'état de surveillance, un test intéressant est celui de la vérification de la direction vers laquelle la caméra regarde.

En résumé de cette expérience du TDD, on peut dire que on peut être vraiment soulagé lorsque tout nos tests passent à chaque refactorisation et surtout en fin de projet. Cette manière de développer est sans aucun doute très efficace. Le seul point négatif est que l'on s'y attache un peu trop car dès que l'on fait un petit script pour quelque chose et qu'il n'est pas rédigé en TDD on est perdu s'il y a des bogues.

4.5 Déroulé du Projet comparaison gantt prévisionnel / effectif

Le gantt prévisionnel nous a permis de bien démarrer le projet. Nous l'avons relativement bien suivi dans le premier mois de développement. Mais à partir du deuxième mois, certaines grosses tâches nous ont pris plus de temps que prévu car nous avions certaines lacunes sur Unity que nous avons comblés en passant du temps à apprendre. Résultat le décalage dans le gantt s'est creusé mais c'était positif car nous étions tout de même productif sur le moyen terme, c'est à dire jusqu'à la fin du projet.

Alors pour la globalité du projet, la détection de noyade par deep learning sur la lecture des caméras des drones n'a pas pu se faire mais pour ce qui est la scène, de l'interface et de la gestion de l'essaim. Nous avons pu faire en très grande partie ce que nous voulions faire.

4.5.1 Les problèmes

Les problèmes que nous avons rencontrés sont en premier le manque de temps.

4.5.2 Les difficultés

Unity nous a posé des soucis au niveau de l'optimisation des caméras avec le mur d'écran de surveillance. Il fallait apprendre à gérer le framework des tests. Également l'architecture du gestionnaire de l'essaim a été réellement difficile à trouver.

4.5.3 Les possibilités écartées

On aurait aimés continuer le projet sur de la surveillance de menaces marines par exemple ou bien de pouvoir apporter de l'assistance avec un bateau autonome.

Remplacement de drones déchargés/perdus

Également, on a pas pu mettre en place un échange entre drones avec différents niveaux de batteries. La plateforme devait permettre de recharger les drones.

4.5.4 Approche du drone vers un personne se noyant

Dans le cas, où l'on avait eu une détection de noyade fonctionnelle, on aurait pu faire que dès lors, le drone vient filmer de manière plus précise la personne qui se noie. Voir même que le drone s'approche d'elle et que le maître nageur puisse lui parler à travers le drone.

Cycle Jour-Nuit

Avec ce cycle, le but aurait été de les faire décoller à l'aurore et les rapatrier au crépuscule.

Cycle de marées

Certainement la possibilité écartée qui nous aurait vraiment eu la plus envie de faire. Le but était de donner une liste de coefficients de marées avec des horaires.

Un script sur la mer aurait pu modifier la hauteur de la mer en fonction de cette liste de coefficients.

Création de terrains aléatoires

Le principe pour cette création de terrains était générer des terrains de plage avec des inclinaisons de terrains différentes ou encore de faire générer des baïnes sur la plage. Avec la montée ou la descente de la mer, la zone de baignade se serait adaptée au terrain.

Affichage d'une carte satellite

Tel une minimap sur un jeu, ce système de carte aurait permis de donner une vue d'ensemble sur l'essaim mais surtout pour le maître nageur, de diriger la plateforme avec des contrôles sur l'interface.

4.6 Conclusion (Bilan/Perspectives)

Durant cette UE de MOCPI (Méthodes et Outils pour la Conduite de Projets Informatiques), nous avons eu comme objectif de trouver une solution incluant des systèmes autonomes à un problème de la vie de tous les jours. Notre travail s'est basé sur la conception d'une application de surveillance de zones de baignades assistée par essaims de drones. La réalisation de cette application a été conçue et réalisée dans un environnement 3D grâce au logiciel Unity3D.

Au cours de la phase de réalisation de notre application, nous avons cherché à recréer un environnement se rapprochant le plus possible de la réalité dans un environnement 3D. Ce point a constitué le point de départ de la réalisation de notre application. Une fois l'environnement créé, nous avons posé les bases pour les choix de spécification des besoins en isolant les objectifs principaux. Après avoir isolé les besoins principaux à réaliser, nous avons commencé à les implémenter.

Pour conclure, notre projet peut être sujet à des améliorations futures, comme par exemple l'ajout de reconnaissance de noyade par "intelligence artificielle". Bien entendu, les objectifs que nous avons écarté sont aussi sujets à de futures améliorations.

FIGURE 14 – Architecture des tests avec les fichiers assembleurs spécifiques

