

# Rapport Projet COCOMA

Abderrahmane CHAABANI - Youssouf HADHBI

December 2016

## 1 Introduction et rappel du sujet

Ce projet a été réalisé dans le cadre de l'UE CoCoMa du M2 ANDROIDE, et encadré par Aurelie Beynier, Amal El Fallah et Cedric Herpson, que nous remercions pour l'aide qu'ils nous ont apporté tout au long de sa réalisation.

Dans ce projet, on dispose d'un environnement 3D, réalisé en NetLogo, où évoluent Voitures, Drones et Ennemis. Les voitures sont organisées sous forme de **Convoi** dont l'une des voitures doit-être **protégée**, et disposent d'un **leader**. Elles démarrent d'un quartier général et cherchent à atteindre une point d'arrivée de la carte en parcourant un environnement contenant **rivières, lacs et montagnes**. Elles sont escortées par une flotte de **Drones** cherchant à les protéger des différents **Ennemis** dont la carte est jonchée. Ces drones peuvent **communiquer** entre eux, et disposent eux aussi d'un **leader** chargé de leurs **coordination et communication** avec le convoi.



*Exemple de carte typique*

Notre objectif dans ce projet est donc de modéliser cet environnement, et d'implémenter les comportements propres à chaque type d'agents présents dans l'environnement.

## 2 Modélisation de l'environnement

Il nous a été fourni dans ce projet un générateur d'environnement permettant de modéliser Rivières et Lacs, et d'obtenir une carte disposant d'un chemin du quartier général des drones au points d'arrivée. Afin de représenter les montagnes, nous avons tout d'abord défini une **taille minimum et une taille maximum** pour les montagnes. Ensuite, nous avons disposé, le nombre de montagnes désirées en sélectionnant un patch au hasard et en rajoutant une **demi-boule** dont le rayon est entre le min et le max des valeurs déterminées auparavant, en faisant en sorte que cette montagne n'empiète pas sur le Quartier Général. Afin d'éviter d'obtenir des montagnes trop lisses, chaque patch de la montagne dispose d'une **probabilité de 10% de ne pas être créé**. On vérifie alors s'il existe un chemin où non un chemin du QG à l'arrivée, et si oui, la nouvelle carte est conservée. Sinon, une nouvelle carte est générée.

## 3 Modélisation et comportement des ennemis

Chaque ennemi dispose d'**attributs** qui lui sont propre : une vitesse, un angle maximal de rotation, une portée de vue et de tir, une cadence, une précision et puissance de tir, ainsi qu'un nombre de munitions. Ces ennemis se déplacent initialement **aléatoirement** sur la carte : ils ne peuvent traverser lacs, rivières (sauf à l'aide des ponts) et montagnes, ni faire le tour de la carte par les bords.

Cependant, si ces ennemis repèrent une ou plus des voitures du convoi, ils chercheront à **se déplacer vers la plus proche**. Ces ennemis n'étant pas des troupes d'élite, si elles se trouvent heurtée à un obstacle en face d'elle, elles **ne chercheront pas** à le contourner ou l'éviter. De plus, chaque ennemi travaille **individuellement** et n'appelle pas ses camarades à l'aide. Une fois à portée de sa cible, l'ennemi **tirera jusqu'à ce que sa cible meurt**, dans ce cas il cherchera alors une prochaine cible, ou jusqu'à ce qu'il n'ai **plus de munition**, dans ce cas il continuera simplement à suivre sa cible.

Dans le cas où les ennemis ne détectent pas de voiture du convoi, mais repèrent un drone, ils se dirigeront, en suivant les mêmes règles que si ils visaient une voiture, vers le drone et tireront dessus lorsqu'ils seront à portée. Néanmoins, tirer sur une cible aérienne étant plus compliqué que tirer sur une cible terrestre, leur précision se verra **divisée par deux**. Si à un moment un ennemi visant un drone **remarque une voiture** du convoi, celle-ci étant de plus haute priorité, il **changera alors de cible**.

## 4 Modélisation et comportement des drones

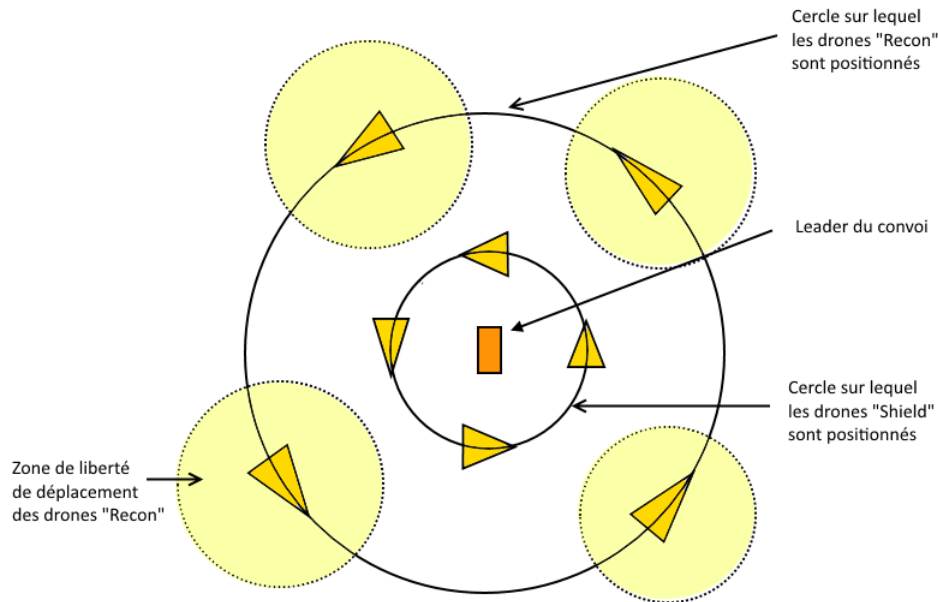
### 4.1 Organisation, rôle et placement

Similairement aux ennemis, chaque drone dispose d'**attributs** qui lui sont propre : une vitesse, un angle maximal de rotation, une portée de vue et de tir, une cadence, une précision et une puissance de tir, ainsi qu'un nombre de munitions. Ils disposent aussi une quantité de **carburant**, qu'ils consomment à chaque pas de temps. Si la quantité de carburant devient trop faible, il leur est nécessaire d'**aller se recharger** à la base. De plus, à l'inverse des ennemis qui ne peuvent plus tirer s'ils ne disposent plus de munitions, les drones **peuvent retourner à la base pour se recharger**.

Les drones se divisent en **deux équipes** de tailles égales ( $\pm 1$  drone si nombre impaire) :

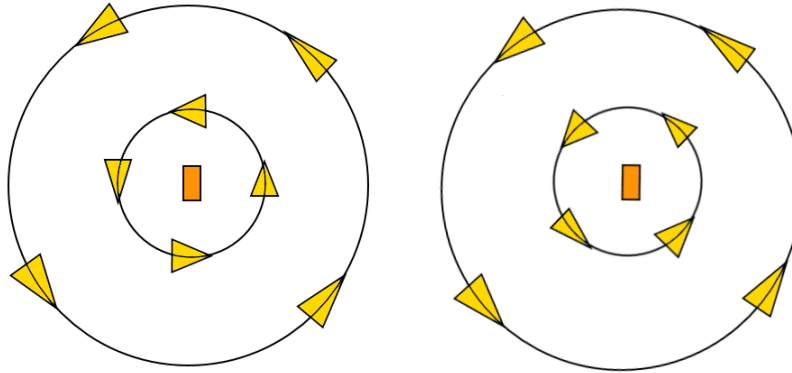
- Les drones "**shield**" : Ils restent au **plus proche** du convoi, tirent sur les ennemis qui s'approchent, mais ne les **poursuivent pas**. Leur distance au convoi ne varie pas lors de leur patrouille.
- Les drones "**recon**" : Chargés de la **reconnaissance du terrain**, ils se trouvent à une distance du convoi plus élevé. Ils disposent d'une position de repos qu'ils adoptent quand ils ne repèrent pas d'ennemis. Cependant, lorsqu'un ennemi est dans leur portée de vue, ils peuvent **se déplacer vers celui-ci** pour l'éliminer, dans la limite d'une **certaine distance** à leur position de repos.

Au sein d'une même équipe, les position de repos des drones sont répartis uniformément autour d'un cercle dont le centre est le leader du convoi et le rayon la distance au convoi de chaque équipe.



*Représentation des positions des drones autour du convoi*

Les drones de reconnaissance devant disposer d'une vitesse angulaire au moins égale à celle des drones de protection, ils sont donc **plus rapides** que ceux-ci. De plus, afin d'éviter d'avoir des "trous" persistant, c'est à dire toujours les mêmes zones à une rotation prêt où l'on ne trouve pas de drone, la **vitesse angulaire** des drones de reconnaissance est **plus élevée** que celle des drones de protection. (L'utilité de cette technique est assez flagrante si l'on ne dispose que de 4 drones. On aura deux drônes opposés à l'intérieur et deux drônes opposés à l'extérieur. Si les 4 drones ont la même vitesse angulaire et sont alignés, il y a un faiblesse évidente sur les flancs où il n'y a pas de drones)



*Position des drônes à deux pas de temps différents. Alors que les drones de protection, à l'intérieur, ont tourné de  $45^\circ$ , ceux de reconnaissance, à l'extérieur, ont tournés de  $90^\circ$*

Lors de leurs déplacements, les drones se verront parfois confrontés à des **montagnes**. Nous avons jugé le contour des montagnes compliqué à effectuer car selon le déplacement du convoi, sa proximité avec la montagne, et d'autres facteurs, cela pourrait rapidement désordonner les positions des drones. Nous avons donc choisi de **survoler** celles-ci. Un drone apercevant une montagne sur son chemin **s'élèvera donc à haute altitude le temps de l'éviter** (en gardant les coordonnées sur x et y de sa destination), **puis redescendra à basse altitude**

Afin de pouvoir gérer leur carburant tout en continuant à défendre efficacement le convoi, nous avons implémenté **un système de roulement** pour le retour à la base des drones. Sur une même couche, les drones se relaient pour retourner à la base : lorsqu'un agent atteint une valeur de **2.5 \* le seuil minimum** de carburant nécessaire pour aller se recharger, il **rentre à la base**. Il indique alors ceci aux autres drones de son équipe, et **un drone sur deux l'accompagne** pour se ravitailler aussi (cela permet de garder une disposition des drones uniforme autour du convoi). Cette valeur de 2.5 permet de gérer l'aller retour du drone et d'éventuels imprévus, tout en permettant aux drones restant assurer la protection d'avoir assez de carburant lorsque ce sera leur tour d'aller se recharger. Une fois les drones partis **revenus à** une distance proche de **la position de repos** à laquelle ils doivent se trouver autour du convoi, les autres drones étant restés **peuvent aller se recharger**. On rappelle que cette synchronisation se fait par équipe : les ravitaillements et aller-retours des drones de protection sont donc **indépendants** de ceux des drones de reconnaissance.

Le ravitaillement d'une équipe peut aussi être enclenché par un **manque de munitions** : lorsqu'il ne reste à un agent qu'un dixième de ses munitions originales, il rentre à la base et le déroulement des actions décrit ci-dessus est suivi. Cette valeur de 1/10 est choisie afin de laisser aux drones la possibilité de **se défendre si nécessaire** sur le chemin du retour.

## 4.2 Comportement vis-à-vis des ennemis

Les drones de protection utilisés pour la défense du convoi étant des plus sophistiqués à la fois dans leur manière de penser, mais aussi dans leur design, ils disposent pour neutraliser leurs ennemis d'un canon à **l'orientation indépendante du mouvement du drone**. Un drone n'a donc pas nécessairement besoin de faire face à un ennemi pour lui tirer dessus. Comme indiqué plus tôt, lors des patrouilles autour du convoi, les drones de protection ne tireront sur les ennemis **que si ceux-ci se trouvent à portée**. La cible est alors la plus proche. Les drones de reconnaissances, eux, ont **une marge de déplacement** pour les pourchasser. Lors du retour à la base, les drones **ignorent** dans leurs déplacements les drones, leur retour au convoi le plus rapidement possible étant **crucial** afin de permettre à ceux étant restés d'aller eux aussi se ravitailler. Cependant, si un ennemi se trouve à portée d'eux durant ce trajet, ils lui tireront dessus, s'ils disposent de munitions. Quand un drone meurt, il en **informe les autres** dans un dernier souffle, et le message est transmis entre les drones. Ceux-ci **se réorganisent alors** pour faire en sorte qu'il y ai bien **autant de drones** dans les deux équipes, et qu'ils soient répartis de **façon uniforme** au sein d'une même équipe. Si le leader des drones vient à mourir, le prochain drone venant dans la liste des drones vivant prend sa place.

Les drones disposent aussi de la capacité de **communiquer** : à chaque étape de la simulation, en dernière action, chaque drone **communiquera, si à portée, une liste comprenant les ennemis qu'il peut observer** au leader, ainsi que leurs coordonnées. Le leader dispose d'une action de plus, après l'envoi des messages de tous les drones : celle de **réception** des messages. Le leader établit une **liste des informations** qu'il a reçu. Pour chaque message reçu des drones, si l'ennemi observé communiqué n'est **pas dans la liste** des ennemis connus, le leader ajoute ses informations à la liste. Une fois le traitement des messages effectués, il **communique alors la liste des ennemis** connus et de leurs position au **chef du convoi** qu'il suit.

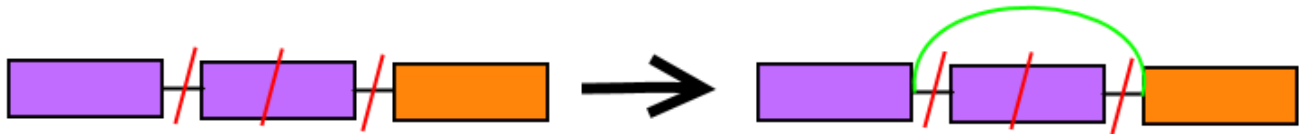


*Schematisation de l'ordre des communications*

## 5 Modélisation et comportement du convoi

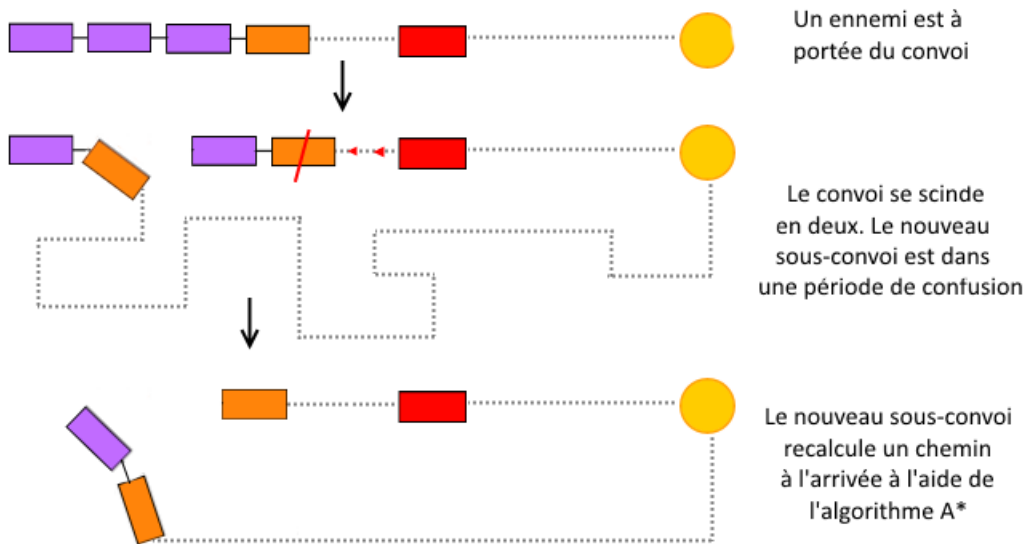
Le convoi est composé d'une **voiture à escorter**, plusieurs **voitures d'escorte** et un **leader**. Dans le projet fourni, un comportement de base du convoi était intégré : un membre du convoi n'ayant pas de voiture devant lui se retrouve leader. Il calcule grâce à l'**algorithme A\*** à l'aide une heuristique simple le meilleur chemin pour aller à l'arrivée et suit bêtement le trajet sans se préoccuper des ennemis de l'environnement. Les autres voitures non-leader suivent alors la voiture se trouvant devant elle.

Nous avons tout d'abord fixé une limite de séparation du convoi : si une voiture meurt, un **lien est créé entre les voitures** auxquelles elle était reliée. Cela fixe donc tout d'abord une **cohésion** du convoi, même si certains éléments meurent.



*Gestion de la mort d'un élément du convoi*

Puis, nous avons implémenté une **distance de danger** entre le convoi et les ennemis. Si un des membres du convoi se trouve à une distance inférieure à l'un des ennemis à la distance de danger, le convoi se scinde en deux équipes de taille égale (pas d'autre divisions récursives) : un lien entre les voitures du milieu est **coupé**, et une **nouvelle voiture devient leader du convoi** scindé. On observe alors une période de confusion ou le nouveau convoi calcule un trajet sous-optimal à l'arrivée (avec la variable **genlongpath**), puis après que cette période se termine, il recalcule un chemin à l'arrivée de la même manière que le convoi initial.

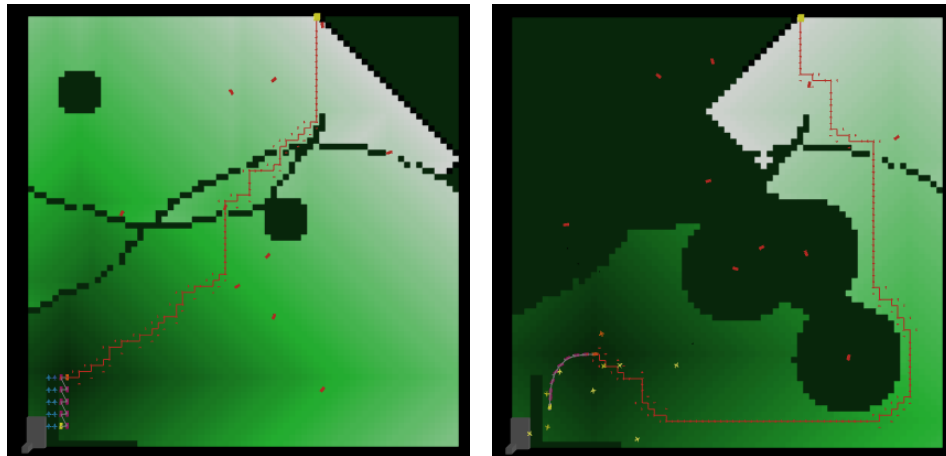


*Gestion de la séparation du convoi*

Les drones protégeant le convoi **se séparent** alors. La moitié des drones restent avec le convoi principal, et l'autre moitié part avec le nouveau convoi. Nous avons alors **deux groupes de drones indépendants** chacun suivant un convoi. Les mouvements des drones sont coordonnées de la même manière qu'expliqué précédemment, mais seulement avec les autres drones du sous-convoi. Les observations des drones sont alors transmises au leader du convoi qu'ils accompagnent, et au leader de l'autre convoi seulement s'il est à portée. Les deux convois ont donc potentiellement des **connaissances différentes**.

Après s'être séparé, nous avons permis au convoi de pouvoir se **reformer** : après un certain laps de temps, le convoi se réunit : **un lien est créé** entre le leader du deuxième convoi (qui perd ses fonctions de leader) et le dernier membre du premier. Les drones se rassemblent alors et récupèrent leurs comportement original, comme un seul et même groupe divisé en deux équipes. Cependant, si des ennemis sont toujours à une portée inférieure à la distance de danger du convoi, celui-ci **se rescinde** instantanément.

Enfin, nous avons parlé précédemment des communications du chef des drones avec le leader dans la section précédente. Après réception d'un message par le leader des drones, le leader du convoi **met à jour ses connaissances concernant les positions des ennemis**. Nous avons donc voulu faire en sorte que le convoi **évite les ennemis** dont il connaît la position. Pour ce faire, nous avons modifié l'**heuristique** utilisée dans l'algorithme A\* fourni : le convoi cherchant à éviter à tous prix les ennemis, nous avons ajouté un **coût très élevé** aux patchs se situant à **proximité d'un ennemi** connu. Nous observons alors généralement un chemin **plus long, mais plus sûr** jusqu'à l'arrivée. Cependant, comme recalculer un meilleur chemin à chaque pas de temps serait très coûteux, nous avons décidé de recalculer un nouveau chemin à des **intervalles réguliers**. Un chemin est donc recalculé avec des positions des ennemis connus mis à jours. Il est important de noter que le nouveau chemin n'évite que les ennemis vus par les drones : il peut passer en **plein milieu d'un groupe d'ennemis situés hors du champs de vision des drones**. Cependant, une fois proche de ce groupe d'ennemi, lors du prochain calcul de chemin, il est probable que le convoi cherche alors à l'éviter.



*A gauche, le chemin initial calculé par le leader du convoi. A droite, le chemin calculé après connaissance des positions des ennemis. Plus un patch est foncé, et plus son coût est important.*

## 6 Améliorations et Conclusion

Nous avons atteint la majorité de nos objectifs et sommes globalement satisfait de notre travail. Nous avons cependant rencontrés quelques problèmes que nous n'avons pas réussi à résoudre : en utilisant l'algorithme A\* fourni, lors de l'utilise de longpath, le patch choisi est celui de pire coût. Cependant, en modifiant l'heuristique pour y inclure le coût des agents, il n'est pas possible d'inclure à la fois l'évitement d'ennemi, et un choix de long chemin, les **patches de pire coût étant ceux proches des ennemis**. De plus, nous n'avons pas réussi à réduire la portée de communication lorsque le message passe à travers une montagne. Enfin, lors du scindement du convoi en cas de danger, lors de la réunification, le second convoi rejoint le premier en traversant les patches d'obstacles comme les rivières ou les lacs. Il aurait fallu déterminer un point de rendez-vous où les deux convois puissent se rejoindre, où recalculer grace à A\* un nouveau chemin du second convoi vers le premier mais cela aurait été couteux en communications et en calculs.