MANEFF Lucas 324942, THEVENOZ Julien 326772

Rapport

L'algorithme de connexion fait deux choses: il active le booléen *connected* de tous les robots connectés à la base et les ajoute au tableau *connected_robs*. Pour accéder aux informations des robots (amis ou concurrents) qu'elle connaît, la base utilise *connected_robs*. Pour donner des ordres uniquement à ses propres robots, elle parcourt sa *flotte* (le tableau de ses propres robots) et interagit avec ceux dont l'attribut *connected* est vrai.

Avant de créer de nouveaux robots, la base vérifie la quantité de ressource qui lui reste. Elle fait en sorte de toujours garder au moins une kilotonne [kT] de ressource afin de ne pas disparaître. Lorsqu'elle s'apprête à créer un Prospecteur ou un robot de Communication, la base vérifie en plus qu'il lui resterait encore au moins 201 [kT] de manière à créer un Foreur et un Transporteur si elle découvrait un nouveau gisement.

Les robots de communication sont importants pour garder le lien avec les Foreurs et les Prospecteurs en tout temps. De plus, leur coût de création étant faible par rapport à la quantité de ressource disponible dans les nouvelles bases, nous avons décidé de commencer par déployer un réseau de communication 5G sur l'ensemble de la planète Donut. Les robots se positionnent donc sur un quadrillage d'une longueur de maille légèrement plus faible que leur rayon de communication. Une fois arrivés, les robots restent immobiles; il est donc inutile de les réparer ou de leur ré-assigner un but.

Une fois le réseau de communication déployé, la base crée jusqu'à 10 Prospecteurs qu'elle envoie en éventail. Ceux-ci avancent en ligne droite jusqu'à trouver un gisement, auquel cas ils retournent à la base. Arrivés à la base, ils sont réparés si leur usure leur laisse moins de (2*√2*dim_max) km à parcourir. Ensuite, ils reçoivent une nouvelle direction aléatoire à explorer.

A tout instant la base se tient prête à déployer ensemble un Foreur et un Transporteur directement sur un gisement nouvellement découvert. Le premier s'immobilise à l'extrémité du gisement (une fois fixé sur un gisement, il ne bougera plus) et le second fait des allers-retours permanents. A chaque fois que le Transporteur revient à la base, il est réparé et renvoyé vers le Foreur dont le gisement possédant la plus grande capacité.

Les Foreurs et Communications ne sont donc jamais réparé alors que les Prospecteurs et Transporteurs le sont régulièrement. Cependant, étant donné que le coût de réparation est négligeable par rapport à la création des robots, cette maintenance est profitable pour la base.

La stratégie des bases met l'accent sur l'information : le but est de découvrir rapidement le plus de gisements possibles. C'est pour cela que le réseau de robots-antennes est mis en place d'abord, afin de permettre à la base de recevoir au plus vite les informations des Prospecteurs. Ceux-ci sont déployés juste après les Comms et envoyés en demi-cercle afin de couvrir (grâce au rebouclement de l'espace) la majorité de la planète. Les 10 vecteurs initiaux des Prospecteurs, plus les directions aléatoires qui leur sont attribuées quand ils reviennent à la base, permettent d'après nos tests empiriques de découvrir la majorité des gisements en peu de temps.

Cependant il faut pouvoir exploiter ces gisements : c'est pourquoi pendant la phases d'exploration la base garde toujours une réserve suffisante pour créer un couple Foreur/Transporteur. Dans le cas improbable où une base trouverait un gisement avant d'avoir envoyé ses 10 Prospecteurs, la priorité est donnée à la création du couple For/Trans plutôt qu'à celle des Transporteurs afin d'exploiter au plus vite le gisement. Enfin, lorsqu'ils reviennent à la base, les Transporteurs sont toujours redirigés vers le gisement le plus prometteur afin d'éviter de faire des trajets inutiles pour n'avoir rien à ramener à l'arrivée.

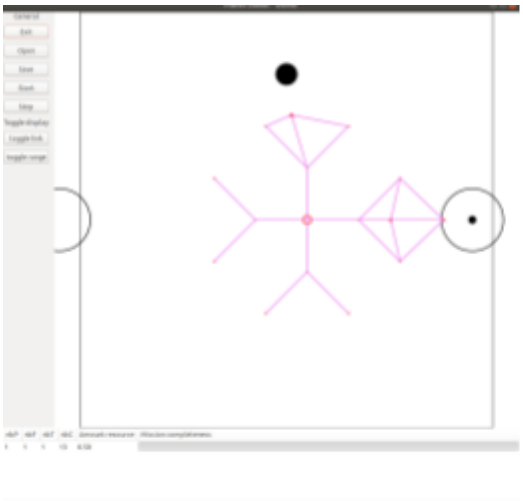
Méthodologie et conclusion:

Nous avons commencé par une phase d'analyse pour déterminer et répartir les différentes tâches à accomplir. Nous nous sommes vite rendu compte que pour tester les algorithmes des bases et des robots, avoir fini l'interface graphique et l'affichage serait un atout. Nous avons donc décidé de programmer cette partie en priorité et de séparer le travail en deux modules. Julien était responsable du module **gui** et Lucas du module **graphic**. C'est à ce moment qu'un bug très embêtant est apparu. En effet, le pointeur sur le contexte du dessin provoquait une "segmentation fault" lorsqu'il était utilisé dans le module **graphic**. Comme la résolution du bug nous a pris un certain temps, Lucas a continué de développer les fonctions de dessin dans un fichier à part. Finalement, grâce à l'aide de notre assistant, le problème a été résolu en appelant les méthodes de dessin uniquement depuis *on_draw()*. Une fois l'interface graphique terminée, il a fallu imaginer une stratégie pour nos bases. Même si nous n'étions pas vraiment bloqué par des bugs lors de la programmation de cette partie, elle nous a pris plus de temps que la précédente car elle était conceptuellement plus compliquée. Nous avons aussi parfois eu un peu de peine à comprendre ce qui était autorisé par la donnée et ce qui ne l'était pas. A partir de ce moment, l'organisation était un peu plus chaotique et comme nous programmions la majeure partie du temps ensemble, l'identification et la répartition des tâches se faisait le jour même. Les tests se faisaient sur des fichiers d'initialisation simples qui nous permettaient de mettre en évidence le comportement de la fonction testée.

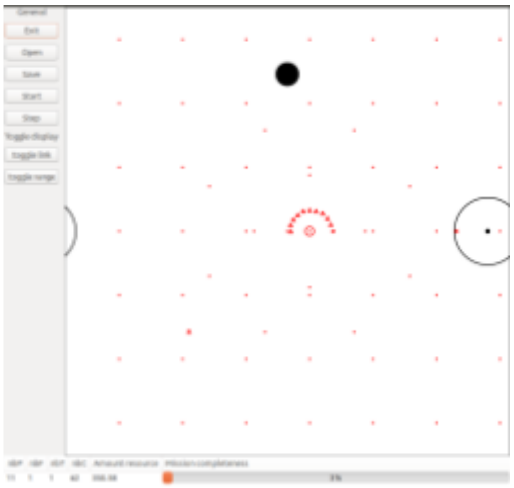
En conclusion, notre organisation et la répartition du travail ont été bien gérés durant les deux premiers rendus et la moitié du troisième, mais ont souffert sur la fin ce qui nous a fait perdre pas mal de temps (notamment lorsqu'il fallait recopier le code de l'un chez l'autre). L'utilisation d'un logiciel comme gitLab, nous aurait fait gagner un temps précieux et il serait peut être judicieux à l'avenir d'en faire une petite présentation en cours. De notre côté, il aurait fallu que nous prenions plus de temps pour discuter de l'architecture globale des algorithmes de comportement. En effet, la modularisation nous a souvent empêchés d'écrire des fonctions que nous avions conçues sans faire attention à cet aspect. Pour finir, notre assistant était au top (Leo Meynent, c'est le boss). Il nous a donné de bons conseils et nous a aidé à déboguer des problèmes récalcitrants.

MANEFF Lucas 324942, THEVENOZ Julien 326772

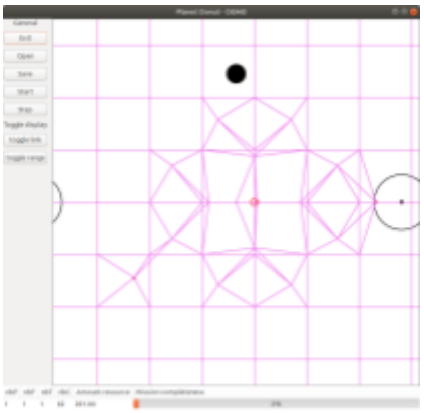
Simulation avec “rendu3_image.txt"



Initialisation

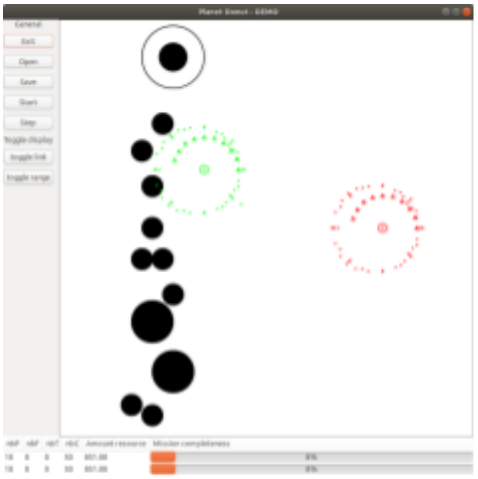


Déploiement des prospecteurs



Le réseau de communication est établi

Simulation avec “rendu3_03.txt”



Les bases déploient leurs Comms
et leurs Prosps

Les réseaux de voisins s’entremêlent