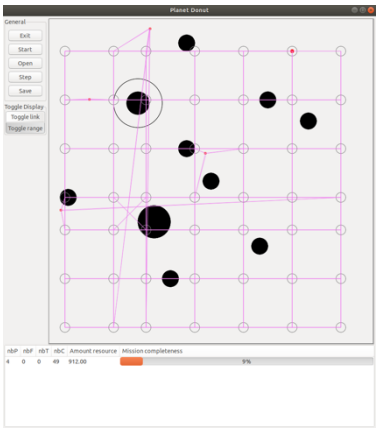
Rapport Final (3) Programmation Orientée Projet (23.05)

Lexique:4G : réseau de communication, R : ressources, combo : un robot de forage et un de transport.

La première étape dans la mise au point de notre algorithme fut une analyse des données et des constantes. Tout d'abord : 49 robots de communication suffissent pour recouvrir la totalité de la planète à travers un réseau que l'on nommera dans la suite 4G. Chacun a un coût de 1 R donc 49 R en total. En deuxième un Robot de transport est inutile sans Robot de forage et vice versa. Ces deux robots sont chacun à un prix de 100 R, la combo coûte donc : 200 R. De plus le coût de réparation est plus attractif que le coût de production d'un robot. Pour un robot prospecteur parcourir 20.000 km coûte 15 R. Envoyer deux robots coûterait 20 R. Enfin le forage d'un gisement rapporte 250 R. Notre stratégie repose sur la phrase : « Une seule entrée de forage suffit pour créer la 4G et détruire un autre gisement ».

La construction de plusieurs stratégie a avancé un point : seulement le robot de prospection a un comportement fondamentalement différent entre le mode autonome (sans connexion à la base) et le mode remote (connecté à la base). Dans le mode autonome le robot ne peut effectuer une prospection que s'il a atteint son but, il doit rentrer à la base après directement. Contrairement au mode remote ou il a le droit d'effectuer une prospection à chaque déplacement. Ceci multiplie par 2 fois la distance au but les possibilités de rencontre de gisement, d'où l'intérêt d'avoir des robots connectés. Les autres robots qu'ils soient en mode remote ou autonome ne se déplacent uniquement que vers leurs but, effectuent leurs actions respectives et rentrent à la base ensuite sauf pour le cas des robots de communication.



Enfin notre stratégie consiste en établir la 4G le plus rapidement possible pour maximiser la surface de recherche et dès qu'un gisement est trouvé et un seul robot de transport est rentré, la base survit toujours.

On peut identifier 4 « intervalles de R » possibles :
Cas 1 -> la base a moins de 200 R et pas de robot de combo, alors elle est morte car elle sera vidée lentement par les maintenances et ne pourra pas se mettre à vider les gisements
Cas 2 -> la base a plus de 70 R avec au moins un combo, alors elle peut générer la 4G et un robot prospection pour une nouvelle recherche car une entrée d'un robot de transport suffit pour remettre la base en situation confortable (250+)
Cas 3 -> le base a plus que 270 R alors elle peut créer la 4G (-49R) un robot de prospection (-10 R) et deux robots pour vider le gisement (-200R) et a 11 R pour la maintenance.
Cas 4 -> la base a entre 200 et 270 ressources et pas de robots de combo, alors elle envoi uniquement un seul robot de prospection jusqu'à avoir un gisement et après elle envoie un combo et crée la 4G que si un robot de transport est rentré.

Le pseudocode suivit lors du projet est le suivant :

```
// entrée modifiée : ensemble des bases taB

Boucle infinie de la simulation : un passage = une mise à jour

   Pour i de 1 à nbB        // dans le même ordre de celui du fichier de test
      Pour j de 1 à nbB     // dans le même ordre de celui du fichier de test
         update_voisin(taB[i].robots, taB[j].robots)
      connexion(taB[i])
      maintenance(taB[i])
      creation(taB[i])
      update_remote(taB[i].robots)
      update_autonomous(taB[i].robots)

   Pour i de 1 à nbB        // dans le même ordre de celui du fichier de test
      Si taB[i].ressource ≤ 0
         destruction(taB[i]) // dont l'ensemble de ses robots
```

Cet algorithme parcourt la liste de bases qui est un attribut de la classe Simulation. Toutes les fonctions suivantes sont des méthodes de cette classe Simulation

- La méthode **update voisins** déjà décrite dans le rendu 2 a pour but de créer pour chaque

robot la liste des robots voisins. Elle nous permet de construire le graphe de connexion de la base et d'afficher les liens. Après considération de notre stratégie nous nous sommes décidés à considérer uniquement les robots d'une même base comme voisins.

- La méthode **connexion** vas parcourir tous les robots qui sont connectés au robot de communication au centre de la base et vas mettre grâce à une fonction de récurrence le booléen de connexion pour chaque robot du graphe à vrai. Tous les robots sont conservés dans un unique tableau « robots_base » qui conservera de manière centralisée toutes les informations. Nous nous sommes servi uniquement de ce dernier tableau et du tableau des robots voisins de chaque robot.
- La méthode de **maintenance** est relativement simple. Elle va remettre tous les compteurs des robots en superposition avec elle à zéro en échange d'un coût donné par : *coût= 0.0005 * distance parcourue*. Cette méthode s'active pour tous les robots peu importe la situation à condition qu'il soit de sa propre base, ceci s'effectue toujours systématiquement vu le coût faible de la maintenance.
- La méthode de **création** effectue 3 points. Tout d'abord elle juge si elle peut/doit encore créer une 4G. Ensuite elle envoie au plus 4 robots de prospection à la recherche de gisement. Enfin elle va chercher dans les robots de prospection connectés si un robot a trouvé un gisement et vas décider la création d'un combo ou pas. De manière générale un robot n'est créé que si on en a besoins, jamais pour confort.
- Enfin **updates_remote** et **update_autonome** sont des méthodes qui appellent les méthodes virtuelles des robots : mode_remote et mode_autonome. Elles indiquent aux robots la procédure à appliquer selon leurs connexion.
- Une méthode de **mise à jour de robots** a été créé en début de code pour remplir un tableau avec tous les robots possibles pour, au besoins, pouvoir toutes les variables d'un seul coup.
- Enfin une méthode de **mise à jour des bases** a été mise au point pour conserver les bases uniquement qui ne sont pas autonome. Les bases qui ont moins de 200R et pas de combo ou qui ont 0 R sont détruits.

Prise de décision des robots en fonction de la connexion

	Remote	Autonome
Communication Représenté par un cercle vide	Déplacement uniquement vers le but. Pas de redéfinition du but	Déplacement uniquement vers le but. Pas de redéfinition du but
Forage Représenté par un triangle avec la pointe vers le bas	Déplacement vers le but uniquement. Il n'y a pas de redéfinition du but car on crée un combo que si le gisement est profitable. Le robot de forage va prélever une quantité de ressources deltaR s'il a un robot de transport qui se superpose avec lui, puis vas le renvoyer. Si le gisement n'est pas vide.	Même situation que dans le cas remote.
Transport Représenté par un triangle avec la pointe vers le haut	Déplacement vers la position d'un robot de forage. Son but pour la recherche de R est défini par la base grâce à une méthode qui renvoi la position d'un robot de forage qui ne se fait pas encore vider et qui a un gisement qui n'est pas vide.	Déplacement vers un robot de forage. Ce dernier lui donnera l'ordre de rentrer si possibilité de forage. Dès qu'il reviendra pour la première fois la 4G sera mise en place et on se réfère au cas remote alors et on trouve le nouveau but comme décrit.
Prospection Représenté par un carré rouge	Déplacement vers le but et vérification tout le long du chemin de la présence un gisement. Son but est fixé aléatoirement et il ne revient à la	Déplacement vers le but, à l'arrivée on vas vérifier la présence d'un gisement puis rentrer à la base. Dès qu'on rentre à nouveau dans la zone de

	base que pour effectuer une maintenance ou bien si la base est devenue autonome.	connexion on choisit le nouveau but aléatoirement ou on rentre pour maintenance.

Remarque : le nombre de robots de communication et de prospection créés au maximum sont fixés à 49 et 4 respectivement. Seulement les robots de combo sont créés lors de chaque découverte de gisement d'intérêt. Le nombre de prospection a été choisi de manière arbitraire pour avoir un nombre gérable à toute étape dans la simulation.

Nous considérons notre stratégie comme robuste car comme les robots de prospection choisissent au hasard leurs but les deux bases doivent compter sur la chance pour trouver un gisement. Ensuite la mise en place de la 4G permet maximiser les chances de survie.

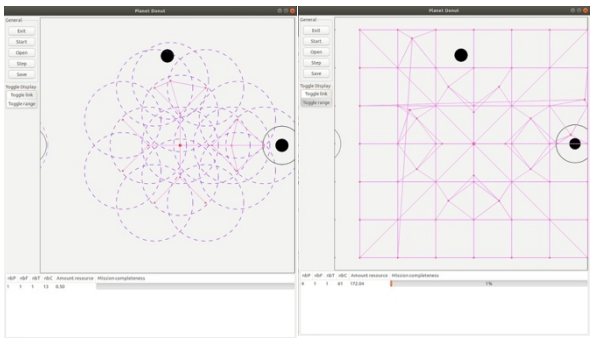
Rétrospectivement le problème/bug le plus fréquent auquel nous avons dû faire face est assez clairement le segmentation fault. Surtout lorsqu'elle survient au milieu de la simulation. Nous avons dû faire appel à une recherche par dichotomie pour cerner les erreurs et les résoudre.

En conclusion la première chose que nous voulons maintenir est la fascination que nous avons ressenti lors de la réalisation de ce projet. Nous étions très heureux de pouvoir appliquer les connaissances apprises pendant le BA1 et le BA2 et vraiment d'en réaliser l'utilité par soit même.

Nous nous sommes toujours très strictement réparti les tâches puis à la fin mis en commun. Georg s'est occupé de la construction du modèle (modules : geomod, gisement, robot, base et simulation) tandis que Daniel a pris en charge toute la partie utilisation des modèles (modules : graphic, graphic_gui, gui, scroll, projet et du Makefile). Lors de la réalisation les codes ont été testés avec soit des fichiers test soit dans un autre fichier pour voir si la fonction en question éffectuait bien la tâche demandée.

Rétrospectivement nous changerions notre approche individuelle du projet. Nous sommes partis trop rapidement sur des scénarios improbables qui pourraient poser problème et nous nous sommes éloigné trop rapidement des grandes lignes.

Description :



rendu3_image.txt : Après que le premier robot de transport est rentré (image 1) la 4G est mise en place et le es autres gisements sont trouvés rapidement

rendu3_fig4.txt : On a plus que 270 ressources on met en place la 4G directement on libère les robots de prospection et on vide les gisements (taille diminue au fur et à mesure des prélèvements)