RAPPORT FINAL

Architecture logicielle et description de l'implémentation

- ### Structures de données robot et particules

Les données sont stockées dans deux tableaux de structures, l'un pour les robots l'autres pour les particules. La structure comprend les éléments de bases (positions, angle, rayon, énergie) et des éléments supplémentaires.

Robot[i]	Double Pos_x	Double Pos_y	Double Angle	S2D occup[1]	Bool actif[2]	Int color[3]	Double vrot[4]	Double vtran[4]
Particule[i]	Double Energie	Double Rayon	Double Pos_x	Double Pos_y	Bool ciblee[5]	Pour 0 < i < nb_entité		

Le tableau de particules est trié par rayons décroissants, évolue et adapte sa taille en fonction du nombre de particules à l'aide d'une fonction de réallocation.

1. La structure occupée indique la position du but du robot
2. Le booléen actif indique si le robot possède un but ou non
3. L'entier color modifie la couleur du robot (si rouge alors il entre en manuel)
4. Les doubles vrot et vtran sont respectivement les vitesses de rotation et de translation des robots
5. Le booléen ciblee indique si une particule est déjà ciblée par un robot ou non

- ### Décription de l'algorithme de coordination entre les 3 modules simulation/robot/particules

Le module simulation décrit la structure de la boucle fonctionnelle durant une Step. Les fonctions de simulations calculent les différents cas de figures que peuvent avoir les robots, et appellent des fonctions des modules robot et particule pour modifier les tableaux de structure correspondants. Les fonctions de simulations calculent également **la collision entre robots et particules** ainsi que **l'édition de but** pour les robots en fonction selon les particules les plus grandes.

Les fonctions du **module Robot** sont principalement des **fonctions d'affectation** de variables aux champs de structures. Elles calculent également **le déplacement optimal** d'un robot vers son but, le déplacement d'un robot selon les vitesses de rotation et translation données et la **correction de la position** lors d'une collision.

Les fonctions du **module Particules** sont principalement des **fonctions d'affectation** de variables aux champs de structures. Elles possèdent de plus une **fonction de tri** par sélection en fonction des rayons décroissante, une **fonction de décomposition** des particules et une **fonction de réallocation** de la mémoire en fonction du nombre de particules.

Lecture du Fichier : Lecture Robots + Erreurs collisions robots
Lecture Particules + Erreurs collisions particules
Collisions robot-particules
Simulation boucle :
Pour i allant de 0 à nb_particules
Décomposition aléatoire
Tri_Particules - Reallocation
Edition de But (appels de fonction de robot)
Pour i allant de 0 à nb_robots
Evaluation collision
Si collision avec une particule k
Rotation i ou Destruction de la particule k
Sinon si aucune collision avec un robot
Déplacement vers son but
Alignement par rapport à toutes les entités
Alignement du robot
Sinon
Déblocage du robot[i] (appels de fonction de robot)
Mode manuel (appels de fonction de robot)

~Module Particule

~Module Robot

~Module Simulation

1 Algorithme Général du Programme

La liste de particule est exploitée comme suit :

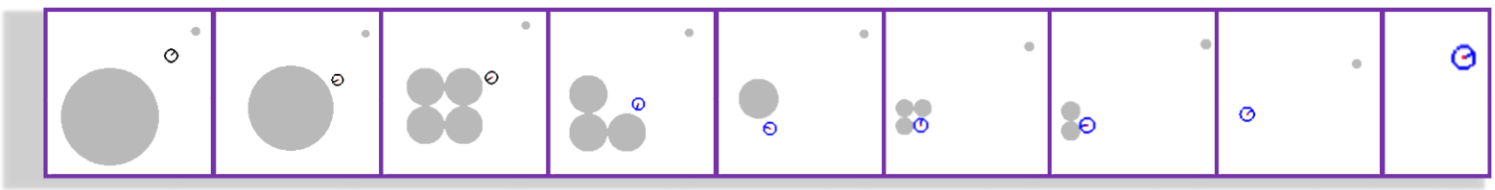
Si décomposition : réallocation supérieure du tableau et affectation de valeurs aux nouvelles structures en fonction de la particule décomposée

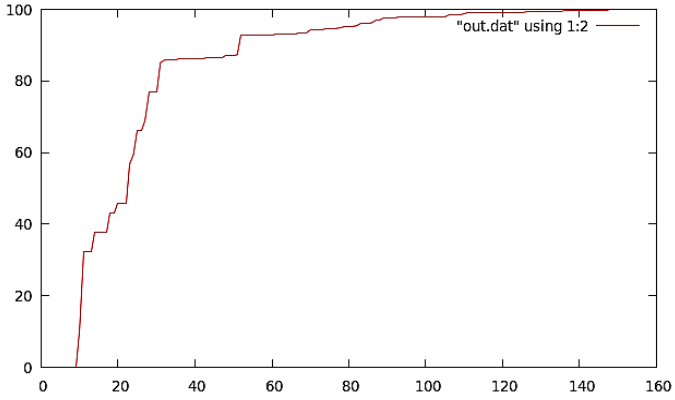
- Le rayon de la particule décomposée devient 0.
- Après tri les particules de rayons 0 se mettent à la fin
- Réallocation Inferieure du tableau : suppression des particules de rayon 0
- Lors de la simulation : si particule détruite son rayon devient 0 et sera triée et supprimée au tour d'après

- **Estimation du coup calcul en fonction de nbRobot et nbParticules pour le mode automatique**

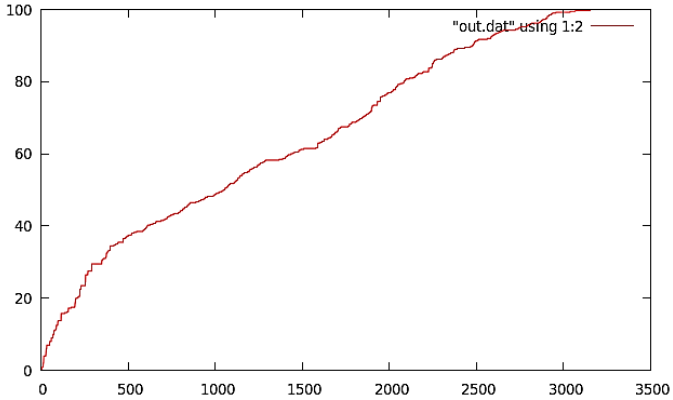
Cout calcul : O(nb_particules2 + nb_particules*nb_robots2)

Cout mémoire : STR_ROBOTS*nb_robots + 2*STR_PARTICULES*nb_particule





2 Graphe Gnuplot test D09.txt



3 Graphe Gnuplot test D03.txt

Méthodologie et conclusion

- **Organisation du travail à plusieurs**

Nous avons choisi de séparer les modules au niveau du rendu 2 entre nous, Lucas s'est occupé du mode Draw et de l'affichage tandis que Andy s'est occupé du mode Error et de l'automate de lecture.

Puis pour le rendu 3 nous avons continué sur ce chemin ou chaque membre du groupe a continué sur le module qu'il avait commencé

- **Personne responsable de chaque module**

Andy a été responsable du module robot, particule et simulation, et Lucas s'est chargé du module principal, du dessin, de l'interface graphique et de la coordination entre les modules

- **Comment le travail a été organisé pour chaque module**

En ce qui concerne l'organisation du travail, nous avons commencé par les fonctions de base, une par une, en testant à chaque fois qu'elles fonctionnent bien, puis une fois que la fonction fonctionne correctement, nous passons à la suivante.

Les tests sont réalisés sur chaque fonctions avec tous les cas de figure disponibles dans les fichiers tests, puis quelque test additionnel un peu plus complexes pour certaines fonctions. En cas de bug, plus de tests sont effectués pour pouvoir localiser l'origine du problème.

Avec un peu plus de recul maintenant, Nous aurions commencé le projet différemment en réfléchissant déjà a l'avance comment ce qu'on a fait servira plus tard dans le projet.

- **Problème le plus fréquent ? Pourquoi ?**

Le Segmentation Fault a été le bug le plus fréquent, Il provenait en général de problèmes avec l'allocation et réallocation dynamique de mémoire pour le tableau de particules qui

changeais de taille au cours de la simulation. Il provenait aussi souvent lors de la manipulation de Fichier comme le fichier de sauvegarde, l'ouverture de fichier ou l'enregistrement du Taux de Décontamination.

- **Problème qui a posé le plus de problèmes**

Le Segmentation Fault provenant de la Réallocation de mémoire pour le tableau de particule a posé le plus de problème à cause de sa taille variable et des particules qui s'effacent et se décomposent durant la simulation.

La collision robot-robot a aussi posé quelque problème au niveau de la détection de collision et de l'alignement qui en dépendait.

Conclusion

- **Brève auto évaluation**

En conclusion, nous sommes parvenus à un résultat plutôt bon, même si le début du projet aurait pu être géré d'une façon plus efficace, qui par exemple ne nous aurais pas forcé à changer tout le système de gestion des particules pour modifier la taille de tableau, ce qui est une perte de temps qui aurais pu être évitée.

- **Evaluation de l'environnement de travail**

L'environnement a été très bon en général, le forum est un très bon moyen d'obtenir plus d'information si quelque chose n'est pas clair dans la donnée ou lorsqu'une réponse du professeur est nécessaire.

Seul point négatif serait la lenteur des ordinateurs en salles lorsque beaucoup de monde est présent à la fin de l'année ou pendant les heures d'exercices, mais avoir la VM palie se problème efficacement.

Les rendus publics peuvent être utiles si jamais un retard est pris sur le projet ou qu'une partie du projet n'est pas comprise par un groupe. C'est un bon moyen de se remettre à jour et de voir une façon différente d'aborder un problème aussi, ce qui peut donner lieu à de nouvelles idées.