Axel Hall 346228
Lucas Michel 363073

Rapport final rendu 3

1. Prises de décision

La prise de décision quant au déploiement d'un nouveau robot se crée selon les besoins actuels de la simulation. Un nouveau robot neutralisateur sera déployé si le nombre de particules est plus de 2 fois supérieur au nombre de robots neutralisateurs déjà déployés. Les robots réparateurs sont également déployés selon les besoins, et un nouveau robot réparateur sera déployé si le nombre de robots neutralisateurs en panne est supérieur au nombre de robots réparateurs déjà déployés.

a) Robots réparateurs

Les robots réparateurs mettent à jour leur cible dès lors qu'un robot neutraliseur tombe en panne. Si ce dernier n'est pas atteignable, le réparateur peut soit prendre en cible un autre robot en panne, soit rentrer à la base s'il n'y a plus aucune réparation à effectuer. Dans le cas où aucun robot réparateur ne se trouve sur le terrain au moment d'une panne, le robot spatial peut en déployer un en suivant le même algorithme de décision.

b) Robots neutralisateurs

Les robots neutralisateurs peuvent être assignés à une nouvelle particule dans plusieurs cas : leur précédente cible s'est faite détruire ; la particule ciblée s'est séparée ; un autre robot se trouve dans une meilleure position après avoir éliminé sa cible ; le robot entre en collision avec une particule se trouvant sur son chemin. Dans les quatre cas cités, nous avons opté pour assignation globale, c'est-à-dire que tous les robots obtiennent leur cible en même temps lors de l'appel d'une méthode appartenant au module Simulation. L'assignation est faite avec les priorités demandées dans la donnée du projet.

c) Simulation

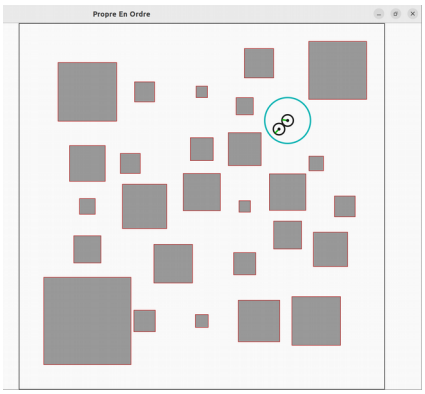

Figure 1: Déploiement d'un neutralisateur sans collision (update 20)

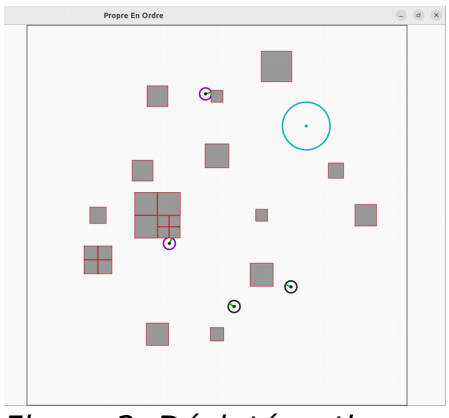

Figure 2: Désintégration de particules (update 855)

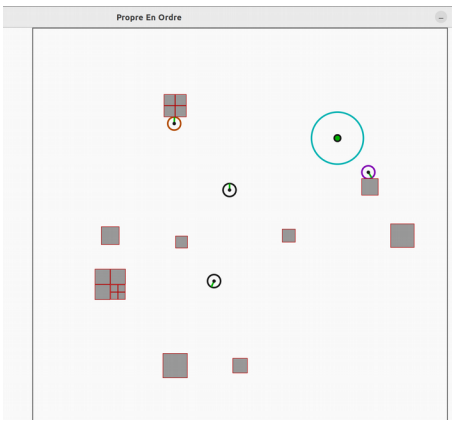

Figure 3: Panne d'un robot Neutralisateur et déploiement d'un réparateur (update 1228)

Axel Hall 346228
Lucas Michel 363073

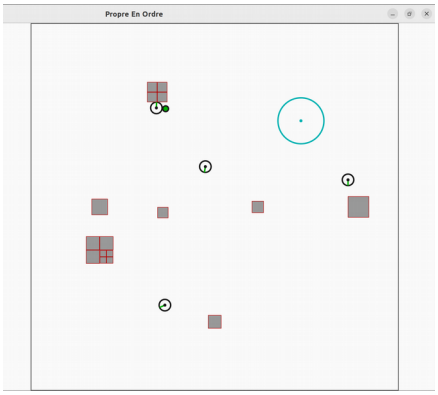

Figure 4: Réparation du robot en panne (update 1418)

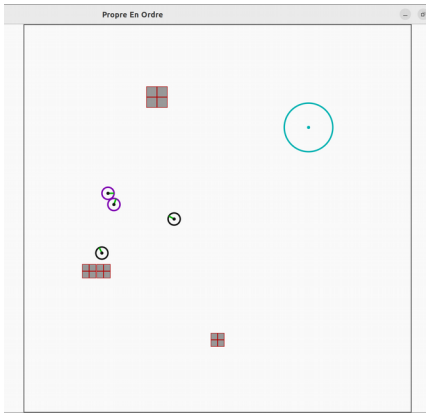

Figure 5: collision entre deux robots (update 1814)

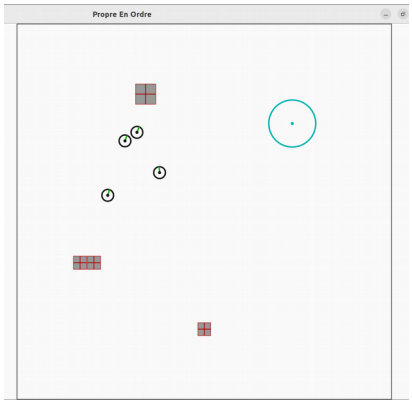

Figure 6: Mouvement des robots vers les 4 particules du haut (update 1932)

Pour réaliser ce projet, nous avons utilisé Visual Studio Code ainsi que Git, qui nous permettaient respectivement d'éditer plus efficacement notre projet et de nous envoyer tout changement apporté au projet.

Contribution

Module	Participation Lucas Michel	Participation Axel Hall
projet	50%	50%
Simulation	30%	70%
Gui	50%	50%
Robot	40%	60%
Particle	70%	30%
shape	70%	30%
graphic	90%	10%

Nous avons utilisé pour ce projet une approche top-down. Nous avons d'abord implémenté les fonctions générales requises par le projet avant d'ajouter des fonctions secondaires qui permettent de finaliser le développement. Nous avons souvent commencé par modifier le module simulation avant d'implémenter les fonctions requises par ce dernier dans les autres modules, notamment robot.cc et particle.cc.

Pour ce projet, nous avons principalement travaillé de manière individuelle (90%). les 10% de travail commun ont servis à la répartition des tâches et à la gestion de bugs majeurs. Ces derniers étant souvent liés à de nouvelles fonctionnalités qui entraient en collision avec notre code précédent.

Selon nous, nous avons réussi à atteindre le but initialement fixé par ce projet. Nous avons réussi à avancer de manière efficace lors des différents rendus en effectuant chacun des tâches nécessaires à l'avancée du projet. Le cadre du projet nous a permis de structurer l'avancée du code, tout en nous fournissant un appui en cas de problème.