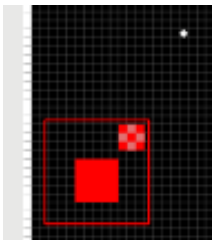
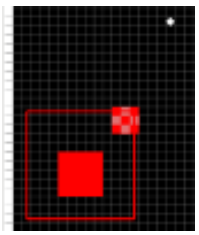
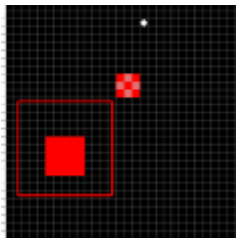
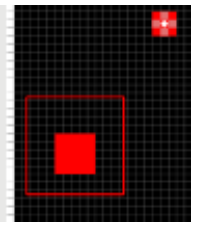
yasmine tligui 341215 zaynab hajroun 346235

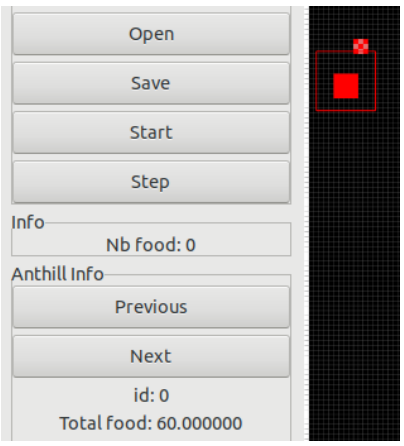
RAPPORT RENDU FINAL

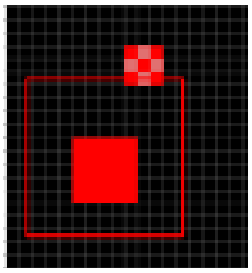
Déplacements des fourmis:
En cas d'agrandissement ou de rétrécissement de la fourmilière, la fourmi Generator se dirige vers le centre de la fourmilière. Ainsi, en cas de rétrécissement il n'y a pas de risque que celle-ci se retrouve en dehors de la fourmilière.
En ce qui concerne la naissance des fourmis, celle-ci se fait d'abord en bas à droite, puis balaye la fourmilière de bas en haut et de droite à gauche. L'agrandissement prioritaire étant celui du coin en haut à droite, et donc celui qui a le plus de probabilité d'avoir lieu, nous avons trouvé judicieux de les créer ainsi.
En cas de deux chemins équivalents, la fourmis collector choisis par défaut le chemin de sens1. Dans la fonction chem_alternatif(), une boucle if/else vérifie si la condition suivante est vérifiée: le nombre d'obstacles du chemin deux est supérieur ou égale à celui du chemin un. C'est pourquoi, en cas d'égalité, la condition est vérifiée et le programme rentre dans la boucle et choisit le chemin de sens1.
Sans nourriture cible, la fourmi collector est immobile et reste au bord de la fourmilière par souci de simplicité. Ainsi le chemin à "l'aller" et au "retour" est similaire.
Le predator revient à la fourmilière si il n'a pas de cible. Ainsi, lorsque la fourmilière passe à l'état free le predator est le plus proche possible de la fourmilière et peut y revenir le plus rapidement possible.
Le Defensor s'arrête lorsqu'il rencontre un obstacle. Cela permettra notamment aux defensors en deuxième position de ne pas mourir en cas de rétrécissement de la fourmilière.

Certaines étapes de la simulation(nombre de mises à jour devant chaque capture):
0.  1.  4.  11. 

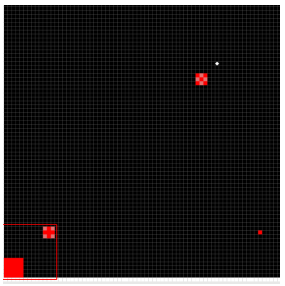
Le fichier est le test 07, on voit qu'à la première mise à jour la fourmis se dirige vers le coin le plus proche. Sachant qu'elle se trouve à équidistance des bords du haut et du bord de droite, elle choisit le chemin qui l'éloigne le plus des bords du monde. Il n'y a pas d'obstacles, ainsi, comme expliqué précédemment la fourmis choisira le chemin de sens 1.
Après avoir récupéré la nourriture la fourmis est loaded, elle change alors d'apparence. On voit dans la capture d'écran suivante le retour de la fourmis à la fourmilière et l'augmentation du total food. Enfin, après plusieurs mises à jour, le collector ne bouge pas puisqu'il n'a plus de nourriture cible.

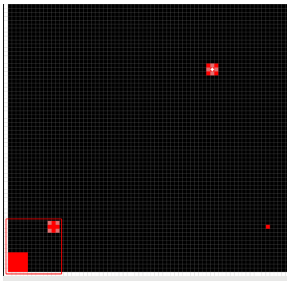
yasmine tligui 341215 zaynab hajroun 346235



25. 

POur ce test là, nous avons, un type d'entité pour la même fourmilière, il n'y a pas de cible pour le predator, il n'a donc rien à attaquer

1. 

5. 

• *Méthodologie et conclusion :*

Le travail a été réparti au fur et à mesure et en fonction du temps que prenait chaque étape. Nous avons commencé par la génération des nourritures, l'agrandissement des fourmilières ainsi que la gestion des vies avant de s'attaquer aux mouvements des différents types de fourmis.
Nous avons fait le choix d'écrire chacune sur une machine pour écrire et tester les différents types de mouvements de fourmis. Par la suite, nous avons mis en commun nos codes respectifs, puis établi les fonctions d'interaction entre les différents types de fourmis.
Malgré notre choix de se partager l'écriture des tâches, nous avons souvent codé en même temps, côte à côte, pour pouvoir demander de l'aide si besoin, et ainsi avoir un regard extérieur sur le code. Avec le recul, nous pensons que cette méthode a été plutôt efficace. Ainsi chacune se concentre sur une tâche,tout en connaissant le fonctionnement du code de l'autre et en pouvant proposer de l'aide si besoin ou bug.
Le bug le plus fréquent que nous avons rencontré était la segmentation fault. De part le choix que nous avons fait d'utiliser des vectors pour stocker les entités, nous avions eu énormément de bugs liés à des tentatives d'accès à des cases d'un vector qui n'existaient pas.A l'aide de sortie de clavier (cout) nous avons cherché à évaluer à quel endroit le bug était présent, puis souvent en affichant la taille du tableau et l'indice de l'élément que nous cherchions à atteindre, nous comprenions d'où venait l'erreur.
Nous pensons que le temps a été notre plus gros point faible, au vu du temps que nous prenaient les segmentation fault a être debuggées. Cependant, et au vu de la charge de travail que demandait ce rendu, nous pensons avoir fait preuve d'organisation dans notre travail d'équipe.

yasmine tligui 341215 zaynab hajroun 346235