Documentation TCHANZ 296190_331471

Choix d'implémentation pour les fourmis :

Le lieu de naissance des fourmis est choisi au hazard pour garantir une homogénéité entre chaque famille de case et donc maximiser le nombre de nourriture qui peuvent être atteintes dans le cas des collectors et d'assurer une certaine homogénéité d'apparition des autres type de fourmis dans la fourmilliere.
La fourmi Generator essaie de se placer au centre de la fourmilière.
Sauf dans le cas où une largeur de 3 carrés ne peut-être garantie
entre la Generator et le bord de la fourmilière où les fourmis peuvent
apparaitre. Ceci dans le but de permettre la génération de fourmis,
même avec la plus petite fourmilière possible.
Dans le cas où deux chemins sont égaux (dans leur changement de direction),
la Collector calcule le nombre de superposition à chaque pas et choisis
le chemin qui cumule le moins de superposition avec d'autres entités.
Une fourmi collector sans nourriture cible se contente de sortir de la fourmilliere sans en faire plus.
Les Defensors calculent la bordure la plus proche et s'y rendent. Une fois sur la bordure, elles ne bouge plus jusqu'à ce que celle-ci bouge
La fourmi Predator tente simplement de retourner dans sa fourmilière en utilisant le meme algo que collector et defensors mais avec sa métrique de distance (euclidienne)

Méthodologie :

Nous avons mis en place un Git depuis le début. Nous étions ainsi capable de programmer ensemble (en général le vendredi en séance d'exercices) et seul quand nous avions du temps libre. Nous travaillons généralement ensemble pour résoudre les bugs les plus coriaces et seul le reste du temps avec des meetings régulier pour tenir l'autre à jour quand à l'avancement global, les prochaines étapes et le fonctionnement du code déjà écris. Nous nous sommes répartis les fonctions et modules en fonction des affinités. L'un de nous ayant plus d'affinité avec la structure du code et l'autre avec l'implémentation d'algorithme.
[Nathann] s'est chargé du module simulation ainsi que des machines d'état de chaque fourmis et de la génération et la mort des entity en faisant appelles aux fonctions que [Felipe] se chargeait de faire pendant ce temps-là. [Felipe] à fait les algorithmes pour permettre aux fourmis de trouver les nourritures, leurs maisons, les autres fourmis à attaquer ou encore la bordure la plus proche ainsi que le "pathfinding" ou encore le calcul de distance. Nous avions mis en place une série de tests automatique que git lançait à chaque push pour le rendu 1. les tests d'intégration automatique de la simulation étant plus compliqué à mettre en place avec un GUI, nous nous sommes contenté de faire des tests manuels sur les fonctions au fur et à mesure et de souvent tester l'exécution du programme dans son ensemble.
Les bugs les plus courants étaient l'oubli de vérifier si un pointeur contenait autre chose que nullptr ce

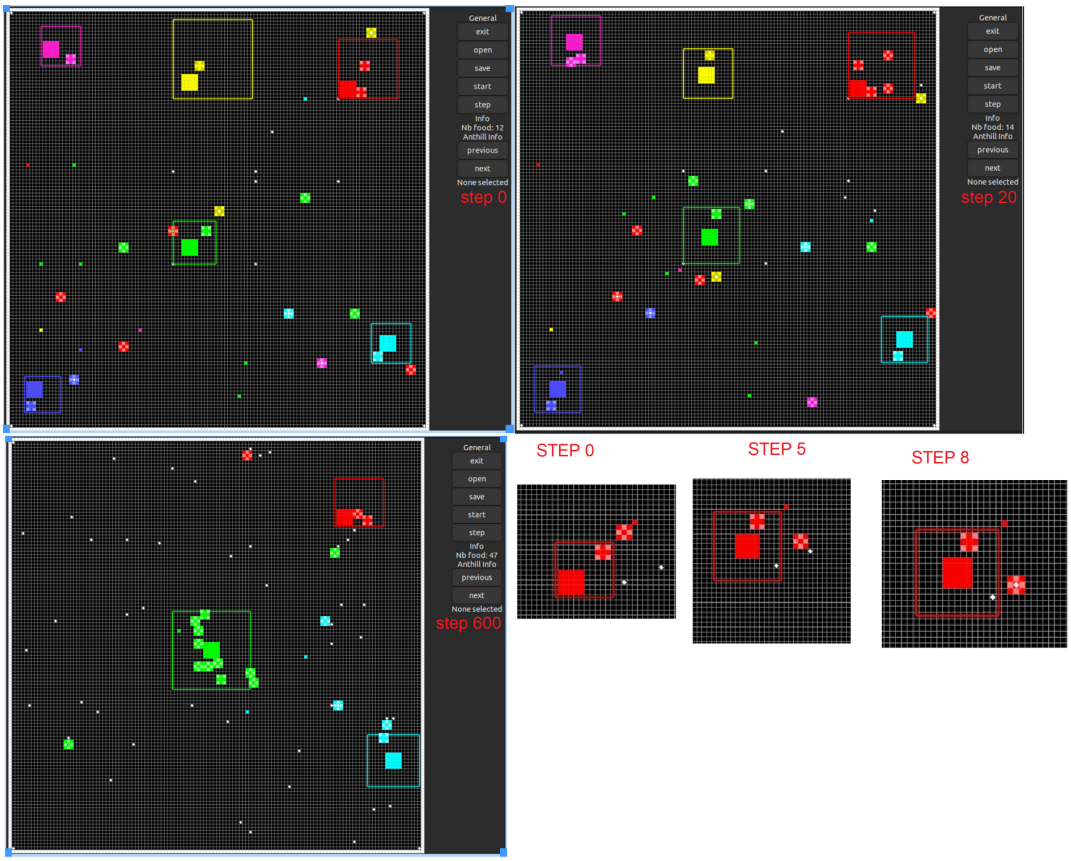
qui résultait en segmentation fault. Mais les plus compliquées a dé-bugger étaient les fautes d'algorithmique.

Conclusion :

Le projet est intéressant dans son ensemble. Nous nous sommes bien amusé même si le projet est relativement chronophage. Nous sommes content du résultat dans l'ensemble et pensons avoir utilisé judicieusement les outils de développement tel que git, WSL (qui supporte le GUI sur windows 11) et les pipelines de test.

Screenshot

Fichier c05.txt. (3 petit screenshot) et notre fichier (gros screen)



Step 20 :Les collector qui ne sont pas coincé se dirige vers des élément de nourriture (la vert clair est sur le point d'arriver a la fourmilliere.) on peut voir aussi un predator cyan qui retourne a sa fourmilliere (centre droite) et les générator qui se sont centré dans leurs fourmilliere
Step 600 :plusieurs fourmilliere sont morte de faim. la fourmilliere cyan est maintenant constrained par le bord et a spawné des prédator qui s'apprete a saccagé la fourmilliere verte n'as plus une fourmis, la cyan a rétrécit car la plus part de ces fourmis sont morte de vieillesse.

c05.txt, step 5 :la fourmilliere est resize, la collector est sur le point de prendre la nourriture et la defensors s'es décalé pour éviter le coin. step 8 : la collector a presque ramené la nourriture.