

DEEP CLEANER

By group 14



ORGANISATION DU CODE



<i>Timeline</i>
States

<i>State</i>
Time
Robots
Particles

<i>Particle</i>
Id
Position
Radius
ExplosionTimes

<i>Robot</i>
Id
Position
Angle
Score
Radius
Speed

- Utilisation de classes pour l'implémentation des méthodes de sérialisation
- Création d'une librairie de fonctions de trajectoires pour la réutilisation dans d'autres projets

SÉPARATION DU TRAVAIL



Jarod Streckeisen:

- UI: Drawing, scores, gestion des signals
- Backend: Logique principale du programme

Guillaume Courbat:

- UI: Fenêtre principale, animation du monde
- Backend: Clean du code
- Rapports

Timothée Van Hove:

- UI: Interactions avec les boutons, timer
- Backend: Menu principal, arguments CLI, JSON

Travail commun:

- Debug, calculs de trajectoire, logique de l'algorithme, structure et organisation du code

ALGORITHME DE GESTION DES ROBOTS



- Algorithme simple : Rotation + ligne droite
- Attribution de particules uniques à chaque robot
- Adaptation des mouvements en fonction des contraintes

RÉTROSPECTIVE



Les plus:

- Le partage des tâches efficace
- Le programme fonctionne selon le cahier des charges
- Algorithme simple, mais fonctionnel
- Affichage fluide
- Backend:Ascii art

Les moins:

- Manque de temps pour implémenter un algorithme plus intéressant
- Perte de temps à debugger
- L'organisation du code