

# Rapport final

## 1. Évolution de l'organisation du code et exécution :

- **Évolution** : Nous n'avons pas modifié les structures de données, jugées suffisamment adaptées. Seule la logique déterminant l'état de la partie (WON/LOST/ONGOING) a été ajustée pour intégrer les spécificités des modes Guidage et Construction. Le reste du travail a porté sur l'ajout des fonctionnalités manquantes liées à ces deux modes.

- **Exécution** :

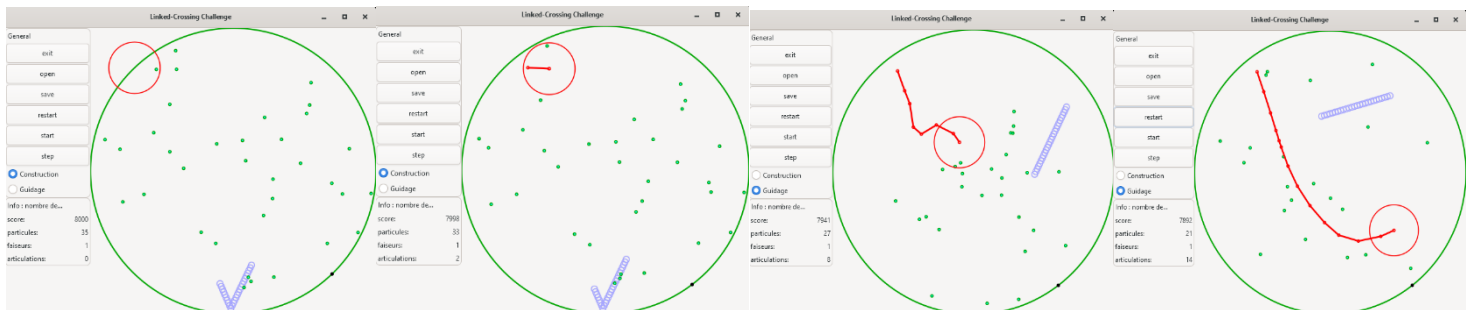
- **t22.txt**

Au début, déplacer la souris crée correctement une zone de capture sur l'arène, avec l'affichage du but noir positionné en diagonale inverse, comme attendu.

Lors de la capture d'une particule, celle-ci devient la racine de la chaîne. Le cercle de capture se recentre automatiquement sur la dernière articulation, permettant ainsi la construction progressive.

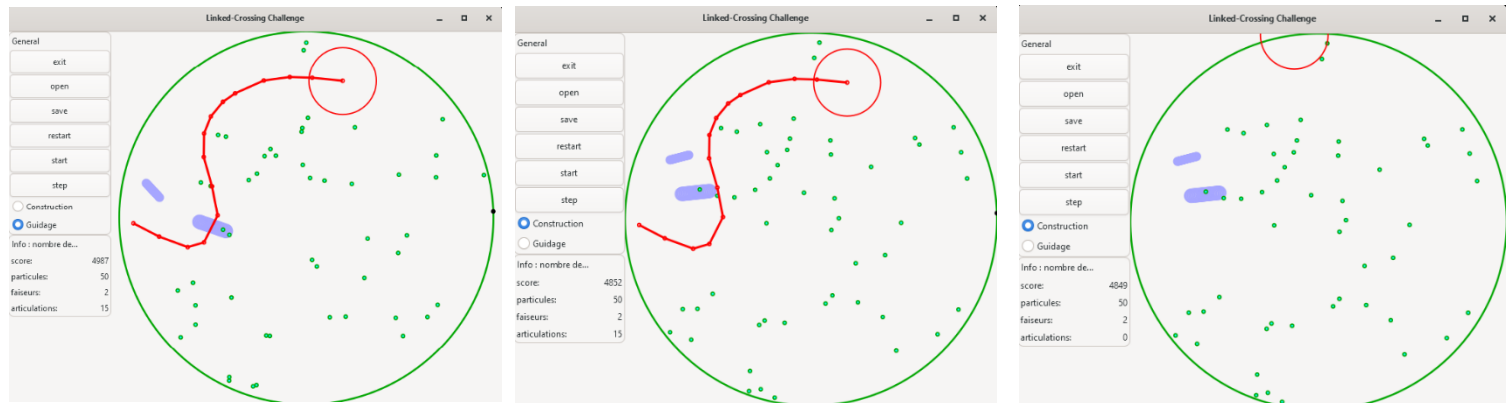
Les informations affichées à l'écran (score, nb particules....) sont mises à jour de manière cohérente à chaque étape.

Enfin, l'activation du mode Guidage exécute correctement l'algorithme prévu, en orientant les articulations vers la cible de manière fluide et attendue.



- **t35.txt**

- Dans l'état initial, la chaîne est affichée correctement et ne bouge pas, conformément au mode Construction.
- À un score de 4852, un faiseur s'approche de la chaîne, précisément de l'articulation 5 (la sixième), sur le point de la toucher.
- À 4849, le faiseur entre en collision avec cette articulation : la chaîne est alors détruite et le jeu se poursuit normalement.
- Une nouvelle zone de capture s'affiche sur l'arène, indiquant que l'utilisateur peut commencer à construire une nouvelle chaîne.
- Les informations à l'écran sont mises à jour de façon cohérente tout au long de cette séquence.



## 2. Méthodologie de travail, contribution individuelle et conclusion :

- **Activité individuelle :**

- Module tools :
- S2d / Pol / Cart / angleNormalise/ reflect : ensemble
- Cercle/ intersection : Adam
- Distance / operator+= / operator+/operator-/projection/ opp/ drawCircle/ drawLine : Yassine
- Module Graphic : Yassine
- Module mobile +capture : Adam
- Module chaine +guidage : Yassine
- Module jeu : ensemble
- Module Gui : ensemble

- **Méthodologie de travail :**

Nous avons utilisé GitHub avec VSCode pour collaborer. À chaque étape, nous lisions l'énoncé ensemble, définissions une stratégie, puis divisions les tâches. Pour les rendus 2 et 3, chacun a continué sur les modules déjà travaillés au rendu précédent, ce qui a facilité l'ajout des nouvelles fonctionnalités.

Environ 80 % du travail s'est fait côte à côte, ce qui a permis une entraide efficace et une meilleure compréhension mutuelle du code. Le reste a été fait à distance. Nous testions nos modules via le scaffolding, et l'intégration s'est faite sans difficulté grâce à une planification claire.

### **Bugs rencontrés :**

Le problème le plus fréquent venait d'un manque de compréhension du fonctionnement des « signal handlers » avec la boucle de « loop\_conn », ce qui causait des comportements inattendus lors des changements de mode et utilisations des boutons. Après plusieurs tests et questions sur ED, nous avons réussi à corriger cela.

### **Usage d'IA :**

Nous avons utilisé ponctuellement des outils d'IA (chatgpt et deepseek), principalement pour comprendre certains messages d'erreur liés à gtkmm et certaines fonctions données dans le module gui. Cela nous a permis de gagner du temps, notamment dans le débogage de l'interface.

- **Conclusion :**

Nous sommes plus que satisfaits du résultat final, tant au niveau de la robustesse que de la performance du code. Ce projet nous a permis de développer de vraies compétences de travail en équipe, en apprenant à organiser notre collaboration efficacement, mais aussi à progresser par essais et erreurs, ce qui a renforcé notre autonomie et notre compréhension des outils utilisés comme gtkmm. Enfin, nous avons particulièrement apprécié la réactivité sur ED, qui a grandement contribué à clarifier les zones d'ombre tout au long du projet.