



Rapport de projet

Projet Intelligence Artificielle

Eva DROSZEWSKI
Sara IKAN
Anesie MARTINIANI
Alix PRATABUY

| | |
|---------------------------------|----------|
| I - Points forts..... | 2 |
| II - Points faibles..... | 2 |
| III- Lacunes..... | 3 |

I - Points forts

Durant ce projet, nous avons su mettre en place des solutions techniques adaptées.

Pour contourner les limites de notre méthode [rechercher_palet](#), nous avons développé un mode de recherche basé sur les positions initiales connues de trois des palets du terrain. Cette stratégie a été cruciale pour maximiser la collecte de points. En effet, grâce à elle, nous étions assurés d'obtenir au minimum trois palets à chaque manche.

De plus, nous avons su trouver une solution alternative lorsque nous avons compris que nous ne pourrions pas utiliser le capteur de couleur. En effet, nous avons choisi de mettre en place un système de boussole, pour que notre robot puisse s'orienter sur le terrain, sans utiliser les lignes de couleur.

Notre point fort principal est notre performance lors de la compétition, car notre robot a réussi à saisir jusqu'à quatre palets, ce qui correspondait largement à nos attentes. Ceci a contribué à notre troisième position au classement général.

Une organisation méthodique nous a permis de gagner en efficacité tout au long du projet.

Contrairement à une approche impulsive qui consisterait à coder rapidement sans réflexion préalable, nous avons choisi de consacrer du temps à la mise en place d'une stratégie globale. Nous avons pour cela créé un automate afin d'avoir un visuel sur les interactions entre méthodes, ainsi qu'un plan de développement détaillant le but de chaque méthode. Cela nous a permis de poser des bases solides pour le projet, évitant ainsi des ajustements majeurs à un stade avancé, et nous permettant une réalisation de la classe Main relativement rapide.

Une autre réussite organisationnelle notable a été la vérification rigoureuse de chaque fonction avant son intégration dans le système principal. Chaque élément du programme a été testé individuellement pour s'assurer de son bon fonctionnement. Cette approche a permis d'éviter des erreurs cumulatives et difficiles à repérer dans les différentes classes.

II - Points faibles

Durant ce projet, certains choix ou réalisations n'ont pas produit les résultats attendus. Nous avons décidé de regrouper des aspects en deux catégories principales : les points faibles d'ordre technique et ceux liés à l'organisation.

Les faiblesses techniques sont principalement dues à des problèmes de détection et de traitement des données par les capteurs. Ces difficultés ont affecté la capacité du robot à repérer correctement les palets sur le terrain, limitant ainsi son efficacité optimale pendant la compétition.

Nous n'avons pas réussi à rendre notre méthode [recherche_palet](#) optimale. En effet, le problème que nous avons rencontré était que le robot détectait des discontinuités dans les murs, interprétant ces interruptions comme des palets inexistants. Nous avons eu des difficultés à déterminer un seuil efficace pour filtrer ces fausses détections, ce qui a conduit à une sur-interprétation au niveau des murs notamment au niveau des angles. Cela a rendu la détection imprécise et diminué les performances du robot pendant la compétition.

Avec plus de temps, nous aurions pu utiliser les coordonnées des palets, et nous aurions pu trouver un moyen de distinguer efficacement les murs, les robots adverses, et surtout les palets à attraper.

Les problèmes organisationnels ont également eu un impact significatif sur la progression et la qualité de notre travail. Certaines décisions prises en amont ou la gestion de notre temps ont entraîné des pertes d'efficacité.

Une part trop importante de notre temps a été consacrée à comprendre le fonctionnement de Git et essayer de résoudre des problèmes liés à Eclipse, pour finalement ne pas l'utiliser.

Un autre point faible que nous pouvons souligner est que nous n'avons pas pris le temps de consulter la Javadoc dès le début du projet, ce qui a retardé notre découverte de fonctions prédéfinies. Par conséquent, nous avons passé du temps à coder certaines méthodes qui auraient pu être directement intégrées, ce qui a inutilement rallongé le temps de développement.

De plus, nous avons sous estimé la dose de travail pour la classe Capteurs, qui nous a posé beaucoup de problèmes. En effet, dans cette classe, les méthodes dépendent de données collectées par les capteurs, et nous avons découvert tardivement que certains d'entre eux étaient défectueux. Nous aurions pu tester dans un premier temps tous les capteurs, afin de nous permettre d'identifier les dysfonctionnements dès le départ. Nous aurions alors pu ajuster notre stratégie et éviter ces pertes de temps considérables.

III- Lacunes

Avec du recul sur le projet, nous pouvons identifier certaines lacunes que nous aurions pu éviter.

Tout d'abord nous pouvons identifier certaines lacunes techniques.

Nous n'avons pas utilisé Git tout au long du projet. Cela a ralenti la gestion des versions du code car nous devions l'envoyer manuellement à chaque membre du groupe dès que nous le modifiions.

De plus, l'affichage de la console était limité à Sully, ce qui a rendu les tests et les débogages plus longs. Nous aurions dû nous assurer que l'affichage était accessible sur l'ordinateur, afin de faciliter le suivi en temps réel des exceptions.

Concernant l'organisation, plusieurs aspects auraient dû être mieux anticipés.

Nous n'avons pas pris le temps de vérifier le code avant le premier combat, ce qui a provoqué des dysfonctionnements avec les pinces et les délais. De plus, après avoir remarqué ce problème, nous n'avons pas demandé de temps mort pour résoudre la situation, ce qui a affecté les performances. Demander un temps mort aurait permis de rectifier rapidement le problème.

Pour les deux premiers matchs, nous n'avons pas prévu de bouton d'activation pour démarrer le robot immédiatement, ce qui a entraîné un temps d'attente inutile.