Université Grenoble Alpes - L3 MIASHS

# Projet Intelligence Artificielle

## Plan de tests

Théo Laigle Délyan Zergua

Année 2022-2023

Dans ce document, nous présentons la stratégie que nous allons mettre en place sur le plan expérimental pour tester les fonctionnalités de notre robot. La stratégie devra nous permettre d'identifier nos erreurs, nos oublis mais aussi ce qui a bien fonctionner.

Comme nous le savons, en robotique, il y a beaucoup d'incertitude. Cette incertitude est causée par ces capteurs mais aussi des actionneurs. Ces instruments qui lui permettent de percevoir et d'agir dans son environnement, ne sont pas fiable à 100%. En effet, nous ne pouvons pas affirmer avec certitude que le robot réalisera à tous les coups la même séquence d'actions pour une situation donné.

Pour tenter de réduire le risque d'imprévus, nous allons systématiquement réaliser un grand nombre de fois la même séquence d'actions. Cette stratégie nous permettra de quantifier le pourcentage d'erreur de chaque fonctionnalité. Bien évidemment, plus on fait d'essais pour une même action, plus notre taux d'erreur sera précis. Pour le moment, nous n'avons pas un nombre exact de répétitions par fonctionnalité. Cependant, il faut au minimum une quinzième pour commencer à avoir des résultats fiables.

Face aux erreurs, durant nos tests, nous avons systématiquement utilisé la même technique pour les réparer. Des dépannages grâce à des tests unitaires (tests successifs de petits bouts de code) pour cibler le problème et tenter de le résoudre ou de l'optimiser.

Notre phase de test s'est divisée en trois étapes :

1) Le test un par un des capteurs du robot
2) Le test de notre stratégie initiale
3) Le test de notre stratégie finale

# Test des capteurs

Dans un premier temps, nous testerons la fiabilité des capteurs du robot pour comprendre au mieux comment les utiliser. C'est-à-dire, minimiser les erreurs qui peuvent être faites avec ce capteur.

## Le capteur ultrason

Pour ce capteur, nous commencerons par le tester à l'aide d'une méthode java qui permettait d'afficher en continue les distances mesurées par le robot sur sa brick EV3.

Cela nous permettra de voir la distance maximale que le robot affiche, la distance minimale, la distance minimale pour détecter un palet, ce qu'il affiche lorsqu'il ne détecte rien… Après ces tests, nous cernerons les limites de ce capteur pour pouvoir les prendre en compte dans la construction de notre stratégies finale.

## Le capteur couleur

Pour ce capteur, nous réaliserons des tests plutôt longs par rapport au capteur ultrason.

Pour ce capteur, il y a plusieurs façons de récupérer la couleur.

Tout d'abord, nous testerons la fiabilité de chacune des techniques pour récupérer la couleur du sol avec le capteur.

Nous écrirons une méthode java qui permettra d'afficher le code RVB sur la brick EV3. Avec cette méthode, nous récolterons 5 mesures par couleur puis nous ferons la moyenne de ces 5 mesures pour chaque couleur. Après cela, nous aurions un code RVB de référence pour chacune des 7 couleurs présentes sur la table.

## Test de notre stratégie initiale

Nous allons essayer seuls chacune des trois grandes méthodes : allerVersPalet, allerAuBut et chercherPalet.

Nous commencerons par chercherPalet car cette méthode est déterminante pour notre stratégie. Pour cela, nous mettrons notre robot sur le terrain et disposerons aléatoirement des palets autour de lui. On demandera au robot de faire sa recherche et de nous afficher les distances mesurées et de nous indiquer la plus petite d'entre elles. Nous vérifierons enfin ces résultats en mesurer à la règle les distances réelles entre le robot et les palets. Nous espérons que le robot sera capable de s'orienter en direction du palet le plus proche.

Nous testerons ensuite allerVersPalet dans deux situations : en situation normale, c'est-à-dire quand un palet est correctement détecté et que le robot et bien orienté dans sa direction, puis dans une situation piège où nous décalerons volontairement le palet pour voir si le robot effectue correctement son petit balayage et arrive à le retrouver.

Enfin, nous testerons la méthode allerAuBut. Pour cela nous nous disposerons des obstacles entre le robot et le camp adverse pour voir s'il est capable de les éviter. Nous ferons cette opération dans des disposition différentes, afin voir si le robot arrive a s'orienter vers le camps adverse, quel que soit l'angle de départ.

## Test de notre stratégie finale

Pour tester la stratégie basée sur les couleurs, nous mettrons le robot en situation de tournois et nous lancerons le début du code, puis en faisant les corrections nécessaires, nous incorporerons d'autres morceaux du code jusqu'à obtenir le déroulé complet.

Idée de Théo, on commence par dessiner un automate très simple, très naïf, qui ne prend pas en compte les erreurs potentielles. Puis, nous testerons cet automate, un grand nombre de fois et nous relèverons les erreurs qu'il y a eu pour ainsi voir les endroits dd lautomate qu'il faut rendre plus robuste.