

1. Introduction :

1.1 Contexte

Dans le cadre d'un cours sur l'intelligence artificielle, notre groupe d'étudiants doit concevoir un programme dans un système embarqué permettant à un robot de détecter, manipuler et déplacer des palets. Nous utilisons le langage Java sur l'IDE Eclipse. A l'issue du semestre, une compétition sur plusieurs matchs a lieu, opposant les autres groupes de la promo. Le gagnant est celui qui arrivera à déplacer le plus de palets dans le temps imparti.

1.2 Objectif principal du plan de développement.

Ce plan de développement permet de définir la direction que doit prendre notre projet, de planifier les différentes étapes de développement et de respecter les délais imposés. De plus, il permet d'améliorer la communication entre les membres du groupe et le commanditaire.

2. Analyse de la situation

2.1 ressources à disposition

- Nous avons à notre disposition un Tribot EV3 LEGO Mindstorms dans lequel nous implémentons un programme nous permettant d'interagir avec le robot de manière asynchrone. Nous avons également accès à un terrain dans lequel nous pouvons tester notre robot.

2.2 limites

- Les capteurs ont des limites avec lesquelles nous devons composer.
- A partir du calcul de la distance entre le robot et l'obstacle, être capable de définir si cet obstacle est un palet ou un autre robot ou un mur.

2.3 composants

Sélection des capteurs EV3 utilisés dans le programme :

Moteurs / Capteurs	Ports
Roues	Droite : M.A
	Gauche : M.B
Pinces	M.D
Ultrason	S.4
Toucher	S.3

3. Objectifs de développement

Ce projet vise à ce que le robot soit capable de ramasser un maximum des palets sur le terrain en un minimum. Pour cela, on cherche à:

- Écrire un code efficace qui permettra au robot de détecter, localiser et ramasser les palets tout en maximisant sa vitesse et minimisant ses erreurs.
- Élaborer une stratégie qui aide à avoir le score maximal possible en moins de temps.
- Rédiger les livrables demandés dans les délais impartis (cahier de charges, plan de développement, plan de tests, code source et documentation ainsi que le rapport final).

4. Analyse des besoins

Pour mieux gérer la réalisation de ce projet, une analyses des besoins nous a aidé à conclure que:

- Nous avons une contrainte de temps, des capteurs dont on dispose ainsi qu'une forme de robot qui nous sont imposés.
- La table sur laquelle tourne le robot a des caractéristiques (longueur, largeur et lignes de couleurs), qui nous permettent d'établir une stratégie de jeu.
- Les capteurs sont la clé qui permet au robot de savoir son état et positionnement afin qu'il puisse agir comme indiqué.
- Il est possible qu'il y ait une collision entre deux robots ou contre le mur et qu'il faut prendre des mesures afin d'éviter que cela arrive.
- La mise en place d'un échéancier est nécessaire afin de pouvoir rendre les livrables dans les délais et d'avancer sur le projet régulièrement.
- La communication entre les membres de l'équipe facilite le travail et favorise l'avancement du projet (à travers la planification de réunions et distributions des tâches), pour cela on a opté pour Discord comme plateforme de communication.

5. Planification du projet

Phase	Tâches	Livrable	Date limite
Conception	Lister les besoins, objectifs, plan, stratégie, risques sur le plan de développement	Plan de développement	9/10/2023
Développement	Implémenter les classes et méthodes (décrites plus en détail dans le paragraphe en-dessous)	Programme	-----
Intégration	Rédiger le plan de tests après avoir implémenté le code	Plan de tests	20/11/2023
Recette	Dernières touches au code suivant les remarques du plan de tests	Code source final + documentation	27/11/2023
Evaluation	Rédiger le rapport final	Rapport final	4/12/2023

6. Stratégie de développement

Construire les classes nécessaires pour l'atteinte des objectifs, avec les méthodes suivantes :

- forward()
- backward()
- rotate()
- rotateTo()
- ouvrirPinces()
- fermerPinces()
- detecterCouleur()
- calculerDistance() en utilisant getDistanceMode()
- stop()
- isPressed() de la classe Button

7. Tests, validation et amélioration

Des tests continus sont essentiels pour détecter tout dysfonctionnement dans les performances du robot et pour vérifier que le programme est conforme aux spécifications et ainsi effectuer les ajustements nécessaires en fonction des résultats des tests.

- Les tests matériels incluent le test de la capacité du robot à ramasser et à transporter les palettes et à les placer du côté de l'adversaire, le second est le test du capteur de couleur du robot qui garantira que le robot se déplacera sur la bonne ligne colorée spécifiée par le programme, et le troisième matériel doit être testé le capteur du détecteur d'objets pour garantir que le robot est capable de distinguer les différents objets qu'il peut rencontrer, tels que les palettes, le mur ou un autre robot.
- Les tests logiciels incluent des tests continus du code, y compris de nouvelles classes, fonctions et constructeurs, pour garantir que le robot est capable de fonctionner et d'agir correctement sans aucune erreur et de pouvoir s'adapter à toutes les situations lors de la compétition finale.
- Tests de performances du robot pour vérifier la coopération entre les composants logiciels et matériels, qui testent le robot sur le terrain et garantissent qu'il est capable de fonctionner avec ce qui convient parfaitement à l'objectif du projet

8. Gestion des risques

La gestion des risques est le processus de traitement, de reconnaissance et de gestion des risques potentiels pouvant être rencontrés tout au long des travaux, elle est principalement composée de trois phases qui sont l'identification du risque potentiel, l'analyse puis le traitement du risque en fonction de sa type, et cela peut inclure des tests et une compilation continue de code pour limiter les pannes logicielles, et une sauvegarde complète du robot pour éviter toute panne matérielle.

Risques pouvant être rencontrés lors du développement du projet :

- Les problèmes techniques pouvant survenir, tels qu'une défaillance ou un retard dans la connexion du logiciel au matériel, peuvent entraîner une perte de temps ou l'impossibilité de défier l'adversaire pendant la durée de la compétition finale.
- Des problèmes matériels peuvent survenir tels qu'une panne d'une des pièces du robot, ou un dysfonctionnement des capteurs pouvant affecter l'attitude du robot pendant la compétition, ou peut-être la décharge de la batterie du robot.
- Les problèmes logiciels principalement représentés par l'implémentation du code (tels que des cas d'exceptions, des bugs de code ou des barrières ou objets non définis que le robot peut rencontrer lors de la compétition) peuvent également affecter les performances du robot,
- Les problèmes logiciels principalement représentés par l'implémentation du code (tels que des cas d'exceptions, des bugs de code ou des barrières ou

objets non définis que le robot peut rencontrer lors de la compétition) peuvent également affecter les performances du robot,,

9. Conclusion

Le plan de développement est un élément essentiel de la réussite de ce projet, il représente la synthèse du travail tout au long des 12 semaines pour atteindre les objectifs requis qui créent un programme permettant à un robot de détecter, manipuler et déplacer efficacement des palettes. Le succès du projet est mesuré par le biais d'une compétition dans laquelle le robot rivalise avec d'autres groupes de robots et le succès est mesuré par la capacité du robot à déplacer le plus de palettes dans un laps de temps spécifique. Notre rapport de plan de développement décrit les principales phases de notre travail de groupe sur le projet et donne un aperçu des étapes de développement et de préparation qui analysent la situation, les objectifs de développement, analysent les besoins, planification du projet, stratégie de développement, tests, validation et gestion des risques.

10. Documentation

https://koor.fr/Java/Tutorial/java_javadoc_introduction.wp#eclipse_javadoc