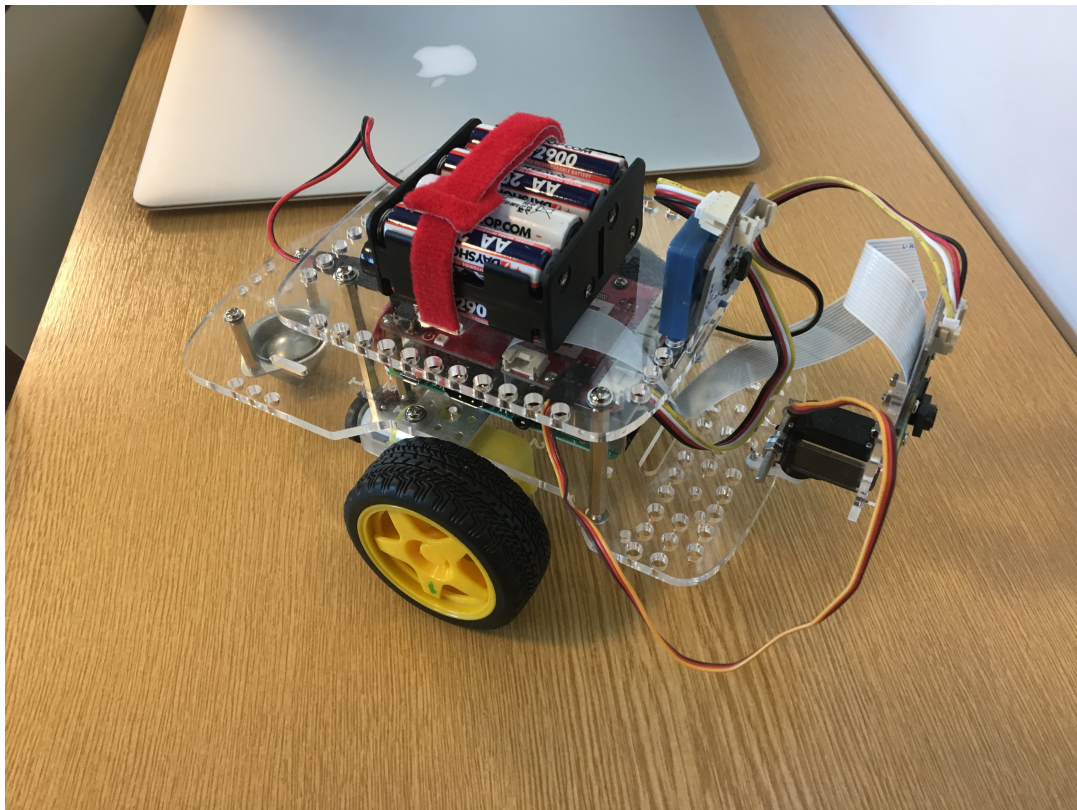


LICENCE D'INFORMATIQUE

Projet Robotique



Groupe FiveGuys
UE 2I013

Chargé de cours : M. BASKIOTIS
Chargé de TME : M. VENIAT

1^{er} Février 2019 — Mai 2019

Table des matières

1	Introduction	2
2	Rapport	3
2.1	Méthodes et outils	3
2.1.1	Agile/Scrum	3
2.1.2	Outils pour le travail d'équipe	3
2.1.3	Hiérarchie du code	
	4	
2.2	Mise en place de briques de base	5
2.2.1	Modélisation du réel	5
2.2.2	Mise en place de l'interface graphique 2D	5
2.2.3	Stratégies de base	5
2.2.4	Passage de la simulation au réel	5
2.3	Développement des stratégies	6
2.3.1	Stratégie carré	6
2.3.2	Stratégie détection de mur	6
2.3.3	Stratégie cercle	6
2.3.4	Stratégie contourner une porte	6
2.3.5	Stratégie détection balise	6
3	Remerciements	7
4	Conclusion	8

1 Introduction

Dans le cadre de l'UE 2I013, notre projet était de concevoir un logiciel permettant de contrôler un robot. Le projet était entièrement à notre charge aucun code ne nous était fourni. Le projet était construit de tel sorte que nos encadrants avait à la fois le rôle de nos superviseurs et de nos clients avec qui nous discussions des éléments a changer et de notre avancement dans le projet.

L'objectif principal de ce projet était d'une part d'être un premier contact avec un projet de robotique et une découverte des méthodes de développement en équipe.

Dans un premier temps, l'objectif est de mettre en place les briques de bases de notre projet comme le simulateur 2D et le modèle physique ainsi que de mettre en place nos outils pour le bon déroulement du travail de groupe.

Dans un second temps, une fois que notre modèle physique et notre simulateur étaient considéré suffisamment stable, nous avons commencé à développer les stratégies qui nous permettent de répondre aux challenges qui nous ont été soumis.

2 Rapport

2.1 Méthodes et outils

2.1.1 Agile/Scrum

Le cadre de notre projet n'était pas seulement le développement d'un logiciel de robotique, il avait également pour but de découvrir et mettre en place des méthodes efficaces pour le développement en équipe.

En effet, dans les projets que nous réalisions jusqu'alors nous ne nous occupions pas particulièrement de la façon dont nous allions gérer le développement de nos projets. Les codes à développer étaient relativement petit et guidé, en groupe de 2 à 3 personnes et sur quelques semaines seulement. Dans la plupart des cas nous avons donc réaliser nos projets d'un bout à l'autre pour obtenir directement notre rendu final avec une méthode dite "en V". Cette méthode consiste en un développement du projet de bout en bout afin à partir des exigences du "cahier des charges" (sujet) afin d'obtenir notre application. Le "retour client" se fait à la fin de celui-ci. Dans nos anciens projets cela ne posait pas problème mais dans le cadre de ce projet oui.

Ce projet était construit de manière à ce que nous développions la totalité du code sur tout le semestre. Il nous fallait donc pouvoir avoir un retour régulier sur l'avancement afin de pouvoir modifier ce qui ne va pas dans la bonne direction et pouvoir s'adapter aux nouvelles exigences pour celui-ci. Nous avons donc cherché à appliquer une autre méthode qui nous a été présentée en cours, la méthode "Agile/Scrum".

Schéma de la méthode Agile/Scrum.

Nous avons donc pour objectif de mettre en place chaque semaine un ensemble de tâches les plus petites et indépendantes possible les unes des autres nommées sprint. Les tâches devaient toutes être définies avec un maximum de précisions et en concertation avec toute l'équipe pour qu'il n'y ait pas de différence de résultat en fonction du membre du groupe qui l'accomplisse. Elles ont toutes pour but de remplir un objectif de démonstration concrète de l'avancement du projet au client en terme de nouvelles fonctionnalités. L'objectif final de cette méthode était que chacun des membres du groupe puisse choisir les tâches qu'il souhaite accomplir durant la semaine et qui seraient ensuite validées par un autre membre.

À l'issue de la semaine, nous présentions une nouvelle démonstration au client pour obtenir ses retours. Grâce aux retours client et au compte-rendu d'équipe rédigé dans la semaine nous pouvions mettre en place le sprint de la semaine à venir.

2.1.2 Outils pour le travail d'équipe

Git & GitHub

Pour pouvoir travailler de façon optimale et sécurisée ils nous fallait pouvoir développer sur un même code sans avoir à se le partager à chaque modification. Nous avons donc utilisé le logiciel de gestion de version Git afin de conserver une version commune du code la plus à jour afin que chacun puisse travailler dessus à tout moment. La centralisation du code nous permet d'éviter un maximum de conflits lors du développement du code et de conserver les versions précédentes du code au cas où nous aurions besoin de revenir sur une modification que nous avons apportée au projet et qui pose problème.

Trello

Pour mettre en place les différentes tâches du sprint de la semaine nous avons utilisé le logiciel Trello. Celui-ci nous a permis de rédiger nos cartes pour les différentes tâches que nous avons dans notre sprint. Dans un premier temps l'élaboration de nos sprints et la préparation de nos cartes n'étaient pas très au point ce qui nous conduisait à faire du travail en double où à une mauvaise compréhension du résultat attendu.

Par la suite, grâce aux premières réunions et aux premiers retours, nous avons pu améliorer la préparation de nos sprints. Nous avons donc pu avancer dans la préparation de notre simulateur avec une première interface et la mise en place de notre modèle.

Slack & Whatsapp

Python

2.1.3 Hiérarchie du code

MVC (Model-View-Controller)

Hiérarchie de notre code

Juste un schéma ?

2.2 Mise en place de briques de base

2.2.1 Modélisation du réel

2.2.2 Mise en place de l'interface graphique 2D

2.2.3 Stratégies de base

2.2.4 Passage de la simulation au réel

2.3 Développement des stratégies

2.3.1 Stratégie carré

2.3.2 Stratégie détection de mur

2.3.3 Stratégie cercle

2.3.4 Stratégie contourner une porte

2.3.5 Stratégie détection balise

3 Remerciements

4 Conclusion