

PROJET FIL ROUGE 2024-2025

Projet d'intégration des compétences acquises en 1A SRI Présentation du projet et Cahier des charges du PFR1

1 Organisation générale

Le Projet Fil Rouge (PFR) est décomposé en deux parties, la première partie (PFR1) est réalisée pendant le semestre 5 et la seconde au semestre 6 (PFR2). La philosophie est la même : se baser sur les compétences acquises lors du semestre, aller au-delà des TP, monter en compétences sur certains aspects techniques et transversaux, comme détaillés dans le tableau 1.

Tableau 1 : Fil rouge et organisation des enseignements en 1A SRI

Semestre 5 (cours concernés)	
UE Informatique Base de l'algorithmique et programmation impérative appliquée en Python	PROJET SRI 1 – PFR 1 (septembre – janvier) <ul style="list-style-type: none">Algorithmique et Programmation impérativeCœur du projet : traitements et représentations des données (image, texte) : langage C et UnixVisualisation et Interaction vocale (STT et TTS) : langage Python <p>Objectif = Développement et validation de briques logicielles de base pour une intégration dans le PFR2</p>
UE Programmation impérative Algorithmique avancée, Unix et langage C <ul style="list-style-type: none">Structures de données avancéesGestion de la mémoire, pointeursCompilation séparéeExécution de scripts Unix	
UE Ingénierie logicielle et système	
Semestre 6 (cours concernés)	
UE Programmation Orientée Objet Concepts de base POO & langage Java	PROJET SRI 2 – PFR 2 (février – mai) <ul style="list-style-type: none">Initiation à la robotique mobileAssemblage, manipulation et intégration sur un dispositif physiqueDéplacement en mode automatique,Déplacement en mode pilotage par l'utilisateurDonnées capteurs, localisation, navigation, interaction avec l'utilisateur
UE Outils de modélisation informatique UML (fin S5)	
UE Introduction aux systèmes interactifs IHM, Image et Son	
UE Modélisation robotique	
UE Gestion de projets	<p>Objectif : Réutilisation des modules du PFR1 et intégration sur une plateforme mobile (code embarqué vs distribué), communication des systèmes. C/Unix/Python/...)</p>
Aspect transversal	
<ul style="list-style-type: none">Communication externe orale/écrite avec le client et l'équipe pédagogique (présentations, rapports, démonstrations...);Gestion planning global et des différents étapes/jalons du projet (livrables, dépôt de code...)Communication interne (entre membres d'une même équipe) et gestion du groupe (5 à 6 étudiants)Outils de gestion de projets (Trello/Altasian, Git...)	

1.1 Équipe projet

Les équipes projet sont composées majoritairement de **5 ou 6 étudiants** suivant les effectifs. Elles doivent être constituées avant le **20 septembre 2024**.

Contraintes : les équipes doivent être **hétérogènes**, c'est-à-dire formées de membres provenant de cursus différents. La constitution des équipes doit donc être validée par l'équipe pédagogique avant l'affectation d'un tuteur.

Le **tuteur** fait office à la fois de *client*, de *conseil* et d'*évaluateur*. La communication avec votre tuteur est aussi un critère d'évaluation, de même que la communication au sein du groupe, l'avancée de la réalisation, la bonne répartition du travail, etc. Chaque groupe sera suivi par le même tuteur sur les deux parties du projet PFR1 et PFR2. Le groupe sera chargé également de **planifier des rendez-vous réguliers avec son tuteur** (et non l'inverse).

1.2 Équipe pédagogique

L'équipe pédagogique PFR est constituée d'enseignants de la formation : Isabelle Ferrané, Julien Pinquier et Julien Vanderstraeten. Pour le PFR2, Nahla Tabti interviendra sur les aspects en lien avec la plateforme robotique.

2 Scénario global

2.1 Vue d'ensemble

Chaque équipe développe les **fonctionnalités logicielles** nécessaires pour faire fonctionner à terme une **plateforme mobile** équipée d'un **ensemble de capteurs** (infrarouge, ultra-sons, caméra, lidar...) permettant de réaliser **différentes tâches**, dans un **environnement évolutif**. Voir la figure 1 pour un exemple de plateforme mis à votre disposition.



Figure 1 : exemple de plateforme mobile

Cette plateforme est assemblée, validée et testée lors de la partie PFR2 où chaque groupe reçoit une caisse contenant un ensemble de composants. Les choix de **configuration et d'architecture** doivent être justifiés. Un inventaire est fait en début et fin de projet. Le non respect du matériel fourni est pris en compte dans l'évaluation.

La plateforme pourra être déplacée selon le **mode pilotage** ou le **mode automatique**.

2.1.1 Mode pilotage

Le **mode pilotage** s'effectue en interaction avec l'utilisateur, soit manuellement soit vocalement :

- **Interface graphique ou un dispositif physique** : pour piloter le robot manuellement vous développez une interface graphique sur ordinateur ou un téléphone. Vous pouvez utiliser une **manette** (type joystick, non fourni).
- **Interaction vocale** : l'utilisateur peut également piloter le robot vocalement (à l'aide de la brique logicielle développée dans le PFR1). Les **commandes vocales** peuvent être de simples commandes de déplacement ou des **commandes plus évoluées** tenant compte de la configuration de l'environnement (objets disponibles et obstacles potentiels) ou encore selon une trajectoire spécifique. Quelques exemples de commandes sont donnés dans le tableau 2.

Les **requêtes** de ce type sont captées avec le **micro** d'un téléphone ou d'un ordinateur et sont traitées, c'est-à-dire (1) transcris (= transformées en texte, on parlera de **requête-texte**) et (2) interprétées (= traduite en une commande ou succession de commandes à exécuter on parle de **requête-commande**).

Avant de réaliser la tâche demandée (exécuter la requête-commande), il faut la valider pour pouvoir l'exécuter. En cas d'impossibilité, le système réagit en conséquence soit en formulant un message via la synthèse vocale (*« je n'ai pas trouvé de balle rouge », « je dois contourner un obstacle », « déplacement impossible » ...*), soit par un autre moyen (événement sonore, bip ...). Le robot doit décider de l'action à faire en cas de problème ou en cas de choix et peut avoir recours à l'utilisateur si besoin (*« Que dois-je faire ? »*).

Tableau 2 : Exemples de commandes vocales à traiter

Commandes basiques	<i>avance ; recule ; tourne à droite ; tourne à gauche ; fais demi-tour ; arrête ; ...</i>
Commandes complexes tenant compte de l'environnement	<i>avance de deux mètres ; avance jusqu'à la balle rouge ; passe entre la balle rouge et le cube bleu ; contourne l'obstacle par la droite ; reviens au point de départ ; ...</i>
Commandes de type trajectoire	<i>trace un carré ; trace un cercle ; avance en zigzag ; effectue un déplacement en spirale ; trace un cercle autour de la balle rouge ; ...</i>
Enchaînement de commandes	<i>avance de deux mètres, tourne à droite et trace un cercle ; ...</i>
Autres possibilités	<i>contourne l'obstacle sans passer à côté de la balle rouge ; passe à côté de la balle rouge puis à côté de la balle bleue ; recherche une balle rouge ou bleue ; ne te positionne pas à côté de la balle verte ; combien vois-tu de balles ? ...</i>

Un exemple de mode pilotage vocal avec successions de commandes peut être consulté [ici](#).

2.1.2. Mode automatique

Plus évolué, ce mode suppose que la plateforme dispose de capteurs et d'actionneurs permettant :

- la **localisation** dans l'environnement,
- le **déplacement**,
- la **détection** d'obstacles (passage impossible) et d'objets spécifiques (balles et objet de couleur présents dans l'environnement et éventuellement recherchés par l'utilisateur),
- et le **contournement d'obstacles**.

La localisation et le déplacement du robot sont représentés à l'écran sur une carte construite au fur et à mesure (phase de cartographie). Ceci permet une visualisation du périmètre dans lequel évolue le robot (espace libre et obstacles détectés) ainsi que du déplacement réalisé (superposé sur la carte).

Le robot peut identifier un ensemble d'objets par leur **forme** et leur **couleur**. Pour cela il est nécessaire d'analyser les images captées par une caméra embarquée sur la plateforme mobile. Cette fonctionnalité est développée dans le cadre du PFR1 et réutilisée dans le PFR2.

Enfin, le robot peut évoluer, *a minima*, dans un **environnement fermé**, connu à l'avance. Il doit surtout pouvoir évoluer dans un **environnement ouvert** (modifiable, différent d'une session à l'autre...). Une gestion de l'historique des déplacements est faite pour un éventuel « rejeu » avec par exemple des commandes du type *refait le même déplacement*.

2.2 Décomposition en 2 parties

PFR1 : La première partie du PFR consiste donc à **développer les fonctionnalités logicielles de base** (= cœur du projet) pour simuler le déplacement du robot, analyser les images pour détecter des objets et leurs propriétés (balles, construction en lego, taille, couleur...) et gérer l'interaction vocale.

PFR2 : La seconde partie du projet vise (1) à construire la plateforme mobile et faire les choix de capteurs en fonction de ce qui est disponible. Une fois la plateforme validée par l'équipe pédagogique, il faut (2) intégrer les briques du PFR1, (3) développer les fonctionnalités relatives au déplacement physique du robot (détection et contournement d'obstacles, cartographie...) (4) gérer la communication entre l'ordinateur, l'utilisateur et le robot (via les moyens mis à disposition, bluetooth, wifi...), etc. Des compléments seront donnés en janvier 2025 sur l'organisation de cette partie (calendrier, jalons, livrables...).

3 Cahier des charges PFR1

Cette partie est développée principalement en **langage C** dans un environnement **Unix/Linux**, sachant que le langage C peut exécuter des commandes système, des scripts Unix ou Python. Elle vient illustrer et compléter les enseignements d'Informatique et de programmation impérative du S5 (cf. Tableau 1).

3.1 Traitement des requêtes utilisateurs

Les commandes vocales ont vocation à être transcris (= passer du signal au texte). Le module nécessaire pour cela vous est fourni (code Python sur Moodle). Il convient donc que chaque groupe développe la **fonctionnalité « Conversion Requête »** qui permette de traiter une **requête-texte** pour la transformer en **requête-commande** (fonction à exécuter, paramètres à prendre en compte, ... cf. Tableau 3).

3.1.1 Conversion Requête-texte en Requête-commande

La mise au point du module de **conversion d'une requête-texte en requête-commande** passe par une étude des formulations possibles, du vocabulaire utilisable ainsi que des commandes à exécuter. **Ces informations ne doivent pas être écrites en dur dans le code.** Il faut passer par des **fichiers de configuration**. La possibilité de changer de langue doit être réalisée, comme illustré dans le tableau 3.

Tableau 3 : exemple de conversion multilingue

Requête-texte français	Requête-commande associée	Requête-texte anglais
avance	forward(dist = default_distance)	move forward
avance de 2 mètres	forward(dist = 2)	move forward by two meters
localise la balle rouge trouve la balle rouge	where(object=ball, color=red)	locates the red ball finds the red ball

Des **paramètres de configuration** peuvent indiquer la langue choisie et les chemins vers les fichiers contenant les ressources lexicales (lexique = mots du vocabulaire, synonymes... propriétés = catégorie des mots color, object, command...), lien avec les commandes (*avance*, *forward()* ; *rouge*, *color=red*). Les fichiers correspondants sont chargés en début de session.

Pour simplifier et permettre une visualisation du déplacement du robot, la librairie python turtle utilisée en TP d'informatique peut être utilisée. Un script python contenant les requêtes-commandes peut alors être exécuté.

3.1.2 Complexité des requêtes-texte

Les requêtes prises en compte sont de complexité croissante, ce qui nécessite une prise en compte des structures syntaxiques (construction des phrases) évoluées (cf. Tableau 2).

3.2 Traitement des images de l'environnement

Dans le volet PFR2, le robot est capable de prendre des photos de son environnement. Ces photos sont analysées pour détecter des objets dans l'environnement. Ces objets sont caractérisés par une **forme**, une **couleur** et éventuellement une **taille ou leur nombre**. Les images couleur sont traitées pour localiser l'objet et déterminer s'il correspond au critère recherché. La Figure 2 donne des exemples d'images qui pourront être traitées.

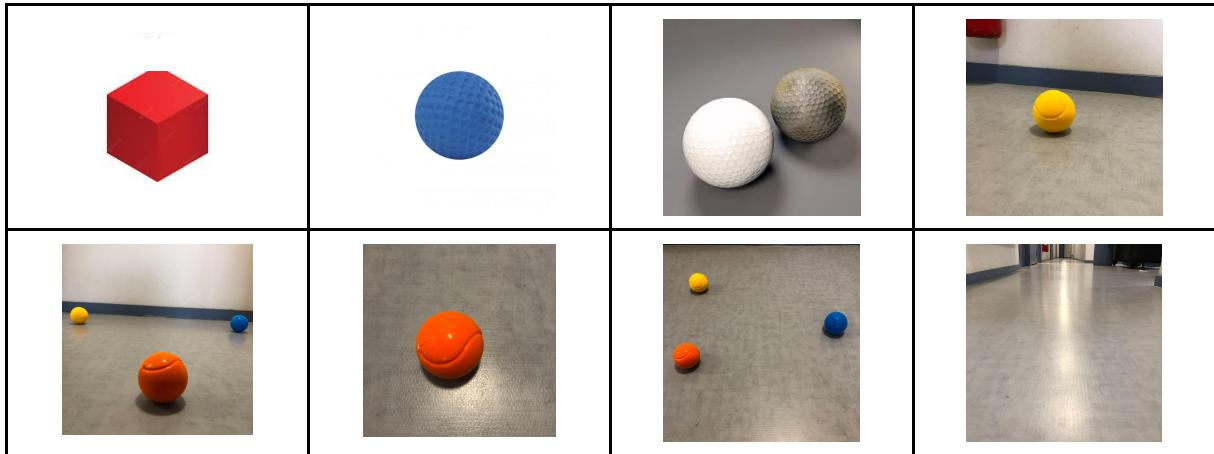


Figure 2 : exemple d'images pouvant être traitées

Les fonctionnalités de **Détection, Localisation et Caractérisation des objets de la scène** sont développées et testées dans la première partie du projet à partir d'un ensemble d'images fournies (et enrichies par chaque groupe). Les coordonnées du centre de l'objet ou des objets détectés ou bien la boîte englobante correspondante sont renvoyées à la partie appelante pour construire la réponse destinée à l'utilisateur (PFR1) ou déterminer l'action à faire (PFR2). Une première approche peut être basée sur les histogrammes de couleurs.

3.3 Configuration

Les différentes fonctionnalités développées pendant la première partie doivent être généralisables à d'autres situations, c'est pourquoi il est nécessaire de permettre à l'utilisateur de configurer chacune des fonctionnalités :

- Langue, lexique... pour la conversion de requête-texte en requête-commande ;
- Paramétrisation du traitement d'images (bit de quantification, plages de couleurs...) ;
- Génération d'un fichier de log comportant les résultats obtenus, même si ceux-ci sont visualisés, il faudra en garder la trace pour être traité dans la partie 2.

Le périmètre du projet, les stratégies et les méthodes à utiliser sont à valider avec votre tuteur. Ces modules peuvent être utilisés dans le PFR2 via les possibilités de communication entre le langage Python et le langage C.

4 Livrables et planning

Le PFR1 est décomposé en plusieurs étapes, comportant des dates de rendu (jalons) et des documents à rendre (livrables) (cf. Tableau 4).

Dès l'affectation du tuteur, chaque groupe doit prendre rendez-vous avec lui pour demander des précisions et des clarifications du sujet et définir les contours de ce qui est demandé et réalisable dans le temps imparti.

Le premier objectif est de rendre un dossier de spécifications. Des séances de TD dédiées vous permettront également de voir quelles sont les attentes en termes de gestion de projet.

Tableau 4 : Planning du PFR1

Date	Jalon/livrable
Semaine du 9/09/2024	Mise à disposition du cahier des charges (Partie 1)
17/09/2024	Présentation générale du projet, réponses aux questions et validation des groupes
18/09/2024	TD FIL ROUGE n°1 : Réponses aux questions sur l'analyse du cahier des charges et sur la constitution d'un dossier de spécifications
20/09/2024	Constitution/validation des groupes
25/09/2024	TD FIL ROUGE n°2 : travail en groupe autour du dossier de spécifications
16/10/2024	TD FIL ROUGE n°3 : finalisation dossier de spécification
18/10/2024	Livrable 1 : dépôt sur Moodle du dossier de spécifications incluant le planning prévisionnel et l'organisation prévue (répartition individuelle des tâches)
4/11/2024	Feu vert pour la conception et le développement
4/11/2024 au 20/12/2025	Décomposition en modules, codage et tests
4/12/2024	TD FIL ROUGE n°4 : principe et attentes de la validation
Semaine du 6/01/2025	Intégration des différentes parties en vue de la validation
Semaine du 13/01/2025	Finalisation : préparation de la validation et des livrables Livrable 2 : rapport global et bilan individuel Livrable 3 : dépôt code structuré, lisible et commenté et manuel d'utilisation associé
24/01/2025 à confirmer	Validation du PFR1 (démonstration scénarisée) et dépôt des livrables