

Mémoire d'Approbation de Projet

RobotFoot

Présenté à

Ruben Gonzalez-Rubio et Ahmed Khoumsi

Département de génie électrique et de génie information



25 février 2013

Table des matières

1	Sommaire exécutif	1
2	Historique et environnement du projet.....	1
2.1	Contraintes.....	2
2.2	Cadre logique	3
2.3	Description technique.....	4
2.3.1	Conception physique	4
2.3.2	Conception logicielle.....	9
2.4	Organisation du projet.....	13
2.5	WBS.....	1
2.6	Structure organisationnelle de la conception physique : OBS.....	1
2.7	Charte des responsabilités.....	1
3	Échéancier du projet et gestion du temps	2
3.1	Charte de Gantt.....	2
4	Coût du projet et gestion des coûts	3
4.1	Budget global	3
4.2	Suivi budgétaire pro forma	4
5	Gestion des communications.....	4
5.1	Coordonnées des principaux acteurs.....	4
5.2	Rapports — pro forma	4
6	Gestion des risques et suivi.....	4
7	Santé et sécurité.....	6
7.1	Procédures de tests à risques	6
7.2	Analyse d'impacts sur la santé, l'environnement et le public	6
8	Recommandations et conclusions.....	8

Liste des figures

Figure 1 : Schéma haut niveau des composantes électroniques.....	4
Figure 2: Schéma haut niveau des modules logiciels.....	10
Figure 3 : Schéma du processus menant au choix du comportement.....	11
Figure 4 : Machine à états finis des comportements.....	11
Figure 5 : Schéma du calcul de la distance.....	12
Figure 6 : WBS.....	1
Figure 7 : OBS.....	1

Liste des tableaux

Tableau 1 : Cadre logique	3
Tableau 2 : Comparaison des batteries	5
Tableau 3 : Comparaison des cartes de contrôle.....	6
Tableau 4 : Comparaison des types de servomoteurs.....	8
Tableau 5 : Comparaison des servomoteurs intelligents.....	9
Tableau 6 : Résumé des composants électroniques.....	9
Tableau 7: Charte des responsabilités	2
Tableau 8 : Sources de revenus	3
Tableau 9 : Dépenses planifiées.....	3
Tableau 10 : Coordonnées des principaux acteurs	4
Tableau 11 : Gestion des risques	5
Tableau 12 : Analyse du risque	7

1 Sommaire exécutif

RobotFoot sera l'implémentation du premier prototype abordable d'une plate-forme robotique humanoïde d'un joueur de soccer de l'Université de Sherbrooke dont le but à long terme est de participer à la compétition internationale Robocup dans la catégorie de taille moyenne. De plus, cette plate-forme pourra être utilisée dans des projets du département de génie électrique et informatique de l'Université et dans les écoles secondaires et Cégeps afin d'intéresser les jeunes à la robotique. Nous sommes la première équipe à travailler en ce sens et nous avons environ huit mois pour concevoir ce prototype afin de le présenter à notre client Wael Suleiman.

Pour la conception matérielle, nous l'avons séparé en trois sections : la section tête, la section torse et la section jambes. La section tête contient deux servomoteurs contrôlant une caméra horizontalement et verticalement. La section torse est composée de l'alimentation, des différents capteurs, d'un ordinateur embarqué et d'une carte de contrôle. Enfin, la section jambes contient les quatorze servomoteurs permettant de créer une marche à six degrés de liberté.

Pour la conception logicielle, nous avons déterminé que la meilleure approche est de répartir l'application sur l'ordinateur embarqué et la carte de contrôle. La carte de contrôle implémentera l'algorithme de marche et communiquera directement avec les servomoteurs alors que l'ordinateur embarqué sera responsable du traitement d'images et du choix des comportements. Les deux cartes communiquent de façon synchronisée afin d'exécuter les actions requises par ces comportements.

Afin d'effectuer la planification et le suivi des tâches, nous utilisons principalement la méthode de gestion agile afin d'être flexibles et de pouvoir réagir aux imprévus. En complément, la méthode de gestion par cascade est utilisée pour la conception physique puisque des dépendances existent entre les différents jalons concernant les pièces électroniques et mécaniques.

2 Historique et environnement du projet

Chaque année se déroule une compétition internationale appelée RoboCup, où plusieurs équipes participent à une compétition de soccer avec des robots de forme humanoïde. Le projet RobotFoot, chapeauté par Wael Suleiman, professeur en robotique, a comme but à long terme de participer à cette compétition dans la catégorie de taille moyenne. Celui-ci a été mis en marche en réponse à l'intérêt de l'Université de Sherbrooke et du département de génie électrique et informatique à développer des compétences dans le domaine de la robotique.

Les robots participant à la compétition sont généralement le résultat de plusieurs années de travail. Ainsi, le projet est amorcé cette année et plusieurs équipes d'étudiants devront se succéder afin d'être en mesure de participer à la compétition. Cette première année de conception a pour but de fournir un prototype du robot qui pourra détecter une balle et marcher dans sa direction pour la botter sur un plancher plat, lisse et sans obstacle.

Ce prototype de robot humanoïde à prix raisonnable d'environ 1000 \$ sera une plate-forme qui pourra aussi être utilisée dans des projets du département de génie électrique et informatique de l'Université et dans les écoles secondaires et Cégeps afin d'intéresser les jeunes à la robotique.

2.1 Contraintes

RobotFoot comporte plusieurs contraintes qui sont énumérées ci-dessous :

- Se limiter à un budget de 2000 \$ provenant du département de génie électrique et informatique en plus d'une commandite d'une valeur de 150 \$ de FreeScale et d'un rabais de 10 % chez RobotShop;
- Respecter un coût final du prototype s'élevant à environ 1000 \$ pour demeurer abordable;
- Réserver un espace pour y installer des bras dans le futur;
- Être de forme humanoïde (caméra dans la tête, rotation de la tête limitée, marche avec des jambes, etc.);
- Respecter une hauteur totale de 30 à 60 cm;
- Avoir une autonomie énergétique d'environ 30 minutes;
- Livrer le prototype pour début décembre 2013.

2.2 Cadre logique

	Indicateurs	Moyens de vérification	Conditions critiques
Finalité			
Avoir une plate-forme robotique humanoïde libre de droits jouant au soccer qui servira au développement futur en vue de la compétition RoboCup	Une plate-forme robotique à forme humanoïde est prête pour du développement futur	<ul style="list-style-type: none"> Le code source et les plans du matériel sont publiquement disponibles La plate-forme comporte une tête, un torse et deux jambes 	<ul style="list-style-type: none"> Un service d'hébergement permet à long terme l'accessibilité au code et aux plans
But			
Construire un prototype de robot humanoïde pouvant se déplacer sur deux jambes afin de suivre une balle en mouvement	<ul style="list-style-type: none"> Le robot se déplace de manière stable selon plusieurs trajectoires différentes La position d'une balle en mouvement peut être détectée Le robot se dirige vers une balle en mouvement 	<ul style="list-style-type: none"> Le robot marche d'un point A à un point B, sur une surface plane et sans obstacle, sans tomber Les coordonnées de la balle en mouvement par rapport au robot sont exactes Le robot utilise les coordonnées calculées afin de déterminer sa trajectoire 	<ul style="list-style-type: none"> Une grande surface plane sans obstacle est disponible La couleur de la balle est distincte des couleurs de l'environnement de test
Extrants			
<ul style="list-style-type: none"> Conception 3D détaillée de la structure cinématique d'une jambe à 6 degrés de liberté Rapport détaillé décrivant l'algorithme du contrôle d'une marche dynamiquement stable Simulation validant la structure cinématique Schéma détaillé d'une carte contrôlant les 14 degrés de liberté (6 par jambes et 2 pour la tête) Plate-forme robotique humanoïde Code source libre de droits et documenté du logiciel embarqué Guide d'utilisation 	<ul style="list-style-type: none"> Une simulation logicielle de l'algorithme présente une marche dynamiquement stable de la structure cinématique Des schémas détaillent les composants électroniques et mécaniques du robot Le code source du logiciel est implémenté et fonctionnel La documentation créée par l'équipe présente toutes les fonctionnalités du logiciel 	<ul style="list-style-type: none"> La simulation présente toutes les caractéristiques d'une marche dynamiquement stable Tous les composants électroniques et mécaniques, ainsi que leurs branchements, sont documentés Le logiciel permet d'effectuer les différentes tâches requises 	<ul style="list-style-type: none"> Le logiciel Solidworks 2013 permet une conception 3D cinématique à 6 degrés de liberté ainsi qu'une simulation Une démarche dynamiquement stable est possible avec 6 degrés de liberté Une forme humanoïde est définie comme ayant un tronc, deux jambes, une tête et deux bras
Intrants			
<ul style="list-style-type: none"> Financement du département de génie électrique et informatique Kit robotique FSLBOT Logiciel de simulation et de modélisation (SolidWorks) Logiciel de contrôle de versions (GitHub) Logiciel de génération de documentation (Doxygen) Dropbox Application de gestion et de planification (ProjectUs) Bibliothèque de traitement d'images (Open CV) Documentation des pièces et logiciels 	<ul style="list-style-type: none"> Les achats sont effectués par l'entremise du département Le kit robotique FSLBOT et les logiciels nécessaires ou utiles à la conception sont accessibles La bibliothèque OpenCV est compilée dans un environnement de travail Des documents disponibles détaillent les spécifications des pièces et les fonctionnalités des logiciels 	<ul style="list-style-type: none"> Les factures des achats indiquent l'acheteur Le kit robotique FSLBOT est assemblé et peut être mis en marche Les logiciels et la bibliothèque OpenCV s'exécutent dans un environnement de travail 	<ul style="list-style-type: none"> Les factures des achats sont archivées et accessibles L'Université de Sherbrooke fournit certains composants matériels et logiciels Certains logiciels utiles, ainsi que certaines bibliothèques, sont disponibles gratuitement

Tableau 1 : Cadre logique

2.3 Description technique

2.3.1 Conception physique

Les pièces du squelette mécanique seront modélisées sur Solidworks avant leur fabrication. Quant aux composantes électroniques, elles sont divisées en trois sections : la tête, le torse et les jambes. Les connexions entre celles-ci sont représentées dans le schéma ci-dessous :

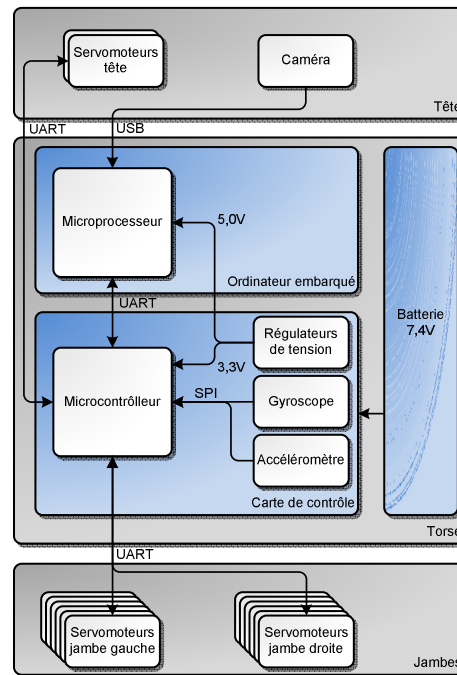


Figure 1 : Schéma haut niveau des composantes électroniques

2.3.1.1 Section tête

La section tête est composée de deux servomoteurs qui contrôlent une caméra verticalement et horizontalement afin de détecter une balle dans l'espace.

2.3.1.1.1 Caméra

Afin d'envoyer les informations nécessaires au traitement d'images, une seule caméra placée dans la tête du robot est nécessaire. L'utilisation de deux caméras en stéréoscopie a été envisagée afin de faciliter le calcul de distances, mais cela a été jugé superflu et plus dispendieux. Ainsi, pour déterminer la distance entre le robot et la balle, le théorème de Pythagore sera utilisé en tenant compte de la hauteur du robot et de l'angle de sa tête.

Une grande résolution d'image facilite une distinction d'objets à une distance plus élevée. Par contre, le traitement d'images demande un temps de calcul plus important. Afin que la qualité de l'image soit acceptable pour détecter la balle à au moins dix mètres, et que le temps de calcul soit raisonnable, une résolution de 640 x 480 pixels a été jugée appropriée après quelques tests effectués sur un ordinateur.

En plus de la résolution, le débit de transmission est aussi un facteur important pour la détection d'un objet qui est en mouvement. Lorsque le champ de vision est à son niveau le plus faible, il faut que la détection d'une balle qui passe ait le temps d'être effectuée. Ainsi, un minimum de quinze images par seconde (fps) est requis, mais un débit de 30 fps a tout de même été choisi étant donné que la plupart des caméras sont configurées à ce débit.

La popularité élevée des caméras Web USB a fixé le choix d'interface sur le standard USB. La flexibilité et la nature *Plug and Play* de l'USB est appropriée pour la pérennité du projet. Effectivement, cela permettra de remplacer facilement la caméra, voire d'ajouter une caméra afin de bénéficier d'une vision stéréo, sans intervention importante de la part des concepteurs. D'ailleurs, l'utilisation de ce type d'interface présente aussi l'avantage qu'une source d'alimentation externe propre à la caméra n'est pas nécessaire.

2.3.1.2 Section torse

La section torse contient l'ordinateur embarqué, la carte de contrôle et la batterie. L'ordinateur embarqué est lui-même constitué d'un microprocesseur qui a pour but de recueillir et analyser les données afin d'envoyer des commandes de haut niveau (avance, tourne, etc.) aux différents modules de contrôle. La carte de contrôle, quant à elle, comporte un régulateur de tension et un microcontrôleur auquel sont branchés un gyroscope et un accéléromètre.

2.3.1.2.1 Batterie

Le tableau suivant illustre les technologies de piles et de batteries les plus courantes et leurs caractéristiques :

Caractéristique	Li-Po	Ni-Mh	LiFePo
Énergie / Poids	100-250 Wh/kg	30-80 Wh/kg	90-110 Wh/kg
Énergie / Volume	200-620 Wh/L, excellent rapport de grosseur	140 à 300 Wh/L, boîtier rigide et plus imposant	220 Wh/L, très similaire à Li-Po
Prix (7-10V, 2000mah)	< 12 \$	< 6 \$	< 15 \$
Tension nominale par cellule	3,7 V	1,2 V	3.3 V
Stabilité	Peut gonfler et brûler lors de mauvais usage	Très stable	Beaucoup plus stable que la Li-Po

Tableau 2 : Comparaison des batteries

Bien qu'il existe une multitude de types de piles et de batteries, notre choix s'est rapidement porté sur les batteries lithium-ion polymères (Li-Po) qui ont l'avantage d'offrir la meilleure capacité énergétique en fonction de leur poids, comme indiqué ci-haut. De plus, comme ce type de batterie est très utilisé pour les modèles réduits radiocommandés, une panoplie de formes et de dimensions de batteries sont disponibles. Enfin, les coûts pour ce type de batterie sont abordables.

Comme les servomoteurs choisis opèrent à une tension nominale de 7,4 V, nous optons pour les batteries à deux cellules, de tension nominale de 7,4 V. Il sera alors possible d'alimenter directement les moteurs avec la batterie en incluant une protection contre les surcharges de courant. De plus, un régulateur à découpage abaisseur pour l'alimentation 5 V à haut rendement sera employé pour alimenter l'ordinateur embarqué. Quant à la capacité de la batterie, nous estimons qu'une capacité d'environ 2000 mAh permettra de rencontrer la contrainte d'autonomie de 30 minutes. En effet, la batterie peut théoriquement fournir 2 A en continu au système complet, incluant les moteurs, pendant une heure. Cette métrique semble réaliste compte tenu des résultats des tests préliminaires sur les servomoteurs. Le choix final de la batterie sera effectué lorsque les plans du torse seront finalisés afin de choisir la plus grosse batterie pouvant y être placée.

2.3.1.2.2 Carte de contrôle

Le tableau suivant compare les caractéristiques de plates-formes qui sont déjà en notre possession pour la carte de contrôle :

Caractéristique	TWR-MECH	STM32F4Discovery	Netduino 2
Prix	100 \$	16 \$	35 \$
Contrôleur/Clock	ColdFire v2/66MHz	ARM Cortex-M4/168MHz	ARM Cortex-M3/120MHz
Mémoire	512Kb Flash 64Kb RAM	1 Mb Flash 192Kb RAM	120Kb Flash 60Kb RAM
Programmation	C/C++ (CodeWarrior)	C/C++	Cadre d'applications .NET (C#)
Unité de calcul à virgule flottante (FPU)	Oui	Oui	Non

Tableau 3 : Comparaison des cartes de contrôle

Pour agir comme une interface efficace entre les moteurs et l'ordinateur embarqué, il est essentiel que la plate-forme puisse effectuer rapidement des calculs de cinématique complexes utilisés pour les mouvements du robot. De plus, il est important que le système ait un très court temps de réponse aux perturbations extérieures, soit que les accès aux entrées/sorties soient très rapides. Pour se conformer aux deux exigences à la fois, il faut de préférence un microcontrôleur performant, idéalement dépourvu de système d'exploitation ou muni d'un système d'exploitation temps réel, afin de garder les latences au minimum.

La plate-forme STM32F4Discovery se démarque par son prix et ses caractéristiques. En effet, le microcontrôleur a la plus grande cadence d'horloge et la plus grande mémoire, en plus de posséder une unité de calcul à virgule flottante (FPU). Cependant, comparé aux autres plates-formes, il n'offre pas un environnement de développement (IDE) adapté avec des bibliothèques de haut niveau prêtes à l'emploi. Par comparaison, le Netduino 2 permet de développer avec le cadre d'applications .NET, contenant une bibliothèque relativement large spécifiquement pour le contrôleur, alors que le TWR-MECH de Freescale est bien intégré avec leur IDE CodeWarrior.

Après des essais sur les trois plates-formes, nous avons constaté qu'une composante essentielle pour développer et déboguer le TWR-MECH avec CodeWarrior ne peut pas fonctionner sur un système d'exploitation 64 bits, ce qui limite fortement son utilité. Le Netduino 2, au contraire, est très facile à prendre en main et ne nécessite pas de configurations préalables. Enfin, le STM32F4Discovery est compatible avec un bon nombre d'environnements de développement (IAR, Keil) dont des versions d'évaluation sont disponibles. Mais, ce qui a retenu notre attention est l'usage d'outils à code ouvert qui permettent de développer le logiciel sur les microcontrôleurs basés sur les ARM. Par contre, cela demande une configuration préalable d'un IDE tel qu'Eclipse.

Le Netduino 2 était aussi une bonne option, mais le cadre d'applications .NET ajoute une latence de plusieurs millisecondes sur les entrées/sorties, ce qui aurait été un problème majeur pour la communication avec les servomoteurs.

À la lumière des essais, notre choix s'est finalement arrêté sur la plate-forme avec le meilleur prix et la plus grande capacité de calcul, c'est-à-dire le STM32F4Discovery.

Cependant, nous envisageons de concevoir et fabriquer notre propre circuit imprimé inspiré de la plate-forme de développement STM32F4Discovery, c'est-à-dire qu'il sera doté du même microcontrôleur. Le circuit imprimé aura les dimensions appropriées pour pouvoir s'attacher mécaniquement sur l'ordinateur embarqué et aura directement les composantes nécessaires telles que l'alimentation 5V, le gyroscope et l'accéléromètre. De plus, il offrira des connecteurs adéquats pour les moteurs et la communication avec l'ordinateur. La plate-forme STM32F4Discovery sera tout de même commode pour la conception du circuit imprimé, car il permet de programmer et déboguer d'autres microcontrôleurs.

2.3.1.2.3 Ordinateur embarqué

Étant donné que le traitement d'images demande beaucoup de mémoire vive, la carte de contrôle ne peut acquiescer cette tâche seule au risque de ralentir considérablement l'exécution des algorithmes. Il est donc nécessaire d'inclure un ordinateur embarqué. Ce dernier doit contenir un microprocesseur qui sera assez rapide pour effectuer des algorithmes complexes, contenir suffisamment de mémoire vive et être conçu pour un système embarqué.

Le microprocesseur choisi nécessite une grande capacité de calcul afin d'analyser la masse de données de l'environnement du robot provenant des capteurs pour ensuite effectuer des actions appropriées aux diverses situations.

De plus, la carte de développement de celui-ci nécessite un minimum de mémoire vive (RAM) nécessaire à l'algorithme de traitement d'images. Après différentes recherches et tests sur un ordinateur, une mémoire vive 1 Go a été jugée amplement suffisante pour emmagasiner les images à traiter sans perturber le déroulement des autres tâches.

Il est aussi nécessaire que l'ordinateur embarqué puisse être positionné sur le robot, ce qui implique qu'il ne doit pas dépasser un volume de 4" x 4" x 2". Aussi, afin de respecter la contrainte d'autonomie,

l'alimentation nécessaire au fonctionnement de celui-ci est tout aussi cruciale. La consommation en courant de tous ses périphériques ne doit pas dépasser 3 A afin d'assurer une autonomie convenable selon la batterie choisie.

Profitant d'une commande de la part de Freescale Semiconductor, la carte i.MX53 Quick Start Board, d'une valeur de 150 \$, nous est offerte gratuitement. Cette carte de développement est une carte de développement à code ouvert, intégrée avec un processeur ARM Cortex-A8 cadencé à 1 GHz. Cette carte a donc été choisie puisque ses caractéristiques conviennent à nos besoins.

2.3.1.3 Section jambes

La section jambes est constituée des quatorze servomoteurs qui organisent la marche. Les jambes auront six degrés de liberté, dont trois situés au niveau de la hanche (*roulis/tangage/lacet*), un au genou (*tangage*) et deux à la cheville (*roulis/tangage*). Pour obtenir une marche de qualité satisfaisante, l'utilisation de six degrés de liberté est un standard en robotique permettant un déplacement et une rotation du pied dans l'espace, dans n'importe quelle direction.

2.3.1.3.1 Servomoteurs

Le tableau suivant compare les différents types de servomoteurs :

Caractéristique	Analogique	Numérique	Intelligent
Prix	< 13 \$	< 24 \$	< 40 \$
Interface	Analogique (PWM) 50 Hz nominaux	Analogique (PWM) 50 Hz nominaux	Numérique : UART TTL full/half duplex, RS482, CAN
Couple	Sous-optimal	Très bon	Très bon
Contrôle du couple	Fixe, diminue lorsque l'erreur est faible	Fixe, reste constant selon l'erreur	Programmable
Rétroaction sur la position	Inexistante	Inexistante	Disponible
Rétroaction sur la vitesse	Inexistante	Inexistante	Disponible
Contrôle de vitesse	Fixe	Fixe	Programmable
Vitesse de réaction aux commandes	Lent (pénalisé par la nature du signal de commande)	Lent (pénalisé par la nature du signal de commande)	Rapide (dépendant de la vitesse de l'interface, mais peut aller à 1 Mb/s)
Vitesse de réaction aux perturbations	Lent	Rapide	Rapide
Type de connexion	Signaux dédiés	Signaux dédiés	Bus

Tableau 4 : Comparaison des types de servomoteurs

L'utilisation de servomoteurs numériques intelligents a été préférée puisqu'il est possible d'obtenir de l'information sur leurs opérations, telles que l'angle actuel, la force exercée et leur température. De

plus, ils sont plus performants que les moteurs analogiques standards et de plus petite taille, ce qui est d'autant plus avantageux afin que le robot soit compact et léger. Le tableau suivant compare donc deux servomoteurs numériques intelligents :

Caractéristiques	AX-12W	DRS-0101
Prix	45 \$	40 \$
Couple	2 kgf × cm	12 kgf × cm
Vitesse	0.023 s/60°	0.166 s/60°
Poids	52.9 g	45 g
Dimensions	50 × 32 × 40 mm HWD	45 × 24 × 31 mm HWD
Interface	UART half-duplex (RX/TX)	UART full duplex (RX & TX)
Vitesse maximale de transmission	1 Mb/s	0,666 Mb/s

Tableau 5 : Comparaison des servomoteurs intelligents

Les fonctionnalités offertes par les deux moteurs sont très similaires : contrôle de couple maximal/minimal, contrôle de vitesse, acquisition de la température/position/vitesse/charge appliquée. Cependant, le DRS-0101 permet en plus d'ajuster les paramètres PID du contrôleur. Il est également beaucoup plus puissant que l'AX-12W, mais beaucoup plus lent. Étant donné que les servomoteurs doivent supporter et déplacer une grande charge pour le couple de l'AX-12W et que la vitesse maximale du DRS-0101 n'est pas exceptionnelle, mais suffisamment rapide pour nos besoins, le DRS-0101 a été favorisé par rapport à l'AX-12W.

2.3.1.4 Résumé des composants électroniques

Section	Composante	Choix
Tête	Caméra	Caméra Web USB 640 x 480 pixels, 30 fps
Torse	Batterie	Lithium Polymère Li-Po 7,4 V
	Carte de contrôle	PCB fait sur mesure à partir du microcontrôleur STM32F4
	Ordinateur embarqué	i.MX Quick Start Board, ARM Cortex-A8 1 Ghz
Jambes	Servomoteurs	Servomoteurs numériques intelligents DRS-0101

Tableau 6 : Résumé des composants électroniques

2.3.2 Conception logicielle

Différents modules seront implémentés dans l'ordinateur embarqué et la carte de contrôle. Les interactions entre ces modules sont représentées dans ce schéma haut niveau :

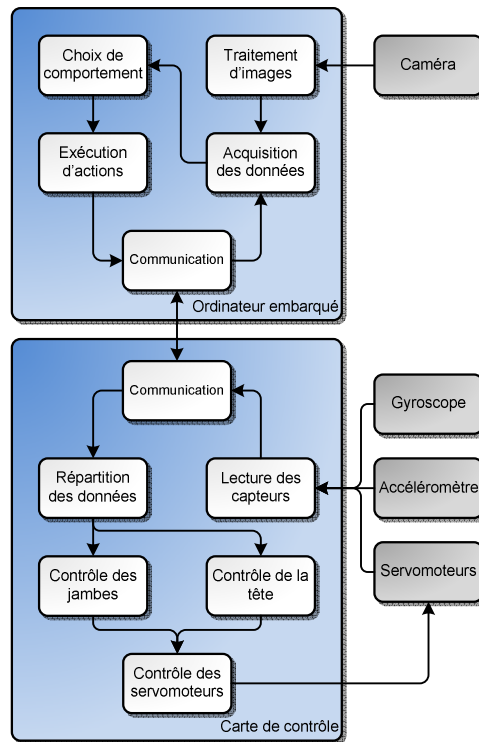


Figure 2: Schéma haut niveau des modules logiciels

La carte de contrôle effectue le contrôle des servomoteurs de la tête et des jambes, ainsi que la lecture des données fournies par le gyroscope, l'accéléromètre et les servomoteurs. Ces données sont communiquées à l'ordinateur embarqué afin qu'elles soient traitées.

L'ordinateur embarqué est responsable du traitement d'images et de la prise de décisions concernant le choix du comportement. Ce choix est déterminé selon les informations acquises des différents capteurs physiques et des résultats d'actions précédemment effectuées.

2.3.2.1 Choix du comportement

Les données acquises des différents capteurs et des actions effectuées par le robot lui permettront d'effectuer un choix par rapport au comportement qu'il doit adopter. Selon le comportement courant, différentes actions seront effectuées et les réactions aux données seront différentes. Le schéma suivant montre le processus menant au choix du comportement :

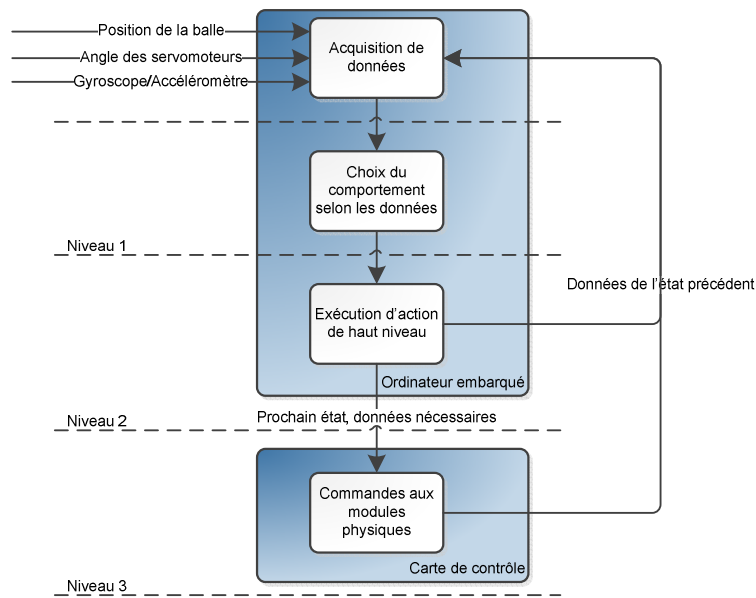


Figure 3 : Schéma du processus menant au choix du comportement

Les comportements et les conditions qui activent une transition vers un comportement différent sont représentés dans cette machine à états finis :

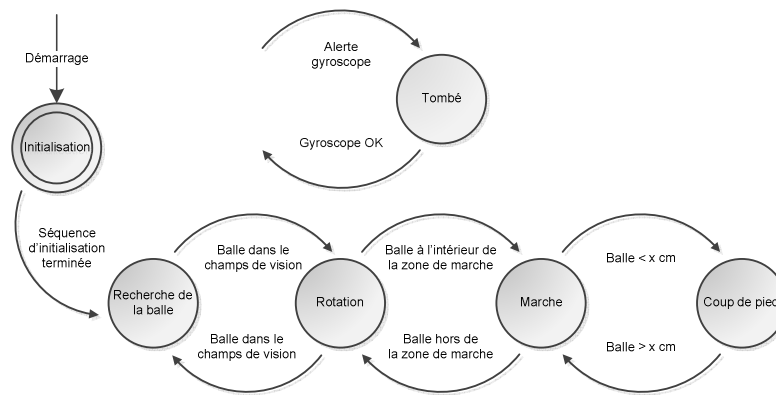


Figure 4 : Machine à états finis des comportements

2.3.2.2 Exécution d'actions

Les actions effectuées par le robot sont définies par le comportement en cours :

- **Initialisation** : Configuration des paramètres et démarrage des tâches;
- **Recherche de la balle** : Le robot tourne sa tête et son corps jusqu'à ce que la balle soit dans son champ de vision;
- **Rotation** : Le robot tourne sur lui-même afin que l'emplacement de la balle soit à l'intérieur de la zone de marche;

- **Marche** : Le robot se déplace vers la balle jusqu'à ce qu'elle soit inférieure à une distance prédéterminée;
- **Coup de pied** : Le robot donne des coups de pied sur la balle jusqu'à ce qu'elle soit supérieure à une distance prédéterminée;
- **Tombé** : Le gyroscope indique que le robot est tombé, il ne doit effectuer aucune action jusqu'à ce que le gyroscope indique qu'il soit debout.

2.3.2.3 Traitement d'images

Étant donné la quantité élevée de pixels à analyser pour chaque image, le traitement d'images en temps réel demande une grande utilisation de mémoire vive, une bibliothèque logicielle graphique dont l'utilisation de ressources est limitée doit donc être utilisée. OpenCV, qui est une bibliothèque graphique spécialisée dans le traitement d'images en temps réel, libre de droits et dont le code est ouvert, a été choisie après une comparaison avec d'autres bibliothèques semblables telle que SimpleCV, CImg et libCVD. Grâce à la nature bas niveau de cette bibliothèque, constituée de fonctions écrites en C, son utilisation de ressources est faible, avoisinant les 25 Mo lors d'un test sur un ordinateur personnel. Toutefois, l'allocation et la libération de la mémoire ne sont pas effectuées automatiquement et doivent être gérées par les développeurs afin d'éviter les fuites de mémoires qui pourraient s'accumuler et causer des ralentissements et des plantages.

Un autre aspect qui a incité à choisir OpenCV est le fait qu'elle n'est pas limitée à un environnement de développement particulier et peut être utilisé avec n'importe quel environnement compatible avec le langage C. D'ailleurs, cette bibliothèque peut être utilisée sur une variété de plates-formes différentes.

Cependant, la documentation officielle d'OpenCV est incomplète et peu organisée. Malgré tout, grâce à sa forte popularité, une très grande quantité de guides et d'exemples sont disponibles en ligne afin de faciliter le développement.

2.3.2.4 Contrôle de la tête

Des ajustements constants seront effectués sur les servomoteurs qui déplacent la caméra de façon horizontale et verticale afin que la balle en mouvement soit toujours au centre du champ de vision. De plus, la distance de la balle par rapport au robot sera calculée selon l'équation suivante, qui est basée sur le théorème de Pythagore et utilise des données connues :

$$\text{Distance de la balle} = (\text{Hauteur de la caméra}) \times \tan(\text{Angle de la caméra})$$

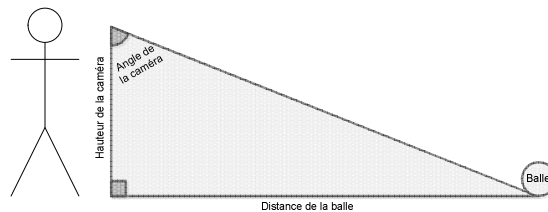


Figure 5 : Schéma du calcul de la distance

2.3.2.5 Contrôle des jambes

Nous avons plusieurs idées afin de réaliser un algorithme de marche stable. Tout d'abord, nous ferons des recherches en biomécanique afin de trouver la façon la plus humaine et facile de faire marcher un robot. Ceci peut être fait par exemple en analysant les huit phases de la marche humaine. Le tout devra être exprimé sous forme d'équations afin de représenter les angles de chacun des servomoteurs à chaque étape d'un pas. Enfin, l'algorithme sera conçu afin de représenter le positionnement et le botté de la balle.

Nous allons tester deux approches pour la marche afin de choisir celle qui convient le mieux. L'une d'elles se base sur des mouvements prédéfinis alors que l'autre sur une approche d'asservissement de la stabilité avec un gyroscope. D'ailleurs, il est possible que la combinaison des deux offre de meilleurs résultats. Afin de qualifier ces approches, il sera intéressant de simuler le mouvement des jambes. De plus, nous allons concevoir deux exemplaires du prototype afin de développer plusieurs approches en parallèle, ce qui maximisera le rendement de l'équipe.

2.4 Organisation du projet

Pour avoir une grande flexibilité dans le développement du projet, nous avons opté pour une méthodologie de travail agile, en utilisant l'application de gestion ProjectUS. Nous travaillerons donc de façon itérative afin de pouvoir effectuer des rectifications rapides lorsque cela sera nécessaire tout au long de l'évolution du projet. De courtes réunions d'avancement auront fréquemment lieu afin de présenter une rétrospective des tâches accomplies et de prévoir ce qui sera développé prochainement. Cette méthodologie définit trois rôles :

- **Propriétaire du produit** (Mathieu Drapeau) : Il représente le client et a comme objectif de maximiser la valeur du produit développé;
- **ScrumMaster** (Mickaël Paradis) : Il doit s'assurer que la méthode est comprise et bien mise en application;
- **Développeurs** (tous) : Ils doivent effectuer le développement de façon autoorganisée, sans notion de hiérarchie interne.

Toutefois, la méthode de gestion par cascade est aussi utilisée pour la conception physique. En effet, certaines tâches ont des dépendances entre elles, comme on peut le constater sur la charte de Gantt disponible sur le SVN de RobotFoot, et un chemin critique sera établi et suivi. Pour organiser la séparation des tâches et des responsabilités liées à la conception physique, nous avons séparé l'équipe en trois sous-équipes : l'équipe tête, l'équipe torse et l'équipe jambes. Chacune de ces équipes est responsable de l'intégration des composants électroniques et mécaniques de leur section respective.

2.5 WBS

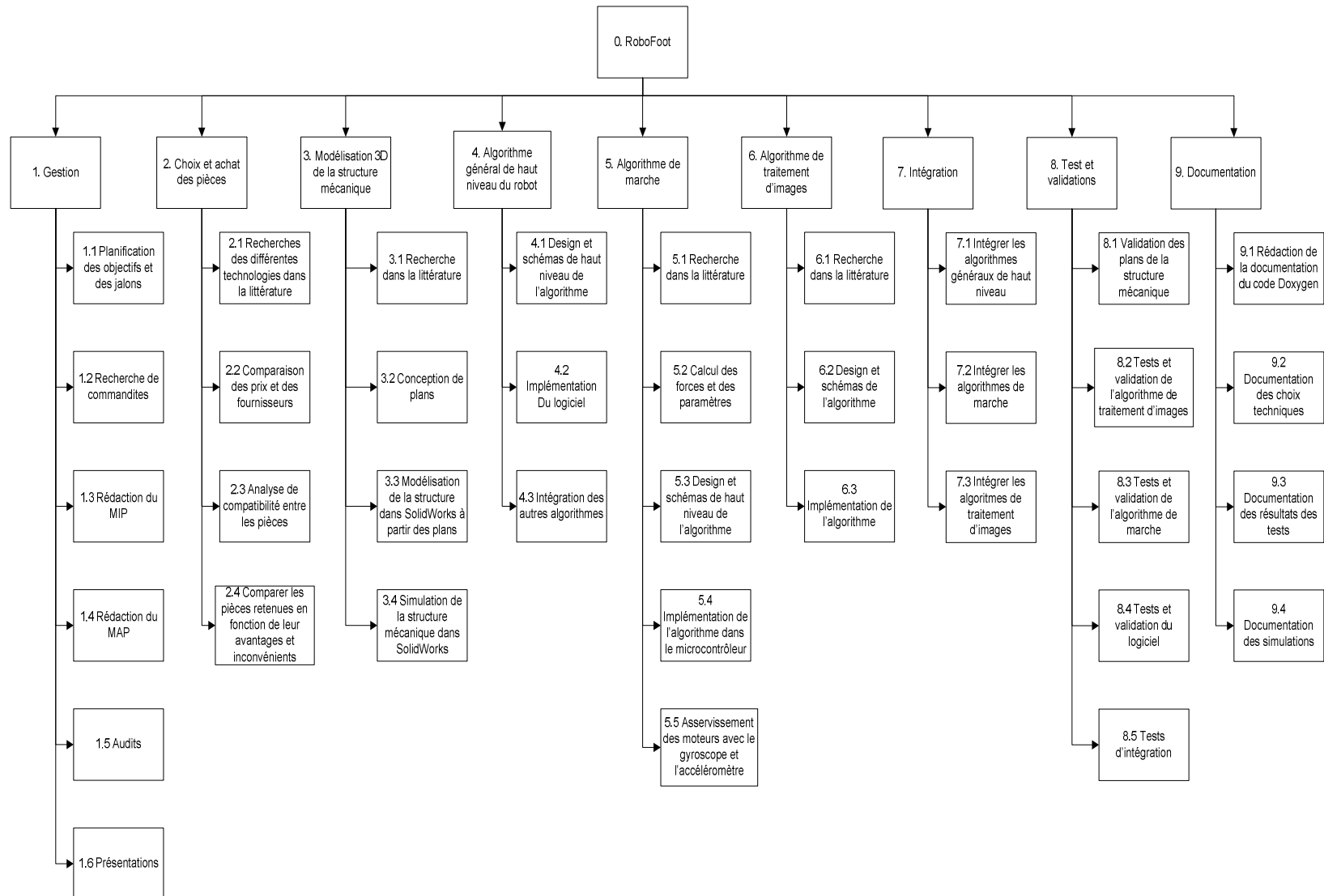


Figure 6 : WBS

2.6 Structure organisationnelle de la conception physique : OBS

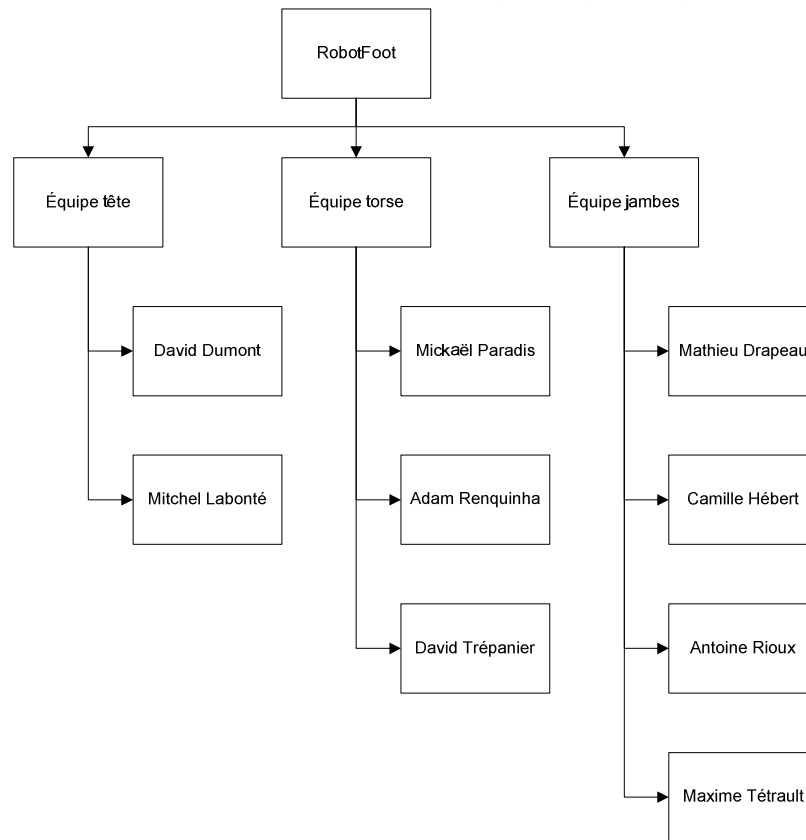


Figure 7 : OBS

2.7 Charte des responsabilités

Les tâches de programmation étant exécutées avec une méthodologie agile, le propriétaire du produit est le responsable et tous les membres de l'équipe seront des développeurs participant activement à leur avancement. Notre charte des responsabilités reflète donc cette méthodologie et ne considère que les tâches du WBS concernant la gestion et les parties physiques, étant suivi en cascade.

WBS	Tâches	Mathieu Drapeau	David Dumont	Camille Hébert	Mitchel Labonté	Mickaël Paradis	Adam Renquinha	Antoine Rioux	Maxime Tétrault	David Trépanier
1	Gestion	R	P	P	P	P	P	P	P	P
1.1	Planification des objectifs et des jalons	R	P	P	P	P	P	P	P	P
1.2	Recherche de commandites	P							R	
1.3	Rédaction du MIP	P	P	P	P	P	P	P	P	R
1.4	Rédaction du MAP	P	P	P	P	P	P	P	P	R
1.5	Préparation des audits	P							R	
1.6	Présentations	P							R	
2	Choix et achat des pièces	I	A		P	R				
2.1	Recherche de différentes technologies dans la littérature	I			P	P				P

2.2	Comparaison des prix et des fournisseurs	I			P	A		R		P
2.3	Analyse de compatibilité entre les pièces	I					A	R		P
2.4	Comparer les pièces retenues en fonction de leurs avantages et inconvénients	I	P	A				R		
3	Modélisation 3D de la structure mécanique	I		O		R		A	P	
3.1	Recherche dans la littérature	I	P	P	P	P	P	P	A	P
3.2	Conception de plans	I		A		R		A		
3.3	Modélisation de la structure dans SolidWorks à partir des plans	I		A		O		R		
3.4	Simulation de la structure mécanique dans Solidworks	I		R				A		
7	Tests et validation	P	P	P	P	P	R	P	P	P
7.1	Validation des plans de la structure mécanique	I		R		A		A		
7.2	Tests et validation de l'algorithme de reconnaissance d'image	I	A		R					
7.3	Tests et validation de l'algorithme de marche	A		R		A		A	A	
7.4	Tests et validation du logiciel	I				R	A			A
7.5	Tests d'intégration	P	P	P	P	P	P	P	P	R
8	Documentation	R	P	P	P	P	P	P	P	P
8.1	Rédaction de la documentation du code avec Doxygen	R	P	P	P	P	P	P	P	P
8.2	Documentation des choix techniques	P	R	P	P	P	P	P	P	P
8.3	Documentation des résultats des tests	P	P	P	P	P	R	P	P	P
8.4	Documentation des simulations	P	P	R	P	P	P	P	P	P
Légende										
<i>N</i> : Autorise/approuve					<i>A</i> : Exécute la tâche					
<i>O</i> : Aide à l'exécution					<i>P</i> : Participe					
<i>R</i> : Responsable de la tâche					<i>I</i> : Est informé					

Tableau 7: Charte des responsabilités

3 Échéancier du projet et gestion du temps

3.1 Charte de Gantt

La charte de Gantt est présente à la racine du SVN de RobotFoot. L'objectif principal de cette charte est d'établir le chemin critique de notre projet pour assurer sa réussite et pour cibler les tâches risquant de causer des retards. Comme la méthode de gestion par cascade est seulement utilisée pour la conception physique, seules les tâches reliées à cette dernière sont représentées dans la charte. Ainsi, nous envisageons de terminer ces tâches en avril, pour ensuite nous consacrer au développement logiciel.

Le chemin critique concerne principalement la marche du robot. Il commence par des recherches et des choix sur la configuration des jambes du robot. Ensuite, les pièces mécaniques seront modélisées à l'aide d'un logiciel. Puis, des soumissions seront envoyées à différentes firmes afin d'obtenir le meilleur prix sur la fabrication. Enfin, les jambes seront assemblées et l'implémentation de l'algorithme de marche pourra commencer.

4 Coût du projet et gestion des coûts

4.1 Budget global

Le département de génie électrique et informatique constitue notre principale source de revenus, en plus de fournir certains composants. De plus, nous avons obtenu des commandites de la part de deux fournisseurs : Freescale fournit gratuitement l'ordinateur embarqué et RobotShop offre un rabais de 10 % sur l'achat des servomoteurs. Ces tableaux présentent nos sources de revenus actuelles et les dépenses planifiées :

Source de revenus	Montant	Quantité	Total
Financement			
Département de génie électrique et informatique	2 000,00 \$	1	2 000,00 \$
Commandites			
Département de génie électrique et informatique	31,05 \$	2	62,10 \$
Freescale	172,50 \$	2	345,00 \$
RobotShop	4,31 \$	26	112,13 \$
Total			2 519,23 \$

Tableau 8 : Sources de revenus

Composant	Coût	Quantité	Total
Tête			
Caméra	25,00 \$	2	50,00 \$
Servomoteur	37,50 \$	4	150,00 \$
Torse			
Batterie	15,00 \$	2	30,00 \$
Ordinateur embarqué i.MX53 Quick Start Board	150,00 \$	2	300,00 \$
Carte de contrôle			
Microcontrôleur STM32F4	13,00 \$	2	26,00 \$
PCB	40,00 \$	2	80,00 \$
Composants (résistances, condensateurs, etc.)	5,00 \$	2	10,00 \$
Régulateur 5V	20,00 \$	2	40,00 \$
Régulateur 3V	2,00 \$	2	4,00 \$
Gyroscope/Accéléromètre	20,00 \$	2	20,00 \$
Jambes			
Servomoteur	37,50 \$	24	900,00 \$
Fabrication des pièces	275,00 \$	2	550,00 \$
Sous-total			2 180,00 \$
Total après taxes			2 507,00 \$

Tableau 9 : Dépenses planifiées

4.2 Suivi budgétaire pro forma

Le suivi du budget est mis à jour chaque semaine avec le tableur Excel. Le suivi budgétaire pro forma est disponible sur le SVN de RobotFoot.

5 Gestion des communications

5.1 Coordonnées des principaux acteurs

Prénom	Nom	Courriel	Numéro de téléphone
Équipe RobotFoot			
Mathieu	Drapeau	m.drapeau9@gmail.com	514-883-7997
David	Dumont	david9dumont@gmail.com	819-239-9498
Camille	Hébert	camille.hebert86@gmail.com	450-776-8500
Mitchel	Labonté	mittchellabonte@hotmail.com	819-674-1383
Mickaël	Paradis	mick.paradis@gmail.com	450-775-8095
James-Adam	Renquinha Henri	jarhmander@gmail.com	819-239-8169
Antoine	Rioux	antoine.rioux@outlook.com	819-674-8975
Maxime	Tétrault	maxime.tetrault@usherbrooke.com	450-501-2653
David	Trépanier	david191999@hotmail.com	819-349-4038
Client			
Wael	Suleiman	Wael.Suleiman@USherbrooke.ca	819-821-8000 poste 62131
Superviseurs			
Ruben	Gonzalez-Rubio	Ruben.Gonzalez-Rubio@USherbrooke.ca	819-821-8000 poste 62931
Ahmed	Khoumsi	Ahmed.Khoumsi@UShebrooke.ca	819-821-8000 poste 63796

Tableau 10 : Coordonnées des principaux acteurs

5.2 Rapports — pro forma

La page titre pro forma, la table des matières pro forma et le compte-rendu des réunions pro forma sont disponibles sur le SVN de RobotFoot.

6 Gestion des risques et suivi

Le tableau de gestion des risques présente la probabilité des risques, leur gravité et leur priorité :

Risque	Probabilité (1-10)	Gravité (1-10)	Priorité (1-10)
Dépassement du budget	8	6	9
Mauvaise conception des pièces mécaniques	4	9	8
Bris de certaines pièces	4	8	6
Inaccessibilité de la plate-forme de développement	3	7	6
Retard dans la réception des commandes	7	4	4
Mauvaise planification des tâches	9	2	3

Tableau 11 : Gestion des risques

Le risque avec la priorité la plus élevée est le dépassement du budget. En effet, étant donné que le budget est restreint et que la marge de manœuvre actuelle est faible, un écart aux prévisions budgétaires nécessitera l'obtention d'autres sources de revenu, ce qui causerait des retards. Ainsi, il faut effectuer nos choix d'achats en considérant les prix de façon prioritaire et saisir toutes les opportunités de commandes. Toutefois, si un dépassement survient, nous pourrions y remédier en ayant recours à des activités de financement ou aux fonds de recherche de notre client Wael Suleiman.

Un autre risque prioritaire est la mauvaise conception des pièces mécaniques. En effet, le coût de fabrication de celles-ci est élevé et notre budget est restreint, une erreur serait donc difficile à absorber. Ainsi, une validation rigoureuse de notre modélisation par des gens ayant une plus grande expertise dans le domaine et des simulations sont donc nécessaires afin de mitiger le risque d'erreurs.

Dans le même ordre d'idées, il est important de choisir judicieusement les pièces avant leur achat et de manipuler celles-ci soigneusement afin d'éviter les bris. Afin de parer à toute éventualité, des pièces de rechange seront prévues dans le budget.

Étant donné que le nombre de plates-formes de développement est limité, l'accès à celles-ci doit être partagé entre les membres de l'équipe. Pour un plus grand accès, deux plates-formes seront assemblées, permettant ainsi d'effectuer des tâches en parallèle.

Des risques de retard pour la réception des pièces sont possibles. Même si les commandes sont passées à l'avance afin de prévenir ce risque, un retard hors de notre contrôle peut survenir malgré tout. Cependant, il est toutefois possible de progresser en accomplissant des tâches qui ne nécessitent pas ces pièces.

Il est très probable que notre planification des tâches soit inadéquate. Par exemple, des tâches peuvent être plus longues que ce qui a été prévu, non planifiées ou comporter des dépendances non anticipées. Le fait d'utiliser la méthode de gestion agile nous permet cependant de nous ajuster rapidement aux imprévus. Si nécessaire, nous compenserons nos erreurs de planification en augmentant nos heures de travail.

7 Santé et sécurité

7.1 Procédures de tests à risques

Comme il est possible de le voir sur le prochain tableau, peu de manipulations requises dans ce projet comportent des risques importants. Presque tous les risques analysés sont de gravité G1 et sont évitables, à l'exception d'un risque de gravité G2, relatif à une mauvaise utilisation de la batterie. Il faudra toutefois s'assurer de respecter certaines normes de sécurité pour réduire les risques.

Tout d'abord, il sera important que chaque membre de l'équipe prenne des pauses régulières lors des séances de programmation. Ceci évitera une fatigue des yeux ainsi que des douleurs au dos causées par des postures potentiellement inadéquates.

Ensuite, lors de travaux sur la structure physique du robot, les membres de l'équipe devront manipuler certains outils de façon sécuritaire. En effet, le port des lunettes de sécurité sera primordial lors des manipulations du fer à souder. De plus, lorsque nous testerons les mouvements du robot, il faudra s'assurer de prendre garde aux pièces métalliques en mouvements qui pourraient se refermer sur les doigts.

Enfin, il faut absolument prévoir diverses protections de nature électrique et mécanique sur le circuit de contrôle. Par exemple, l'usage de fusibles empêchera qu'un court-circuit provoque la rupture, voire l'explosion de la batterie.

7.2 Analyse d'impacts sur la santé, l'environnement et le public

RobotFoot comporte peu d'impacts sur la santé ou l'environnement, mais il comporte un aspect social important. En effet, d'ici quelques années, le robot pourra être utilisé à des fins éducatives dans des écoles secondaires et des Cégeps. Ce prototype servirait ainsi à promouvoir l'apprentissage et la relève en robotique. C'est pourquoi notre équipe a décidé d'opter pour une philosophie libre de droits pour les algorithmes qui seront créés ainsi que pour la modélisation des pièces mécaniques (matériel ouvert, code ouvert). Les choix de conception et les résultats de nos recherches pourront donc être réutilisés par des équipes subséquentes pour assurer la pérennité du projet.

Tâches ou manipulations	Composantes du risque				Estimation du risque (sans protection)					Évaluation du risque (1-3)	Mesures de réduction du risque	Remarque
	Phénomène dangereux	Situation dangereuse	Événement dangereux	Dommages possibles	Gravité (G1/G2)	Fréquence d'exposition (F1/F2)	Probabilité d'occurrence (O1/O2/O3)	Possibilité d'évitement (P1/P2)	Indice de risque (1-6)			
Programmer	Mauvaise ergonomie du poste de travail, mauvais éclairage	Corps exposé à de mauvaises postures, fatigue des yeux	Travailler avec une posture inappropriée ou avec un éclairage inapproprié pour une durée prolongée	Douleurs au dos, fatigue des yeux, tunnel carpien	G1	F2	O3	P1	3	1	Utiliser un environnement ergonomique adapté	
Monter la structure physique du robot	Utilisation d'outils, travailler avec des petites pièces comme des vis	Blessures physiques, blessures aux yeux	Se blesser avec des outils, pièces métalliques projetées dans les yeux	Écorchures, bleu, blessures aux yeux	G1	F1	O2	P1	2	2	Utiliser des lunettes de sécurité	
Travailler avec des servomoteurs	Pièces en mouvements	Deux membres du robot qui se referment l'un sur l'autre	Se coincer les doigts entre deux pièces en mouvements	Bleu, peau écorchée	G1	F1	O1	P1	2	3	Déterminer l'axe et la direction de mouvement des membres du robot avant de les mettre en mouvement	
Soudure	Exposition à un fer chaud	Utilisation d'un fer à souder chaud	Se brûler avec le fer à souder, recevoir de l'étain dans les yeux	Brûlures, blessures aux yeux	G1	F1	O2	P1	3	2	Utiliser des lunettes de sécurité	
Utilisation de la batterie	Exposition à une batterie	Travail à proximité de la batterie	Subir l'explosion de la batterie	Brûlures chimiques, blessures aux yeux, dommages auditifs	G2	F1	O1	P2	2	1	Implémenter un circuit de protection contre les courts-circuits Utiliser un chargeur adéquat	
Légende												
Gravité	G1 : lésion légère, normalement réversible (écorchure, bleu, etc.) G2 : lésion grave, normalement irréversible, y compris le décès (membre brisé, coupé ou arraché, maladie professionnelle permanente, brûlure chimique, etc.)											
Fréquence d'exposition	F1 : rare à assez fréquente F2 : fréquente à continue											
Probabilité d'occurrence	O1 : très faible, car technologie éprouvée et stable O2 : faible O3 : élève ou événement relié à une action humaine											
Possibilité d'évitement	P1 : possible dans certaines conditions (ex. : réflexe) P2 : impossible											

Tableau 12 : Analyse du risque

8 Recommandations et conclusions

Il est important que la conception physique du robot, qui inclut les pièces mécaniques et électroniques, soit terminée dans les délais fixés, afin de ne pas empiéter sur les périodes prévues du développement logiciel. En effet, il sera bénéfique de disposer d'une période de développement plus longue pour l'implémentation de l'algorithme de marche, car c'est un élément critique de la conception. Aussi, l'implémentation des autres modules logiciels, tels que le traitement d'images, le choix de comportement et l'exécution d'actions ne doit pas être négligée.

Étant donné que RobotFoot s'échelonnera sur plusieurs années pour finalement se concrétiser par la participation à la compétition RoboCup dans la catégorie de taille moyenne, plusieurs équipes se succéderont. Ainsi, à la fin de notre engagement en décembre 2013, nous sommes persuadés de respecter les objectifs déterminés. Ceux-ci se résument à la conception de deux exemplaires d'un prototype qui marchera de façon stable et pourra détecter une balle mobile d'une couleur déterminée afin de se diriger vers celle-ci. Parmi les fonctionnalités qui ne seront pas encore développées à la fin de notre engagement, il y a l'ajout de bras servant de mécanisme de relèvement, la reconnaissance de sons, la communication sans fil entre les joueurs et l'ajout de comportements spécifiques aux rôles des membres d'une équipe de soccer.

Nous sommes confiants de la pertinence de RobotFoot, initié par notre client Wael Suleiman, qui viendra enrichir l'expertise en robotique de l'Université de Sherbrooke. D'ailleurs, nous sommes ravis de l'intérêt porté par nos commanditaires actuels Freescale et RobotShop et nous espérons que ces partenariats subsistent pendant plusieurs années afin que les prochaines équipes puissent en bénéficier.