

# ***Cahier des Charges - Semestre 4 - EPITA***

Clerc Killian (Chef de Projet)  
Bellot Baptiste  
Bruneteau Alexis  
Genillon Paul

***Janvier 2022***



## Table des Matières

<b>I.</b>	<b>Introduction</b>	<b>3</b>
<b>II.</b>	<b>Présentation du groupe</b>	<b>5</b>
I.	CLERC Killian (Chef de projet) . . . . .	5
II.	BELLOT Baptiste . . . . .	5
III.	BRUNETEAU Alexis . . . . .	6
IV.	GENILLON Paul . . . . .	6
<b>III.</b>	<b>Objectif</b>	<b>7</b>
<b>IV.</b>	<b>État de l'art</b>	<b>7</b>
<b>V.</b>	<b>Explication des tâches</b>	<b>8</b>
I.	Recuperation des données du capteur . . . . .	8
I -	Communication entre le capteur et l'ATmega (ou esp32) . .	8
II -	Communication entre l'ATmega et le PC . . . . .	8
II.	Création d'une surface à partir d'un nuage de points . . . .	8
III.	Rendu Graphique . . . . .	9
I -	Interface graphique : GTK . . . . .	9
II -	OpenGL . . . . .	10
IV.	Sauvegarde d'un fichier . . . . .	11
V.	Recherche du plus court chemin . . . . .	12
I -	Création de la classe "Graph" . . . . .	12
II -	Plus court chemin . . . . .	12
<b>VI.</b>	<b>Répartition des tâches et planning de réalisation</b>	<b>13</b>
<b>VII.</b>	<b>Coûts</b>	<b>13</b>
<b>VIII.</b>	<b>Site Web</b>	<b>14</b>
<b>IX.</b>	<b>Conclusion</b>	<b>15</b>

## I. Introduction

Dans le cadre du projet de programmation du semestre 4, nous avons prévu de réaliser un robot autonome qui serait capable de se déplacer dans un espace et de matérialiser en trois dimensions l'environnement dans lequel il se trouve. Ce projet sera programmé en C, et nous utiliserons une arduino pour la réalisation du robot. Il s'agit donc d'une reconstruction tridimensionnelle de l'environnement d'un robot mobile dans un environnement statique.

Nous avons réfléchi à de multiples idées avant d'en arriver à celle que nous envisageons de réaliser au cours de ce semestre. Pendant nos premières discussions, nous avons pensé à réaliser un "plugin" de conception automatisée de slides de powerpoint, au final nous avons remarqué que cela semblait compliqué voire impossible de faire ceci en C. Nous avons également pensé à la conception d'une calculatrice scientifique mais nous n'avons pas retenu l'idée car nous avons peur d'avoir des programmes qui se ressemblaient trop. Nous avons également pensé à faire un logiciel de traitement audio ou un logiciel de gestion (transports, routes...) mais avons rapidement abandonné ces idées. Après cela, nous est venue une nouvelle idée qui semblait déjà bien plus avancée et ambitieuse, il s'agissait d'un robot autonome qui, en se déplaçant, traçait sur l'ordinateur ses déplacements. L'objectif de ce robot aurait été de lui faire trouver le plus court chemin entre sa position et un point déterminé par l'utilisateur. Nous avons pensé que cela était trop simple et pas assez algorithmiquement intéressant. Après discussion avec notre enseignant, nous en sommes donc venu à l'idée citée dans le premier paragraphe.

Le nom de notre projet vient directement de notre cette idée de robot autonome. En effet, l'idée de ce sujet vient de la Cartographie et Localisation Simultanée, "Simultaneous localization and mapping". L'acronyme étant donc SLAM, il signifie "claquer" en français, d'où le nom de notre projet. Étonnant n'est-il pas ?

Nous avons decoupé les tâches de façon à ce que chacun d'entre nous puisse travailler sur quelque chose qui lui plaise et qui lui fasse envie. Notre but étant avant tout de prendre du plaisir. La répartition des tâches à été faites de façon homogène de manière à ce que nous ayons chacun une quantité de travail similaire entre chaque soutenance.

Ce cahier des charges vous présentera les différents aspects de notre projet, quelles ont été les étapes de notre reflexion, l'origine du projet et ce qu'il implique. Il n'y a aucun doute sur le fait qu'il nous apportera beaucoup, autant pour améliorer nos compétences de développement que pour la cohésion de groupe.

## II. Présentation du groupe

### I. CLERC Killian (Chef de projet)

Pour ma part, ce projet est celui qui m'enthousiasme le plus ! J'ai déjà pu faire deux autres projets comme C#erberus lors du semestre 2 et l'OCR lors du 3ème semestre. J'ai pu m'améliorer et apprendre de nouvelles choses qui me serviront surement pour ce projet ! Le langage C ne s'aborde pas du tout comme du langage POO, grâce à l'OCR j'ai pu m'habituer à manier ce langage et à faire ce que je voulais très rapidement !

J'espère que ces compétences acquises vont m'aider pour pouvoir faire quelque chose de qualitatif pour ce projet.

### II. BELLOT Baptiste

Ce nouveau projet représente un nouveau défi de taille conséquente je pense, puisque cela fait maintenant un an que je n'ai pas travaillé en groupe. Néanmoins, je pense que nous posons dans ce cahier des charges de bonnes bases pour réaliser ce projet, chose dont nous avons manqué au S2 avec mon groupe.

Je pense que le sujet sur lequel nous nous lançons ici est très intéressant dans les nombreux domaines qu'il touche. Notamment dans la modélisation, car j'avais eu la chance de me perdre sur Wikipedia et de tomber sur les notions de triangulation de Delaunay et Diagramme de Voronoï. Ainsi, j'espère pouvoir comprendre ces théorèmes et les implémenter dans notre programme.

### III. BRUNETEAU Alexis

Pour moi ce projet est tout particulièrement intéressant, en effet, le fait que le sujet soit libre et choisis par nous-même nous permet d'allier travail et passion. De plus la confection d'un robot m'a toujours intéressé.

L'idée de ce projet me plait donc beaucoup, d'une part par le côté passionnant du sujet, et d'autre part par le défi technique que nous nous sommes imposés en faisant ce choix.

### IV. GENILLON Paul

En ce qui me concerne, ce projet de programmation est mon troisième plus gros projet ! Après le jeu C#erberus au semestre 2 et l'OCR au semestre 3, j'ai pu acquérir une certaine expérience qui me permettra de mieux aborder ce projet.

L'idée de ce projet me plait énormément et j'ai hâte de commencer à travailler dessus ! Le projet OCR en C m'a déjà permis d'avoir un premier aperçu de réalisation de projet dans ce langage que je ne connaissais pas au début de l'année, et j'espère que ce projet ambitieux me permettra d'acquérir de nouvelles compétences intéressantes.

### III. Objectif

L'objectif du projet sera, grâce à un détecteur accroché sur un robot, de modéliser un environnement **statique** en 3D sur un ordinateur. On pourra se déplacer dans cette modélisation pour pouvoir découvrir l'environnement. Il sera possible d'ordonner au robot de se déplacer vers un point précis, il pourra alors prendre le plus court chemin pour s'y rendre et continuer à modéliser l'environnement à partir de cet endroit. Le logiciel pourra sauvegarder la modélisation afin de pouvoir la recharger dans le futur au besoin.

### IV. État de l'art

De part nos recherches, nous avons trouvé de multiples projets et thèses qui pourront et sauront nous inspirer ainsi que nous aiguiller dans la conception de ce projet.

- Thèse de reconstruction tridimensionnelle de l'environnement d'un robot mobile à partir d'informations de vision omnidirectionnelle, pour la préparation d'interventions, Rémi Boutteau
- L'aspirateur Roborock S6 MaxV modélise en 2 dimensions l'environnement dans lequel il doit se déplacer pour le nettoyer

## **V. Explication des tâches**

### **I. Recupération des données du capteur**

#### **I - Communication entre le capteur et l'ATmega (ou esp32)**

Pour la récupération des données du capteur (VL53L0X / VL53L1X) par notre microprocesseur, nous utiliserons de l'I2C, le moyen de communication de nos capteurs.

#### **II - Communication entre l'ATmega et le PC**

Pour la communication entre le microprocesseur et le PC nous pouvons soit passer en filaire soit en sans fil, le filaire est plutôt simple à mettre en place mais cependant il sera peu pratique pour le déplacement du robot. Pour le sans fil, soit bluetooth, soit Wi-Fi, les difficultés techniques sont plus grandes, cependant il sera beaucoup plus pratique pour le robot de se déplacer.

### **II. Création d'une surface à partir d'un nuage de points**

Pour cette partie, nous allons nous aider d'un cours sur la reconstruction de surface de l'Inria, qui présente plusieurs thèses dont:

- Inverse geometry : from the raw point cloud to the 3d surface : theory and algorithm, Julie Digne
- Surface Reconstruction from Unorganized Points, Hugues Hoppe
- Voronoi-based Variational Reconstruction of Unoriented Point Sets, P.Alliez et al., SGP 2007

Notre plan d'action sera de choisir plusieurs méthodes et de déterminer les meilleurs pour notre projet.



### **III. Rendu Graphique**

Concernant le rendu graphique de notre logiciel, nous allons utiliser deux outils: Le premier GTK pour la mise en place de l'interface utilisateur (UI), et le second OpenGL qui sera intégré dans notre UI et qui servira à montrer le rendu de notre modélisation.

#### **I - Interface graphique : GTK**

L'interface graphique permettra à l'utilisateur de connecter le robot pour qu'il effectue sa modélisation de l'environnement, d'afficher cette modélisation qui est réalisée grâce à OpenGL et d'enregistrer cette modélisation.

Cette interface graphique sera réalisée grâce à GTK et à Glade. Glade nous permettra notamment de concevoir l'aspect visuel de l'interface. Nous allons faire dans un premier temps une interface assez simple pour assurer un résultat fonctionnel, et si nous avons le temps nous tenterons de l'améliorer pour la rendre plus agréable d'utilisation.

## II - OpenGL

Pour afficher le rendu de notre modélisation, nous allons utiliser OpenGL. En sachant qu'il existe plusieurs version de cette librairie, nous allons utiliser les versions supérieures à la 3.0. Nous hésitons à choisir la version Vulkan (dernière 1.3 le 25/01/2022), mais celle-ci n'assure pas de compabilité avec les anciennes versions. L'avantage de cette dernière est la possibilité de créer une application multi-plateforme, dont mobile.

Afin d'apprendre à utiliser cette librairie (OpenGL en général), nous allons avoir recours à différents tutoriels conseillés sur des slides de cours de l'université de Lille.

- <http://www.arcsynthesis.org/gltut/>
- <http://www.opengl-tutorial.org/>
- <http://ogldev.atspace.co.uk/>

Dans la fenêtre OpenGL ouverte dans GTK, nous afficherons donc le nuage de points avec une possibilité de se mouvoir dans la modélisation. Comme nous aurons un affichage avec des surfaces, nous utiliserons aussi du ray-tracing pour mettre en transparence la surface qui se situe entre la caméra et le point représentant notre capteur sur la modélisation. Dans le but de pouvoir utiliser un algorithme de plus court chemin nous devront placer un point dans la modélisation. Nous avons deux solutions pour le placer. Avec la souris ou bien avec un sextuplet de touches. Nous pensons que le plus réalisable dans ces 5 mois est la seconde option mais si l'occasion se présente nous implémenterons la première ou une option hybride.

#### IV. Sauvegarde d'un fichier

Étant donné que nous allons travailler sur des nuages de points, il est important de choisir des formats de stockages intéressants. Parmi ceux-ci, on peut retrouver :

- Le format XML (pas très économique en place donc on évitera)
- Le format OFF, pour Object File Format dont les spécifications sont

##### Spécification :

- entête : *OFF*
- [*nombre de sommets*] [*nombre de faces*] [*nombre d'arêtes*]
- liste de points [*x y z*]
- liste de faces [*nombre de sommets*] [*n<sub>0</sub> n<sub>1</sub> ... n<sub>i</sub>*]

- Le format OBJ(MOD en binaire) qui stock les informations sous forme de polygones ou surfaces lisses. Il a les spécifications suivantes

##### Spécification (partielle) :

- liste de points [**v** *x y z*] (*v* pour vertex)
- liste de coordonnées de texture [**vt** *u v*] (*facultatif*)
- liste de normales [**vn** *x y z*] (*facultatif*)
- liste de points [**f** *v<sub>0</sub>/vt<sub>0</sub>/vn<sub>0</sub> v<sub>1</sub>/vt<sub>1</sub>/vn<sub>1</sub> ... v<sub>i</sub>/vt<sub>i</sub>/vn<sub>i</sub>*] (ou simplement [**f** *v<sub>0</sub> v<sub>1</sub> ... v<sub>i</sub>*])

- Le format PLY, pour Polygon File Format. Qui stock les informations sous listes de polygones nominalement plats.
- Le format STL, STereo-Lithography. Il décrit un objet par sa surface externe (fermée), composée de multiples triangles. Il ne stock pas les données de textures, couleurs etc.

A noter, la plupart de ces formats ont une version ASCII, facile à créer, et une version binaire. La seconde étant moins volumineuse, c'est celle que nous allons utiliser.

## V. Recherche du plus court chemin

### I - Création de la classe "Graph"

Pour la recherche du plus court chemin, nous allons implémenter notre propre classe pour utiliser les graphes. Nous implémenterons les fonctions principales, notamment :

- La construction de la liste d'adjacence.
- Un booléen indiquant si le graphe est orienté ou non.
- La possibilité d'ajouter des noeuds et arcs/arrêtes.

### II - Plus court chemin

Nous utiliserons la classe que nous avons créée pour la recherche du plus court chemin entre les différents sommets. Il sera possible d'indiquer au robot où se déplacer, et il s'y rendra en passant par le chemin le plus court qu'il trouvera.

Pour cela, on utilisera l'un des algorithmes suivant :

- Algorithme de Dijkstra
- Algorithme A\*
- Algorithme de Bellman-Ford
- Algorithme de Floyd-Warshall

Tout ces algorithmes ont des petites différences notables mais nous ne pouvons pas choisir lequel sera le meilleur à utiliser pour le moment.

## VI. Répartition des tâches et planning de réalisation

Planning de réalisation du projet :

Tâches	Soutenance 1	Soutenance 2	Soutenance 3
Site Web	50%	75%	100%
Données capteur	10%	50%	100%
Nuage de points	0%	50%	100%
OpenGL	25%	75%	100%
GTK	20%	50%	100%
Voxel	10%	50%	100%
Sauvegarde	50%	100%	-
Plus court chemin	25%	75%	100%

Répartition des tâches :

Tâches	Killian	Alexis	Baptiste	Paul
Site Web				X
Données capteur		X		
Nuage de points			X	
OpenGL			X	
GTK				X
Voxel	X			
Sauvegarde			X	
Plus court chemin				X

## VII. Coûts

Dans la pratique notre projet n'engendre que peu de coûts, en effet presque aucun coût logiciel, cependant le matériel permettant la fabrication du robot pourrait être coûteux (à déterminer), à savoir les moteurs, la puce ATmega ou ESP32 (sous forme d'arduino).

## VIII. Site Web

Ce site sera le support principal pour suivre notre avancée. Il permettra à des personnes extérieures de consulter nos différents rapports (Cahier des charges, rapports de soutenance ..) ainsi que de télécharger le projet.

Le site sera également composé... :

- d'une page d'accueil
- de l'historique du projet
- une page contenant un devblog
- d'une présentation des membres du groupe
- de la chronologie de réalisation
- d'une page de ressources (liens vers les sites et outils qui nous ont été utiles pour la conception du projet)
- d'une page pour télécharger les différents rapports de soutenance ainsi que le cahier des charges
- d'une page pour télécharger le projet avec ses différents manuels (avec une version lite sans les documents)

N'ayant pas de compétences dans le développement de site web, nous avons pris la décision d'héberger notre site web grâce à l'hébergeur Wix.com qui nous permettra d'avoir un rendu satisfaisant pour ce que nous avons à faire.

Le site a déjà été créé, il est accessible en cliquant [ici](#)

## IX. Conclusion

Pour conclure, le projet que nous envisageons de faire est une reconstruction tridimensionnelle de l'environnement d'un robot mobile, sachant que l'environnement sera **statique** pour se simplifier le développement du projet.

Nous sommes conscient que nous nous lançons sur un projet très ambitieux. Chacun des membres du groupe a des qualités qui pourront être mises à profit pour la réalisation de cet ambitieux projet qui nous prendra probablement beaucoup de temps mais nous sommes plus que prêts à relever ce défi d'envergure. Notre objectif est de voir ce projet se concrétiser et ce robot se déplacer tout en modélisant en trois dimensions l'environnement dans lequel il se trouve.

Nous avons plusieurs supports qui pourront nous aider dans la réalisation de ce projet et nous sommes impatients de commencer son développement.