Rapport de soutenance 1 - Semestre 4 - EPITA

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

Janvier 2022

Table des Matières

I. Introduction

I. Rappel du sujet

L'objectif du projet est de modéliser l'environnement **statique** en 3D d'un robot grâce à un capteur de distance LIDAR. 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.

II. Etat d'avancée du projet

Nous avons pu durant ces deux semaines commencer à développer l'infrastructure du projet et de réfléchir à différents aspects pour partir rapidement sur les meilleures bases possibles. Ce rapport soutenance ne représente pas un monstre en terme de technicité, mais plus une façon de présenter de façon plus détaillée notre projet. Nous nous réservons cette partie technique pour les deux prochains rapports qui pourront aborder nos algorithmes de façon plus élaborée.

II. Rappel de la répartition des tâches

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

III. Explication des tâches

I. Mise en place de l'ESP32

Pour compiler sur l'ESP32, il a fallu installer l'IDE fourni par ESP, le fabricant de la puce, cependant, nous pouvons compiler sans forcément passer par l'IDE, il suffit d'utiliser un script python nommé build.py dans notre dossier de projet et notre projet peut être compiler et téléverser sur la puce facilement.

Pour tester la compilation de l'ESP32, j'ai créer un code permettant de faire clignoter la LED présente par défaut sur l'ESP32. Ainsi j'ai pu comprendre comment fonctionnaient les entrées sorties de la carte, et ainsi pu les contrôler. Du fait que j'ai de l'experience avec l'environnement Arduino, j'ai déjà une certaine aisance à coder sur ce genre de système, cependant, Arduino étant un système à visée tout public, beaucoup de choses sont gérer automatiquement. De ce fait j'ai du apprendre à tout gérer manuellement en C. L'ESP32 étant une alternative de l'ESP8266, une puce très utilisée, il est très facile de trouver de la documentation sur la facon dont on peut la faire fonctionner.

II. Récupération des données capteur

Du fait que les capteurs ne sois pas encore arrivés, le code de cette partie là n'as pas pu être commencé, cependant j'ai commencé à m'interesser au mode de communication des capteurs, à leur fonctionnement, et à la facon dont nous allons les agencer.

III. Création d'une surface à partir d'un nuage de points

Nous avons mis en place un prototype de structure se nommant frame nous permettant de facilement mettre en mémoire les points détectés par le robot. Voici les structures:

```
struct frame
{
    struct off_point* points;
}

struct off_point
{
    char sentinel; // 0 if is a point, otherwise is a sentinel
    double x;
    double y;
    double z;
    struct off_point* prev_x;
    struct off_point* prev_y;
    struct off_point* prev_z;
    struct off_point* next_x;
    struct off_point* next_y;
    struct off_point* next_z;
    struct off_point* sent_x;
    struct off_point* sent_y;
    struct off_point* sent_z;
}
```

Nous aurons deux listes 3D, une de sentinelles et une de points détéctés par le robot.

L'idée de cette structure est d'avoir une liste 3D de sentinelles nous permettant d'aller vers n'importe quels points en la parcourant grâce aux pointeurs "prev" et aux pointeurs "next". Depuis l'un des points de cette liste de sentinelles nous pourrons accéder à la liste 3D des points grâce aux pointeurs "sent" à un endroit voulu pour pouvoir la parcourir et trouver notre point. (Voir explications plus bas).

En effet les sentinelles nous permettent de créer des plans qui nous permettent par la suite d'accéder à l'intersection de ces plans et d'accéder à cette même intersection sur la liste 3D des points.

Si nous sommes dans la liste des sentinelles nous pouvons accéder à la listes des points 3D en utilisant les pointeurs "sent" et vice versa, si nous sommes dans la liste des points 3D nous pouvons accéder à la liste des pointeurs en utilisant les pointeurs "sent".

Explications:

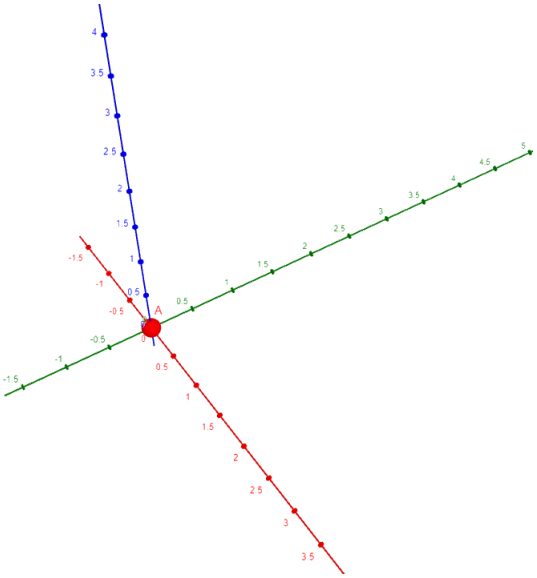

Figure 1: Initialisation de notre structure

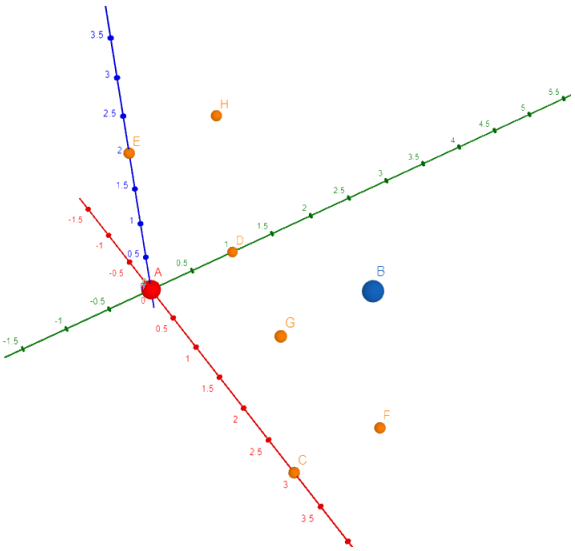

Figure 2: Ajout du point B, x = 3, y = 2, z = 1, dans la liste 3D de points

Nous remarquons la creation de sentinelles, qui nous permettent d'atteindre le point B à partir de n'importe quel solutions qui prends comme sentinelle de départ A, par exemple nous pouvons chercher le plan x = 3 puis ensuite chercher l'intersection avec le plan y = 2, qui nous donnera la sentinelle qui nous renvoie une liste de point, une ligne avec x = 3 et y = 2, on aura donc juste à parcourir s'est "next_x"/"prev_x" et à essayer de trouver un point ayant z = 1. Mais nous pouvons faire ceci par exemple en commençant par z = 1 ensuite l'intersection avec x = 3, etc...

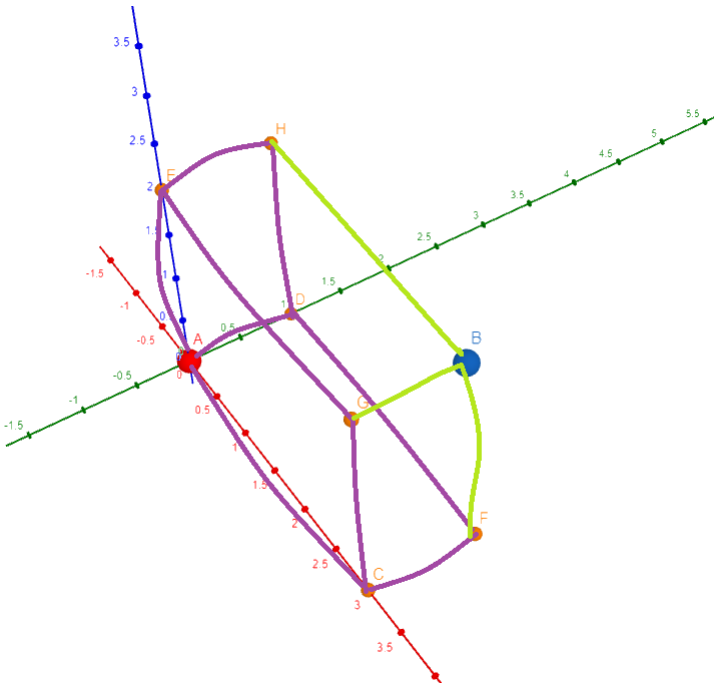

Figure 3: Visualisations des liens

Les liens en violets correspondent aux "next"/"prev" des sentinelles, des directions correspondantes aux liens

Les liens en verts correspondent aux "sent" des sentinelles/points, des directions correspondantes aux liens

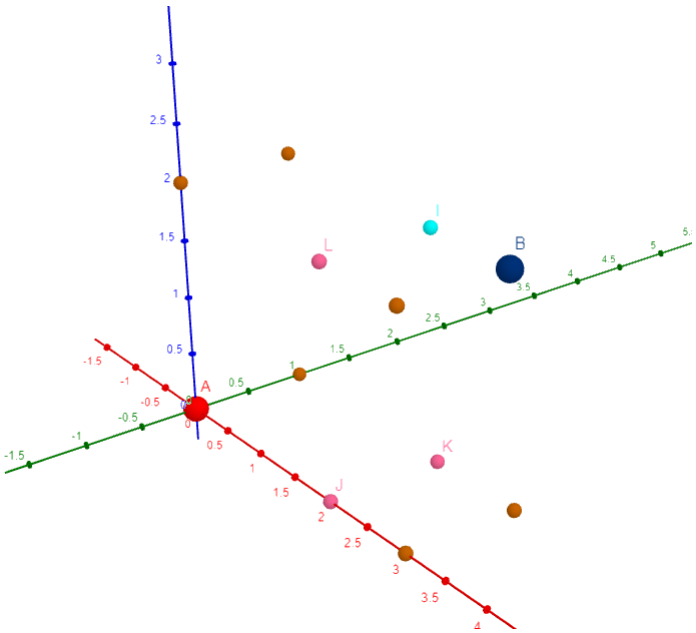

Figure 4: Ajout du point I, x = 2, y = 2, z = 1, dans la liste 3D de points

Nous remarquons la création des sentinelles, qui est en réalité la créations du plan x = 3.

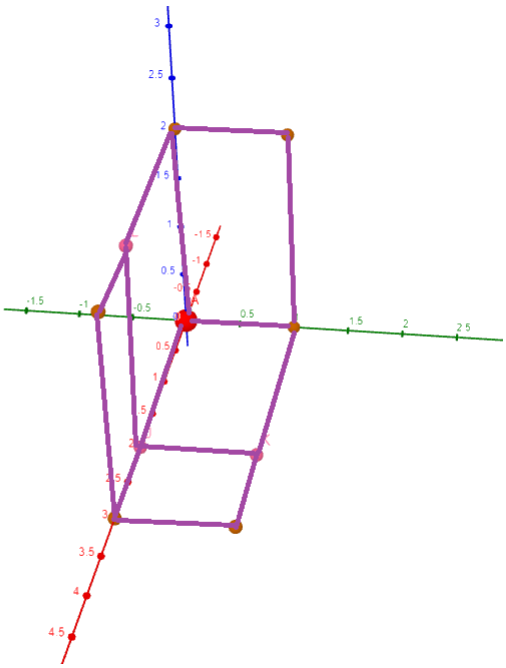

Figure 5: Visualisation de la listes 3D des sentinelles avec les liens "next"/"prev"

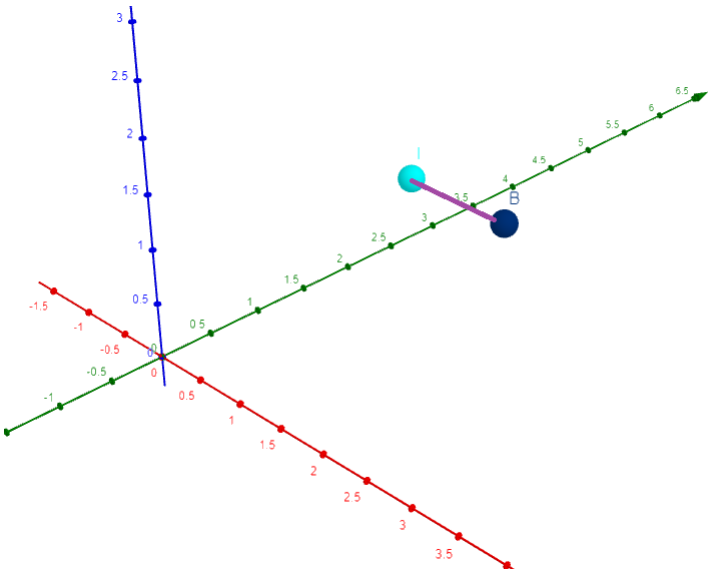

Figure 6: Visualisation de la listes 3D des points avec les liens "next"/"prev"

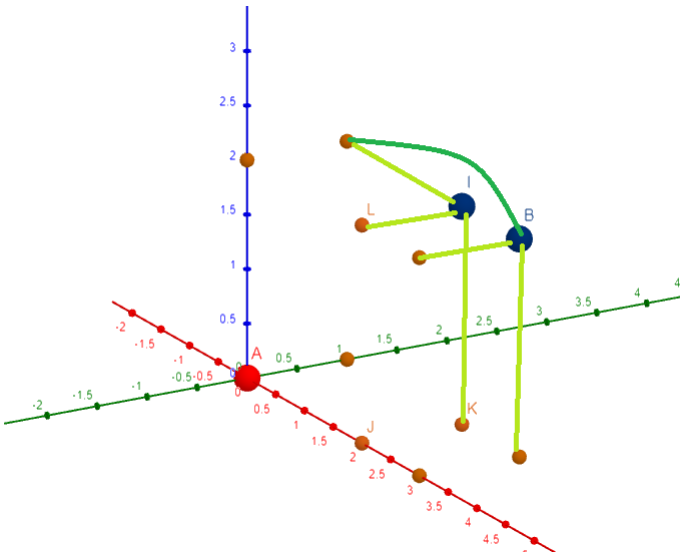

Figure 7: Visualisation des liens "sent"

Les liens vers verts claires vont dans les deux sens, alors que les liens verts foncés vont seulement du point vers la sentinelle.

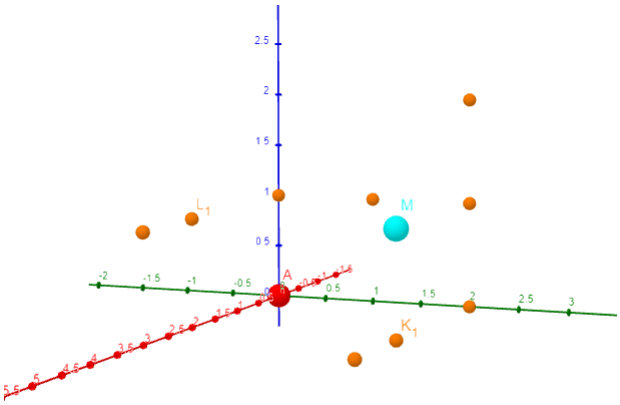

Figure 8: Ajout du point I, x = 2, y = 1, z = 2, dans la liste 3D de points

Seulement les ajouts sont montrés, nous remarquons la création des sentinelles, qui est en réalité la créations des plans y = 1 et z = 2 puisqu'ils n'existaient pas.

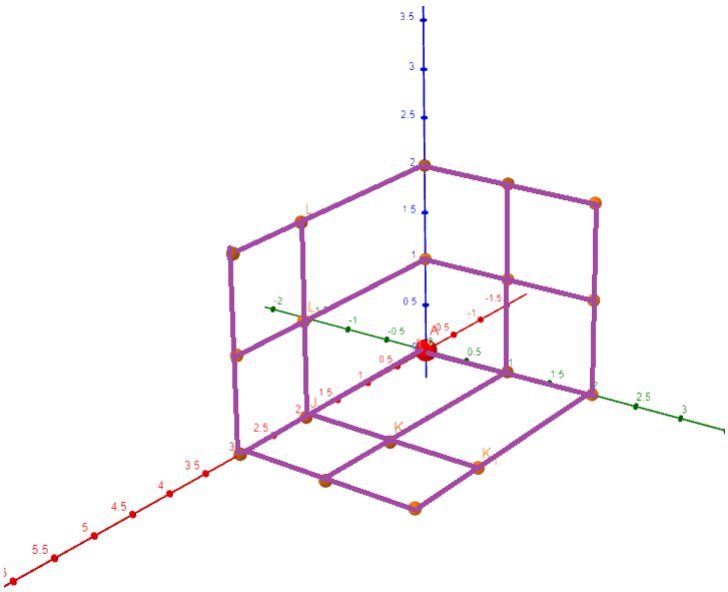

Figure 9: Visualisation de la listes 3D des sentinelles avec les liens "next"/"prev"

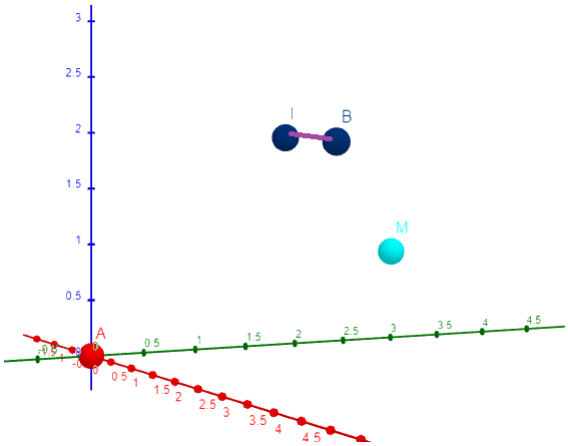

Figure 10: Visualisation de la listes 3D des points avec les liens "next"/"prev"

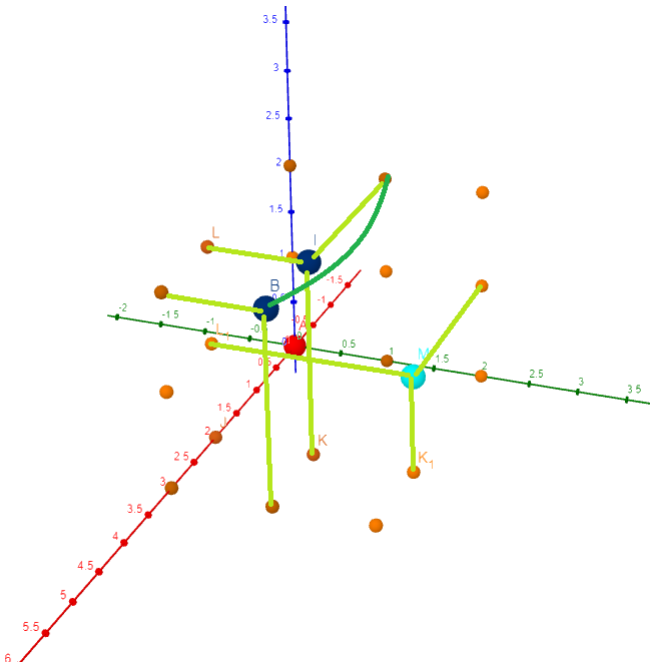

Figure 11: Visualisation des liens "sent"

Les liens vers verts claires vont dans les deux sens, alors que les liens verts foncés vont seulement du point vers la sentinelle.

Fonctionnalités:

Nous avons mis en place plusieurs fonctionnalités, nous pouvons ajouter des points, en supprimer. Ou encore en chercher. Nous pouvons chercher des plans (une seule coordonnée fix) qui nous renvoie une sentinelle, des lignes (deux coordonnées fix) qui nous renvoie une sentinelle.

IV. Rendu graphique

I - 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.

Nous avons commencé à réaliser l'aspect visuel de l'interface graphique mais nous n'avons pas eu le temps d'aller plus loin. Nous sommes confrontés à un problème avec GTK et OpenGL car ils ne semblent pas être compatibles entre eux et nous recherchons une solution pour pallier au problème. Pour afficher du OpenGL, nous avions envisagé d'utiliser un "GtkGLArea widget" mais ne semble pas être compatible avec GTK.

II - OpenGL

A l'inverse de ce qui était prévu, nous n'avons pas commencé à nous occuper d'OpenGL. Si ce n'est l'ajout des différentes librairies.

V. Sauvegarde d'un fichier

La partie sauvegarde des fichiers est terminée, dans son utilité la plus basique pour la première soutenance. C'est-à-dire que nous sauvegardons pour le moment uniquement le nuage de point. Nous ne somme pas encore soucié des arrêtes reliant plusieurs points. En effet, nous n'avons pas encore choisi la structure de ces dernières.

VI. Recherche d'un plus court chemin

I - Création de la structure de Graph

Pour la recherche du plus court chemin, nous avons implémenté les bases de la structure de Graph pour pouvoir faire une recherche de plus cours chemin.

Voici les différentes structures mises en place :

- Une structure de liste de sommets : elle permet d'accéder au sommet suivant celui sur lequel on pointe, ainsi que la valeur du sommet pointé (lors d'une recherche de plus court chemin, le graph doit être valué).
- Une structure pour les listes d'adjacence : elle permet d'accéder à la liste d'adjacence de chaque sommet.
- Une structure de Graph : elle permet de connaitre l'ordre du graph (son nombre de sommets), s'il est orienté ou non, ainsi que d'accéder à la liste d'adjacence de chaque noeud.

II - Recherche du plus court chemin

Nous n'avons pas encore pu commencer le développement de cette partie. Nous avons commencé à réfléchir à la solution la plus adaptée pour ce projet. Il semblerait que nous essayions de faire l'algorithme d'Astar (A*) qui semblerait être le plus optimisé.

VII. 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.

Sa création a déjà bien commencé. L'ensemble de la structure a été réalisée. Il comporte actuellement :

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

Parmi les pages citées au dessus, bon nombre d'entre elles ne sont pas encore fonctionnelles. Nous avons d'abord voulu travailler sur les bases du site et avons commencé à développer certaines pages plus en profondeur.

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

Le site est accessible en cliquant ici

VIII. Git

Pour ce projet, nous utiliserons GitLab. Le projet est accessible (accessible ici). Il est constitué de six branches. Une par développeur plus deux autres qui sont develop et main. Pour cette soutenance, nous avons choisi de ne pas merge nos versions puisque celles-ci ne se complète pas ni ne dépende les unes des autres. Ainsi, dans la branche main, nous avons 4 dossiers aux noms de leur branche de base respective (donc pas de develop). Le README.md est celui généré par GitLab mais il sera modifié par le futur pour en faire une première introduction à notre projet.

En ce qui concerne les merges, nous voulons en faire un par semaine pendant une réunion ou nous serons tous présents.

IV. Coûts

En ce qui concerne les coûts, nous avons du acheter du matériel et notamment :

- 1 ESP 32 : 9€
- 1 TOF 4M : 5,65€
- 4 TOF 2M : 13,92€

V. 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.

Au cours de ces deux premières semaines de projet, nous avons pu commencer à définir les bases de notre projet. Le git, les structures et classes de base, mais aussi les librairies pour la connection à l'esp-32.

Les semaines à venir seront nettement plus interessantes dans le developpement de ce projet, nous aurons plus de temps pour developper des systèmes plus interessants qui donneront vie à notre projet.

Nous avons plusieurs supports qui pourront nous aider dans la réalisation de ce projet et qui nous ont déjà servi pour certains.