

2017

CPE Lyon Projet 4ETI

Jacques SARAYDARYAN

[SPECIALITE : CAHIER DES CHARGES]

Ce document est un cahier des charges livré par le client auquel doit répondre le participant.

Table des matières

1. Introduction.....	3
2. Information générale.....	3
3. Création d'un simulateur de robot.....	4
3.1. Présentation [FCT-INFO0-Simulator].....	4
3.1.1. Objectif	4
3.2. [FCT-INFO0- Simulator -Env].....	4
3.3. [FCT-INFO0- Simulator -Robot].....	4
3.4. [FCT-INFO0- Simulator - Measures].....	5
3.5. [FCT-INFO0- Simulator - RobotCrt].....	5
4. Création des interfaces d'interactions	5
4.1. [FCT-INFO1-TelCom].....	5
4.1.1. Objectif	5
4.1.2. Moyen.....	5
4.1.3. Informations complémentaires	6
4.2. [FCT-INFO2-Mesure].....	7
4.2.1. Objectif	7
4.2.2. Moyen.....	8
4.2.3. Informations complémentaires	8
4.3. [FCT-INFO3-Carte]	9
4.3.1. Objectif	9
4.3.2. Moyen.....	9
4.3.3. Informations complémentaires	9
4.1. [FCT-INFO4-AutoNav].....	11
4.1.1. Objectif	11
4.1.2. Moyen.....	11
4.1.3. Informations complémentaires	11
5. Annexes techniques	13
5.1. Utilisation de JAVA (Servlet) comme fournisseur d'information (JAX-RS)	13
L'exemple ci-dessus permet d'envoyer des informations à l'url 127.0.0.1 :8080/<your projet name>/rest/cmd/<command>.....	13
5.2. Afficher une matrice de string en java (text).....	13
5.3. Utilisation des Canvas HTML5	14
5.3.1. Références	14

5.4.	Utilisation de JSON MAP	14
5.4.1.	Références	14
5.5.	Affichage des graphes	14
5.5.1.	Références	14

1. Introduction

Le présent document recense l'ensemble des fonctionnalités à fournir par le participant de la spécialité.

Des groupes de fonctionnalités seront regroupés en lots. Les premiers lots devront être réalisés séquentiellement.

- **[FCT-INFO1-Simulator]**-Création d'un simulateur de robot
- **[FCT-INFO2-Mesure]**-Affichage des informations contextuelles envoyées par le robot
- **[FCT-INFO2-Carte]**-Affichage d'une carte de déplacement du robot
- **[FCT-INFO2-AutoNav]**-Navigation automatique du robot

2. Information générale

L'ensemble des lots et sous lots peuvent présenter des difficultés différents notées ci-dessous:

Lots	Objectif	Difficultés		
		1	2	3
[FCT-INFO0-Simulator]				
	[FCT-INFO0-TelCom-Env]	X		
	[FCT-INFO0-TelCom-Robot]		X	
	[FCT-INFO0-TelCom-Measures]	X		
	[FCT-INFO0-TelCom-RobotCrt]	X		
[FCT-INFO0-TelCom]				
	[FCT-INFO1-TelCom-Conn]	X		
	[FCT-INFO1-TelCom-WebCmd]	X		
	[FCT-INFO1-TelCom-WebAdm]	X		
[FCT-INFO2-Mesure]				
	[FCT-INFO2-Mesure-Aff]		X	
	[FCT-INFO2-Mesure-AffAuto]		X	
	[FCT-INFO2-Mesure-AffGraph]			X
[FCT-INFO3-Carte]				
	[FCT-INFO3-Carte -Traj]	X		
	[FCT-INFO3-Carte -AffTraj]		X	
	[FCT-INFO3-Carte -Obs]			X
[FCT-INFO4-AutoNav]				
	[FCT-INFO4-AutoNav-customdisplay]		X	
	[FCT-INFO4-AutoNav-autoLauncher]		X	
	[FCT-INFO4-AutoNav-slam]			X

L'évaluation de la difficulté tient compte de la complexité de la tâche à réaliser ainsi que du temps nécessaire pour son développement. Ce tableau a pour vocation à vous donner une indication de l'effort d'investissement à fournir sur chacun des lots et sous-lots.

3. Création d'un simulateur de robot

3.1. Présentation [FCT-INFO0-Simulator]

3.1.1.Objectif

Le lot suivant devra remplir les fonctionnalités suivantes :

- **[FCT-INFO1- Simulator -Env]** : Générer automatique un environnement avec des obstacles
- **[FCT-INFO1- Simulator -Robot]** Générer un modèle de robot pouvant évoluer dans l'environnement
- **[FCT-INFO1- Simulator -Measures]** récupérer des mesures lors de l'exécution du simulateur
- **[FCT-INFO1- Simulator -RobotCrt]** Créer un contrôleur permettant de déplacer le robot et de renvoyer des informations

3.2.[FCT-INFO0- Simulator -Env]

L'objectif de ce lot est de créer un environnement sous forme de matrice/grille de taille paramétrable contenant zone libre et des obstacles.

L'objet Environnement, **Env**, devra posséder les propriétés suivantes :

- Paramètre d'initialisation
 - Taille de la grille/matrice
 - Pourcentage de change d'avoir un obstacle dans une cellule
- Une méthode de génération d'environnement : création de la matrice, ajout d'obstacle aléatoirement respectant les paramètres d'initialisation
- Une méthode permettant de déterminer le contenu d'une case/cellule (FREE/OBSTACLE/UNKNOWN) à des coordonnées x,y
- Une méthode permettant d'afficher la matrice sous forme ascii (cf. Annexe)

3.3.[FCT-INFO0- Simulator -Robot]

L'objectif de ce lot est de créer un modèle de robot qui évoluera dans cet environnement

L'objet Robot, **Robot**, devra posséder les propriétés suivantes :

- Paramètre d'initialisation
 - Position initiale
 - Orientation initiale
- Attributs :
 - Coordonnées x,y dans l'environnement
 - Orientation du robot (N ,E,S,W)

- L'environnement découvert par le robot
- Un capteur de vision (objet définissant un pattern de vision du robot en fonction de son orientation)

3.4.[FCT-INFO0- Simulator - Measures]

L'objectif de ce lot est de créer un conteneur de compteur

L'objet Measures, **Measures**, devra posséder les propriétés suivantes :

- Attributs :
 - Nombre de commandes exécutés sur le robot depuis le lancement du simulateur
 - Nombre d'obstacles rencontrés par le robot depuis le lancement du simulateur
 - Nombre d'obstacles visibles par le robot
 - Distance parcouru par le robot depuis le lancement du simulateur

3.5. [FCT-INFO0- Simulator - RobotCrt]

L'objectif de ce lot est de créer un controleur permettant de déplacer le robot dans l'environnement

L'objet **RobotCrt**, devra posséder les propriétés suivantes :

- Paramètre d'initialisation
 - Un environnement **Env**
 - Un robot **Robot**
- Des méthodes permettant de déplacer le robot (**UP,DOWN,LEFT,RIGHT**)
- Une méthode permettant de retourner ce que perçoit le capteur de vision du robot
- Une méthode permettant de récupérer l'environnement utilisé **Env**.
- Une méthode permettant de récupérer l'environnement perçu par le **Robot**

4. Création des interfaces d'interactions

4.1.[FCT-INFO1-TelCom]

4.1.1.Objectif

Le lot suivant devra remplir les fonctionnalités suivantes :

- **[FCT-INFO1-TelCom-Conn]** : Connecter le robot à un serveur web
- **[FCT-INFO1-TelCom-WebCmd]** Interagir avec le robot depuis une page web
- **[FCT-INFO1-TelCom-WebAdm]** Arrêter/démarrer la commande du robot depuis une page d'administration

4.1.2.Moyen

- Utiliser l'objet **RobotCrt** positionné dans l'espace de mémoire application du serveur web.
- Utiliser une base de données Mysql permettant de stocker les utilisateurs administrateurs

4.1.3. Informations complémentaires

Deux pages web devront être créées :

-cmd.html : permettant à l'utilisateur de prendre la main sur la commande (réserver la commande), permettant à un utilisateur de commander le robot

-admin.html : accessible uniquement par les administrateurs du système permettant d'arrêter/démarrer la télécommande du robot depuis le web

-Un seul utilisateur de la télécommande pourra déplacer le robot, les autres utilisateurs recevront un message d'attente.

Les pages web devront contenir les informations suivantes (ergonomie et position à titre de proposition) :

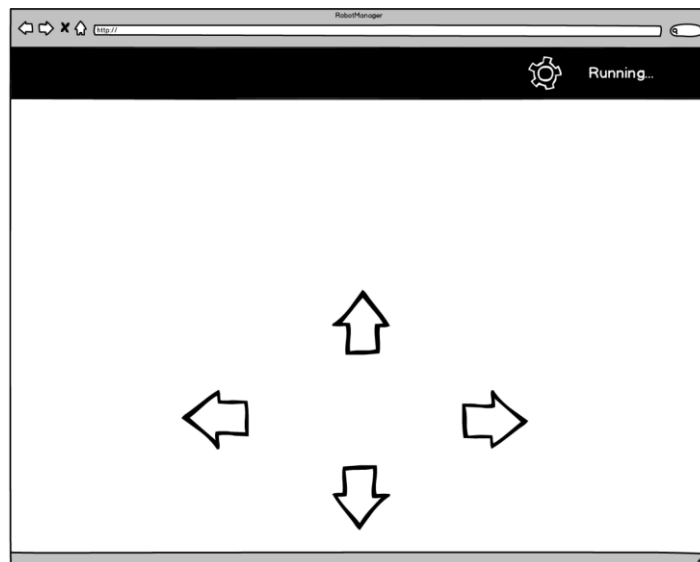


Figure 1: cmd.html

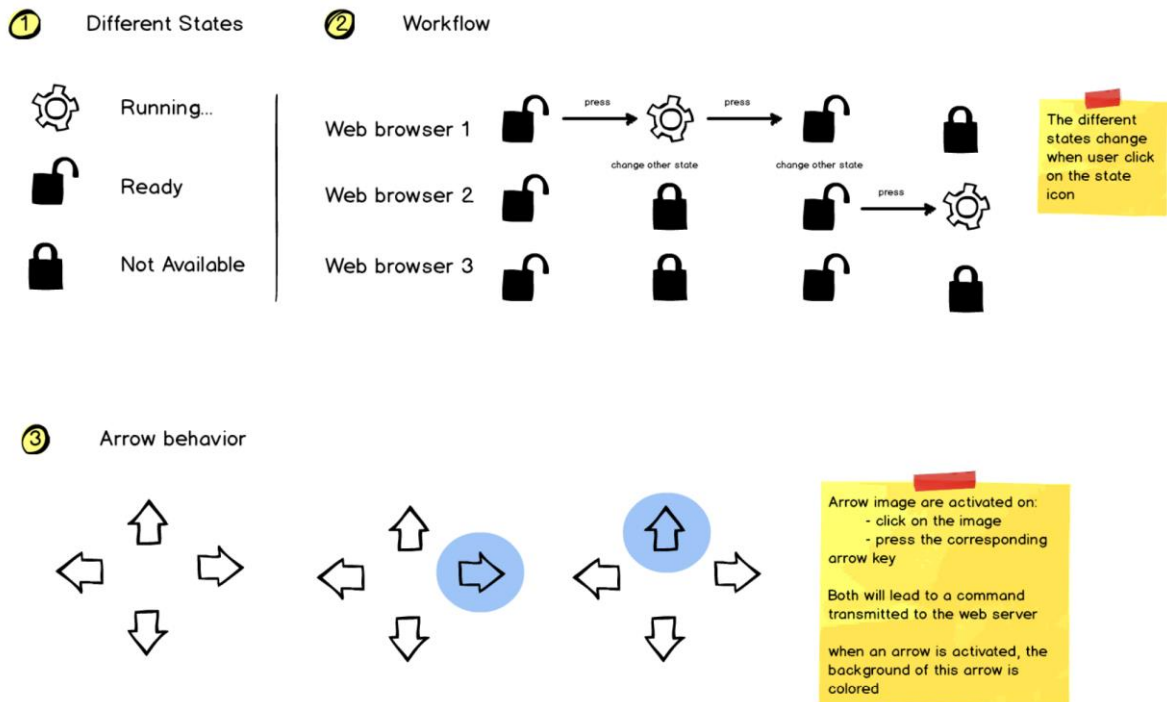


Figure 2: cmd.html comportement

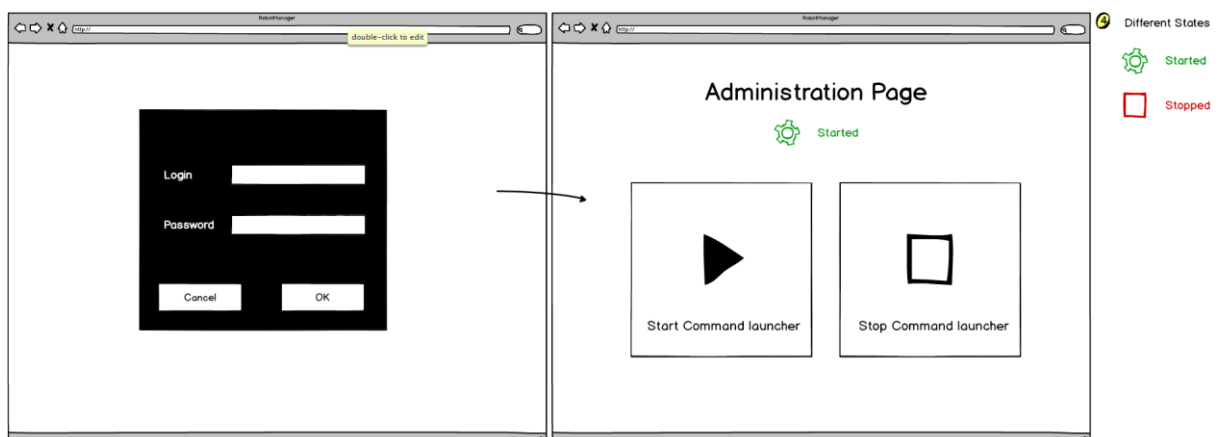


Figure 3: Admin.html

4.2.[FCT-INFO2-Mesure]

4.2.1.Objectif

Le lot suivant devra remplir les fonctionnalités suivantes :

- **[FCT-INFO2-Mesure-Aff]** : Afficher les différentes informations disponibles sur le robot (rafraichissement par bouton)
- **[FCT-INFO2-Mesure-AffAuto]** Afficher les différentes informations disponibles sur le robot (rafraichissement auto toute les 5 secondes)
- **[FCT-INFO2-Mesure-AffGraph]** Afficher un historique des données collectées sous forme de graphe.

4.2.2.Moyen

- Utiliser l'envoi de commande du lot [FCT-INFO1-TelCom] pour demander des informations sur le robot.
- Utiliser un bouton javascript + AJAX afin de demander et recevoir les informations du robot
- Utiliser des librairies javascripts pour afficher les graphes (exemple Chart.js)

4.2.3.Informations complémentaires

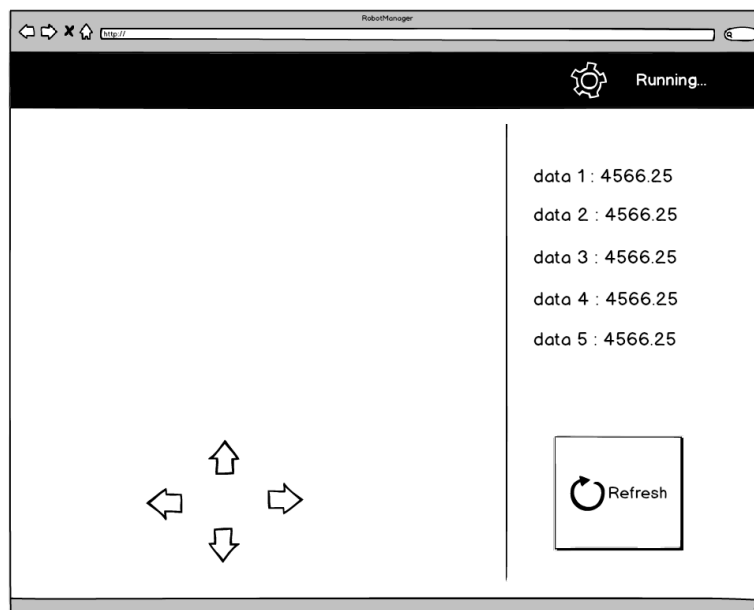


Figure 4: Collecte des données + rafraichissement manuel

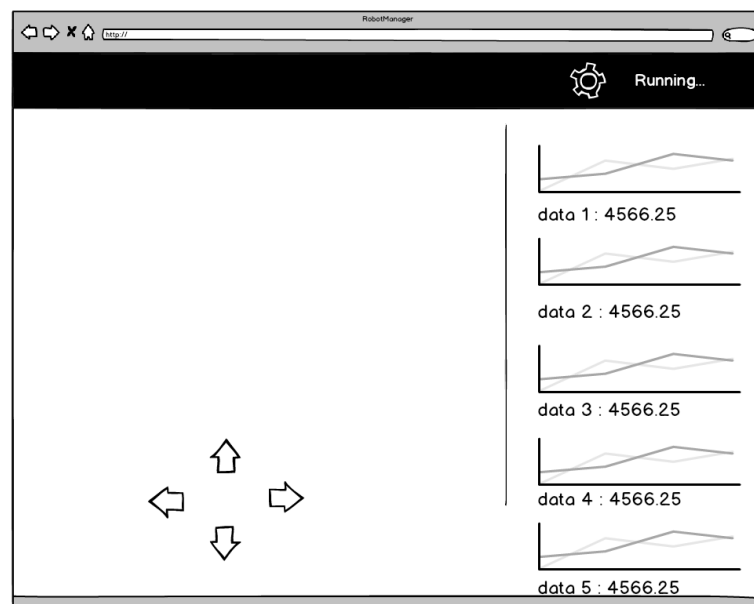


Figure 5: Collecte d'information + graphe + update auto

4.3.[FCT-INFO3-Carte]

4.3.1.Objectif

Le lot suivant devra remplir les fonctionnalités suivantes :

- **[FCT-INFO3-Carte –Traj]** Réaliser un enregistrement de la trajectoire du robot
- **[FCT-INFO3-Carte –AffTraj]** Afficher cette trajectoire
- **[FCT-INFO3-Carte –Obs]** Marquer et afficher les obstacles rencontrés par le robot sur l'interface

4.3.2.Moyen

- Utilisation des canvas afin d'afficher la position depuis la télécommande (attention à l'échelle).
- Ajout graphique simple dans le canvas (forme géométrique, image).

4.3.3.Informations complémentaires

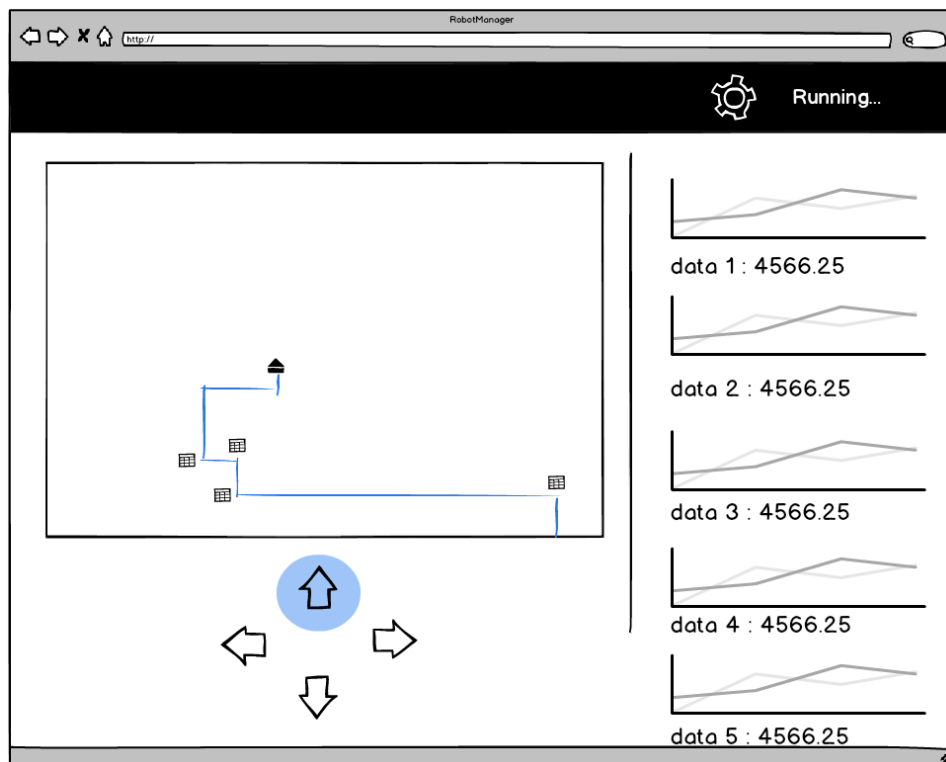


Figure 6: Affichage de trajectoire et d'obstacles

- A** Trajectory display
- ① ▲ Current position of the robot
 - ② — Robot trajectory
 - ③ ■ Detected Obstacles

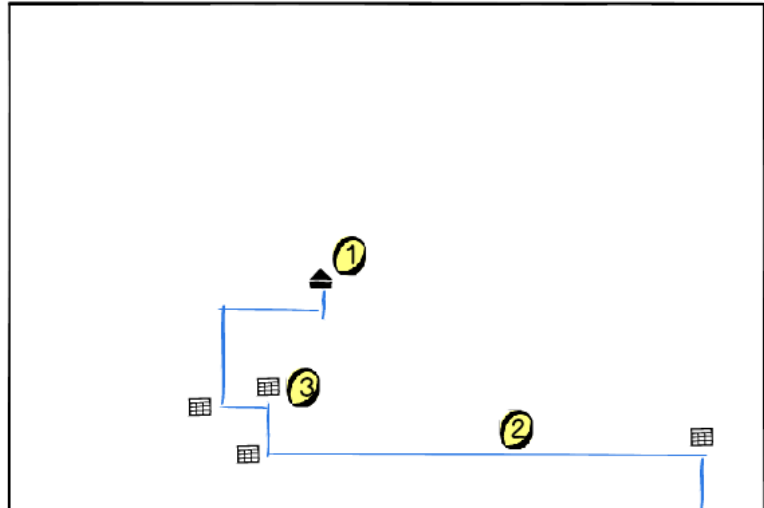


Figure 7: Description détaillée de l'affichage de la trajectoire

4.1.[FCT-INFO4-AutoNav]

4.1.1.Objectif

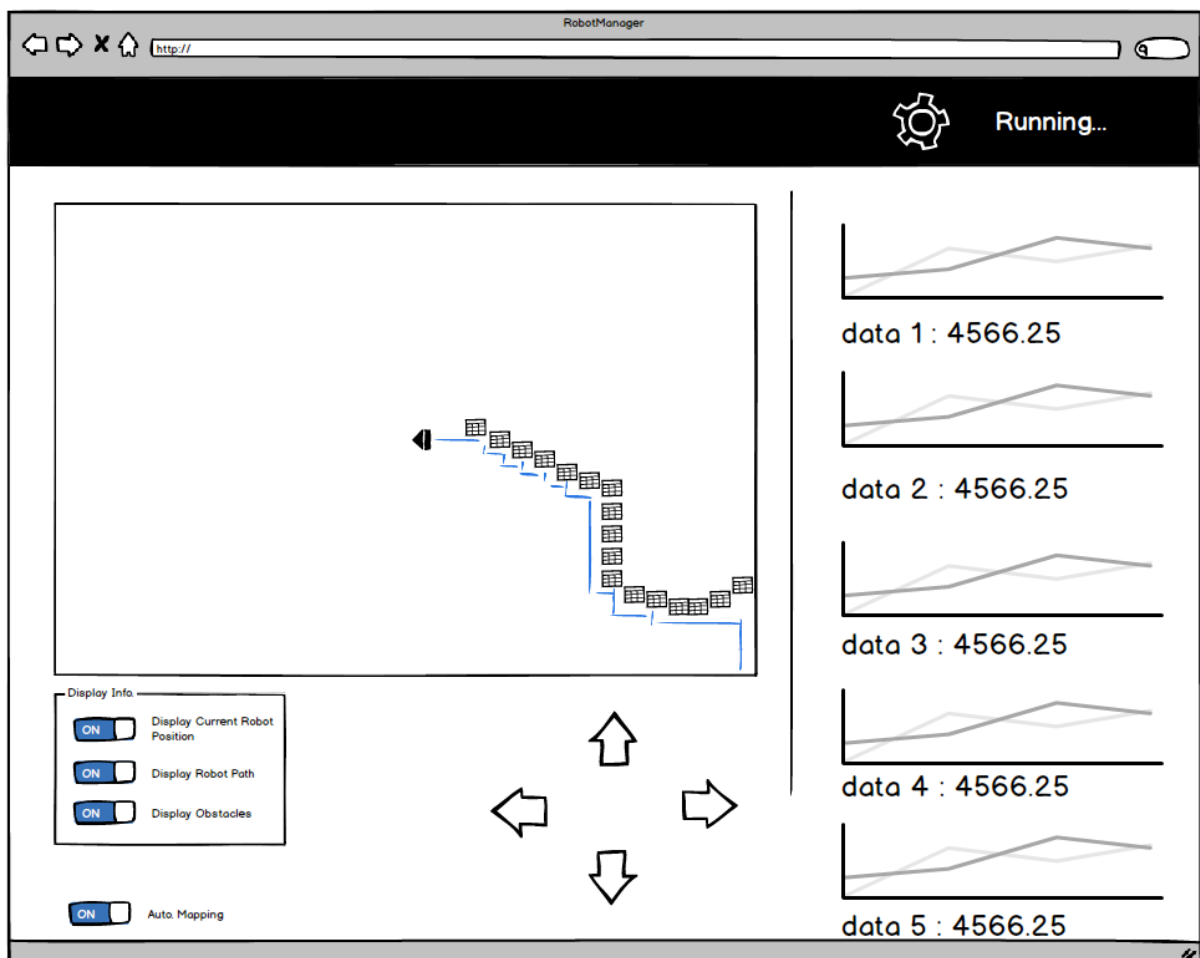
Le lot suivant devra remplir les fonctionnalités suivantes :

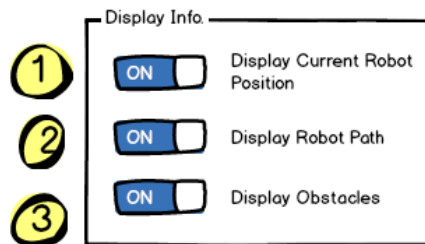
- **[FCT-INFO4-AutoNav-customdisplay]** Permettre à l'utilisateur d'afficher ou non les éléments du canvas
- **[FCT-INFO4-AutoNav-autoLauncher]** Permettre à l'utilisateur de lancer le robot en mode automatique simple
- **[FCT-INFO4-AutoNav-slam]** Permettre à l'utilisateur de lancer le robot en mode automatique évolué

4.1.2.Moyen

- Utilisation des canvas afin d'afficher la position depuis la télécommande (attention à l'échelle).
- Enregistre vos objets avec leur positions dans le canvas(obstacle et trajectoire)

4.1.3.Informations complémentaires





(1)
Push or switch button
Display or hide Current robot position on the Canvas

(2)
Push or switch button
Display or hide Current robot path on the Canvas

(3)
Push or switch button
Display or hide obstacles on the Canvas



(4)
Push or switch button
Enable or disable the environment mapping. When environment mapping is enabled, the robot automatically navigates in the room and detects and avoids obstacles.
The target of such navigation is to detect all obstacles in a room

La fonctionnalité **[FCT-INFO4-AutoNav-autoLauncher]** va permettre de démarrer le robot qui sera piloté automatiquement par le serveur. Ce dernier fera avancer le robot jusqu'à ce que ce dernier rencontre un obstacle. Une fois l'obstacle rencontré le robot effectuera une rotation et ira à nouveau tout droit.

La fonctionnalité **[FCT- INFO4-AutoNav-slam]** va permettre de démarrer le robot qui sera piloté automatiquement par le serveur. Dans ce mode le serveur devra optimiser la trajectoire du robot afin qu'il détecte **TOUS** les obstacles possibles dans une zone.

5. Annexes techniques

5.1.Utilisation de JAVA (Servlet) comme fournisseur d'information (JAX-RS)

```
@Path("/cmd")
public class OtherSample {

    @POST
    @Produces(MediaType.TEXT_PLAIN)
    @Path("/UP")
    public String goUp() {
        return "Ask to go Up";
    }

    @POST
    @Produces(MediaType.TEXT_PLAIN)
    @Path("/LEFT")
    public String goUp() {
        return "Ask to go Left";
    }

    @GET
    @Produces(MediaType.APPLICATION_JSON)
    @Path("env")
    public String getData(@PathParam("eventId") Long eventId) {
        return "'envMatrix':" + robotCrt.getEnv() + " ";
    }

    @GET
    @Produces(MediaType.APPLICATION_JSON)
    @Path("envRobot")
    public String getData(@PathParam("eventId") Long eventId) {
        return "'envMatrix':" + robotCrt.getEnvRobot() + " ";
    }
}
```

L'exemple ci-dessus permet d'envoyer des informations à l'url 127.0.0.1 :8080/<your projet name>/rest/cmd/<command>

5.2.Afficher une matrice de string en java (text)

```
public static String printMatrix(String[][] matrix,int sizeX,int sizeY) {
    StringBuilder sb = new StringBuilder();
    Formatter formatter = new Formatter(sb, Locale.FRENCH);

    String formatS = "%1$5s";
    String[] valueTab = new String[sizeX+1];
    valueTab[0]="";

    for (int index = 0; index < sizeX; index++) {
        formatS = formatS + " %" + (index + 2) + "$5s";
        valueTab[index+1] = String.valueOf(index);
    }
    formatter.format(formatS + "\n", valueTab);
    formatter.format("%1$5s | %2$47s\n", "",
```

```

        " _____ " );

    for (int i = 0; i < sizeY; i++) {
        String formatS2 = "%1$5s | ";
        String[] valueTab2 = new String[sizeY+1];
        valueTab2[0]=String.valueOf(i);
        for (int j = 0; j < sizeX; j++) {
            formatS2 = formatS2 + " %" + (j + 2) + "$5s";
            valueTab2[j+1] = matrix[i][j];
        }
        formatter.format(formatS2 + "\n", valueTab2);
    }
    return formatter.toString() ;
}

```

5.3.Utilisation des Canvas HTML5

5.3.1.Références

http://www.w3schools.com/tags/ref_canvas.asp

<http://devhammer.net/blog/exploring-html5-canvas-part-6-managing-animated-shapes/>

<http://www.creativeblog.com/html5/build-tile-based-html5-game-31410992>

<http://www.html5canvastutorials.com/tutorials/html5-canvas-lines/>

<http://www.html5canvastutorials.com/tutorials/html5-canvas-images/>

http://www.w3schools.com/tags/canvas_lineto.asp

5.4.Utilisation de JSON MAP

5.4.1.Références

<https://github.com/bjorn/tiled/wiki/JSON-Map-Format>

<http://www.creativeblog.com/html5/build-tile-based-html5-game-31410992>

<https://hashrocket.com/blog/posts/using-tiled-and-canvas-to-render-game-screens>

<https://gist.github.com/shaneriley/4078905>

<https://openclassrooms.com/courses/creer-un-mini-rpg-en-javascript-avec-canvas/mise-en-place-du-terrain>

<https://github.com/rbcasperson/canvas-tile-map>

5.5.Affichage des graphes

5.5.1.Références

<http://www.highcharts.com/demo>

<http://jsfiddle.net/gh/get/jquery/1.9.1/highslide->

<software/highcharts.com/tree/master/samples/highcharts/demo/area-basic/>

<http://www.highcharts.com/demo/dynamic-update>

