

28/07/2021

BETAILLE

Antonin

Compilation des comptes-rendus
hebdomadaire
Projet : Bras manipulateur

SECTION : Interface web

Amélioration de l'interface web

| | |
|--|-----------|
| AMELIORATION ESTHETIQUE | 3 |
| PREMIERS CHANGEMENTS | 3 |
| LIGHT MODE ET DARK MODE : | 6 |
| AMELIORATION DE L'EXPERIENCE UTILISATEUR..... | 7 |
| AJOUT D'UNE DESCRIPTION ORALE A L'ACTIVATION D'UN BOUTON : | 7 |
| LISTE DE LIENS UTILES : | 8 |
| AJOUT D'ELEMENTS PRATIQUES | 8 |
| NECESSITE D'UN BOUTON D'ARRET D'URGENCE | 8 |
| NECESSITE DE SEPARER LES DEUX FONCTIONNALITES DE L'ARRET D'URGENCE | 9 |
| CREATION D'UN NOUVEAU MENU POUR LE BRAS..... | 10 |
| VERSION FINALE..... | 11 |
| LA PAGE D'ACCUEIL | 11 |
| LE MENU DU MEUBLE | 13 |
| LE MENU DE COMMANDE PAR JOYSTICK | 14 |
| LE MENU DU BRAS | 15 |

Amélioration esthétique

Premiers changements

L'interface web proposée par Killian était fonctionnelle est assez lisible. Cependant, de mon point de vue, l'esthétique globale pouvait être améliorée. C'est pourquoi je me suis penché sur les fonctionnalité du bootstrap pour modifier l'apparence des boutons (leur nombre avec un menu déroulant) et la taille des zones de d'information. J'ai également pris soin de changer l'apparence de la barre de batterie (qui est maintenant animée). Enfin j'ai ajouté un petit bandeau en haut de l'interface contenant le logo de la chaire (via Gimp 2).

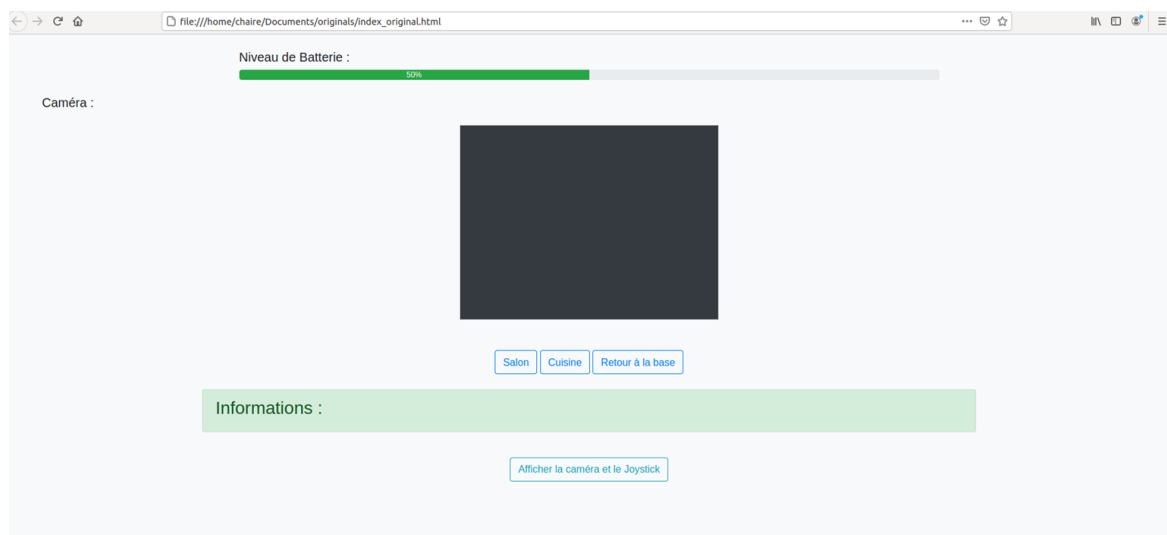


Illustration 1: Interface web originale p1

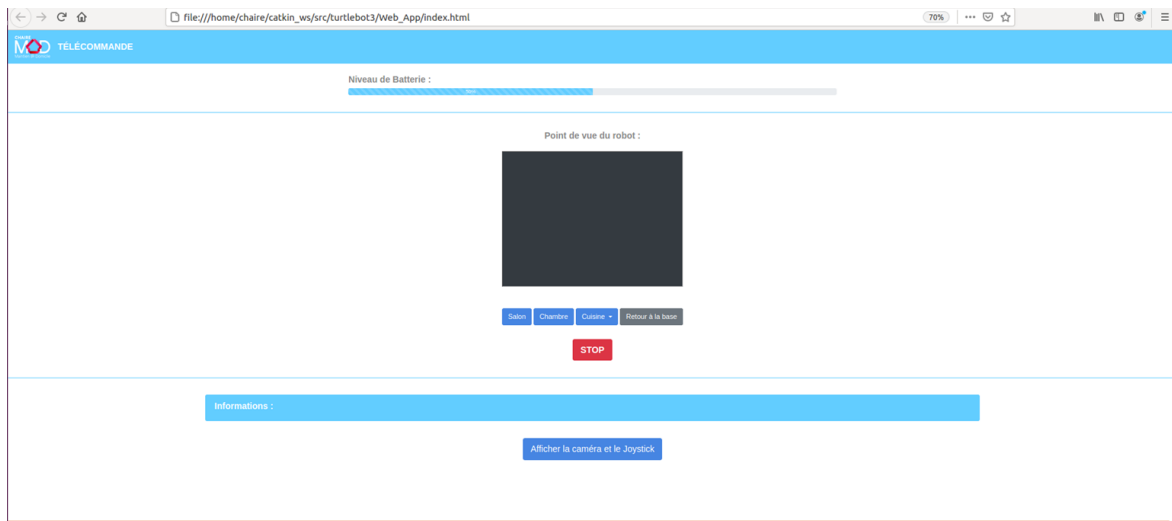


Illustration 2: Nouvelle interface web p1 (light mode)

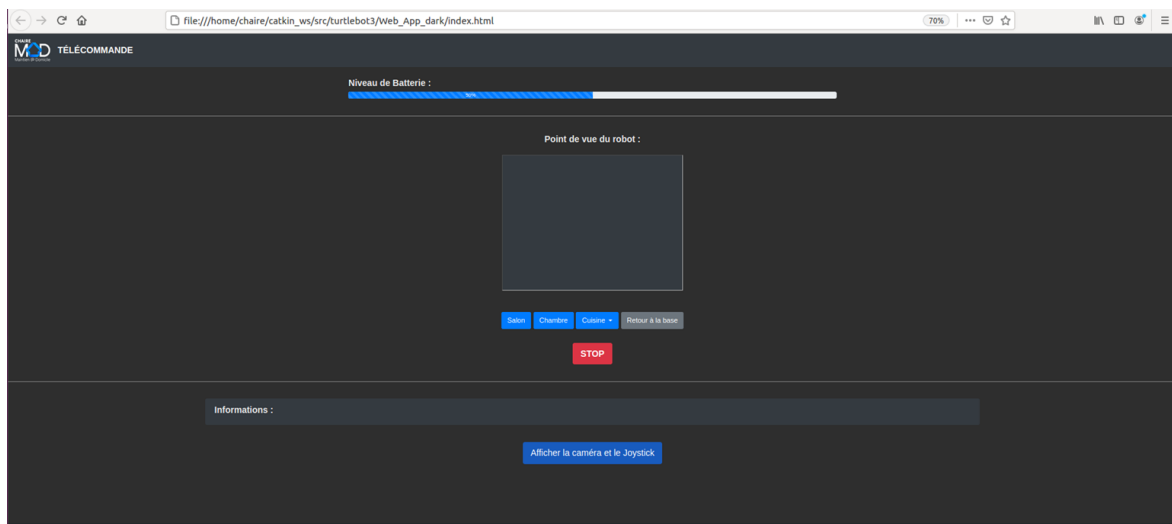


Illustration 3: Nouvelle interface web p1 (dark mode)

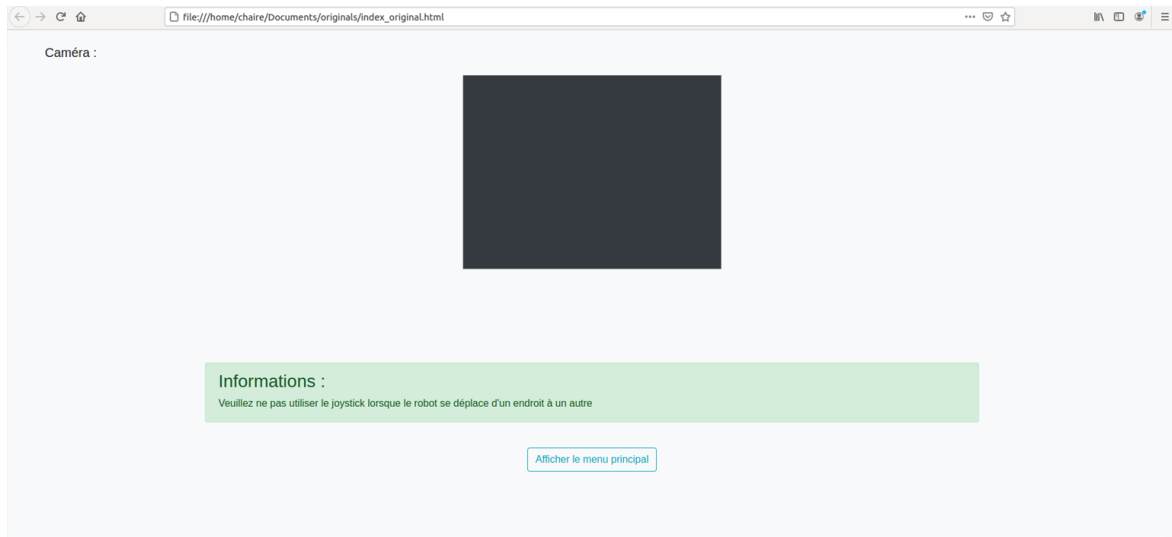


Illustration 4: Interface web originale p2

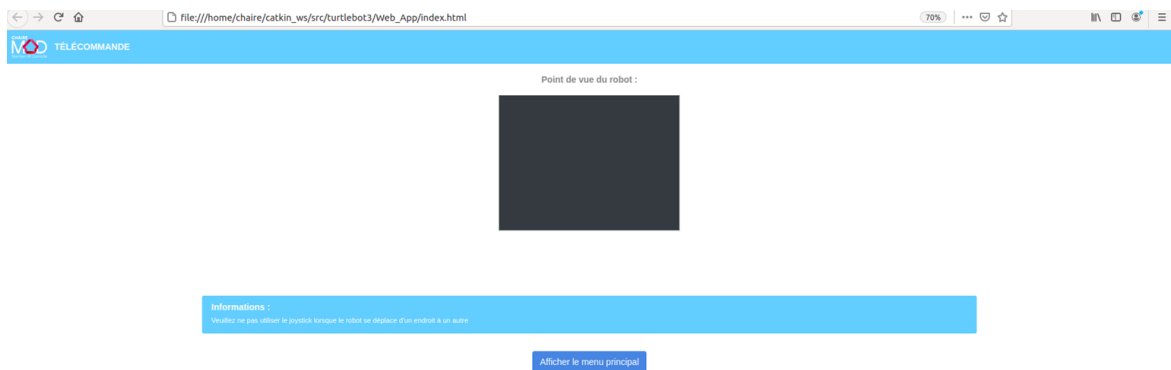


Illustration 5: Nouvelle interface web p2 (light mode)

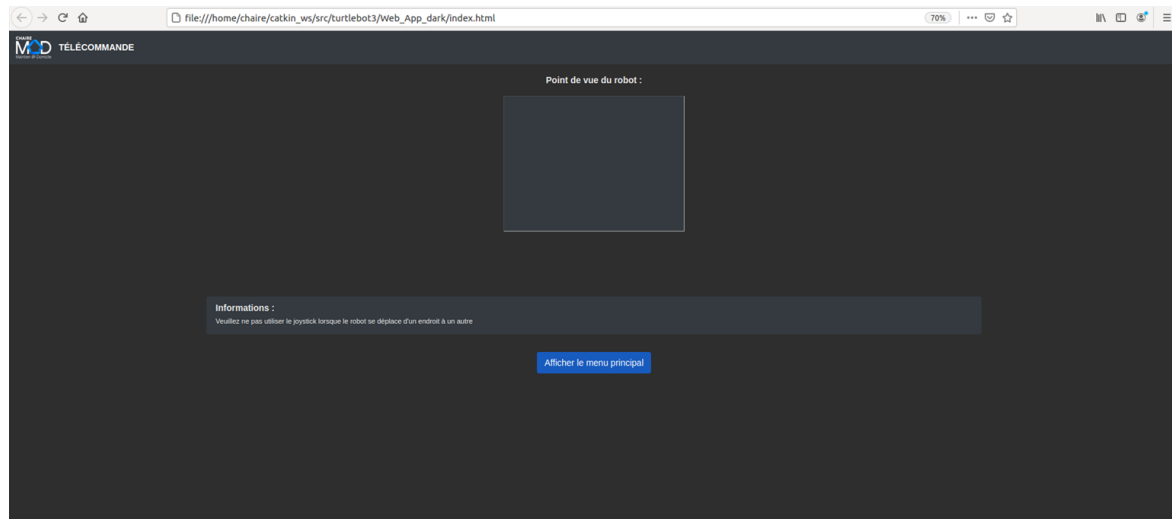


Illustration 6: Nouvelle interface web p2 (dark mode)

Light mode et dark mode :

Comme vous pouvez le constater j'ai pris soin de créer deux versions de l'interface web. De cette façon j'ai à disposition une version adaptée à la lecture de jour et une version adaptée à la lecture en soirée.

Les deux versions sont disponibles au lancement du fichier turtle.sh. L'affichage de l'une ou l'autre dépend uniquement de l'heure à laquelle le fichier est exécuté. Pour cela j'ai modifié le script en ajoutant les lignes suivantes :

```
printf -v heure '%(%H)T' -1
```

→ la variable heure prend la valeur de l'heure actuelle (0-23)

```
echo "Il est actuellement plus de : $heure h"
```

```
if [[ $heure -gt 14 ]] → s'il est au moins 15h (valeur de test)
```

```
then
```

```
    gnome-terminal --working-
```

```
    directory=/home/chaire/catkin_ws/src/turtlebot3/Web_App_dark/ -x live-server . --
```

```
    port=5500 --no-browser & # --no-browser
```

→ Lancement de l'interface web en dark mode

```
else
```

```
    gnome-terminal --working-directory=/home/chaire/catkin_ws/src/turtlebot3/Web_App/ -
```

```
    x live-server . --port=5500 --no-browser & # --no-browser
```

→ Lancement de l'interface web en light mode

```
fi → Fin de la structure (if... then... else)
```

Il est important de noter que je n'ai pas pris en compte les premières heure du matin entre 0 et 7h. Ce cas de figure est à compléter

Amélioration de l'expérience utilisateur

Ajout d'une description orale à l'activation d'un bouton :

Killian avait suggéré l'idée d'ajouter un haut parleur sur le turtlebot afin de créer une meilleure interaction avec son utilisateur. Faute de haut parleur, j'ai décidé de donner une voix au robot directement depuis l'interface web. Chaque bouton a donc un fichier audio qui est activé lors d'un clic.

Pour créer la liste des fichiers audio j'ai d'abord installé le package **libtts-pico-utils** :

```
sudo apt-get install libtts-pico-utils
```

Pour enregistrer une phrase en français il faut respecter le schéma suivant :

```
pico2wave -l fr-FR -w chambre.wav "Je vais dans la chambre"
```

La phrase entre guillemets sera prononcée par une voix de synthèse similaire à celle de google traduction.

Une fois le fichier créé il faut programmer son utilisation en l'utilisant dans une fonction du fichier **webui.js**

```
function Chambre() {  
  document.getElementById("chambre-btn");  
  var music = new Audio('https://raw.githubusercontent.com/Anton1B/Base_mobile_MaD/main/chambre.wav');  
  music.play();  
  my_message.data=2;  
  talker.publish(my_message);  
  console.log(my_message.data);  
}
```

Remarque :

Les fichiers utilisés pour l'affichage des différentes versions du logo et les fichiers audio ont été déposés sur le même dépôt github. Ce dépôt servira à sauvegarder tous les fichiers liés au stage à l'avenir :

https://github.com/Anton1B/Base_mobile_MaD

Liste de liens utiles :

<https://www.computerhope.com/issues/ch000059.htm>

→ « How to link an image to another page in HTML »

<https://getbootstrap.com/docs/4.0/components/buttons/>

→ Aide bootstrap pour les boutons. D'autres catégories sont disponible dans le menu de gauche

<https://www.wikihow.com/Change-the-Button-Color-in-HTML>

→ Autres informations concernant les boutons

<https://stackoverflow.com/questions/59722914/how-to-change-color-of-header-in-html>

→ Modifier la couleur des titres

Ajout d'éléments pratiques

Nécessité d'un bouton d'arrêt d'urgence

Bien que la plupart des fonctionnalités de l'interface web aient été mises en place. Un cas de figure concernant le docking m'a forcé à reconsidérer la capacité de l'utilisateur à contrôler le robot en cas d'urgence. En effet, même si j'ai mis un bouton stop en place, ce bouton n'est actif que sur le node `/simple_navigation_goals`. Il n'a donc pas d'effet sur le node `/automatic_parking_vision`, ou quelconque autre node agissant sur le topic `/cmd_vel`. Donc dans le cas où certains nodes mèneraient à des bugs où le robot reste bloqué sur une commande, il me fallait avoir accès à un bouton d'urgence.

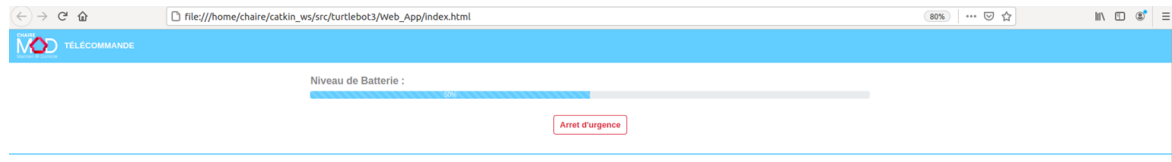


Illustration 7: Bouton d'arrêt d'urgence

Suite de la description audio sur l'arrêt d'urgence :

Pas de réelle avancée à ce niveau. J'ai simplement pris soin de bien choisir la phrase à convertir en synthèse vocale pour l'arrêt d'urgence. En effet il est important de rappeler pourquoi le robot s'est arrêté et ce qu'il faut faire une fois les nodes réactivés. J'ai précisé qu'il fallait attendre 30s puis recharger la page web. Une fois le bouton d'activation sera disponible je changerai sûrement de phrase pour faire

Nécessité de séparer les deux fonctionnalités de l'arrêt d'urgence

Plus haut, j'expliquais avoir mis un bouton d'arrêt d'urgence me permettant de tuer les nodes agissant sur le topic `/cmd_vel` puis de les réactiver au bout de quelques instants. Ce dispositif avait pour avantage d'être rapide mais assez contre intuitif concernant le but premier de l'arrêt d'urgence. Le docking par exemple se contentait de reprendre depuis le début et faisait tourner le robot dès sa réactivation. Alors certes, le node `/emergency_stop` pouvait être utilisé à la fois depuis l'interface web et depuis un terminal mais dans le cadre d'une utilisation normale, il était essentiel de pouvoir activer le node `/emergency_activation` à n'importe quel instant, en fonction de la volonté de l'utilisateur.

C'est pourquoi j'ai créé un bouton [Réactivation] assurant le rôle de remise en fonctionnement du robot.

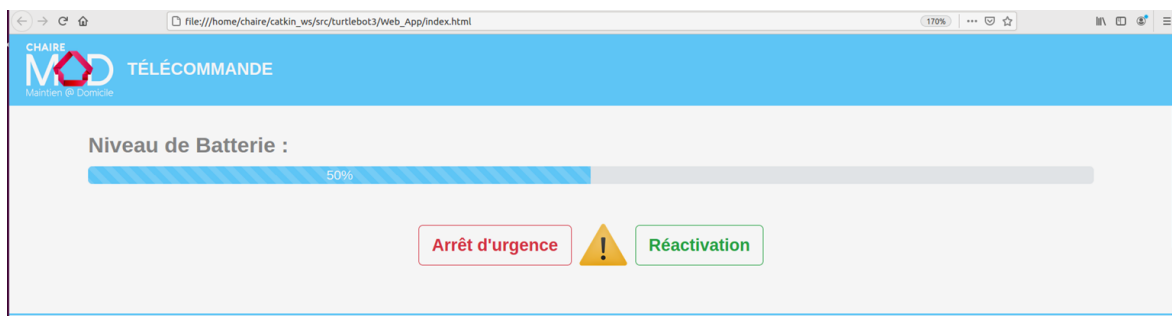


Illustration 8: Nouvelle configuration de l'arrêt d'urgence

Remarque :

Les fichiers audio et les messages ont été mis à jour. Leur contenu explique la marche à suivre une fois qu'un des deux boutons d'arrêt d'urgence a été activé

Création d'un nouveau menu pour le bras

Puisque le bras doit être intégré dans le scénario de fonctionnement de l'ensemble du système il est légitime de devoir accompagner l'interface web d'un menu pour le Niryo. En effet au-delà de pouvoir effectuer une action après le docking, il est important de garder un contrôle sur le robot à chaque mouvement. C'est pourquoi j'ai voulu mettre en place un système de pause et un système d'arrêt en plus du choix de l'action à effectuer.

J'ai donc créé plusieurs boutons ayant pour but d'activer les fonctions qui leur sont attribuées. Ces boutons sont positionnés sous le retour de la vidéo de la caméra du niryo. Pour y accéder il y a un bouton [Afficher le menu du bras] disposé sous le bouton STOP du menu du turtlebot.

Pour faciliter la compréhension de l'interface lors du changement de menu j'ai adopté un code couleur légèrement différent pour ce qui concerne le bras

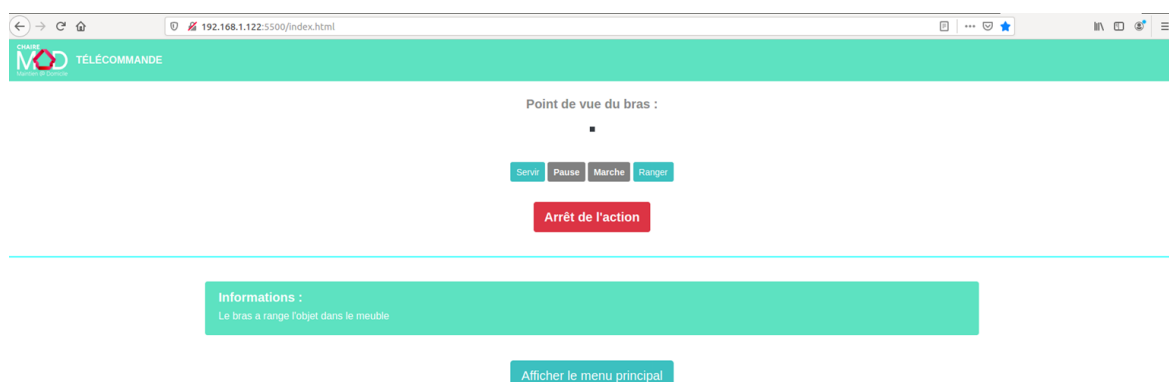
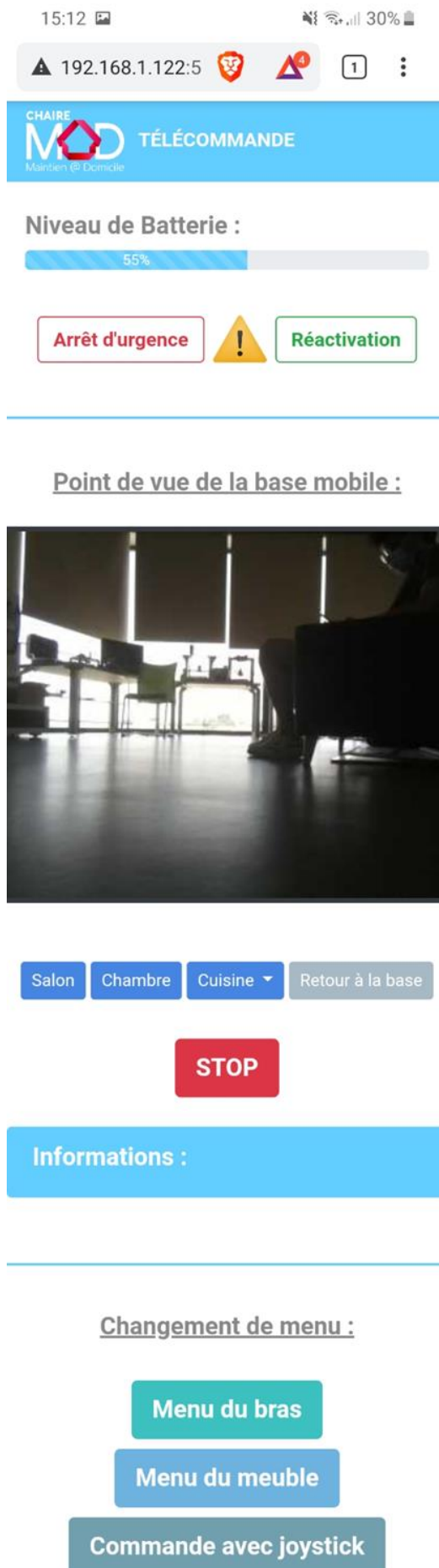


Illustration 9: Menu du bras dans l'interface web

Version finale

La page d'accueil

La page d'accueil a été légèrement revue de manière à intégrer de façon plus claire le changement de menu



Le menu du meuble

En prévision de l'intégration du meuble intelligent au fonctionnement général du système, j'ai décidé de travailler sur un menu dédié à son contrôle. J'ai donc pris en compte chaque élément prévu dans la réalisation du meuble pour créer les boutons nécessaires. Il y a donc pour l'instant 3 boutons principaux liés aux 3 mécanismes mis en place par Quentin Mérian :

- l'élévateur, avec dans son menu déroulant les fonctions « monter » et « descendre »
- le convoyeur avec les fonctions « avancer » et « reculer »
- le tiroir avec les fonctions « sortir » et « rentrer »

En plus de ces 3 boutons sont toujours présents les boutons [Pause] et [Marche] afin d'interrompre et de reprendre les actions en cours.

Et enfin le bouton [Arrêt de l'action], qui tout comme pour le Niryo, est destiné à arrêter l'action en cours définitivement pour ensuite faire revenir le meuble dans son état initial.



Illustration 11: Screenshot du menu du meuble intelligent de l'interface web pris depuis un smartphone

Il est important de noter l'absence de caméra pour ce menu. En effet, il n'y a pas de caméra embarquée sur le meuble. Cependant avec l'ajout d'une caméra, il faudrait penser à intégrer une zone d'affichage pour la nouvelle source de flux vidéo.

Le menu de commande par joystick

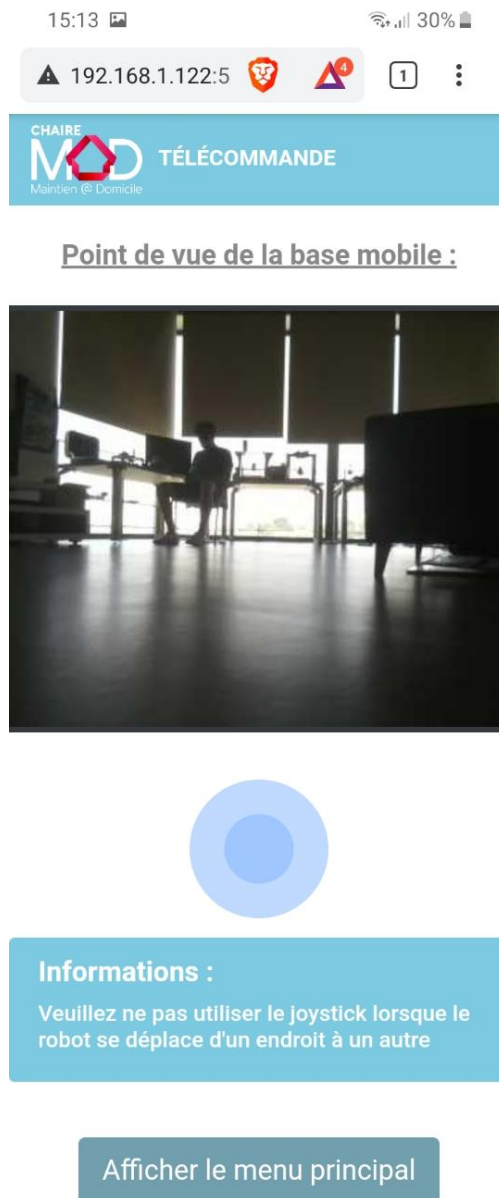


Illustration 12: Screenshot du menu de contrôle par joystick pris depuis un smartphone

Le menu du bras

L'ajout d'un mode d'observation a entraîné la division du menu du bras en deux parties. La première est destinée au contrôle du bras dans le cadre du scénario d'utilisation basique. La seconde est dédiée à la fonctionnalité d'observation mise en place avant la fin du stage.

En cliquant sur le bouton [Démarrer l'observation], le robot se positionne à mi-hauteur avec un angle d'azimut nul. Puis les flèches qui sont situées en dessous permettent d'agir sur l'angle d'azimut du bras et sur l'angle de tangage de la caméra pour être en mesure de balayer la pièce. Le bouton central est un peu différent puisqu'il permet de revenir à la position d'origine en début d'observation.



Illustration 13: Screenshot du menu du bras pris par un smartphone