



**POLYTECHNIQUE  
MONTRÉAL**

UNIVERSITÉ  
D'INGÉNIERIE



Département de génie informatique et génie logiciel

**INF3995**

**Projet de conception d'un système informatique**

Document des Tests

Équipe No **104**

<i>Aymane Chalh</i>	A.C
<i>Anass El Kettani</i>	A E K
<i>Abderrahim Zebiri</i>	A.Z
<i>Thibault Demagny</i>	
<i>Nada Alami Chentoufi</i>	
<i>Lina Khial</i>	L.K

03 Novembre 2023

**Ce guide vous explique comment accéder à l'interface utilisateur.**

**Étape 1** : Cloner le projet

```
user$ git clone  
https://gitlab.com/polytechnique-montr-al/inf3995/20233/equipe-104/INF3995-104.git
```

**Étape 2**: démarrer le site

```
INF3995-104/webApp/front-end$ npm start
```

**Étape 3**: démarrer le serveur

```
INF3995-104/webApp/back-end$ python3 run.py
```

**Ce guide vous explique comment utiliser la simulation Limo Gazebo.**

**Étape 1** : sudo docker compose up

## **Plan de Test pour R.F.4: Exploration Autonome des Robots (simulation)**

*Objectif du Test* : S'assurer que les robots explorent l'environnement de façon autonome après le démarrage.

### **Étapes du Test :**

- a. Démarrez une mission en utilisant l'application web dans la section simulation, en testant les fonctionnalités disponibles pour CDR.
- b. Les robots explorent l'environnement de manière autonome dans un environnement simulé avec des obstacles statiques et dynamiques (l'autre robot) positionnés de manière aléatoire.
- c. Surveillez attentivement les mouvements des robots et leur interaction avec l'environnement.

**Résultats Attendus** : Les robots explorent l'environnement de manière autonome, en effectuant des mouvements aléatoires.

## **Plan de Test pour R.F.5: Évitement d'Obstacles (simulation)**

*Objectif du Test* : S'assurer que les robots évitent les obstacles détectés par leurs capteurs, qu'ils soient statiques ou d'autres robots.

### **Prérequis :**

Des obstacles de différentes natures (statiques et dynamiques) sont présents dans l'environnement de simulation.

### **Étapes du Test :**

- a. Démarrez une mission en utilisant l'application web dans la section simulation.
- b. Placez des obstacles sur le chemin des robots, comprenant des obstacles statiques positionnés de manière aléatoire. tels que des murs et des boîtes, ainsi que des obstacles dynamiques comme des robots mobiles.
- c. Surveillez attentivement les réactions des robots lorsqu'ils détectent des obstacles à l'aide de leurs capteurs.
- d. Assurez-vous qu'ils évitent avec succès ces obstacles en ajustant leur trajectoire et qu'ils poursuivent leur mission.

### **Résultats Attendus :**

Les robots doivent démontrer leur capacité à détecter et éviter avec succès les obstacles positionnés de manière aléatoire et de différentes natures (statiques et dynamiques) présents dans l'environnement. Cela mettra en évidence leur efficacité dans l'évitement d'obstacles, garantissant ainsi une navigation autonome.

**Ce guide vous explique comment démarrer un robot.**

**Étape 1** : Compiler la Simulation

~\$ ./start\_script.sh

### **Plan de Test pour R.F.1: Identification Individuelle des Robots**

*Objectif du Test* : S'assurer que chaque robot physique peut répondre à la commande "Identifier" disponible dans l'interface utilisateur.

#### **Étapes du Test :**

- a. Accédez à l'interface utilisateur.
- b. Vérifiez que les robots physiques sont allumés et prêts.
- c. Cliquez sur le bouton "Identifier" dans l'interface utilisateur.
- d. Observez la réponse des robots physiques (émission d'un son ou dessin sur l'écran).

#### **Résultats Attendus :**

Chaque robot physique répond à la commande "Identifier" par une action unique (émission d'un son ou dessin sur l'écran).

### **Plan de Test pour R.F.2: Commandes de Contrôle de Mission**

*Objectif du Test* : S'assurer que l'équipe de robots répond aux commandes "Démarrer la Mission" et "Terminer la Mission" de l'interface utilisateur.

### **Étapes du Test :**

- a. Cliquez sur le bouton "Démarrer la Mission".
- b. Observez les robots physiques alors qu'ils commencent la mission ainsi que la map qui se dessine sur l'interface utilisateur.
- c. Cliquez sur le bouton "Terminer la Mission" pour arrêter la mission.

### **Résultats Attendus :**

Les robots physiques commencent leur mission lorsqu'on clique sur "Démarrer la Mission".

Les robots physiques arrêtent leur mission immédiatement lorsqu'on clique sur "Terminer la Mission".

### **Plan de Test pour R.F.3: Affichage de l'État des Robots**

*Objectif du Test* : S'assurer que l'interface utilisateur affiche l'état de chaque robot, mis à jour avec une fréquence minimale de 1 Hz.

### **Étapes du Test :**

- a. Lors d'une mission, surveillez continuellement l'interface utilisateur.
- b. Assurez-vous que l'état de chaque robot est affiché et se met à jour au moins une fois par seconde.

### **Résultats Attendus :**

L'interface utilisateur affiche l'état de chaque robot, se mettant à jour au moins une fois par seconde.

## **Plan de Test pour R.F.4: Exploration Autonome des Robots**

*Objectif du Test* : S'assurer que les robots explorent l'environnement de façon autonome après le démarrage.

### **Étapes du Test :**

- a. Démarrez une mission.
- b. Laissez les robots explorer l'environnement de manière autonome.
- c. Surveillez les mouvements des robots et leur interaction avec l'environnement.

### **Résultats Attendus :**

Les robots explorent l'environnement de manière autonome, en effectuant des mouvements aléatoires.

## **Plan de Test pour R.F.5: Évitement d'Obstacles**

*Objectif du Test* : S'assurer que les robots évitent les obstacles détectés par leurs capteurs, qu'ils soient statiques ou d'autres robots.

### **Prérequis :**

Des obstacles sont présents dans l'environnement.

### **Étapes du Test :**

- a. Démarrez une mission.
- b. Placez des obstacles sur le chemin des robots.
- c. Surveillez les réactions des robots lorsqu'ils détectent des obstacles.
- d. Assurez-vous qu'ils évitent ces obstacles et poursuivent leur mission.

### **Résultats Attendus :**

Les robots détectent et évitent avec succès les obstacles, qu'ils soient statiques ou d'autres robots.

### **Plan de Test pour R.F.6: Commandes de Contrôle de Mission**

*Objectif du Test* : S'assurer que l'équipe de robots répond aux commandes "Retourner à la base".

### **Étapes du Test :**

- a. Cliquez sur le bouton "Démarrer la Mission".
- b. Observez les robots physiques alors qu'ils commencent la mission ainsi que la map qui se dessine sur l'interface utilisateur.
- c. Attendez que le robot s'éloigne un peu de la base.
- d. Cliquez sur le bouton "Retourner à la base" pour arrêter la mission.

### **Résultats Attendus :**

Les robots physiques reviennent à leur base en affichant les logs correspondants.

### **Plan de Test pour R.F.6: Commandes de Contrôle de Mission**

*Objectif du Test* : S'assurer que l'équipe de robots répond aux commandes "Retourner à la base" automatiquement dès que le niveau de batterie devient de moins de 30%.

### **Étapes du Test :**

- a. Assurez vous que la batterie est proche de 30%



- b. Cliquez sur le bouton "Démarrer la Mission".
- c. Observez les robots physiques alors qu'ils commencent la mission ainsi que la map qui se dessine sur l'interface utilisateur.
- d. Attendez que le robot s'éloigne un peu de la base et le pourcentage de la batterie diminue.
- e. Une fois que le niveau de la batterie devient de moins de 30%, le robot effectue immédiatement un retour à la base.

### **Résultats Attendus :**

Les robots physiques reviennent à leur base en affichant les logs correspondants.

### **Plan de Test pour R.F.10: Interface Utilisateur en tant que Service Web**

*Objectif du Test* : S'assurer que l'interface utilisateur est disponible en tant que service Web et peut être visualisée sur différents appareils (PC, tablette, téléphone) via le réseau.

### **Étapes du Test :**

- a. Accédez à l'interface utilisateur depuis un PC, une tablette et un téléphone via le réseau.
- b. Vérifiez que l'interface utilisateur s'affiche correctement sur chaque appareil.

### **Résultats Attendus :**

L'interface utilisateur est accessible depuis plusieurs types d'appareils et s'affiche correctement.

Dans le cadre de notre engagement continu à répondre à ce requis, nous avons récemment testé notre interface utilisateur

pour s'assurer qu'elle est pleinement accessible en tant que service Web. Notre objectif est de garantir une expérience utilisateur cohérente et optimisée, indépendamment de l'appareil utilisé pour accéder au service.

En effet, la vérification a porté sur les points suivants :

- Disponibilité du service Web sur diverses plateformes (PC, tablette, téléphone).
- Compatibilité avec les principaux navigateurs web.
- Préservation de la fonctionnalité et de l'esthétique de l'interface utilisateur sur différents formats d'écran.
- Performance et temps de chargement adaptés aux capacités de connexion variables.

Pour des raisons de validation, nous avons enregistré une vidéo qui est stockée dans notre dépôt GitLab pour attester du bon fonctionnement de notre interface.

En ce qui concerne les boutons de notre interface, nous avons méticuleusement élaboré une série de tests unitaires. Ces derniers jouent un rôle crucial en s'assurant que l'interaction entre le frontend et le backend est impeccable. Plus précisément, ils vérifient que le message transmis du frontend au backend est correct, et que la réponse retournée par le backend correspond précisément à ce qui est attendu. Cette démarche garantit une intégrité et une fiabilité de communication, éléments essentiels pour une bonne expérience utilisateur.

### **Plan de Test pour R.F.14 et R.F.16: Mise à jour et éditeur de code**

*Objectif du Test* : Vérifier le bon fonctionnement de la mise à jour de fichiers des robots physiques, en utilisant l'éditeur de code pour effectuer le test.

**Étapes du Test :**

- a. Connecter les deux robots à la station au sol pour avoir accès au bouton “mettre à jour les robots”
- b. entrer le mot de passe de connexion SSH (par défaut ‘nvidia’) dans le dialogue prompt.
- c. attendre la réception des fichiers
- d. Modifier le fichier `save_test_file.py` en remplaçant la valeur de `range(x)` par une valeur différente pour voir une mise à jour.
- e. Sauvegarder et observer le lancement automatique du code de `save_test_file.py` (effectuant `x` demi-tour)

### **Résultats Attendus :**

Le mot de passe laisse bien l'accès à la page de l'éditeur de code. La récupération des fichiers s'effectue dans un délai acceptable, la modification d'un des fichiers prévient l'utilisateur de quitter sans sauvegarder. La sauvegarde ne s'effectue seulement si les robots ne sont pas en mission. Le code de test se lance une fois les fichiers sauvegardés sur les robots. Les fichiers sont sauvegardés lorsque le message *Saving files* disparaît.

### **Plan de Test pour R.C.1: Logs de Débogage**

*Objectif du Test* : S'assurer que des logs de débogage sont disponibles en continu lors de chaque mission, et qu'ils contiennent les informations nécessaires au développement, notamment les lectures des capteurs de chaque robot et les commandes envoyées par la station au sol.

### **Étapes du Test :**

- a. Démarrez une mission.
- b. Surveillez l'interface utilisateur pour afficher les logs de débogage en temps réel pendant la mission.

Vérifiez que les logs contiennent les informations suivantes :

- a. Les lectures des capteurs de chaque robot (y compris la distance et la position) avec une fréquence d'au moins 1 Hz.
- b. Les commandes envoyées par la station au sol.

### **Résultats Attendus :**

Les logs de débogage sont activés pour la mission en cours.  
Les logs de débogage s'affichent en temps réel sur l'interface utilisateur pendant la mission.

Les logs contiennent des informations détaillées, notamment les lectures des capteurs de chaque robot avec une fréquence minimale de 1 Hz et les commandes envoyées par la station au sol.

Les logs sont formatés de manière lisible et sont accessibles à des fins de diagnostic.

Les logs permettent de suivre et de comprendre le déroulement de la mission.