

# WIFIBOT

## Activité n°4 : Réalisation de l'application WIFIBOT

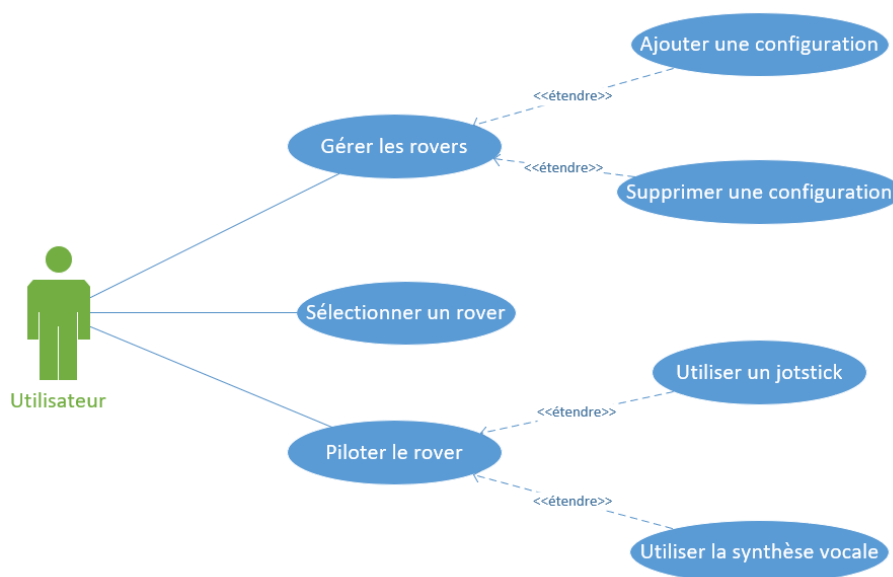
### 1. Objectifs du TP

Concevoir une application graphique en langage C# permettant à un utilisateur de disposer des fonctionnalités définies par le diagramme des cas d'utilisation présenté au paragraphe suivant.

Le plan de développement à suivre sera le suivant :

- 1- Réaliser la modélisation de l'application en langage UML.
- 2- Réaliser une maquette de l'IHM qui devra permettre au client de valider l'ergonomie et de vérifier que toutes les fonctionnalités attendues sont bien prises en compte pour le développement.
- 3- Développer le logiciel de l'application en langage C#.
- 4- Tester les fonctionnalités et démontrer qu'elles sont bien réalisées.
- 5- Réaliser un compte rendu contenant les résultats des tests de chaque fonctionnalité
- 6- Rédiger le cahier de recette
- 7- Effectuer la recette en présence du client

### 2. Fonctionnalités attendues (DCU)



## Description des cas d'utilisation

### Cas d'utilisation « Gérer les rovers »

**Objectif** : Permettre à l'utilisateur de gérer la flotte de rovers.

**Acteurs concernés** : L'utilisateur de l'application

**Pré conditions** : Aucune

**Scénario nominal** :

1. L'utilisateur sélectionne cette fonctionnalité dans le menu de l'IHM
2. A partir du sous menu affiché il peut choisir :
  - 2.1. D'ajouter une configuration
  - 2.2. De supprimer une configuration

**Scénario alternatif** : Aucun

### Cas d'utilisation « Ajouter une configuration »

**Objectif** : Permettre à l'utilisateur d'ajouter un rover dans la liste des rovers existants.

**Acteurs concernés** : L'utilisateur de l'application

**Pré conditions** : Aucune

**Scénario nominal** :

1. L'utilisateur sélectionne cette fonctionnalité depuis le cas « gérer les rovers »
2. Une IHM secondaire lui permet de saisir les informations suivantes :
  - Nom du rover
  - Adresse IP du serveur embarqué du rover (format IP v4)
  - Numéro du port TCP sur lequel le serveur écoute (de 15015 à 15025)
3. Après avoir validé la configuration, l'IHM secondaire est fermée et la configuration est sauvegardée dans le fichier

Remarque : l'utilisateur peut annuler sa configuration à tout moment.

**Scénario alternatif** : Aucun

## Cas d'utilisation « Supprimer une configuration »

**Objectif :** Permettre à l'utilisateur de supprimer un rover dans la liste des rovers existants.

**Acteurs concernés :** L'utilisateur de l'application

**Pré conditions :** Aucune

**Scénario nominal :**

1. L'utilisateur sélectionne cette fonctionnalité depuis le cas « gérer les rovers »
2. Une IHM secondaire lui permet de visualiser la liste des rovers enregistrés dans le fichier
3. L'utilisateur choisit le rover à supprimer dans la liste
4. L'utilisateur confirme la suppression du rover
5. Le rover sélectionné est supprimé de la liste et le fichier est mis à jour.

Remarque : l'utilisateur peut annuler cette opération à tout moment.

**Scénario alternatif :** Aucun

## Cas d'utilisation « Sélectionner un rover »

**Objectif :** Permettre à l'utilisateur de sélectionner un rover dans la liste des rovers existants.

**Acteurs concernés :** L'utilisateur de l'application

**Pré conditions :** Aucune

**Scénario nominal :**

1. L'utilisateur sélectionne cette fonctionnalité dans le menu de l'IHM
2. Une IHM secondaire lui permet de visualiser la liste des rovers enregistrés dans le fichier
3. L'utilisateur sélectionne le rover dans la liste
4. L'utilisateur confirme la sélection du rover, à ce moment l'IHM secondaire se ferme et les paramètres de configuration du rover sont récupérés dans l'application.

Remarque : l'utilisateur peut annuler cette opération à tout moment.

**Scénario alternatif :** Aucun

## Cas d'utilisation « Piloter le rover »

**Objectif :** Permettre à l'utilisateur de contrôler les déplacements du rover.

**Acteurs concernés :** L'utilisateur de l'application

**Pré conditions :** Un rover est sélectionné

**Scénario nominal :**

1. L'utilisateur sélectionne cette fonctionnalité dans le menu de l'IHM
2. Une IHM secondaire lui permet de contrôler les mouvements du rover à l'aide d'un ensemble de boutons, l'utilisateur peut également régler les vitesses de déplacement.

Remarque : l'utilisateur peut annuler cette opération à tout moment.

**Scénario alternatif** : Aucun

### Cas d'utilisation « Utiliser un joystick »

**Objectif** : Permettre à l'utilisateur de contrôler les déplacements du rover grâce à un joystick virtuel.

**Acteurs concernés** : L'utilisateur de l'application

**Pré conditions** : Un rover est sélectionné

**Scénario nominal** :

1. L'utilisateur sélectionne cette fonctionnalité le cas « Piloter le rover »
2. Une IHM secondaire lui permet contrôler les mouvements du rover à l'aide d'un joystick virtuel, la vitesse de déplacement du rover est proportionnelle au déplacement du joystick.

Remarque : l'utilisateur peut annuler cette opération à tout moment.

**Scénario alternatif** : Aucun

### Cas d'utilisation « Utiliser la synthèse vocale »

**Objectif** : Permettre à l'utilisateur de contrôler les déplacements du rover grâce à la voix.

**Acteurs concernés** : L'utilisateur de l'application

**Pré conditions** : Un rover est sélectionné

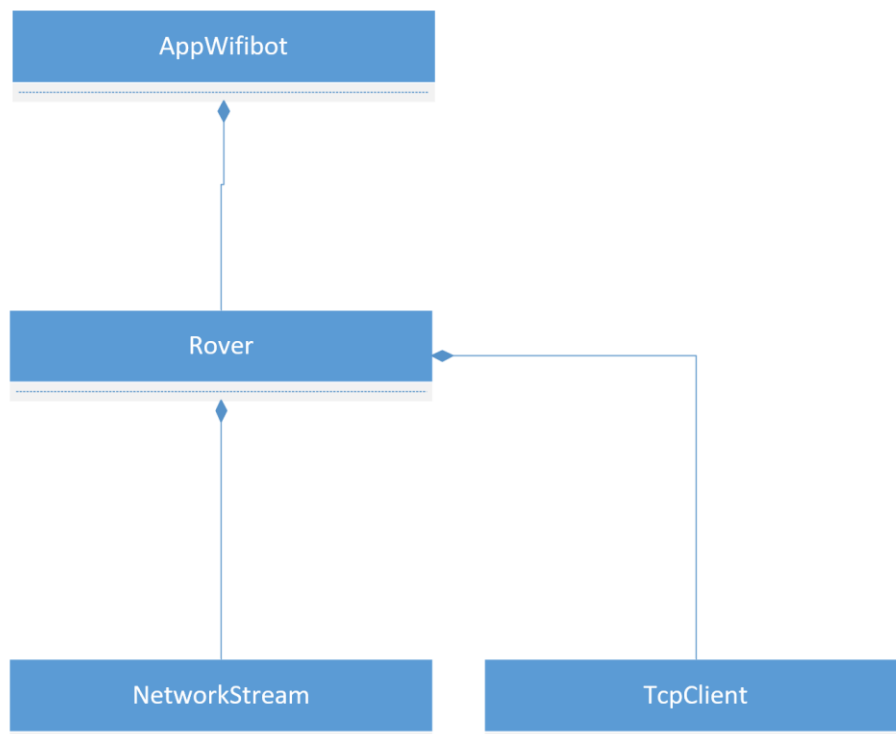
**Scénario nominal** :

1. L'utilisateur sélectionne cette fonctionnalité le cas « Piloter le rover »
2. Une IHM secondaire lui permet contrôler les mouvements du rover en donnant oralement les ordres de déplacement. Les ordres de déplacement sont définis dans un catalogue.
3. Chaque ordre donné est également est affiché dans l'IHM secondaire

Remarque : l'utilisateur peut annuler cette opération à tout moment.

**Scénario alternatif** : Aucun

### 3. Diagramme de classe partiel



« **AppWifibot** » correspond à l'application attendue par le client. Elle met à disposition de l'utilisateur une IHM qui propose toutes les fonctionnalités décrites dans le DCU.

« **Rover** » est le module logiciel qui permet de contrôler un rover. Ce module a déjà été développé dans l'activité 3.

Les classes « **TcpClient** » et « **NetworkStream** » sont des composants issus du framework .net

## 4. TRAVAIL DEMANDÉ

### 4.1. Modélisation UML

Sous MagicDraw, créer un nouveau projet puis éditer les diagrammes suivants :

- Le diagramme des cas d'utilisation.
- Le diagramme de classes en y faisant apparaître les classes de votre application ainsi que leurs méthodes.
- Le diagramme de séquence de chaque cas d'utilisation.

## 4.2. Maquettage des IHM

Télécharger installer le logiciel « Balsamiq Mockups ».

Concevoir à l'aide de ce logiciel une interface graphique permettant de simuler et de vérifier toutes les fonctionnalités attendues par le client au regard du diagramme des cas d'utilisation.

## 4.3. Codage et développement de l'application

- Réaliser les IHM en langage C# sous l'environnement de développement Visual Studio 2017.
- Coder et tester toutes les fonctionnalités attendues.

## 4.4. Compte rendu des tests

- Réaliser un compte rendu contenant les résultats des tests de chaque fonctionnalité.

**Fin de l'activité 4.**