

Rapport projet I62

Tom Bartier, Nino Cantera, Balboni Ivano

May 13, 2024

Ce rapport concerne le projet du module I62 Génie Logiciel de la L3 Informatique de l'université de Toulon. Le but de ce projet était de nous faire découvrir les différents concepts théoriques du génie logiciel et de les appliquer sur un projet pratique travaillé pendant les séances de TP ainsi qu'en dehors des heures de cours.

Contents

1	Introduction	2
1.1	Infos Pratiques	2
2	Informations générales	2
3	Exigences	4
3.1	Interactions client/serveur	4
3.2	GUI	4
3.3	Caméra	6
3.4	Rover	7
3.5	Hélicoptère	8
3.6	Environnement	9
4	Scénarios	10
4.1	Admin Modifier Environnement	10
4.2	Scénario Client Piloter Hélico	11
4.3	Scénario Piloter Rover	12
5	Sources	14
6	Diagrammes	14

1 Introduction

Notre projet était de mettre au point un simulateur de rover explorant la planète Mars. L'application est en mode client / serveur, elle permet à l'utilisateur de contrôler un rover et d'explorer la planète Mars avec d'autres utilisateurs, d'analyser différents matériaux ou encore de déployer un drone afin de faciliter l'exploration.

1.1 Infos Pratiques

L'application ne peut se lancer que si le serveur est allumé. Pour lancer le serveur, exécutez la commande `python ./src/server/app-server.py <ip> <port>` où <ip> correspond à l'adresse ip de votre machine et <port> désigne le port sur lequel le serveur attendra les connexions. Exemple : `python ./src/server/app-server.py 127.0.0.1 25025`

Pour lancer l'application client, exécutez la commande `python ./src/client/app-client.py <ip> <port>`

Côté client vous arrivez sur la page de connexion, deux comptes sont disponibles :

1. Utilisateur : bob Mot de Passe : 1234
2. Utilisateur : alice Mot de Passe : 5678

Si la connexion ne fonctionne pas c'est que vous avez entré la mauvaise IP / Port, ou que le serveur n'était pas allumé lorsque vous avez lancé le client.

Vous devez arriver sur l'écran principal, utilisez les flèches directionnelles pour vous déplacer et cliquez sur les boutons à l'écran pour changer entre le rover et l'hélicoptère, ou pour casser et analyser un rocher. À noter que pour ranger l'hélicoptère vous devez vous placer sur le rover, et pour analyser un rocher vous devez être devant orienté dans la bonne direction. Il n'y a pas encore de manière propre de quitter le client et le serveur, fermez juste la fenêtre du client et le terminal du serveur.

2 Informations générales

Ce projet est donc un simulateur de rover sur Mars écrit en Python et qui a été séparé en deux grandes phases : une première phase de conception en séance de TP où l'on a réfléchi à la conception du projet en réalisant les différents diagrammes UML vu en cours (voir 6). C'est durant cette phase que nous nous sommes mis d'accord sur la structuration du projet ainsi que les différentes fonctionnalités proposées.

Le logiciel est dans l'idée destiné à une entreprise pour un usage privé, il n'est donc pas disponible au public, c'est pourquoi nous avons décidé de ne pas implémenter de fonction pour créer un compte directement depuis l'écran de connexion. L'application administrateur n'a pas pu être développée par manque de temps, mais elle est prévue dans la conception, elle est très fortement liée au serveur, c'est à dire qu'elle se lance sur la même machine que le serveur en même temps et elle est unique, il n'y a donc qu'un

seul administrateur. C'est lui qui doit créer un nouveau compte si une nouvelle personne veut avoir accès au simulateur.

Parmi les différentes fonctionnalités imaginées mais que nous n'avons pas pu mettre en place, nous pouvons citer la prise de photos et vidéo, que nous n'avons pas jugés pertinents d'implémenter dans notre programme en 2D, mais si le simulateur avait été fait en 3D cette fonctionnalité aurait eu sa place. Nous avons aussi imaginé un système d'exploration en équipe avec d'autres utilisateurs : chaque utilisateur aurait une représentation minimaliste de la planète dans un coin de l'écran donnant des informations sur l'emplacement actuel de chaque utilisateur ainsi que les zones qu'ils ont explorées. Les zones pas encore découvertes seraient cachées et lorsque qu'un utilisateur découvrirait une nouvelle zone les informations sur cette dernière seraient retransmises à tous les autres utilisateurs.

Nous avons aussi imaginé et essayé d'implanter un système de topographie. La carte de Mars que nous avons est colorée de façon à représenter la topographie de Mars (voir 5). Nous avons essayé de déterminer un niveau de hauteur en fonction de la couleur de chaque pixel afin de simuler la présence de montagnes et cratères. Le rover serait gêné dans sa progression par ces derniers, d'où l'intérêt de déployer le drone afin de faciliter l'exploration sur des terrains escarpés. Mais l'implémentation de ce système a posé trop de problèmes, il est cependant toujours présent dans le code source mais non utilisé.

Nous avons aussi mis en place un système permettant de simuler la météo sur Mars. La météo est générée de manière pseudo-aléatoire à l'aide de la bibliothèque Random de python. Cela se base sur une série de chiffres qui garantit qu'on obtiendra les memes meteos en fonction de la graine. Chaque météo est créée avec comme Id unique sa date de création, il ne peut donc pas avoir 2 météos créés exactement en même temps. Ensuite, chacun de ses paramètres est généré dans une fourchette de valeurs ce qui inclut la position de départ, le sens et vitesse de déplacement, la durée et le type de météo. Ces météos sont stockées dans un dictionnaire ce qui les rend facilement compatibles avec le format json et des BDD SQL. Le pseudo-aléatoire permet de générer les météos au fur et à mesure au besoin, il est possible de rajouter/supprimer des types de météos au besoin. Concernant la mise en place du serveur, nous nous sommes basés sur un modèle de serveur déjà existant (voir 5) que nous avons modifiés afin de répondre à nos attentes. Il utilise la librairie socket de Python ainsi que selectors, qui est une manière de faire un serveur asynchrone non bloquant en évitant d'avoir à gérer plusieurs threads. Nous utilisons une unique classe Message transitant sur le réseau et contenant les différentes requêtes et réponses, mais nous ne l'avons pas mentionné dans les diagrammes car nous le considérons comme une API, un peu comme Tkinter, nous avons donc pas jugés nécessaire de faire apparaître cette classe dans les diagrammes.

Nous avons aussi pensés au fait que le serveur communiquerait avec une base de données. Toutes les données concernant les utilisateurs seraient stockées dans une base de donnée SQL mais par manque de temps nous avons dans notre programme simulés cette base de données avec un fichier json dans lequel nous lisons les données à l'allumage du serveur et écrivons les nouvelles données à stocker (en théorie) périodiquement ainsi qu'à l'extinction du serveur.

Concernant la répartition des tâches, Nino Cantera s'est occupé d'une grande partie de

l'IHM, Ivano Balboni s'est occupé de la génération de la topographie, de la météo et de quelques classes serveur et Tom BARTIER s'est chargé de la communication entre le client et le serveur, le traitement des requêtes des deux côtés, quelques classes serveur et quelques éléments de l'IHM. La quasi totalité du code a été écrit en dehors des séances de TP.

3 Exigences

Voici les exigences mises en place pour ce projet, à noter qu'elles n'ont pas toutes pu être implémentées mais qu'elles pourraient l'être si le projet venait à être poursuivi après ce rendu.

3.1 Interactions client/serveur

- EX_SERV_0002
 - S'authentifier
 - Le SI doit permettre à l'utilisateur de s'authentifier grâce à un identifiant et un mot de passe.
- EX_SERV_0003
 - Voir les autres
 - Le SI doit permettre à l'utilisateur de voir les autres autres utilisateur dans l'environnement
- EX_SERV_0004
 - Stockage environnement
 - Le SI héberge sur le serveur l'environnement dans lequel évoluent les rovers.

3.2 GUI

- EX_GUI_0001
 - Avancer
 - Le SI doit permettre à l'utilisateur de faire avancer le rover.
- EX_GUI_0002
 - Reculer
 - Le SI doit permettre à l'utilisateur de faire reculer le rover.
- EX_GUI_0003
 - Pivoter

- Le SI doit permettre à l'utilisateur de faire pivoter le rover dans les deux sens (sens horaire et antihoraire).
- EX_GUI_0004
 - Contrôler la vitesse
 - Le SI doit permettre à l'utilisateur de contrôler la vitesse du rover.
- EX_GUI_0005
 - Utiliser Laser
 - Le SI doit permettre à l'utilisateur de tirer un laser sur un rocher.
- EX_GUI_0006
 - Surveiller énergie
 - Le SI doit permettre à l'utilisateur de surveiller la quantité d'énergie restante au rover.
- EX_GUI_0007
 - Analyser
 - Le SI doit permettre à l'utilisateur d'obtenir les informations sur la matière pulvérisée et analysée par le rover.
- EX_GUI_0008
 - Creuser
 - Le SI doit permettre à l'utilisateur d'indiquer au rover de creuser dans le sol à l'aide de sa foreuse
- EX_GUI_0009
 - Pilote Automatique
 - Le SI doit permettre à l'utilisateur de mettre le rover en mode pilote automatique.
- EX_GUI_00010
 - Mode Manuel
 - Le SI doit permettre à l'utilisateur de mettre le rover en mode manuel.
- EX_GUI_00011
 - Allumer
 - Le SI doit permettre à l'utilisateur d'allumer le rover.
- EX_GUI_00012
 - Eteindre
 - Le SI doit permettre à l'utilisateur d'éteindre le rover.

- EX_GUI_0013
 - Mini carte
 - Le SI doit permettre à l'utilisateur de consulter une mini carte de la planète (en ne voyant clairement que les zones découvertes).
- EX_GUI_0014
 - Affichage température
 - Le SI doit permettre à l'utilisateur de consulter la température actuelle de la zone.

3.3 Caméra

- EX_CAM_0001
 - Affichage Caméra
 - Le SI doit permettre à l'utilisateur de voir le retour de la caméra en temps réel.
- EX_CAM_0002
 - Pivoter Caméra
 - Le SI doit permettre à l'utilisateur de faire pivoter la caméra dans tous les sens.
- EX_CAM_0003
 - Zoom
 - Le SI doit permettre à l'utilisateur de zoomer et dézoomer la caméra.
- EX_CAM_0004
 - Prendre des photos
 - Le SI doit permettre à l'utilisateur de prendre des photos avec la caméra et de les enregistrer sur sa machine.
- EX_CAM_0005
 - Prendre des vidéos
 - Le SI doit permettre à l'utilisateur de prendre des vidéos.
- EX_CAM_0006
 - Pivoter
 - Le SI doit permettre à l'utilisateur de faire pivoter la caméra.
- EX_CAM_0007

- Enregistrer Photos
 - Le SI doit permettre à l'utilisateur d'enregistrer les photos prises.
- EX_CAM_0008
 - Enregistrer Photos
 - Le SI doit permettre à l'utilisateur d'enregistrer les vidéos prises.

3.4 Rover

- EX_ROVER_0001
 - S'abîmer
 - Le SI doit infliger des dégâts au rover en cas de colision avec un obstacle ou en cas de chute.
- EX_ROVER_0002
 - Panne d'énergie
 - Le SI doit permettre au rover de tomber en panne d'énergie.
- EX_ROVER_0003
 - Recharge d'énergie
 - Le SI doit permettre au rover de se recharger en énergie.
- EX_ROVER_0004
 - Remplacer foreuse
 - Le SI doit permettre au rover d'abandonner sa foreuse et la remplacer par une autre si la première se retrouve coincée ou endommagée.
- EX_ROVER_0005
 - Découvrir les alentours
 - Le SI doit permettre au rover de découvrir les alentours.
- EX_ROVER_0006
 - Prévoir météo
 - Le SI doit permettre au rover de prévoir de prévoir le prochain évènement météorologique de la zone où il se trouve.

3.5 Hélicoptère

- EX_HELI_0001
 - Déployer hélicoptère
 - Le SI doit permettre au rover de déployer l'hélicoptère
- EX_HELI_0002
 - Ranger hélicoptère
 - Le SI doit permettre au rover de ranger l'hélicoptère
- EX_HELI_0003
 - Monter
 - Le SI doit permettre à l'utilisateur de faire monter l'hélicoptère
- EX_HELI_0004
 - Descendre
 - Le SI doit permettre à l'utilisateur de faire descendre l'hélicoptère
- EX_HELI_0005
 - Avancer
 - Le SI doit permettre à l'utilisateur de faire avancer l'hélicoptère
- EX_HELI_0006
 - Reculer
 - Le SI doit permettre à l'utilisateur de faire reculer l'hélicoptère
- EX_HELI_0007
 - Aller à gauche
 - Le SI doit permettre à l'utilisateur de faire aller l'hélicoptère vers la gauche
- EX_HELI_0008
 - Aller à droite
 - Le SI doit permettre à l'utilisateur de faire aller l'hélicoptère vers la droite
- EX_HELI_0009
 - Pivoter
 - Le SI doit permettre à l'utilisateur de faire pivoter l'hélicoptère
- EX_HELI_0010
 - Caméra Hélico

- Le SI doit permettre à l'utilisateur contrôler la caméra de l'hélicoptère (voir rubrique Caméra)
- EX_HELI_0011
 - Energie
 - Le SI doit permettre à l'hélicoptère de pouvoir tomber en panne d'énergie
- EX_HELI_0012
 - Energie
 - L'hélicoptère doit pouvoir recharger son énergie en se posant rentrant à terre.
- EX_HELI_0013
 - Décollage
 - Le SI doit permettre à l'hélicoptère de décoller
- EX_HELI_0014
 - Atterrissage
 - Le SI doit permettre à l'hélicoptère d'atterrir
- EX_HELI_0015
 - Vitesse
 - Le SI doit permettre à l'utilisateur de changer la vitesse de l'hélicoptère
- EX_HELI_0016
 - Décollage
 - Le SI doit permettre à l'utilisateur de mettre l'hélicoptère en mode automatique.
- EX_HELI_0016
 - Cartographier
 - Le SI doit permettre à l'hélicoptère de cartographier ses environs

3.6 Environnement

- EX_ENV_0001
 - Brouillard
 - Le SI affiche à l'utilisateur un brouillard dans les zones non découvertes par le rover et l'hélicoptère
- EX_ENV_0002
 - Rocher

- Le SI dispose des rochers dans l'environnement
- EX_ENV_0002
 - Rocher
 - Le SI dispose de différents niveaux de hauteur dans l'environnement
- EX_ENV_0003
 - Tempête de poussière
 - Le SI dispose d'un évènement "tempête de poussière" sur une zone qui réduit la distance de vision et endommage un petit peu l'hélicoptère lorsqu'il est en vol et empêche la recharge d'énergie solaire.
- EX_ENV_0004
 - Vent
 - Le SI dispose d'un évènement "vent" sur une zone avec différents niveaux d'intensité qui vont avoir une influence sur l'hélicoptère.

4 Scénarios

4.1 Admin Modifier Environnement

Scénario nominal :

1. L'administrateur clique sur le bouton "Modifier Environnement"
2. Le système affiche l'interface de modification de l'environnement
3. L'administrateur clique sur l'icône du rocher
4. Le système demande les coordonnées et le type du rocher à ajouter
5. L'administrateur renseigne les coordonnées auxquelles le rocher sera ajouté ainsi que le type
6. Le système confirme l'ajout du rocher sur la carte

Scénario alternatif :

3. L'administrateur clique sur l'icône tempête de poussière
4. Le système demande dans quelle zone sera la tempête de poussière ainsi que sa date et sa durée
5. L'administrateur renseigne les informations
6. Le système confirme l'ajout de la tempête de poussière

Scénario exception :

5. L'administrateur renseigne les coordonnées auxquelles le rocher sera ajouté ainsi que le type

6. Le système indique qu'un rocher est déjà présent à ces coordonnées, echec de l'ajout

4.2 Scénario Client Piloter Hélico

Scénario nominal:

1. L'utilisateur appuie sur la touche permettant de déployer l'hélicoptère
2. L'hélicoptère virtuel se détache du rover et remplace temporairement le rover en tant que véhicule piloté par l'utilisateur
3. Le système affiche la jauge d'énergie de l'hélicoptère
4. L'utilisateur appuie sur la touche permettant d'avancer
5. L'hélicoptère avance tant que la touche est enfoncée
6. L'utilisateur relâche la touche avant que l'hélicoptère rencontre un obstacle
7. L'hélicoptère arrête d'avancer
8. L'utilisateur appuie sur la touche permettant de monter
9. L'altitude de l'hélicoptère augmente
10. L'utilisateur appuie sur la touche pour avancer
11. L'hélicoptère avance et passe au-dessus de l'obstacle
12. L'utilisateur remarque que la jauge d'énergie de l'hélicoptère est basse
13. L'utilisateur relâche la touche
14. L'hélicoptère arrête d'avancer
15. L'utilisateur appuie sur la touche permettant d'atterrir
16. L'hélicoptère atterrit et commence à recharger sa batterie
17. L'utilisateur attend que la jauge d'énergie se remplisse
18. Une fois la jauge remplie l'utilisateur appuie sur la touche permettant de décoller
19. L'hélicoptère se remet en route
20. L'utilisateur appuie sur la touche permettant de pivoter
21. L'hélicoptère pivote sur lui-même tant que la touche est enfoncée
22. L'utilisateur relâche la touche
23. L'hélicoptère arrête de pivoter
25. L'utilisateur appuie sur la touche permettant de ranger l'hélicoptère
26. L'hélicoptère vient se poser sur le rover
27. L'utilisateur reprend le contrôle du rover

Cas alternatif: Contournement de l'obstacle

7. L'utilisateur appuie sur la touche permettant de pivoter
8. L'hélicoptère pivote sur lui-même tant que la touche est enfoncée
9. L'utilisateur relâche la touche après avoir fait pivoter l'hélicoptère à 45°
10. L'utilisateur appuie sur la touche permettant d'avancer
11. L'hélicoptère avance tant que la touche est enfoncée
12. L'utilisateur relâche la touche après avoir fait parcourir une courte distance à l'hélicoptère

13. L'hélicoptère arrête d'avancer
12. L'utilisateur appuie sur la touche permettant de pivoter
13. L'hélicoptère pivote sur lui-même tant que la touche est enfoncée
14. L'utilisateur relâche la touche après avoir fait pivoter l'hélicoptère à 45°
15. L'utilisateur appuie sur la touche permettant d'avancer
16. L'hélicoptère avance tant que la touche est enfoncée
17. L'utilisateur relâche la touche
18. L'hélicoptère arrête d'avancer

Le scénario reprend à l'étape 11.

Cas d'exception: Batterie vide

11. L'utilisateur reste appuyé sur la touche pendant plusieurs minutes
12. La jauge d'énergie de l'hélicoptère diminue et se vide
13. L'hélicoptère tombe au sol
14. L'hélicoptère perd des points de vie
15. L'hélicoptère tombe à court de points de vie
16. Le système indique que l'hélicoptère est trop endommagé pour décoller
17. L'utilisateur reprend le contrôle du rover
18. L'utilisateur appuie sur la touche permettant de contrôler l'hélicoptère
19. Le système indique que l'hélicoptère est hors-service

4.3 Scénario Piloter Rover

Scénario nominal: Piloter le rover

Scénario principal

1. L'utilisateur lance une partie en appuyant sur un bouton
2. Le programme principal se lance et affiche l'interface
3. L'utilisateur appuie sur la touche permettant d'avancer
4. Le rover virtuel avance tant que la touche est enfoncée
5. L'utilisateur relâche la touche
6. Le rover s'arrête
7. L'utilisateur appuie sur la touche permettant de reculer
8. Le rover recule tant que la touche est enfoncée
9. L'utilisateur relâche la touche
10. Le rover s'arrête
11. L'utilisateur appuie sur la touche permettant de pivoter
12. Le rover pivote sur lui-même tant que la touche est enfoncée
13. L'utilisateur relâche la touche
14. Le rover s'arrête

15. L'utilisateur appuie sur la touche permettant d'avancer
16. Le rover avance tant que la touche est enfoncée
17. L'utilisateur relâche la touche avant de percuter un rocher
18. Le rover s'arrête
19. L'utilisateur appuie sur la touche permettant de creuser
20. Le rover utilise sa foreuse pour détruire le rocher et en collecter un échantillon
21. L'utilisateur appuie sur la touche permettant de lancer une analyse
22. Le résultat de l'analyse est affiché à l'écran
23. L'utilisateur appuie sur la touche permettant de régler la vitesse
24. L'utilisateur appuie sur la touche permettant d'augmenter la vitesse
25. La jauge de vitesse augmente tant que la touche est enfoncée
26. L'utilisateur relâche la touche
27. La jauge de vitesse arrête d'augmenter
28. L'utilisateur appuie sur la touche permettant de confirmer la vitesse choisie
29. La vitesse choisie est appliquée
30. L'utilisateur appuie sur la touche permettant d'avancer
31. Le rover avance tant que la touche est enfoncée
32. L'utilisateur relâche la touche
33. Le rover s'arrête
34. L'utilisateur clique sur le bouton permettant d'afficher le menu
35. Le simulateur se met en pause et le menu s'affiche
36. L'utilisateur clique sur le bouton «revenir à l'écran-titre»
37. Le programme retourne à l'écran d'accueil
38. L'utilisateur clique sur le bouton «déconnexion»
39. L'utilisateur est déconnecté de son compte
40. L'utilisateur ferme le programme

Cas alternatif: Collision avec un obstacle

11. L'utilisateur reste appuyé sur la touche
12. Le rover ne s'arrête pas et percute un rocher
13. Le système indique que le rover est bloqué par un rocher
14. Le rover est endommagé et perd des points de vie

Le scénario reprend à l'étape 19

Cas d'exception: Echec de la mission

11. L'utilisateur reste appuyé sur la touche
12. Le rover ne s'arrête pas et continue d'avancer jusqu'à tomber du haut d'une montagne
13. Le rover tombe à court de points de vie
14. Le système indique que le rover est trop endommagé pour continuer sa mission
15. Le système indique que la partie est terminée

Le scénario reprend à l'étape 37

5 Sources

Base du modèle client / serveur (app-server.py, libserver.py, app-client.py, libclient.py) :

<https://realpython.com/python-sockets/>

Rover: https://www.researchgate.net/figure/D-Top-view-of-the-rover-with-dimensions_fig1_356749945

Rochers: <https://www.pngwing.com/en/free-png-zrzud>

Hélicoptère: <https://www.creativefabrica.com/fr/png/drone-png-ORY0xgTGA/>

Mars (arrière-plan authentification): <https://www.frostscience.org/volcanic-activity-on-mars-t>

Carte: <https://attic.gsfc.nasa.gov/mola/images.html>

Tempête : https://pngtree.com/freepng/strong-wind-vector_9038035.html

Tempête de poussière : <https://www.vecteezy.com/png/21913770-brown-watercolor-modern-brus>

6 Diagrammes

Voici les différents diagrammes UML de la conception du projet.

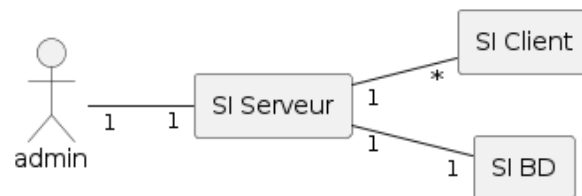


Figure 1: Diagramme de Contexte Statique Serveur

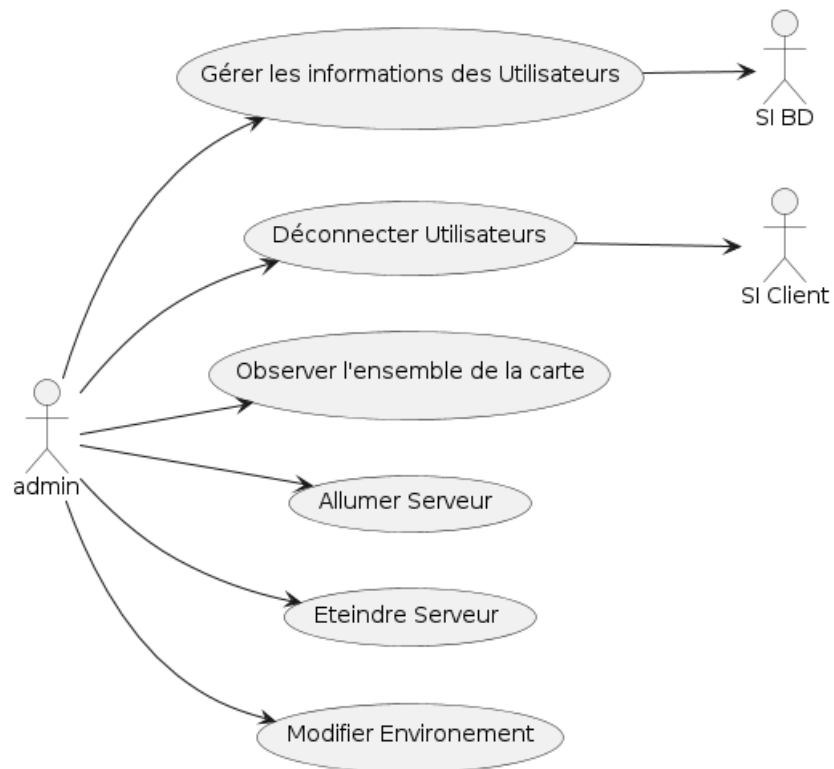


Figure 2: Diagramme de Cas D'utilisation Serveur

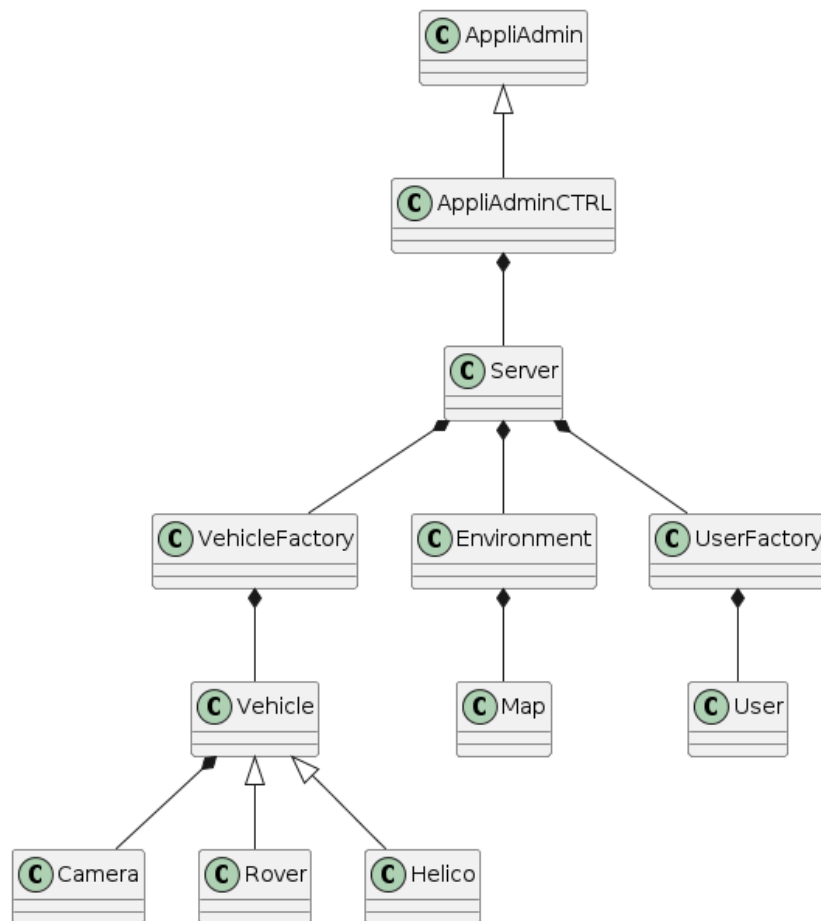


Figure 3: Diagramme de Classes Serveur

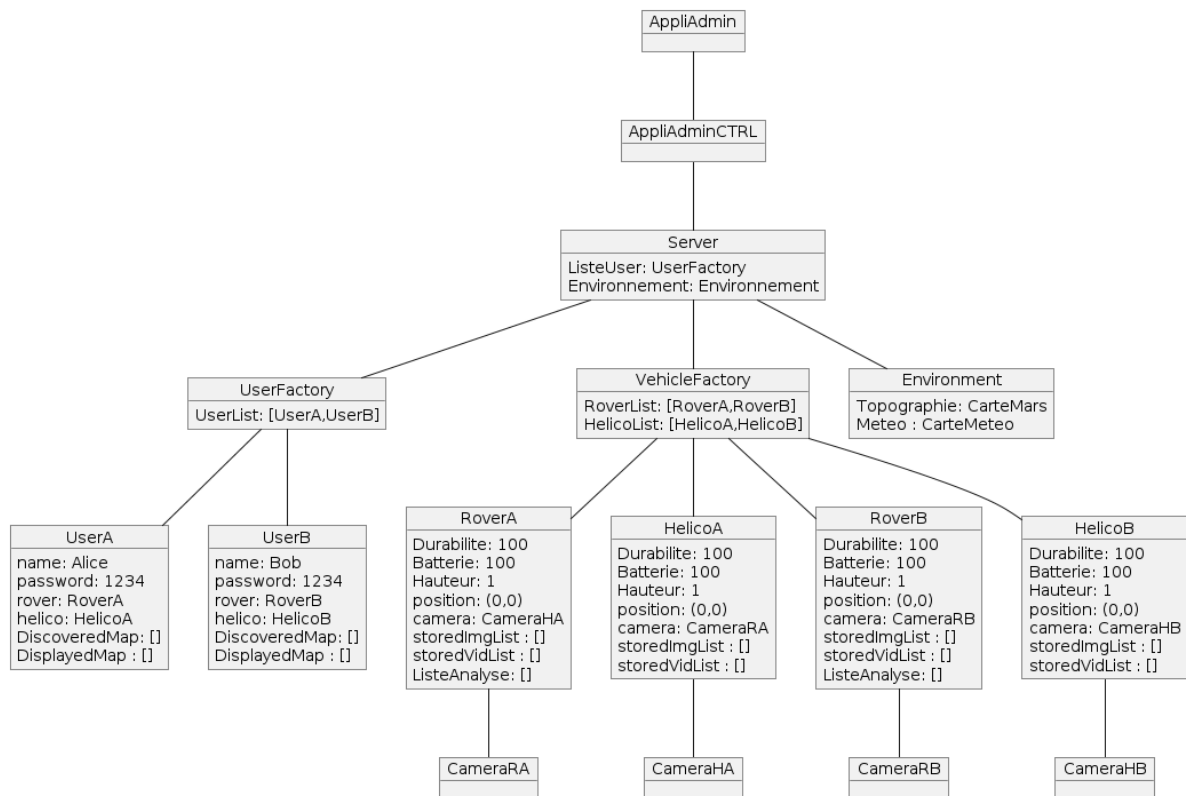


Figure 4: Diagramme d'Objet Serveur

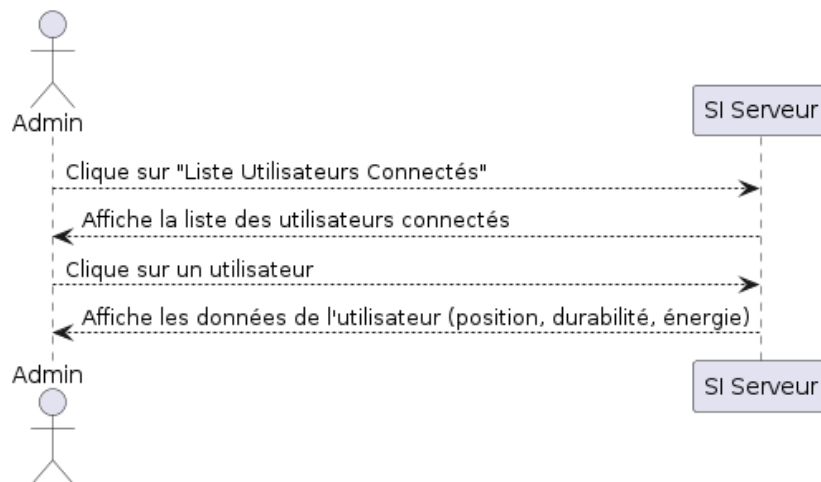


Figure 5: Diagramme de séquence Serveur UC Consulter Données Utilisateurs

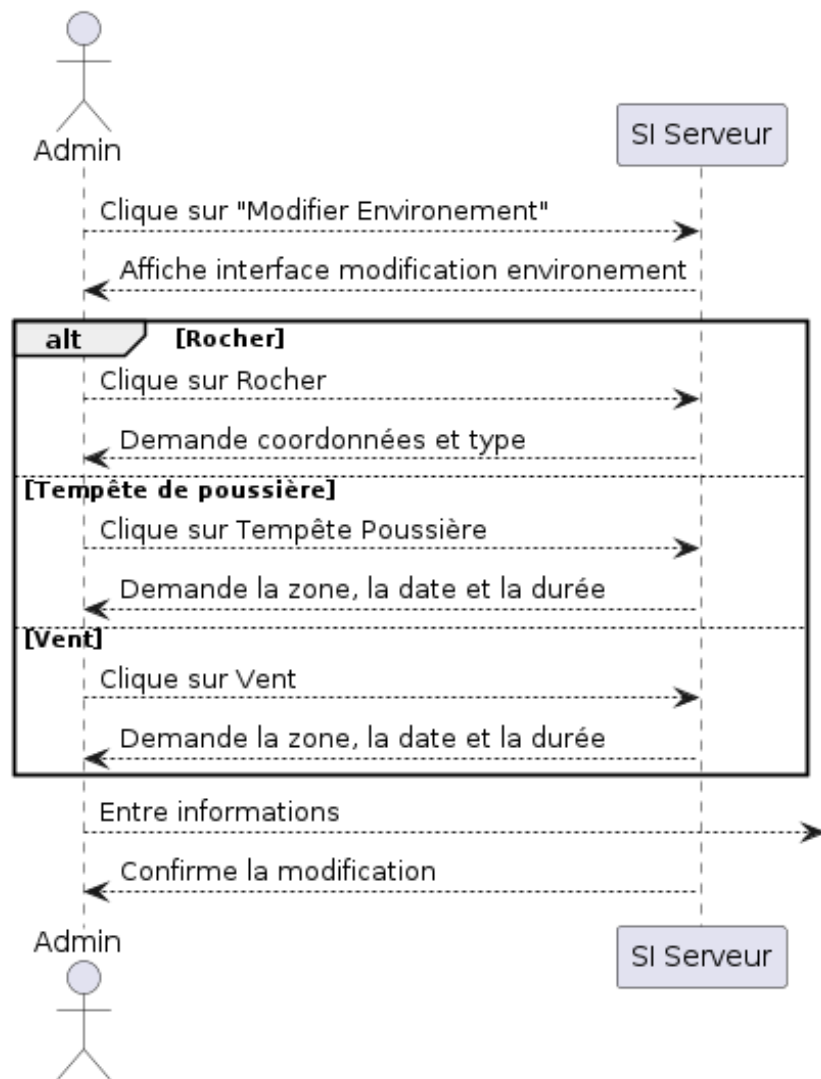


Figure 6: Diagramme de séquence Serveur UC Modifier l'Environnement



Figure 7: Diagramme de Contexte Statique Client

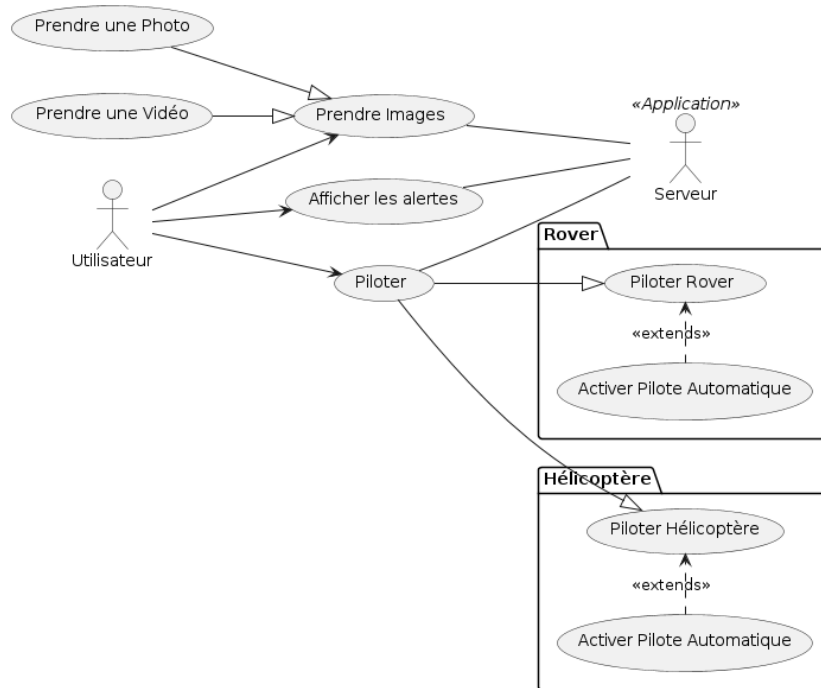


Figure 8: Diagramme de Cas d'Utilisations Client

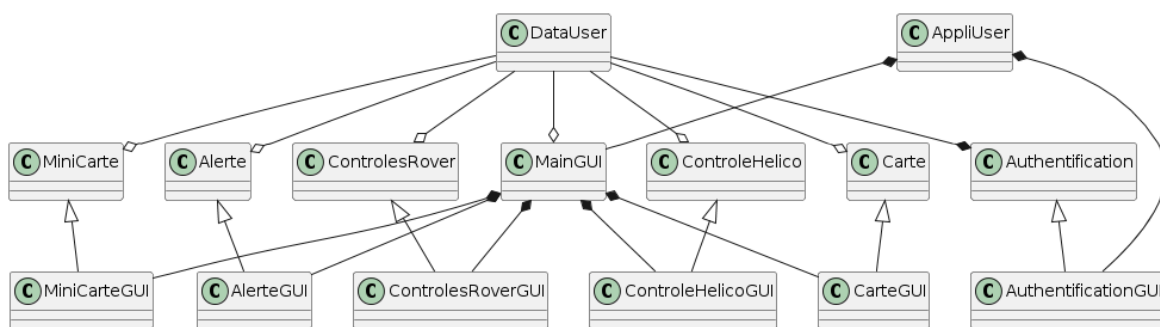


Figure 9: Diagramme de Classes Client

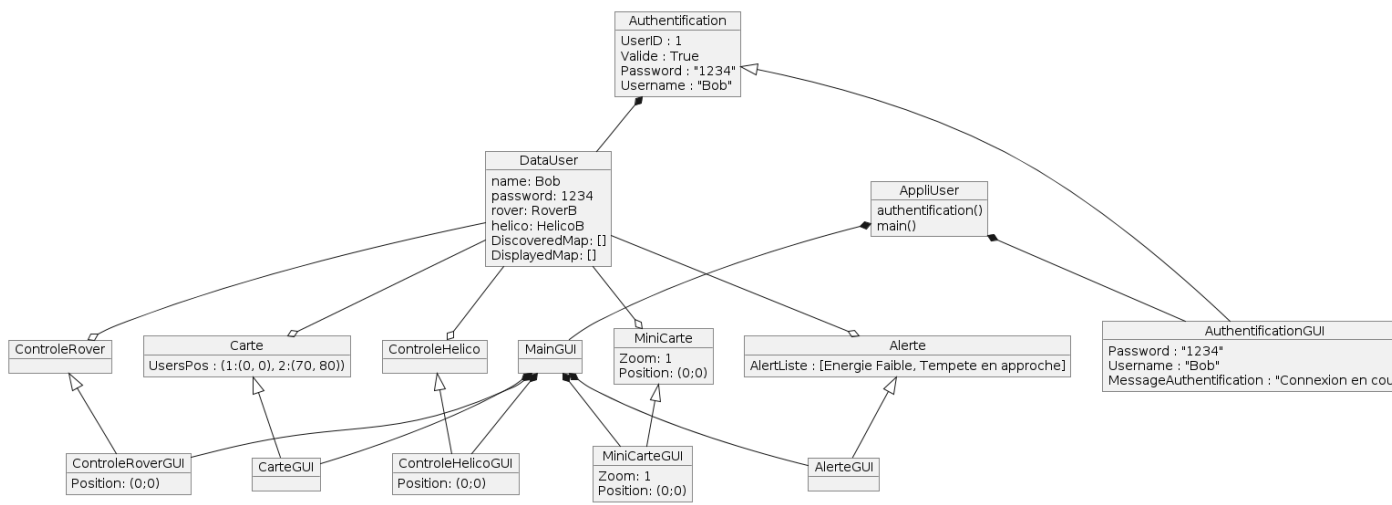


Figure 10: Diagramme d'Objets Client

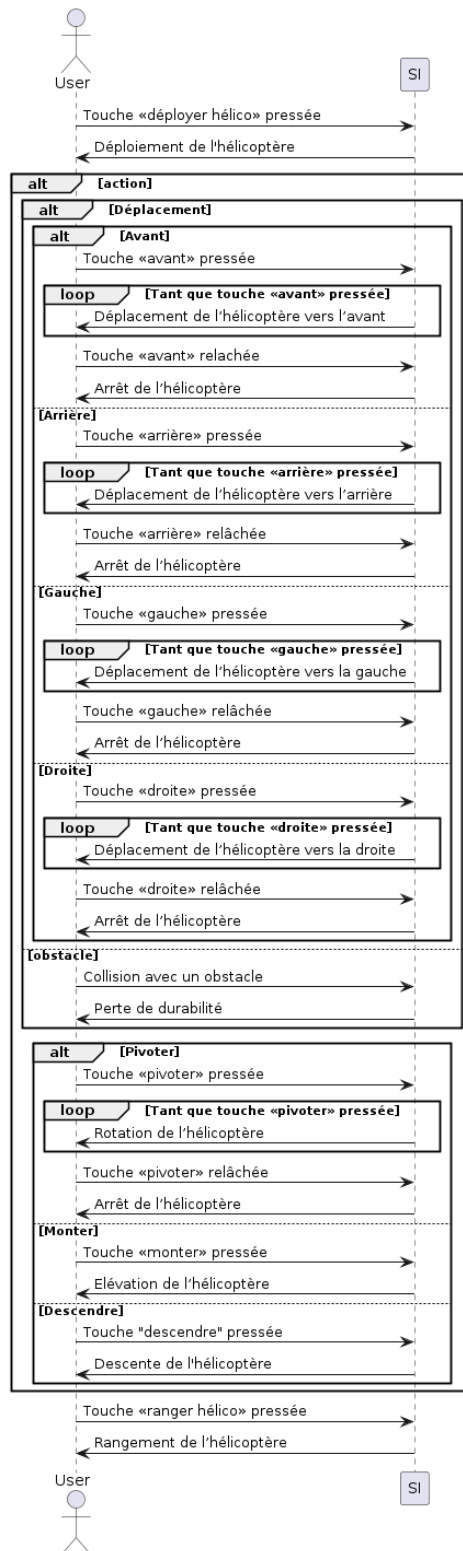


Figure 11: Diagramme de Séquence Client UC "Piloter hélicoptère"

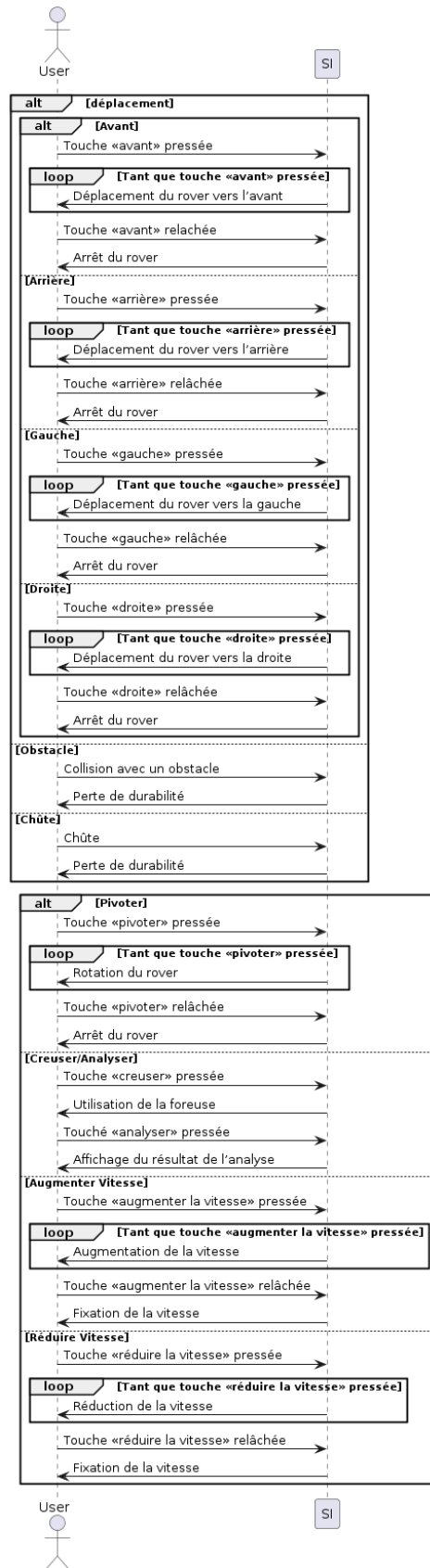


Figure 12: Diagramme de Séquence Client UC "Piloter rover"