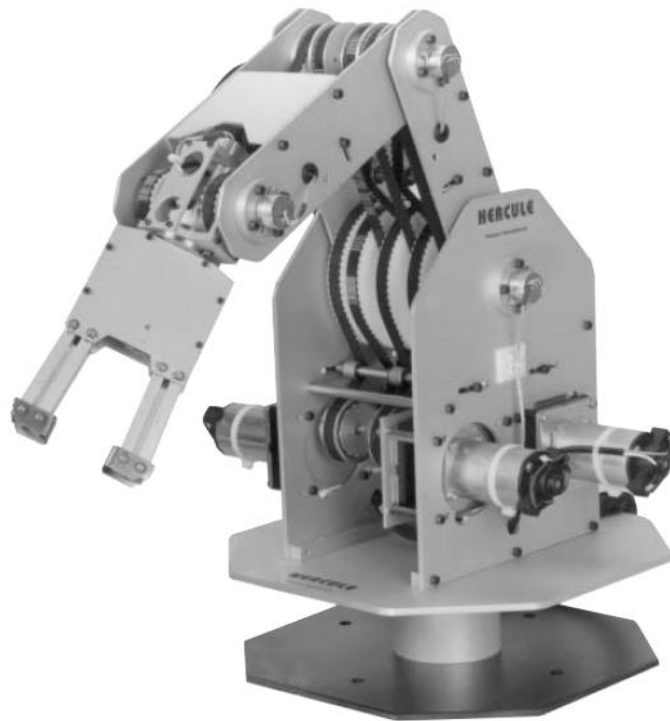


Pilotage à distance d'un robot Hercule



1 : KHEIRAT Abdellah
2 : ZAIR Aymane
3 : OUZEAU Alexandre

Description du projet

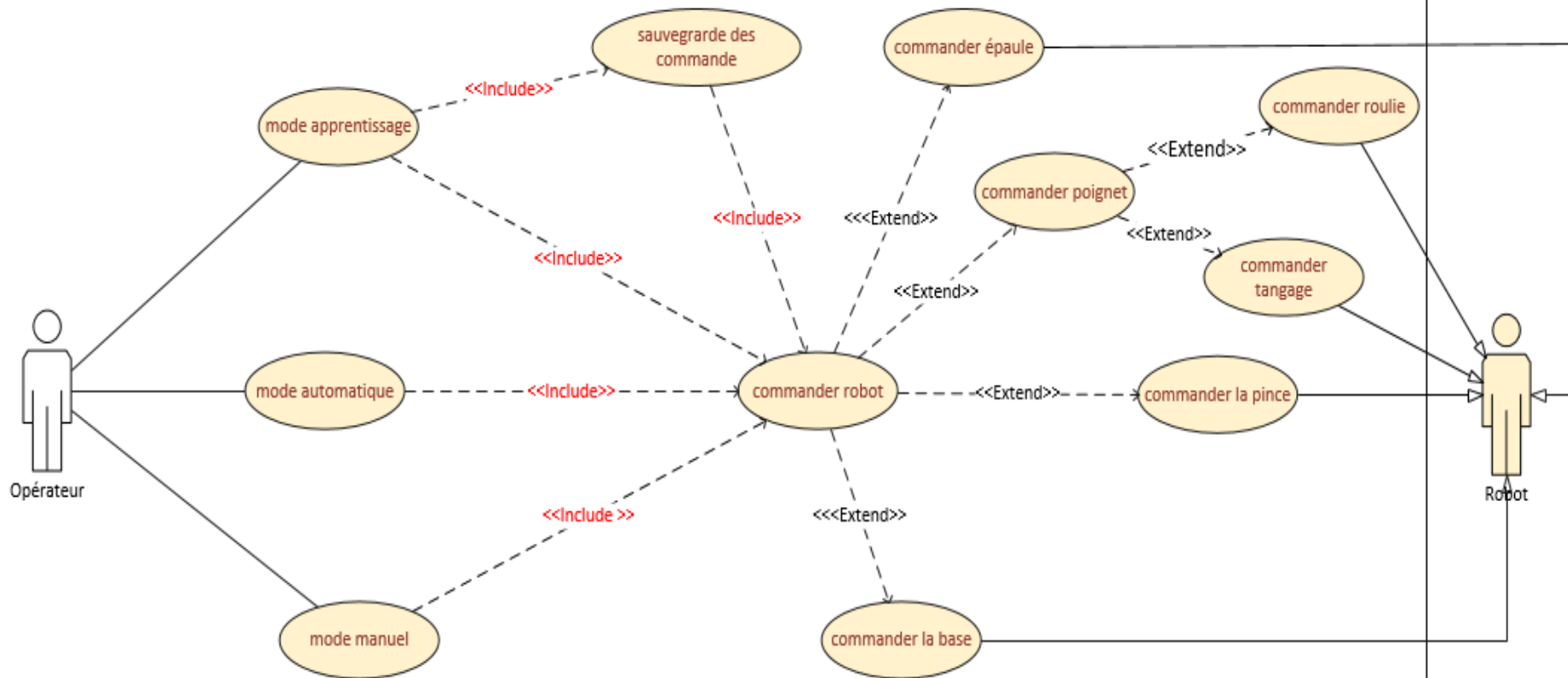
- ▶ Utilisation courante.
- ▶ Mode manuel.
- ▶ Mode apprentissage.
- ▶ Mode autonome.

Sommaire

- ▶ I) Diagramme des cas d'utilisation général
- ▶ II) Environnement matériel
- ▶ III) Cahier des charges personnel
- ▶ IV) Analyse UML
- ▶ V) Fonctionnement de l'application Android
- ▶ VI) Etat d'avancement du projet

Diagramme des cas d'utilisation général

► Diagramme des cas d'utilisation général



Environnement matériel



Joystick

USB



PC de pilotage

Ethernet



Caméra IP



PC de commande

Boîtier de commande



RS 232

Ethernet

Switch



Ethernet



Robot



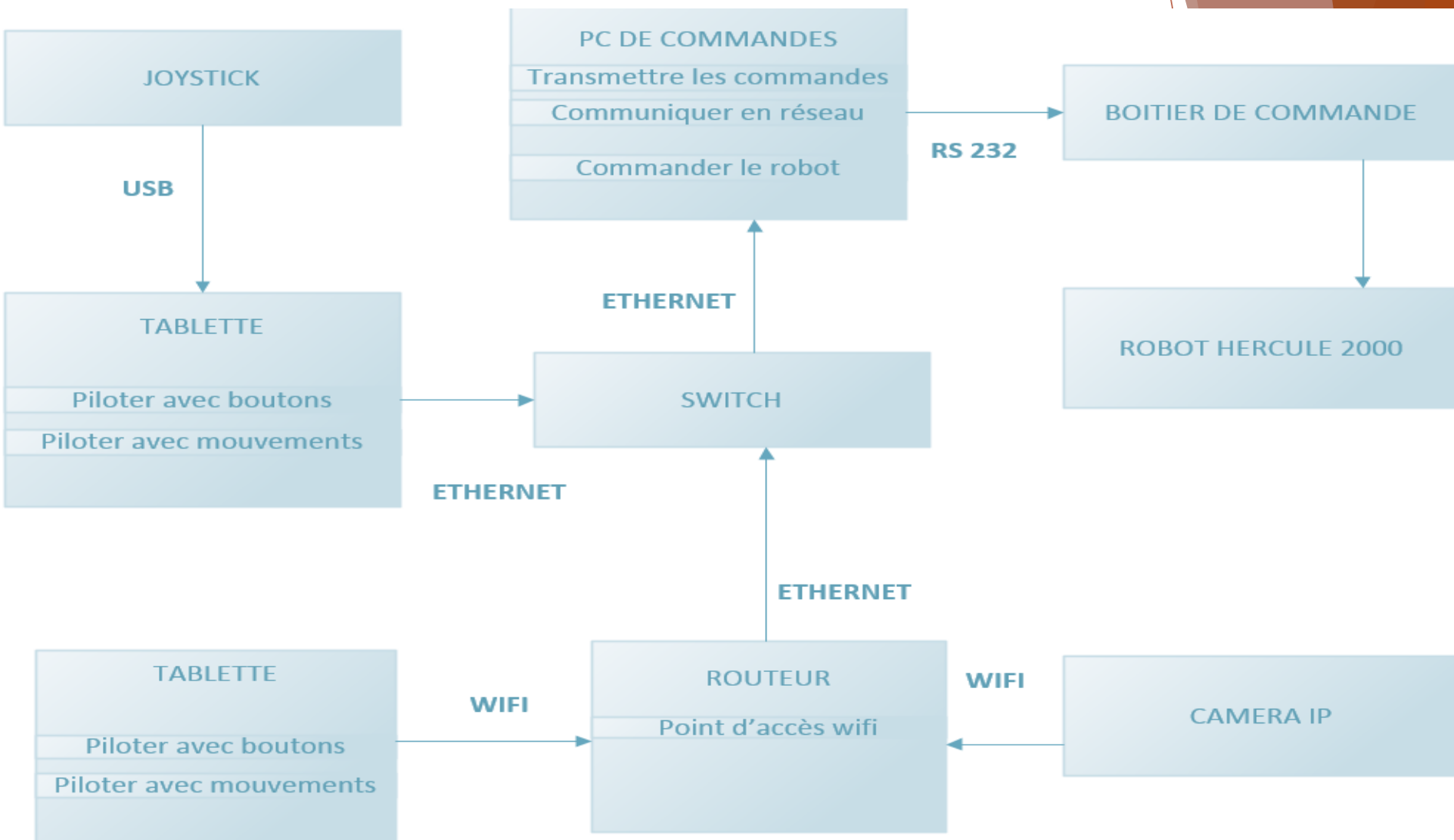
Point d'accès wifi

Wifi



Tablette

Diagramme de déploiement



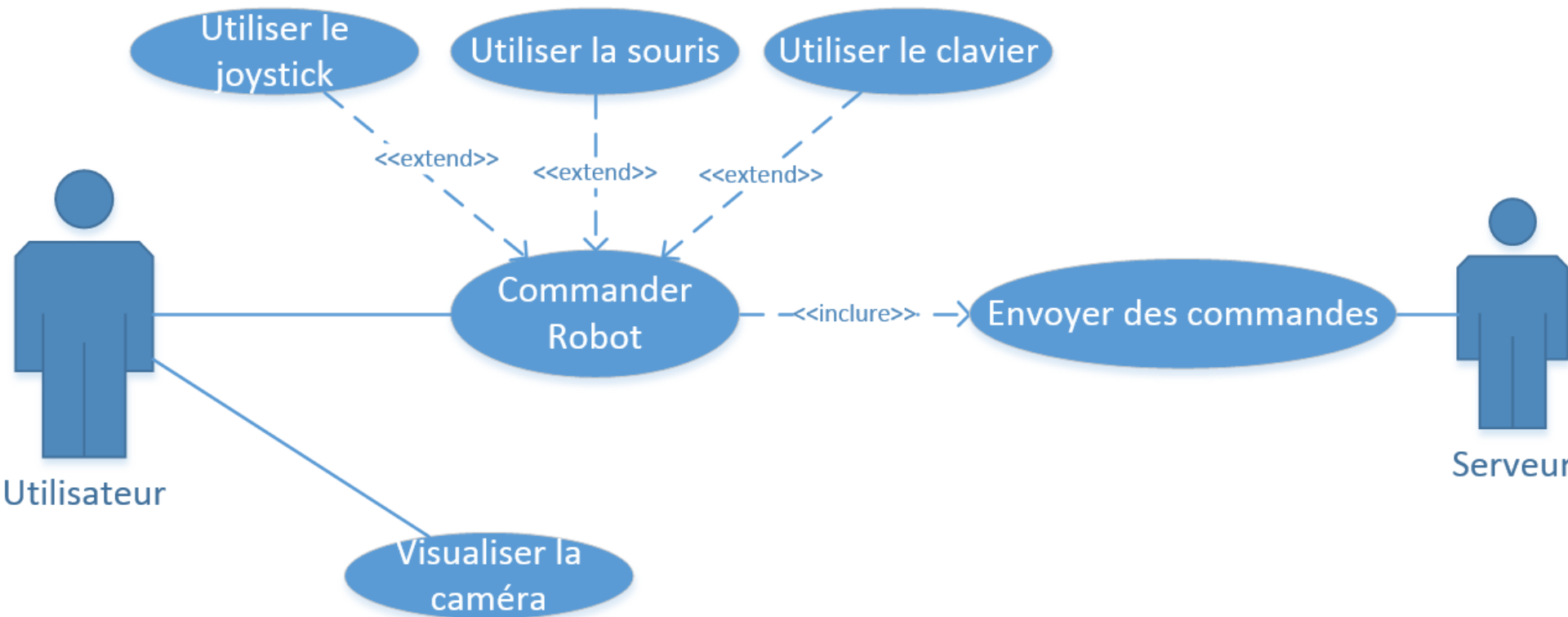
Présentation du projet

► Travail personnel demandé

- Modélisation UML du système à réaliser.
- Environnement et matériels de développement.
- Mettre au point une IHM pour contrôler le robot.
- Protocole de communication.
- Programmation d'un module client sur le pc pour le pilotage.
- Coder un module d'affichage du flux vidéo de la caméra.
- Coder un module d'utilisation d'un joystick.

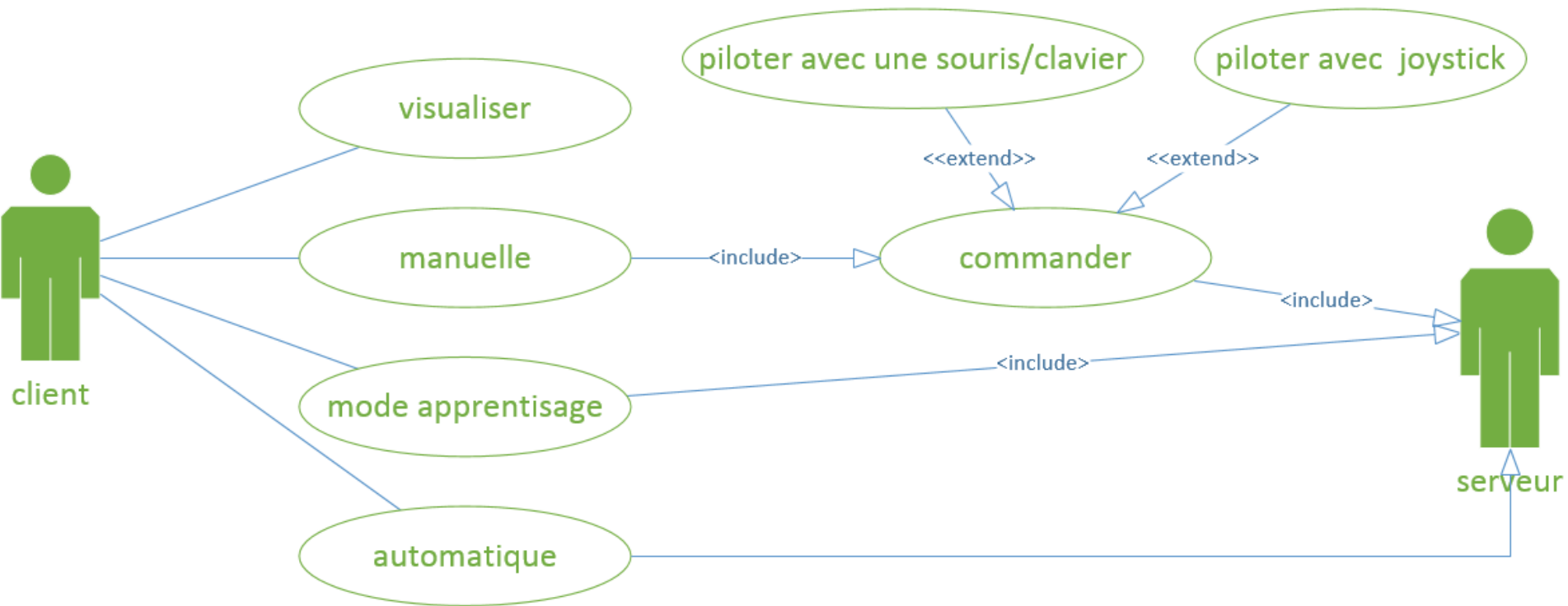
Analyse UML

► Diagramme des cas d'utilisation personnel



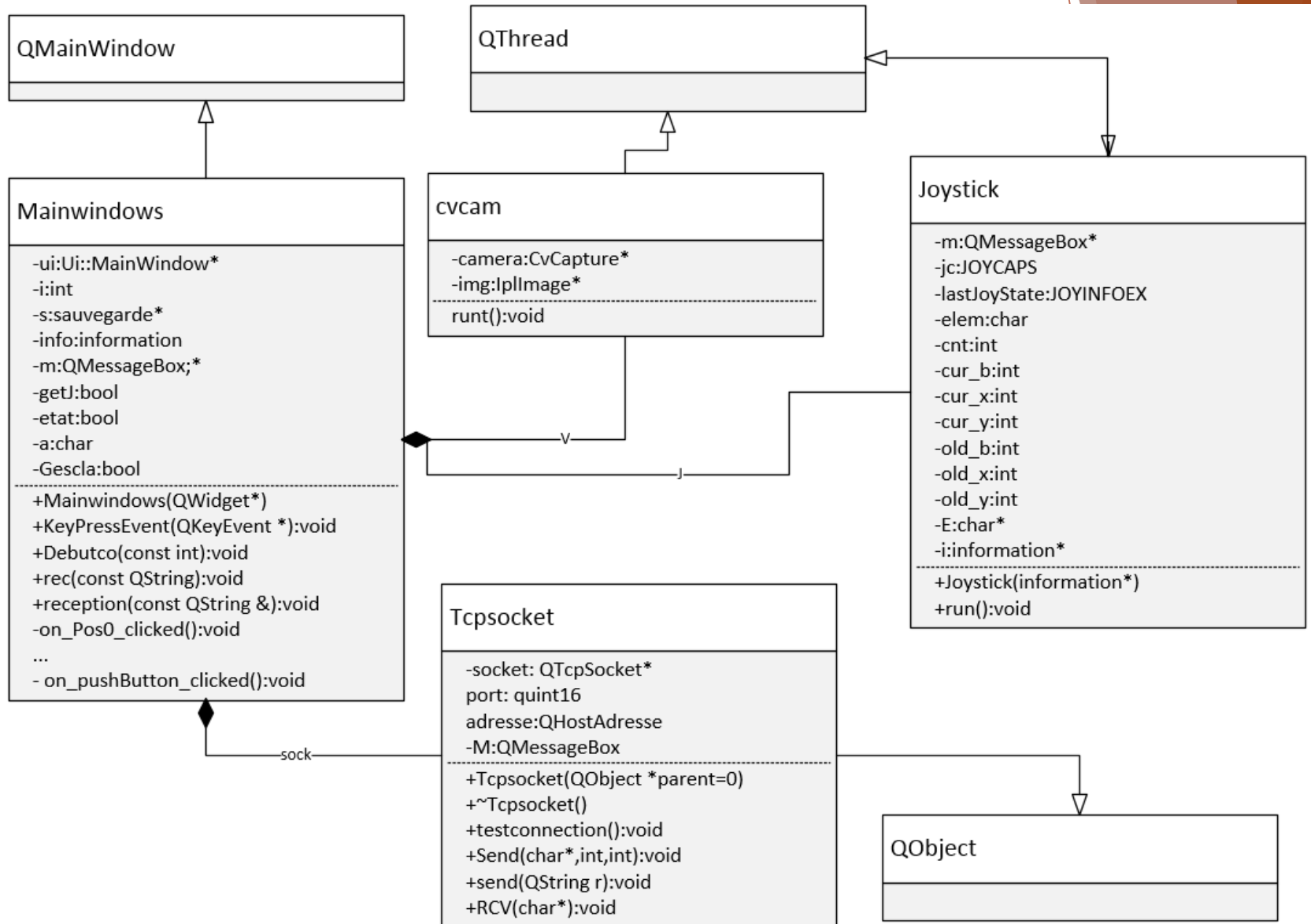
Analyse UML

► Diagramme des cas d'utilisation personnel



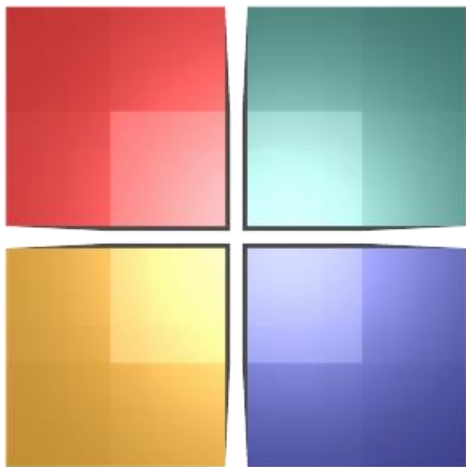
Analyse UML

Diagramme des classes



Environnement et matériels de développement

► Environnement utiliser



Environnement et matériels de développement

► Matériel utiliser



Mettre au point une IHM pour contrôler le robot

The image shows a graphical user interface (GUI) for controlling a robot. The interface is organized into several sections:

- Paramètre**: A section at the top for parameter settings.
- Connexion**: A section containing three buttons: **Connexion Serveur**, **Deconnexion Serveur**, and **Connexion Camera**.
- Choix du mode**: A section with two buttons: **Position 0** and **Position actuelle**.
- mode de controle**: A section with four radio buttons for selecting the control mode: **Joystick**, **Apprentissage**, **automatique**, and **Clavier**.
- Choix des elements**: A section with six radio buttons for selecting the robot's components: **Base**, **Epaule**, **Coude**, **Tangage**, **Roulis**, and **Prince**.
- Controle**: A section containing a **Vitesse** slider, a directional pad with **-** and **+** buttons, and two buttons labeled **Ouvrir pince** and **Fermer pince**.
- Controle clavier**: A section listing keyboard controls: **Pour Controler la Base B**, **Pour Controler l'Epaule E**, **Pour Controler le Coude C**, **Pour Controler la Tangage T**, **Pour Controler le Roulis R**, **Pour ouvrir ou fermer la pince P**, and **Pour se deplacer la touche + ou -**.

Mettre au point une IHM pour contrôler le robot



A screenshot of a graphical user interface (GUI) for configuring a robot control system. The interface is set against a light gray background with a fine grid of dots. It features three input fields for configuration: 'Adresse du serveur', 'Port du serveur', and 'Adresse de la caméra'. Each field is a white rectangle with a thin gray border. To the right of these fields are two buttons: 'Valider' and 'Quitter', both with gray borders and text. The labels for the input fields are in a bold, black, sans-serif font.

Adresse du serveur	<input type="text"/>
Port du serveur	<input type="text"/>
Adresse de la caméra	<input type="text"/>
	<input type="button" value="Valider"/> <input type="button" value="Quitter"/>

Protocole de communication.

- ▶ Mode de communication couche 4 TCP
- ▶ Master
- ▶ Android
- ▶ Auto
- ▶ FDS
- ▶ DDS
- ▶ FDS
- ▶ RAZ

Programmation d'un module client sur le pc pour le pilotage

- ▶ QTcpSocket
- ▶ network
- ▶ connectToHost(QHostAdress, quint16)
- ▶ waitForConnected(temps en msec)
- ▶ write(QbyteArray)
- ▶ Char *readAll()


Coder un module d'affichage du flux vidéo de la caméra.

- ▶ LIBS += : -L -lopencv -lopencv_world -lopencv_ffmpeg300
- ▶ INCLUDEPATH += \include
- ▶ #include ...\cv.h ...\highgui.h
- ▶ cvCaptureFromFile(adresse)
- ▶ cvNamedWindows(nf, CV_WINDOW_AUTOSIZE)

Coder un module d'affichage du flux vidéo de la caméra.

- ▶ `CvWaitKey(10)`
- ▶ `CvQueryFrame`
- ▶ `CvShowImage`

Coder un module d'affichage du flux vidéo de la caméra.



AXIS 210 Network Camera

Live View | Setup | Help

Basic Configuration

Instructions

1. Users

2. TCP/IP

3. Date & Time

4. Video & Image

Video & Image

Live View Config

Event Configuration

System Options

About

Basic TCP/IP Settings

IP Address Configuration

☐ Obtain IP address via DHCP

View

☒ Use the following IP address:

IP address:192.168.12.200

Subnet mask:255.255.255.0

Default router:192.168.0.1

Test

Services

Options for notification of IP address change

Settings...

Axis Internet Dynamic DNS service

Settings...

Save

Reset

See also the [advanced TCP/IP settings](#)

Coder un module d'utilisation d'un joystick.

- ▶ LIBS += -l winmm
- ▶ joyGetNumDevs()
- ▶ joyGetDevCaps()
- ▶ Windows.h

Coder un module d'utilisation d'un joystick.



Contrôle avec le clavier

- ▶ `KeyPressEvent(QKeyEvent*event)`
- ▶ `QKeyEvent -> text()`

Etat d'avancement du projet

fiche d'état des travaux - candidat 2

description de la tâche

Analyser et modéliser avec UML le système à réaliser

Etudier la documentation du robot.

Analyser et coder un module d'affichage du flux vidéo de la caméra.

Définir un protocole de communication.

Programmer un module client sur le pc "pilote" pour envoyer les ordres.

Programmer le module d'utilisation de la manette.

Analyser et coder l'IHM de pilotage du robot sur le pc "pilotage".

Finaliser le pilotage manuel du robot.

Analyser et coder le fonctionnement automatique du robot.

Intégrer les diverses parties du projet.

