BotCleaner

Documentationtechnique

**Enzo ROY**

CFPT - Informatique

TPI - 2019

Table des matières

[1 Introduction 2](#_Toc8048165)

[2 Rappel du cahier des charge 2](#_Toc8048166)

[2.1 Objectifs 2](#_Toc8048167)

[2.2 Spécifications 2](#_Toc8048168)

[2.3 Restrictions 2](#_Toc8048169)

[2.4 Environnement 2](#_Toc8048170)

[2.5 Organisation 2](#_Toc8048171)

[2.6 Livrables 2](#_Toc8048172)

[3 Analyse fonctionnelle 2](#_Toc8048173)

[3.1 Fonctionnalités 2](#_Toc8048174)

[3.2 Interfaces 2](#_Toc8048175)

[4 Analyse organique 2](#_Toc8048176)

[4.1 Architecture du code 2](#_Toc8048177)

[4.1.1 Arborescence de fichier 2](#_Toc8048178)

[4.1.2 Classe 2](#_Toc8048179)

[4.2 Outils 2](#_Toc8048180)

[5 Tests 2](#_Toc8048181)

[6 Conclusion 2](#_Toc8048182)

[7 Sources 2](#_Toc8048183)

[7.1 Codes repris 3](#_Toc8048184)

[7.2 Sites utilisés 3](#_Toc8048185)

[7.3 Aides reçues 3](#_Toc8048186)

[8 Planning 3](#_Toc8048187)

[8.1 Planning prévisionnel 3](#_Toc8048188)

[8.2 Planning Effectif 3](#_Toc8048189)

[9 Table des illustrations 3](#_Toc8048190)

[9.1 Figures 3](#_Toc8048191)

[9.2 Tableaux 3](#_Toc8048192)

[10 Annexes 3](#_Toc8048193)

# Introduction

Cette documentation a pour but de détailler les étapes nécessaires à la réalisation de l’application python/web « BotCleaner » sur Raspberry pi 3, réalisé dans le cadre du Travail Pratique Individuel (TPI).

L’application que je dois réaliser permet à un robot (AlphaBot2) de chercher les intrus sur la zone qui se trouve autour de lui via la caméra. Depuis un navigateur, un utilisateur peut visualiser en temps réel les essais du robot, de suivre son analyse d’image et les décisions qui en découlent. De plus, l’application permet de télécommander le robot tout en ayant un traitement d’image (cadre vert autour des intrus).

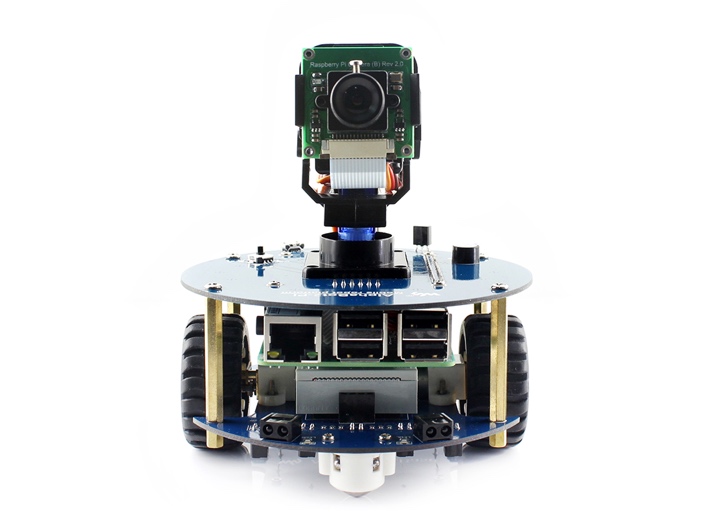


Figure 1 AlphaBot2 de waveshare

# Rappel du cahier des charges

## Objectifs

En 11 jours, réaliser une application python/web qui permet de télécommander un robot et de le voir chercher à nettoyer en repoussant tous objets se trouvant sur sa zone de travail, en utilisant uniquement sa caméra embarquée.

## Spécifications

Les points principaux de l’application sont les suivants :

* Le flux vidéo de la caméra est accessible par un connecteur TCP/IP
* L’analyseur d’image trace un cadre vert autour des intrus et donne la distance par rapport au centre de l’image.
* Une page web permet de télécommander le robot (formulaire web + script python)
* Une page web permet de visualiser le travail du robot

## Restriction

* L’application doit être réalisée en OOP selon le pattern MVC.

## Environnement

* PC : macOS Mojave (10.14.4)
* Raspberry pi 3 : raspbian (9.9)
* Alphabot2 version PI
* IDE : PyCharm (2018.3), PhpStorm (2018.3)
* Git client : GitKraken (5.0.4)
* Serveur: Flask (1.0.3)
* Bureautique : Office 360 (2019)
* Outil de prototypage graphique : Pencil (3.0.5)

## Organisation

Élève : Enzo Roy [enzo.r@eduge.ch](mailto:enzo.r@eduge.ch)

Maître : Pascal Bonvin [pascal.bonvin@edu.ge.ch](mailto:pascal.bonvin@edu.ge.ch)

Experts: Arnold Rullo arnold.rullo@etat.ge.ch

Jean Sottas jean.sottas@etat.ge.ch

## Livrables -> TODO

* Version électronique :
  + Résumé rapport TPI
  + Documentation technique
  + Documentation utilisateur
  + Code source (imprimable)
* Version papier :
  + Journal de bord

Tous les documents cités précédemment seront disponibles sur le GitHub :  
https://github.com/RoyEnzo/BotCleaner

# Analyse fonctionnelle

## Fonctionnalités

### Flux vidéo

Le flux vidéo de la caméra sera accessible par un connecteur TCP/IP.

### Analyse d’image

Les objets intrusifs dans la zone du robot seront entourés par un cadre vert et la distance par rapport au centre de l’image est calculée.

### Manette de contrôle

Une manette de contrôle (formulaire web) sera disponible pour télécommander le robot (avancer/reculer, rotation droite/gauche).

### Contrôleur moteur

Lors du mode automatique, un contrôleur décide de la rotation des deux moteurs selon les résultats de l’analyse d’image.

## Interfaces

### Mode automatique



Figure 2 Page web - Mode automatique

### Mode manuel -> TODO refaire changement !

Une image contenant capture d’écran

Description générée automatiquement

Figure 3 Page Web - Mode manuel

## Cas d’utilisation

# Analyse organique

## Architecture du code

### Arborescence de fichiers

### Classe

## Outils

# Bug

## Pi camera : out of ressources

### Description :

Lorsque je lançais un script python sur le Raspberry pi, une erreur « pi camera : out of ressources » se déclenchait. Cela m’indique que la pi camera est déjà utilisée par un autre script. Mais à ma connaissance, tous les scripts que j’avais lancés précédemment étaient arrêtés.

Sauf que des scripts Python étaient toujours en fonctionnement dans les processus actifs.

J’ai quand même vérifié que la caméra fonctionne avec la commande « raspistill »

### Solution :

L’un des outils de PyCharm permet d’avoir un interpréteur distant via SSH (Secure Shell). Je lance les scripts présents sur le Raspberry pi avec l’outil cité précédemment. Lorsque le programme se ferme, il interrompt le script lancé par SSH. Mais à cause une erreur de manipulation ou un glitch, le script a continué de tourner sur le Raspberry pi.

Il faut donc enlever l’interpréteur distant sur PyCharm, redémarrer les deux machines. Une fois rallumé, il faut vérifier que PyCharm n’a pas relancé le script en arrière-plan sur le Raspberry pi.

# Tests

# Conclusion

## Bilan personnel

# Glossaire

# Sources

## Codes repris

Pour mettre en place le flux vidéo, j’ai utilisé des parties de code présentent dans les projets suivants :

https://blog.miguelgrinberg.com/post/video-streaming-with-flask  
https://raw.githubusercontent.com/RuiSantosdotme/Random-Nerd-Tutorials/master/Projects/rpi\_camera\_surveillance\_system.py

## Sites utilisés

<https://sourceforge.net/projects/mjpg-streamer/>

<http://www.magdiblog.fr/divers/raspberry-pi-camera-5-facons-de-faire-du-streaming/>

<https://stackoverflow.com/questions/42601478/flask-calling-python-function-on-button-onclick-event>

<https://www.digitalocean.com/community/questions/running-mutliple-flask-application>

<https://openclassrooms.com/fr/courses/1654786-creez-vos-applications-web-avec-flask/1655538-tp-service-web-dupload-dimages>

<https://stackoverflow.com/questions/7391945/how-do-i-read-image-data-from-a-url-in-python>

## Aides reçues

M. Bonvin Notamment lors de l’erreur avec la caméra

# Planning

## Planning prévisionnel



Tableau 1 Planning prévisionnel

## Planning effectif

# Table des illustrations

## Figures

[Figure 1 AlphaBot2 de waveshare 2](#_Toc8716832)

[Figure 2 Page web - Mode automatique 4](#_Toc8716833)

[Figure 3 Page Web - Mode manuel 5](#_Toc8716834)

## Tableaux

[Tableau 1 Planning prévisionnel 7](#_Toc8716843)

# Annexes

EnzoRoy\_Planning\_Taches.xlsx Planning prévisionnel/effectif, tâches effectuées  
EnzoRoy\_documentation\_utilisateur.docx Documentation pour la mise en place du robot