07/05/2019

Quentin FORESTIER

CFPT – Informatique

TPI 2018 – 2019

Pascal Bonvin

RedBallBot2

Documentation Technique

Table des matières

[1 !!!!!!!!!!!! Résumé du rapport !!!!!!!!!! 2](#_Toc8308363)

[1.1 Situation de départ 2](#_Toc8308364)

[1.2 Mise en œuvre 2](#_Toc8308365)

[1.3 Résultats 2](#_Toc8308366)

[2 Introduction 2](#_Toc8308367)

[3 Rappel du cahier des charges 3](#_Toc8308368)

[3.1 Objectifs 3](#_Toc8308369)

[3.2 Spécifications 3](#_Toc8308370)

[3.3 Environnement 3](#_Toc8308371)

[3.4 Livrables 3](#_Toc8308372)

[4 Analyse fonctionnelle 4](#_Toc8308373)

[4.1 Fonctionnalités 4](#_Toc8308374)

[4.1.1 Streaming vidéo 4](#_Toc8308375)

[4.1.2 Analyse d’image 4](#_Toc8308376)

[4.1.3 Déplacement du robot 4](#_Toc8308377)

[4.1.4 Manette de contrôle 4](#_Toc8308378)

[4.2 Interfaces 5](#_Toc8308379)

[4.3 Cas d’utilisation 5](#_Toc8308380)

[4.4 Planning prévisionnel 5](#_Toc8308381)

[5 Analyse organique 5](#_Toc8308382)

[5.1 Architecture du code 5](#_Toc8308383)

[5.1.1 Arborescence des fichiers 5](#_Toc8308384)

[5.1.2 Diagrammes de classes 5](#_Toc8308385)

[5.2 Outils 5](#_Toc8308386)

[6 Tests 5](#_Toc8308387)

[7 Conclusion 6](#_Toc8308388)

[7.1 Bilan Personnel 6](#_Toc8308389)

[8 Planning 6](#_Toc8308390)

[8.1 Planning prévisionnel 6](#_Toc8308391)

[8.2 Planning effectif 6](#_Toc8308392)

[9 Glossaire 6](#_Toc8308393)

[10 Bibliographie 6](#_Toc8308394)

[10.1 Codes repris 6](#_Toc8308395)

[10.2 Sites utilisés 6](#_Toc8308396)

[10.3 Aides reçues 6](#_Toc8308397)

[11 Table des illustrations 6](#_Toc8308398)

[11.1 Figures 6](#_Toc8308399)

[11.2 Tableaux 6](#_Toc8308400)

[12 Annexe 6](#_Toc8308401)

# !!!!!!!!!!!! Résumé du rapport !!!!!!!!!!

## Situation de départ

## Mise en œuvre

## Résultats

# Introduction

Cette documentation permet d’obtenir des détails sur le projet « RedBallBot », réalisé dans le cadre du TPI (Travail Pratique Individuel).

L’objectif est de déplacer un robot pour que celui-ci se colle à une balle rouge. L’utilisateur pourra cependant prendre la main et le déplacer à l’aide de boutons disponible sur une page Web. Le mode automatique permettra au robot d’analyser son environnement afin de trouver la balle rouge et de se déplacer vers elle.

# Rappel du cahier des charges

## Objectifs

Le but du projet est de réalisé une application Python / Web qui permet de télécommander un robot et de le voir chercher une balle rouge en utilisant uniquement sa caméra embarquée à travers une page / application Web.

## Spécifications

L’application permet de :

* Télécommander un AlphaBot2 au travers d’une page / application Web
* Obtenir le flux vidéo de la caméra sur une page Web
* De savoir ou le robot a analysé la balle grâce à un cadre vert
* Connaitre la distance entre le centre de l’image et le centre de la balle

## Environnement

J’ai utilisé un PC Windows 10 afin de réaliser ce projet. Pour la partie Python, j’ai utilisé PyCharm 2018.3.5, avec la fonctionnalité de transfert de fichier SSH.

Pour la partie de gestion de version, j’ai utilisé Git avec un répertoire en ligne GitHub, et je fais des sauvegardes récurrentes sur Google Drive.

Le serveur http disponible sur le Raspberry Pi est un serveur Flask. Il me permet de rester uniquement en python pour toute la partie traitement.

Pour les outils de bureautique, j’ai utilisé Word et Excel de la suite Office.

## Livrables

Plusieurs documents seront disponibles :

* La documentation technique
* Le manuel utilisateur
* Le code source
* Un journal de bord
* Un répertoire GitHub comprenant tous les documents cité ci-dessus

## Contraintes

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

# Analyse fonctionnelle

## Fonctionnalités

### Streaming vidéo

Le flux vidéo sera disponible par le biais d’un connecteur TCP/IP. La seule restriction est d’être connecté au même réseau que le robot. Un carré vert sera ajouté autour de la balle lorsque celle-ci sera analysée.

### Analyse d’image

L’analyse d’image se fera, en premier lieu, sur le robot. Cependant, il sera possible de lancer le script d’analyse d’image sur un ordinateur externe afin que le processus soit plus rapide.

### Déplacement du robot

Le robot a 2 modes de fonctionnement. Le mode manuel, qui permet à l’utilisateur de contrôler le robot par l’intermédiaire d’une page WEB. Le mode automatique, qui se dirige vers une balle rouge en fonction de l’image analysé, soit par le robot, soit par un ordinateur distant.

### Manette de contrôle

La manette de contrôle se présentera sous forme de page WEB contenant des entrées. Entrées qui permette à l’utilisateur de déplacer le robot, de changer la vitesse, ou de changer son mode de déplacement.

## Interfaces

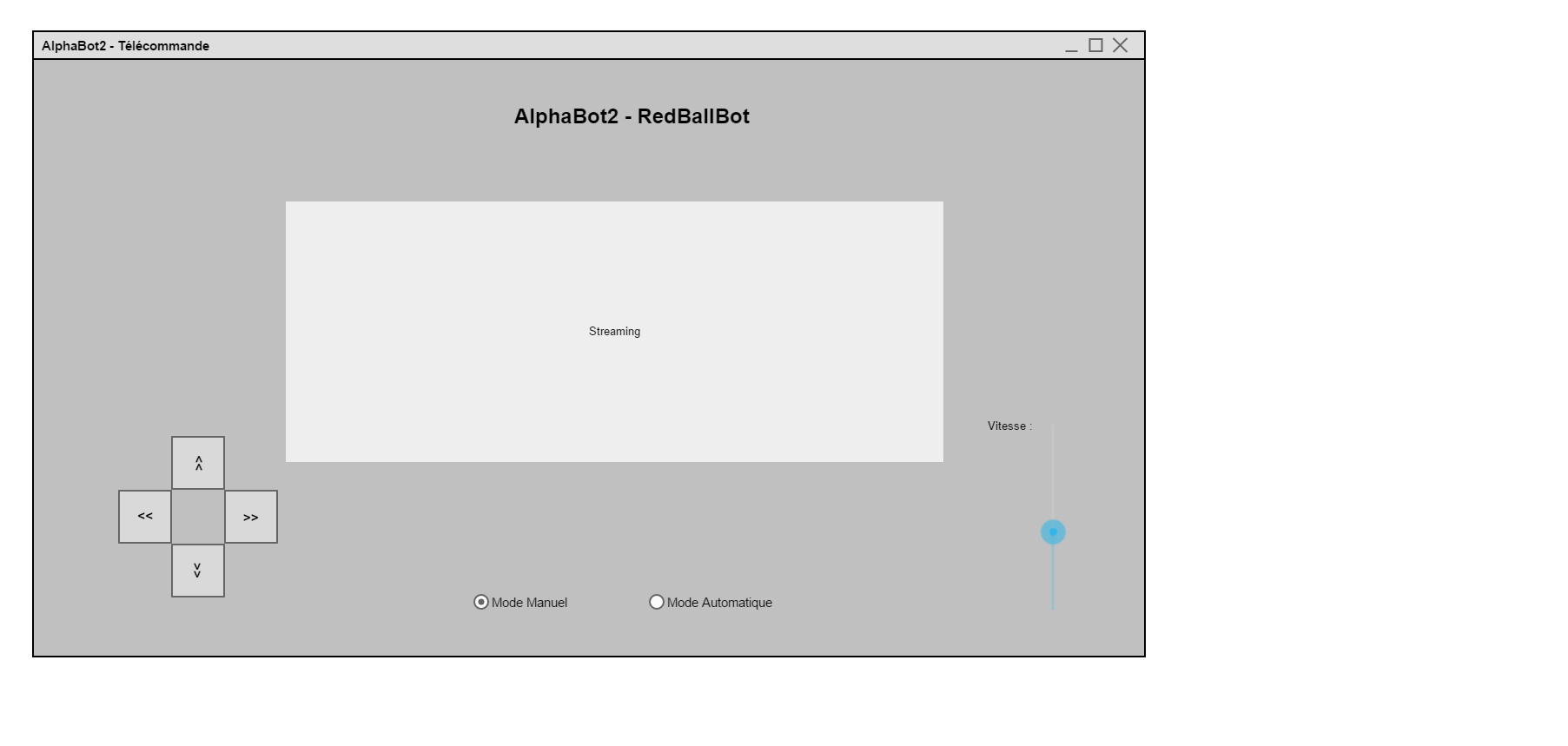


Figure 1 : Page principal du projet

Cependant, j’ai décidé par la suite d’ajouter un autre slider pour compenser la différence des moteurs.

## Cas d’utilisation

## Planning prévisionnel



# Analyse organique

## Architecture du code

### Arborescence des fichiers

### Diagrammes de classes

## Outils

# Tests

# Conclusion

## Bilan Personnel

# Planning

## Planning prévisionnel

## Planning effectif

# Glossaire

# Bibliographie

## Codes repris

## Sites utilisés

<https://blog.miguelgrinberg.com/post/video-streaming-with-flask>

<https://randomnerdtutorials.com/video-streaming-with-raspberry-pi-camera/$>

<https://raspbian-france.fr/installer-serveur-web-raspberry-lamp/>

## Aides reçues

# Table des illustrations

## Figures

## Tableaux

# Annexe