



UFR SEN  
Faculté des Sciences  
Exactes et Naturelles



**Aymerick LAURETTA-PERONNE**

*Université des Antilles  
Équipe Biologie de la Mangrove*

---

**Rapport de stage de 3ème année de Licence Informatique**

*Réalisation d'un dispositif de capture vidéo pour l'acquisition de données dans le cadre d'une  
manipulation en biologie.*

---

Enseignant référent :  
Wilfried SEGRETIER

Tuteur de stage :  
Manuel CLERGUE  
Co-tuteur :  
Olivier GROS

---

**27 février 2022**



## **Remerciements**

Je tiens à remercier tous ceux qui ont contribué à la réalisation de ce rapport de stage.

Tout d'abord, je à exprimer ma gratitude à Monsieur Manuel CLERGUE de m'avoir  
accepté et accompagné durant mon stage.

J'adresse mes remerciements à Monsieur Olivier GROS pour l'accueil chaleureux au sein  
de l'équipe et de m'avoir fait partager ses connaissances et son expérience dans le  
domaine de la biologie.

Merci à tous les membres de l'équipe de Biologie de la Mangrove pour leur aide et leur  
soutien.

## Résumé

Dans ce rapport sont présentées les différentes étapes de conception du dispositif de capture vidéo et de la réalisation de l'application pour les biologistes du laboratoire ISYEB. Ce dispositif a pour but de remplacer celui existant du laboratoire ISYEB en offrant une meilleure polyvalence aux biologistes. L'application quant à elle doit permettre aux biologistes de réaliser différents types de prises de vue de manière plus efficace et autonome en fonction de leurs besoins. De ce fait, l'application doit permettre aux biologistes de lancer un enregistrement en ayant la possibilité de choisir l'emplacement et le nom du fichier vidéo enregistré. Permettre aussi la prise de photo en ayant la possibilité tout comme l'enregistrement vidéo de choisir l'emplacement et le nom du fichier photo enregistré. Ce dispositif doit permettre aux biologistes de récupérer les différents enregistrements via différents types de périphériques externes de stockage en ayant également une possibilité de les visualiser depuis le dispositif.

## Abstract

In this report are presented the different stages of design of the video capture device and the realization of the application for the biologists of the ISYEB laboratory. The purpose of this device is to replace the existing one in the ISYEB laboratory by offering better versatility to biologists. As for the application, it should enable biologists to take different types of shots more efficiently and autonomously according to their needs. Therefore, the application must allow biologists to launch a recording by having the possibility of choosing the location and the name of the recorded video file. Also allow photo taking with the possibility, like video recording, of choosing the location and name of the saved photo file. This device must allow biologists to retrieve the different recordings via different types of external storage devices while also having the possibility of viewing them from the device.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Environnement</b>	<b>6</b>
2.1	Présentation de la structure . . . . .	6
2.2	Organisation du stage . . . . .	7
<b>3</b>	<b>Présentation de la problématique</b>	<b>9</b>
3.1	Présentation du sujet . . . . .	9
3.2	Présentation de l'aspect biologiste . . . . .	9
3.3	Objectif . . . . .	10
<b>4</b>	<b>Travail réalisé</b>	<b>12</b>
4.1	Mon travail . . . . .	13
4.2	Réception du matériel . . . . .	13
4.3	Montage, Installation et Configuration . . . . .	15
4.4	Choix des technologies . . . . .	17
4.5	Réalisation de l'application . . . . .	18
<b>5</b>	<b>Conclusion</b>	<b>26</b>
5.1	Ressentis par rapport au stage . . . . .	26
5.2	Perspectives futures . . . . .	26

# Chapitre 1

## Introduction

Dans le cadre de ma dernière année de licence informatique à l'université des Antilles de Guadeloupe, je dois effectuer un stage d'une durée de 24 jours. Ce stage vise à clôturer mon cursus universitaire. Il me permet de mettre en pratique mes acquis en informatique et de me familiariser avec la vie professionnelle.

Le projet avec le laboratoire de biologie concerne l'analyse de déplacements d'insectes afin de déterminer leur préférence sur des zones marquées par des odeurs, en environnement contrôlé (aquarium en laboratoire).

La mission est de configurer un système d'acquisition vidéo à base d'un Raspberry Pi, et de développer une interface intuitive afin de piloter cette acquisition de manière à ce qu'elle puisse être conduite par un biologiste.

Comment pourrons nous concevoir un dispositif de prise de vue qui soit performant et avec une application qui soit facile à utiliser? En m'aidant de mon expérience en informatique et de ma recherche scientifique, nous allons essayer de mettre œuvre un dispositif de prise de vue qui soit performant et avec une application dédiée à la prise de vue.

Dans ce rapport, je présente mon environnement de travail (chapitre 2), ainsi que la mission principale que j'ai été amené à accomplir au sein du laboratoire ISYEB (chapitre 3), à savoir, la conception d'un dispositif de prise de vue, ainsi que la réalisation d'une application permettant l'acquisition de prise de vue (chapitre 4). En effet, le laboratoire ISYEB possède déjà un dispositif permettant les prises de vue, mais il s'agit d'un dispositif provisoire que les biologistes ont décidé de remplacer.

# Chapitre 2

## Environnement

### 2.1 Présentation de la structure

#### 2.1.1 L'université des Antilles

L'université des Antilles est une université pluridisciplinaire implantée sur deux régions, Guadeloupe et Martinique née de la scission de l'université des Antilles et de la Guyane (UAG) en 2014, en université de Guyane, d'une part et en université des Antilles, d'autre part.

Elle comprend l'une des 204 écoles d'ingénieurs françaises accréditées au 1er septembre 2020 à délivrer un diplôme d'ingénieur.

#### 2.1.2 L'UFR SEN

L'Unité de Formation et de Recherche (UFR) en Sciences Exactes et Naturelles (SEN), communément appelée UFR SEN, compte près de 1 800 étudiants, 110 enseignants et enseignants chercheurs, 32 personnels BIATSS, et a la particularité d'être la composante de l'Université des Antilles qui porte le plus de diplômes de formation (15) et de structures de recherche (9 sur les 25 que compte toute l'université). Ses domaines de recherche et de formation couvrent les six pôles thématiques de l'Université des Antilles : Risques et Énergies, Numérique, Mer et Océan, Biodiversité en milieu tropical insulaire, Santé insulaire en environnement tropical, Dynamique des sociétés et territoires Caraïbes.

Les équipes de recherche portent à elles seules près de 70% de l'ensemble des projets de recherche réalisés à l'Université des Antilles. Des impacts environnementaux des sargasses,

à l'étude de la durabilité des matériaux, en passant par les risques naturels majeurs et les transitions énergétiques, climatiques et écologiques, la Faculté des Sciences est porteuse de projets innovants.

### 2.1.3 L'Equipe de la Biologie de la Mangrove

L'équipe **Biologie de la Mangrove** fait partie intégrante de l'UMR 7205 MNHN CNRS-Sorbonne Université-UA «Institut de Systématique, Evolution, Biodiversité» dirigée par Philippe Grandcolas.

Elle représente l'une des 19 équipes constituant actuellement cette UMR qui est répartie sur 2 sites (Paris et Guadeloupe). L'équipe **Biologie de la Mangrove** est composée exclusivement de personnels de l'Université des Antilles et est localisée en Guadeloupe sur le campus de Fouillole.

L'équipe de la Biologie de la Mangrove intègre l'UMR 7205 ISYEB en janvier 2019 en proposant d'étudier la biologie et les adaptations évolutives (par le biais de la symbiose essentiellement) de modèles littoraux côtiers tropicaux évoluant au sein d'écosystèmes extrêmes (forte teneur en composés soufrés réduits comme la mangrove et les herbiers à phanérogames marines) faciles d'accès.

Mon tuteur en entreprise durant le stage est Monsieur Olivier GROS responsable de l'équipe Biologie de la Mangrove et Professeur des Universités. Les étapes de développement de l'application est validée par Monsieur Manuel CLERGUE Maître de Conférences à l'Université des Antilles. Les membres de l'équipe projet sont Monsieur Mathieu BONNEAU intervenant Mathématique et Madame SUZANNE doctorante et la stagiaire en Biologie qui aura la gestion des expériences.

## 2.2 Organisation du stage

Tout au long du stage, des réunions régulières ont été organisées avec mon encadrant, afin de faire le point sur l'avancement du projet.

Le travail a été réparti en trois axes :

- **Partie Informatique** : La partie centrale du stage comportant le montage et l'installation du dispositif de capture vidéo ainsi que la mise en place de



l'application.

- **Partie en laboratoire** : La partie secondaire comportant l'installation et l'ajustement du dispositif sur l'équipement de laboratoire.
- **Partie sur le terrain** : Immersion sur le terrain afin de découvrir les différents aspects de l'environnement et de la biodiversité de la mangrove de la Guadeloupe.

### 2.2.1 Ressources fournies

Plusieurs ressources m'ont été fournies dans le cadre de mon stage. Dans un premier temps, j'ai reçu le matériel permettant l'acquisition vidéo qui sera abordé dans la section 4.2. Une connexion filaire au réseau de l'Université des Antilles a été fournie me permettant d'installer les différents logiciels nécessaires à l'installation du dispositif ainsi que les mises à jour.

### 2.2.2 Contraintes particulière

Durant ce stage j'ai eu deux contraintes particulières :

- **Premièrement** : Premièrement, le matériel a été commandé avant le début de mon stage. de ce fait je n'ai pas pu lister avec précision le matériel optimal pour réaliser le projet. A cela s'ajoute différentes difficultés techniques.
- **Deuxièmement** : Il y avait un calendrier strict à suivre. Un système minimal devait avoir vu le jour 10 jours après le début du stage.

Dans la section suivante, nous allons présenter la problématique, notamment le sujet ainsi que l'objectif du stage.

# Chapitre 3

## Présentation de la problématique

### 3.1 Présentation du sujet

Le projet avec le laboratoire *Biologie* de l'*Université des Antilles*, concerne l'analyse de déplacement de **Gerridés**.

Le but étant de déterminer leur préférence entre différentes zones marquées par des odeurs, en environnement contrôlé.

### 3.2 Présentation de l'aspect biologiste

Les **Gerridés** sont une famille d'insectes de l'ordre des Hémiptères et du sous-ordre des Hétéroptères c'est-à-dire des punaises.



FIGURE 3.1 – Gérridés

Les membres de cette famille sont communément appelés araignées d'eau, mais ce sont des insectes, on ne peut donc pas parler d'araignées. Cette appellation vient sans doute du fait de leurs longues pattes. Leur capacité à se déplacer sur l'eau leur vaut aussi le nom de patineurs de l'eau.

Dans un environnement contrôlé, on place deux zones avec différentes odeurs (aliments) et on observe où les **Gerridés** préfèrent se déplacer. Afin de faciliter l'enregistrement de l'expérience, il est nécessaire d'avoir un système adéquat de capture vidéo.

### 3.2.1 Positionnement du travail dans le projet

Les biologistes accrochaient une GoPro au-dessus du bac et ensuite devaient la retirer afin de pouvoir récupérer les données enregistrées. Ce dispositif était fonctionnel et leur permettait de réaliser des captures vidéo mais il leur fallait placer le bac dans le champ de vision de la GoPro à chaque fois qu'ils voulaient faire une capture.

Il fallait aussi que la GoPro soit connectée à un ordinateur afin de pouvoir récupérer les données. Ajouter à cela le démarrage et l'arrêt de l'enregistrement de manière manuelle entraînant régulièrement le déplacement du bac (avec les mouvements). La disposition de la GoPro ne permettait de bien voir la prévisualisation de la capture vidéo. Les biologistes ont donc souhaité d'améliorer leur dispositif afin de pouvoir réaliser l'expérience de manière plus pratique.

Ce qui nous permet d'enchaîner sur l'objectif qui en découlent : comment faire en sorte que les biologistes puissent réaliser l'expérience de manière plus pratique ?

## 3.3 Objectif

L'objectif du stage est de réaliser l'installation et de configurer un dispositif de capture vidéo réalisé par un Raspberry Pi. Il faudra aussi développer une application conviviale pour la gestion des vidéos. Il sera donc question de permettre au biologiste de réaliser des captures vidéo de manière plus pratique. Plutôt que d'avoir un dispositif mobile comme auparavant, nous réaliserons un dispositif qui sera fixe.

Le dispositif doit répondre à deux besoins principaux :

- **1** : La fixité du dispositif.
- **2** : Une interface simple.

Le but de cette application est de permettre aux biologistes de réaliser des captures vidéo enregistrées directement sur la carte SD de la Raspberry Pi. Il sera également

possible de visualiser les capture vidéo enregistrées sur la carte SD du Raspberry Pi et si besoin de les récupérer à l'aide d'une clé USB ou un disque dur externe sans problème. Ce dispositif sera installé sur une potence au-dessus du bac et restera fixe en permanence permettant de limiter voire supprimer les divers déplacements.

Voyons maintenant comment nous allons nous y prendre pour réaliser ce dispositif ainsi que l'application dédiée.

# Chapitre 4

## Travail réalisé

### Sommaire

---

<b>4.1</b>	<b>Mon travail . . . . .</b>	<b>13</b>
<b>4.2</b>	<b>Réception du matériel . . . . .</b>	<b>13</b>
4.2.1	Raspberry Pi . . . . .	13
4.2.2	Ecran LCD (Raspberry Pi) . . . . .	14
4.2.3	Module de capture vidéo (Raspberry Pi) V2 . . . . .	14
<b>4.3</b>	<b>Montage, Installation et Configuration . . . . .</b>	<b>15</b>
4.3.1	Montage du matériel . . . . .	15
4.3.2	Installation du matériel . . . . .	15
4.3.3	Configuration du matériel . . . . .	16
<b>4.4</b>	<b>Choix des technologies . . . . .</b>	<b>17</b>
<b>4.5</b>	<b>Réalisation de l'application . . . . .</b>	<b>18</b>
4.5.1	Création de l'interface graphique . . . . .	19
4.5.2	Fonction Start Recording . . . . .	20
4.5.3	Prendre une capture . . . . .	21
4.5.4	Travail en laboratoire . . . . .	23
4.5.5	Travail sur la caméra . . . . .	24
4.5.6	Travail sur vidéo . . . . .	24

---

Avant de commencer nous allons introduire le travail à réaliser. Dans un premier temps, nous allons présenter les différents matériels utilisés afin de concevoir le dispositif de prise de vue. Par la suite, le déroulement du montage, ainsi que de l'installation et la

configuration du dispositif. Une fois cela fait, nous allons réaliser l'application permettant l'interaction entre le dispositif et l'utilisateur afin de pouvoir effectuer la prise de vue.

## 4.1 Mon travail

Ma mission est de réaliser l'installation et de configurer un dispositif de capture vidéo réalisé par un *Raspberry Pi* ainsi que de développer une application conviviale pour la gestion des vidéos.

## 4.2 Réception du matériel

### 4.2.1 Raspberry Pi

Le *Raspberry Pi* est un ordinateur portable de petite taille, doté d'un processeur ARM<sup>1</sup> et d'un système d'exploitation Linux.

- **Intérêt** : Le principal intérêt de ce type de matériel est son coût qui est très faible. Comme le montre le tableau ci-dessous.

Nom	Processeur	RAM	Prix
Raspberry Pi 2w	1.5 GHz Quad-Core A72	512Mb	15€
Raspberry Pi 3	1.4 GHz Quad-Core A53	1Gb	35€
Raspberry Pi 4	1.8 GHz Quad-Core A72	2Gb	45€

Le Raspberry Pi est également simple à manipuler. Néanmoins il possède certains inconvénients.

- **Inconvénients** : En vue de sa petite taille, il est très fragile mais également très peu performant. Sa configuration n'est pas toujours simple à réaliser. Nous pouvons toujours augmenter les performances du Raspberry Pi via à de l'Overclocking<sup>2</sup> mais il faudrait prévoir un système de refroidissement.



FIGURE 4.1 – Raspberry Pi 4

### 4.2.2 Ecran LCD (Raspberry Pi)

Un écran LCD de 7 pouces est branché sur le Raspberry Pi et permet d'interagir avec l'ordinateur.

L'intérêt majeur de ce type de périphérique est sa finesse. En effet, dans un laboratoire il n'est pas toujours facile d'avoir un grand écran. De ce fait avoir un écran de petite taille qu'on peut tenir dans nos mains tout en ayant la possibilité d'utiliser l'écran tactile pour interagir avec l'ordinateur nous permettant de nous dispenser de clavier et de souris.



FIGURE 4.2 – Ecran LCD de 7 pouces

### 4.2.3 Module de capture vidéo (Raspberry Pi) V2

Le module de capture vidéo (Raspberry Pi) V2 est un module de captation vidéo qui permet de capturer des images et des vidéos.

- **Intérêt** : Le principal intérêt de ce module est son coût qui est très faible. De plus tout comme le Raspberry Pi, il est très facile à manipuler.
- **Inconvénients** : la qualité de ce module caméra est relativement mauvaise, sa configuration et son utilisation ne sont pas toujours évidentes. De plus il est relativement fragile.



FIGURE 4.3 – Module de capture vidéo (Raspberry Pi) V2

## 4.3 Montage, Installation et Configuration

Il faut savoir que le Raspberry Pi et son module caméra arrive séparément ce qui signifie que le module n'est pas monté sur le Raspberry Pi. Le Raspberry Pi arrive sans son système d'exploitation préinstallé.

### 4.3.1 Montage du matériel

Le Raspberry Pi possède un port CSI 'Camera Serial Interface' (signifiant en anglais interface série pour caméra, CSI) est un standard d'interface électronique entre une caméra (un capteur ou une source vidéo) et un microprocesseur.

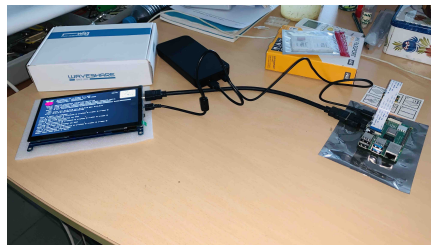


FIGURE 4.4 – Matériel assemblé et prêt à l'emploi

Afin de pouvoir utiliser le module caméra il nous faut donc insérer la nappe de connecteur de la caméra dans le port CSI du Raspberry Pi.

### 4.3.2 Installation du matériel

Il nous faut installer le système d'exploitation (OS<sup>3</sup>). Pour ce faire nous allons utiliser Raspberry Pi Imager, logiciel développé par Raspberry Pi Foundation nous permettant de télécharger et installer l'OS Raspbian. Raspbian est l'OS original ainsi que le plus connu, libre et gratuit basé sur Debian, il est optimisé pour fonctionner sur les différents Raspberry Pi.



Raspberry Pi Imager est le moyen rapide et facile d'installer Raspberry Pi OS (Raspbian) sur une microSD.

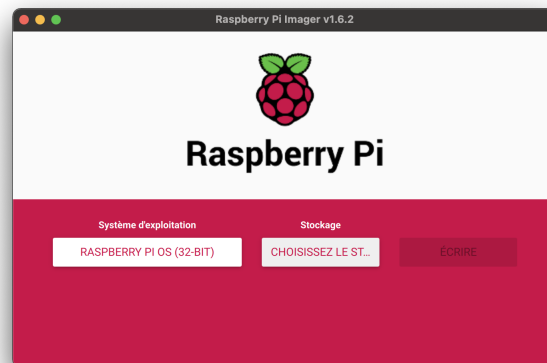


FIGURE 4.5 – Interface "Raspberry Pi Imager"

Nous obtenons alors l'aperçu de la figure 4.5. Comme nous pouvons le voir sur cette figure, le logiciel nous demande de choisir l'OS que nous souhaitons installer. On lui indique donc que nous voulons installer l'OS Raspbian en 32 bits. Et on choisit notre stockage qui dans notre cas est une microSD.

Une fois cela fait, on lance l'écriture, le logiciel va télécharger l'OS et l'installer sur la microSD.

### 4.3.3 Configuration du matériel

Avant de pouvoir utiliser le module caméra, activer l'interface caméra. Pour ce faire nous allons utiliser la commande **sudo raspi-config**.

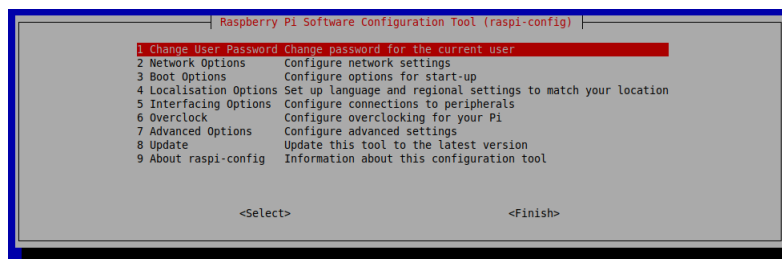


FIGURE 4.6 – Interface "raspi-config"

Une fois dans **Raspi-config**, nous allons choisir la section **Interface section**. On sélectionne **P1 Camera** et on met sur **Enable**.

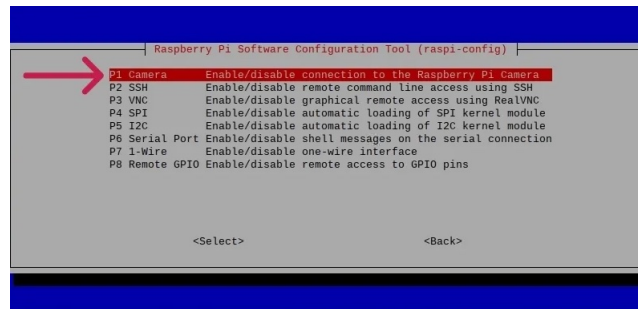


FIGURE 4.7 – camera option (raspi-config)

Une fois cela fait on peut quitter **Raspi-config** et tester si la caméra est bien activée et fonctionne correctement.

Pour ce faire, on ouvre le terminal et on tape la commande suivante :

```
raspistill -o Desktop/image.jpg
```

Cette commande nous permet de prendre une photo et de la sauvegarder dans le dossier Desktop. Si photo est bien prise on peut en deduire que la caméra est opérationnelle. L'opération suivant consiste à essayer de faire une vidéo.

```
raspivid -o Desktop/video.h264
```

Cette commande nous permet de prendre une vidéo et de la sauvegarder dans le dossier Desktop, la vidéo est bien prise.

## 4.4 Choix des technologies

Pour le choix du langage de programmation, Nous nous sommes basé sur les langages de programmation les plus utilisés avec le système d'exploitation Raspbian. Lors de ma recherche, 5 langages de programmation sont sortis :

- Python
- C
- Java/BlueJ
- Perl
- Scratch

Nous constatons que le langage Python est le plus utilisé pour le système d'exploitation Raspbian. Ce qui est normal, car le Pi dans "Raspberry Pi" signifie "Python". Il a donc le mérite d'être le choix par défaut.

Afin de pouvoir nous décider sur le langage de programmation, il me faut chercher la compatibilité des API<sup>5</sup> pour la camera. Il s'avère que l'API PiCamera est installée par défaut sur le système d'exploitation Raspbian. Puisque le langage Python et l'API PiCamera sont compatibles, et installé par défaut sur le système d'exploitation Raspbian, il est possible de choisir le langage Python.

Même si le langage de programmation JavaScript ne figure pas dans cette liste, il est très utilisé, il est donc très intéressant de le choisir. Avec electron et nodejs, nous pouvons très bien créer une application multiplateforme. De plus, l'API PiCamera est disponible pour le langage JavaScript.

NodeJs et Electron ne sont pas installés par défaut sur le Raspberry Pi, donc cela nous prendra plus de temps afin de réaliser les phases d'installation et de configuration de l'environnement de travail, et plus d'incertitudes qu'avec le langage Python. En vu du délais court, nous allons choisir le langage Python.

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.

## 4.5 Réalisation de l'application

Avant toutes choses, nous allons importer les modules nécessaires. Nous allons utiliser les modules :

- **picamera** : *Module de la caméra*
- **tkinter** : *Module de l'interface graphique*
- **datetime** : *Module de gestion de la date et de l'heure*
- **os** : *Module de gestion de l'OS*
- **time** : *Module de gestion du temps*

Nous allons créer une maquette du menu principal de l'application. Cette maquette nous permettra d'avoir un aperçu de l'application. Et nous permettra de concevoir l'interface graphique (GUI<sup>6</sup>) de l'application à l'aide de Tkinter.

### 4.5.1 Création de l'interface graphique

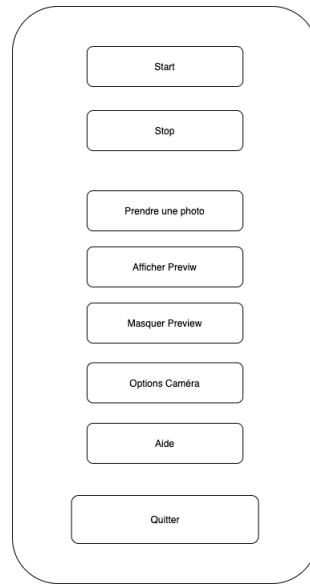


FIGURE 4.8 – Maquette de l'application

Nous allons créer 5 classes pour notre GUI qui sont :

- **class** *App*
- **class** *AppPhotoSection*
- **class** *AppOptionsSection*
- **class** *AppUtilitySection*
- **class** *AppHelpSection*

La classe *App* est la classe principale de notre application. C'est ici que nous allons définir la taille de la fenêtre, la couleur de fond, le titre de la fenêtre, le logo de l'application, etc.

C'est également dans cette classe que nous allons lancer notre menu principal par la méthode : **def \_\_init\_\_(self, master)**. La méthode `__init__` peut être appelée lorsqu'un objet est créé à partir de la classe, et un accès est nécessaire pour initialiser les attributs de la classe. Le menu principal est représenté par la maquette de la figure 4.8.

Les 4 autres class sont les classes qui représentent les sections de l'application. Lorsque que l'utilisateur clique sur un des boutons du menu principal, il se rend dans la section correspondante.

## 4.5.2 Fonction Start Recording

Afin de réaliser l'application, nous allons commencer par concevoir la fonction cœur de l'application. C'est à dire la fonction qui va nous permettre de lancer un enregistrement et de l'arrêter. Cette fonction, son fonctionnement est représenté par la figure 4.9

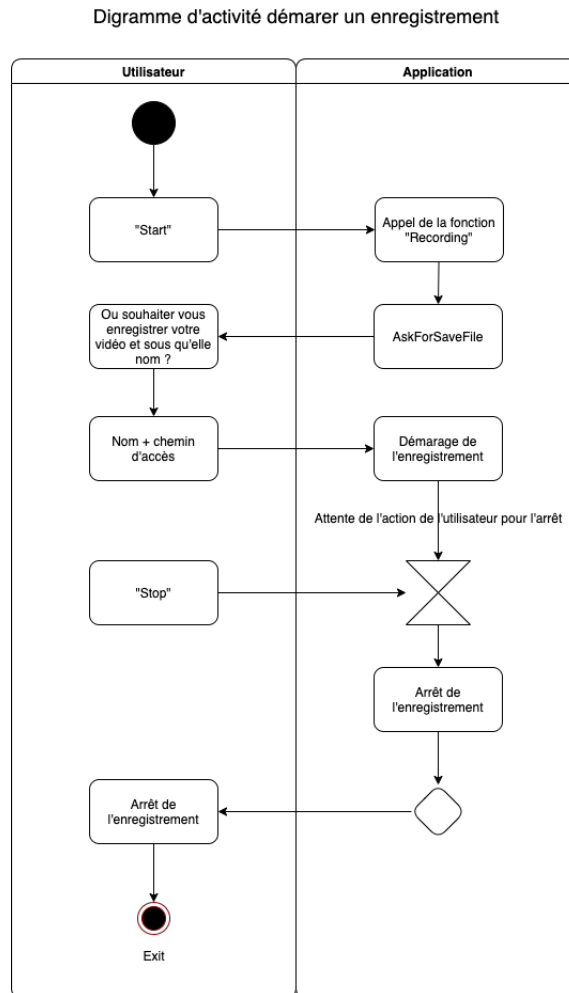


FIGURE 4.9 – Diagramme d'activité de la fonction "Démarrer un Enregistrement"

D'après notre diagramme d'activité, il nous faut commencer l'enregistrement lorsque l'utilisateur clique sur le bouton "Start". Avant de pouvoir démarrer un enregistrement, il faut initialiser la caméra, pour cela, la documentation nous indique une fonction qui est : **camera = PiCamera()**. Une fois la caméra initialisée, nous pouvons commencer à traiter la demande de l'utilisateur.

**AskForSaveFile** est une fonction de **tkinter** qui nous permet de demander à l'utilisateur de saisir un nom de fichier et de lui permettre de choisir l'emplacement de sauvegarde. Une fois que l'utilisateur a saisi le nom de fichier, nous pouvons commencer à enre-

gistrer avec la fonction de l'API PiCamera : **camera.start\_recording(nomDuFichier, format)**. L'enregistrement restera actif tant que l'utilisateur ne clique pas sur le bouton "Stop". L'arrêt de l'enregistrement se fait avec la fonction de l'API PiCamera : **camera.stop\_recording()**.

Lors des tests, nous avons remarqué qu'arrêter l'enregistrement de manière manuelle n'est parfois pas très optimale. Dans notre cas, une expérience est d'une durée minimum de 20 minutes, l'utilisateur doit donc rester attentif afin d'arrêter l'enregistrement. Il se peut que l'utilisateur oublie l'enregistrement en cours.

Nous avons donc décidé de créer une fonction nous permettant de stopper de manière automatique l'enregistrement. Il s'agit de permettre à l'utilisateur de saisir un temps en minutes et de lancer l'enregistrement. Le temps saisi en minute de l'utilisateur sera converti en secondes.

De ce fait, l'API PiCamera nous permet de définir le temps d'enregistrement en secondes avec la fonction : **camera.wait\_recording(tempsEnSeconde)**. Cependant cette fonction possède un défaut, elle agit comme une fonction **wait()** classique. Par conséquent, durant l'enregistrement, l'utilisateur ne peut pas interagir avec l'application.

Une fois que nous n'utilisons plus la caméra, nous pouvons couper toute utilisation de la caméra avec la fonction de l'API PiCamera : **camera.close()**.

### 4.5.3 Prendre une capture

La fonction prendre une capture est une fonction qui va nous permettre de prendre une photo. Cette fonction est représenté par la figure 4.10.

Contrairement à la fonction **Démarrer un Enregistrement**, cette fonction est plus simple dans l'ensemble de son fonctionnement.

Tout d'abord, comme nous l'avons vu pour la fonction **Démarrer un Enregistrement**, nous allons initialiser la caméra. Une fois celle-ci initialisée, nous allons demander à l'utilisateur de saisir un nom de fichier et l'emplacement de sauvegarde. Une fois cela fait, il nous faudra préchauffer la caméra afin de pouvoir lancer la capture d'image. Avec la fonction **sleep()**, nous allons attendre un certain délai avant de lancer la capture. Pour le préchauffage, 2-3 secondes suffiront.

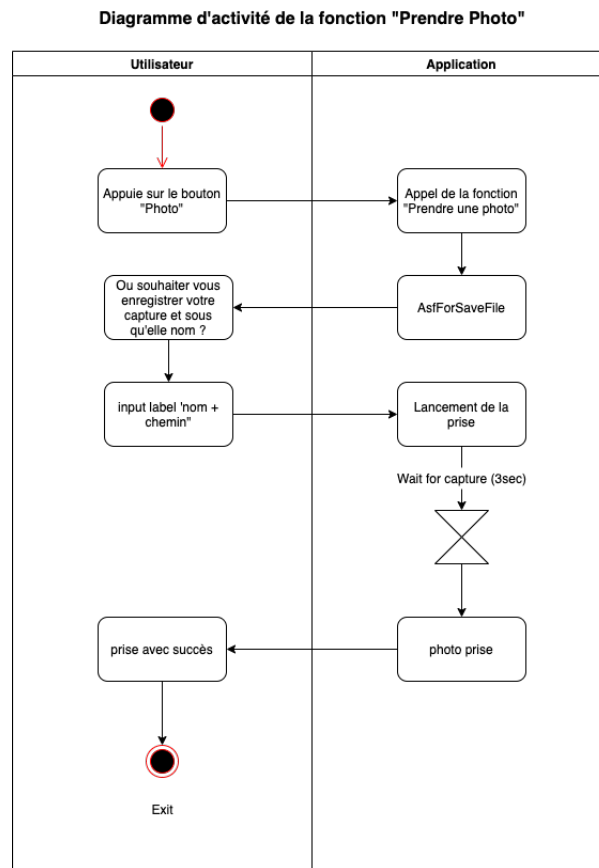


FIGURE 4.10 – Diagramme d'activité de la fonction "Prendre une capture"

Une fois cela fait, nous pouvons lancer la capture avec la fonction de l'API **camera.capture(nomDuFichier, format)**. Nous avons également décidé d'ajouter une fonction permettant de prendre des photos en continu.

Nous allons donc créer une fonction permettant de prendre des photos en continu. Afin de pouvoir faire cela, nous allons demander à l'utilisateur de saisir un nom de fichier et l'emplacement de sauvegarde comme précédemment. Mais avant cela, nous demandons à l'utilisateur de choisir le nombre de photos à prendre.

Pour cela, nous utilisons une boucle telle que :

```

for nomDuFichier in camera.record_sequence(
    '%d.h264' % i for i in range(1, nombreDePhoto)):
    camera.wait_recording(6)
  
```

Tant que le nombre de photos voulues n'est pas atteint, nous allons continuer à prendre

des photos toutes les 6 secondes. Nous avons mis un délai de 6 secondes entre chaque photo afin que la prise ne soit pas exactement identique à la précédente.

#### 4.5.4 Travail en laboratoire

Tout le système que nous avons créé jusqu'à maintenant est destiné à être utilisé dans un laboratoire. De ce fait, le module caméra sera installé sur une potence au-dessus d'un bac où sera installé diverses espèces lors des expérimentations. Cette installation suit la figure 4.11.



FIGURE 4.11 – Le dispositif installé dans le laboratoire

Nous remarquons que le Raspberry Pi est installé sur une potence dans un boîtier afin de le protéger des diverses éclaboussures, il en va de même pour la caméra qui se trouve en bas de la potence. Plus à gauche, nous pouvons voir l'écran tactile qui ne prend pas énormément de place, et qui est relativement amovible.

Les **Gerridés** utilisés pour les expériences sont directement attrapés à l'épuisette dans la Mangrove. Nous avons eu l'occasion de partir à la recherche de ces **Gerridés** dans la Mangrove de la Guadeloupe (voir Figure 4.12).

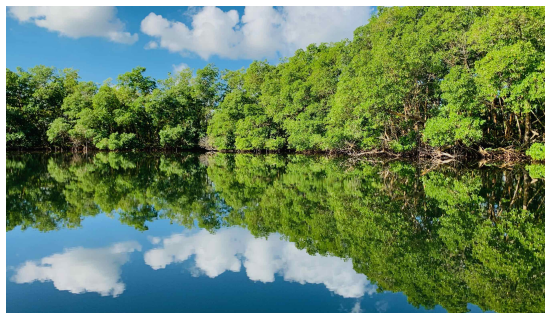


FIGURE 4.12 – Mangrove de la Guadeloupe



### 4.5.5 Travail sur la caméra

Le module caméra est très certainement celui qui nous a posé le plus de problème. Pendant une grande partie des expérimentations, nous avons eu divers problèmes lors des enregistrements.

Commençons par des problèmes de stockage de la vidéo. Pour nos expérimentations, nous devons réaliser des enregistrements d'une vingtaine de minutes. Cela représentait plus de 4Go de mémoire. Cela était trop grand pour le stockage de la vidéo sur le Raspberry Pi.

Nous avons également eu des soucis d'éclairage, qui provoquaient des effets d'anneaux sur la vidéo, cela a plus ou moins été réglé en modifiant le nombre de données traitées par unité de temps (bitrate) de l'enregistrement, ainsi qu'en mettant le dispositif dans un endroit sans éclairage. L'utilisateur peut également modifier le contraste et la luminosité de la vidéo afin d'avoir un meilleur rendu.

Nous avons également rencontré deux autres problèmes, le premier concernent les effets de zoom lors de nos enregistrements qui étaient dus à une trop forte résolution de la caméra. Le deuxième était un problème de saut d'image dus à un souci au niveau du format de la vidéo. Lors de conversion de la vidéo en format mp4, ce dernier problème a été résolu. Cela laisse à supposer que l'écriture au format H264<sup>7</sup> n'était pas adapté pour la configuration de notre Raspberry Pi.

### 4.5.6 Travail sur vidéo

Après avoir utilisé notre dispositif lors des expérimentations, les biologistes ont besoin de résultats et pour ce faire nous devons extraire un certain nombre d'images des vidéos, avec le framework<sup>8</sup> *FFmpeg*.

FFmpeg est un projet de logiciel libre qui produit des bibliothèques et des programmes pour gérer et manipuler des données multimédias. FFmpeg peut gérer l'ensemble du processus de transcodage, de manipulation de vidéos et d'images (redimensionnement, débruitage, etc.), de conditionnement, de diffusion en continu et de lecture. Il s'agit du logiciel de traitement de vidéo et d'image le plus populaire et est utilisé par de nombreuses entreprises dans divers secteurs.

Pour ce faire, nous utilisons une commande de ffmpeg :

```
ffmpeg -i Nom_Vid.h264 -r .6 -q:v 1 ./Nom_Vid_%6d.jpg
```

Nous remarquons que cela n'est pas simple d'utiliser cette commande, c'est pourquoi nous avons créé une fonction permettant à l'utilisateur de choisir une vidéo ainsi que le dossier où il souhaite enregistrer les images extraites.

Une fois que nous avons nos images, nous pouvons les annoter avec un logiciel en ligne : **www.robots.ox.ac.uk**. Mais le gros problème, c'est qu'il nous faut annoter les images de façon manuelle et une par une, et afin d'avoir des résultats précis, il nous faut travailler avec d'énorme quantité d'image dans le but d'entraîner un système d'apprentissage automatique, pour un individu annoter environs 900 images n'est pas chose facile.

Nous allons donc voir dans la conclusion comment nous améliorer notre système pour que les biologistes puissent travailler avec plus de confort et de rapidité tout en ayant des résultats d'expérimentations de qualité.

# Chapitre 5

## Conclusion

Ce stage a permis le début du développement d'un système de reconnaissance de Gerridés afin de faciliter le travail des biologistes durant leur expérimentations.

### 5.1 Ressentis par rapport au stage

Le travail scientifique que je réalise durant mes études m'a permis de comprendre les différentes documentations allant du montage du Raspberry Pi à la programmation et l'utilisation de l'API PiCamera afin de concevoir l'application.

### 5.2 Perspectives futures

Durant ce stage, nous avons remplacé un dispositif permettant la capture vidéo de qualité en termes de vidéo grâce à la GoPro, mais n'était pas optimisé pour le travail en laboratoire.

De ce fait nous avons conçu un dispositif permettant la capture vidéo avec divers fonctionnalités grâce à l'application comme l'arrêt automatique des vidéos, prendre le nombre de photos souhaité en rafale, avec la possibilité d'avoir un rendu de la vidéo en temps réel dans les mains via à l'écran tactile. Bien que nous perdions en qualité d'enregistrement vidéo et photo, nous avons réussi à réaliser un système optimal pour un moindre coût.

Mais le système n'est pas fixe, c'est à dire que nous pouvons changer le module caméra par un modèle plus performant afin d'avoir un meilleur rendu vidéo. Concernant

l'application, nous pouvons améliorer le fait que l'application se bloque durant le mode rafale ou le mode d'enregistrement automatique en faisant du multiprocessus. Cela nous permettrait de réaliser une tâche tout en faisant une autre.

Il serait également intéressant d'automatiser l'annotation d'image, en effet annoter une grande quantité d'image est une tâche très longue. Avec la prise de vue en temps réelle, nous pourrions annoter de manière automatique les images et en grande quantité sans que cela soit une tâche fastidieuse.

# Définition

**Définition 1.** *ARM : ARM est une société britannique spécialisée dans le développement de processeurs d'architecture 32 bits et d'architecture 64 bits de type RISC. Filiale de SoftBank<sup>1</sup> depuis 2016. ARM développe également un grand nombre de blocs de propriété intellectuelle (IP). ARM est désormais présent dans le monde entier et son siège historique se situe à Cambridge.*

**Définition 2.** *Overclocking : L'overclocking, ou parfois sur-cadencement<sup>1</sup>, ou surcadencage est une manipulation ayant pour but d'augmenter la fréquence du signal d'horloge d'un processeur au-delà de la fréquence nominale afin d'augmenter les performances de l'ordinateur.*

**Définition 3.** *OS : Un système d'exploitation (ou système d'exploitation) est un ensemble de programmes et d'outils qui permettent d'exécuter des programmes sur un ordinateur.*

**Définition 4.** *Paradigme : Un paradigme de programmation est une façon d'approcher la programmation informatique et de traiter les solutions aux problèmes et leur formulation dans un langage de programmation approprié.*

**Définition 5.** *API : (Application Programming Interface) est une interface de programmation qui permet d'accéder à des fonctionnalités d'un logiciel.*

**Définition 6.** *GUI : (Graphical User Interface) est une interface utilisateur graphique qui permet d'interagir avec un logiciel.*

**Définition 7.** *H264 : H264 est un codec compressé de type video qui permet de compresser des images en une seule image.*

**Définition 8.** *framework : Un framework est un ensemble de composants qui sont utilisés pour construire un logiciel.*

# Webographie

## [1] Installation du Raspberry Pi

- <https://www.raspberrypi.com/documentation/computers/getting-started.html>
- <https://www.raspberrypi-france.fr/guide/installer-raspbian-raspberry-pi/>

## [2] Premiers pas avec le module caméra

- <https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/0>

## [3] Configuration du Raspberry Pi

- <https://projects.raspberrypi.org/en/projects/raspberry-pi-using/9>
- <https://www.raspberrypi-france.fr/guide/configurer-raspbian/>
- <https://www.raspberrypi-france.fr/guide/connecter-ssh-raspbian/>
- <https://www.waveshare.com/wiki/7HP-CAPQLED>

## [4] Documentation API et librairies

- <https://picamera.readthedocs.io/en/release-1.13/>
- <https://www.npmjs.com/package/pi-camera>
- [https://developer.mozilla.org/en-US/docs/Web/API/File\\_System\\_Access\\_API](https://developer.mozilla.org/en-US/docs/Web/API/File_System_Access_API)
- <https://docs.python.org/fr/3/library/tk.html>
- <https://realpython.com/python-gui-tkinter/>
- <https://www.ffmpeg.org/documentation.html>

## [5] Ressources Informative

- <https://wethegeek.com/programming-languages-for-raspberry-pi/>
- [https://fr.wikipedia.org/wiki/Wikipédia:Accueil\\_principal](https://fr.wikipedia.org/wiki/Wikipédia:Accueil_principal)
- <https://www.ffmpeg.org/about.html>