



UFR SEN
Faculté des Sciences
Exactes et Naturelles



Aymerick LAURETTA-PERONNE

*Université des Antilles
Équipe Biologie de la Mangroove*

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

15 février 2022

Remerciements

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

Tout d'abord, je tiens exprimé 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 chaleureuse au sein de l'équipe et de m'avoir fait partager ses connaissances et son expériences 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.

Table des matières

1	Introduction	3
2	Environnement	4
2.1	Présentation de la structure	4
2.2	Organisation du stage	5
3	Présentation de la problématique	6
3.1	Présentation du sujet	6
3.2	Présentation de l'aspect biologiste	6
3.3	Objectifs	7
4	Travail réalisé	8
4.1	Mon travail	8
4.2	Réception du matériel	8
4.3	Montage, Installation et Configuration	10
4.4	Choix des technologies	12
4.5	Réalisation de l'application	13
5	Conclusion	19
5.1	Interface labo / traitement informatique	19
5.2	Ressentis par rapport au stage	19
5.3	Perspectives futures	19

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 cloturer mon cursus universitaire. Il me permet de mettre en pratique mes acquis en informatique et de me familiariser avec la vie professionnelle.

Dans ce rapport de stage, je vais décrire le contexte de l'entreprise, la problématique, les objectifs, les étapes de réalisation du logiciel de capture vidéo.

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 intégré 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 teneurs 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 également Professeur et chercheur à l'Université des Antilles. Les membres de l'équipe projet Monsieur Matheiu 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 Monsieur Manuel CLERGUE afin de faire le point sur l'avancement du projet.

Le travail a été réparti en deux axes :

- **Partie en laboratoire**
- **Partie sur le terrain**

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 pour me permettant d'installer des mise à jour de logiciels.

2.2.2 Contraintes particulière

Durant ce stage j'ai eu deux contraintes particulières :

- Premièrement, le matériel ayant été commandé avant le début de mon stage et des différents aspects techniques.
- Deuxièmement, le calendrier stricte en début de stage demandant un système minimal en 10 jours.

Chapitre 3

Présentation de la problématique

3.1 Présentation du sujet

Projet avec le laboratoire de *Biologie* de l' *Université des Antilles*, le projet concerne en l'analyse de déplacement de **Gerridés** afin de déterminer leur préférence sur des 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.

Afin de déterminer leur préférence, il est nécessaire d'avoir un dispositif adéquat de capture vidéo pour l'acquisition de données.

3.2.1 Positionnement du travail dans le projet

Le tout étant fait de façon manuelle, 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 ils

leur fallait placé le bac dans le champ de vision de la GoPro a chaque fois qu'ils voulaient faire une capture.

Il fallait aussi que la GoPro soit connecté 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 décidé d'améliorer leur dispositif afin de pouvoir réaliser l'expérience de manière plus pratique.

Ce qui nous permet de nous enchaîner sur l'objectif qui en découlent, comment nous pourrions améliorer le dispositif.

3.3 Objectifs

L'objectif du stage 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.

Il sera donc question de permettre au biologiste de réaliser des captures vidéo de manière plus pratique. A défaut d'avoir un dispositif non-fixe comme énoncé, nous réaliserons un dispositif qui sera fixe.

Ce dispositif sera installé sur une potence au-dessus du bac et restera fixe en permanence permettant de limiter voir supprimer les divers déplacements.

Une application conviviale sera développée afin de permettre au biologiste de réaliser des captures vidéo plus facilement.

Avec un écran nous pourrions interagir avec l'application permettant de démarrer et d'arrêter un enregistrement.

Ainsi avec l'application nous pourrions avoir une prévisualisation de la capture vidéo.

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 captures vidéos 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.

Voyons maintenant comment faire.

Chapitre 4

Travail réalisé

Avant de commencer, je vous présente mon travail sur le sujet du stage.

Dans un premier temps, il faudra installer le système Raspbian. Une fois l'installation faite, il faudra configurer le système. Avant de commencer l'application, il me faudra faire quelques tests afin de vérifier le bon fonctionnement. Une fois cela terminé, il me faut vérifier que la caméra fonctionne correctement.

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.

Dans un premier temps, il faudra installer le système Raspbian. Une fois l'installation faite, il faudra configurer le système. Avant de commencer l'application, il me faudra faire quelques tests afin de vérifier le bon fonctionnement. Une fois cela terminé, il me faut vérifier que la caméra fonctionne correctement.

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 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 vu de sa petite taille, il est très fragile mais égalemnt 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 mais il faudrait un système de refroidissement pour le Raspberry Pi.



FIGURE 4.1 – Raspberry Pi 4

4.2.2 Ecran LCD (Raspberry Pi)

Avec le *Raspberry Pi* un écran LCD de 7 pouces permettant d'interagir avec l'ordinateur grâce à son écran tactile.

L'Intérêt majeur de ce type de périphérique c'est sa finesse. En effet, dans un laboratoire il est pas toujours facile d'avoir un grand écran. De ce fait avoir un écran de petite taille qu'on peut tenir dans nos main 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 a manipuler.
- **Inconvénients** : Cependant la qualité de ce module caméra est relativement mauvaise. et la configuration et l'utilisation est pas toujours évident. De plus il est très 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 est pas monté sur le Raspberry Pi. Le Raspberry Pi arrive sans son système pré-installé.

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.

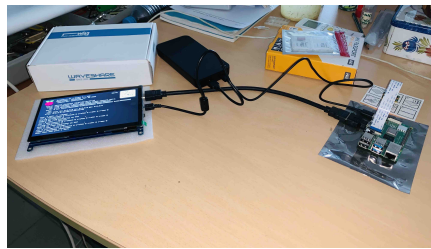


FIGURE 4.4 – Matériel assemblé et prêt à l'emploi

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

Le système étant pas pré-installé, il nous faut installer le système d'exploitation. Pour ce faire nous allons utiliser Raspberry Pi Imager.

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

Il nous faut télécharger le fichier d'installation sur le site web Raspberry.com et de l'installer.

Nous obtenons alors l'aperçu de la figure 4.4.

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.

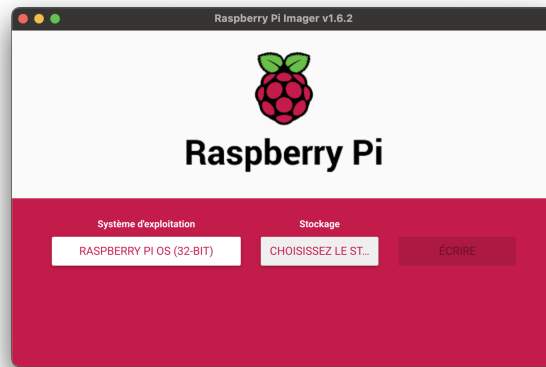


FIGURE 4.5 – Interface "Raspberry Pi Imager"

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

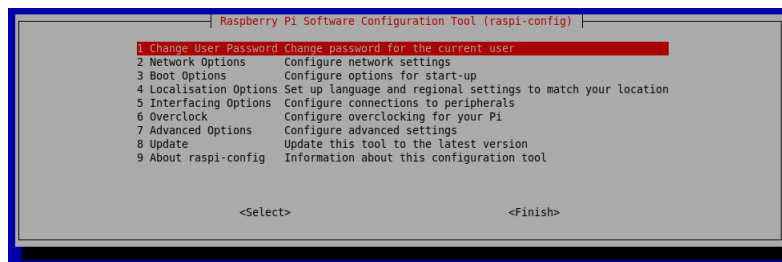


FIGURE 4.6 – Interface "raspi-config"

Une fois dans **Raspi-config**, nous allons choisir la section **Interface section**.

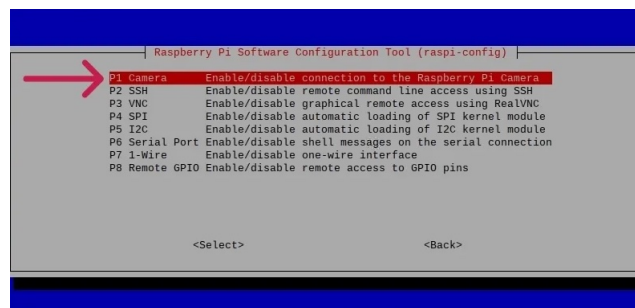


FIGURE 4.7 – camera option (raspi-config)

On selectionne **P1 Camera** et on met sur **Enable**.

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.

La photo est bien prise. De ce fait la caméra est opérationnelle.
Nous pouvons tout de même 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. Donc tout est fonctionnel.

4.4 Choix des technologies

Pour le choix du langage de programmation, je me suis basé sur les langages de programmation les plus classés pour le (Raspberry Pi) système d'exploitation Raspbian. Lors de ma recherche 5 langages de programmation en sont sortis :

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

On peut voir que le langage Python est le plus utilisé pour le (Raspberry Pi) 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 me décider sur le langage de programmation, il me faut chercher la compatibilité des API pour la camera.

Il s'avère que l'API PiCamera est installer par défaut sur le système d'exploitation Raspbian.

Puisque le langage Python et l'API PiCamera sont compatibles, et installer 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.

Mais par manque de temps, nodejs et electron ne sont pas installés par défaut sur le Raspberry Pi.

De ce fait nous allons utiliser le langage Python pour la programmation du logiciel. 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**
- **tkinter**
- **datetime**
- **os**
- **time**

Maintenant que nos modules sont importés, 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 le GUI de l'application à l'aide de Tkinter.

4.5.1 Création de l'interface graphique

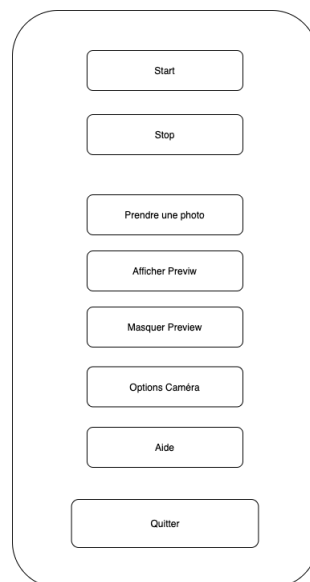


FIGURE 4.8 – Maquette de l'application

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, 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 coeur 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 érigés par la figure 4.9

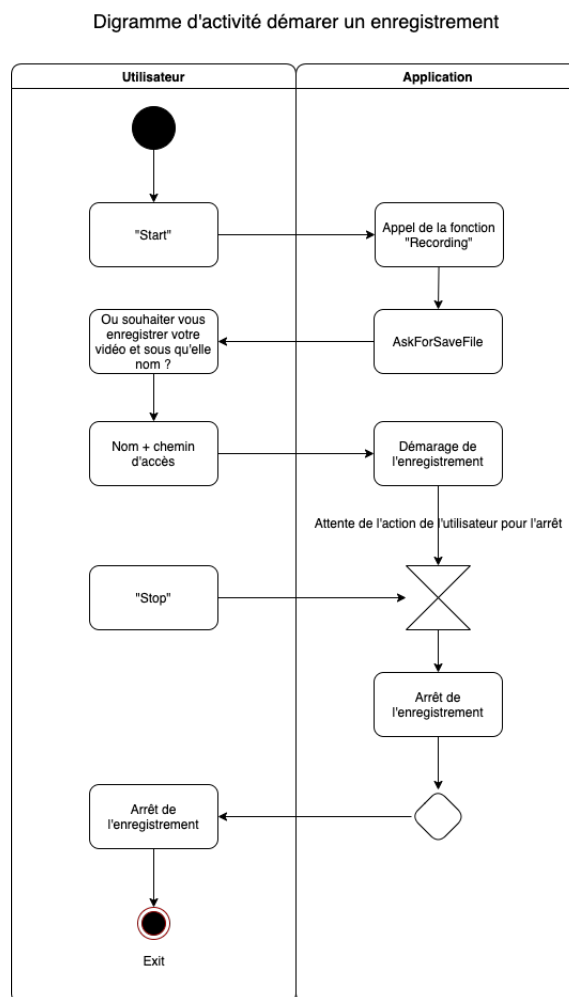


FIGURE 4.9 – Diagramme d'activité de la fonction "Démarer un Enregistrement"

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 pour cela 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 va nous permettre 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 enregistrer 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 est parfois pas très optimale. Nous avons donc décidé de créer une fonction nous permettant de stopper de manière automatisée 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. De ce fait, 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 érigée 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 pré-chauffer 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 pré-chauffage, 2-3 secondes suffiront.

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

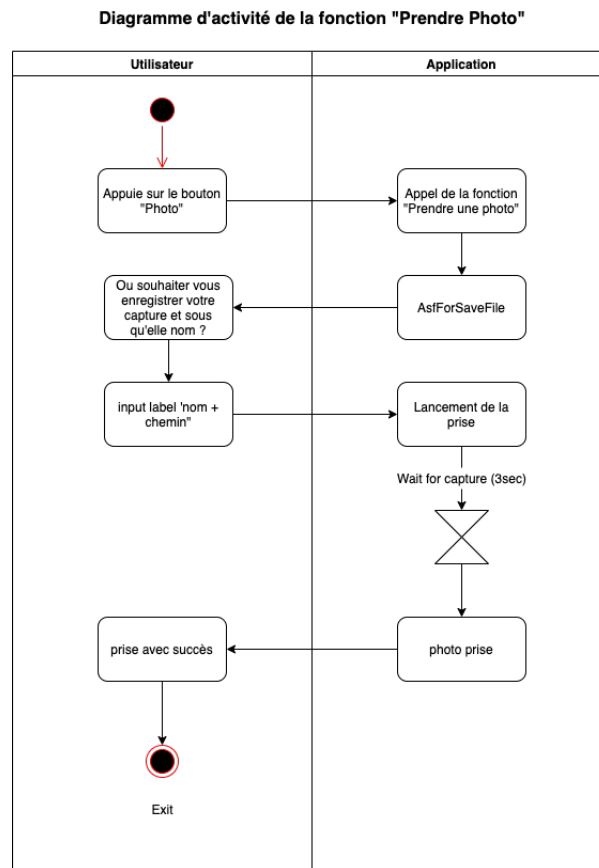


FIGURE 4.10 – Diagramme d'activité de la fonction "Prendre une capture"

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 voulue 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 un potence au dessus d'un bac ou sera installé divers espèce lors des expérimentations.

Cette installation suit la figure 4.11.

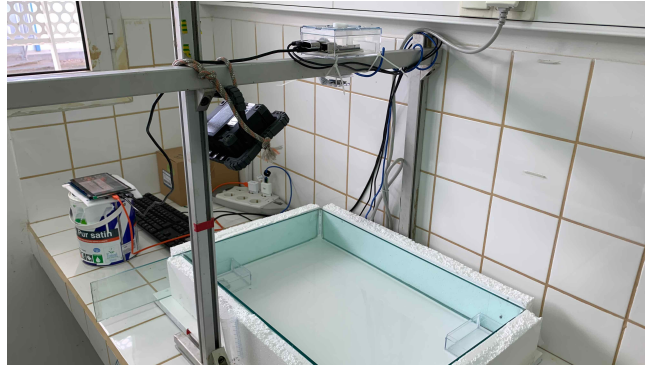


FIGURE 4.11 – Le dispositif installé dans le laboratoire

Nous pouvons voir que le Raspberry Pi est installé sur un potence dans un boîtier afin de le protéger des divers 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.

J'ai également eu l'occasion de partir à la recherche de Gérridés dans la Mangrove de la Guadeloupe et d'en attraper a l'épuisette (voir Figure 4.12).

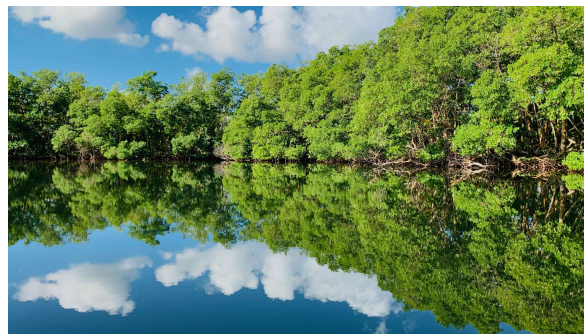


FIGURE 4.12 – Mangrove de la Guadeloupe

4.5.5 Travail sur la caméra

Le module caméra est très certainement celle qui nous à posé le plus de problème.

Pendant une grande partie des expérimentations, nous avons eu divers problème lors des enregistrements.

Commencent par des problèmes de stockage de la vidéo, pour nos expérimentations, nous devons réaliser des enregistrement d'une vingtaine de minutes. Pour une vidéo de 20 minutes, nous avons besoin de plus de 4Go de mémoire. Cela était trop grand pour le stockage de la vidéo sur le Raspberry Pi.

Nous avons également eue des soucis d'éclairage, qui provoquait des effets d'anneaux sur la vidéo, cela à plus ou moins été réglé en modifiant le bitrate de l'enregistrement ainsi qu'en mettant le dispositif projet dans un endroit où il n'y a pas d'éclairage. L'utilisateur peut également modifier le contraste et la luminosité de la vidéo afin d'avoir un meilleur rendu.

Nous avons rencontré également deux autres problèmes qui sont notamment les effets de zoom lors de nos enregistrement qui étaient du à une trop forte résolution de la caméra ainsi qu'un problème de saut d'image dans nos enregistrement. Cela étant du au format de la vidéo, lors de la conversion vidéo en format mp4, ce soucis n'apparaisait plus.

Cela laisse à supposer que l'écriture au format H264 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 résultat et pour ce faire nous devons extraire un certain nombre d'image des vidéos.

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, de ce fait nous avons créé une fonction permettant à l'utilisateur de choisir une vidéo ainsi que le dossier où il souhaite enregistrées 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, 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 Interface labo / traitement informatique

5.2 Ressentis par rapport au stage

Le travail scientifique que je réalise durant mes études mon 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.

5.3 Perspectives futures

Durant ce stage, nous avons remplacer un dispositif permettant la capture vidéo de qualité en terme 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 perdons 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 la fait que l'application se bloque durant le mode rafale ou le mode d'enregistrement automatique en faisant du multi-processus. 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 tache très longue.

Avec la prise de vue en temps réelle, nous pourrions annoter anoter de manière automatique les images et en grande quantité sans que cela soit une tâche fastidieuse.

Définition 1. *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.*

Définition 2. *Overclocking : L'overclocking, ou parfois sur-cadencement¹, 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. *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 4. *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 5. *API : L'API (Application Programming Interface) est une interface de programmation qui permet d'accéder à des fonctionnalités d'un logiciel.*

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