

# Projet Pynect

Par LORAIN Antoine et MARQUET Quentin

Sommaire :

## Table des matières

I) Présentation générale du projet.....	2
II) Organisation du travail.....	2
III) Présentation des différentes étapes du projet.....	2
a. Choix du module Python.....	2
b. Installation et tests du module.....	2
c. Modification de l'image.....	3
IV) Validation de l'opérationnalité et du fonctionnement et ouvertures possibles.....	3

## **I) Présentation générale du projet**

Le projet Pynect a été inspiré du projet SARndbox. Il s'agit d'un projet ayant pour but de dessiner des formes dans un bac rempli de sable et que l'on projette une image mettant en avant les formes tracées. Ces formes sont détectées par le capteur de profondeur de la Kinect de Microsoft. La différence entre SARndbox et le projet Pynect est que Pynect est fait en Python alors que SARndbox est fait en C++. Le projet Pynect n'est fonctionnel que sur Linux et n'a été testé qu'avec une Kinect de 1<sup>er</sup> génération pour Xbox 360. Le modèle utilisé ici est le modèle 1414. Ce projet a été réalisé pour le FabLab du Chalonais. Ce projet avait été commencé par certains adhérents mais n'était pas terminé. Ce projet reprenait SARndbox et était donc fait en C++ également. Nous avons donc du tout reprendre depuis le début. Fort heureusement, le bac à sable avec la potence pour tenir la Kinect au-dessus du bac ainsi que celle pour y accrocher un vidéo projecteur afin de voir le dessin était déjà construit et opérationnel. Nous avons donc pu récupérer ce travail.

## **II) Organisation du travail**

Pour réaliser ce projet, nous étions deux élèves de 1<sup>er</sup> Générale. Il fallait donc que nous trouvions un moyen de récupérer une image depuis le capteur de la Kinect avec Python puis que nous traitions cette image afin d'y afficher quelque chose de cohérent avec le tracé sur le sable.

## **III) Présentation des différentes étapes du projet**

### **a. Choix du module Python**

Après nos recherches, nous avons trouvé quatre options différentes. Nous avons le choix entre le module PyKinect, PyKinect2, libfreenect et libfreenect2. Deux de ces modules peuvent être retirés de la liste, il s'agit de PyKinect2 et libfreenect2 qui ont été faits pour la Kinect de 2<sup>nd</sup> génération pour la Xbox One et ne sont donc pas compatibles avec notre Kinect. Les deux modules restants ne sont pas disponibles dans les dépôts PyPi et ne peuvent donc pas être installés « pip install {module} » ou « sudo apt install python3-{module} ». Le module PyKinect doit être téléchargé sur un repository GitHub et doit être extrait dans le dossier sites-package de l'environnement Python. Cependant, une fois la procédure d'installation complète, nous n'avons jamais réussi à l'utiliser pour récupérer une image. Libfreenect est en revanche un projet Open Source pour Linux pour la Kinect. Heureusement, OpenKinect (les développeurs de libfreenect) ont créé un wrapper pour Python. Nous avons donc choisi d'utiliser un Raspberry Pi 4 pour la réalisation de ce projet.

### **b. Installation et tests du module**

Toute la procédure d'installation avec les commandes à utiliser est détaillée dans le fichier readme.md du projet et du repo GitHub de Pynect. L'installation consiste à installer les dépendances nécessaires à la compilation du pilote, à la compilation du pilote avec cmake et make, à l'installation du pilote et à la

compilation et à l'installation du wrapper python. Un fois le module python installé, nous avons réussi à exécuter un fichier de démonstration de OpenKinect. Afin de simplifier la tâche de développeur Python qui utilisent libfreenect, OpenKinect a mis à disposition un fichier Python appelé `frame_convert2` qui permet de convertir les données de la kinect en un format facile à modifier et à afficher avec le module Python `cv2`.

### c. Modification de l'image

La fonction `modifiedColors()`. Elle a pour but de traiter les données de l'image en niveaux de gris et les transformer en une image en couleurs. La fonction commence par récupérer les données avec `getdepth().tolist()`. Il s'agit des données de l'image en nuance de gris qui est reconvertie en tableau Python. Ensuite, la fonction va récupérer la liste contenant toutes les informations de remplacement de couleur qui sont stockées dans le fichier `config.ini`. Un boucle va ensuite vérifier chaque pixel afin de savoir dans quel intervalle ce dernier se trouve. Afin d'optimiser ce processus, la fonction vérifie si la dernière nuance est la même que la nuance actuelle afin de ne pas effectuer trop de tests. Une fois cette liste modifiée, elle peut être affichée avec le module `OpenCV`.

## **IV) Validation de l'opérationnalité et du fonctionnement et ouvertures possibles**

Comme annoncé plus tôt, ce projet a été fait pour le Fablab du Chalonais. Nous nous sommes donc rendu là-bas afin de mettre en place la kinect (voir photos) et tester l'ensemble du projet. Nous avons à ce moment remarqué un temps de latence entre chaque image. Nous avons donc utilisé le module `datetime` de python afin d'obtenir les heures/minutes/secondes du démarrage de l'application ainsi que celles de l'affichage de la première image. Nous avons remarqué à ce moment qu'il y avait un décalage d'environ une seconde entre deux images. Après avoir testé chaque ligne de code qui traitait l'image, nous nous sommes aperçu que nous ne pourrions pas résoudre ce problème car il venait de la façon que nous utilisions pour traiter l'image. Nous avons calculé qu'il y avait 307200 pixels par images. Chaque pixels contient en plus trois valeurs. Il y a donc 921600 valeurs par image et 921600 valeurs à modifier. Nous avons également tester de supprimer le traitement de l'image afin de voir si le problème ne venait pas du module python en lui-même. Nous avons également tester d'utiliser le projet Pynect sur un PC bien plus puissant qu'un Raspberry Pi 4 et qui tournait sous Ubuntu 24.10 (au lieu de Raspberry Pi OS datant du 19 novembre 2024) Par conséquent, les possibles modifications et ajouts sont donc :

- Modification du traitement de l'image afin d'augmenter la vitesse de traitement.
- Ajout de transition entre deux images afin d'avoir un mouvement d'eau par exemple là où il y a une ligne qui est tracée dans le sable.

## Projet Pynect

- Port du projet sous Windows et par conséquent, modification du module python afin de support Windows. (Ou faire fonctionner libfreenect sous Windows)