



UNIVERSITÉ / UCLouvain

---

## APP3 Informatique

---

*Auteurs :*

Mathias DE VOS

John DECORTE

Gaspard ROBIN

Maïcky DAOUST

Rapport du projet Raspberry PI

14 décembre 2022

# 1 Introduction

Le contexte du projet est que l'état nous a commandé un appareil permettant de retrouver facilement des mots de passe et de pouvoir inscrire une liste de course et la modifier à tout moment. Il nous est demandé de faire cet appareil sur un Raspberry PI en utilisant le langage informatique "Python". Pour améliorer et rendre unique notre Raspberry, il faut créer une fonction personnalisée qui utilise un de ses capteurs. On a décidé en groupe de créer un programme d'aide pour les personnes âgées(SOS) et ajouter un jeu pour pouvoir les divertir.(Tetris)

## 2 Répartition des tâches

Tout le monde a eu des tâches similaires au niveau du temps, mais relativement différent, tout en essayant que tout le monde crée une partie de code. Par exemple, Maïcky et Gaspard se sont occupés à deux du gestionnaire de mots de passe et de la liste de course, dû aux nombreux points communs de ces deux fonctions. Mathias s'est occupé de la fonction SOS qu'on va développer dans son propre paragraphe. John a programmé et optimisé le Tetris pour qu'il tourne bien sûr le Raspberry.

## 3 Mots de passe et liste de course

La fonction mots de passe sert à stocker un mot de passe complexe en toute sécurité en ayant la capacité de le retrouver facilement, si les données sont vides, on peut implémenter un nouveau mot de passe ou supprimer un déjà existant grâce à la reconnaissance vocal pour que ce soit plus facile d'accès pour les personnes âgées. Les mots de passes sont stockés sur le fichier sentence.ini du Raspberry qui est crypté par hachage pour éviter l'accès aux mots de passe par des personnes malveillantes.

Cette fonction sert à stocker une liste de course et sert de pense-bête pour l'utilisateur. On peut supprimer, ajouter des produits pour créer une liste de course complètement personnalisée. Notre sélection dispose d'un grand éventail diversifié de produit de grande surface, toutes reconnues par la reconnaissance vocal.

## 4 SOS()

Cette fonction permet de détecter certains mouvements, si les mouvements effectués correspondent à notre paterne, cela va directement envoyer un message d'alerte à une ambulance ou un proche (au choix). Cette procédure fonctionne grâce au capteur de mouvement qui détecte tous les mouvements du Raspberry. Le paterne pré-enregistré dans le code est trois rotations de suite. Après avoir effectué le paterne, le programme nous laisse du temps pour confirmer ou annuler l'envoi pour éviter les erreurs et ne pas appeler une ambulance pour rien. Comment on fait pour envoyer un message ? On va utiliser une API qui transmet une requête à un serveur qui va envoyer le message pour nous.

## 5 Tetris()

Ce magnifique programme nous propose de prendre une partie de plaisir en jouant à Tetris sur notre Raspberry. Les pièces tombent toutes seules et vous pouvez les changer de sens avec le joystick du haut pour les imbriquer. Si vous dépassez le cadre du haut, vous perdez la partie et vous recevez votre score que vous pouvez améliorer en rejouant d'autre partie.

## 6 Détails techniques

Les détails techniques du code sont inscrits et bien détaillés à l'intérieure même du code.

## 7 Les problèmes rencontrés

Nos problèmes rencontrés étaient le fait que la reconnaissance vocale était vraiment médiocre et qu'on a perdu beaucoup de temps pour le faire comprendre notre mot de passe. Aussi, c'était d'avoir un Tetris fonctionnel sur un petit écran de 64 pixels. Heureusement, avec du temps et de la réflexion, on a réussi à régler la plupart de nos problèmes.

## 8 Optimisation

Notre menu est très fonctionnel et on ne perd pas de temps, car on peut "scroller", on ne doit pas attendre chaque fonction un par un. Simple d'accès pour les personnes âgées car toutes les actions peuvent être exécutés vocalement. Le programme est lisible et simplifié, ce qui aide si nous ou quelqu'un d'autre doit modifier quelque chose dans le programme.

## Références

<https://pythonhosted.org/sense-hat/api/sense-hat-api-reference>  
<https://trinket.io/sense-hat>