

Challenge européen Astro Pi

2016 - 2017

Phase 2 – Rapport de mission

I– MISSION PRINCIPALE

Principal objectif : détecter la présence d'un/plusieurs membre(s) d'équipage dans le module Columbus en utilisant l'Astro Pi de l'ISS et ses capteurs.

Procédure

Nous avons commencé par analyser les données tweetées par le compte @RaspberrySpy, qui est l'AstroPi présent dans Columbus. Notre professeur a tracé des courbes avec les données récoltées et nous avons pu nous apercevoir que la température et la pression avaient des variations lentes, au contraire de l'humidité qui semblait fluctuer plus rapidement. Nous avons donc décidé d'utiliser le capteur d'humidité de l'AstroPi pour détecter une présence. Comme nous n'avions pas encore reçu notre AstroPi, nous avons commencé à coder sur Trinket, ce qui permis à notre professeur de commencer notre initiation à Python. Notre classe s'est alors séparée en deux groupes :

- une équipe (16 élèves) a travaillé sur les algorithmes de détection (Mission 1 et 2 + Easter Eggs)
- une équipe (19 élèves) a travaillé sur les animations graphiques qui se déclencheraient au cours de ces algorithmes.

Une fois l'AstroPi reçu et installé dans son boîtier, nous avons commencé par faire afficher les valeurs d'humidité en boucle sur l'afficheur LED. Nous souhaitions utiliser une toute petite salle vide située à côté de notre salle informatique : elle faisait une parfaite réplique du laboratoire Columbus ! Mais il fallait pour ça faire en sorte que l'AstroPi démarre notre programme au démarrage : c'était difficile mais après quelques échecs notre professeur a réussi. Nous avons donc pu constater comme prévu que l'humidité augmentait assez sensiblement au bout d'une minute ou deux lorsque nous rentrions tous dans la pièce.

Il nous restait donc à établir :

- une valeur de référence, que nous avons décidé de calculer au démarrage de notre programme en faisant la moyenne de l'humidité pendant environ 1 minute.
- Une valeur de seuil de déclenchement : après plusieurs tests, nous avons choisi de prendre un seuil de +4% rH.

explication du code par Gaëtan : <https://youtu.be/7sPoiToSbSs>

Résultats attendus

Notre programme commence par une fonction `calibrage()` qui, après une minute de mesures, doit écrire dans le fichier `data_FM.csv` la valeur moyenne de l'humidité. Ensuite, notre programme entre dans sa boucle principale et écrit toutes les 10 secondes dans le fichier `data_FM.csv` les valeurs de l'humidité mesurée. Toutes les secondes environ, un test de dépassement du seuil est effectué : si il est positif, la mention "*présence détectée*" apparaît dans notre fichier data, et une animation se lance sur l'écran de l'AstroPi. Ceci se répète tant que l'humidité reste au dessus du seuil calculé.

Cette méthode est malheureusement sensible à la qualité du calibrage : si une personne est présente dans Columbus lors de la récolte des données nécessaires au calcul de la moyenne, celle-ci sera haute et le seuil ne sera pas dépassé !

II– MISSION SECONDAIRE

Principal objectif : détecter les moments où a lieu une poussée de maintien sur orbite de l'ISS.

Procédure

En étudiant le document *How to collect data from the AstroPi*, nous avons trouvé à la fin de celui-ci un lien vers un fichier de données collectées dans le module Columbus entre le 16/02/2016 et le 29/02/2016.

Ce document mettait aussi en lien un graphique permettant de voir qu'une poussée de remise sur orbite avait eu lieu vers mi-février.

En ouvrant avec un tableur les plus de 100 000 lignes du fichier de données, et en classant par ordre décroissant les colonnes relatives à l'accélération en X, Y, ou Z, nous avons trouvé !

Le 19/02/2016 à 13h24, l'accélération en Z est passée brutalement de 0,014 (valeur stable depuis le début des mesures) à 0,029 pendant 30 secondes avant de se restablir à 0,014.

Nous avons donc décidé d'appliquer la même méthode qu'à la mission 1 (calibrage puis détection d'un dépassement de seuil), mais sur les trois axes X, Y, et Z. Le seuil a été fixé à 0,01, d'après les valeurs relevées lors de la poussée du 19/02/2016.

Comme nous ne savons pas quelle sera l'orientation de l'AstroPi lors de cette poussée, nous avons décidé de faire un test simultané sur les trois axes, par des conditions reliées par le mot `or`. De même, ne sachant pas dans quel sens se ferait la variation, nous avons dû utiliser une valeur absolue (expliquée par notre professeur).

Nous avons donc le test final :

```
if abs(accX-MaccX)>seuilP or abs(accY-MaccY)>seuilP or  
abs(accZ-MaccZ)>seuilP:
```

qui déclenche alors, comme pour la Mission 1, l'écriture dans le fichier de données, et une animation sur l'afficheur LED.

explication du code par Aymerick : <https://youtu.be/A0Zrz5AHxfs>

Résultats attendus

Comme pour la Mission 1, nous procédons d'abord à un calibrage, mais cette fois avec moins d'inquiétude. En effet les relevés de données nous ont montré un très grande stabilité des valeurs d'accélération en X, Y et Z. De plus, il y a peu de chances qu'une poussée intervienne durant notre minute de calibrage !

Notre procédure de détection écrit ensuite toutes les 10 secondes dans notre fichier de données les valeurs relevées. En cas de détection, la mention "*poussée détectée*" s'inscrit, suivie d'une animation sur l'afficheur à LED.

Le seuil de 0,01 que nous avons fixé d'après les relevés est un seuil très bas : nos tests nous ont montré qu'un très léger choc sur l'AstroPi déclenchait la détection de poussée ! Nous pourrions donc aussi détecter les chocs involontaires des astronautes sur l'AstroPi...

III– EASTER EGGS

Nous avons très vite souhaité agrémenter notre programme de petites surprises, semblables aux fonctionnalités cachées de certains logiciels, appelées Easter Eggs.

1. Anniversaires

Un groupe d'élèves s'est occupé de relever les dates d'anniversaires des 6 astronautes actuellement présents dans l'ISS. Lorsque la date du jour (récupérée grâce au module `time` de Python) est égale à la date de naissance d'un des astronautes, un message apparaît sur l'afficheur LCD, suivi du drapeau du pays d'origine de l'astronaute.

explication du code par Manon <https://youtu.be/M1CHsF3WAM8>

2. Lever et coucher du soleil à Rome

Un groupe d'élèves a eu l'idée de faire afficher une animation de lever et coucher du soleil. Mais à quelle heure ? 16 fois par jour, comme sur l'ISS, ou aux heures d'une ville précise sur Terre ? Nous avons décidé de choisir une ville sur Terre. Au départ fixés sur Dieppe, puisque T.Pesquet venait de tweeter une photo de la ville où il fut lycéen, nous avons changé d'avis lorsque nous avons compris que la compagne de T.Pesquet habitait Rome.

Notre professeur a donc récupéré sur internet dans un fichier `rome.csv` les heures de lever et coucher du soleil à Rome pour l'année 2017. Grâce au module `csv` de Python, notre programme parcourt ce fichier de données à la recherche de la date du jour, et stocke ensuite dans deux variables `Hlever` et `Hcoucher` les heures correspondantes.

Dans la boucle principale de notre code, il ne nous reste plus qu'à comparer toutes les minutes l'heure actuelle à ces deux variables pour déclencher éventuellement les animations de lever ou coucher du soleil.

3. Position de l'ISS

Nous avons été impressionnés par le code proposé l'année dernière par l'école Thirsk School (Royaume-Uni), qui permettait à l'AstroPi de calculer en temps réel sa position grâce au module `ephem`.

Nous avons donc repris leur idée et, grâce à la « carte de visite » TLE de l'ISS récupérée fréquemment sur le site celestrak.com, nous avons été capables d'obtenir deux variables `latitude` et `longitude`.

Grâce à [Google Maps](https://www.google.com/maps) nous avons alors délimité les coordonnées de deux rectangles, pour la France et pour l'Italie. Le test de survol de la France se fait alors par exemple par :

```
if 41<int(latitude)<49 and 0<int(longitude)<9:
```

Ces tests déclenchent des animations spécifiques aux deux pays.

explication du code par Tom : <https://youtu.be/LeeHrmD9yEY>

IV– BILAN DU PROJET

Tout d'abord, nous avons dû nous plonger dans l'ISS ! À l'aide de notre professeur et du professeur d'Anglais d'une partie de la classe, nous avons étudié les conditions de vie dans l'ISS, l'histoire de la Station Spatiale, ses différents modules... Les nombreux articles sur internet et le compte Twitter de Thomas Pesquet nous ont beaucoup aidé et nous ont bien motivés.

Le code en Python (que nous avons débuté avec ce projet) a au début souvent nécessité l'aide de notre professeur, mais nous avons peu à peu gagné en autonomie.

Nous avons beaucoup aimé la partie « communication » de ce projet, par notre compte Twitter, nos actions au sein du lycée ou même la visite du journal Sud-Ouest qui a fait un grand article sur notre participation à ce concours.

Nous sommes très contents de notre code final, qui marche, et ceci est déjà une victoire ! Même si l'aventure s'arrête peut-être ici, ce projet restera une très grande réussite pour nous et notre professeur : nous continuerons à regarder passer l'ISS dans le ciel en ayant l'impression de la connaître un peu mieux que les autres...

déroulement du programme final : <https://youtu.be/saWXrP1HzSY>