



Édition 2025

PRÉSENTATION DU PROJET

MICROTAMAGOTCHI
ON MICROBIT V2

Nom de votre projet	MicroTamagotchi
Membre de l'équipe n°1 (prénom/nom)	Killian NALLET

Membre de l'équipe n°2 (prénom/nom)	Mattéo MARTIN-BOILEUX
Membre de l'équipe N°3 (prénom/nom)	
Membre de l'équipe n°4 (prénom/nom)	
Membre de l'équipe n°5 (prénom/nom)	
Niveau d'étude (première ou terminale)	Première
Établissement scolaire	Lycée Camille Claudel, Vaureal
Responsable du dépôt (professeur de NSI)	Céline DELAGE

1 / PRÉSENTATION GÉNÉRALE

Ce projet a pour but de recréer un tamagotchi (petit personnage virtuel avec lequel on peut interagir et qui affiche des émotions) sur le microcontrôleur micro:bit v2. La base de notre projet est née d'un TP en NSI (créer un projet avec des micro:bit interagissant entre eux) et d'une envie de participer à un concours de NSI, pour valoriser notre travail et s'amuser. La problématique générale de départ était d'imaginer un projet faisant interagir plusieurs micro:bit ensemble (avec python3) et de créer un logigramme pour l'algorithme de chaque micro:bit. Au moment de la proposition de participation au concours, nous voulions conserver notre projet initial mais nous avons fini par choisir de modifier celui-ci car il n'était pas assez étoffé. Pendant que nous cherchions une idée de projet un peu plus complexe qu'un jeu avec des leds, l'un de nous deux a pensé au tamagotchi, qui a des similitudes avec le micro:bit (3 boutons disponibles, écran avec des pixels et la taille). Et nous avons repris notre projet du jeu avec une bande de leds neopixels pour l'inclure à un petit personnage virtuel avec lequel on peut interagir.

2 / ORGANISATION DU TRAVAIL

Répartition des tâches :

- Killian NALLET : Passionné de robotique, dev python sur ses temps libres, en option NSI. A développé principalement l'application MicroTamagotchi_Tool (le Microbit_Backend pour communiquer avec le micro:bit, la disposition des éléments de l'application, les fonctionnalités de l'interface graphique et l'adaptation du simulateur de micro:bit).
- Mattéo MARTIN-BOILEUX : Passionné de Japonais, de dessin et de Jeux Vidéos, en option NSI. A développé en grande partie le programme du micro:bit (la gestion des événements, les animations associées la communication entre les cartes Microbit et le stockage des données).

Toutes ces tâches ont été réalisées en fonction des compétences et du temps disponible de chacun.

Nous avons passé en tout 4 semaines sur ce projet (et il nous a manqué un peu de temps pour le finir). Ce court temps de développement est lié à la proposition tardive de participer au concours Trophées NSI 2025.

Les outils que nous avons utilisés pour le projet sont les suivants :

- Visual Studio Code : pour le développement de MicroTamagotchi_Tool
- Thonny IDE : pour le développement du programme du MicroTamagotchi
- <https://python.microbit.org/v/3/>: pour flasher le firmware micropython le plus récent sur le micro:bit (pour Thonny IDE)
- Les Emails : nous avons hésité à utiliser GitHub pour se partager le code mais pas manque de connaissance sur ce vaste outil, nous nous sommes contentés d'utiliser des emails pour se partager des fichiers .zip avec le code et la répartition des tâches restantes
-

3 / ÉTAPES DU PROJET

Les Etapes :

- TP de NSI (2 mois avant la nouvelle idée) : création d'un jeu avec une bande de neopixels et des micro:bit
- Remaniement du projet : tamagotchi
- Création d'un premier personnage affichable sur l'écran du micro:bit : pas une image fixe de toute la matrice de leds où chaque position du personnage est définie par une image, mais une liste de pixels à afficher (ex : [5,9,5, 9,9,9, 5,9,5] size=[3,3]) en leur attribuant une valeur (intensité de la led) et une position dans la liste avec la taille (en x et en y). Dans l'exemple précédent, l'image affichée à l'écran du micro:bit aurait été le symbole '+', de largeur 3 et de longueur 3.
- Idée d'une application permettant d'installer le programme MicroTamagotchi dans le micro:bit et de pouvoir le customiser les personnages du MicroTamagotchi
- Version Beta de MicroTamagotchi_Tool : permet de se connecter au MicroTamagotchi et de lui envoyer des fichiers de données (des images des personnages) avec l'outil pyboard.py
- Création de la gestion des événements du MicroTamagotchi : changement de position (x, y) de l'image en fonctions de variables et création d'un menu
- Version Alpha de l'application lui permettant de flasher le projet (un fichier .hex de micropython et de downloader les autres fichiers) dans le micro:bit (et aussi pour le reinitialiser), stockage des configurations sous format .json
- Rencontre d'un gros problème avec le module json : pas compatible avec micropython sur micro:bit (manque le module 're' pour fonctionner dedans ...), mais solution trouvée : fonction 'repr' et 'eval' permettant de stocker les 'dict' sous forme de 'str' dans un fichier et de les retransformer en 'dict' après

- Version Finale de MicroTamagotchi_Tool lui permettant de se connecter au MicroTamagotchi, de downloader / uploader des fichiers de données, de créer des personnages depuis un panneau de son interface graphique.
- Version finale du programme du MicroTamagotchi, lui permettant aussi de s'endormir après de l'inactivité et de jouer des sons.

4 / FONCTIONNEMENT ET OPÉRATIONNALITÉ

Au moment du dépôt du projet, celui-ci est presque terminé. Il manque juste des fonctionnalités dans le menu du MicroTamagotchi (manque de temps). Il manque aussi la fonctionnalité communiquer / détecter d'autres MicroTamagotchi avec la radio (qui utilise le matériel bluetooth) pour interagir avec eux (mode love, mode échange de place).

Pour éviter les bugs, nous avons choisi de nous occuper chacun d'une partie distincte du projet de notre côté pour ne pas trop avoir à travailler sur le même code l'un après l'autre (même si cela a été pratique d'envoyer des archives .zip avec l'avancement de notre partie de projet pour que l'autre puisse s'assurer que son programme est bien compatible avec), et de bien documenter nos fonctions avec des commentaires partout pour une meilleure compréhension du code et un gain de temps à la compréhension.

Pour garantir une facilité d'utilisation de notre projet, nous avons regardé les versions des modules que nous utilisons et la version de python associée pour trouver la bonne version de python et de ses librairies à partir de laquelle le projet fonctionne.

Nous avons rencontré pas mal de problèmes et difficultés à résoudre pendant le développement de notre projet. Par exemple le fait que le micro:bit ne possède qu'une matrice de 5x5 leds, qui empêche d'afficher des personnages complexes avec des expressions sur la matrice. La solution a été de créer des petit personnages (du genre carré, d'une taille variable), avec des pixels d'intensité variable pour les reconnaître et d'afficher un visage (à la place du personnage) exprimant une émotion particulière quand le celui-ci en a besoin (pendant un court instant). Il y a aussi le problème de compatibilité de la librairie json avec micropython, qui a empêché l'interprétation des données reçues par le microcontrôleur (car on pensait que cela fonctionnerait sur le micro:bit mais la version de python présente dessus ne le permet pas), qui a été résolu en trouvant des fonctions remplaçantes, natives à python3 et micropython (les fonctions 'repr' et 'eval') qui ont permis de résoudre le problème en permettant l'utilisation des données sans changer complètement leur structure.

5 / OUVERTURE

Les nouvelles fonctionnalités que nous pouvons apporter sont : (moyen terme)

- Ajouter des fonctionnalités d'interaction supplémentaires pour le MicroTamagotchi (communiquer avec d'autres, échange de place ...)
- Personnage du MicroTamagotchi qui grandit

Nous pouvons améliorer notre projet en :

- Ajoutant plus de personnages et d'émotion qu'il peut afficher
- Adapter le projet pour le micro:bit v1 (trouver une solution pour remplacer le pinlogo du v2, par exemple, si la luminosité reçue par l'écran de leds est à zéro ...)

Si le projet était à refaire, il nous faudrait une meilleure organisation :

- *logicielle*, en apprenant à utiliser GitHub pour développer ce genre de projet,
- mais aussi *temporelle*, en prévoyant aussi des moments (pendant certaines heures de permanence par exemple) pour travailler ensemble sur un problème rencontré (et en étant prévenu plus tôt pour participer au concours).

Par exemple, le problème de compatibilité de la librairie json nous a fait perdre du temps car nous avons chacun essayé de résoudre le problème de chacun notre côté en faisant à peu près les mêmes recherches et retrouvant les mêmes problèmes (comme chercher la librairie micropython json/ujson sur le net, la trouver mais qui dépends du module 're' non disponible sur le micro:bit).

Grâce à ce concours, nous avons appris à un peu mieux gérer le travail d'équipe, nous avons appris à utiliser des nouveaux modules, comme 'customtkinter' et à utiliser le matériel et les fonctions de la carte micro:bit. Ce projet nous a aussi permis de consolider nos compétences sur certains types de données en python (comme 'dict' ou 'list') et pris l'habitude de documenter notre code de façon claire en d'anglais.

En faisant notre projet, lors de problèmes rencontrés, nous n'avons pas hésité à partager les problèmes trop compliqués avec notre coéquipier pour les résoudre, ce qui participe à la démarche d'inclusion.