

Mise à jour :

Objectif : Préparer aux étudiants de I1 de l'année prochaine un manuel qui va les aider à utiliser Scratch pour programmer le robot pour répondre à un cahier de charge.

Enoncée :

1. Utiliser Python pour programmer des boîtes scratch associées aux fonctions déjà existant comme *avancer*, *tourner*, etc et les exécuter sur le robot à partir de Scratch directement.
2. Utiliser Python pour programmer des boîtes scratch pour de nouvelles fonctions (reconnaissance visuelle, vocale, etc) et les exécuter sur le robot à partir de Scratch directement.

Etapes :

- **Installations :** Python, VS Code, Scratch, Scratch Link, SDK RoboMaster.
Scratch Link permet de relier Scratch et Python.
 - Installer Python : <https://www.python.org/downloads/release/python-378/>
 - Installer VS Code : <https://code.visualstudio.com/>
 - Configurer VS Code pour Python : VS Code -> Extensions -> Recherche "**Python**" et installe l'**extension officielle de Microsoft** -> Fermer et relancer VS Code
 - Installer ScratchLink : <https://downloads.scratch.mit.edu/Scratch-Link-Win.exe>
 - Installer Scratch : <https://scratch.mit.edu/download>
 - Installer les bibliothèques nécessaires (cmd) :
`pip install robomaster flask websockets`
 - Installer le SDK RoboMaster :
https://robomaster-dev.readthedocs.io/en/latest/python_sdk/installs.html
(déjà installé ! A vérifier !)
 - Connecter le RoboMaster au PC
- **Définir l'extension Scratch (fichier JSON) :** Les fichiers JSON décrivent les **blocs Scratch personnalisés**.
 - Dans VS Code, créer un fichier JSON pour notre extension (`robomaster_extension.json`, ce fichier contient la définition des blocs Scratch)
 - Ajouter l'extension à Scratch (Extensions -> Ajouter une extension personnalisée -> Entrer l'URL `ws://localhost:8765/`)
 - **Un seul fichier JSON** peut contenir **plusieurs blocs Scratch**.
- **Créer un serveur WebSockets en Python et Lier Scratch et Python via WebSockets :**
Créer un **serveur WebSockets** en Python qui **écoute les commandes Scratch** et les envoie au RoboMaster.
 - Installer les bibliothèques nécessaires : `pip install websockets robomaster`
 - Créer le serveur WebSockets en Python : Dans VS Code, créer un fichier `server.py`
 - Lancer le serveur WebSockets : `python server.py` (Dans VS Code)
- **Test :** Scratch -> WebSockets -> Python -> RoboMaster.
 - Lancer le serveur WebSockets : `python server.py`
 - Revenir sur Scratch, glisser le bloc "Avancer de X mètres" par exemple, dans le script.
 - Changer la valeur (ex : "**Avancer de 1 mètre**").
 - Cliquer sur le bloc pour l'exécuter.

Fichiers annexes :

Fichier `servy.py` :

```
import asyncio
import websockets
from robomaster import robot
# Initialisation du RoboMaster
ep_robot = robot.Robot()
ep_robot.initialize(conn_type="ap") # Connexion en Wi-Fi direct au robot
async def handle_command(websocket, path):
    async for message in websocket:
        print(f"Commande reçue : {message}")
        if message.startswith("move_forward"):
            try:
                # Extraire la distance depuis la commande
                _, distance = message.split(":")
                distance = float(distance)
                # Convertir la distance en déplacement (1 mètre = 1)
                ep_robot.chassis.move(x=distance, y=0, z=0, xy_speed=0.5).wait_for_completed()
                print(f"Le robot avance de {distance} mètre(s)")
                await websocket.send("ok") # Confirmer à Scratch que l'action est terminée
            except Exception as e:
                await websocket.send(f"error:{str(e)}")
# Démarrer le serveur WebSockets
start_server = websockets.serve(handle_command, "localhost", 8765)
print("Serveur WebSockets en attente de connexions...")
asyncio.get_event_loop().run_until_complete(start_server)
asyncio.get_event_loop().run_forever()
```

Fichier `robomaster_extension.json` :

```
{
  "id": "robomaster",
  "name": "RoboMaster EP Core",
  "blocks": [
    {
      "opcode": "move_forward",
      "blockType": "command",
      "text": "Avancer de [DISTANCE] mètre(s)",
      "arguments": {
        "DISTANCE": {
          "type": "number",
          "defaultValue": 1
        }
      }
    }
  ],
  "menus": {},
  "extensionURL": "ws://localhost:8765/"
}
```