Thème : informatique embarquée et objets connectés

TI-83 Premium CE
EDITION PYTHON

Classe enseignée : 2nde SNT

Mélange de couleurs

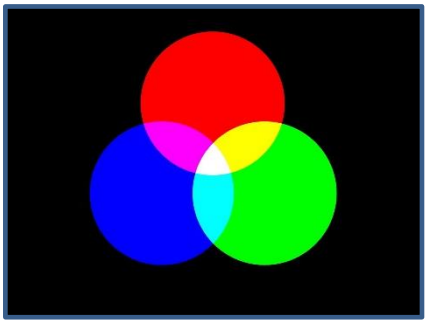
F. GIROD

Compétences visées

Un des objectifs de l'enseignement de SNT est de développer et de coder des scripts PYTHON afin d'apporter une réponse à une problématique précise. A travers le thème "**informatique embarquée et objets connectés**", nous pouvons notamment travailler les compétences suivantes dans l'activité proposée :

- Coder des scripts simples d'acquisition de données.
- Gérer des entrées/sorties à travers les ports utilisés par le système.
- Écrire et développer des algorithmes pour résoudre une problématique.
- Identifier des algorithmes de contrôle des comportements physiques à travers les données des capteurs.

Situation déclenchante



Comment créer et modifier facilement des couleurs en utilisant le code RGB ?

Ici, une **vidéo** illustrant ce projet.

Problématique

Comment modifier un code RGB facilement et en voir les effets immédiats ?

1. Identifier un capteur possible ;
2. Identifier un (des) actionneur(s) possible(s) ;
3. Élaborer un algorithme modélisant la création de couleurs en code RGB (au moins deux couleurs parmi les trois) ;
4. Mettre cet algorithme en application.

Matériel disponible

Les élèves disposent, en plus de leur TI-83 Premium CE :

- d'un TI-Innovator HUB ;
- de deux capteurs de distance à ultrasons.

Thème : informatique embarquée et objets connectés

TI-83 Premium CE
EDITION PYTHON

Classe enseignée : 2nde SNT

Fiche méthode

F. GIROD

Déroulement possible du projet

En amont du projet :

Les propositions suivantes permettent de préparer le projet et de le rendre possible sur 2 séances d'1h30. Elles peuvent aussi être préparées par l'enseignant sous forme de fiches et/ou d'un exposé au tableau.

- Un groupe d'élèves peut préparer une fiche ou présenter un rapide exposé illustrant les principaux points pour maîtriser la programmation en PYTHON spécifique au **TI-Innovator HUB** en s'appuyant sur les '**10 mn de code**' (Unité 6).
- Un groupe d'élèves peut expliquer le **mode de codage RGB** pour les couleurs.
- Un groupe d'élèves peut décrire le **TI-Innovator HUB**.

Différentes évolutions possibles pour le projet :

Les propositions suivantes donnent des pistes pour gérer les différences de vitesse d'exécution des élèves.
On peut aussi considérer ces différentes étapes comme une manière de séquencer le projet pour ceux qui en auraient besoin.

- 1ère étape : on mesure la distance entre sa main et un premier capteur à ultrasons ; même chose avec un second capteur.
- 2ème étape : on utilise ces données en les transformant en un nombre entier compris entre 0 et 255.
- 3ème étape : on produit une image RG ou RB ou GB avec les deux valeurs précédemment trouvées.

A la suite du projet :

Synthétiser ce travail par rapport au cours de SNT : le(s) capteur(s) et le(s) actionneur(s) mis en jeu / l'algorithme qui gère les données d'entrée / le langage de programmation utilisé.

Proposition de résolutions

Choix du capteur :

- Capteur de distance à ultrasons : en passant sa main à proximité, la distance mesurée nous permet de modifier la valeur de R, G ou B.

  Remarque : on ne peut utiliser que deux capteurs car seules les entrées IN 1 et IN 2 permettent de brancher un tel capteur.

Choix des actionneurs :

- LED RGB du HUB : elle pourra délivrer une lumière de la couleur de son choix ; c'est cet actionneur qui est choisi ici.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus !

Thème : informatique embarquée et objets connectés

TI-83 Premium CE
EDITION PYTHON

Classe enseignée : 2nde SNT

Fiche méthode

F. GIROD

Etapes de résolution

L'importation de ces bibliothèques est explicitée dans la remarque qui suit cette partie.

```
ÉDITEUR : COULEURS
LIGNE DU SCRIPT 0011
# Projets STEM Hub
from ti_system import *
from time import *

import color
from ranger import *
```

`r1=ranger("IN 1")` crée la variable **r1** qui est de type 'ranger'.
r1 et **r2** représentent les entrées des deux rangers (capteurs de distance) reliés aux ports d'entrée **IN 1** et **IN 2**.
Les instructions se trouvent dans le module *ranger* importé précédemment.

```
ÉDITEUR : COULEURS
LIGNE DU SCRIPT 0001
# Projets STEM Hub
from ti_system import *
from time import *

import color
from ranger import *

r1=ranger("IN 1")
r2=ranger("IN 2")
```

- `while not escape():` crée une boucle 'infinie' qui sera interrompue par l'appui sur la touche **on** ; l'instruction se trouve dans le module *ti_system*.
- `m1=r1.measurement()` crée la variable **m1** qui stocke la mesure donnée par **r1** (le ranger branché à l'entrée 1).
- La variable **r** permet de transformer la valeur **m1** en un nombre entier compris entre 0 et 255 ; de même pour **b** avec la valeur de **m2**.
- L'instruction **color.rgb(r,0,b)** permet de colorer la led avec les paramètres **r** et **b**. Cette fonction se trouve dans la bibliothèque *color* importée en préambule.
  - En mettant un obstacle près du ranger branché en **IN 1**, on rend la valeur proche **r** de 0 : la led est proche d'un bleu pur.
  - En mettant un obstacle près du ranger branché en **IN 2**, on rend la valeur proche **b** de 0 : la led est proche d'un rouge pur.
  - En mettant un obstacle à une distance équivalente des deux rangers, on obtient des violets.

```
ÉDITEUR : COULEURS
LIGNE DU SCRIPT 0022
while not escape():
  m1=r1.measurement()
  m2=r2.measurement()
  #print("m1=",m1)
  #print("m2=",m2)
  r=min(255,int(255/0.5*m1))
  b=min(255,int(255/0.5*m2))
  #print("r=",r)
  #print("b=",b)
  color.rgb(r,0,b)
  #sleep(1)
```

- Pour faciliter la compréhension du script en cours de réalisation, on peut proposer des affichages.
  - Affichage des valeurs de **m1**, **m2**, **r** et **b**. Si les valeurs de **r** et **b** ne conviennent pas, on peut modifier à sa guise les lignes de calculs correspondantes.
  - Pour que l'affichage ait le temps d'être lu, on peut proposer une pause par l'instruction **sleep(1)** (pause d'une seconde) ; cette instruction se trouve dans la bibliothèque *time*.
  - Pour commenter ces instructions par la suite, on insère un **#** (par exemple par le raccourci : touche **2nde** suivi de la touche **3**).

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus

Thème : informatique embarquée et objets connectés

TI-83 Premium CE
EDITION PYTHON

Classe enseignée : 2nde SNT

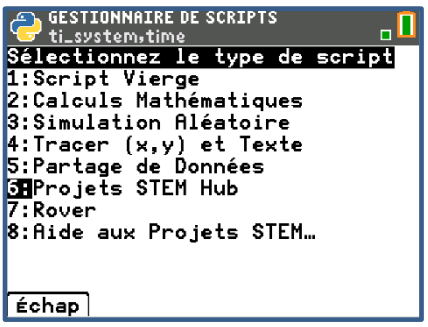
Fiche méthode

F. GIROD

Remarques

En préambule du code, on trouve l'importation des bibliothèques spécifiques au projet :

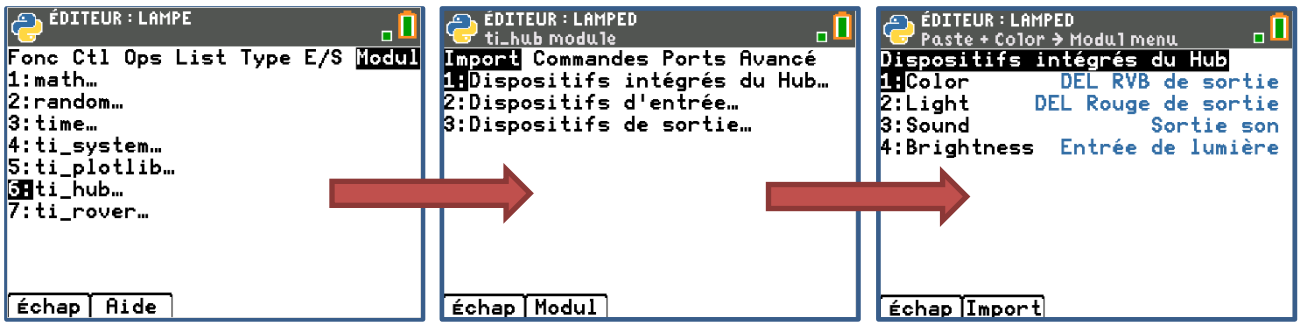
- elles peuvent être importées une à une en allant les chercher dans le menu **Modul** ;
- elles peuvent s'implémenter en choisissant le type de programme « **Projets STEM Hub** » (choix 6 des types de programmes proposés) à la création du nouveau programme.



La bibliothèque *ti_system* permet d'importer ou d'exporter des données dans les listes de la calculatrice.
Pour ce projet, elle contient l'instruction **while not escape() :** qui permet de lancer une boucle qui ne s'interrompt que si on appuie sur la touche **on**.

La bibliothèque *time* permet en particulier d'utiliser la fonction **sleep** qui prend comme argument un temps de pause donné en secondes.

On va ensuite importer une bibliothèque spécifique au projet, à savoir la bibliothèque *color* en suivant les instructions suivantes :



Au fur et à mesure que les bibliothèques sont importées, elles s'ajoutent aux bibliothèques déjà présentes (*math*, *random*, …etc.) et permettent d'accéder aux instructions qu'elles contiennent.
On procède de même pour importer la bibliothèque *ranger* qui s'obtiendra en choisissant '**dispositifs d'entrée**' à l'étape 2 de la procédure précédente.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus