

Tracé de polygones réguliers

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 :

- Écrire et développer des algorithmes pour répondre à des problèmes.
- Faire preuve d'autonomie, d'initiative et de créativité.
- Coopérer au sein d'une équipe.
- Rechercher de l'information, apprendre à utiliser des sources de qualité, partager des ressources.
- Présenter un problème ou sa solution, développer une argumentation dans le cadre d'un débat.

Situation déclenchante



Dans la vie de tous les jours, nous sommes entourés de formes géométriques : bâtiments, œuvres d'art, ...etc. Dans certains cas, il s'agit de polygones réguliers. Mais qu'est-ce qu'un polygone régulier ? Comment le tracer ?

Problématique

Comment programmer le robot Rover pour qu'il puisse tracer un polygone régulier au sol et sur l'écran de la calculatrice, connaissant le nombre de côtés et la longueur d'un côté ?

Fiche méthode

Matériel nécessaire



- Calculatrice TI-83 Premium CE
- Câble (calculatrice/Hub)
- TI-Innovator Hub
- Rover
- Marqueur

Déroulement possible du projet

Travail de groupe possible. Chaque groupe disposera du matériel ci-dessus et devra concevoir une réponse à la problématique sous forme d'une démonstration munie d'une documentation qui explique les scripts PYTHON pilotant le rover. A la fin du projet, les groupes pourront voter pour élire la production qui répond le mieux à la problématique.

La production devra répondre aux critères suivants :

Critère 1 : Définir succinctement un polygone régulier et ses principales propriétés.

Critère 2 : Le rover devra tracer un polygone au sol connaissant le nombre de côtés et la longueur d'un côté.

Critère 3 : Tracer les déplacements du Rover sur l'écran de la calculatrice.

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



Fiche méthode

Proposition de résolution

La fonction polygone prend comme arguments la longueur d'un côté (notée l) et le nombre de côtés du polygone régulier (noté c).

Importation des bibliothèques nécessaires (voir remarques).

Instructions de tracé sur la calculatrice :

- L'instruction `plt.cls()` permet d'effacer l'écran.
- L'instruction `plt.axes("on")` permet d'afficher les axes.
- L'instruction `plt.title("polygone regulier a"+str(c)+"cotés")` permet d'afficher le titre.
- L'instruction `plt.pen("medium","solid")` permet de modifier le style de tracé.
- L'instruction `plt.grid(1,1,"dot")` permet d'afficher une grille.

Les instructions de déplacement.

L'instruction `rv.wait_until_done()` permet d'attendre la fin du déplacement précédent de manière à synchroniser le tracé sur la calculatrice avec le tracé au sol.

L'instruction `rv.waypoint_x()` permet de récupérer l'abscisse de la fin du déplacement.

L'instruction `plt.line(x,y,X,Y,"arrow")` permet de tracer une flèche du point de coordonnées (x,y) au point de coordonnées (X,Y).

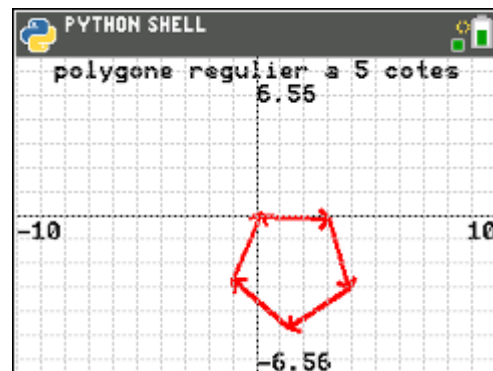
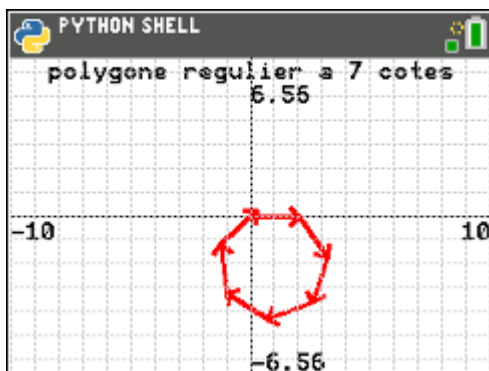
L'instruction `plt.show_plot()` permet d'afficher le graphique.

```

ÉDITEUR : POLYGONE
LIGNE DU SCRIPT 0001
# TI-Rover
from time import *
from ti_system import *
import ti_rover as rv
import ti_plotlib as plt

def polygone(l,c):
    plt.cls()
    plt.axes("on")
    plt.pen("medium","solid")
    plt.color(255,0,0)
    plt.grid(1,1,"dot")
    plt.title("polygone regulier a
              "+str(c)+" cotés")
    x=0
    y=0
    b=360/c
    for i in range(c):
        rv.forward(l)
        rv.right(b)
        rv.wait_until_done()
        X=rv.waypoint_x()
        Y=rv.waypoint_y()
        plt.line(x,y,X,Y,"arrow")
        x=X
        y=Y
    plt.show_plot()
    rv.disconnect_rv()
    
```

Exemples de représentation graphique obtenues :



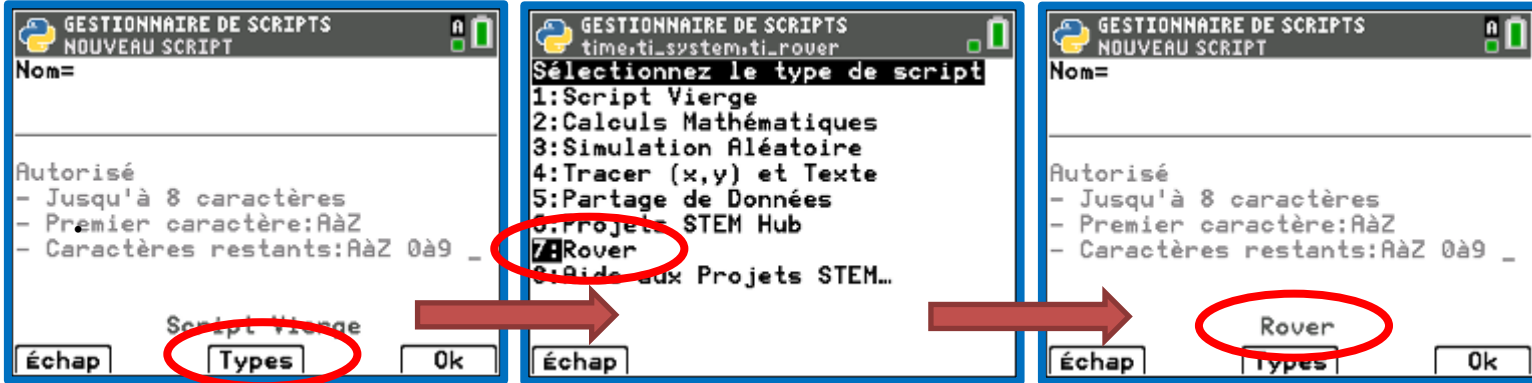
Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



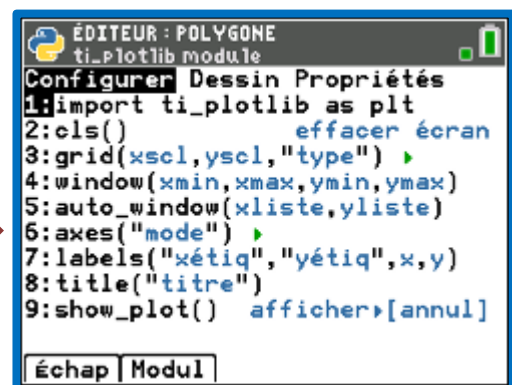
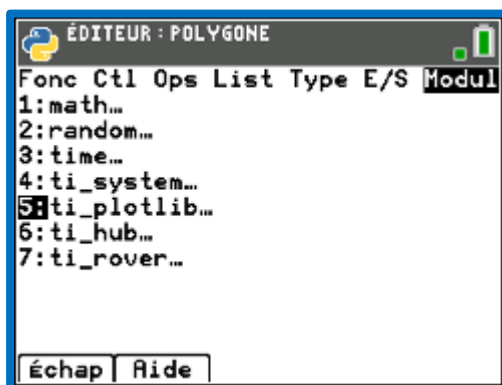
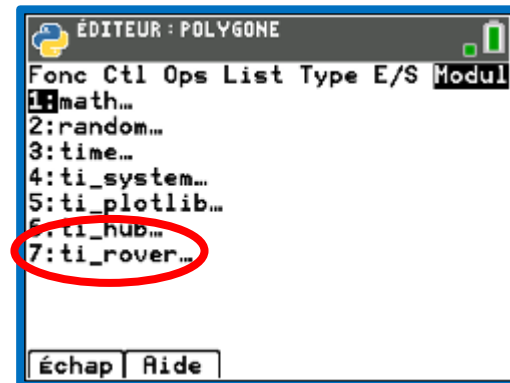
Fiche méthode

Utilisation des bibliothèques

- Lors de la création du script, sélectionner la rubrique Types, puis Rover. Cela permettra de charger automatiquement les trois bibliothèques suivantes : *time*, *ti_system* et *ti_rover*.



- Pour utiliser la bibliothèque *ti_rover* importée, appuyer sur la touche `f(x)`, sélectionner la rubrique Modul, puis sélectionner 7 : *ti_rover*...
- 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.
- Une fois le nouveau script créé, pour importer la bibliothèque *ti_plotlib*, appuyer sur la touche `f(x)`, sélectionner la rubrique Modul, puis suivre les copies d'écran suivantes :



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

