Séquence 3

Algorithme et programmation

SI

LYCÉE LÉONARD DE VINCI

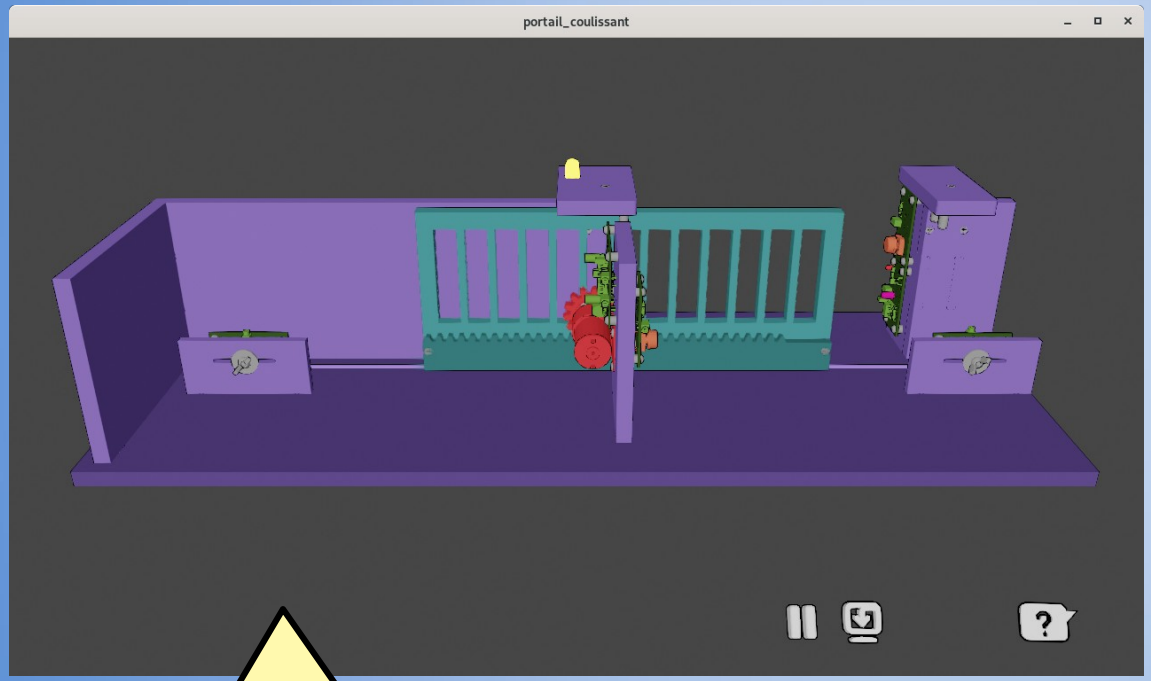
Simulateur 3D pour la programmation de système pluritechnique

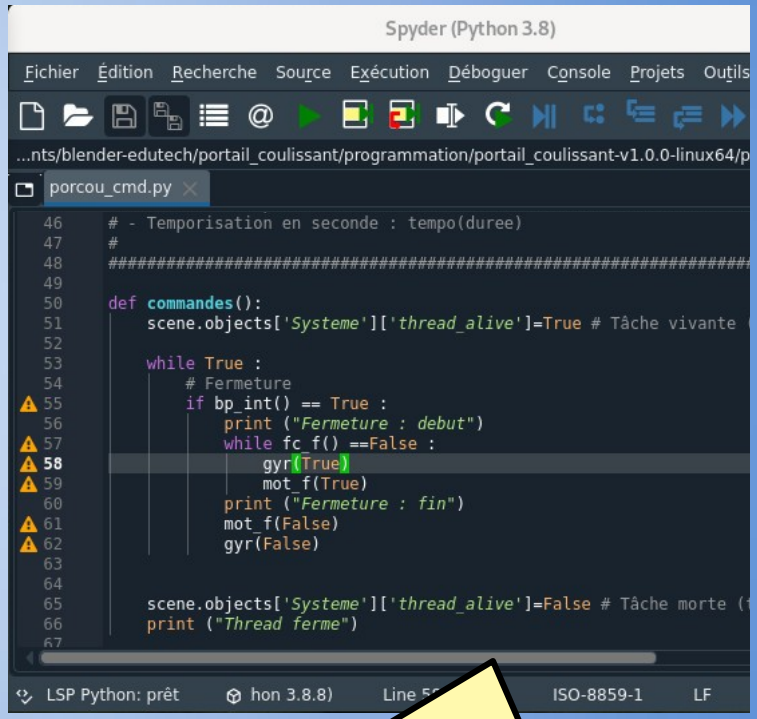
Projet Blender-EduTech (Blender/UPBGE for Technology Education)

https://gitlab.com/blender-edutech

Présentation du simulateur 3D et de son environnement de programmation

Le simulateur 3D est une maquette numérique qui se commande grâce au langage **Python**. L'interface de programmation se décompose en 2 fenêtres : un éditeur de texte et le simulateur.

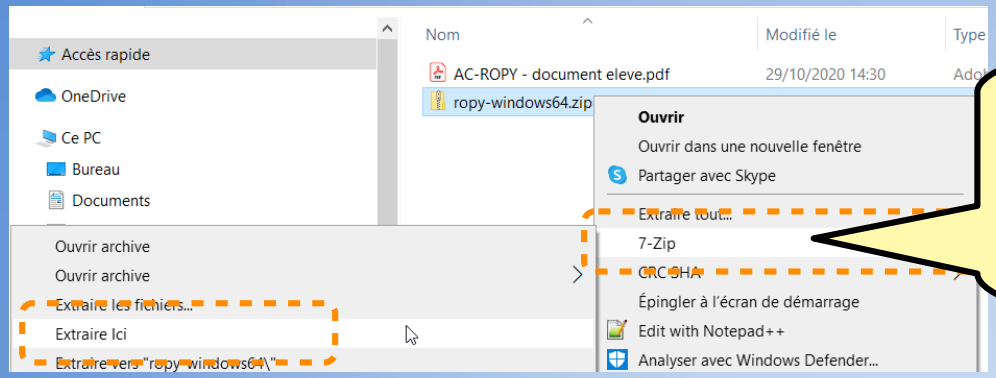




Le **simulateur** permet de **visualiser l'évolution du système**.

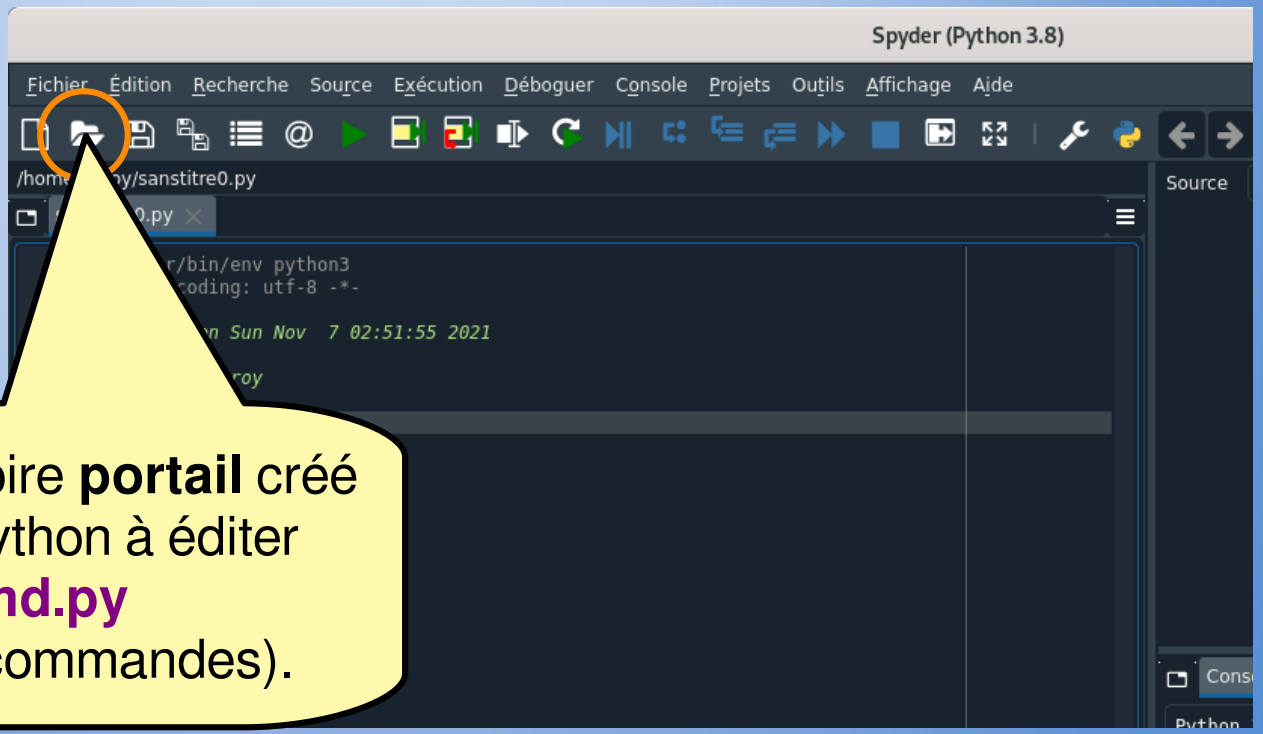
Un **éditeur de texte** (Notepad++, Spyder, Emacs, ...) pour **écrire le programme** en **Python**.

Éditer le programme avec Spyder



1 : Récupérer l'archive **portail-windows64.zip** et la décompresser avec **7-Zip** dans votre répertoire.

2 : Lancer le Logiciel **Spyder**.



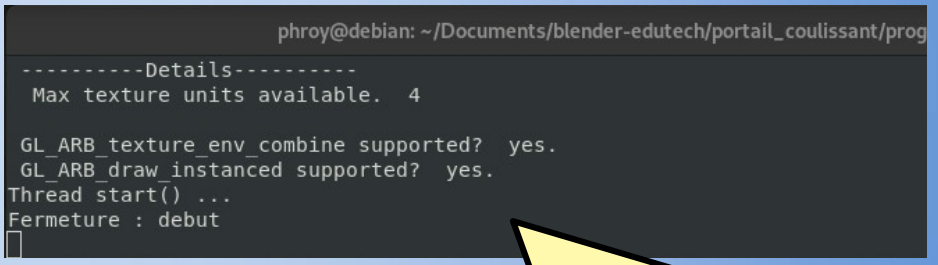
3: Aller dans le répertoire **portail** créé et ouvrir le fichier Python à éditer **porcoul_cmd.py** (Portail coulissant commandes).

Éditer le programme avec Spyder

5 : **Sauvegarder** (Ctrl-s) le fichier

Attention !
Toujours sauvegarder le fichier avant son exécution avec le simulateur.

Le simulateur est le programme **portail_coulissant.exe**

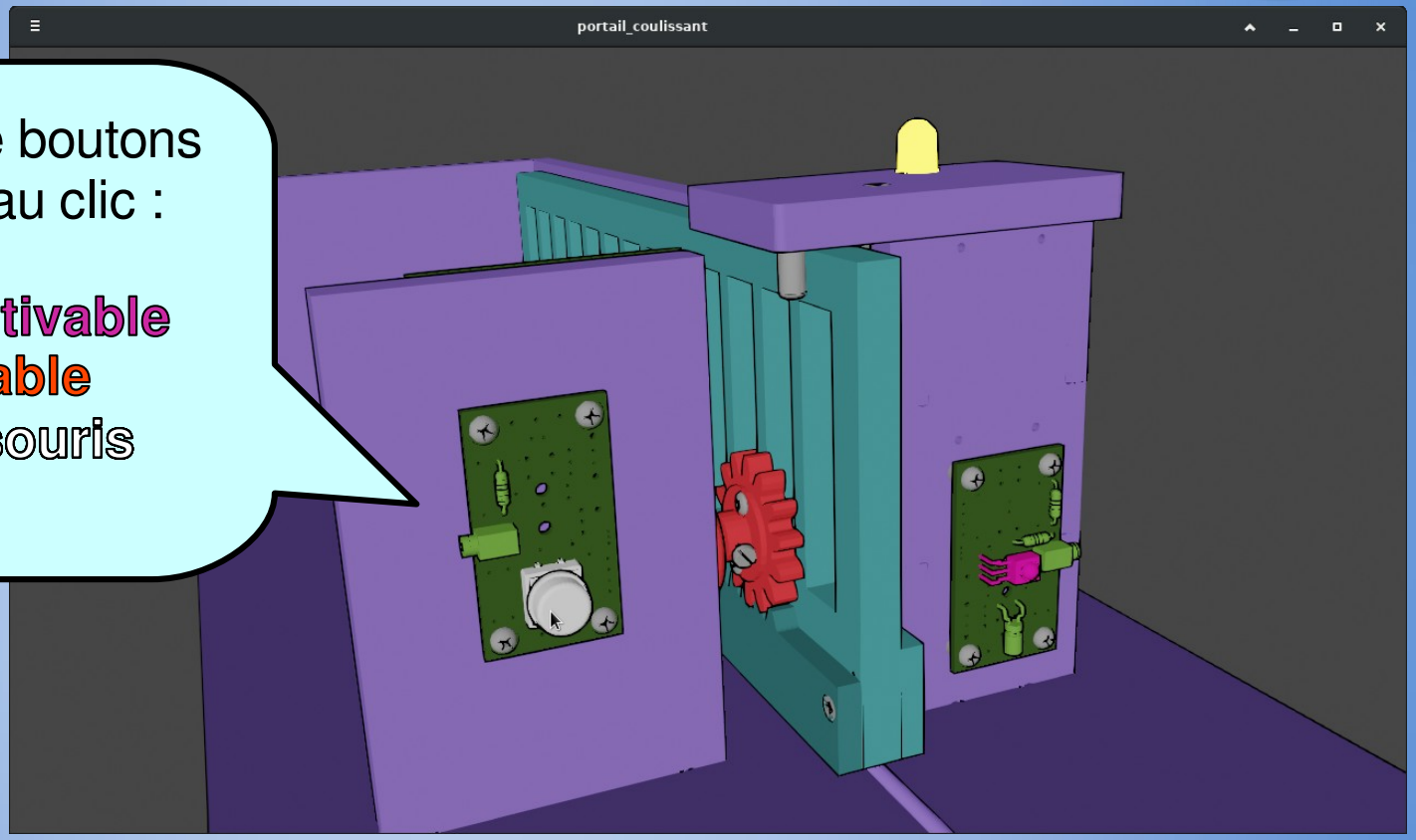


4 : **Écrire** le code Python

Ce n'est pas obligatoire, mais l'exécution du simulateur à partir d'un **terminal** permet d'afficher les **messages** de la sortie standard (**`print`**) afin de faciliter de **débogage**.

Interface du simulateur

Manipulation de la maquette numérique



Les capteurs et le boutons sont sensibles au clic :

- **Magenta : inactivable**
- **Orange : activable**
- **Blanc : focus souris**
- **Jaune : activé**

Le **bouton du centre** sert à **manipuler** le modèle 3D :

- **Clic centre** : rotation du mécanisme (orbit)
- **Clic centre + Maj** : déplacement du mécanisme (pan)
- **Clic centre + Ctrl** : zoom
- **Molette** : zoom

Carte de référence du portail coulissant

Boutons :
- Bouton poussoir coté rue : `bp_ext()`
- Bouton poussoir coté cour : `bp_int()`

Capteur de fin de course :
- Capteur portail ouvert : `fc_o()`
- Capteur portail fermé : `fc_f()`

Moteur :
- Ouvrir le portail : `mo_o(ordre)`
- Fermer le portail : `mo_f(ordre)`

Gyrophare :
- Allumé/éteindre : `gyr(ordre)`

Capteur barrage :
- Activation de l'émetteur : `ir_emet(ordre)`
- État du récepteur (absence d'obstacle) : `ir_recep()`

Valeur retournée par les capteurs et les boutons
- `True` : actif
- `False` : inactif

Ordre pour les actionneurs
- `True` : activer
- `False` : désactiver