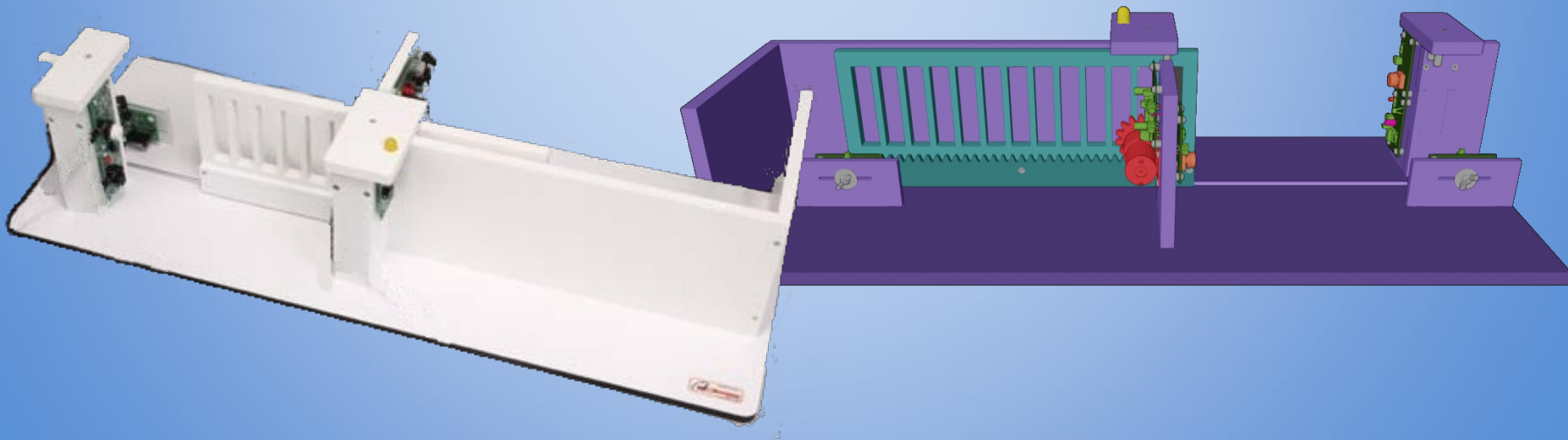


# Séquence 3

*Algorithme et programmation*

## Document Technique Jumeau numérique d'un portail coulissant



# Présentation du jumeau numérique et de son environnement de programmation

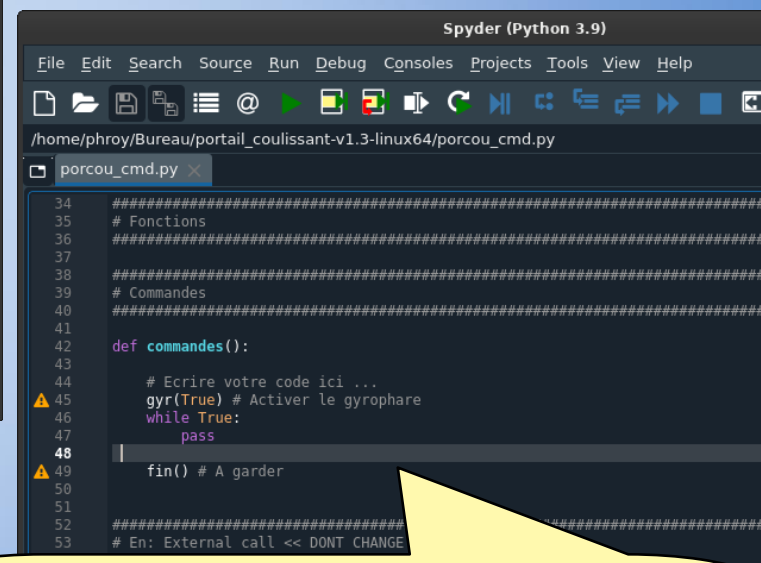
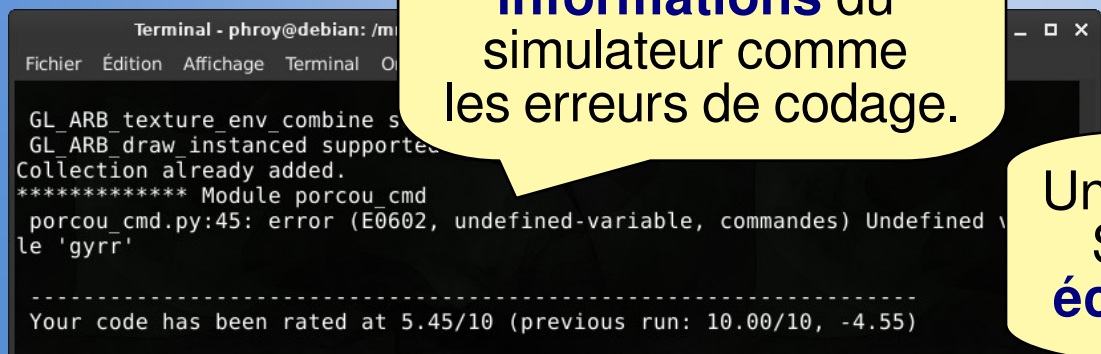


Le jumeau numérique est une maquette numérique qui se commande grâce au langage **Python**. L'interface de programmation se décompose en **3 fenêtres** : un éditeur de texte, le simulateur et la console.



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

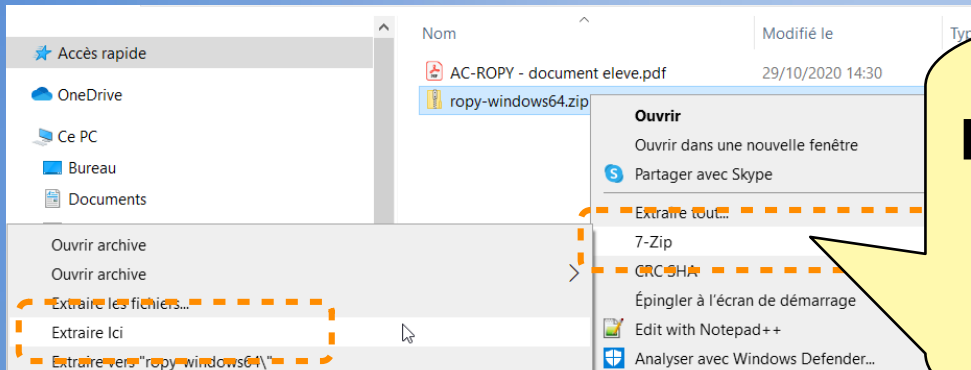
La **console** pour **visualiser les informations** du simulateur comme les **erreurs de codage**.



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

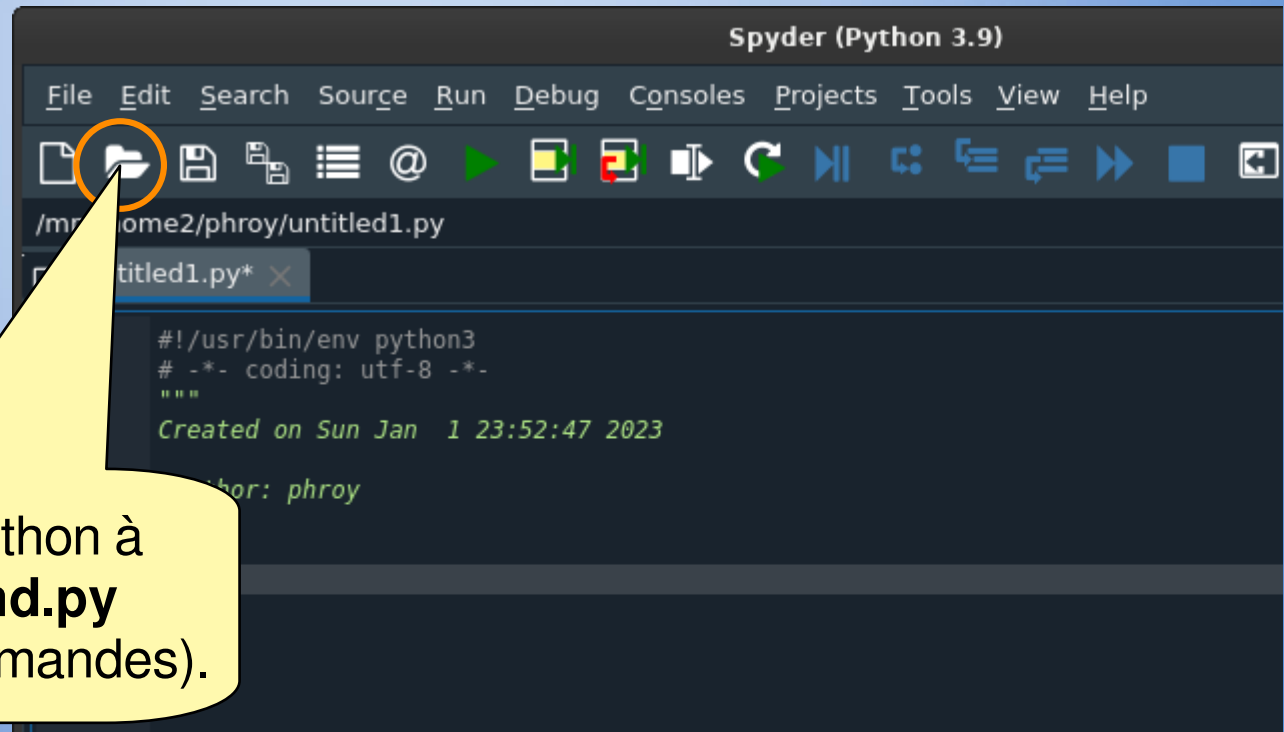
# Éditer le programme avec Spyder

## Ouvrir le fichier porcoule\_cmd.py



1 : Récupérer l'archive **portail\_coulissant-windows64.zip** et la décompresser avec **7-Zip** dans votre répertoire. L'extraction va créer le répertoire **portail\_coulissant**

2 : Lancer le logiciel **Spyder**.



3: Ouvrir le fichier Python à éditer **porcoule\_cmd.py** (Portail coulissant commandes).

# Éditer le programme avec Spyder

## Exécution du programme



5 : **Sauvegarder** le fichier

**Attention !**

Toujours sauvegarder le fichier avant son exécution avec le simulateur.

Le **simulateur** et la **console** se lancent en même temps avec **portail\_coulissant.bat**

4 : **Écrire** le code Python

**Arrêter et réinitialiser**

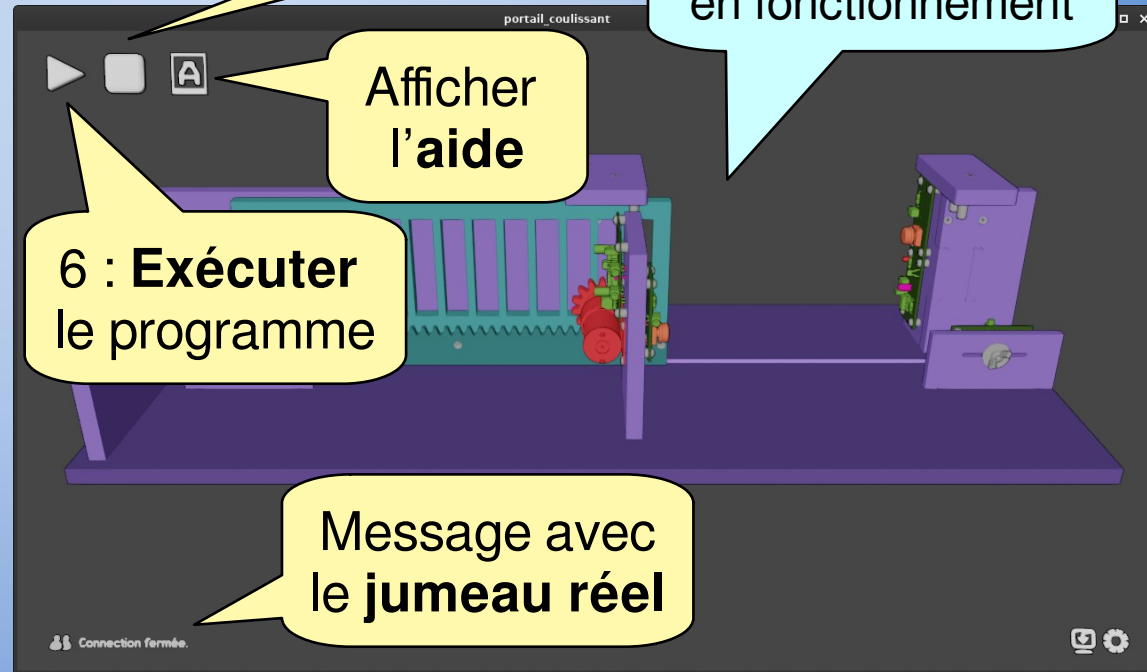
Le système en fonctionnement

**Afficher l'aide**

6 : **Exécuter** le programme

Message avec le **jumeau réel**

```
File Edit Search Settings View Help
/home/phroy/Bureau
porcou_cmd.py
16 #
17 # Capteur
18 # - Capteur
19 # - Capteur
20 # - Capteur
21 #
22 # Consignes du pupitre (valeur retournée = True)
23 # - Bouton poussoir coté rue : bp_ext()
24 # - Bouton poussoir coté cour : bp_int()
25 #
26 # Gestion du temps :
27 # - Temporisation en seconde : tempo(duree)
28 #
29 #####
30 # Fonction de gestion du temps
31 # Fonction de gestion du temps
32 # Fonction de gestion du temps
33 #####
34 # Fonction de gestion du temps
35 #####
36 # Fonction de gestion du temps
37 #####
38 #####
39 # Commandes
40 #####
41
42 def commandes():
43     # Ecrire votre code ici ...
44     gyrr(True) # Activer le gyrophare
45     while True:
46         pass
47
48
49 fin() # A garder
50
```

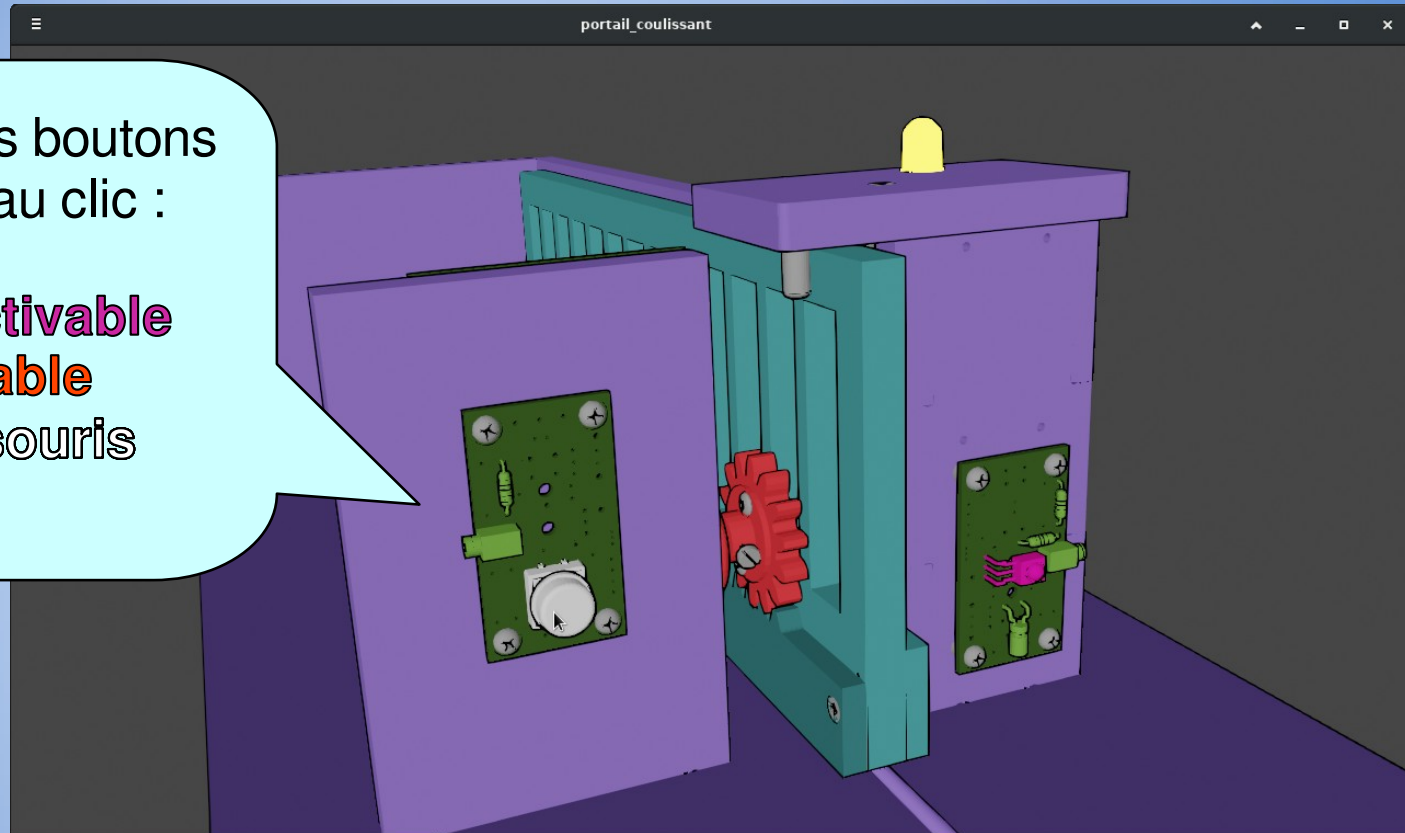


# Manipulation de la maquette numérique



Les capteurs et les 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 : `fdc_o()`
- Capteur portail fermé : `fdc_f()`

## Moteur :

- Ouvrir le portail : `mot_o(ordre)`
- Fermer le portail : `mot_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

## Activation du jumelage

- `jumeau()`