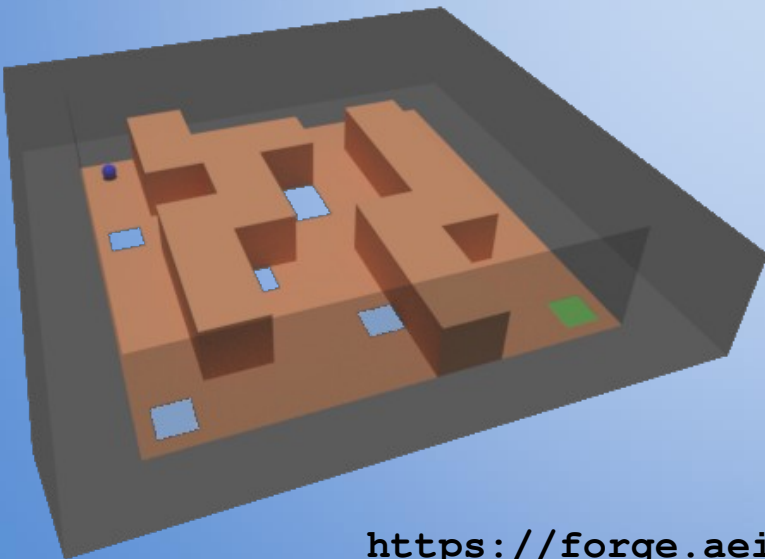


# Labyrinthe à bille

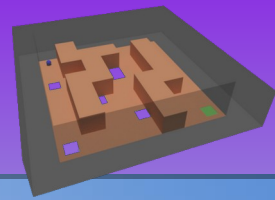
## Créer une scène 3D interactive

# Tutoriel 1

## Ma première scène



# Objectif



L'objectif de ce tutoriel est de créer un **scène animée et interactive**. Le support est le **labyrinthe à bille**. Le principe est de déplacer en rotation le plateau par le clavier afin d'amener la bille du départ à l'arrivée.

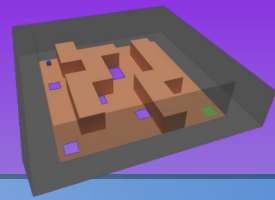
Ce tutoriel est une déclinaison pour UPBGE du projet n° 1 du livre "[Créez vos propres jeux 3D comme les pros](#)" (Éditions Graziel) de Grégory Gosselin De Bénicourt.

Le tutoriel se décompose en 8 étapes :

- Installer Blender/UPBGE
- **Modéliser** le plateau
- **Déplacer** le plateau avec les **briques logiques**
- Créer la bille et définir sa **physique**
- **Définir le game play** (règles d'échec et de réussite)
- Créer la fenêtre de fin
- **Animer** la fenêtre de fin par des **images-clés**
- Fermer la fenêtre de fin par un **bouton cliquable**

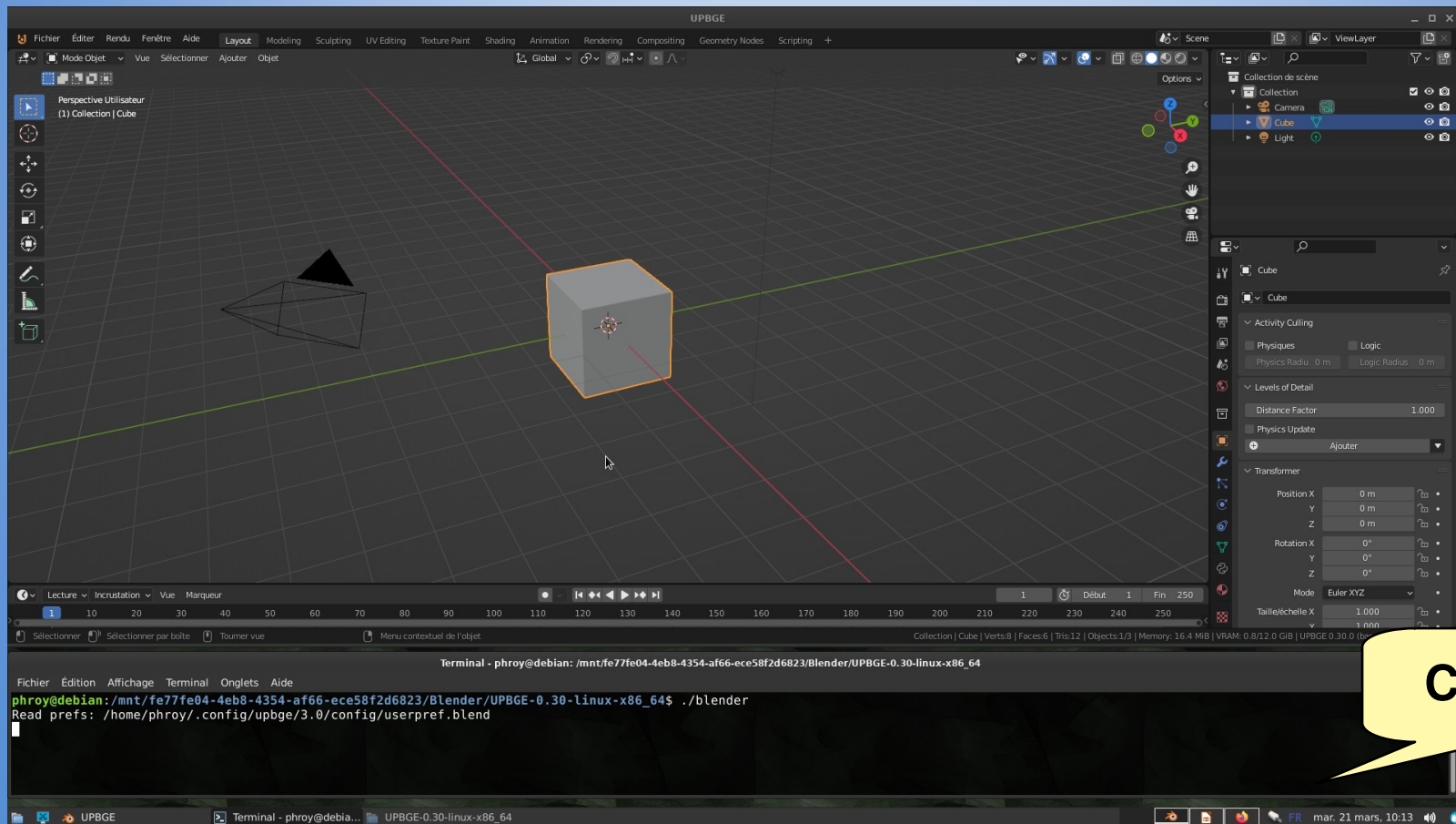


# 1. Installation de Blender/UPBGE



Le moteur de jeu **UPBGE** (**UP** Blender **G**ame **E**ngine) intègre déjà Blender, donc seule son installation suffit. Lancer UPBGE dans une **console** permet de visualiser les messages (erreurs, sortie standard, ...).

- **UPBGE** (version 0.3+) se trouve à cette adresse : <https://upbge.org>



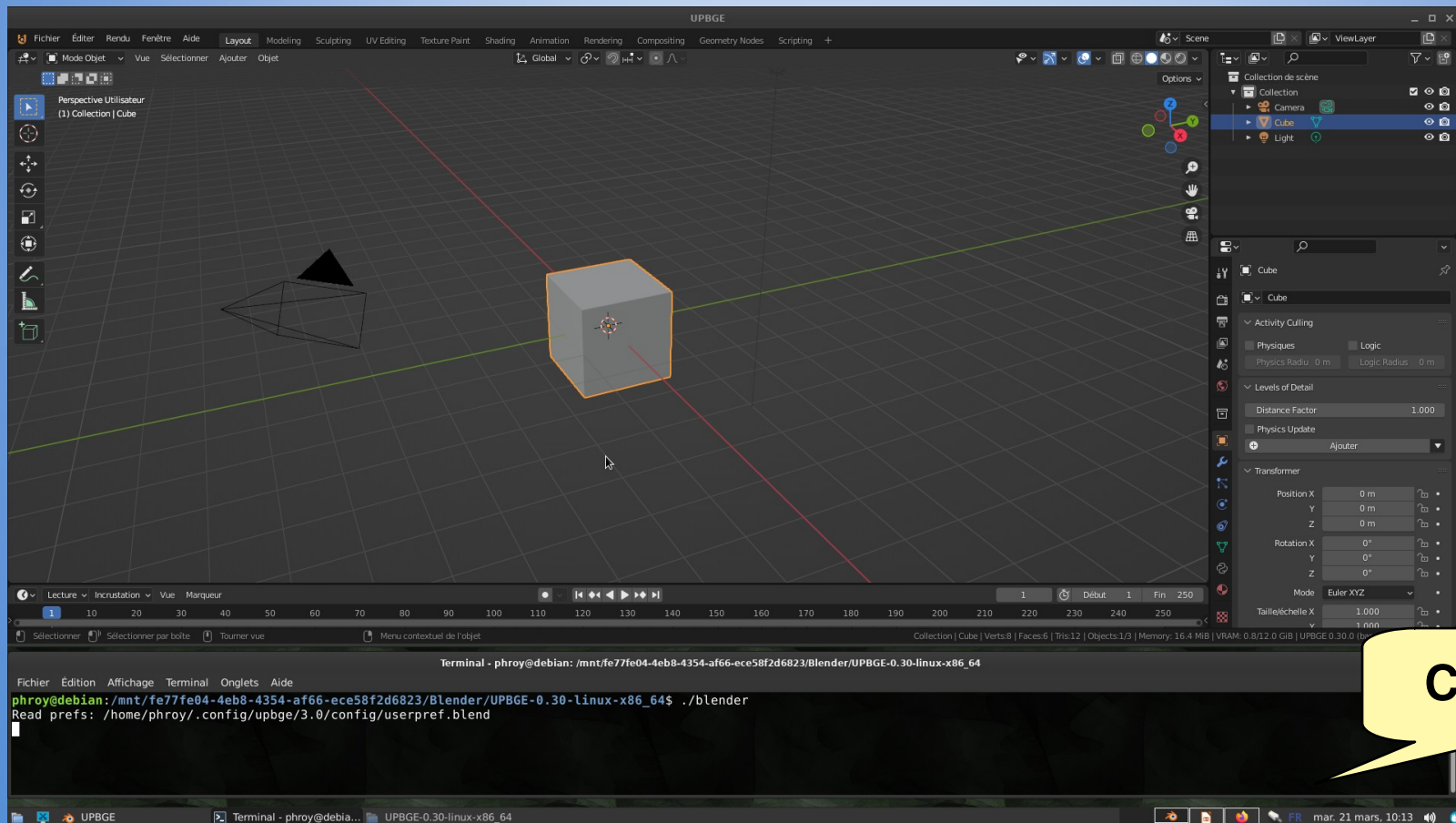
Console

## 2. Créer le plateau



Le moteur de jeu **UPBGE** (**UP** Blender **G**ame **E**ngine) intègre déjà Blender, donc seule son installation suffit. Lancer UPBGE dans une **console** permet de visualiser les messages (erreurs, sortie standard, ...).

- **UPBGE** (version 0.3+) se trouve à cette adresse : <https://upbge.org>

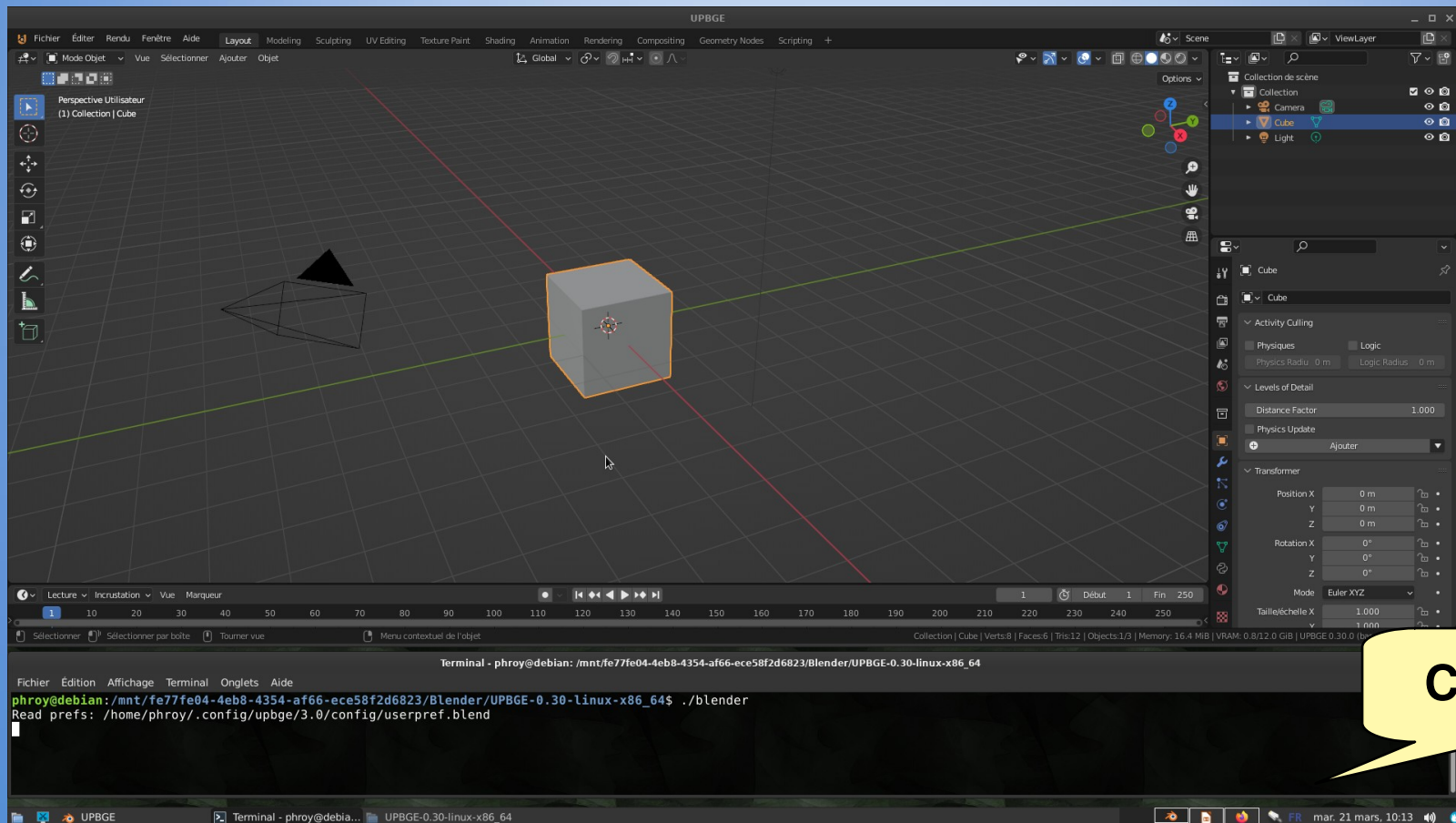


## 2. Déplacer le plateau avec les briques logiques

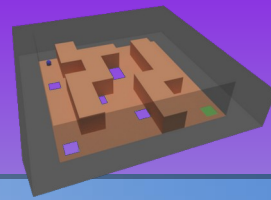


Le moteur de jeu **UPBGE** (**UP** Blender **G**ame **E**ngine) intègre déjà Blender, donc seule son installation suffit. Lancer UPBGE dans une **console** permet de visualiser les messages (erreurs, sortie standard, ...).

- **UPBGE** (version 0.3+) se trouve à cette adresse : <https://upbge.org>

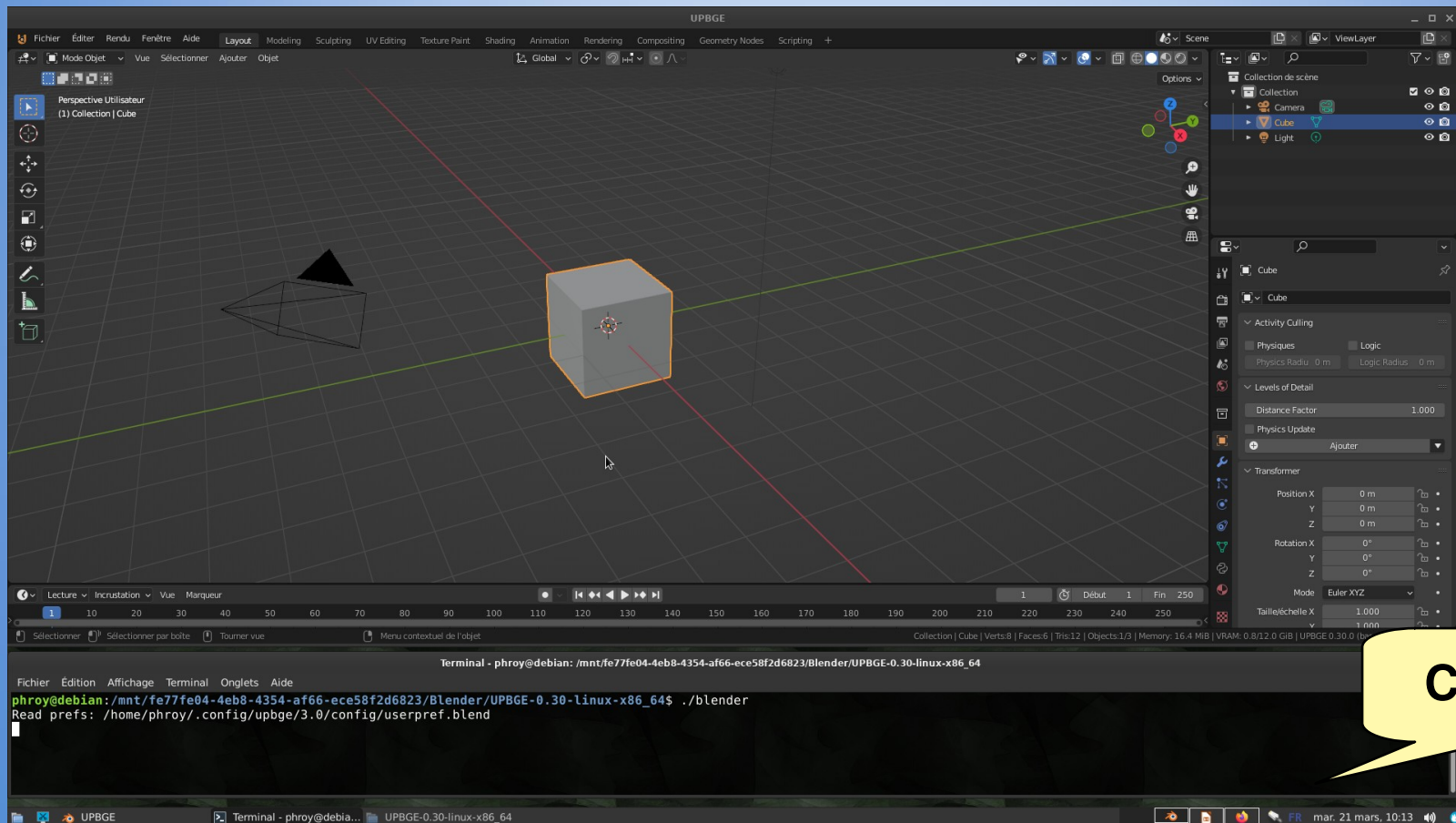


# 3. Créer la bille et définir sa physique



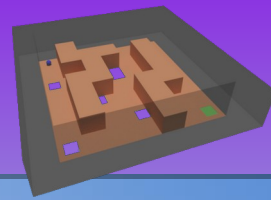
Le moteur de jeu **UPBGE** (**UP** Blender **G**ame **E**ngine) intègre déjà Blender, donc seule son installation suffit. Lancer UPBGE dans une **console** permet de visualiser les messages (erreurs, sortie standard, ...).

- **UPBGE** (version 0.3+) se trouve à cette adresse : <https://upbge.org>



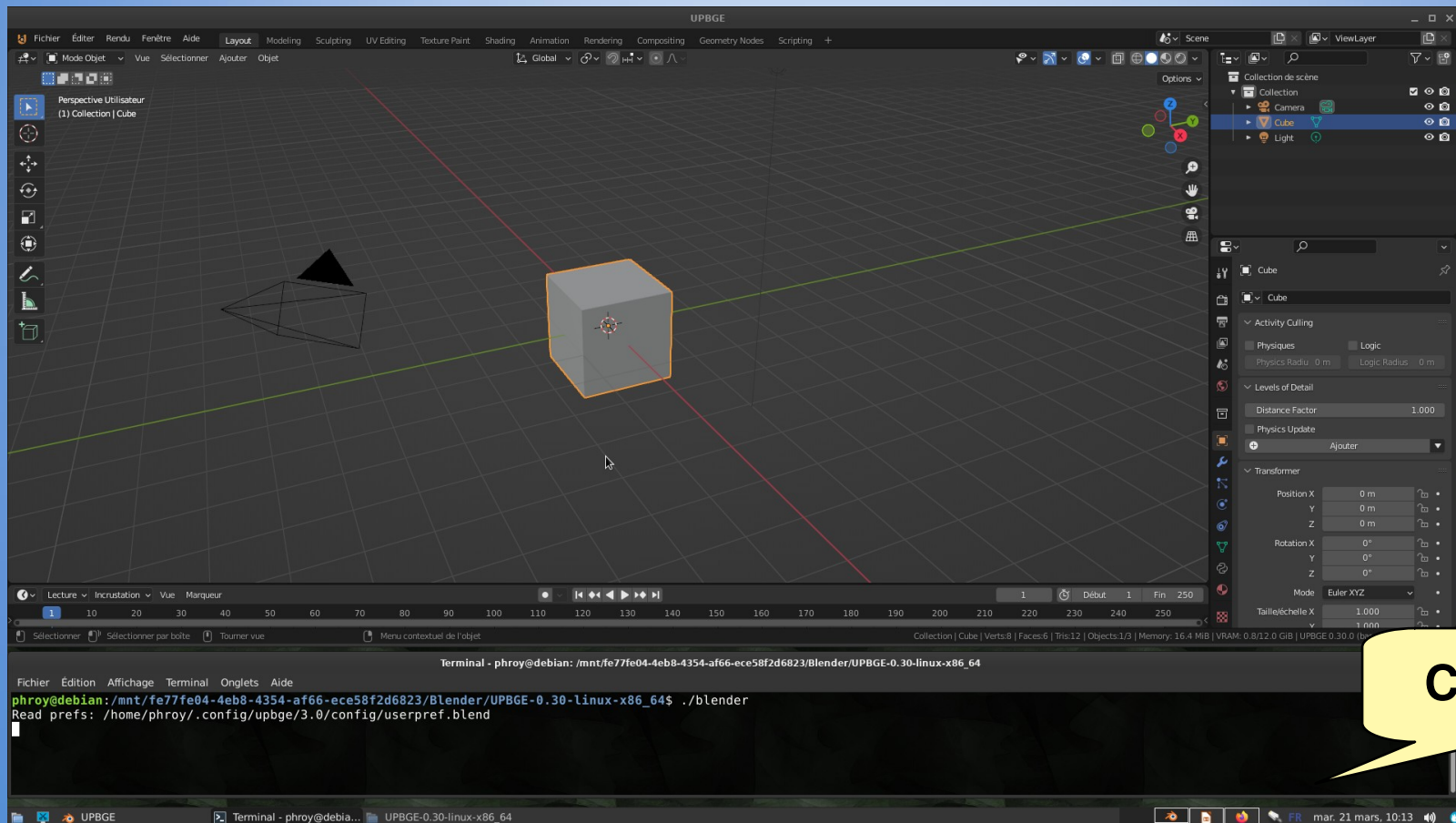
Console

# 4. Définir le game play (règles d'échec et de réussite)



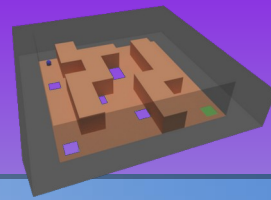
Le moteur de jeu **UPBGE** (**UP** Blender **G**ame **E**ngine) intègre déjà Blender, donc seule son installation suffit. Lancer UPBGE dans une **console** permet de visualiser les messages (erreurs, sortie standard, ...).

- **UPBGE** (version 0.3+) se trouve à cette adresse : <https://upbge.org>



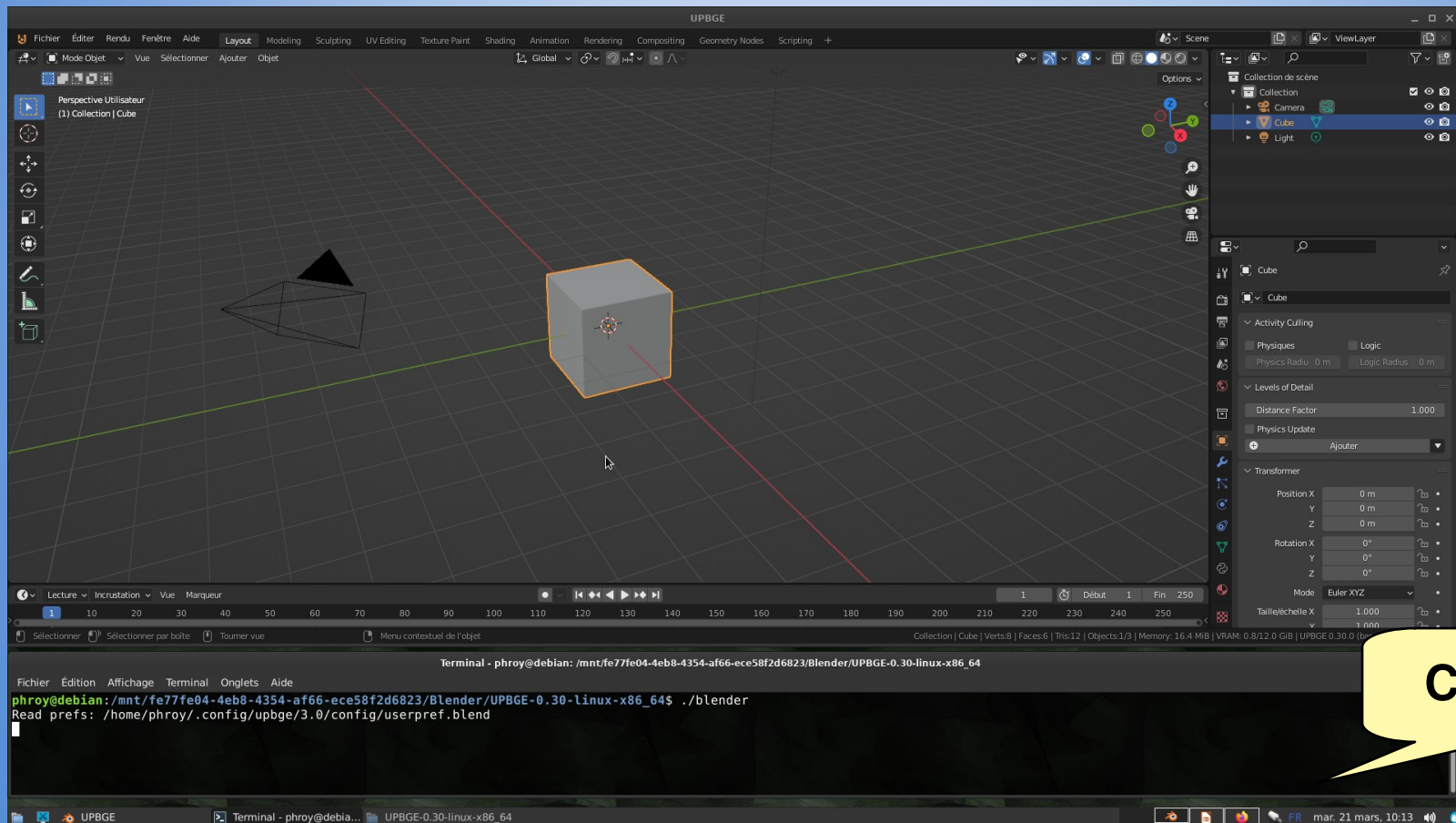
Console

# 5. Créer la fenêtre de fin



Le moteur de jeu **UPBGE** (**UP** Blender **G**ame **E**ngine) intègre déjà Blender, donc seule son installation suffit. Lancer UPBGE dans une **console** permet de visualiser les messages (erreurs, sortie standard, ...).

- **UPBGE** (version 0.3+) se trouve à cette adresse : <https://upbge.org>



Console

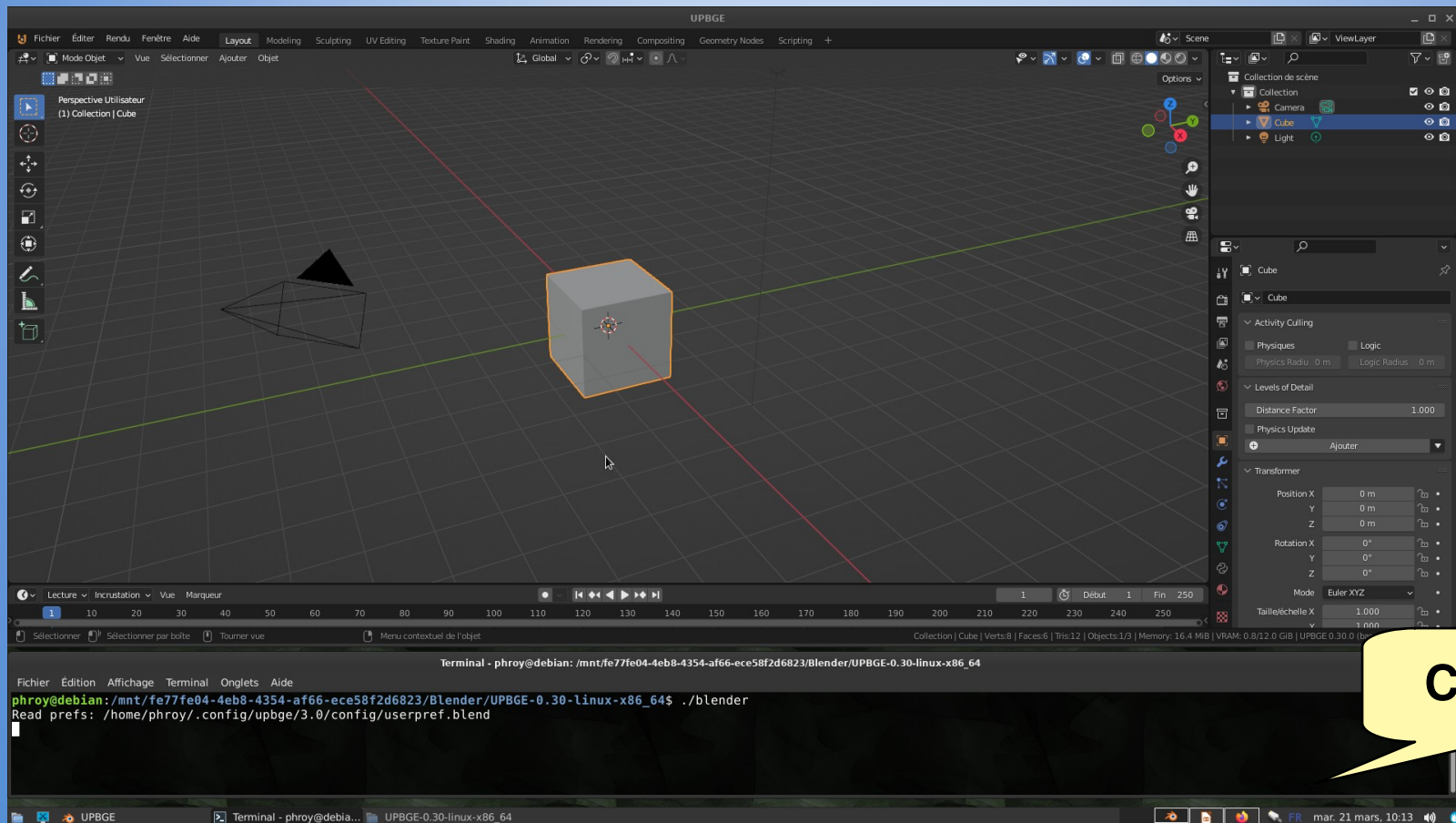


# 6. Animer la fenêtre de fin par des images-clés



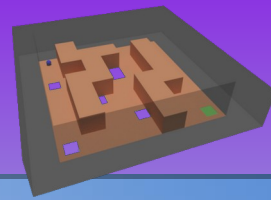
Le moteur de jeu **UPBGE** (**UP** Blender **G**ame **E**ngine) intègre déjà Blender, donc seule son installation suffit. Lancer UPBGE dans une **console** permet de visualiser les messages (erreurs, sortie standard, ...).

- **UPBGE** (version 0.3+) se trouve à cette adresse : <https://upbge.org>



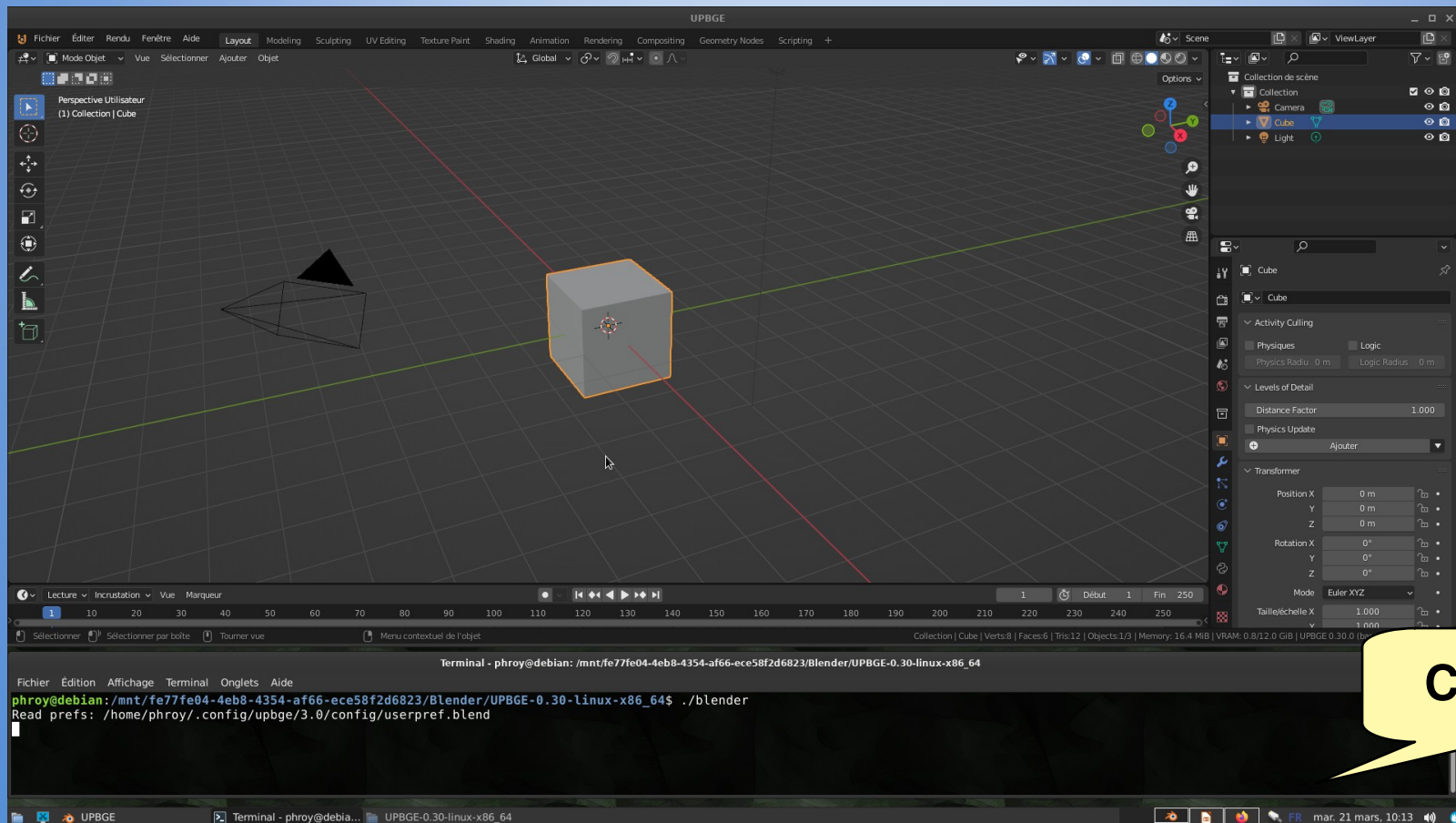
Console

# 7. Fermer la fenêtre de fin par un bouton cliquable



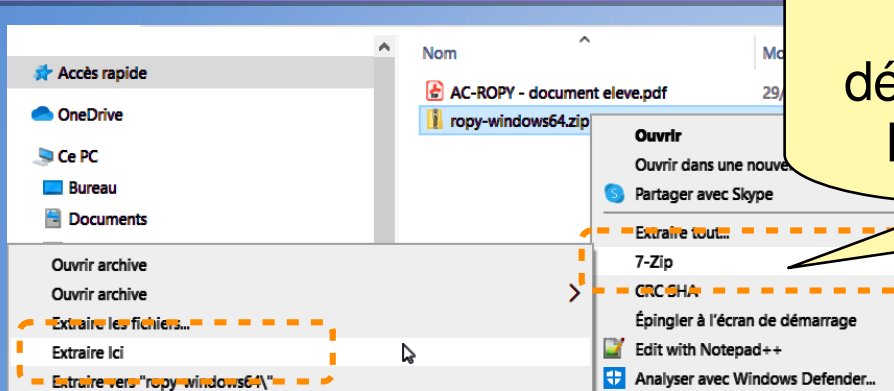
Le moteur de jeu **UPBGE** (**UP** Blender **G**ame **E**ngine) intègre déjà Blender, donc seule son installation suffit. Lancer UPBGE dans une **console** permet de visualiser les messages (erreurs, sortie standard, ...).

- **UPBGE** (version 0.3+) se trouve à cette adresse : <https://upbge.org>



Console

# Mettre en place l'environnement de développement



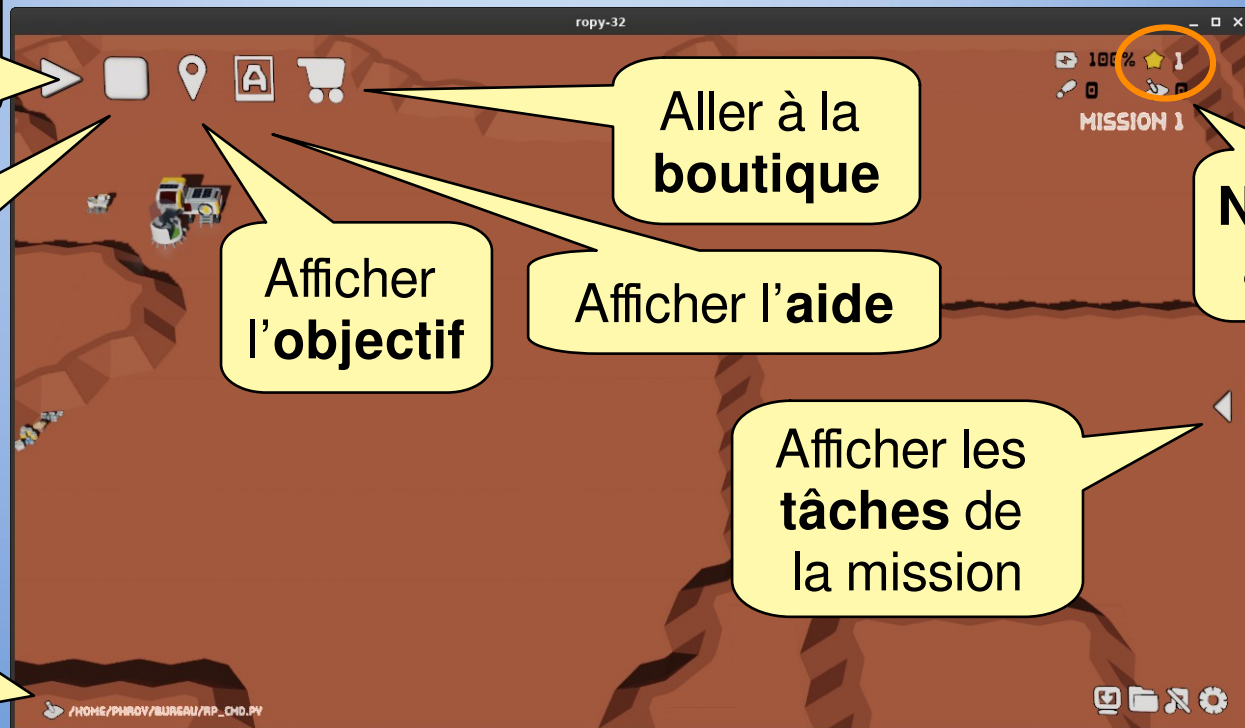
1 : Récupérer l'archive **ropy-windows64.zip** et la décompresser avec **7-Zip** sur le **bureau**. L'extraction va créer le répertoire ropy

Le **simulateur** et la **console** se lancent en même temps avec **ropy.bat** (situé dans le répertoire créé).

Exécuter le programme

Arrêter et réinitialiser

Fichier de commandes



Aller à la boutique

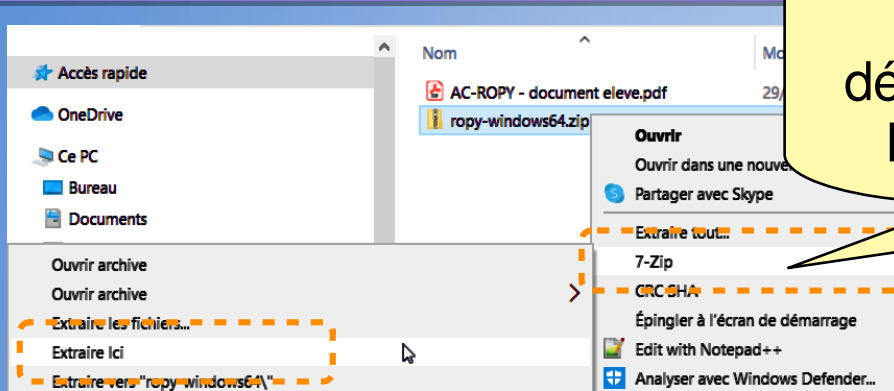
Afficher l'objectif

Afficher l'aide

Niveau actuel

Afficher les tâches de la mission

# Mettre en place l'environnement de développement



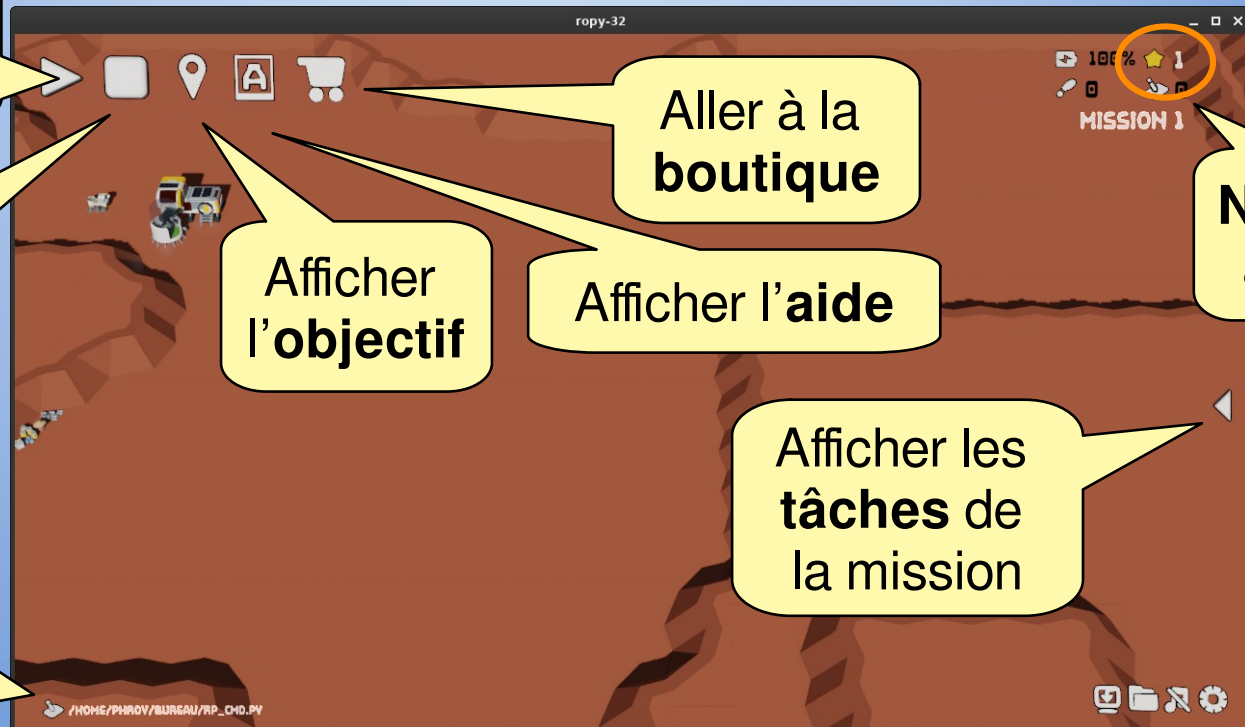
1 : Récupérer l'archive **ropy-windows64.zip** et la décompresser avec **7-Zip** sur le **bureau**. L'extraction va créer le répertoire ropy

Le **simulateur** et la **console** se lancent en même temps avec **ropy.bat** (situé dans le répertoire créé).

Exécuter le programme

Arrêter et réinitialiser

Fichier de commandes



Aller à la boutique

Afficher l'objectif

Afficher l'aide

Niveau actuel

Afficher les tâches de la mission

# Mettre en place l'environnement de développement

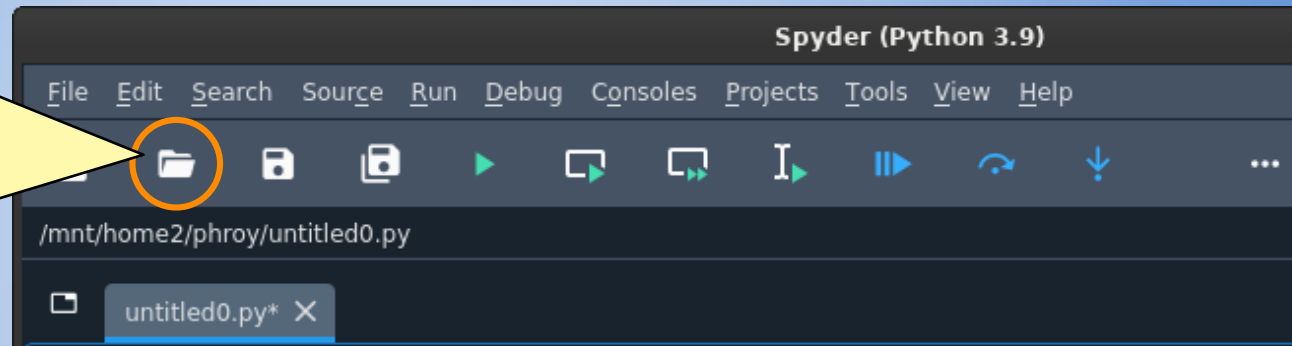


2 : Copier **dans votre répertoire** le fichier de commandes : **ropy\_cmd.py** (ropy commandes).

3 : Lancer **Spyder**.



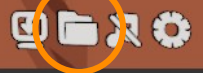
4 : Dans **Spyder** ouvrir le fichier de commandes qui a été précédemment copié dans votre répertoire.



6 : Le nom de votre fichier doit apparaître ici.

/HOME/PHROY/BUREAU/RP\_CMD.PY

5 : Dans le **simulateur**, définir votre fichier comme fichier de commandes.



# Mettre en place l'environnement de développement



```
File Edit Search Source Run Del
/home/phroy/Bureau/rp_cmd.py
rp_cmd.py X
7  # @project: Ropy (Blender-EduTech)
8  #####
9
10 #####
11 # Initialisation du niveau :
12 # Niveau 1 : Les premiers pas de Ropy
13 # Niveau 2 : Ma première fonction
14 # Niveau 3 : Sécuriser Ropy
15 # Niveau 4 : Partir au bout du monde
16 # Niveau 5 : Faire face à l'inconnu
17 # Niveau 6 : Se
18 #####
19
20 #####
21 # Fonctions
22 #####
23
24 #####
25 # Commandes
26 #####
27
28 def commandes():
29
30     rp_gauche()
31     rp_avancerr()
32     rp_avancer()
33     rp_avancer()
34     rp_avancer()
35
36     rp_fin() # A garder
37
38     #####
conda: base (Python 3.9.13)  Comp
```

**8 : Sauvegarder le fichier**  
**Attention !**  
Toujours sauvegarder le fichier avant son exécution.



**9 : Exécuter le programme.**

**7 : Écrire le code Python.**

**10 : Si votre code a une erreur, la console indique où elle se trouve. Ici la ligne 31 contient une commande indéfinie : 'rp\_avancerr'.**

```
Fichier Édition Affichage Te
***** Module rp_cmd
rp_cmd.py:31: error (E0602, undefined-variable, commandes) Undefined variable 'rp_avancerr'

Your code has been rated at 4.44/10 (previous run: 10.00/10, -5.56)
```

# Contenu du fichier rp\_cmd.py



Le fichier `rp_cmd.py` comporte 4 sections.

```
import bge # Bibliothèque Blender Game Engine (UPBGE)
import time
from rp_lib import * # Bibliothèque Ropy

#####
# rp_cmd.py
# @title: Commandes pour le Rover Ropy
# @project: Ropy (Blender-EduTech)
#####

#####
# Initialisation du niveau :
# Niveau 1 : Les premiers pas de Ropy
# Niveau 2 : Ma première fonction
# Niveau 3 : Sécuriser Ropy
# Niveau 4 : Partir au bout du monde
# Niveau 5 : Faire face à l'inconnu
# Niveau 6 : Se rendre utile
#####

#####
# Fonctions
#####

#####
# Commandes
#####

def commandes():
    ➔ rp_gauche()
    rp_avancer()
    rp_avancer()
    rp_avancer()
    rp_avancer()

    rp_fin() # A garder

#####
# En: Externals calls << DONT CHANGE THIS SECTION >>
# Fr: Appels externes << NE PAS MODIFIER CETTE SECTION >>
#####

if __name__=='start':
    thread_cmd_start(commandes)
if __name__=='stop':
    thread_cmd_stop()
```

Le code doit être indenté  
(décalé sur la droite) avec  
la touche Tab

**Import des bibliothèques**  
**Ne pas modifier cette section**

**Fonctions** : section pour le  
codage de **vos fonctions**

**Commandes** : section pour le  
codage des commandes du robot

**La commande `rp_fin()`  
est à conserver.**

**Appels du simulateur**  
(Blender Game Engine)  
**Ne pas modifier cette section**