NOM1 Prénom1

NOM2 Prénom2

Date du TP (jj/mm/aaa)

# Rapport de TPX – Titre du TP

## Introduction

### Question (1).

import pygame

pygame.init()

ecran = pygame.display.set\_mode((300, 200))

pygame.quit()

Ces quatre lignes de programme permettent d’importer le module pygame et de modéliser une fenêtre.

### Question (2).

import pygame

pygame.init()

ecran = pygame.display.set\_mode((300, 200))

continuer = True

while continuer:

for event in pygame.event.get():

if event.type == pygame.KEYDOWN:

continuer = False

pygame.quit()

Ce programme permet d’ouvrir une fenêtre. Tant que l’on appuie sur aucune touche du clavier, la fenêtre « écran » reste ouvre. Lorsqu’on appie sur une touche, la fenêtre se ferme.

## Préparation

### Question (1).

glu.gluPerspective(45, (display[0] / display[1]), 0.1, 50.0)

### Question (2).

gl.glBegin(gl.GL\_LINES) # Indique que l'on va commencer un trace en mode lignes (segments)

gl.glColor3fv([255, 0, 0]) # Indique la couleur du prochian segment en RGB

gl.glVertex3fv((0,0, -2)) # Premier vertice : départ de la ligne

gl.glVertex3fv((1, 0, -2)) # Deuxième vertice : fin de la ligne

gl.glEnd() # Find du tracé

gl.glBegin(gl.GL\_LINES)

gl.glColor3fv([0, 255, 0])

gl.glVertex3fv((0,0, -2))

gl.glVertex3fv((0, 1, -2))

gl.glEnd()

gl.glBegin(gl.GL\_LINES)

gl.glColor3fv([0, 0, 255])

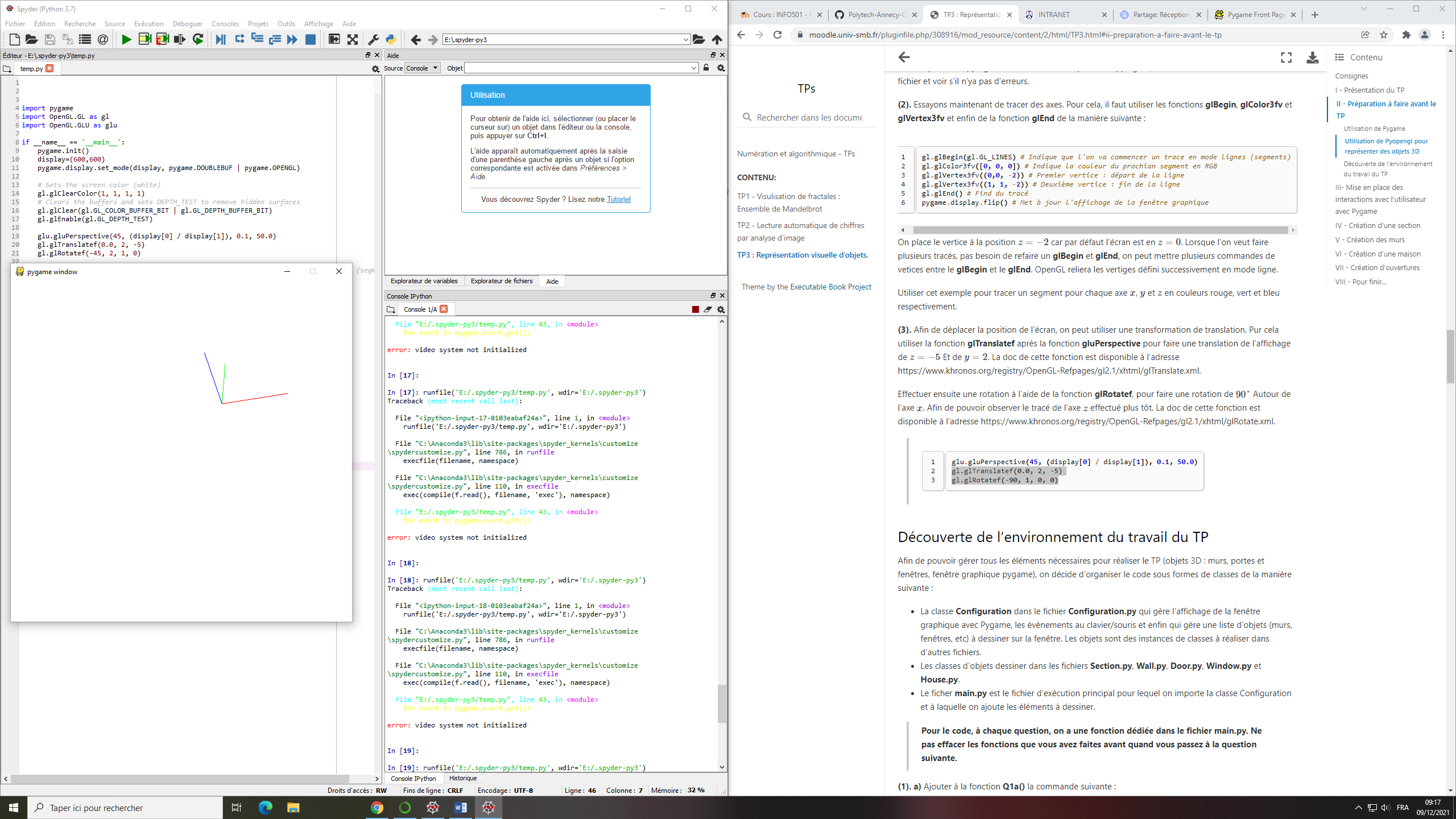
gl.glVertex3fv((0,0, -2))

gl.glVertex3fv((0, 0, -1))

gl.glEnd()

pygame.display.flip()

### Question (3).

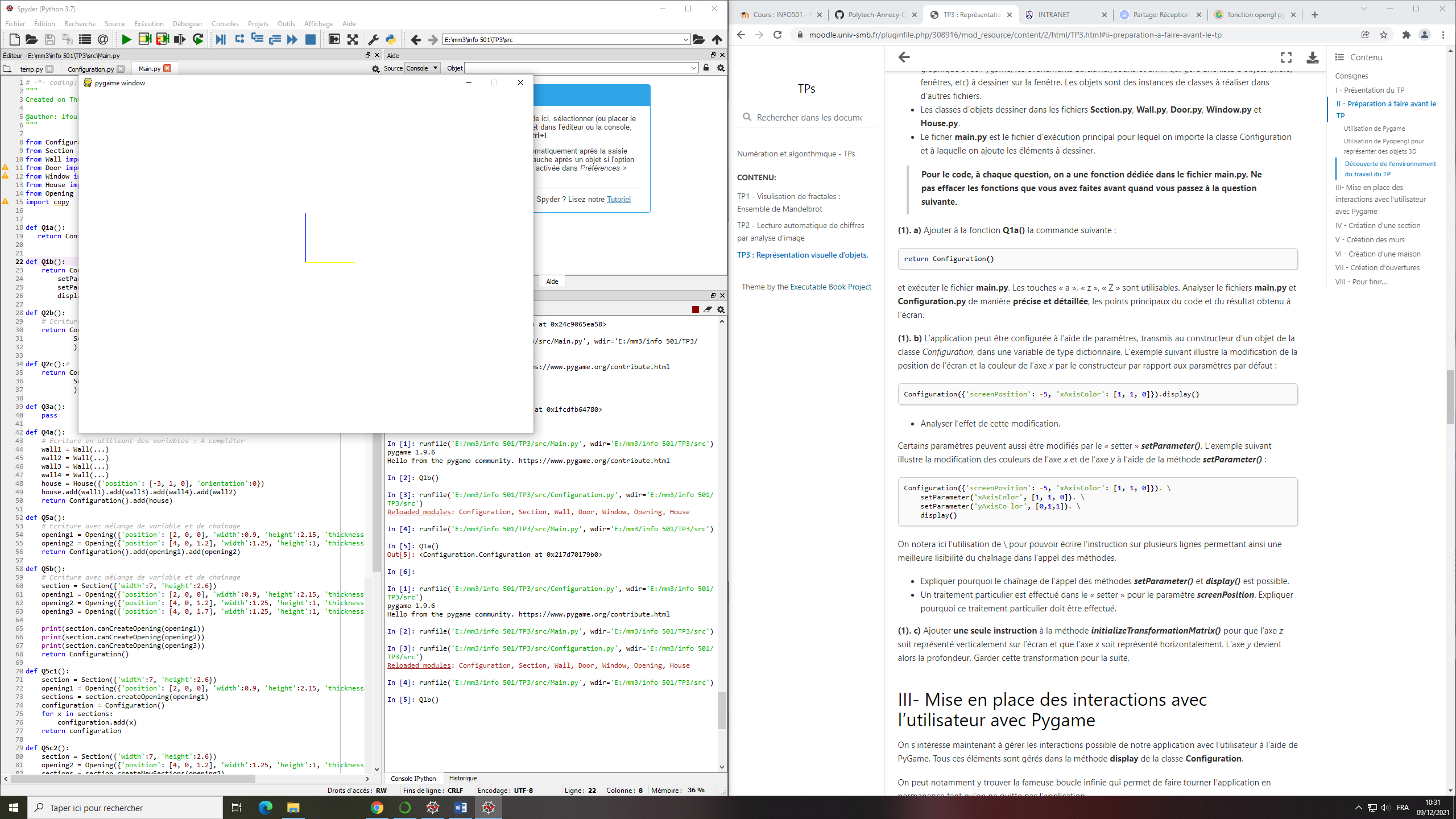
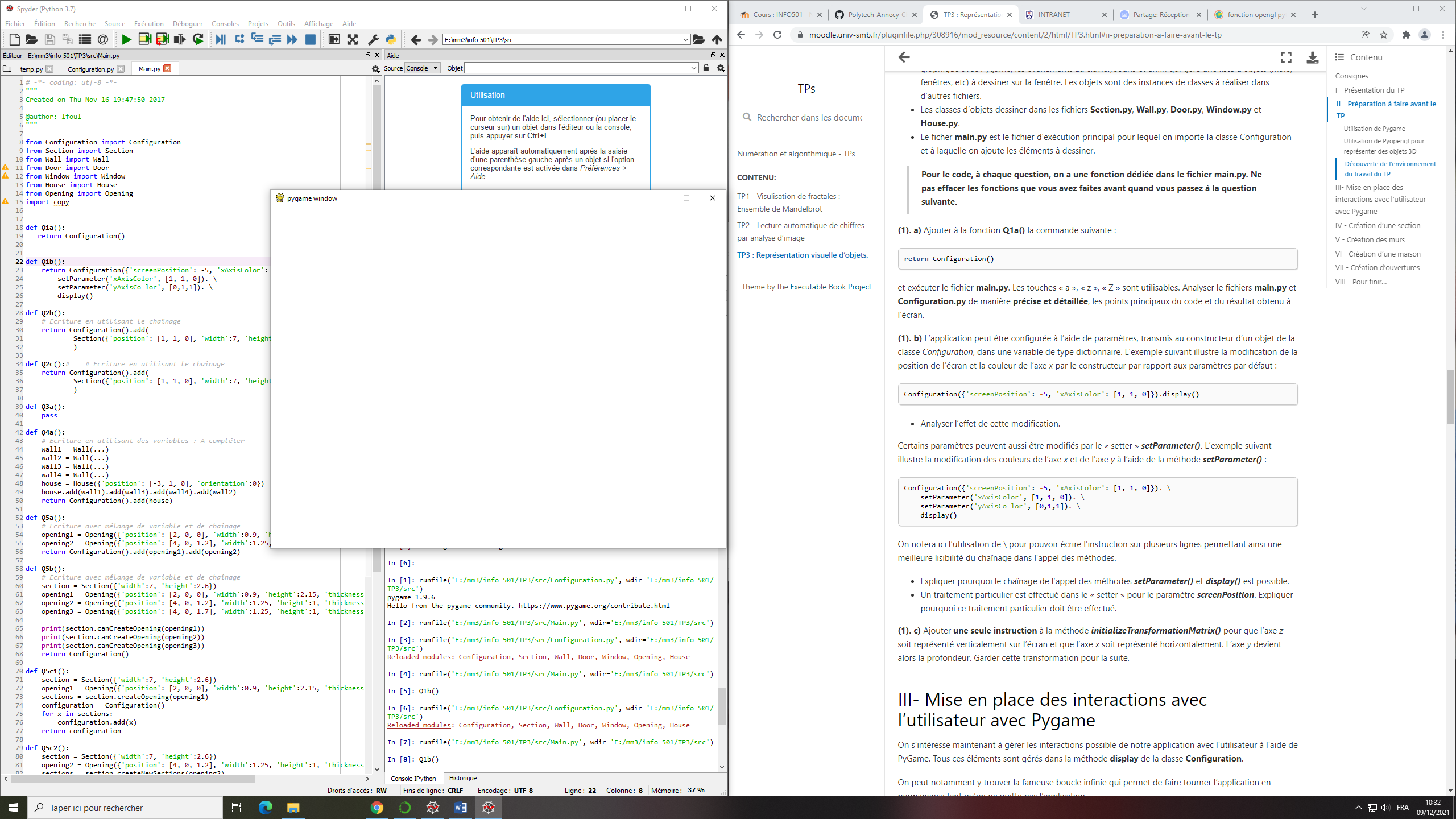


gl.glTranslatef(0.0, 2, -5)

gl.glRotatef(-45, 2, 1, 0)

### Question (1).

1. Dans Configuration.py, on retrouve le tracé des différents axes. On retrouve aussi les fonctions qui sont attribuées au clavier. Le fichier Main.py …
2. On utilise la chaîné de setter pour changer les paramètres du trait. On change ainsi sa couleur.
3. gl.glRotatef(-90, 1, 0, 0)



Après la rotation

Avant la rotation

## Mise en place des interactions avec l’utilisateur

### Question (1).



elif self.event.dict['unicode'] == 'p' or (self.event.mod & pygame.KMOD\_SHIFT and self.event.key == pygame.K\_p):

gl.glScalef(1.1, 1.1, 1.1)

elif self.event.dict['unicode'] == 'm' or (self.event.mod & pygame.KMOD\_SHIFT and self.event.key == pygame.K\_m):

gl.glScalef(0.9,0.9,0.9)



if self.event.type == pygame.MOUSEBUTTONDOWN and self.event.button == 4:

gl.glScalef(1.1, 1.1, 1.1)

elif self.event.type == pygame.MOUSEBUTTONDOWN and self.event.button == 5:

gl.glScalef(0.9, 0.9, 0.9)



if pygame.mouse.get\_pressed()[0] == 1:

x= self.event.rel[0]

y= self.event.rel[1]

gl.glRotate(x+y, 1, 0, 1)

elif pygame.mouse.get\_pressed()[2] == 1:

x= self.event.rel[0]

y= self.event.rel[1]

gl.glTranslate(x\*0.05, 0, y\*0.05)

## Création d’une section

### Question (2)

## Conclusion

Expliquer ici l’état d’avancement du TP actuel, les difficultés principales que vous avez rencontrées ainsi que ce que vous avez appris.