Compte Rendus

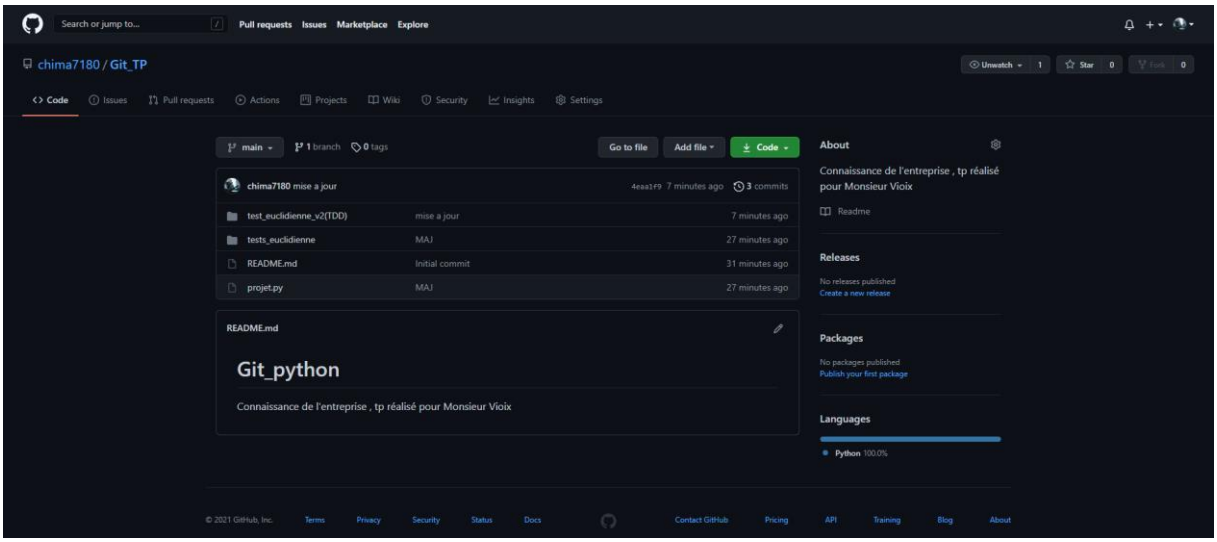
Connaissance de l'entreprise

Le 08/04/2021
Gressette Théo
Million Florian

Table des matières

Partie 1 : Initialisation de GIT

Nous avons commencé par Créer un compte sur github :

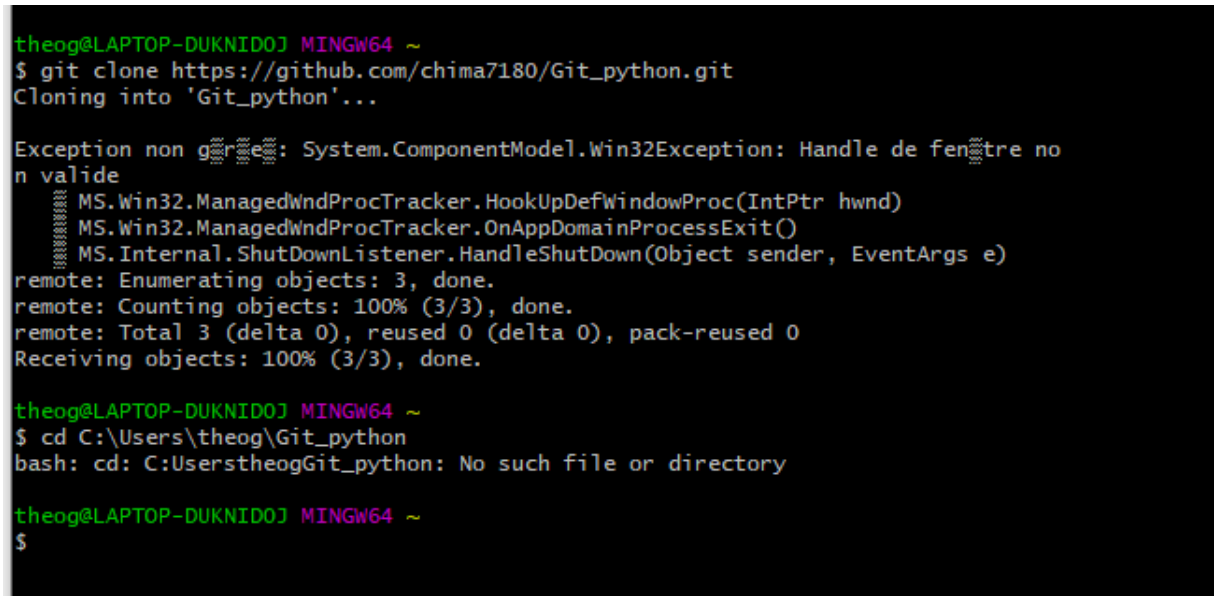


Nous avons ensuite récupéré le lien de notre répertoire créer :
https://github.com/chima7180/Git_TP

Nous avons Ouvert l'interface Git Bash pour Windows

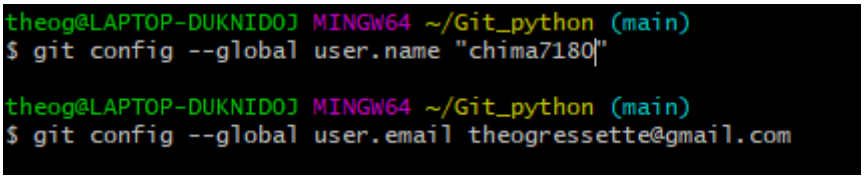
Fait la commande suivante :

Git clone https://github.com/chima7180/Git_TP



Ensuite ouvert git bash dans le dossier créer :

S’identifier sur git bash si nous sommes sur windows :



Pour mettre à jour le dossier on utilise les commandes suivantes :

Git add *

Git commit -a -m « ton message »

Git push -u origin main

Partie 2 : Programmation Python

1.*Erreur de compréhension*

Lors du début de notre Tp nous n'avons pas utiliser la Méthode TDD mais nous avons quand même réalisé la fonction *distance_euclidienne* avec un fichier Projet et un fichier de Test.

projet.py

```
import numpy as np
point_a = np.array((0,0))
point_b = np.array((1,1))

def distance_euclidienne(a,b):
    distance = np.linalg.norm(a-b)
    return distance

print(distance_euclidienne(point_a,point_b))
```

Test_euclidienne

```
import numpy as np
from projet import distance_euclidienne
from random import randint

Bilan = True

point_a =  np.array((randint(0,10000),randint(0,10000)))
point_b =  np.array((randint(0,10000),randint(0,10000)))
point_c =  np.array((randint(0,10000),randint(0,10000)))
if distance_euclidienne(point_a,point_a) == 0:
    Bilan = True
else :
    Bilan = False
if Bilan == False:
    print("echec du test")


if distance_euclidienne(point_b,point_b) == 0:
    Bilan = True
else :
    Bilan = False
if Bilan == False:
    print("echec du test")

if distance_euclidienne(point_a,point_b) == 0 or
distance_euclidienne(point_a,point_b) > 0:
    Bilan = True
else :
    Bilan = False
if Bilan == False:
    print("echec du test")

ab = distance_euclidienne(point_a,point_b)
ac = distance_euclidienne(point_a,point_c)
```

```
bc = distance_euclidienne(point_b,point_c)
somme = ab + bc
if ac < somme or ac == somme :
    Bilan = True
else :
    Bilan = False
if Bilan == False:
    print("echec du test")a,point_b))
```

2. mise en place avec la procédure TDD

2.1 tests pour la fonction distance_euclidienne

```
import numpy as np
import math
from projet import distance_euclidienne
from random import randint
import unittest
class test_distance_euclidienne(unittest.TestCase):
    def setUp(self):
        point_a =  np.array((randint(0,10000),randint(0,10000)))
        point_b =  np.array((randint(0,10000),randint(0,10000)))
        point_c =  np.array((randint(0,10000),randint(0,10000)))
    def PermierTest(self):
        self.assertEqual( 0,distance_euclidienne(point_a,point_a))
    def DeuxièmeTest(self):
        self.assertGreaterEqual(distance_euclidienne(point_a,point_b),0)
    def TroisèmeTest(self):

self.assertEqual(distance_euclidienne(point_a,point_b),distance_euclidienne
(point_b,point_a))
    def QuatrièmeTest(self):
        ab = distance_euclidienne(point_a,point_b)
        ac = distance_euclidienne(point_a,point_c)
        bc = distance_euclidienne(point_b,point_c)
        somme = ab + bc
        self.assertLessEqual(ac,somme)
    def Derniertest(self):
        self.assertAlmostEqual(distance_euclidienne((0,1)(1,0)),sqrt(2))
if __name__ == '__main__':
unittest.main()
```

Conclusion

1. *Problèmes rencontrés*

Lors de ce Tp nous avons rencontré plusieurs difficultés comme le fait que nous avons trouvé le tp assez difficile probablement par manque d'explication ou le fait que nous avions pris beaucoup trop de temps à comprendre le fonctionnement de Git ainsi que la méthode TDD.

2. Rendus

Avec le compte rendu vous allez trouver le dossier de travail avec les programmes mais je vous dépose aussi le lien de mon Dossier GitHub :

https://github.com/chima7180/Git_TP