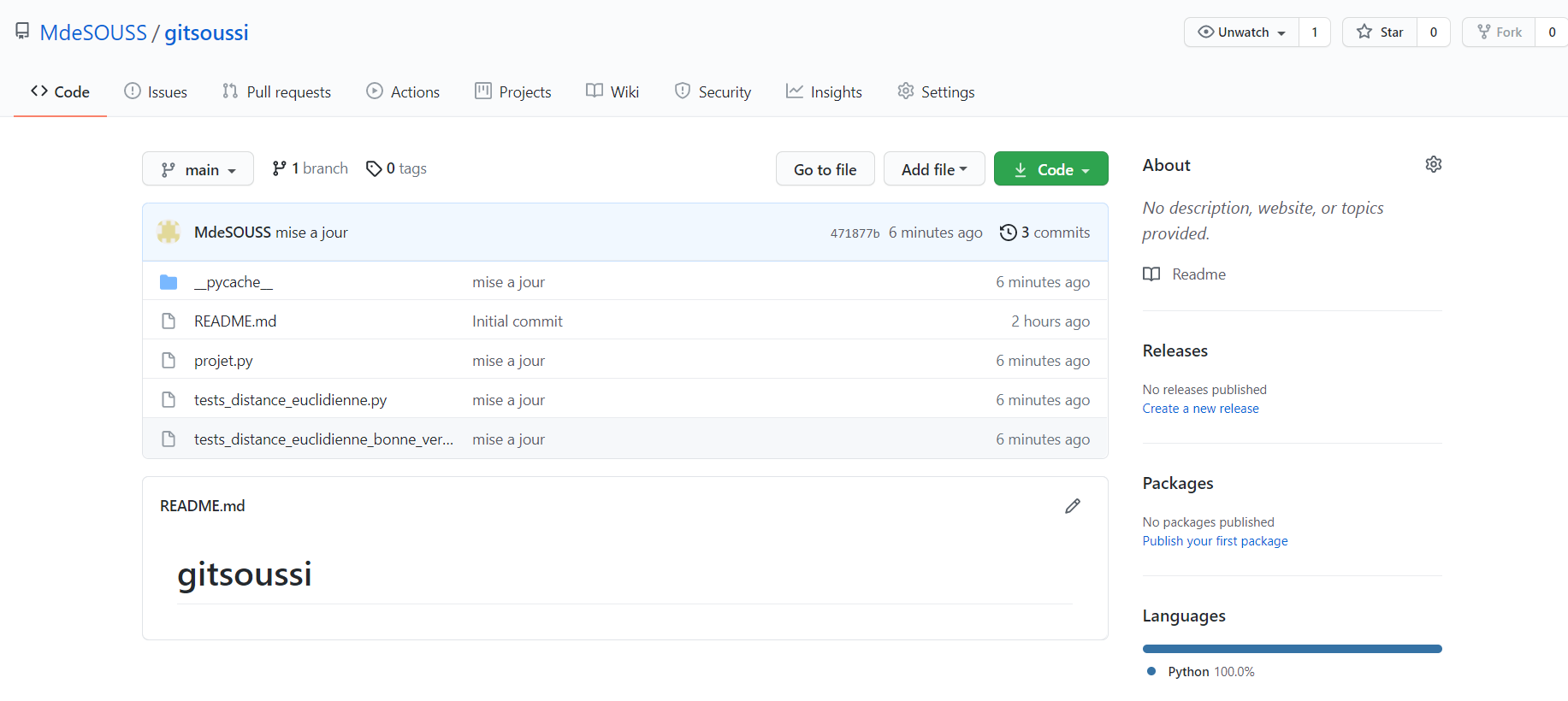
Monde de l’entreprise



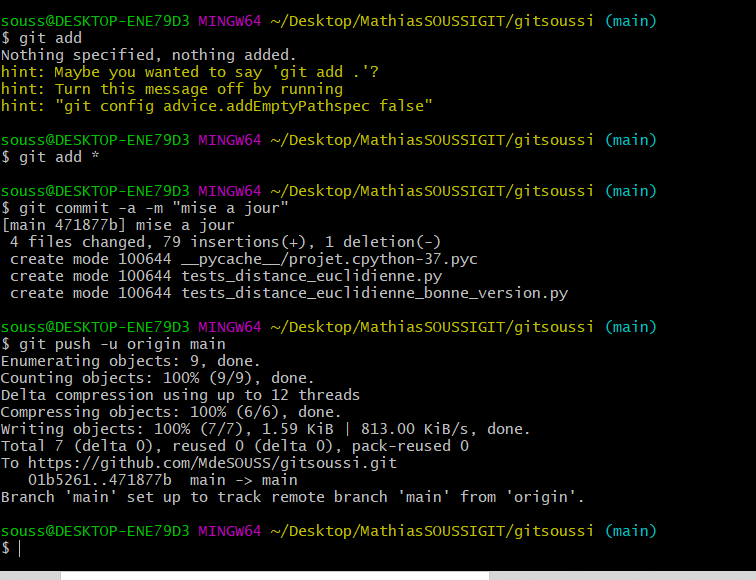
SOUSSI MATHIAS

Sauvegarde projet GIT sur Github



Lien du github : [MdeSOUSS/gitsoussi (github.com)](https://github.com/MdeSOUSS/gitsoussi)

Commandes pour mettre a jour sur le site :



Lors du début de ce Tp je ‘avais pas compris ce qu’il fallait faire et j’ai donc créer un programme a base de if et de else pour tester la validité de la fonction. Environ 1 heure avant la fin du temp règlementaire pour rendre le tp un camarade m’a fait remarquer mon erreur et donc il ya deux fichier test dont l’un avec la bonne méthode, la méthode TDD

Programme Main :

# Créé par souss, le 24/04/2021 en Python 3.7

**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 avec TDD :

# Créé par souss, le 30/04/2021 en Python 3.7

**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()

Test BARBARE :

# Créé par souss, le 24/04/2021 en Python 3.7

**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")