

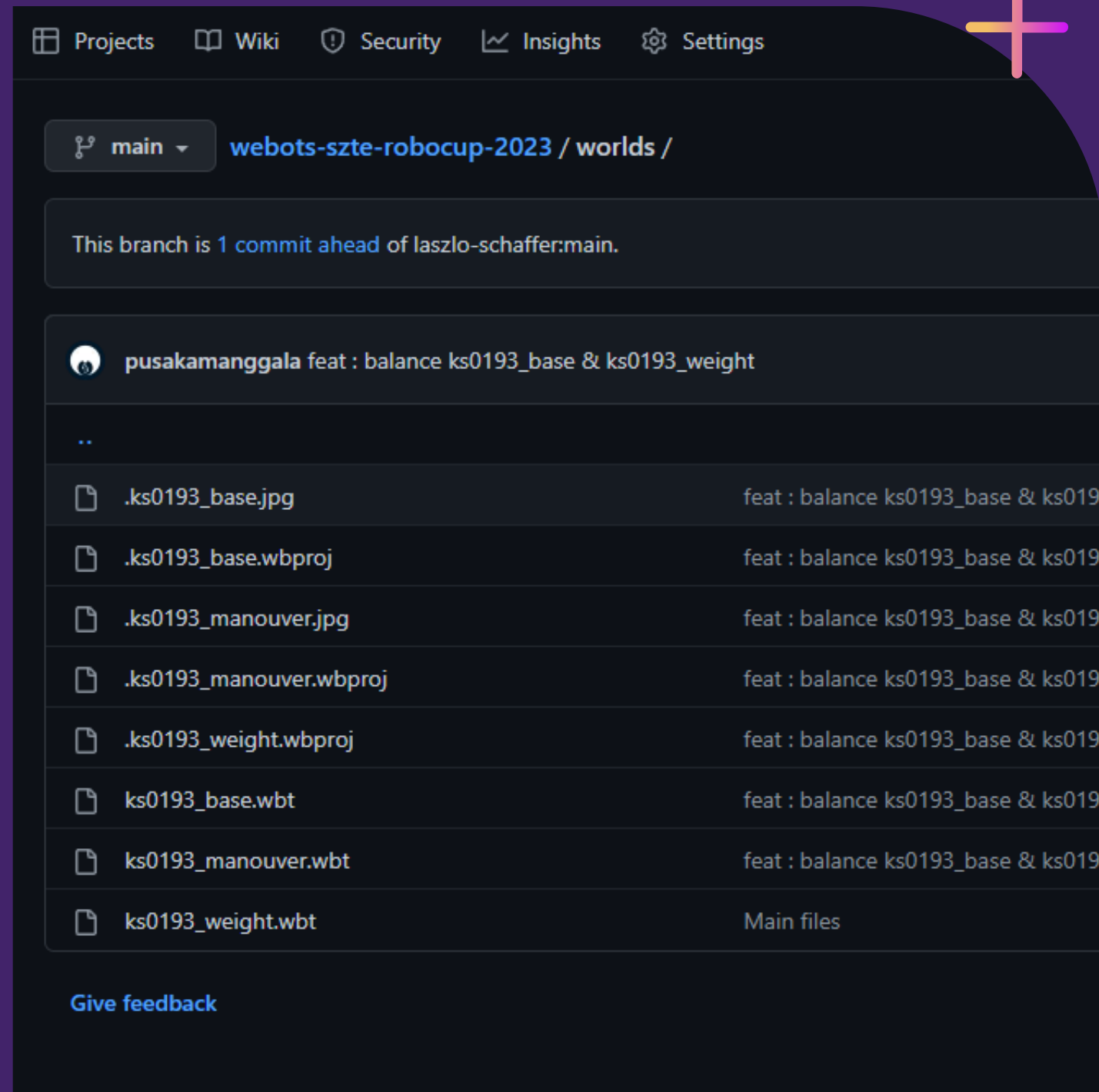


UTS ROBOTIKA

Pusaka Manggala (1103194021)



PUSAKA MANGGALA



Fork Repository

<https://github.com/laszlo-schaffer/webots-szte-robocup-2022>

Fork repository kedalam repository anda sendiri, lalu clone repository and untuk membuka dan menjalankan projek di komputer pribadi.

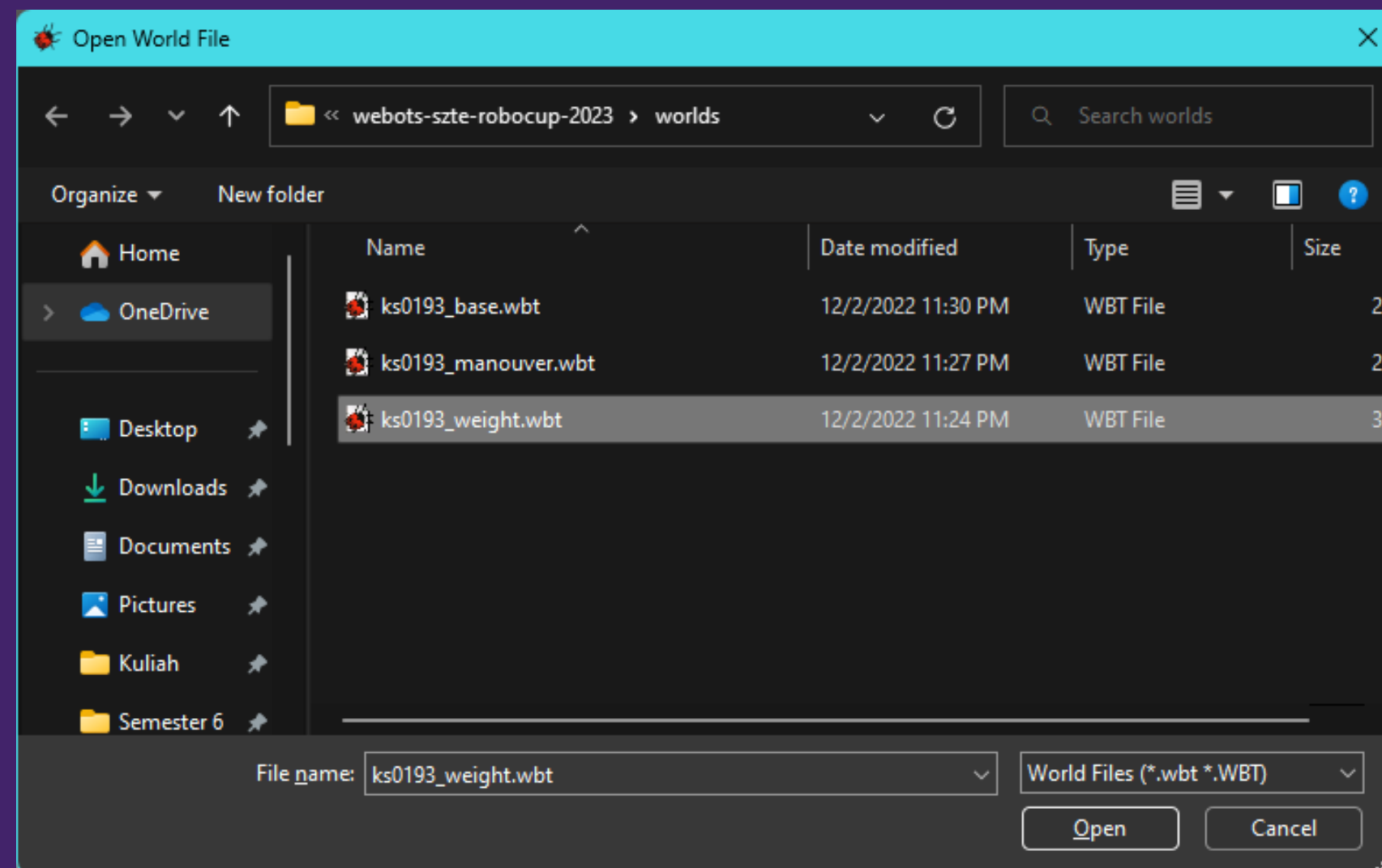


STUDIO SHODWE

Open World di Webots

Buka 'world' yang terletak didalam folder
..\webots-szte-robocup-2023\worlds

File world tersebut berisikan robot, properti, dan lingkungan yang akan dipakai. Disini saya memakai ks0193_weight.wbt

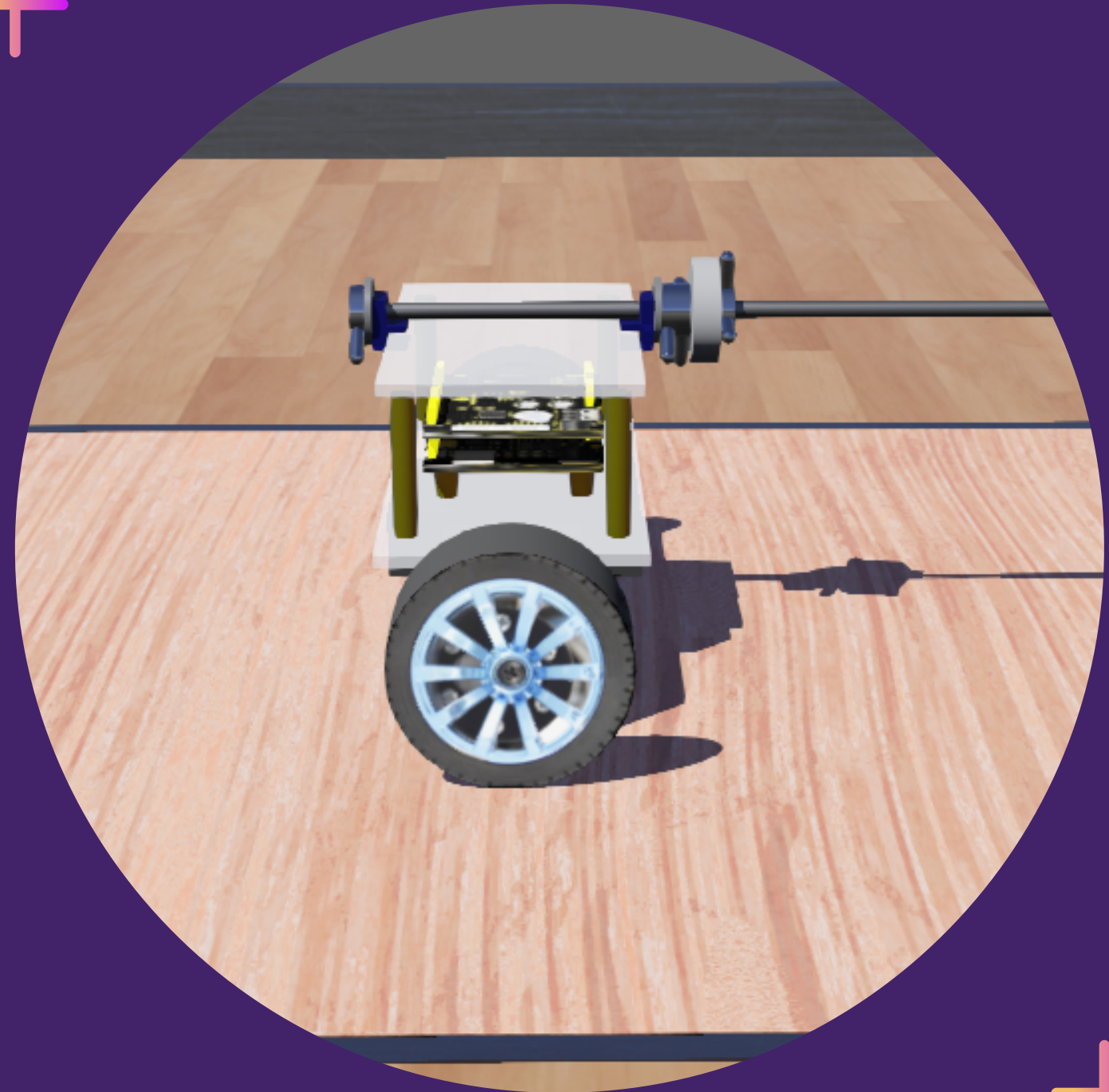




STUDIO SHODWE

ks0193_weight

Robot ini memiliki 2 roda yang bisa berputar. Robot tersebut dipasangkan beban yang tidak seimbang di atasnya, yang bertujuan untuk membuktikan bahwa robot tersebut bisa menyeimbangkan diri.





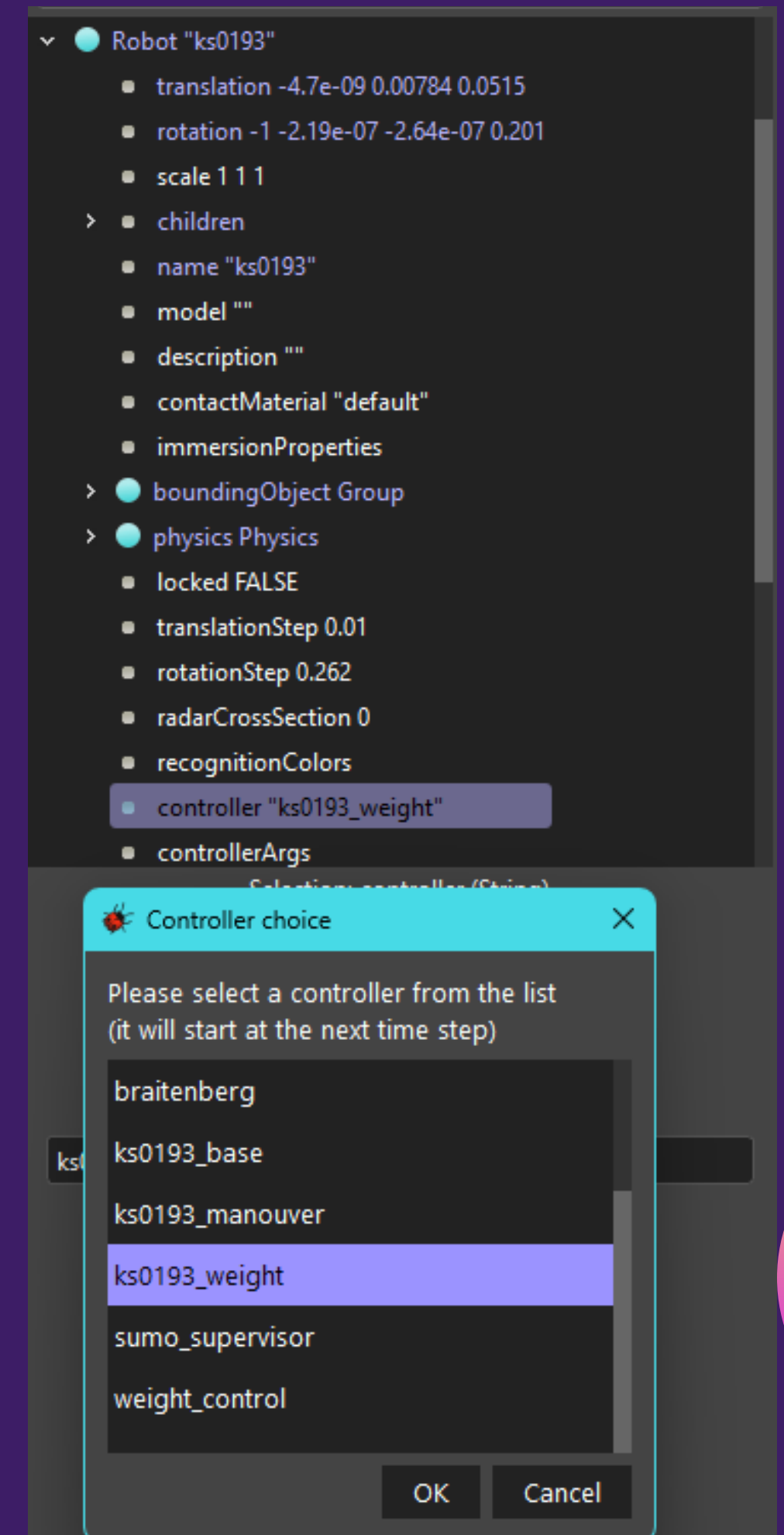
STUDIO SHODWE



Controller choice

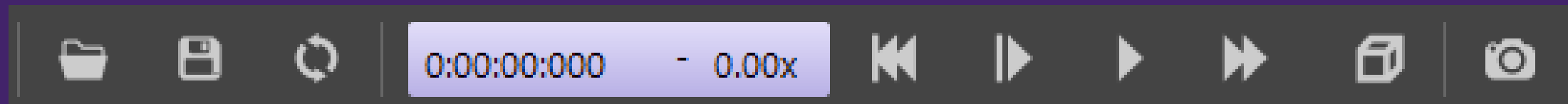
webots-szte-robocup-2023\controllers\ks0193_weight.py

ks0193_weight.py adalah file yang berisi program yang berfungsi untuk mengontrol dan menyeimbangkan robot sehingga robot tersebut bisa tetap berdiri walaupun diberi beban yang tidak seimbang.






STUDIO SHODWE



Run Simulation

klik tombol  untuk menjalankan simulasi robot agar berjalan sesuai program controllanya. Proses sukses jika robot bisa tetap berdiri tegak walaupun sedikit bergoyang ke depan dan kebelakang karena beban yang tidak seimbang.

IMPORTABLE_EXTERNPROTO

PROTO that may be imported during the execution must be declared

Insert new



```

1 """ks0193_base controller"""
2
3 # You may need to import some classes of the controller
4 from controller import Robot, Keyboard
5 import numpy as np
6 from simple_pid import PID
7
8 PLOTTING = False # plotting angle data
9 if PLOTTING:
10     from pandas import Series
11     import matplotlib.pyplot as plt
12
13 # prototype Kalman filter based on
14 # https://wiki.keyestudio.com/Ks0193_keyestudio_Self-b
15 class Kalman:
16     def __init__(self):
17         self.Q_angle = 0.001 #Covariance of gyroscope
18         self.Q_gyro = 0.003 #Covariance of gyroscope
19         self.R_angle = 0.5 #Covariance of accelerometer
20         self.C_0 = 1
21         self.dt = 0.005 #The value of dt is the filter
22         self.K1 = 0.05 # a function containing the Kalman
23         self.q_bias = 0.0
24         self.angle=0.0
25
26         self.Pdot = np.zeros((4,1))
27         self.P = np.array([[1,0], [0,1]])
28
29     def estimate(self, angle_m, gyro_m):
30         self.angle += (gyro_m - self.q_bias) * self.dt
31         self.angle_err = angle_m - self.angle
32
33         self.Pdot[0] = self.Q_angle - self.P[0][1] - s
34         self.Pdot[1] = -self.P[1][1]
35         self.Pdot[2] = -self.P[1][1]
36         self.Pdot[3] = self.Q_gyro
37
38         self.P[0][0] += self.Pdot[0] * self.dt #The
39         self.P[0][1] += self.Pdot[1] * self.dt
40         self.P[1][0] += self.Pdot[2] * self.dt
41         self.P[1][1] += self.Pdot[3] * self.dt
42
43         #Intermediate variable of matrix multiplication
44         self.PCT_0 = self.C_0 * self.P[0][0]
45         self.PCT_1 = self.C_0 * self.P[1][0]

```

Console - All

Pitch: -13.06 pos: 0.04 speed 0.0

-13.048891151553013

Pitch: -13.05 pos: 0.04 speed -0.0

43 05300000740005