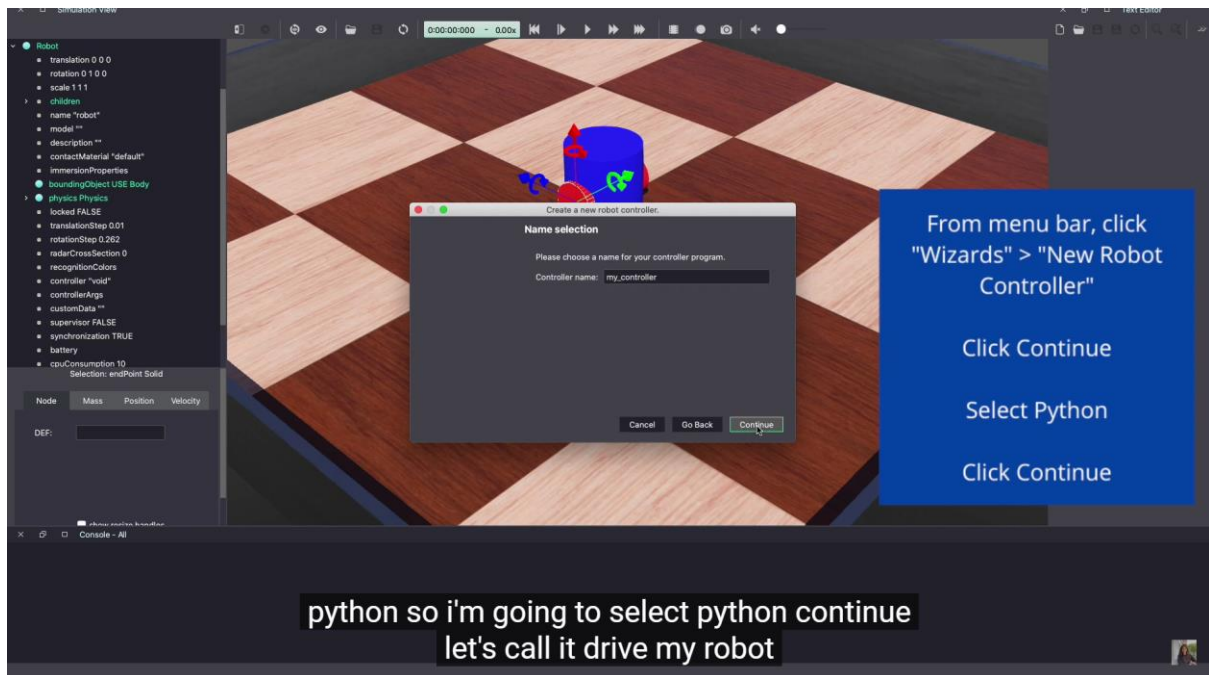


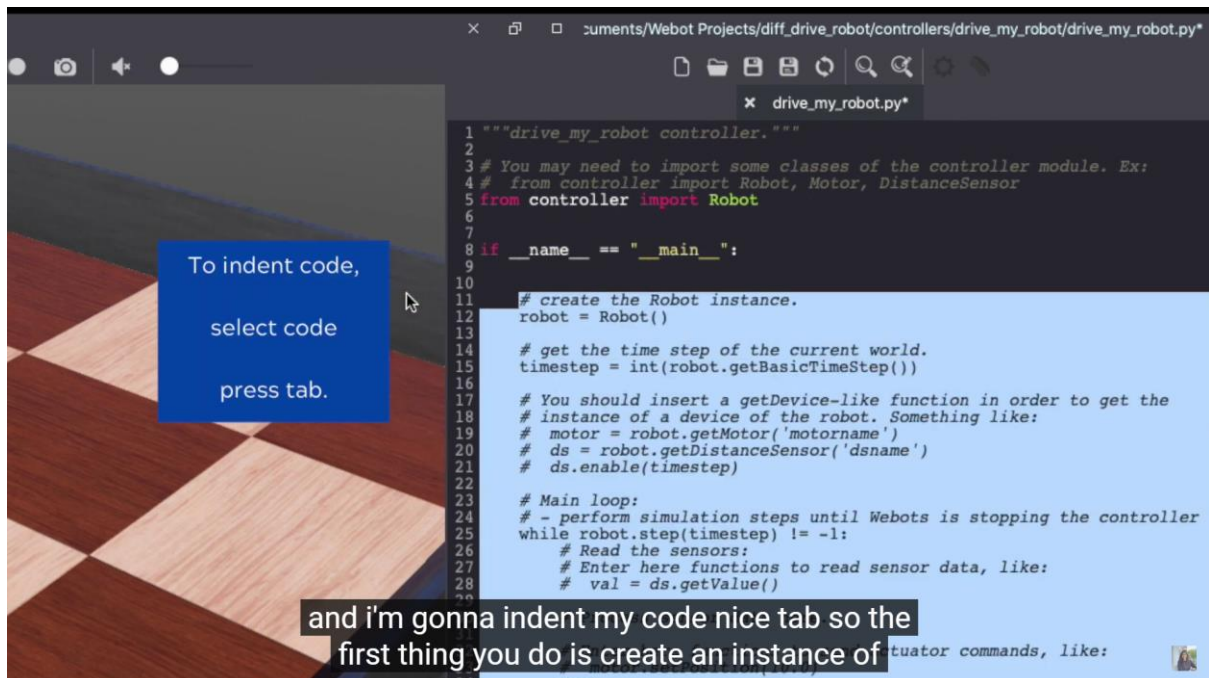
Nama : Ichlasul Al Gifari

NIM : 1103204236

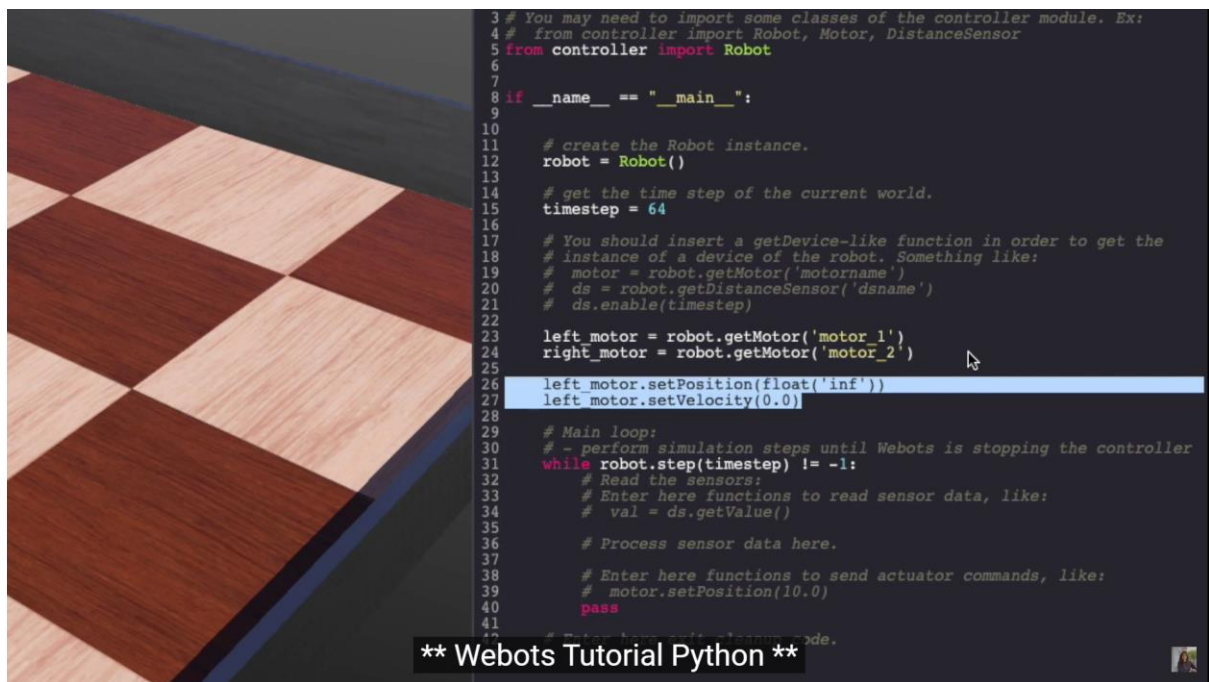
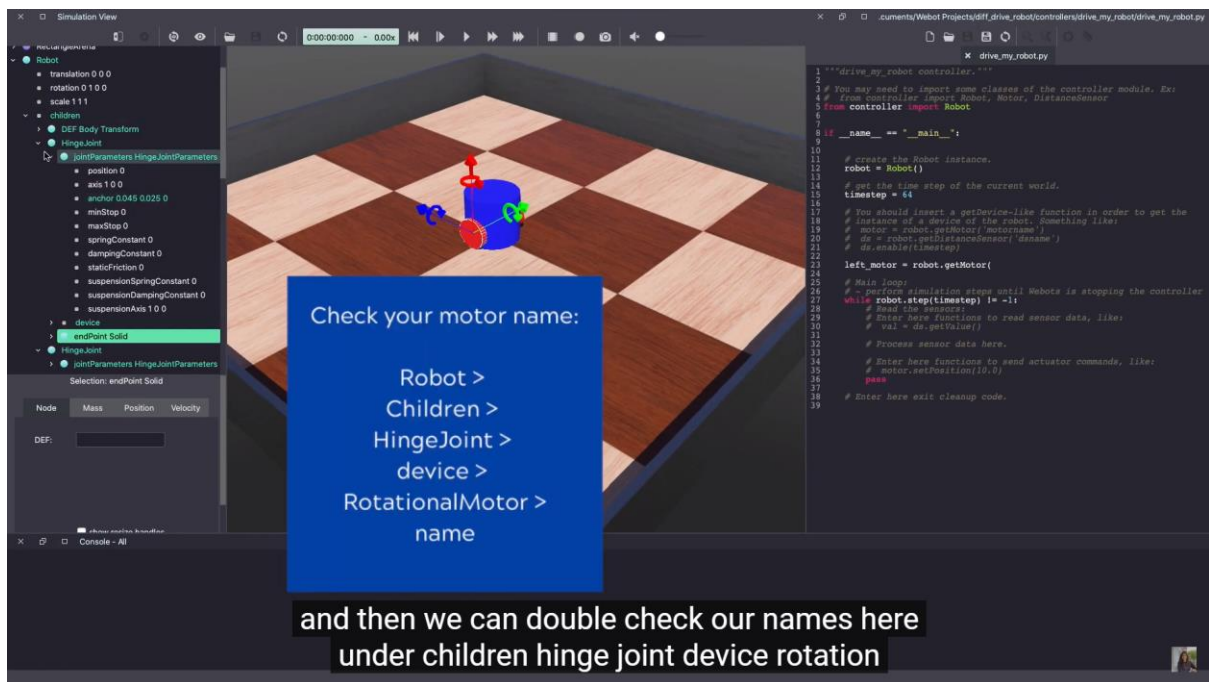
## Tugas Lecture 10 Robotik



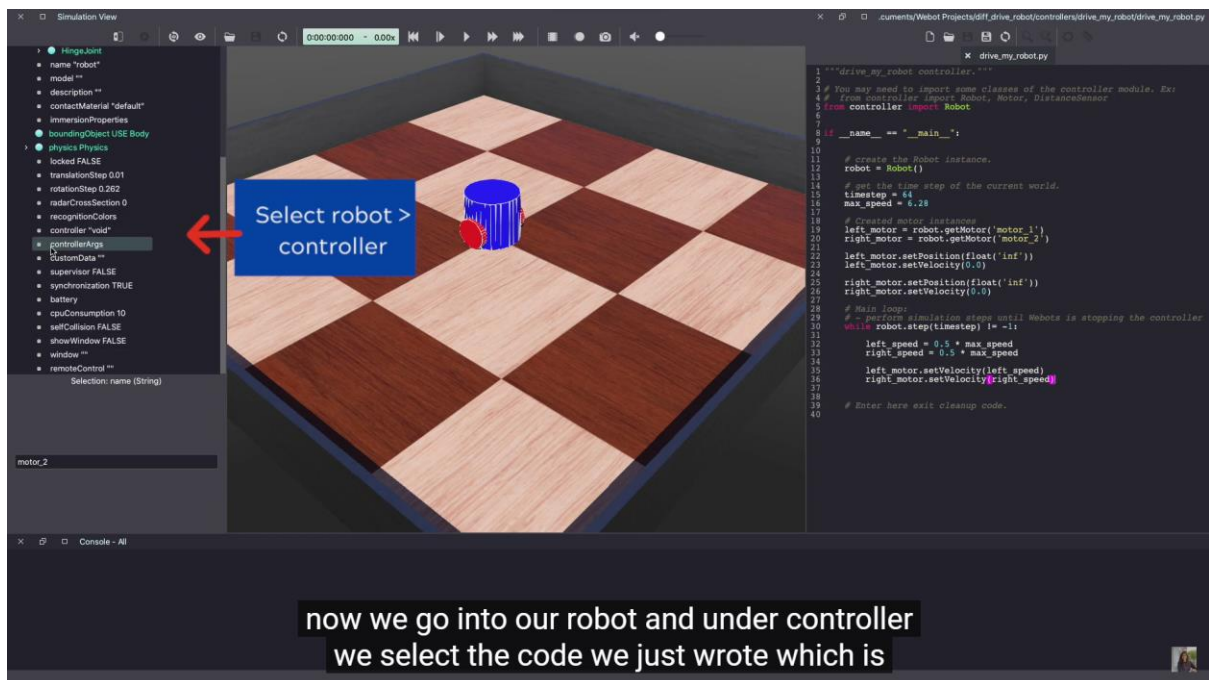
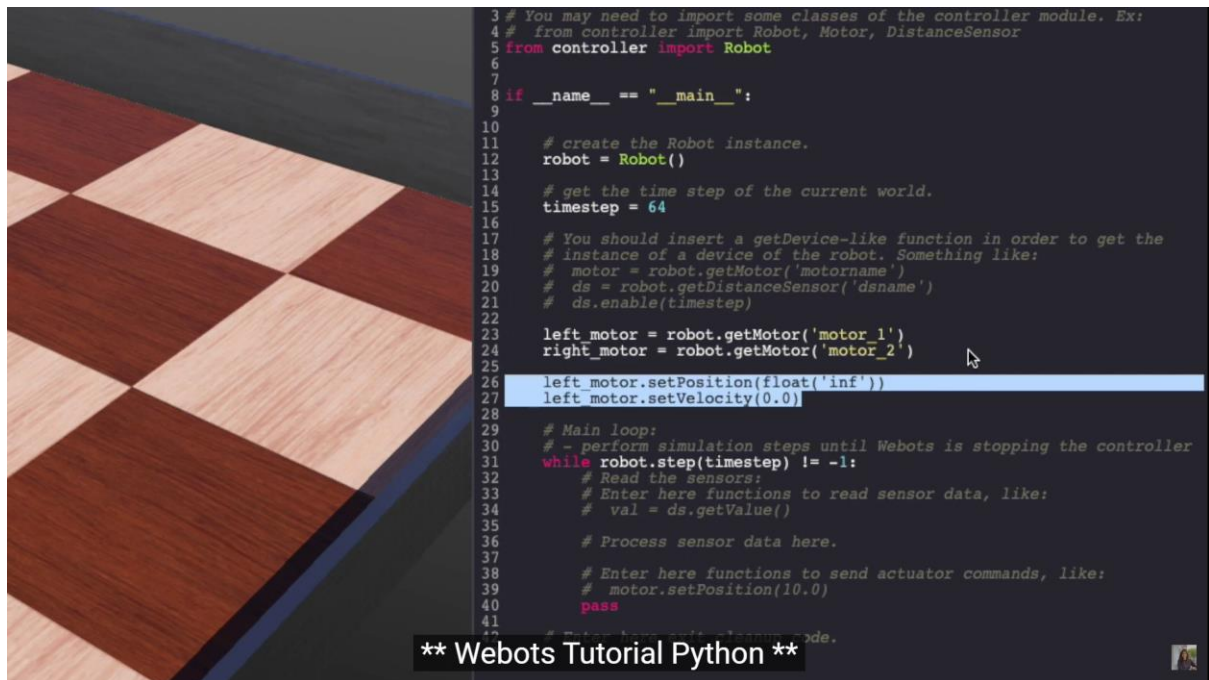
Tahap pertama, buat sistem control untuk robot webots dengan python



Pada gambar diatas terlihat tampilan code yang ada pada Bahasa python di visual code.



Pada perintah yang ada pada bahasa python diatas diperintahkan untuk membuat perintah baru dengan motor controller untuk menjalankan robot.



Lalu menambahkan driving straight atau berjalan lurus untuk robot setelah menambahkan perintah pada kodingan Bahasa python di sebelah kanan.



Simulation View

- HingeJoint
  - name "robot"
  - model ""
  - description ""
  - contactMaterial "default"
  - immersionProperties
  - boundingObject USE Body
  - physics Physics
    - locked FALSE
    - translationStep 0.01
    - rotationStep 0.252
    - radiusCrossSection 0
    - recognitionColors
    - controller "drive\_my\_robot"
    - controllerArgs
    - customData ""
    - supervisor FALSE
    - synchronization TRUE
    - battery
    - cpuConsumption 10
    - selfCollision FALSE
    - showWindow FALSE
    - window ""
    - remoteControl ""
- Selection: controller (String)

drive\_my\_robot

Select... Edit

INFO: drive\_my\_robot: Starting controller: python -u drive\_my\_robot.py  
DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 won't be maintained after that date. A future version of Webots will drop support for Python 2.7.

Save the current text file.

left\_speed = -0.5 \* max\_speed  
right\_speed = 0.5 \* max\_speed

speed to be positive so that it goes forward  
hit save and then play

Simulation View

- translation
  - DEF Body Transform
  - HingeJoint
    - jointParameters HingeJointParameters
      - position 0.85
      - axis 1 0 0
      - anchor -0.045 0.025 0
      - minStop 0
      - maxStop 0
      - springConstant 0
      - dampingCoefficient 0
      - staticFriction 0
      - suspensionSpringConstant 0
      - suspensionDampingConstant 0
      - suspensionAxis 1 0 0
    - device
      - RotationalMotor "motor\_2"
    - endPoint Solid
      - translation 0.045 0.025 -7.96e-07
      - rotation -0.701 0.698 0.147 2.85
  - Selection: translation (Vector3)

translation (Vector3)

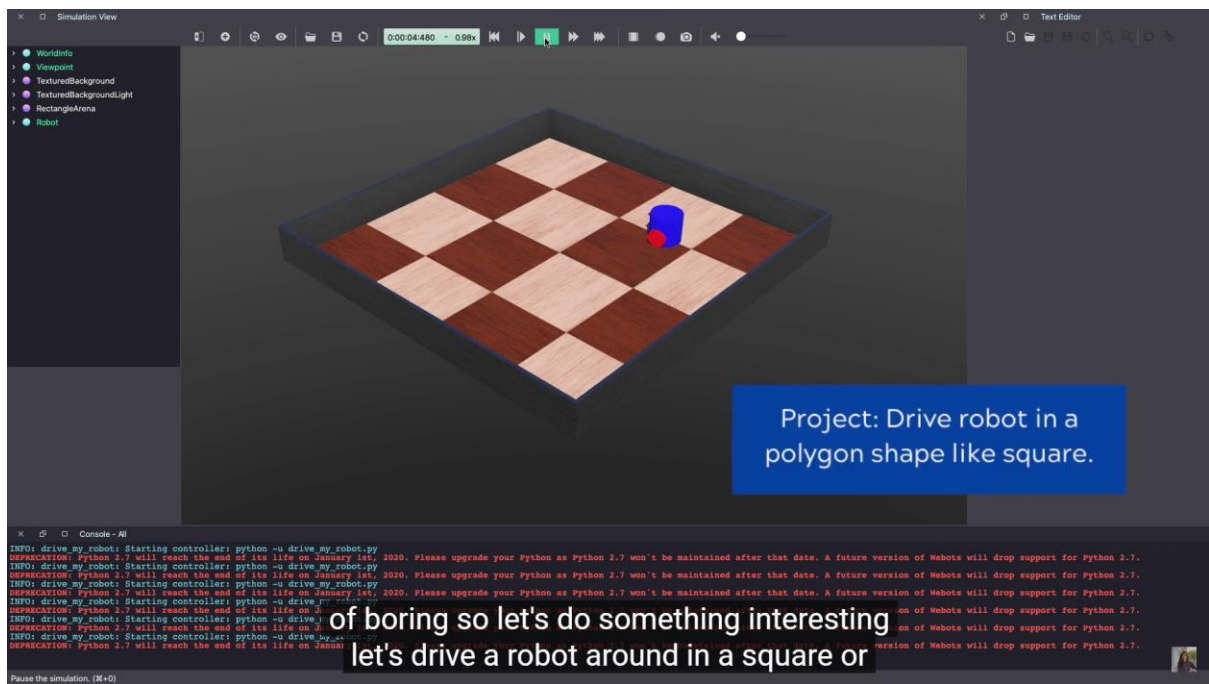
x: -0.045 m  
y: 0.025 m  
z: 0 m

INFO: drive\_my\_robot: Starting controller: python -u drive\_my\_robot.py  
DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 won't be maintained after that date. A future version of Webots will drop support for Python 2.7.

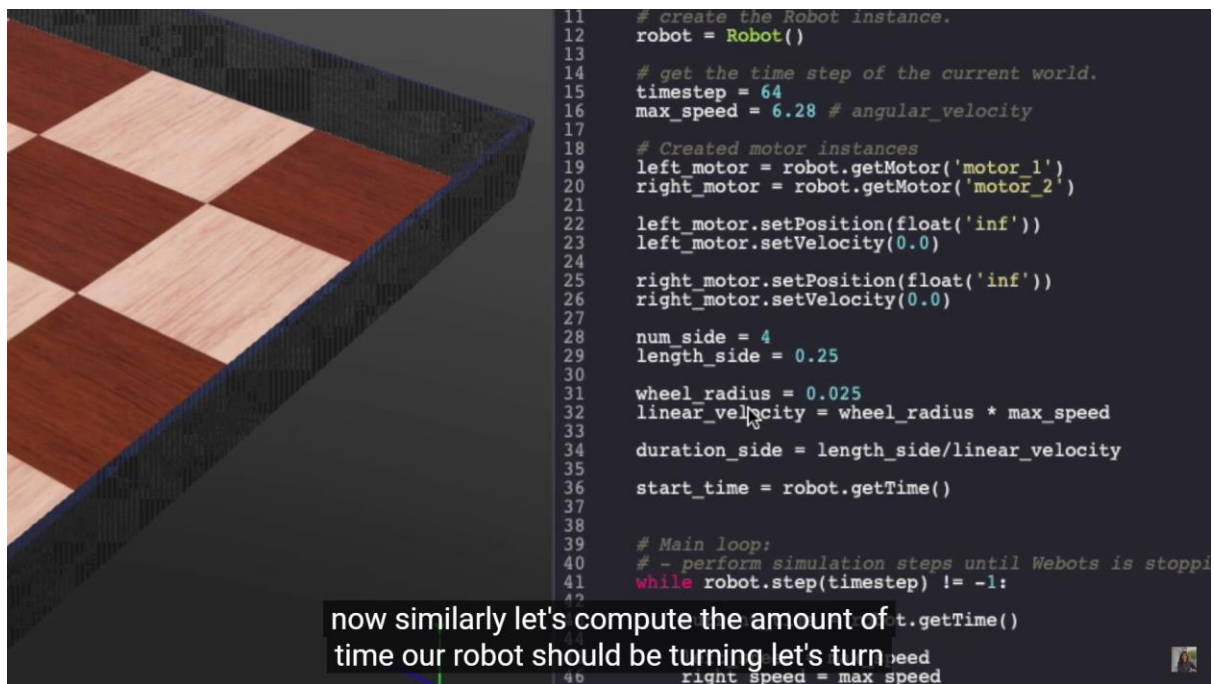
left\_speed = 0.25 \* max\_speed  
right\_speed = 0.5 \* max\_speed

and when we reverse the speed you can see  
the robot is now driving in the opposite direction

Sesuaikan kecepatan kanan atau kiri robot saat bergerak dengan mengganti perintah yang ada pada codingan Bahasa python.



Tampilan project robot saat bergerak memutar peta.



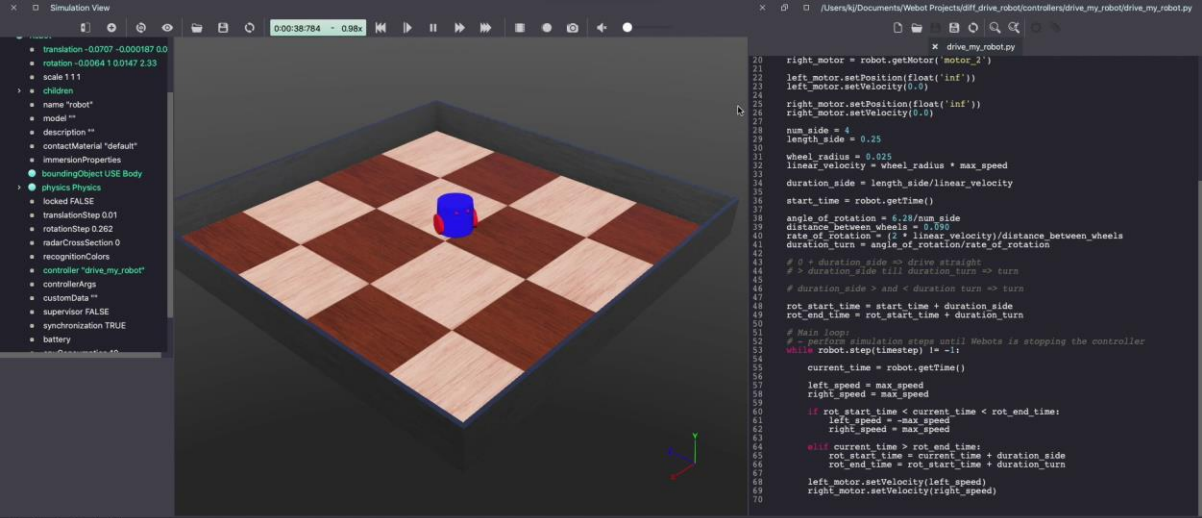
```

44 # > duration_side till duration_turn => turn
45
46 # duration_side > and < duration_turn => turn
47
48 rot_start_time = start_time + duration_side
49 rot_end_time = rot_start_time + duration_turn
50
51 # Main loop:
52 # - perform simulation steps until Webots is stopping the controller
53 while robot.step(timestep) != -1:
54
55     current_time = robot.getTime()
56
57     left_speed = max_speed
58     right_speed = max_speed
59
60     if rot_start_time < current_time < rot_end_time:
61         left_speed = -max_speed
62         right_speed = max_speed
63
64     elif current_time > rot_end_time:
65         rot_start_time = current_time + duration_side
66
67     left_motor.setVelocity(left_speed)
68     right_motor.setVelocity(right_speed)

```

recomputing our time for rotation for upcoming  
 ned after that date. A rotations let's save our work and see what: for Python 2.7.

Tambahkan beberapa kodingan untuk menjalankan polygon shape pada robot webots.



```

20 right_motor = robot.getMotor('motor_2')
21 left_motor.setPosition(float('inf'))
22 left_motor.setVelocity(0.0)
23 right_motor.setPosition(float('inf'))
24 right_motor.setVelocity(0.0)
25
26 num_side = 4
27 length_side = 0.25
28
29 wheel_radius = 0.025
30 linear_velocity = wheel_radius * max_speed
31 duration_side = length_side/linear_velocity
32
33 start_time = robot.getTime()
34
35 angle_of_rotation = 6.28/num_side
36 distance_between_wheels = 0.090
37 rate_of_rotation = 2 * linear_velocity/distance_between_wheels
38 duration_turn = angle_of_rotation/rate_of_rotation
39
40 # 0 < duration_side => drive straight
41 # > duration_side till duration_turn => turn
42
43 # duration_side > and < duration_turn => turn
44
45 rot_start_time = start_time + duration_side
46 rot_end_time = rot_start_time + duration_turn
47
48 # Main loop:
49 # - perform simulation steps until Webots is stopping the controller
50 while robot.step(timestep) != -1:
51
52     current_time = robot.getTime()
53
54     left_speed = max_speed
55     right_speed = max_speed
56
57     if rot_start_time < current_time < rot_end_time:
58         left_speed = -max_speed
59         right_speed = max_speed
60
61     elif current_time > rot_end_time:
62         rot_start_time = current_time + duration_side
63         rot_end_time = rot_start_time + duration_turn
64
65     left_motor.setVelocity(left_speed)
66     right_motor.setVelocity(right_speed)
67
68
69

```

robot itself has no idea about its own state  
 or the environment to correct our action we

Setelah di run maka akan terlihat open loop yang terjadi pada robot dan juga kebalikannya closed loop system pada robot.