

Webots Tutorial Series in Python Summary

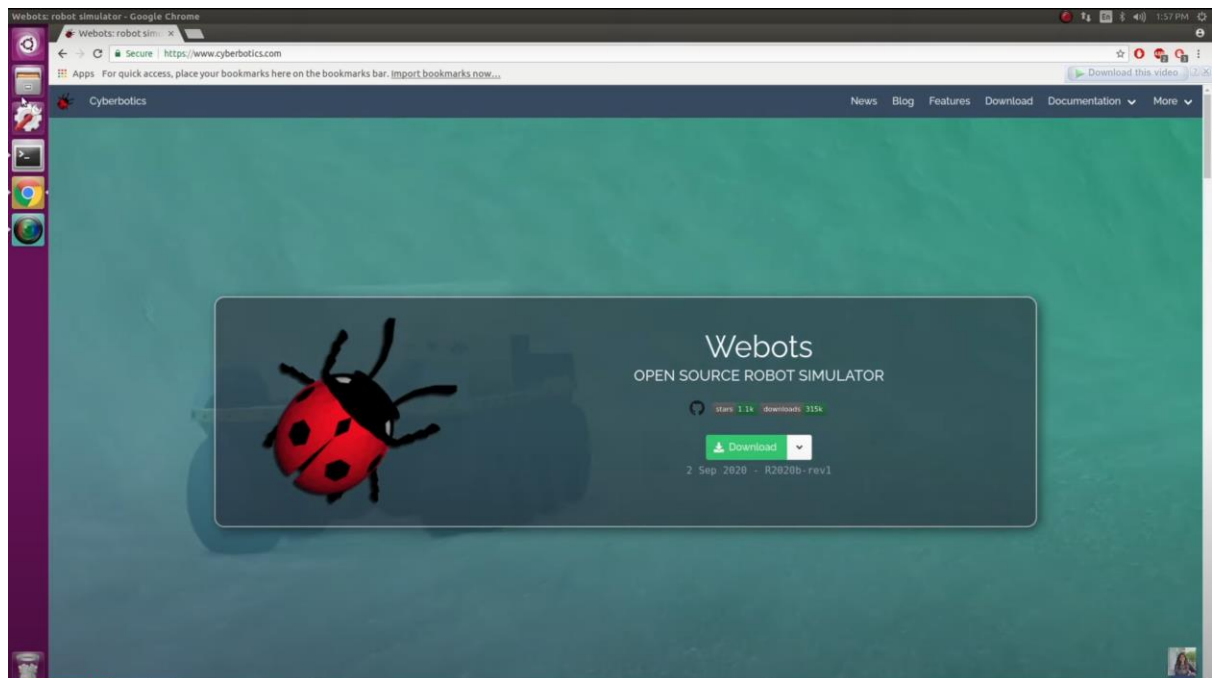
Nama : Al Ghifary Akmal Nasheeri

NIM : 1103201242

Kelas : TK-44-06

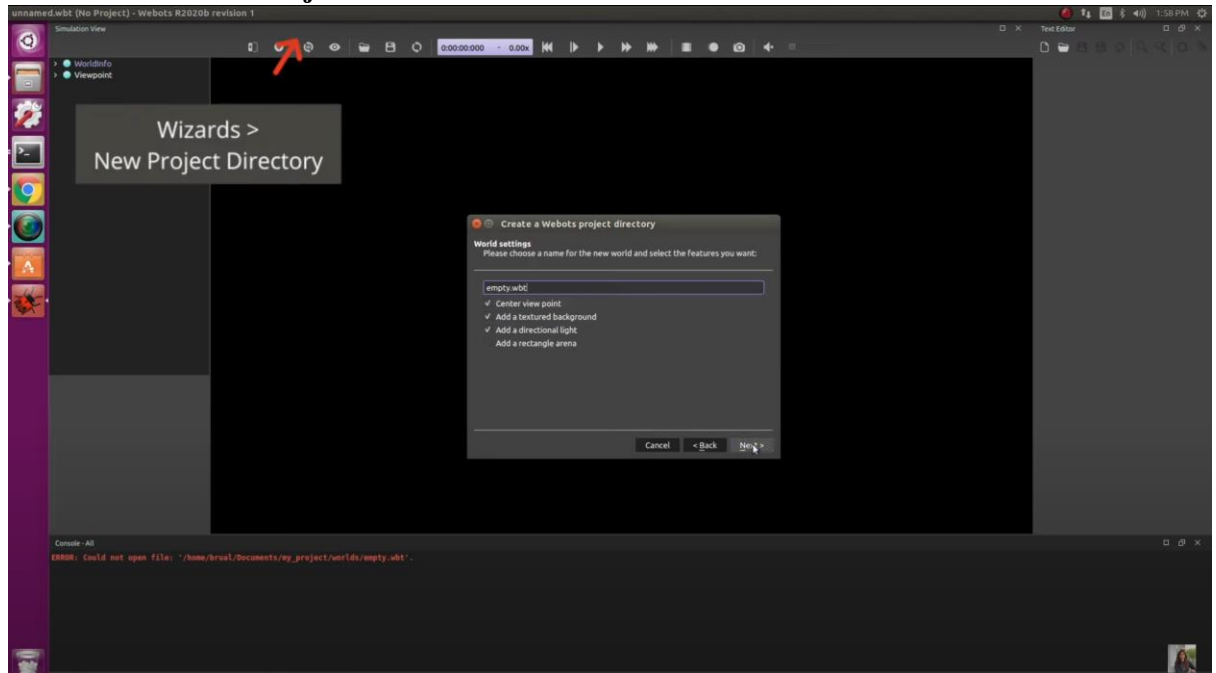
Introduction to webots: how to install and run your first simulation in 10 min | Webots Tutorial 1

1. Download and Install:



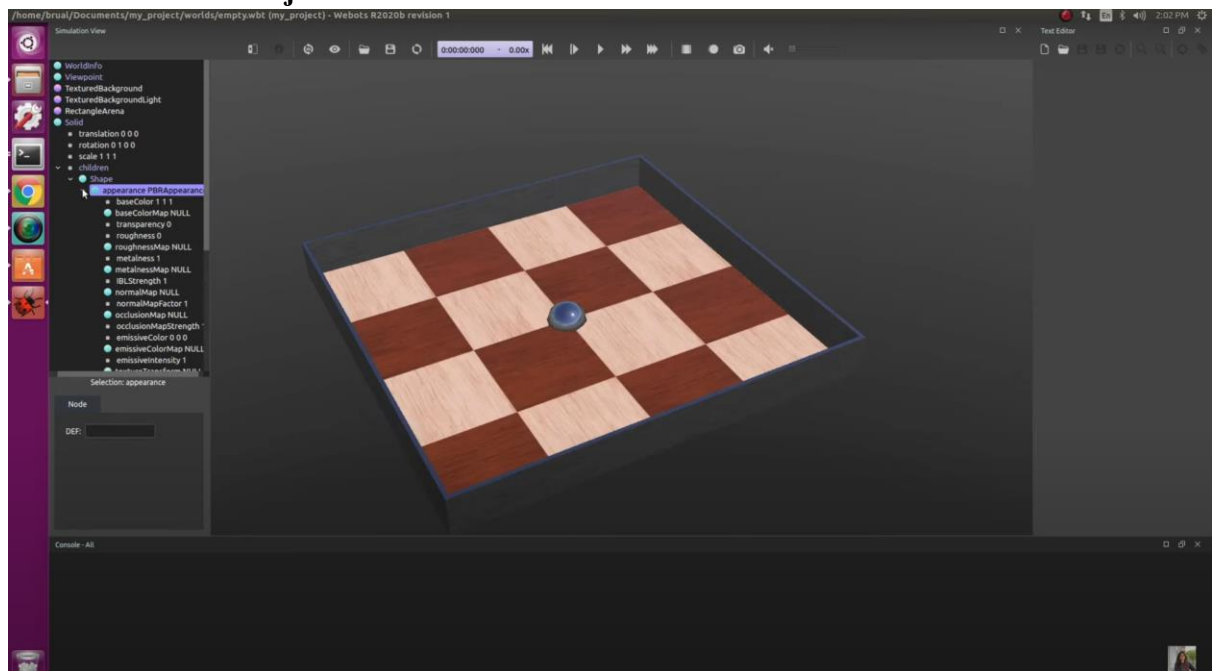
Pertama, langkah awal dalam tutorial Webots ini adalah mengunduh dan menginstal aplikasi. Video memberikan panduan langkah demi langkah untuk memastikan pengguna dapat dengan mudah mengakses Webots di sistem mereka.

2. Create Your First Project:



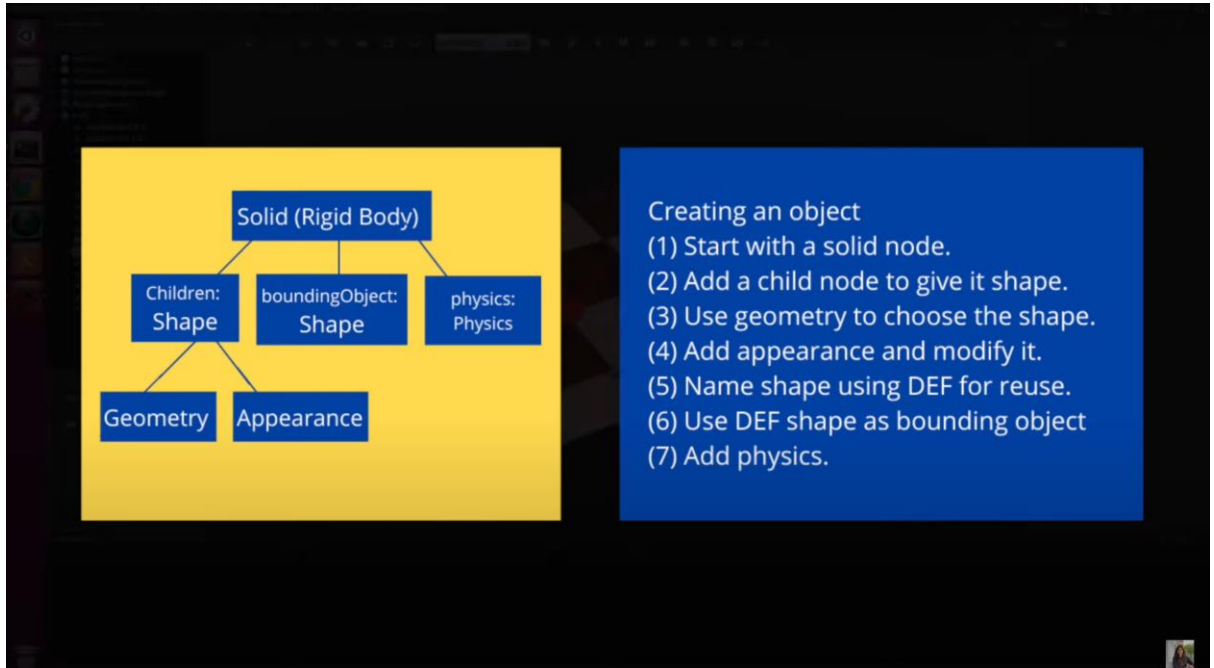
Setelah instalasi selesai, langkah berikutnya adalah membuat proyek pertama Anda. Tutorial ini memberikan instruksi tentang cara membuat proyek di Webots, memberikan dasar bagi pengguna untuk memulai eksplorasi dan pengembangan di lingkungan simulasi.

3. Make Your Own Object:



Video ini mengajarkan cara membuat objek khusus sesuai kebutuhan Anda. Ini termasuk langkah-langkah untuk membuat objek 3D atau menggunakan objek bawaan Webots, memberi pengguna fleksibilitas dalam desain simulasi mereka.

4. Understand Fundamentals:



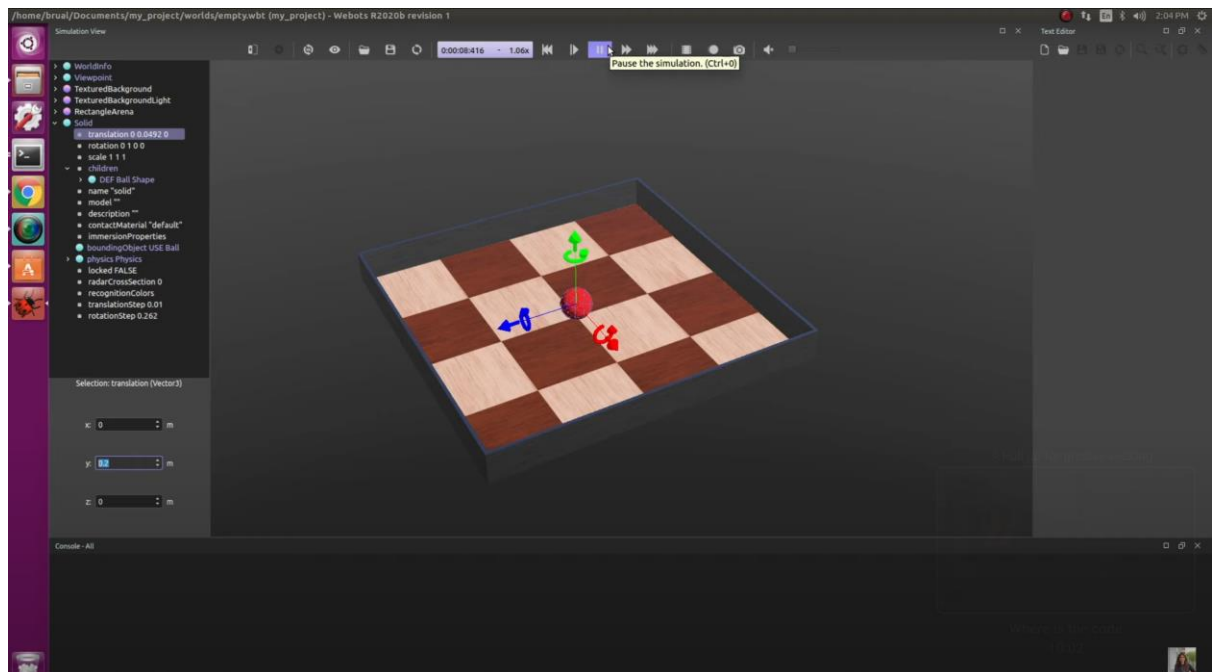
The diagram illustrates the structure of a Solid (Rigid Body) in Webots. It is a hierarchical tree where the root node is 'Solid (Rigid Body)'. This node has three children: 'Children: Shape', 'boundingObject: Shape', and 'physics: Physics'. The 'Children: Shape' node further branches into 'Geometry' and 'Appearance'.

Creating an object

- (1) Start with a solid node.
- (2) Add a child node to give it shape.
- (3) Use geometry to choose the shape.
- (4) Add appearance and modify it.
- (5) Name shape using DEF for reuse.
- (6) Use DEF shape as bounding object
- (7) Add physics.

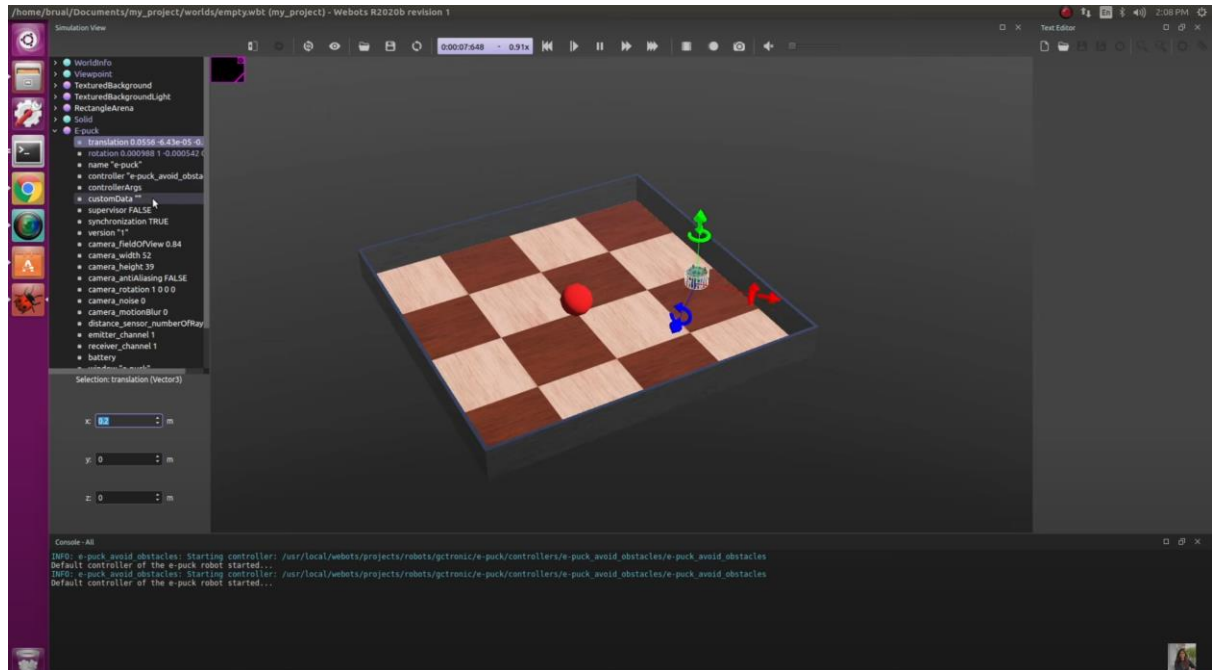
Langkah ini memberikan pemahaman dasar tentang konsep-konsep kunci Webots. Pengguna diberikan penjelasan singkat tentang konsep simulasi fisika, sensor, dan elemen-elemen fundamental lainnya yang membentuk lingkungan simulasi Webots.

5. Run Your First Simulation:



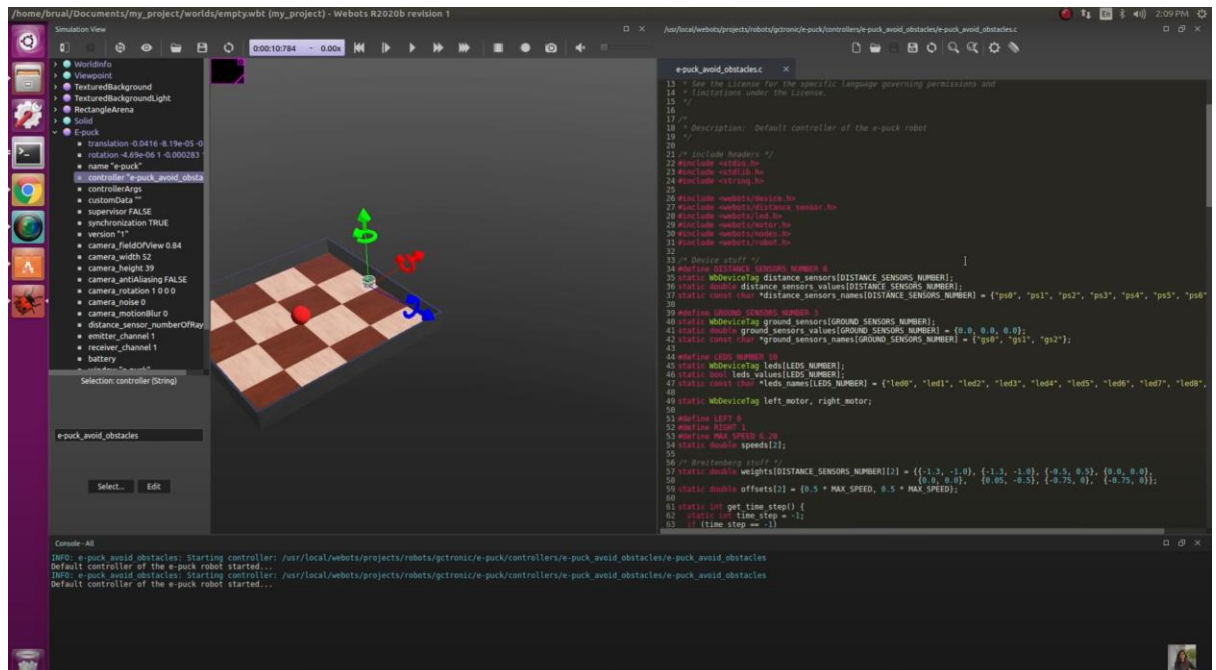
Setelah memahami dasar-dasar, langkah selanjutnya adalah menjalankan simulasi pertama. Tutorial ini membimbing pengguna melalui proses ini, memastikan bahwa pengguna dapat melihat hasil simulasi mereka dengan sukses.

6. Add a Robot & Run Simulation:



Pada tahap ini, pengguna diajak untuk menambahkan robot ke dalam simulasi mereka. Video memberikan petunjuk tentang cara menambahkan robot ke lingkungan simulasi Webots dan menjalankan simulasi dengan kehadiran robot tersebut.

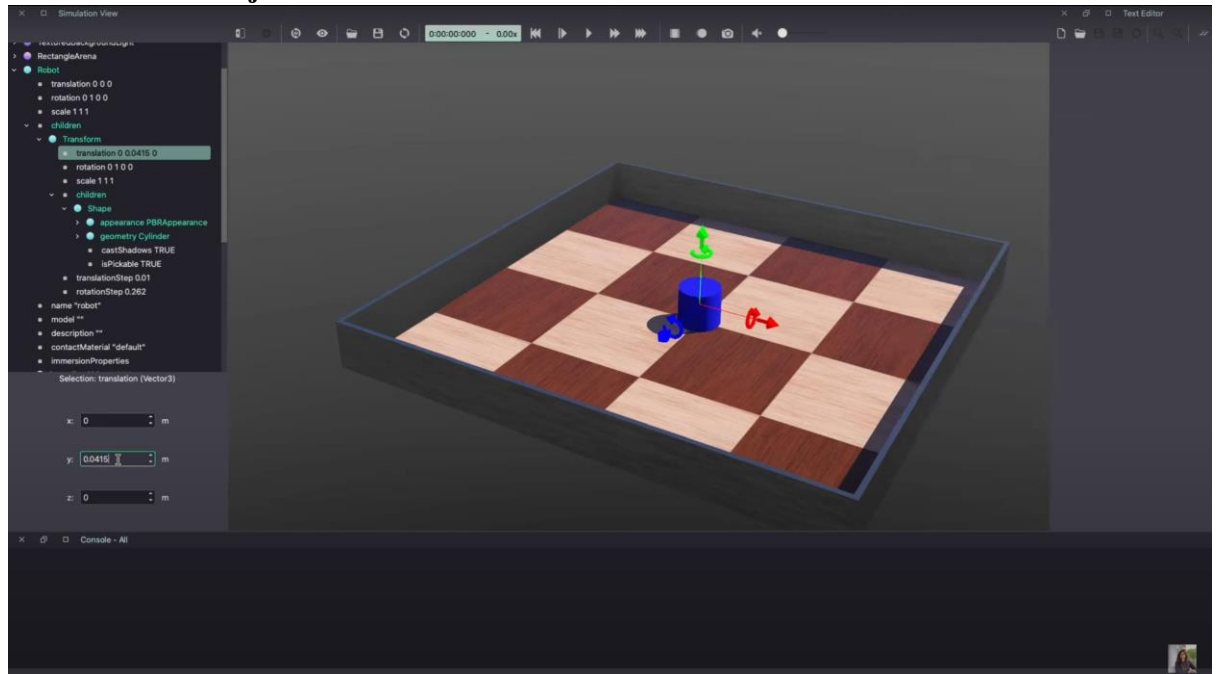
7. Where Is The Code:



Terakhir, tutorial menjelaskan di mana kode-kode terkait simulasi disimpan. Pengguna diberikan wawasan tentang struktur direktori proyek dan di mana mereka dapat menemukan dan mengelola kode-kode yang berkaitan dengan simulasi yang telah mereka buat.

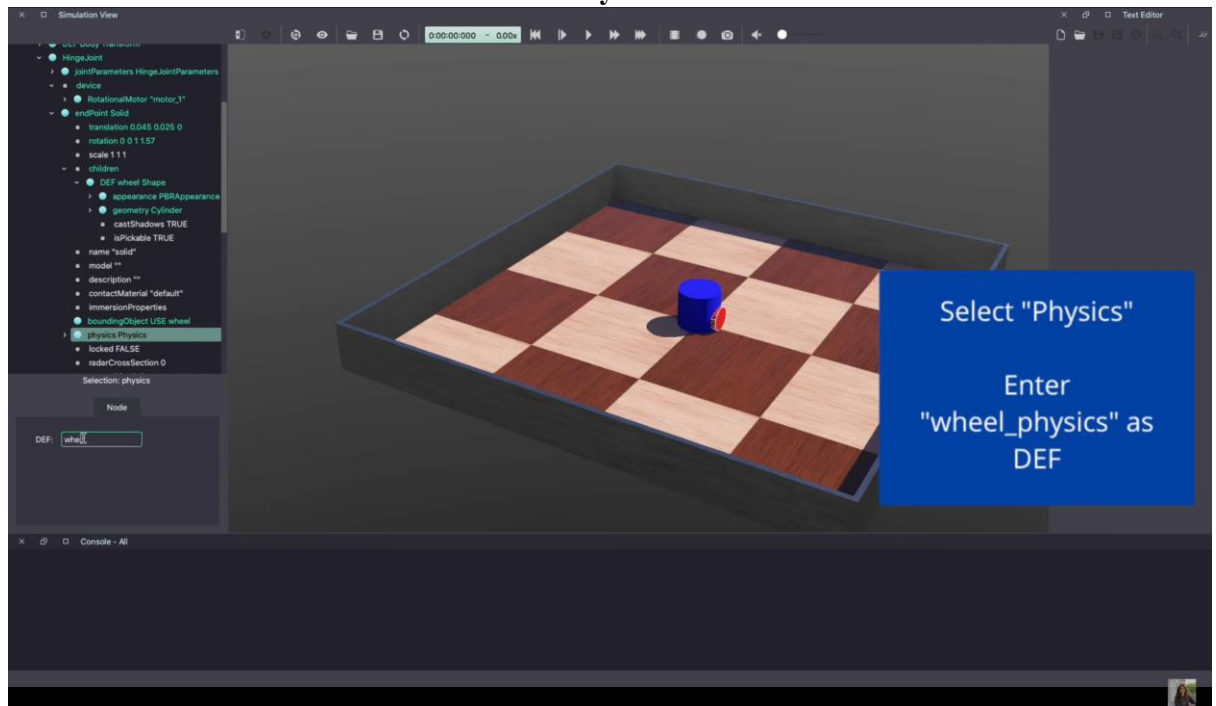
How to design a 2 wheel differential drive robot in Webots? // Webots tutorial 2 // Kaja Gada

1. Create a New Project in Webots



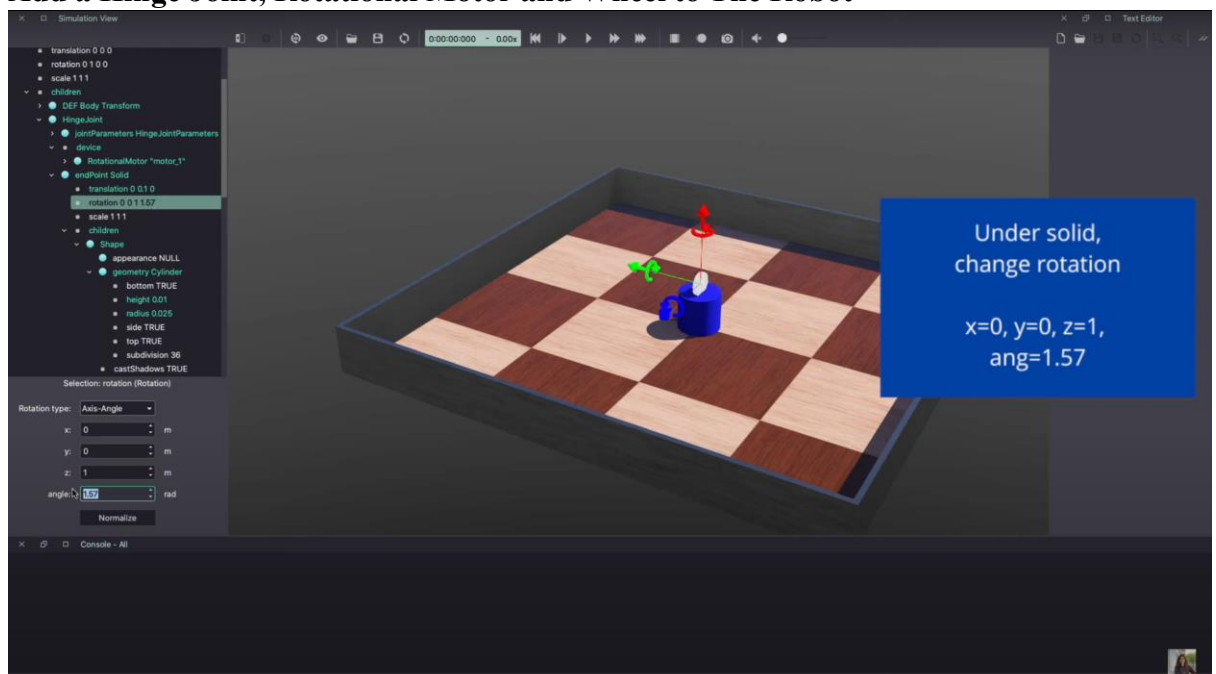
Tutorial ini dimulai dengan langkah pertama dalam membuat robot penggerak differential drive di Webots, yaitu membuat proyek baru. Penjelasan detail diberikan untuk memastikan pengguna dapat membuat proyek dengan benar dan memasuki tahap desain robot.

2. Add a Robot Node and Create a Robot Body



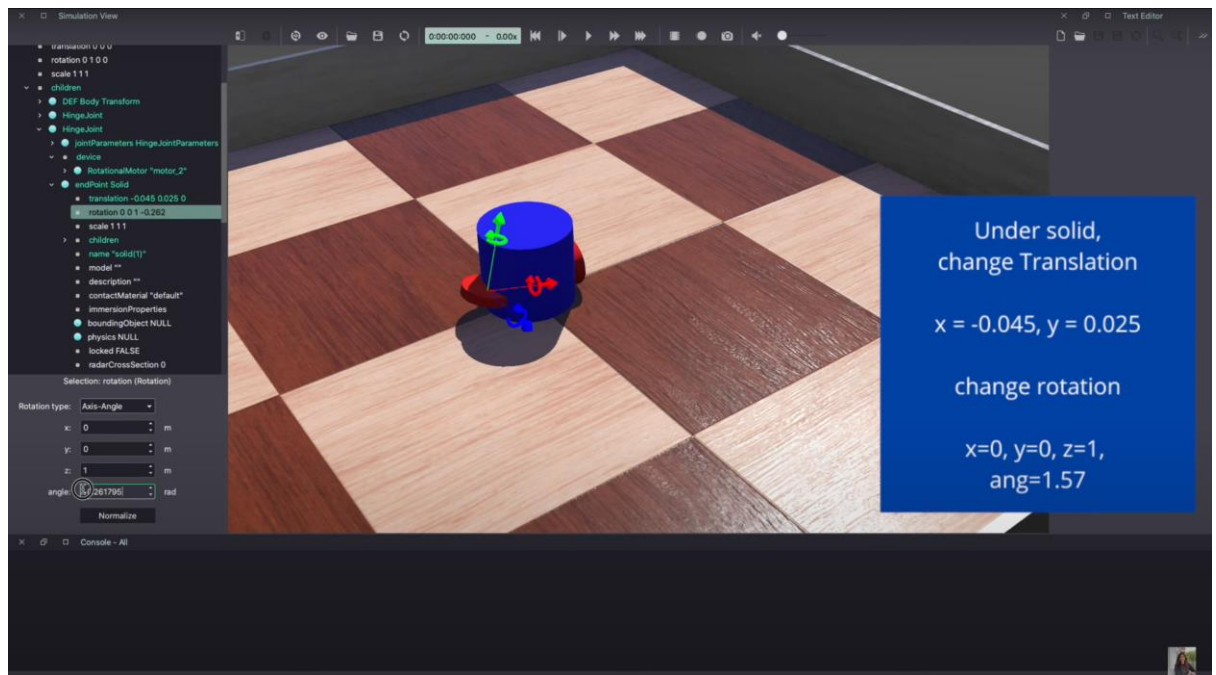
Langkah berikutnya melibatkan penambahan node robot dan pembuatan struktur tubuh robot. Pengguna dipandu melalui proses menambahkan elemen-elemen dasar yang membentuk kerangka robot, memberikan fondasi untuk desain differential drive.

3. Add a Hinge Joint, Rotational Motor and Wheel to The Robot



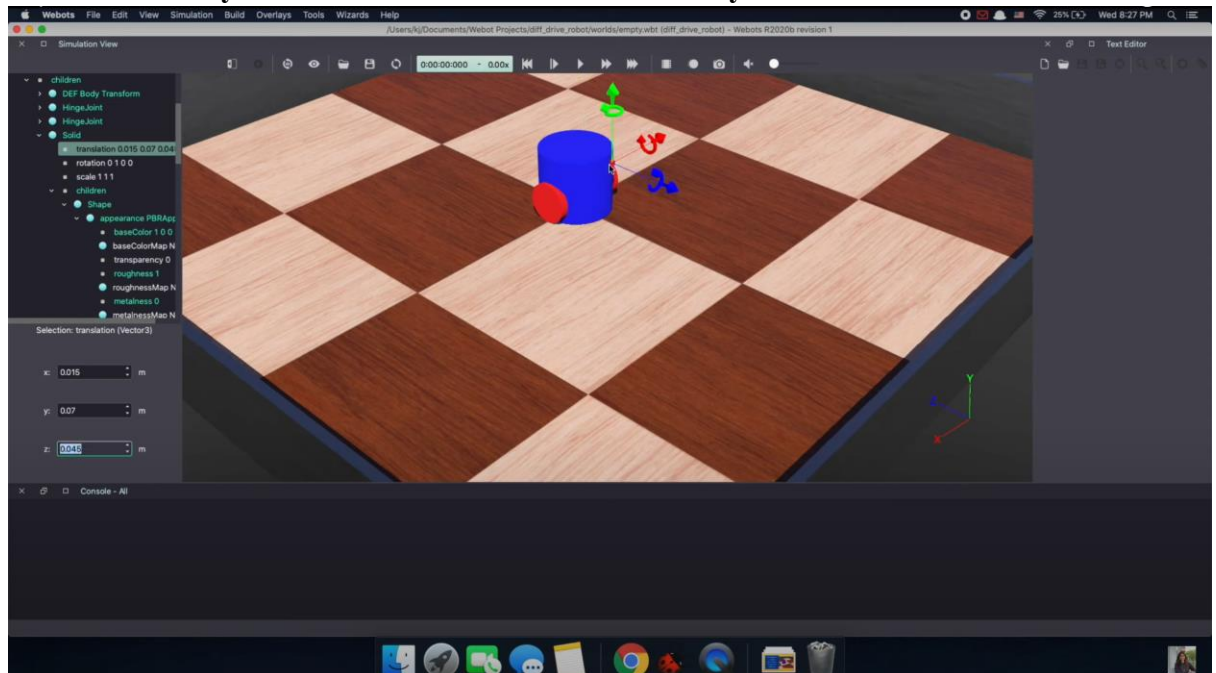
Tutorial ini menitikberatkan pada konfigurasi komponen-komponen kunci seperti Hinge Joint, Rotational Motor, dan Wheel pada robot. Pengguna diberikan langkah-langkah untuk menambahkan dan mengonfigurasi setiap komponen ini untuk membangun struktur dasar robot penggerak differential drive.

4. Add a Second Wheel to The Differential Drive Robot



Langkah selanjutnya mencakup penambahan roda kedua pada robot differential drive. Tutorial ini membimbing pengguna dalam menyesuaikan konfigurasi dan memastikan bahwa robot memiliki dua roda yang sesuai dengan prinsip penggerak differential drive.

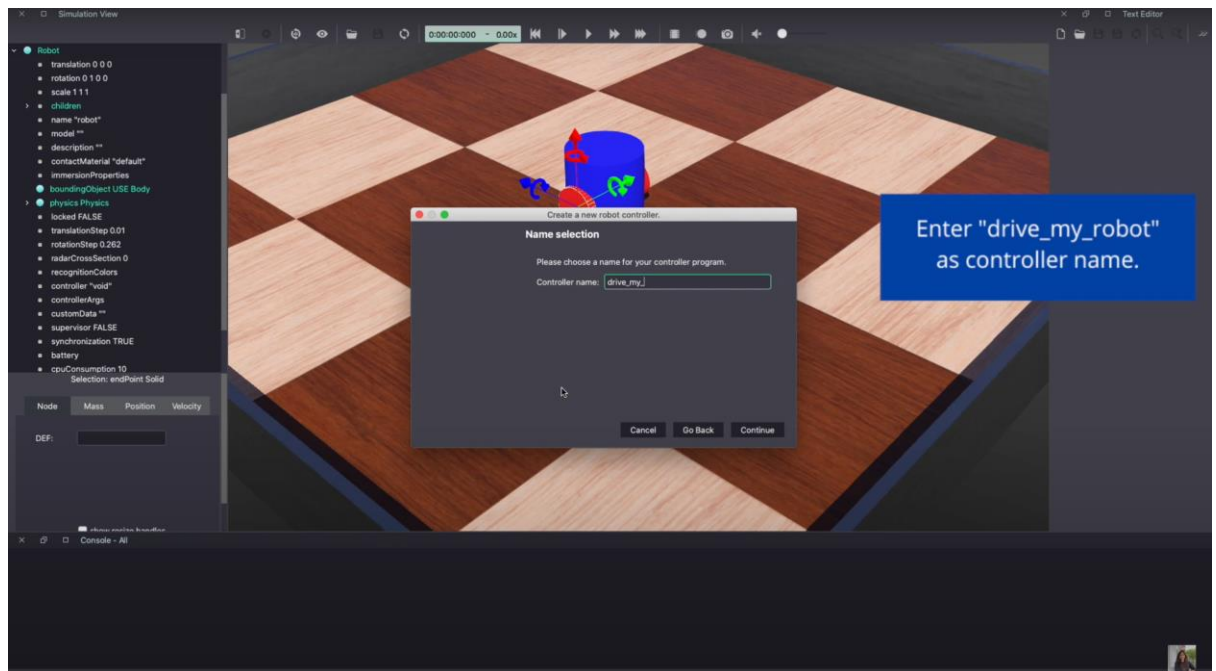
5. Add a Pair of Eyes to Make Our Robot Look Friendly



Selain aspek teknis, tutorial juga mengajarkan cara menambahkan elemen desain ke robot. Pada langkah ini, pengguna diajak untuk menambahkan sepasang mata untuk memberikan karakter dan penampilan yang ramah pada robot yang telah dibuat.

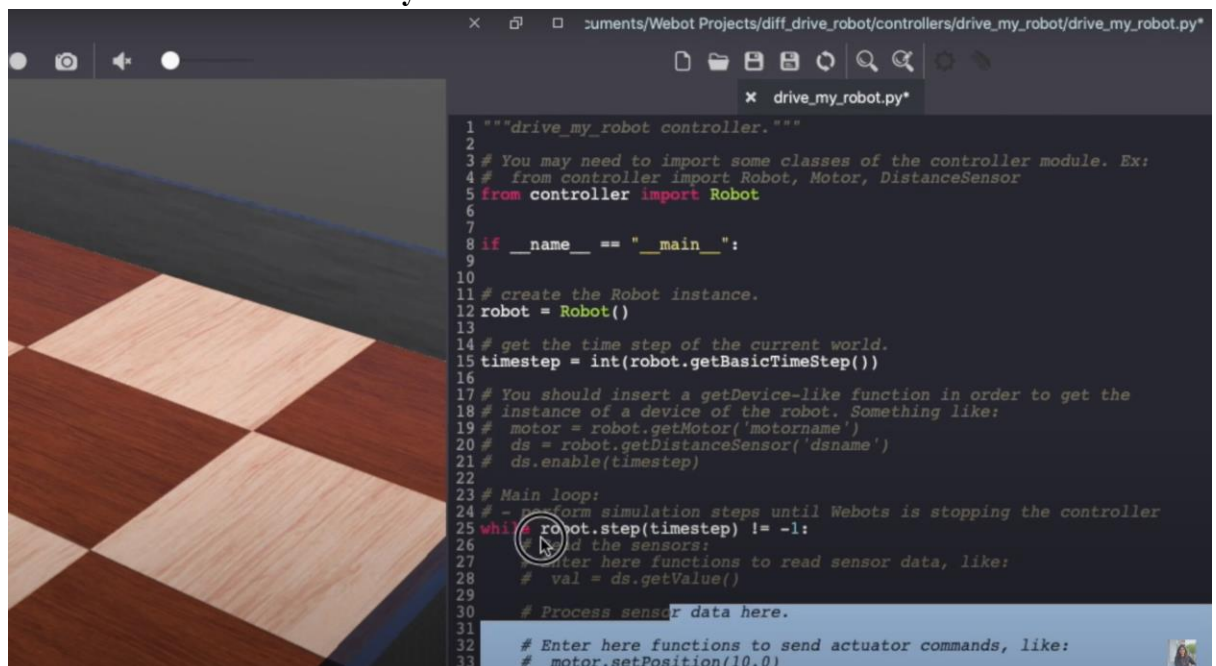
Webots tutorial 3: Controller code to drive a differential drive robot // Webots Tutorial Python

1. Create Controller in Webots



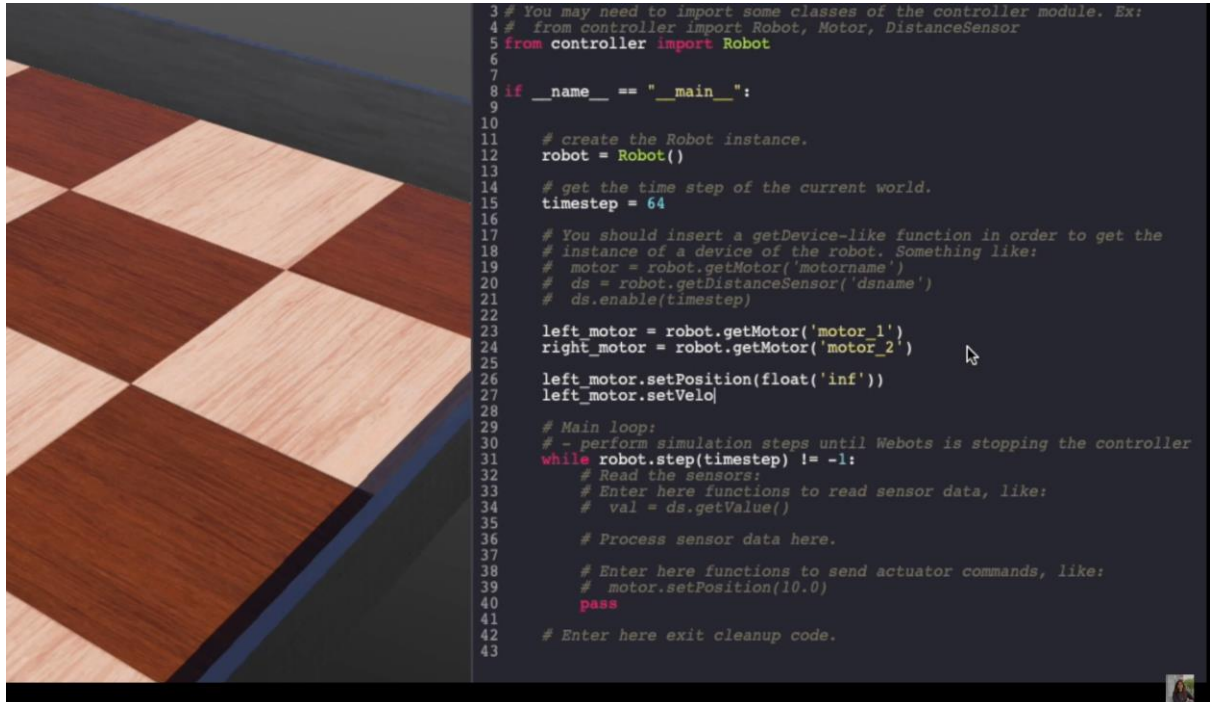
Pembelajaran dimulai dengan langkah pertama yaitu membuat controller di Webots. Tutorial memberikan panduan langkah demi langkah untuk menciptakan kontroler yang diperlukan untuk mengendalikan robot penggerak differential drive.

2. Webots Controller Code in Python



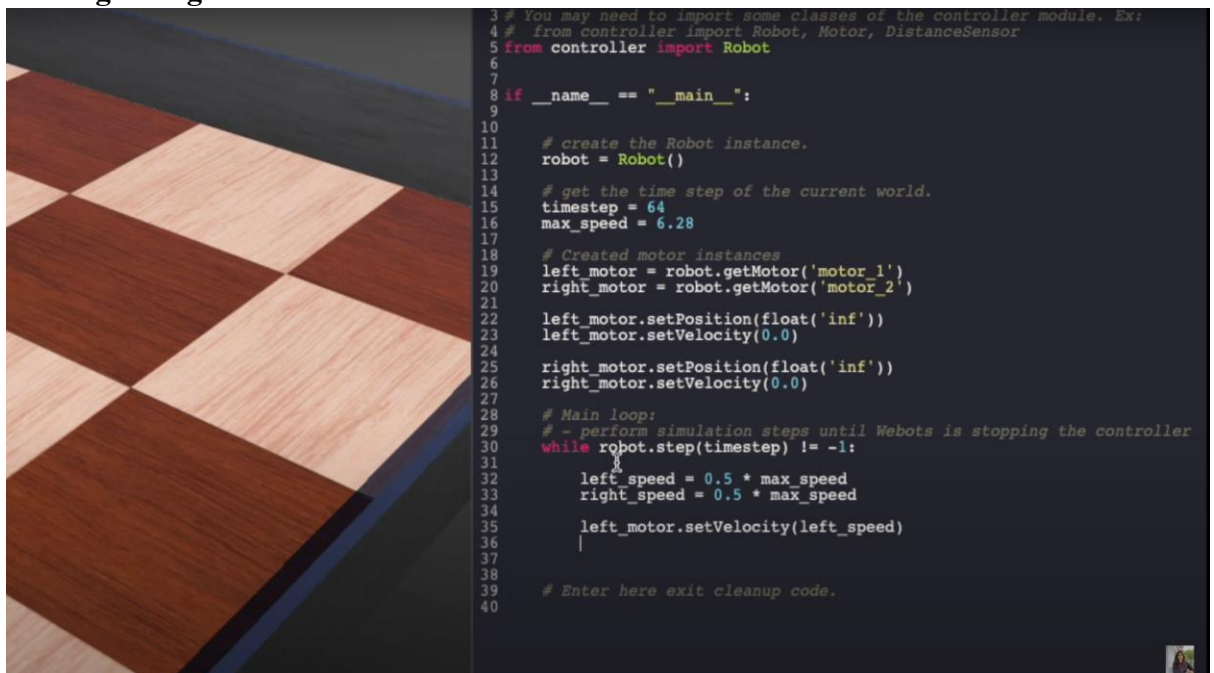
Pada tahap ini, fokus diberikan pada penulisan kode kontroler menggunakan bahasa pemrograman Python di Webots. Tutorial ini memberikan contoh kode dan menjelaskan sintaksis yang digunakan untuk mengimplementasikan kontroler pada simulasi robot.

3. Motor Control in Webots



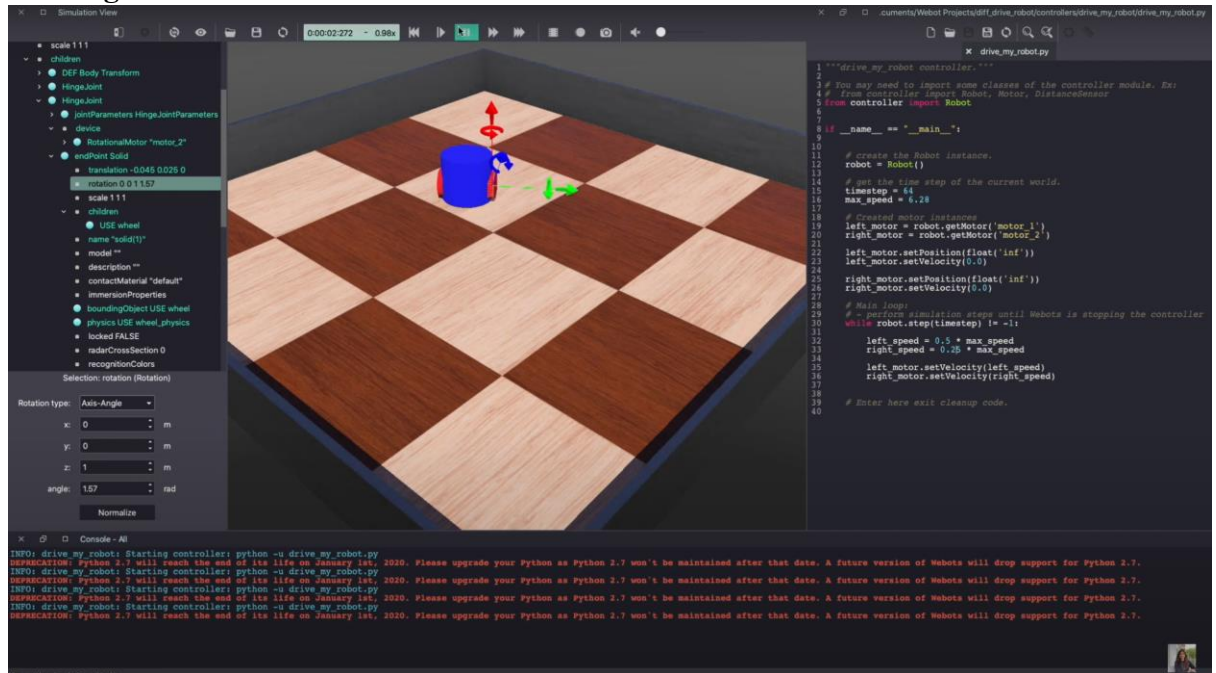
Langkah berikutnya adalah memahami kontrol motor dalam Webots. Video memberikan penjelasan mendalam tentang cara mengontrol motor pada robot penggerak differential drive dan memastikan pengguna dapat mengimplementasikannya dengan benar.

4. Driving Straight



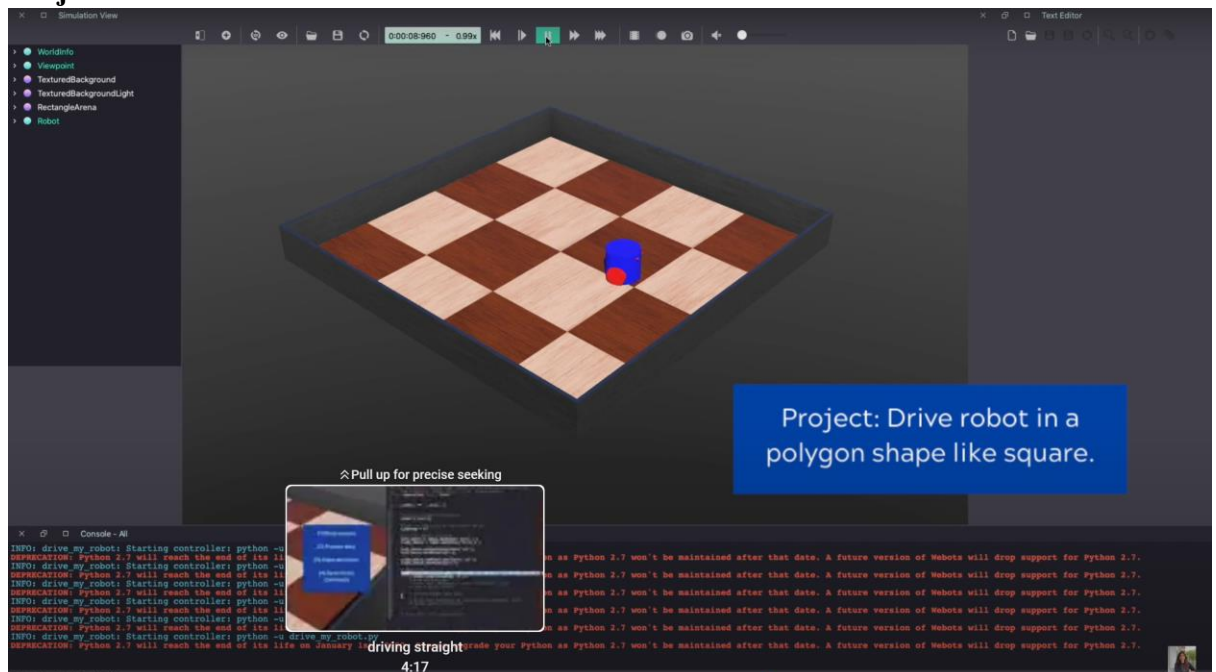
Tutorial ini melibatkan implementasi kode untuk mengarahkan robot secara lurus. Pengguna diajak untuk memahami bagaimana mengontrol kecepatan roda agar robot dapat bergerak dengan lurus pada simulasi Webots.

5. Turning Robot



Langkah ini menitikberatkan pada kemampuan mengendalikan putaran robot. Video memberikan contoh pengkodean untuk memastikan robot dapat berputar dengan kontrol yang diinginkan di lingkungan simulasi Webots.

6. Project Details



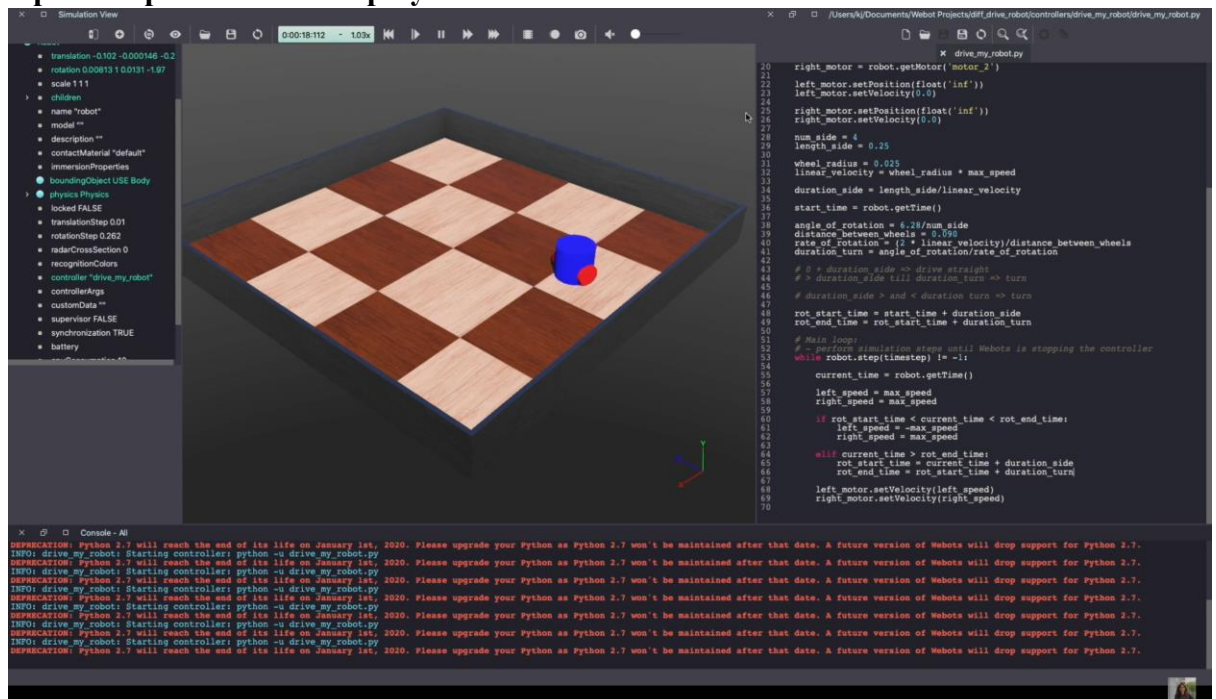
Bagian ini memberikan rincian tambahan tentang proyek yang sedang dikembangkan, termasuk aspek-aspek teknis dan desain yang telah diimplementasikan pada robot penggerak differential drive.

7. Drive in Polygon Shape

```
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
66         left_motor.setVelocity(left_speed)
67         right_motor.setVelocity(right_speed)
68
69
70 ned after that date. A future version of Webots will drop support for Python 2.7.
```

Pengguna diajak untuk mengimplementasikan kemampuan robot untuk bergerak dalam pola bentuk poligon. Tutorial memberikan panduan langkah demi langkah untuk mengkodekan kontroler agar robot dapat mengikuti pola tertentu.

8. Open Loop Vs Closed Loop System



```
20 right_motor = robot.getMotor('motor_2')
21
22 left_motor.setPosition(float('inf'))
23 left_motor.setVelocity(0.0)
24
25 right_motor.setPosition(float('inf'))
26 right_motor.setVelocity(0.0)
27
28 num_side = 4
29 length_side = 0.25
30
31 wheel_radius = 0.025
32 linear_velocity = wheel_radius * max_speed
33
34 duration_side = length_side/linear_velocity
35
36 start_time = robot.getTime()
37
38 angle_of_rotation = 0.28/num_side
39 distance_between_wheels = 0.095
40 rate_of_rotation = (2 * linear_velocity)/distance_between_wheels
41 duration_turn = angle_of_rotation/rate_of_rotation
42
43 # > duration_side => drive straight
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
66         left_motor.setVelocity(left_speed)
67         right_motor.setVelocity(right_speed)
68
69
70
```

Video ini membahas perbandingan antara sistem open loop dan closed loop dalam konteks pengendalian robot. Pengguna diberikan pemahaman tentang perbedaan kunci antara kedua sistem ini dan kapan harus menggunakan masing-masing dalam pengembangan robotik.

