

NAMA : Albizhar Zidane Budi Laksana
NIM : 1103202116
KELAS : TK-44-G7

“Lecture Week 10”

REPORT WEEK 10 Video 1-3

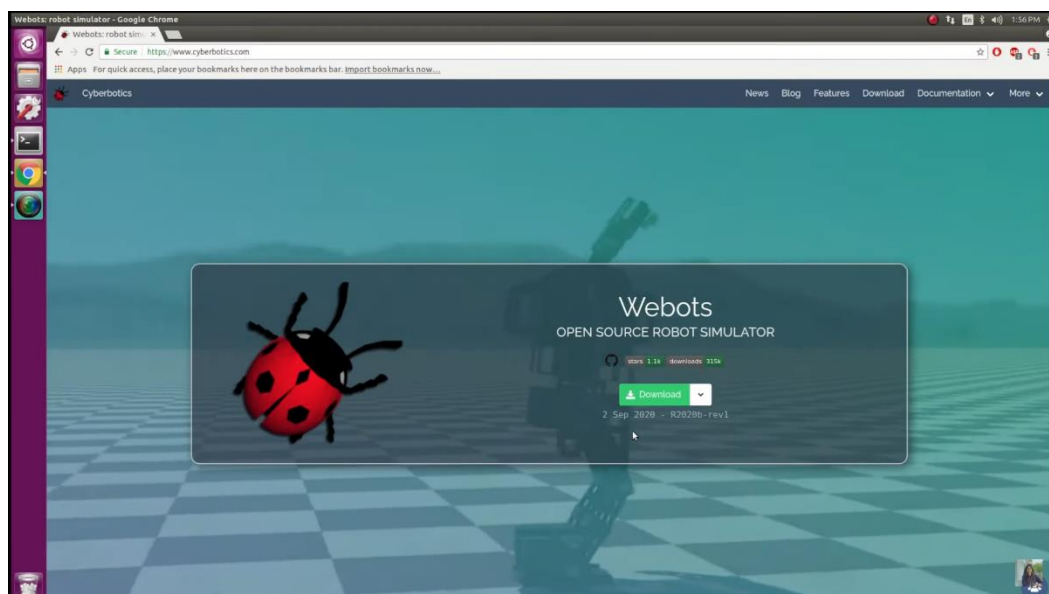
Video 1 : Introduction to webots: how to install and run your first simulation in 10 min | Webots Tutorial 1



https://www.youtube.com/watch?v=luyg3plGuig&list=PLbEU0vp_OQkUwANRMUOM00SXybYQ4TXNF&index=2&ab_channel=KajalGada

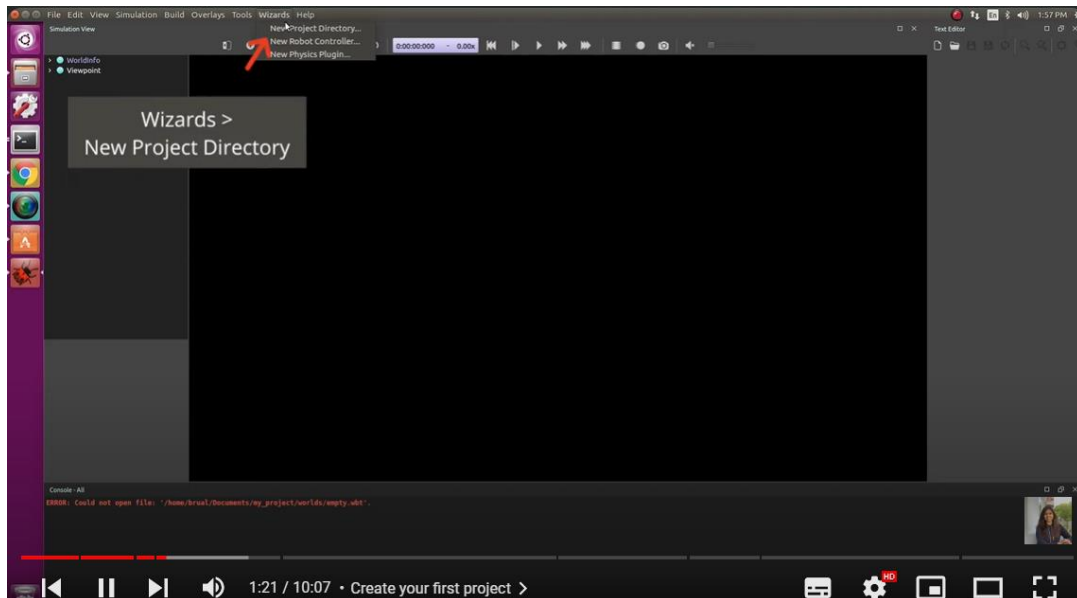
1. Download and Install

Tahap permulaan dalam tutorial Webots ini dimulai dengan mengunduh serta menginstal program aplikasi. Video memberikan panduan rinci langkah demi langkah untuk memastikan pengguna dapat dengan lancar mengakses Webots pada sistem komputer mereka.



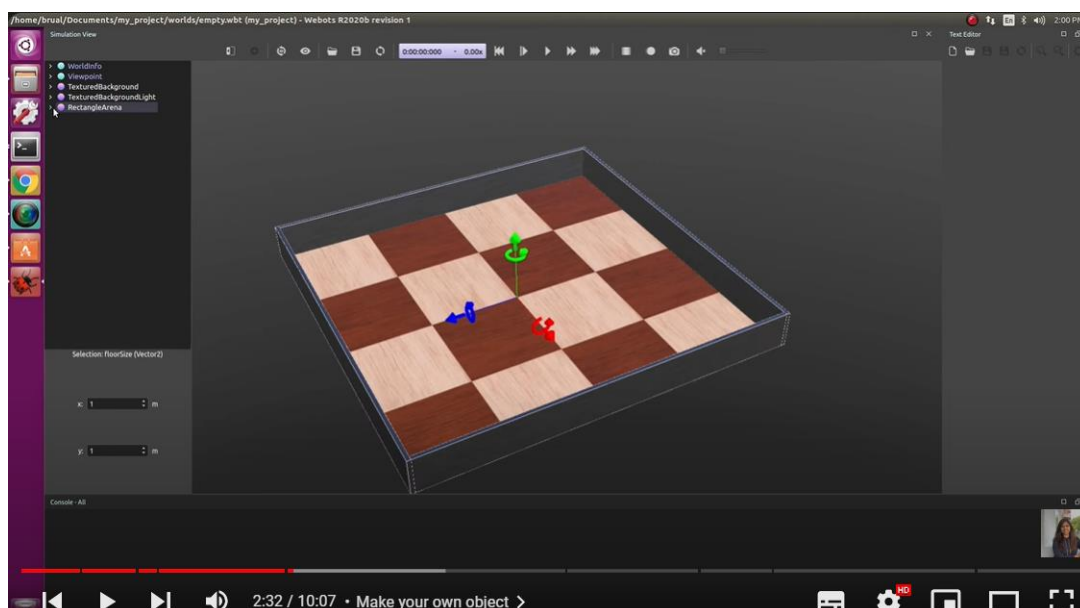
2. Create Project

Setelah selesai melakukan instalasi, langkah selanjutnya adalah memulai proyek awal Anda. Panduan ini memberikan instruksi rinci tentang cara menciptakan proyek di Webots, memberikan landasan bagi pengguna untuk memulai penjelajahan dan pengembangan di dalam lingkungan simulasi.



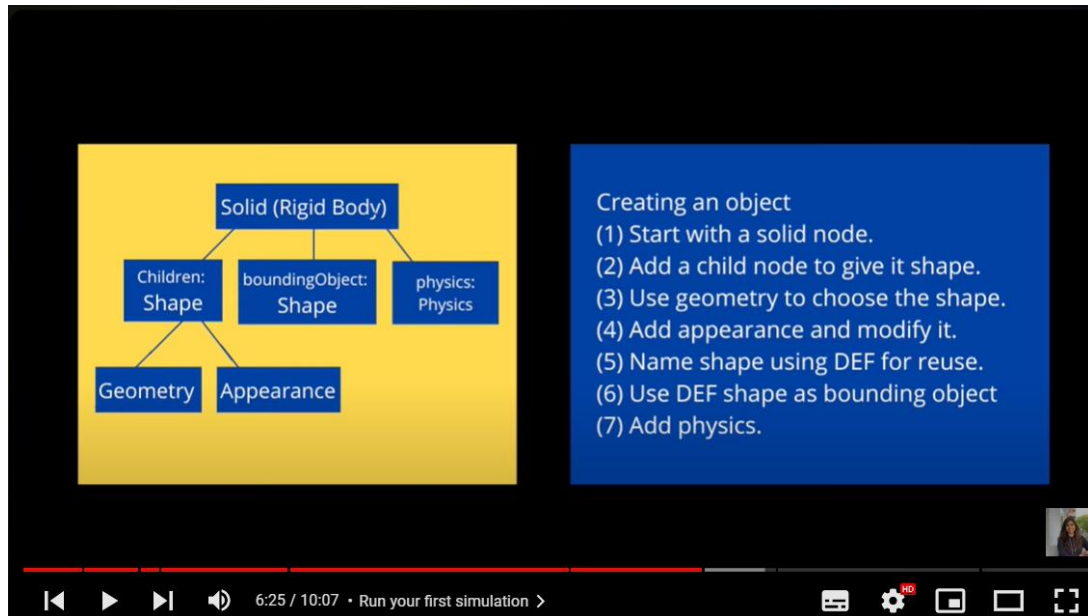
3. Make your own object

Video ini memberikan panduan tentang pembuatan objek kustom sesuai dengan kebutuhan individual. Ini meliputi langkah-langkah dalam menciptakan objek 3D atau menggunakan objek yang sudah ada di Webots, memberikan pengguna keleluasaan dalam merancang simulasi mereka.



4. Understand Fundamental

Tahapan ini memberikan landasan pemahaman tentang konsep-konsep utama dalam Webots. Pengguna diberi gambaran singkat mengenai konsep simulasi fisika, sensor, dan elemen-elemen dasar lainnya yang membentuk kerangka lingkungan simulasi Webots.



The screenshot shows a video player interface. On the left, a yellow box contains a hierarchical diagram of a Solid (Rigid Body) structure. The root node is 'Solid (Rigid Body)', which branches into 'Children: Shape', 'boundingObject: Shape', and 'physics: Physics'. 'Children: Shape' further branches into 'Geometry' and 'Appearance'. On the right, a blue box contains a list of seven steps for creating an object. The video player controls at the bottom show a progress bar at 6:25 / 10:07 and a play button.

Solid (Rigid Body)

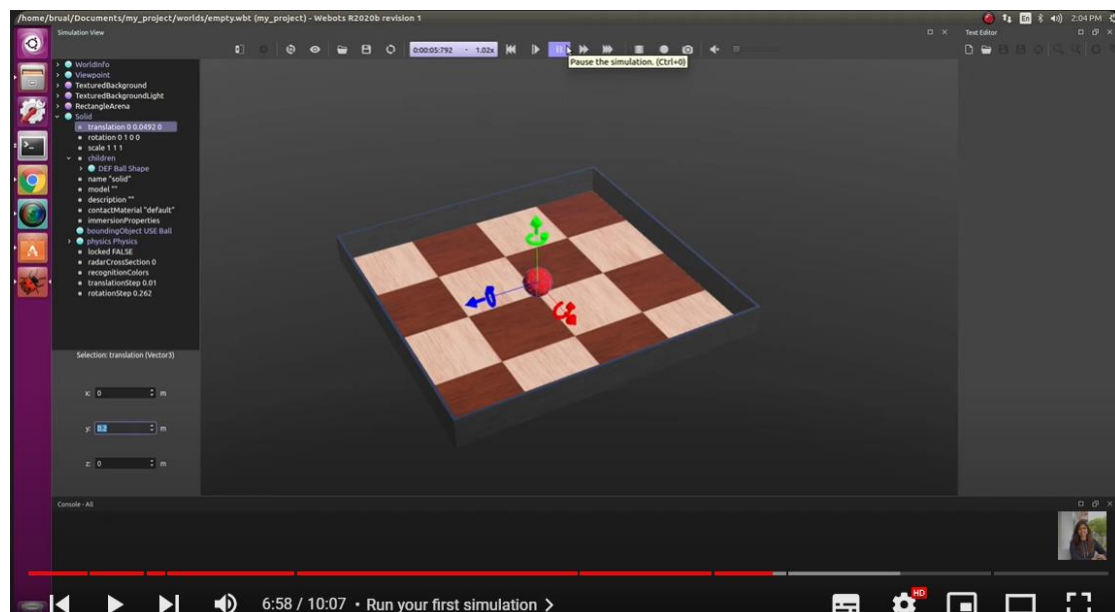
- Children: Shape
 - Geometry
 - Appearance
- boundingObject: Shape
- physics: Physics

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.

5. Run Simulation

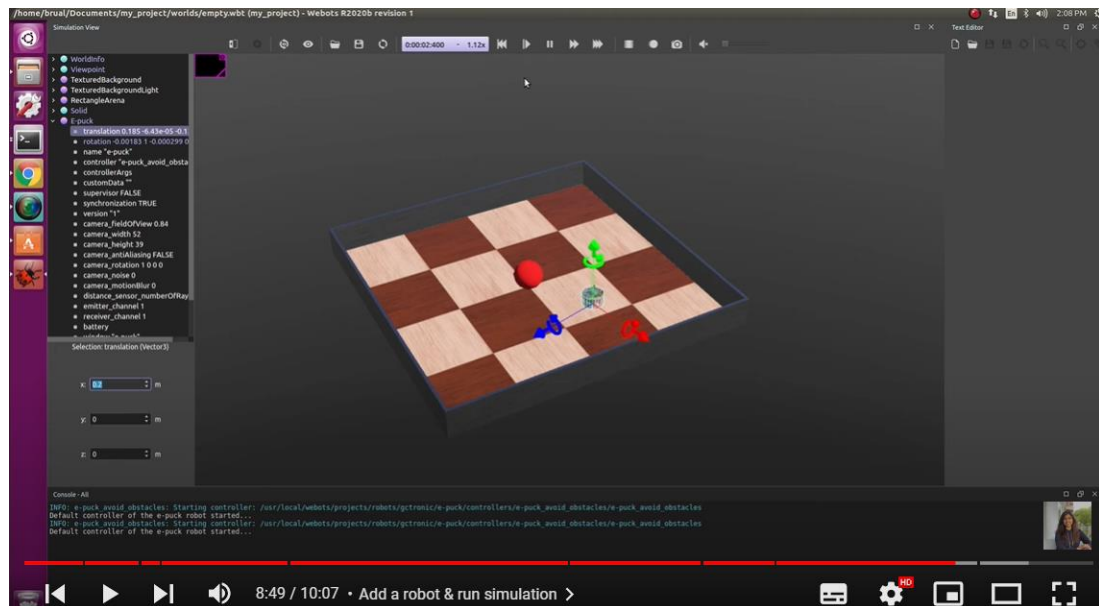
Setelah memperoleh pemahaman dasar, langkah berikutnya adalah menjalankan simulasi awal. Panduan ini mengarahkan pengguna melalui proses tersebut, memastikan bahwa mereka dapat berhasil melihat hasil simulasi mereka dengan sukses.



The screenshot shows the Webots simulation environment. The main window displays a 3D scene with a chessboard on a table. A red ball is on the chessboard, and a small robot is nearby. The left sidebar shows a tree view of the simulation components, including 'WorldInfo', 'Viewpoint', 'TextureBackground', 'TextureBackgroundLight', 'RectangularArea', 'Solid', 'DEF Ball Shape', 'Physics', and 'Robot'. The bottom of the window shows a progress bar at 6:58 / 10:07 and a play button.

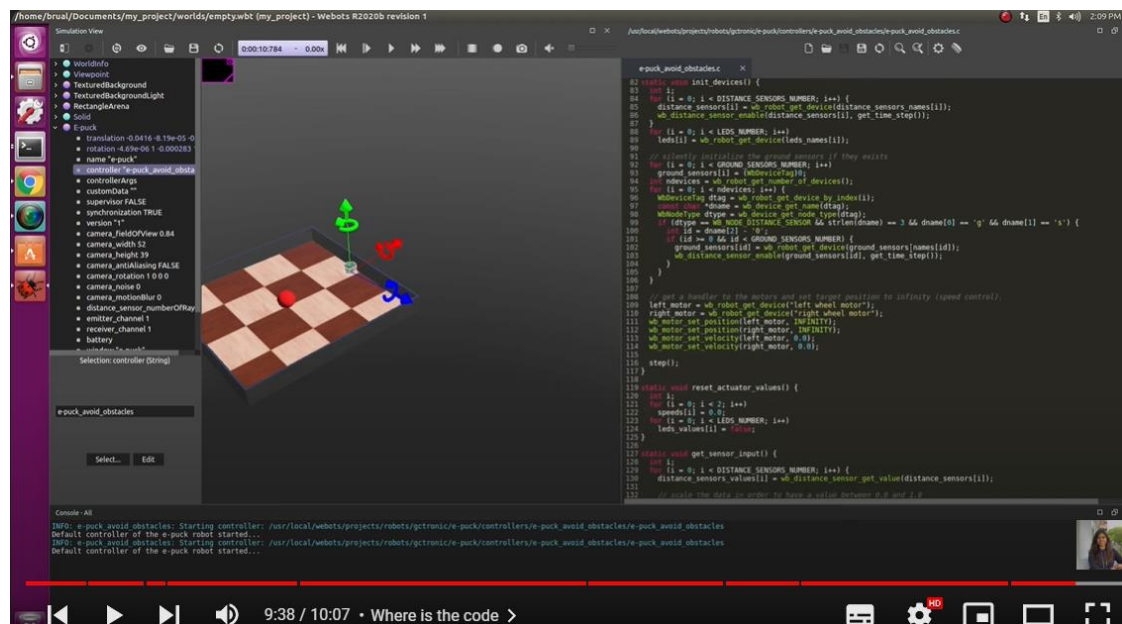
6. Add Robot and Run Simulation

Pada langkah ini, pengguna diarahkan untuk mengintegrasikan robot ke dalam simulasi mereka. Video ini memberikan arahan tentang cara menambahkan robot ke dalam lingkungan simulasi Webots dan menjalankan simulasi dengan kehadiran robot tersebut.

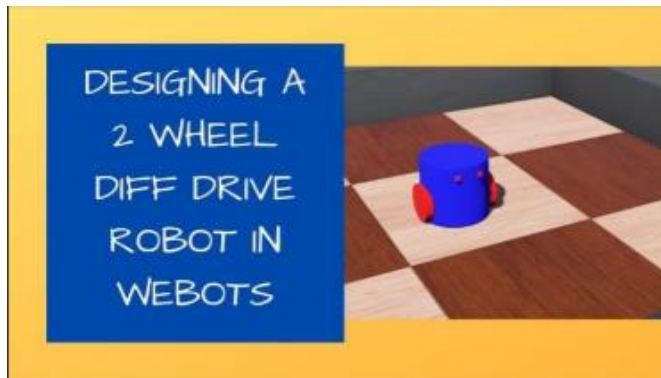


7. The Code

Pada akhirnya, tutorial menguraikan lokasi penyimpanan kode terkait simulasi. Pengguna diberikan pemahaman tentang struktur direktori proyek dan dimana mereka dapat menemukan serta mengelola kode-kode yang terkait dengan simulasi yang telah mereka buat.



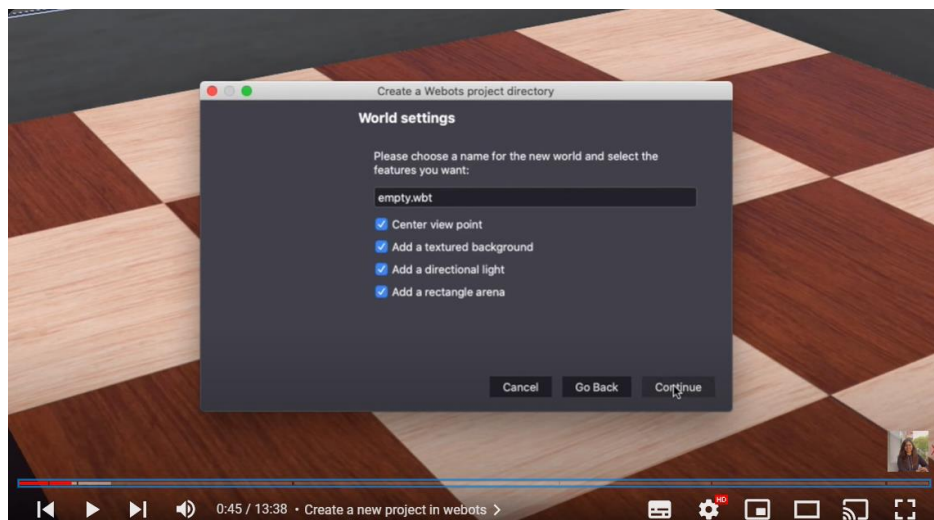
Video 2 : How to design a 2 wheel differential drive robot in Webots? | Webots tutorial 2



https://www.youtube.com/watch?v=ebGJzymXv-o&list=PLbEU0vp_OQkUwANRMUOM00SXybYQ4TXNF&index=2&ab_channel=KajalGada

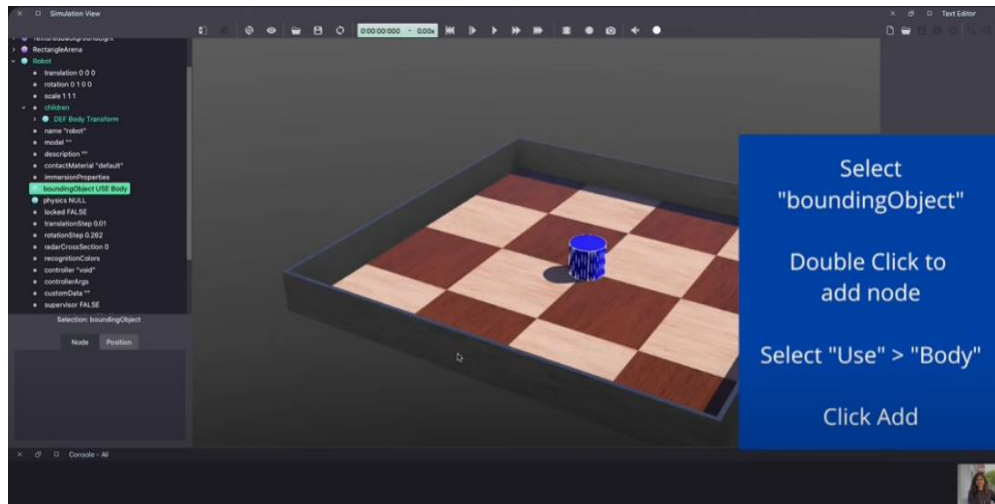
1. Create a new project in webots

Panduan ini diawali dengan langkah pertama dalam menciptakan robot penggerak differential drive di Webots, yakni menciptakan proyek baru. Penjelasan yang terperinci disajikan untuk memastikan pengguna mampu membuat proyek dengan tepat dan siap memasuki tahapan desain robot.



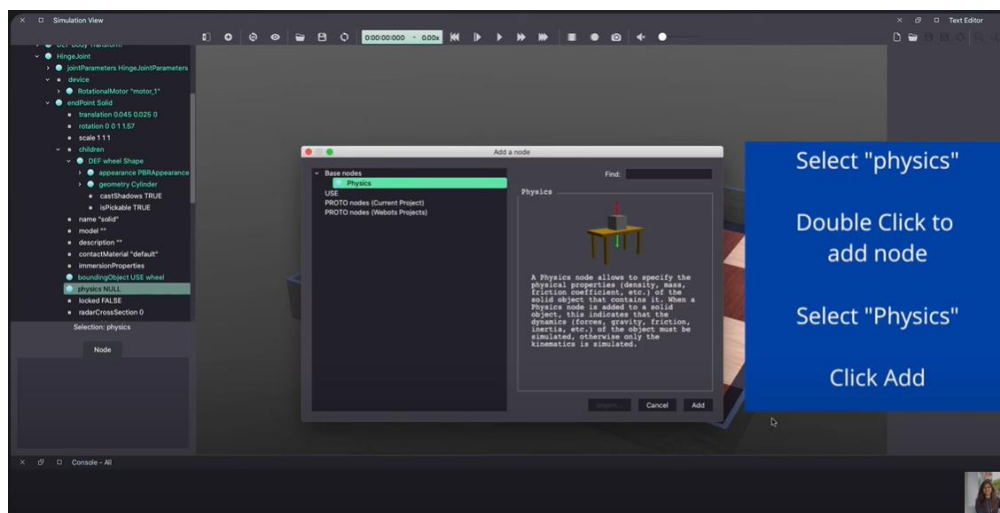
2. Add a robot node and create a robot body

Tahap selanjutnya melibatkan penambahan node robot dan pembentukan struktur tubuh robot. Pengguna didampingi melalui proses menambahkan elemen-elemen dasar yang membentuk kerangka robot, yang memberikan fondasi bagi desain differential drive.



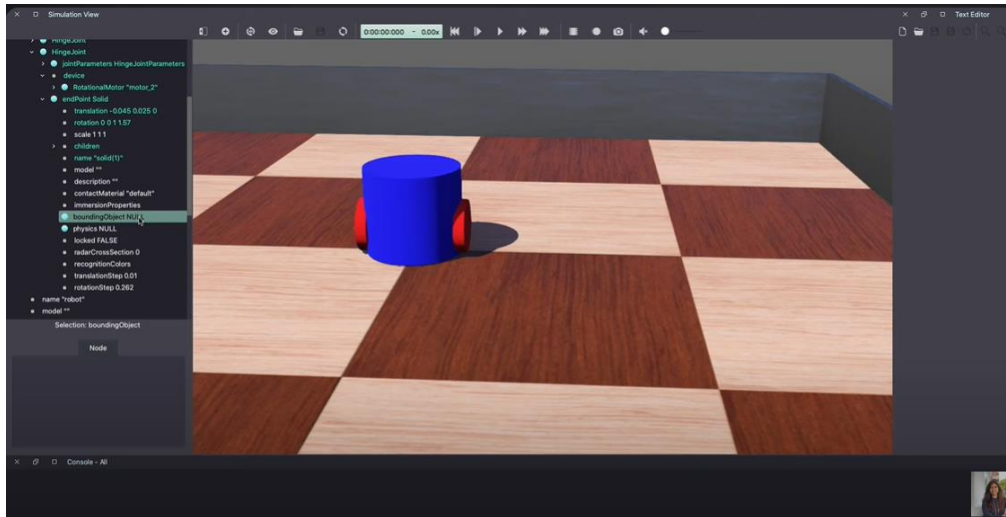
3. Add a hinge joint, rotational motor and wheel to the robot

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



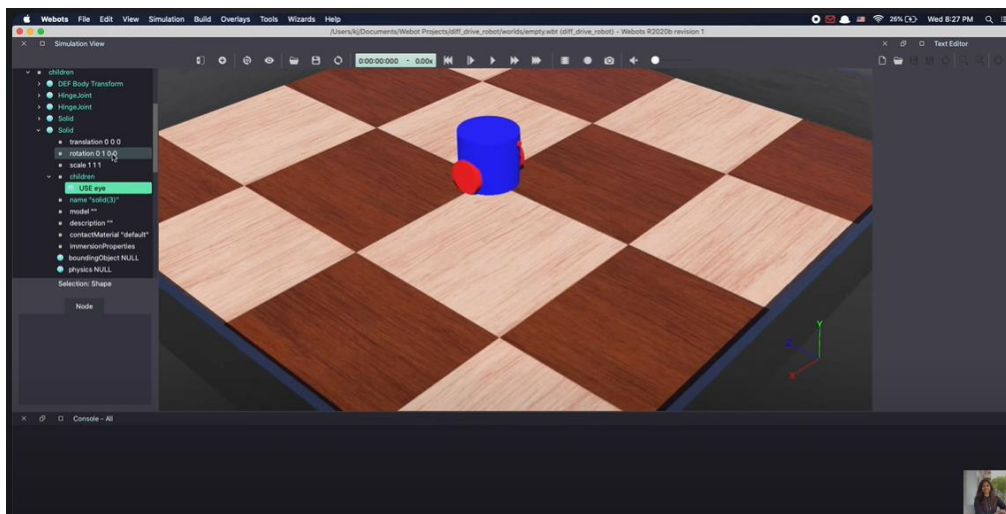
4. Add a second wheel to differential drive robot

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

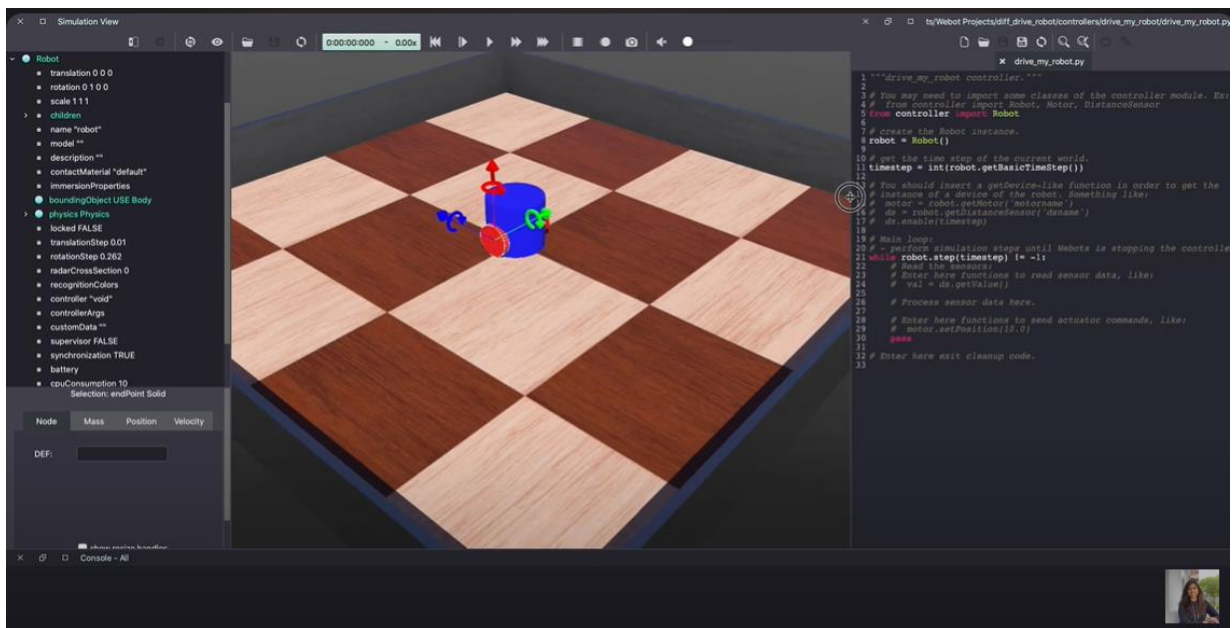
Selain memperhatikan aspek teknis, tutorial juga memberikan pembelajaran tentang menambahkan elemen desain ke robot. Pada langkah ini, pengguna diarahkan untuk menambahkan sepasang mata guna memberikan karakter dan penampilan yang ramah pada robot yang telah dibuat.



Video 3 : Webots tutorial 3: Controller code to drive a differential drive robot

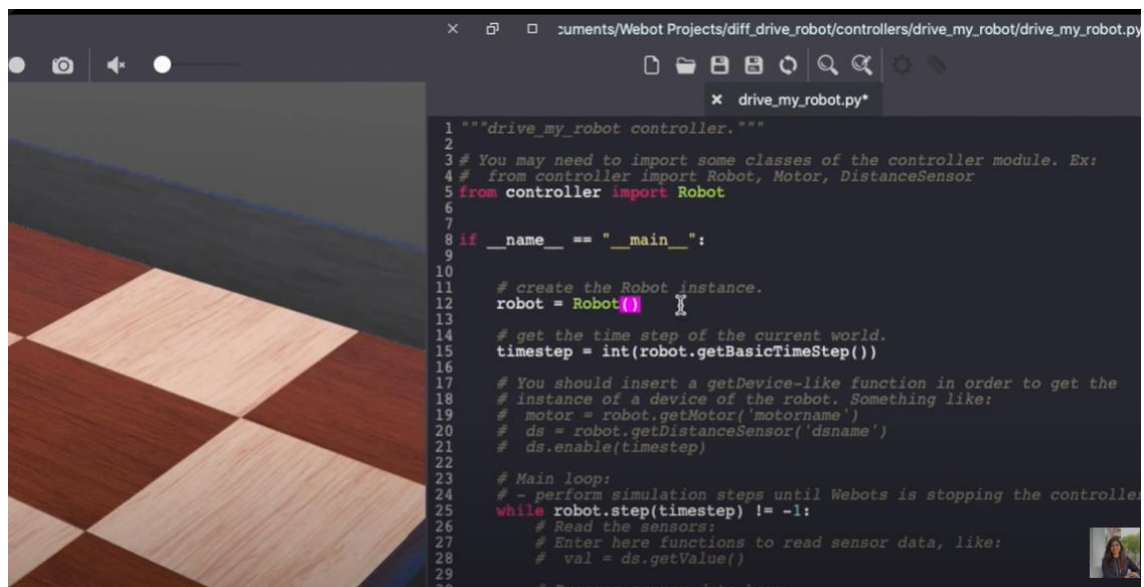
1. Create controller in webots

Pembelajaran dimulai dengan langkah awal, yaitu pembuatan controller di Webots. Tutorial ini memberikan panduan secara detail langkah demi langkah untuk menciptakan kontroler yang diperlukan guna mengendalikan robot penggerak differential drive.



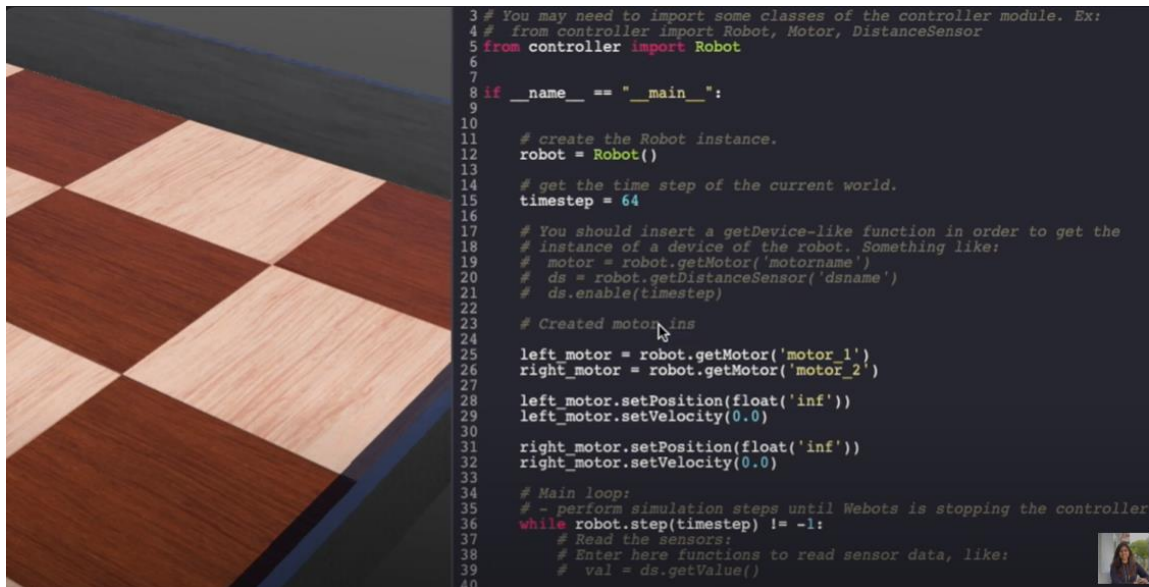
2. Webots controller code in python

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.



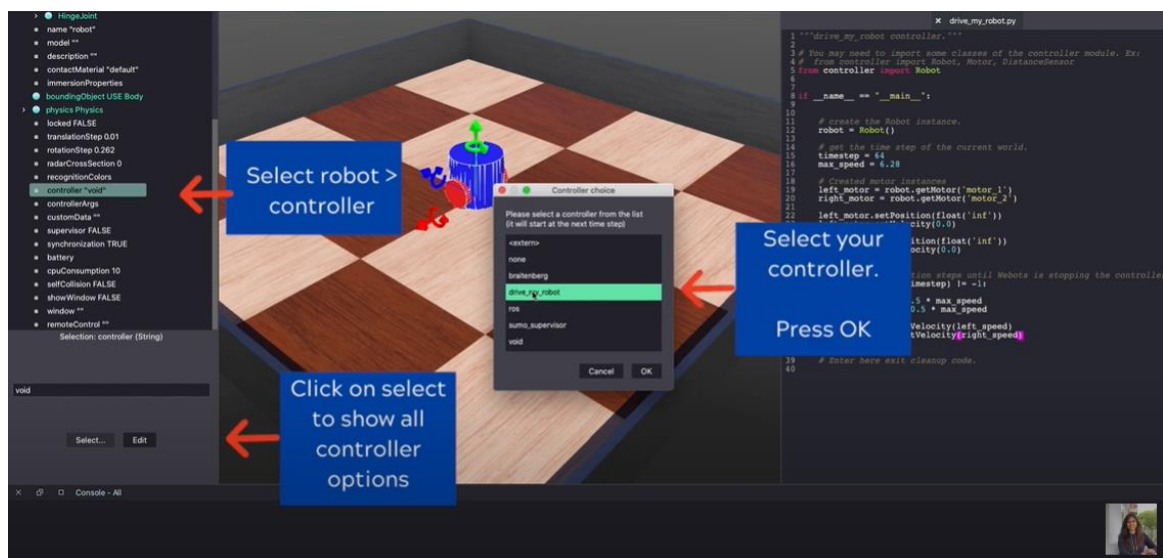
3. Motor Control in webots

Tahapan selanjutnya adalah memahami kontrol motor dalam Webots. Video ini menyajikan penjelasan mendalam tentang pengendalian motor pada robot penggerak differential drive, memastikan pengguna bisa mengimplementasikannya secara akurat.



4. Driving Straight

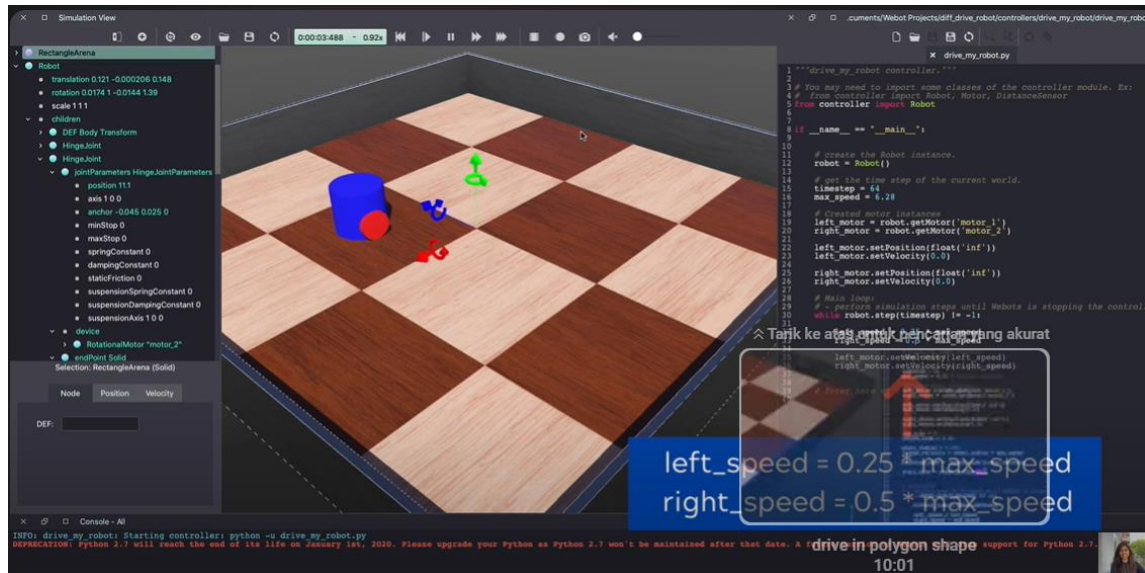
Panduan ini terlibat dalam implementasi kode untuk mengarahkan pergerakan lurus robot. Pengguna dibimbing untuk memahami cara mengontrol kecepatan roda agar robot dapat bergerak lurus dengan sukses dalam simulasi Webots.



5. Turning robot

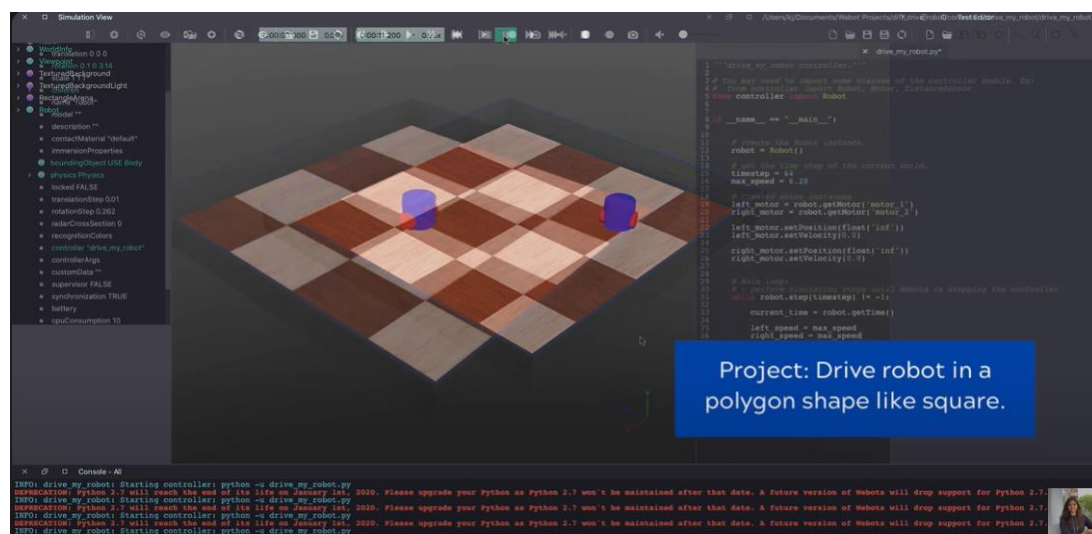
Langkah ini memberikan fokus pada kemampuan mengendalikan perputaran robot.

Video ini memberikan contoh pengkodean untuk memastikan robot dapat berputar sesuai dengan kendali yang diinginkan dalam lingkungan simulasi Webots.



6. Project desain

Bagian ini memberikan tambahan detail tentang proyek yang sedang dikerjakan, mencakup aspek-aspek teknis serta desain yang telah diimplementasikan pada robot penggerak differential drive.



7. Drave in polygon shape

Bagian ini menyajikan tambahan rincian mengenai proyek yang sedang dalam pengembangan, mencakup aspek-aspek teknis serta desain yang telah diterapkan pada robot penggerak differential drive.

```
40 distance_between_wheels = 0.095
41 rate_of_rotation = (2 * linear_velocity)/distance_between_wheels
42 duration_turn = angle_of_rotation/rate_of_rotation
43
44 # 0 < duration_side => drive straight
45 # > duration_side till duration_turn => turn
46
47 # duration_side > and < duration_turn => turn
48
49 rot_start_time = start_time + duration_side
50 rot_end_time = rot_start_time + duration_turn
51
52 # Main loop:
53 # - perform simulation steps until Webots is stopping the controller
54 while robot.step(timestep) != -1:
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_motor.setVelocity(-left_speed)
62         right_motor.setVelocity(right_speed)
63
64
```

Tampilan bioskop (t)

8. Open l dan close loop fiter

Video ini mengulas perbandingan antara sistem open loop dan closed loop dalam konteks pengendalian robot. Pengguna diberikan pemahaman mendalam mengenai perbedaan kunci antara kedua sistem ini serta panduan kapan sebaiknya menggunakan masing-masing dalam pengembangan robotic.

