

Muhammad Fauzan Nur'ilham

1103204085

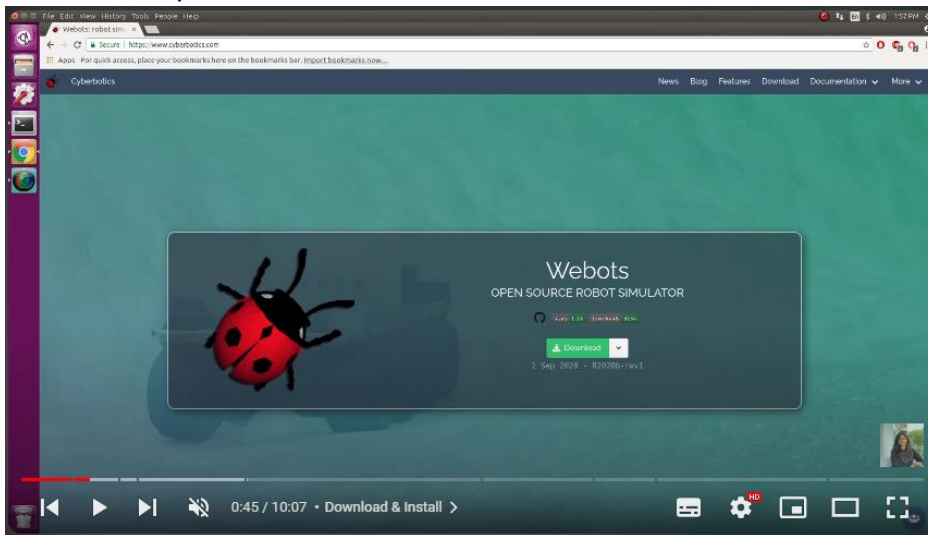
Robotika

TK-44-G7

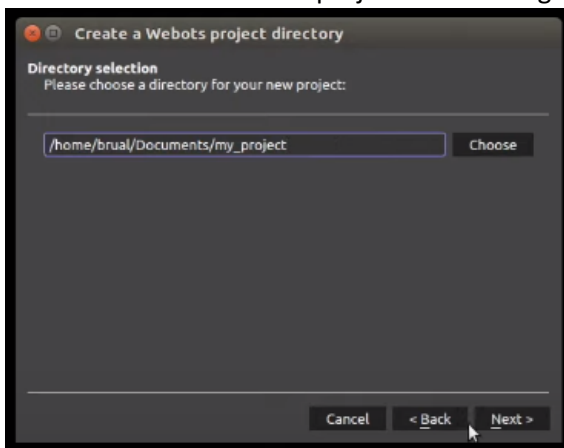
Webots in Python

Video 1

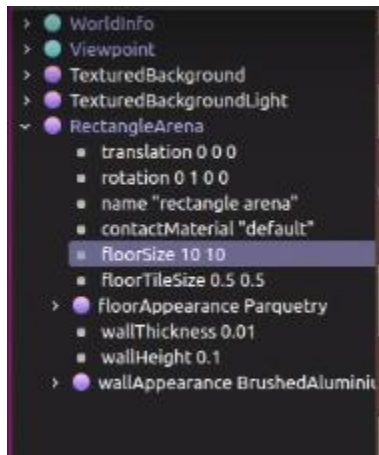
Pada video pertama memberitahu bagaimana cara mendownload dan menginstall webots, masuk kedalam link cyberbotics.com untuk mendownload webots.



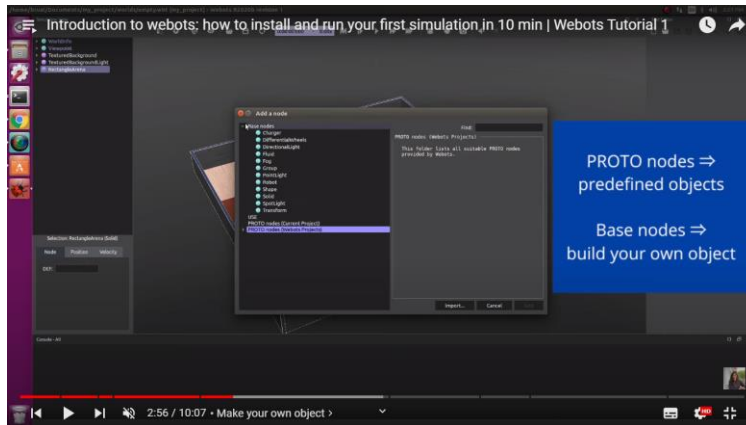
Untuk membuat project baru kita dapat membuatnya dari wizard pada option menu dan menentukan dimana project kita ingin disimpan atau dibuat pada PC kita



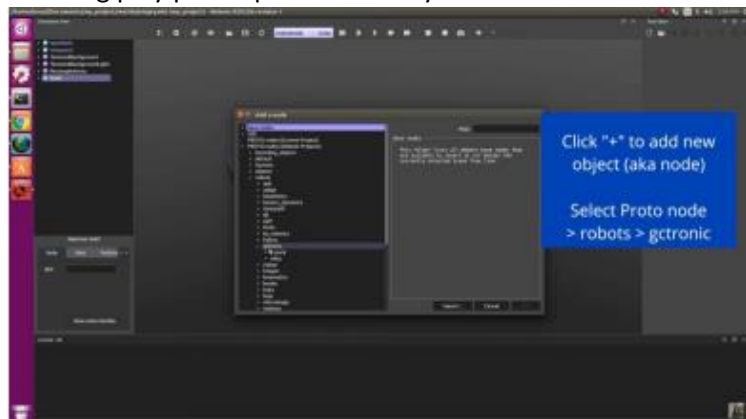
Setelah membuat project. Kita dapat melihat scene tree disebelah kiri layar kita. Scene tree akan berguna seperti node yang dapat mengubah konfigurasi world atau robot yang akan kita gunakan di dalam simulasi.



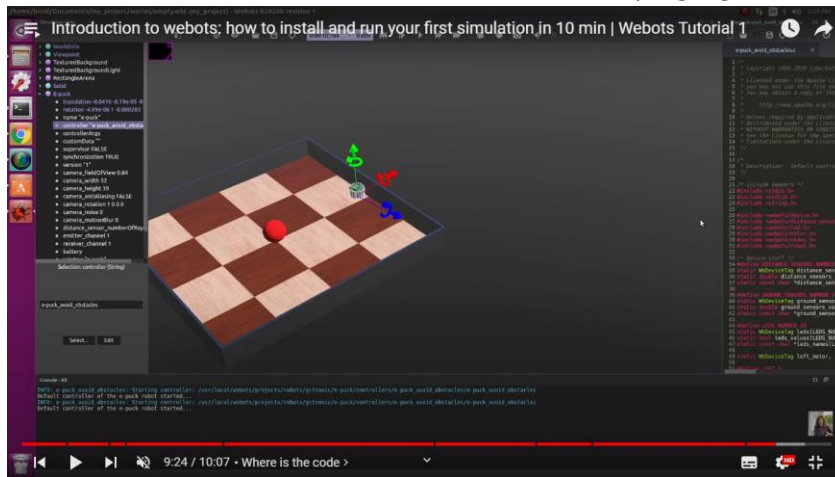
Kita juga dapat memasukkan object dengan menggunakan option add object. Pada dasarnya object juga merupakan sebuah node. Pada webots kita dapat memilih untuk memasukkan object yang telah dibuat atau membuat object kita sendiri yang nantinya object tersebut akan tersedia di scene tree milik kita untuk menentukan bentuk atau konfigurasi object yang ingin kita gunakan



Selain membuat object sendiri kita dapat memasukkan robot yang telah disediakan oleh webots seperti epuck. Dan untuk melakukan simulasi kita dapat menjalankannya dengan menekan lambang play pada option diatas layar

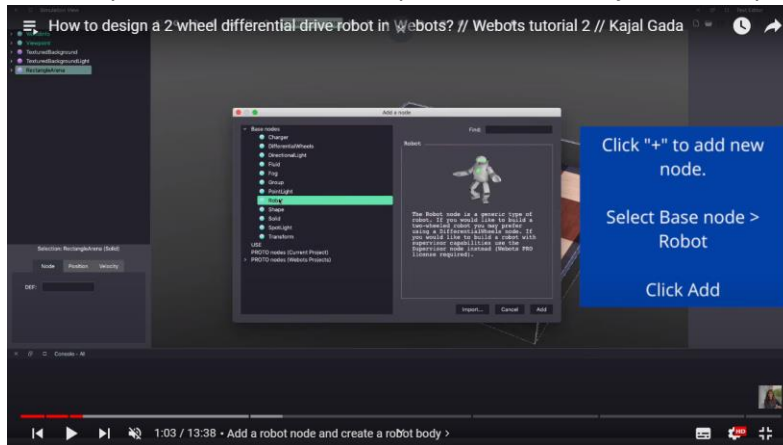


Untuk melakukan koding atau mengubah konfigurasi pada robot. Kita dapat membuka code dari robot tersebut melalui tree scene dan memilih code yang ingin kita ubah

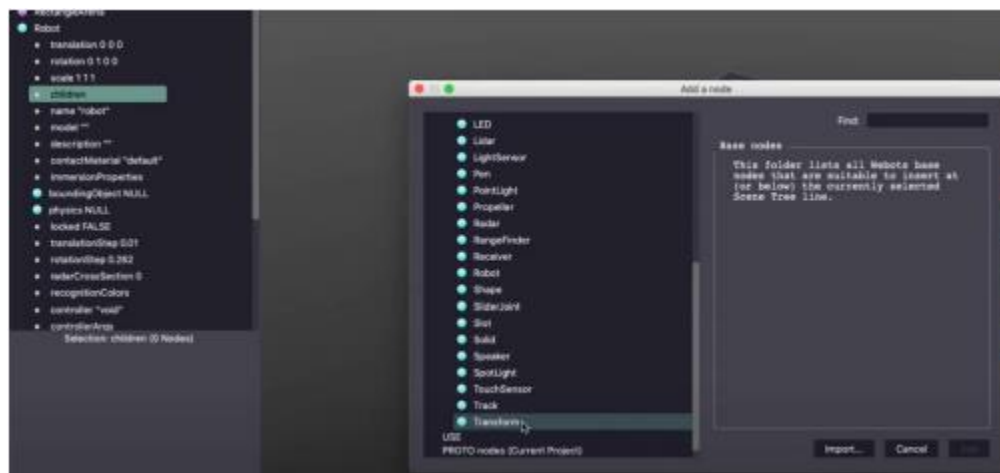


Video 2

Dalam video kedua ini, kita akan diajak untuk memahami secara mendalam bagaimana cara menciptakan sebuah robot yang menggunakan prinsip differential drive dengan dua roda. Proses dimulai dengan membuat proyek baru, di mana langkah pertama adalah mendesain bentuk robot yang akan digunakan. Untuk menginisiasi pembuatan robot, kita dapat memanfaatkan fitur "Tambah" dengan mengeklik ikon tambah (+), selanjutnya memilih node dasar, dan pada tahap berikutnya, kita akan diberikan opsi untuk memilih jenis robot yang akan diimplementasikan.

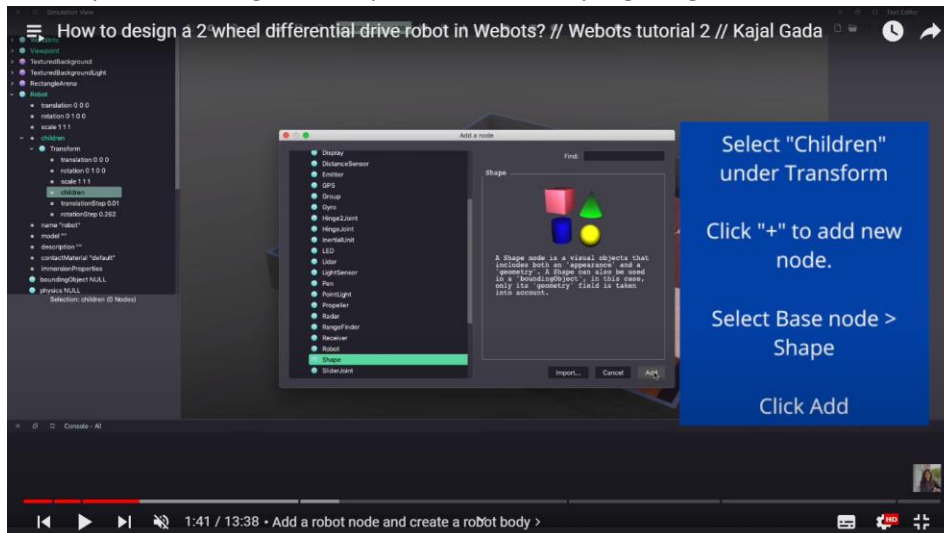


Setelah memilih base node, lanjutkan dengan memilih node children sebagai langkah berikutnya dalam proses pembuatan robot. Untuk memungkinkan robot dapat bergerak, tambahkan node transform ke dalam struktur desain robot. Node transform ini nantinya akan memungkinkan pengaturan pergerakan dan orientasi robot dalam lingkungan simulasi. Dengan langkah ini, kita dapat memastikan bahwa robot yang dibuat memiliki kemampuan untuk bergerak dan berinteraksi sesuai kebutuhan dalam simulasi.

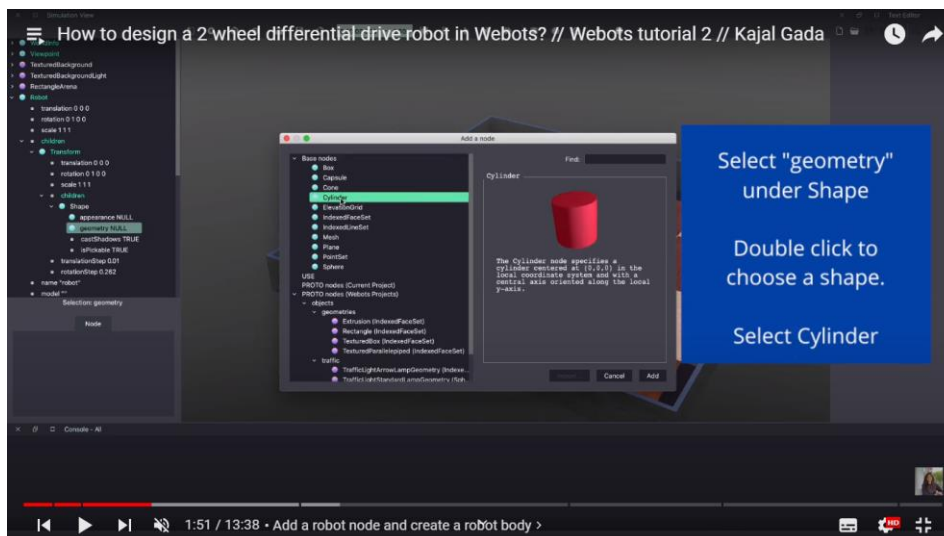


Setelah berhasil menambahkan node transform, selanjutnya adalah membuka node tersebut. Di dalamnya, kita akan menemui node lain yang disebut "children". Buka node "children" tersebut, dan di dalamnya, tambahkan node baru yang dinamakan "shape" untuk membentuk struktur robot yang akan digunakan dalam simulasi.

Dengan menambahkan node "shape", kita memberikan representasi visual pada robot, membangun geometri dan tampilan fisiknya. Proses ini membantu dalam menciptakan gambaran yang lebih nyata dan terlihat dari robot yang sedang kita desain, memastikan bahwa elemen visualnya sesuai dengan konsep dan kebutuhan yang diinginkan.

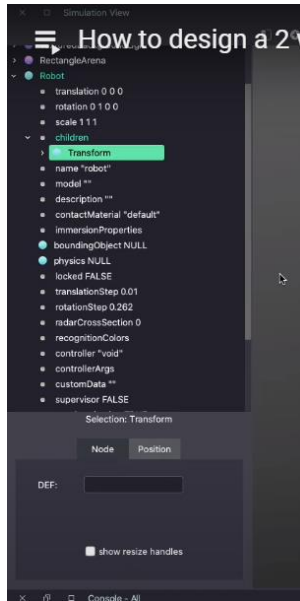


Setelah membuka node "shape", kita dapat melanjutkan dengan memilih node yang disebut "geometry". Di dalam node "geometry", kita dapat menyesuaikan dan memilih bentuk tubuh atau geometri yang diinginkan untuk robot yang sedang kita rancang. Pilihan ini memungkinkan kita untuk mengatur parameter visual seperti ukuran, bentuk, dan detail lainnya yang akan menciptakan representasi yang akurat dan sesuai dengan desain yang diinginkan untuk tubuh robot tersebut dalam simulasi.

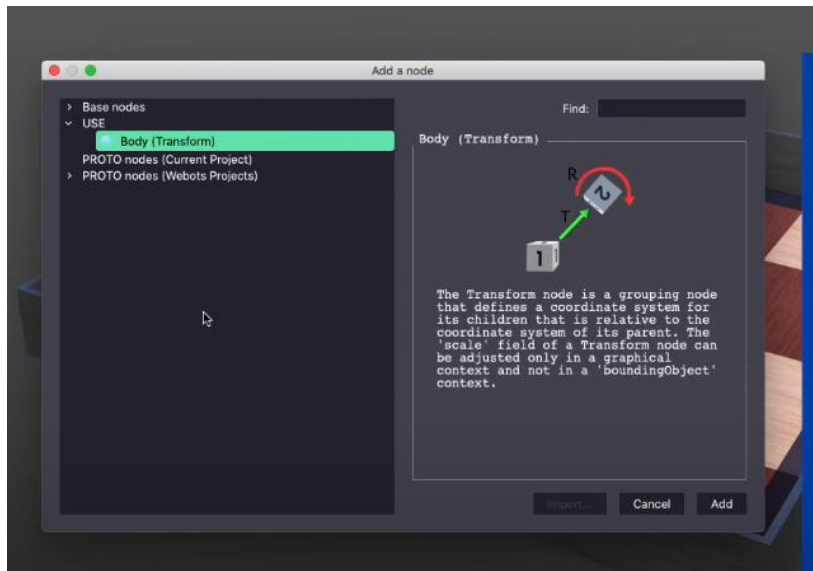


Selanjutnya, di dalam node "transform", berikan nama "body" pada atribut "DEF". Penyebutan DEF (Definition) memberikan pengenal atau label unik pada node tersebut, sehingga nantinya

dapat dengan mudah diakses dan digunakan kembali dalam proses desain atau pengaturan lainnya. Memberikan nama yang bermakna seperti "body" dapat membantu memudahkan identifikasi dan pengelolaan elemen robot yang terlibat dalam proses desain dan simulasi.

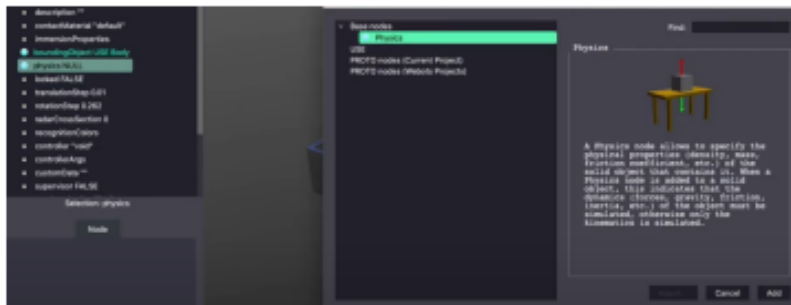


Selanjutnya, pada node "boundingobject", tambahkan node "body" ke dalamnya. Langkah ini akan mengaitkan objek pembatas (bounding object) dengan elemen "body" yang telah dibuat sebelumnya. Dengan menghubungkan node-body ke dalam node bounding object, kita dapat menentukan batas atau area yang dikenakan pada elemen "body". Hal ini akan memengaruhi respons fisika dan interaksi dengan lingkungan sekitar selama simulasi robot bergerak. Proses ini menjadi langkah penting dalam membangun model robot yang akurat dan dapat diandalkan dalam simulasi.

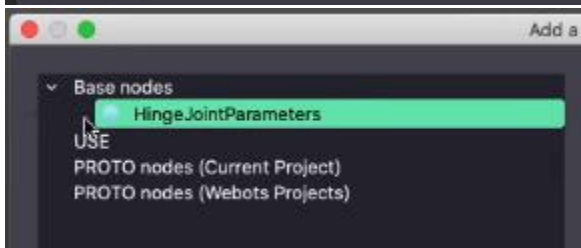
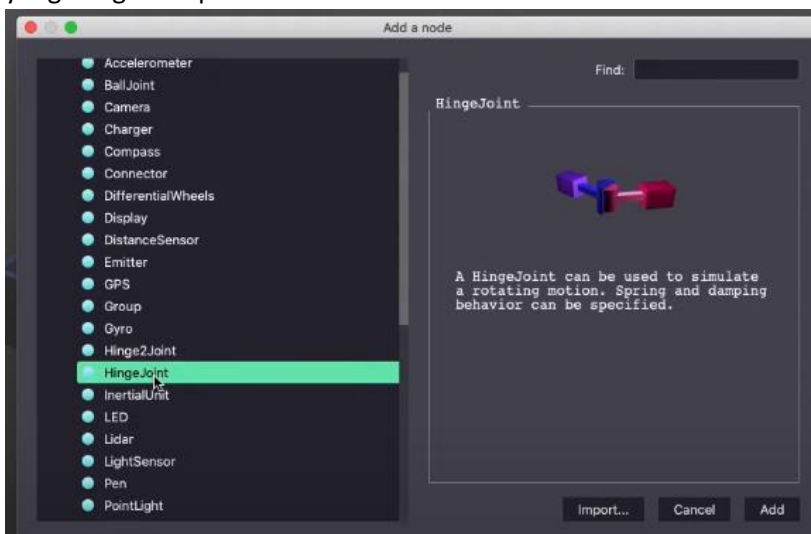


Setelah menambahkan node "body", langkah selanjutnya adalah menambahkan node "physics" pada model robot. Node "physics" akan memberikan atribut fisika pada robot, seperti massa, friksi, atau atribut lainnya yang memengaruhi perilaku robot dalam simulasi. Dengan menetapkan properti fisika yang sesuai, kita dapat mensimulasikan gerakan robot sesuai dengan prinsip-

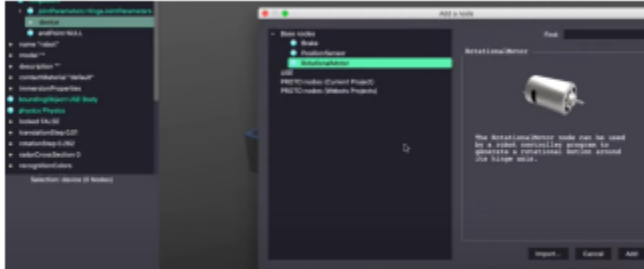
prinsip fisika yang berlaku di dunia nyata. Ini mencakup respons terhadap gaya, percepatan, dan perubahan kecepatan yang dapat diamati selama simulasi. Dengan adanya node "physics", kita dapat lebih mendekati kondisi fisik yang sebenarnya dalam lingkungan simulasi.



Setelah kembali ke node "body transform", selanjutnya kita akan menghubungkan robot dengan roda menggunakan node "hingejoint". Node "hingejoint" memungkinkan kita membuat sambungan sendi atau engsel yang memungkinkan gerakan berputar. Pada node "hingejoint parameter", kita dapat menentukan parameter seperti batas pergerakan, kekakuan, atau atribut lainnya yang memengaruhi hubungan antara tubuh robot dan roda. Dengan cara ini, kita dapat merancang bagaimana roda akan berinteraksi dengan tubuh robot dan memberikan sifat gerak yang diinginkan pada model simulasi kita.



Selanjutnya, pada bagian "Device", kita akan menambahkan "rotational motor" dan memberinya nama "motor 1". Rotational motor akan berfungsi sebagai penggerak roda atau perputaran pada robot. Dengan memberikan nama "motor 1", kita dapat dengan mudah mengidentifikasi dan mengelola motor ini di dalam lingkungan simulasi. Penggunaan motor ini akan memungkinkan kita untuk mengendalikan pergerakan dan putaran roda pada robot.



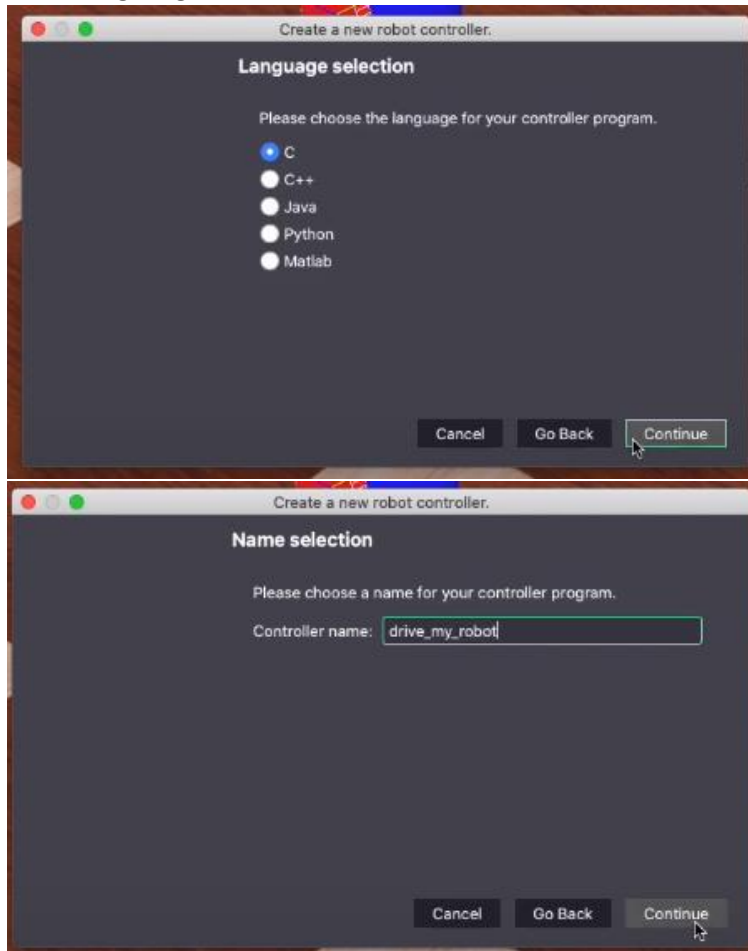
Selanjutnya, pada bagian hingejoint, pilih "endpoint" dan tambahkan node "solid" dari base node. Dengan menambahkan solid pada endpoint hingejoint, kita dapat membuat suatu titik atau bagian yang tetap untuk dijadikan dasar roda robot. Setelah menambahkan solid, akan muncul node dengan nama "children" di dalam "endpoint solid". Selanjutnya, masukkan node "shape" pada node tersebut. Di dalam node shape, kita akan menambahkan node "children" lagi, dan di dalam node ini, kita akan mencoba membuat representasi dari roda sebagai penggerak pada robot kita.

Setelah itu, lakukan langkah yang sama untuk membuat roda kedua seperti yang telah dijelaskan sebelumnya. Dengan cara ini, kita dapat membuat dua roda yang terhubung ke robot dan siap digunakan sebagai penggerak. Proses ini memastikan bahwa robot yang dibuat memiliki dua roda differential drive, yang merupakan karakteristik umum pada robot gerak dua roda.



Video 3

Setelah menyelesaikan pembuatan robot dengan 2 wheels differential drive, kita sekarang akan mempelajari cara menulis kode yang akan mengendalikan robot tersebut. Pertama, kita akan membuat controller robot baru menggunakan opsi wizard yang tersedia di bagian atas layar. Pilih bahasa pemrograman yang akan digunakan, dan berikan nama untuk controller tersebut. Langkah ini merupakan awal dari proses penulisan kode yang akan memberikan instruksi kepada robot dalam lingkungan simulasi.



Setelah berhasil membuat controller untuk robot, sebuah jendela untuk menulis kode controller akan muncul di layar. Selanjutnya, masukkan kode seperti di bawah ini untuk menjalankan modul kita sebagai kode utama:

```
# from controller import Robot, Motor, DistanceSensor
from controller import Robot

if __name__ == "__main__":

    # create the Robot instance.
    robot = Robot()

    # get the time step of the current world.
    timestep = 64
```

Berikut adalah contoh kode untuk mengendalikan motor pada robot kita:

```
# Created motor instances
left_motor = robot.getMotor('motor_1')
right_motor = robot.getMotor('motor_2')

left_motor.setPosition(float('inf'))
left_motor.setVelocity(0.0)

right_motor.setPosition(float('inf'))
right_motor.setVelocity(0.0)
```

Kemudian tambahkan kode untuk menentukan apa yang akan dilakukan oleh robot di dalam simulasi dan untuk mengatur kecepatan maksimal:

```
# - perform simulation steps until Webots is stoppi
while robot.step(timestep) != -1:
    left_speed = 0.5 * max_speed
    right_speed = 0.5 * max_speed

    left_motor.setVelocity(left_speed)
    right_motor.setVelocity(right_speed)

# get the time step of the current world.
timestep = 64
max_speed = 6.28
```

untuk membuat robot dapat berbelok atau berputar, Anda dapat mengatur kecepatan roda kiri dan roda kanan dengan nilai yang berbeda. Berikut adalah contoh modifikasi pada kode untuk membuat robot berputar:

```
left_speed = 0.5 * max_speed
right_speed = 0.25 * max_speed
```