

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**Tương tác người – Robot
BÁO CÁO THỰC HÀNH BUỔI 1
LAB01 - Làm quen với Webots**

Sinh viên thực hiện
Nguyễn Huy Thắng – 22027545
Mai Quốc Hiếu – 22027521
Lê Công Phú – 22027524

Hà Nội, ngày 22 tháng 9 năm 2025

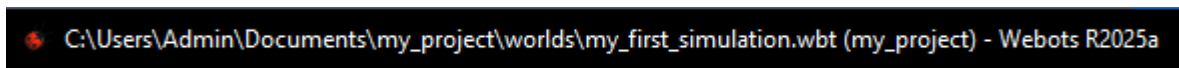
Tutorial 1 : Mô phỏng đầu tiên trong phần mềm WEBOTS

Mục tiêu của hướng dẫn đầu tiên này là giúp ta làm quen với giao diện người dùng và các khái niệm cơ bản về Webots. Ta sẽ tạo mô phỏng đầu tiên của mình bao gồm một môi trường đơn giản: một đấu trường với sàn và tường, một vài hộp, một robot e-puck và một chương trình điều khiển giúp robot di chuyển.

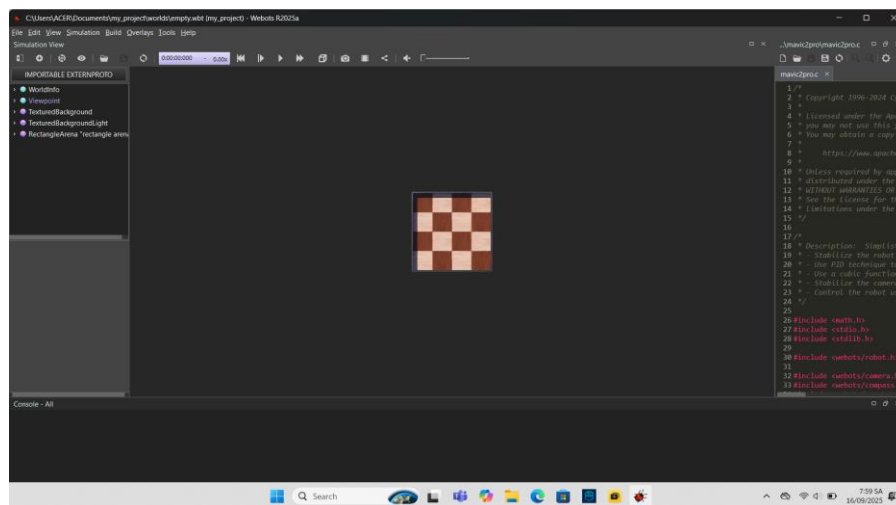
Hands-on 1: Khởi động Webots

Hands-on 2:

1. Đặt tên thư mục dự án **my_first_simulation** thay vì tên được đề xuất là **my_project**.
2. Đặt tên tệp thế giới **my_first_simulation.wbt** thay vì tên được đề xuất là **empty.wbt**.
3. Nhấp vào tất cả các ô đánh dấu, bao gồm cả ô "Thêm đấu trường hình chữ nhật" không được đánh dấu theo mặc định.



Đường dẫn của thư mục theo tên đã đặt



Giao diện của WEBOTS

Bên trái của giao diện sẽ gồm các node của WEBOTS được xây dựng dựa trên cấu trúc **Scene Tree**. Chế độ **Scene Tree** là nơi có thể chỉnh sửa các nút và trường. Hiện tại, nó sẽ liệt kê các nút sau:

- **WorldInfo**: chứa các tham số toàn cục của mô phỏng.
- **Viewpoint**: xác định các thông số chính của camera góc nhìn.
- **TexturedBackground**: xác định nền của cảnh (Ta sẽ thấy những ngọn núi ở xa nếu bạn xoay một chút góc nhìn)
- **TexturedBackgroundLight**: xác định ánh sáng liên quan đến nền ở trên.

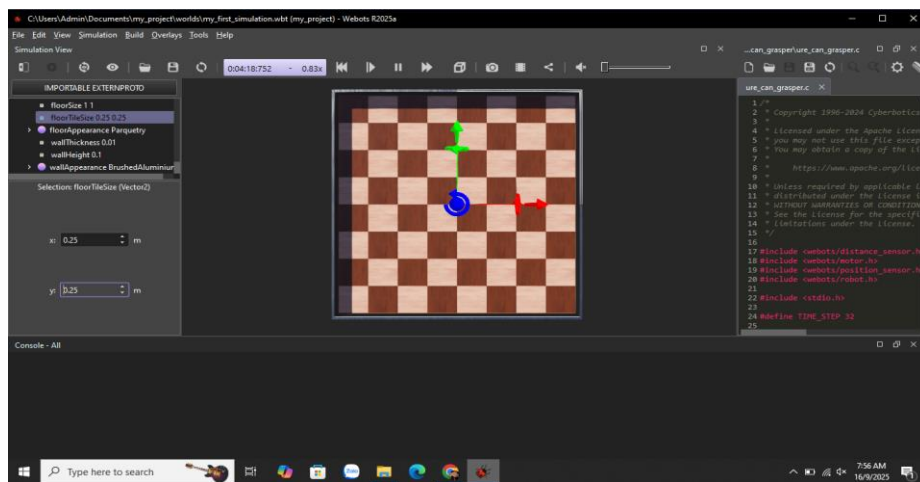
- **RectangleArena**: xác định đối tượng duy nhất mà bạn nhìn thấy cho đến thời điểm này trong cảnh này.

Mỗi nút có một số thuộc tính có thể tùy chỉnh được **Fields**, sửa đổi các trường này để thay đổi đầu trường hình chữ nhật:

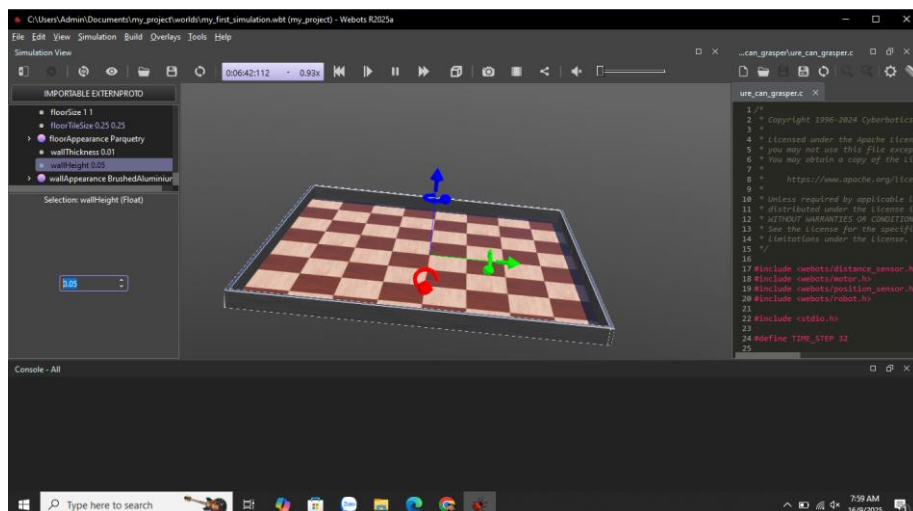
Hands-on 3:

Nhấp đúp vào **RectangleArena** node trong cây cảnh. Thao tác này sẽ mở nút và hiển thị các trường của nút.

1. Chọn **floorTileSize** trường và đặt giá trị của nó thành 0.25 0.25. 0.5 0.5 sẽ thấy hiệu ứng trong chế độ xem 3D.
2. Chọn **wallHeight** trường và thay đổi giá trị của nó 0.05 thành 0.1. Bức tường của đầu trường bây giờ sẽ thấp hơn.



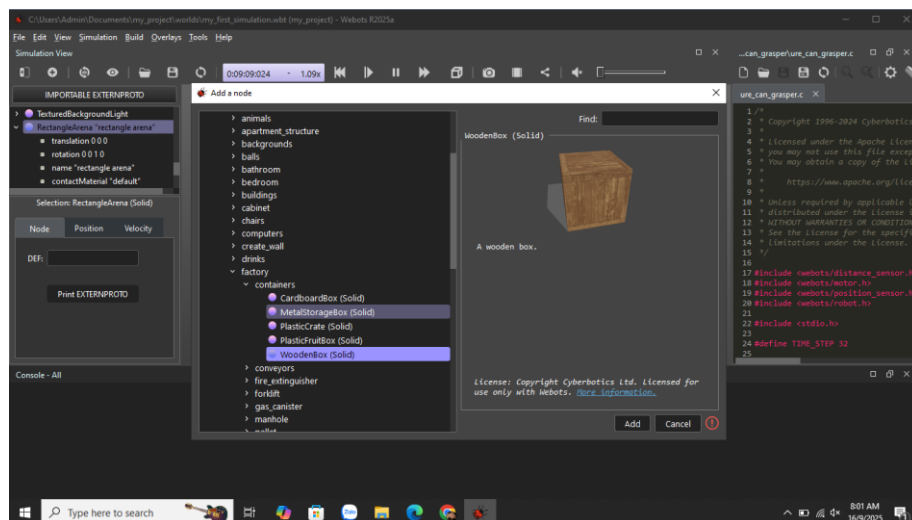
Hình ảnh kích thước ô sàn thay đổi



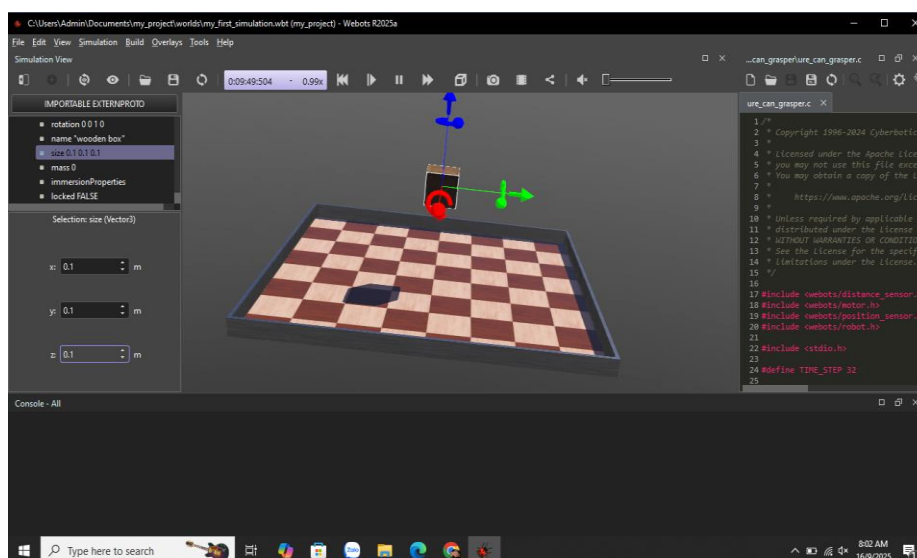
Hình ảnh chiều cao tường xung quanh thay đổi

Hands-on 4: Thêm 1 node a **wooden box** vào trong môi trường

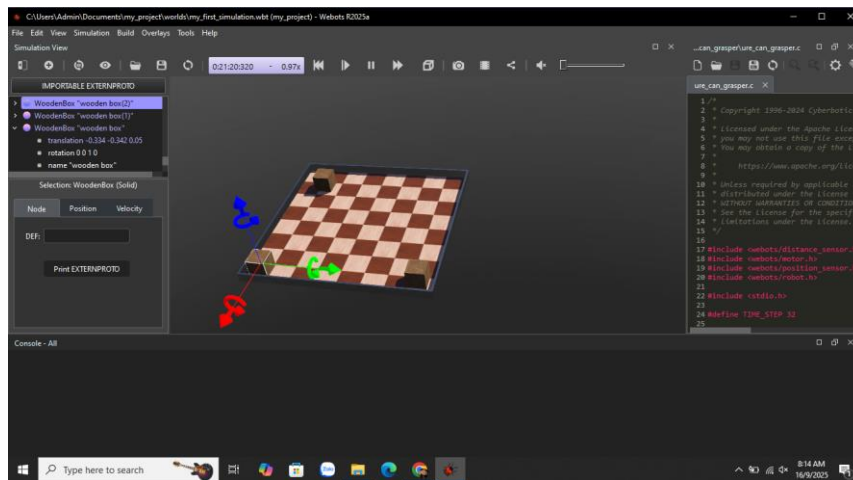
1. Chọn **RectangularArena**: Trong **Scene Tree** double click vào node **Rectangle Arena**.
2. Với **RectangularArena** đang được chọn, nhấn nút **Add**, chọn thư mục **PROTO nodes (Webots Projects)** → **objects** → **factory** → **containers** → **WoodenBox (Solid)**.
3. Chỉnh kích thước và vị trí hộp: trong **Scene Tree** double click vào **WoodenBox** để mở ra các trường (fields). Tìm **field size** và **field translation** để thay đổi kích thước hộp và vị trí của hộp so với sàn.
4. Di chuyển hộp: giữ Shift + chuột trái và kéo hộp để thay đổi vị trí.
5. Copy thành nhiều hộp: Chọn hộp trong **Scene Tree** → nhấn **Ctrl + C**, **Ctrl + V** → Hộp mới xuất hiện ở vị trí hộp cũ
6. Xoay hộp: trong 3D view Shift + kéo chuột phải để xoay theo trục đứng



Add thêm node wooden box vào scene tree



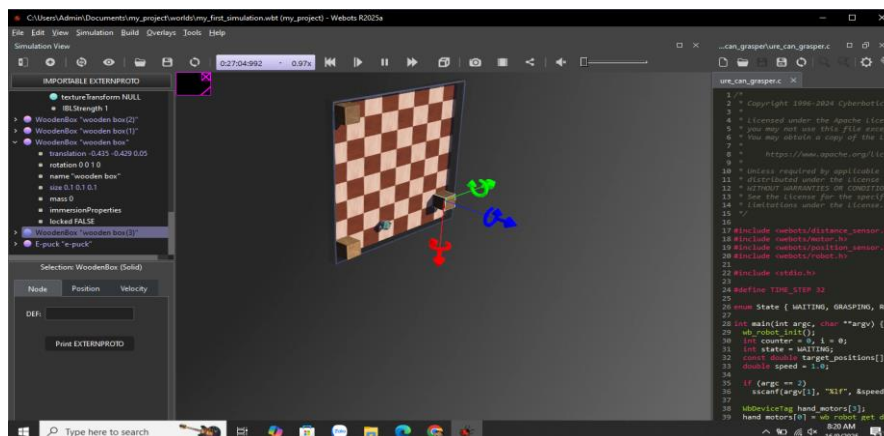
Điều chỉnh kích thước trục x,y,z khối hộp



Tạo thêm hộp mới và di chuyển đến điểm mong muốn

Hands-on 5: Thêm một Robot e-puck

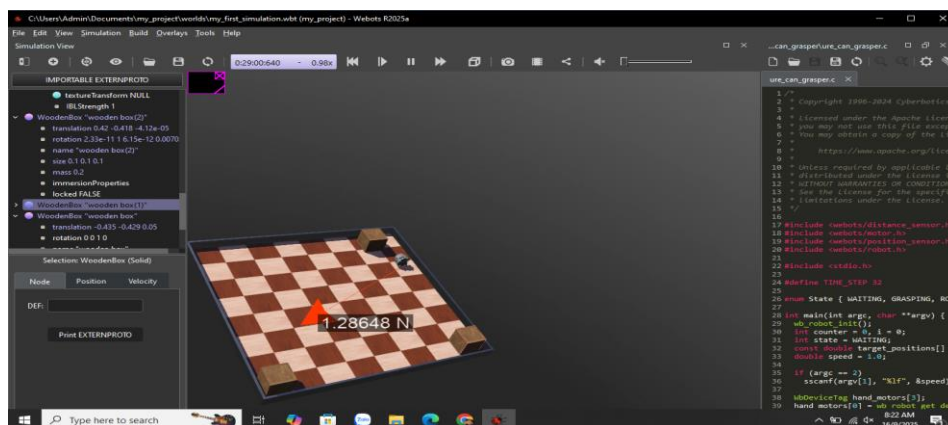
1. Chọn vị trí thêm robot: chọn **RectangleArena**
2. Nhấn nút **Add: PROTO nodes (Webots Projects)** → **robots** → **gctronic** → **e-puck** → **E-puck (Robot)**



Robot E-pucks xuất hiện trong môi trường

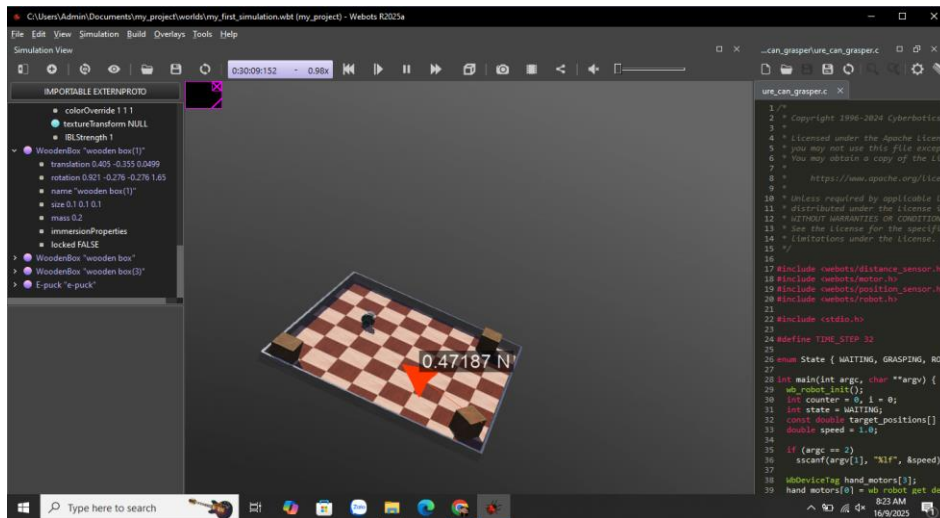
Hands-on 6: Tác dụng 1 lực lên robot

- Nhấn giữ Shift, nhấp chuột trái và kéo robot để thêm lực tác dụng lên robot



Lực tác dụng lên robot và có thể ảnh hưởng đến khả năng di chuyển của robot

Tương tự thêm lực tác dụng lên hộp

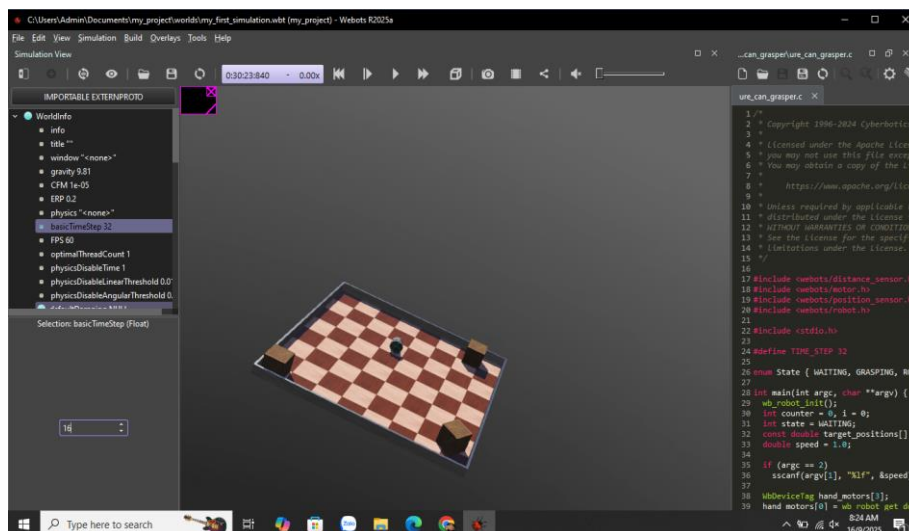


Lực tác dụng lên các khối gỗ và kéo nó di chuyển khỏi vị trí ban đầu

→ WEBOTS hỗ trợ mô phỏng các tương tác vật

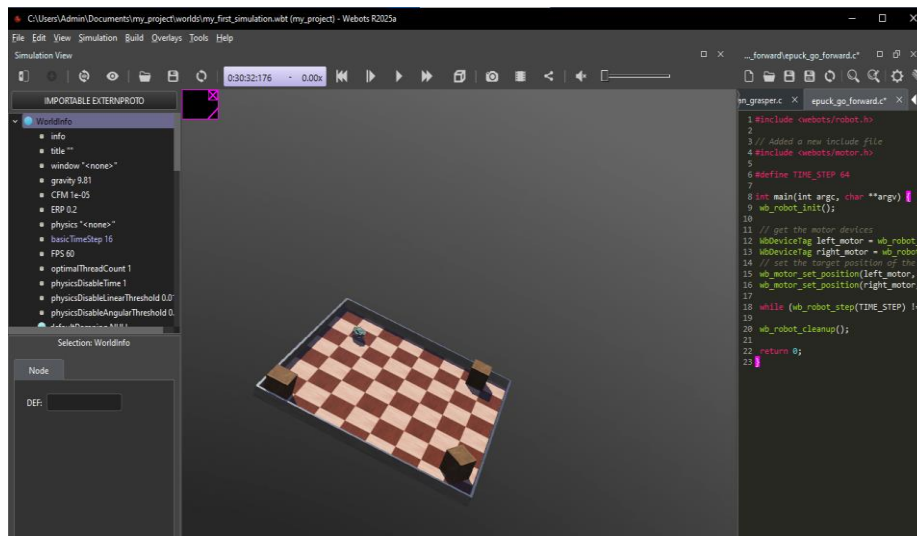
Hands-on 7:

1. Dừng mô phỏng và đưa về trạng thái ban đầu
 - Nhấn **Pause** trên thanh công cụ để dừng mô phỏng
 - Nhấn **Revert** để đưa thế giới về trạng thái ban đầu
2. Chỉnh **basicTimestep**
 - Mở rộng node **WorldInfo**
 - Tìm field **basicTimestep** đổi giá trị thành 16 giúp mô phỏng vật lý mượt và chính xác hơn



*Điều chỉnh **basicTimestep** giúp tăng độ chính xác mô phỏng*

Hands-on 8,9: Tạo bộ điều khiển

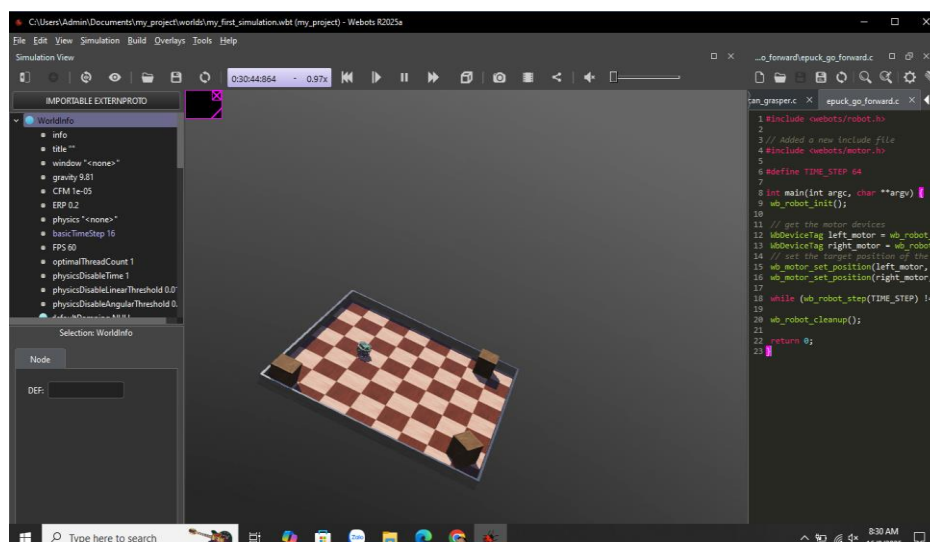


Tạo controller để điều khiển Robot

Trong đó:

- **<webots/robot.h>**: Cung cấp các hàm cơ bản để khởi tạo và quản lý robot trong Webots (như **wb_robot_init()**, **wb_robot_step()**, **wb_robot_cleanup()**).
- **<webots/motor.h>**: Cung cấp các hàm để điều khiển động cơ, như **wb_robot_get_device()** và **wb_motor_set_position()**.
- **wb_robot_get_device()** lấy tham chiếu đến các thiết bị động cơ dựa trên tên ("left wheel motor" và "right wheel motor")
- **wb_motor_set_position()** đặt vị trí mục tiêu cho động cơ. Robot sẽ di chuyển với tốc độ tối đa trong một lúc và dừng lại khi bánh xe quay được 10 radian

Hands-on 10: Mở rộng bộ điều khiển để kiểm soát tốc độ



E-pucks thực hiện đi thẳng theo code

KẾT LUẬN:

Qua những phần thực hành trên giúp chúng ta nắm được những khái niệm :

- Một thế giới được tạo thành từ các nút được sắp xếp theo cấu trúc cây.
- Một thế giới được lưu trong một **.wbt** tệp được lưu trữ trong dự án Webots.
- Dự án cũng bao gồm các chương trình điều khiển robot xác định hành vi của robot.
- Bộ điều khiển có thể được viết bằng C hoặc ngôn ngữ khác.
- Bộ điều khiển C, C++ và Java phải được biên dịch rõ ràng trước khi có thể thực thi.
- Bộ điều khiển được liên kết với robot thông qua **controller** các trường của **Robot** node.

Video kết quả

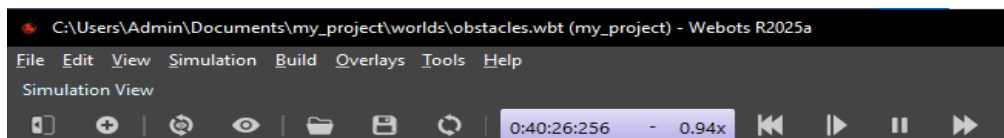
[Tutorial 1.mp4](#)

➔ Video kết quả có môi trường giống với môi trường trong Tutorial 1 và e-puck đã thực hiện chạy tránh vật cản và sau đó chạy thẳng như code.

TUTORIAL 2: Thay đổi môi trường

Trong hướng dẫn này, chúng tôi sẽ hướng dẫn bạn cách tạo các vật thể đơn giản trong môi trường. Bước đầu tiên là tạo một quả bóng tương tác với môi trường. Chúng ta sẽ tìm hiểu một số khái niệm liên quan đến các nút: ý nghĩa của chúng, cách tạo ra chúng, cách chúng phải được liên kết, v.v. Ngoài ra, chúng ta sẽ xem cách thiết lập vật lý.

Hands-on 1:

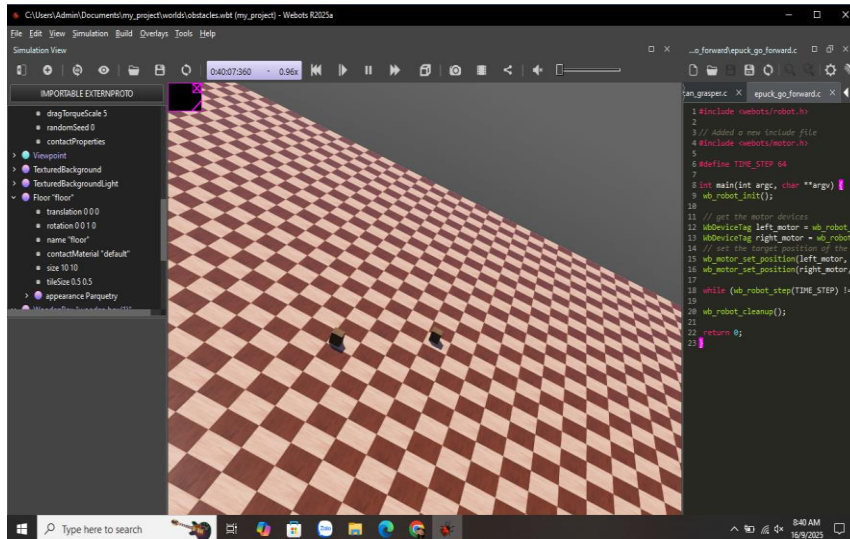


*Đường dẫn file mới có tên **obstacles.wbt***

Hands-on 2:

1. Xóa **RectangleArena**: trong **Scene Tree** tìm node **RectangleArena** Click trái và chọn **Delete**
2. Thêm **Floor(solid)**: Nhấn **Add : PROTO nodes (Webots Projects)** → **objects** → **floors** → **Floor (Solid)**

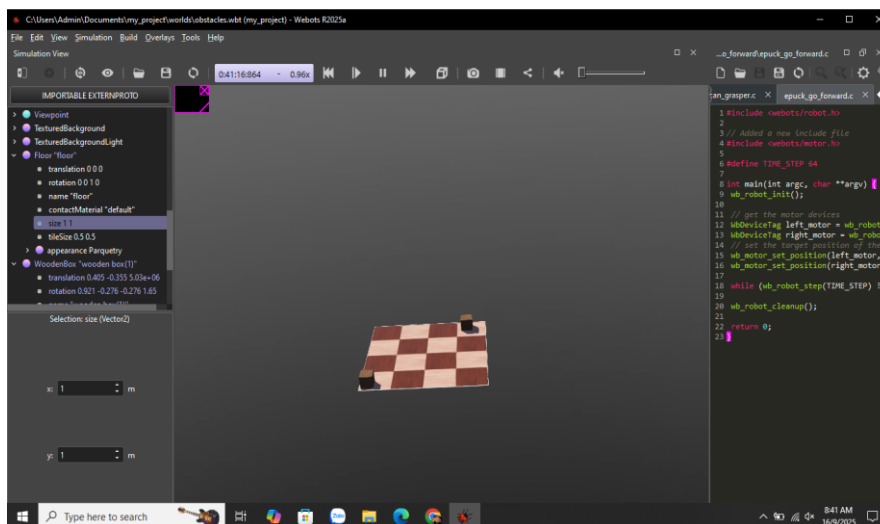
Kết quả một sản phẩm 10x10 xuất hiện trong môi trường



Hình ảnh mặt sàn với kích thước 10x10

Hands-on 3:

1. Trong **Scene Tree** mở rộng các trường của node **Floor**
2. Tìm **field size** và chỉnh kích thước



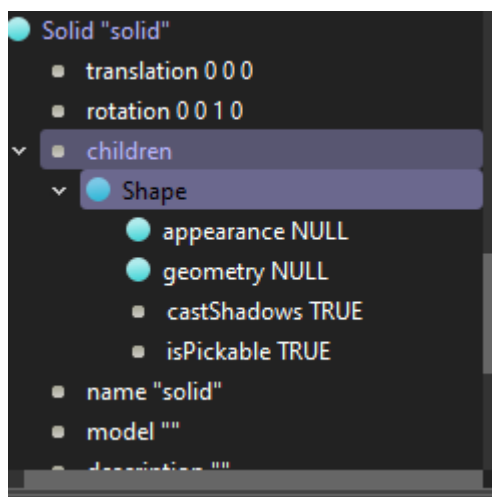
Thay đổi kích thước sang thành 1x1

Hands-on 4:

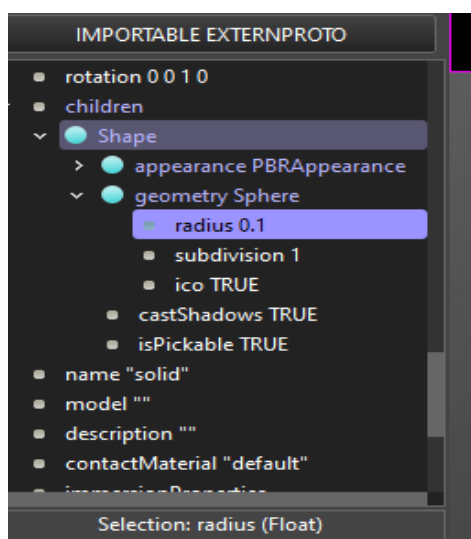
Add → chọn **Base nodes** → **Solid**.

children → nhấn **Add** → **Shape**

Chỉnh hình dạng và tính chất vật thể

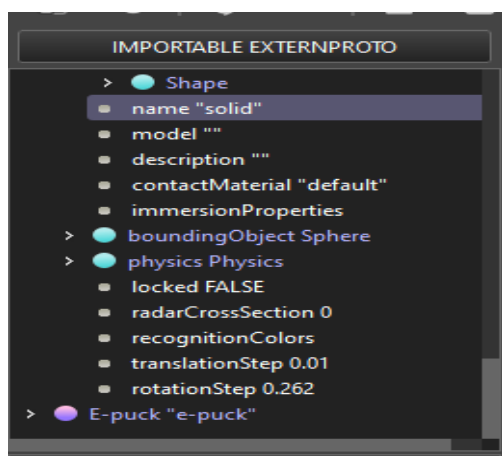


Chỉnh sửa metalness, roughness



Chỉnh sửa Sphere tại geometry

Ngoài ra ta còn có thể chỉnh đường viền và thêm tính chất vật lý cho vật thể



Thêm đường viền và tính vật lý cho vật thể

Trong Shape:

- **appearance** → Add → **PBRAppearance** → đặt **metalness = 0**, **roughness = 1**.

metalness = 0 : Vật thể không có tính kim loại

roughness = 1: bề mặt nhám, không phản xạ ánh sáng rõ nét

- **geometry** → Add → **Sphere**.

Chọn hình dạng cho vật rắn là hình cầu (Sphere)

Trong Solid:

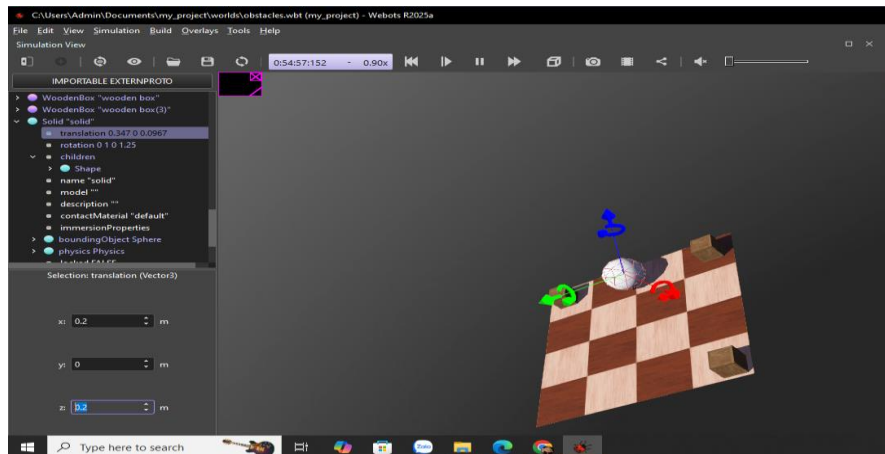
Tạo đường viền cho cho vật rắn

- **boundingObject** → Add → **Sphere**.

Thêm mô phỏng vật lý cho vật rắn

- **physics** → Add → **Physics**.

Thay đổi vị trí vật rắn **translation = {0.2, 0, 0.2}**

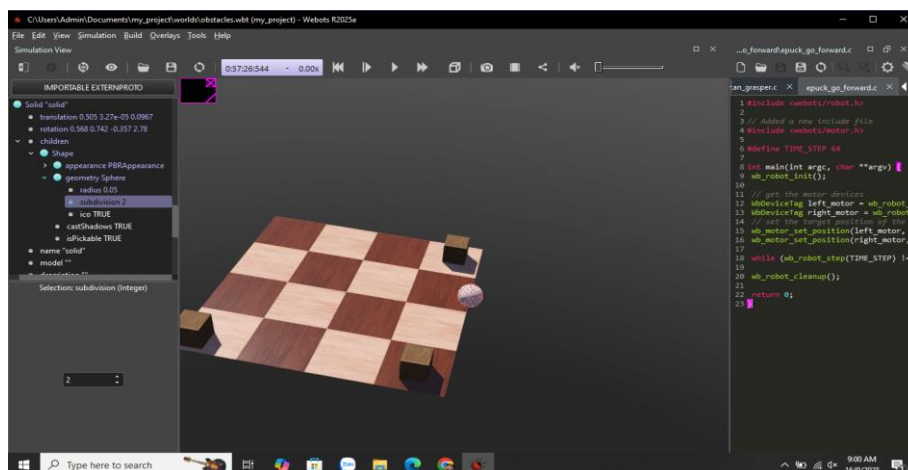


Vật rắn xuất hiện ở vị trí có tọa độ (0.2, 0, 0.2)

Hands-on 5:

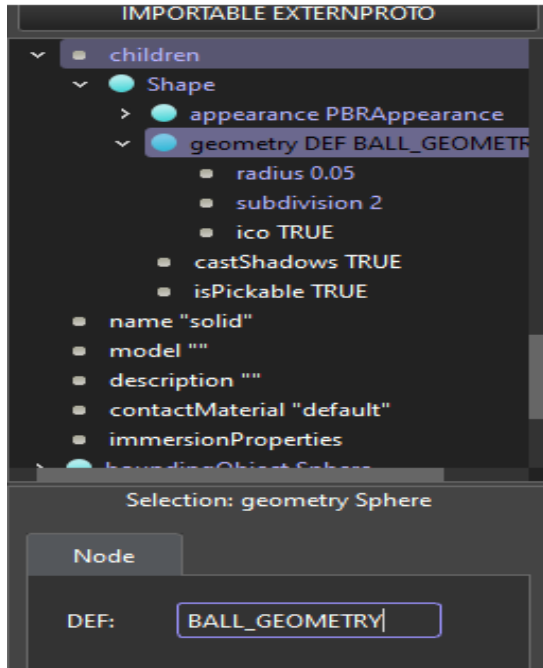
Thay đổi kích thước vật rắn và bề mặt kết xuất

Vào cả 2 **Sphere** (geometry + boundingObject) → đặt **radius = 0.05**, **subdivision = 2**.



Quả bóng sau khi chỉnh thông số nhỏ hơn và mượt hơn

Hands-on 6:

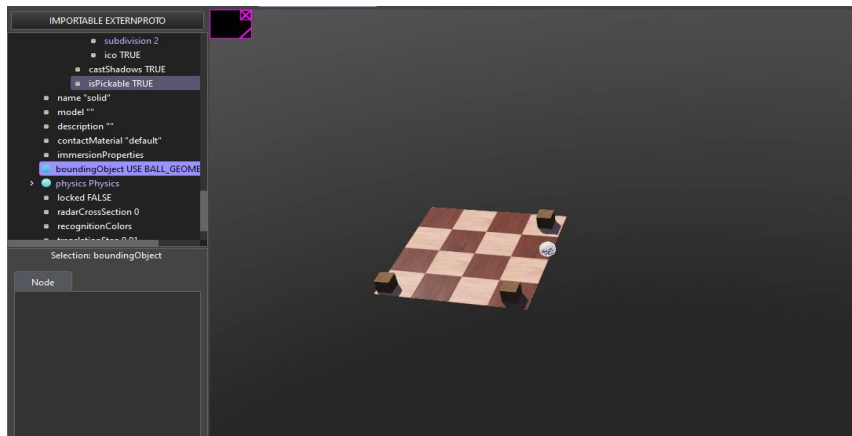


DEF:

Xoá **boundingObject Sphere** và chỉnh
boundingObject = BALL_GEOMETRY

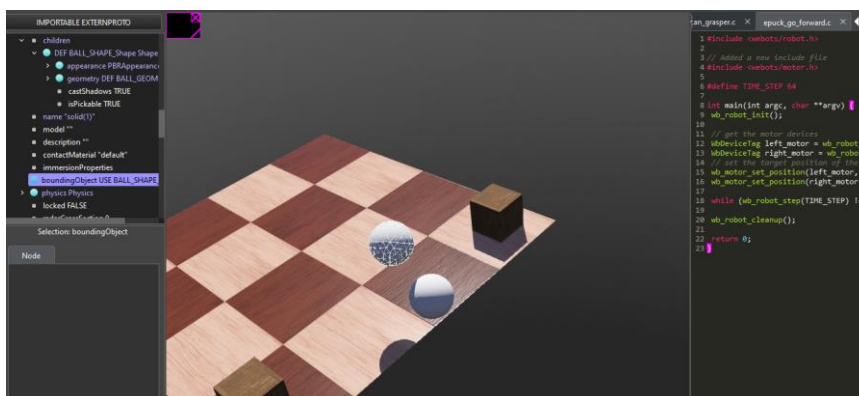
Đặt lại giá trị DEF

Ta có thể sử dụng lại node với cùng thuộc tính nhờ DEF



Đặt USE DEF để có thể sử dụng lại node đó với cùng thuộc tính

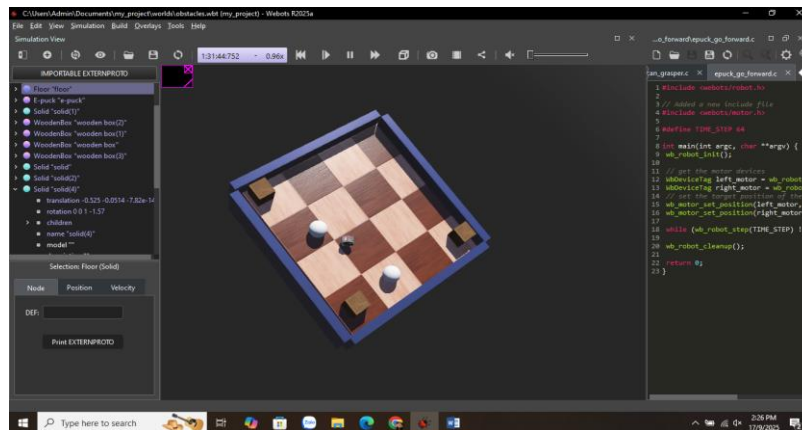
Hands-on 7:



Tạo quả bóng thứ 2 SHAPE với translation

Hands-on 8: Tạo 4 bức tường quanh floor bằng DEF/USE

1. Thêm tường đầu tiên:
 - Trong **Floor: Add-> Solid**
 - Trong Solid mới mở **field children ->Add->Shape**
 - Trong **Shape-> geometry ->Box**
 - Trong **Shape -> appearance ->PBRAppearance**
 - Trong **boundingObject** của **Solid -> Add->USE**
2. Đặt tên **Shape** bằng **DEF**: Trong node **Shape** vừa tạo điền **DEF WALL_SHAPE** như vậy có thể dùng lại cho các tường khác
3. Điều chỉnh kích thước hộp để thành bức tường
4. Copy thành 3 tường còn lại và chỉnh lại translation để đặt ở các cạnh khác của **Floor**



Môi trường đã có thêm tường bao quanh

KẾT LUẬN:

Ở cuối hướng dẫn này, bạn có thể tạo các môi trường đơn giản dựa trên các vật thể rắn. Bạn có thể thêm các nút từ chế độ xem cây cảnh và chỉnh sửa các trường của chúng. Bạn đã làm quen với các nút **Solid**, **Physics**, **Shape**, **Sphere** và **Box**. Bạn cũng đã thấy cơ chế **DEF-USE** cho phép giảm thiểu sự dư thừa node của cây cảnh.

Video kết quả

[Tutorial 2.mp4](#)

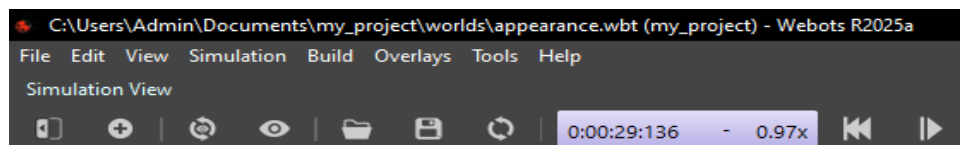
- ➔ Video kết quả đã có đủ 4 bức tường bao quanh đấu trường (arena) và đủ hai quả bóng với các tham số, đường viền và tính chất vật lý như trong Tutorial 2.

TUTORIAL 3: Appearance

Mục đích của hướng dẫn này là giúp bạn làm quen với một số nút liên quan đến kết xuất đồ họa. Các mô phỏng đẹp mắt có thể được tạo ra rất nhanh chóng khi các nút này được sử dụng hợp lý. Chất lượng đồ họa tốt không chỉ nâng cao trải nghiệm của người dùng mà còn rất cần

thiết cho các mô phỏng mà robot nhận thức được môi trường của chúng (xử lý hình ảnh camera, theo dõi đường nét, v.v.).

Hands-on 1: Tạo một file mới với tên “**appearance.wbt**”

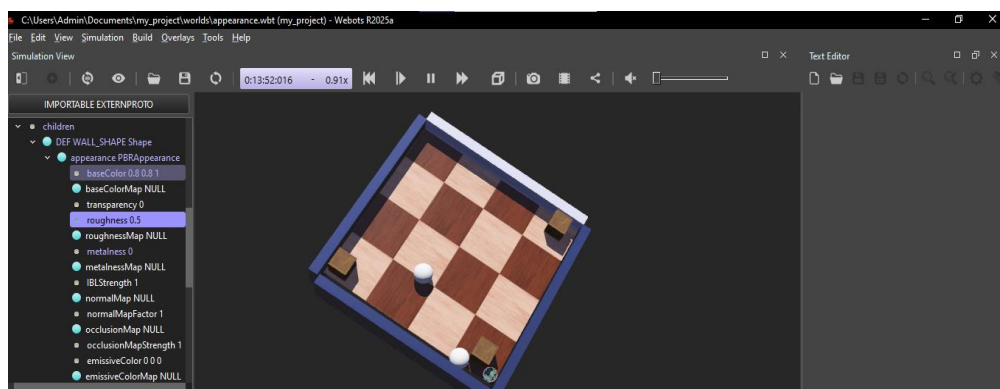


*Đường dẫn thư mục mới với tên **appearance.wbt***

Hands-on 2: Chỉnh màu sắc và chất liệu bức tường

1. Chọn **Shape** node đã tạo bức tường
2. Thêm node **PBRAppearance** vào field **apperacce**
3. Trong **PBRAppearance**:
 - Thay đổi **baseColor** để thay đổi màu sắc
 - Thay đổi **metalness** và **roughness** để thay đổi chất liệu

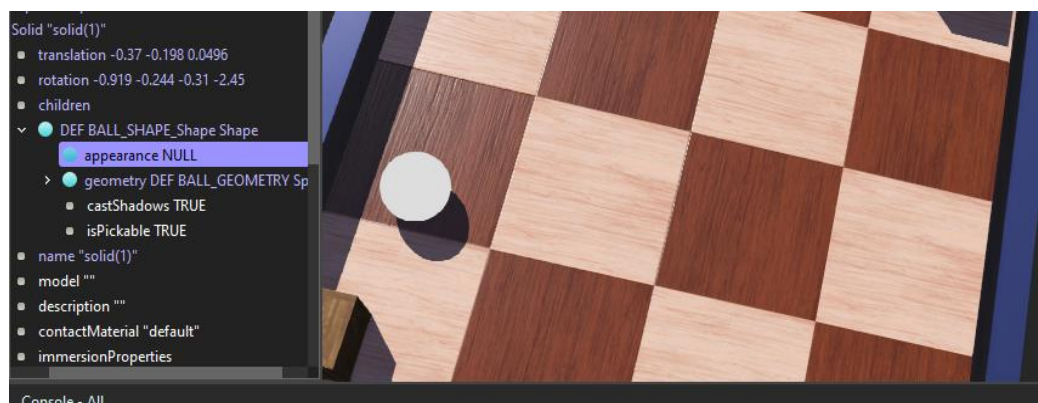
Vì Shape này được gán **DEF WALL_SHAPE** và các tường khác dùng **USE WALL_SHAPE** nên cả 4 bức tường sẽ đổi màu thành xanh dương cùng 1 lúc



*“First wall” sau khi chỉnh tham số **metalness roughness***

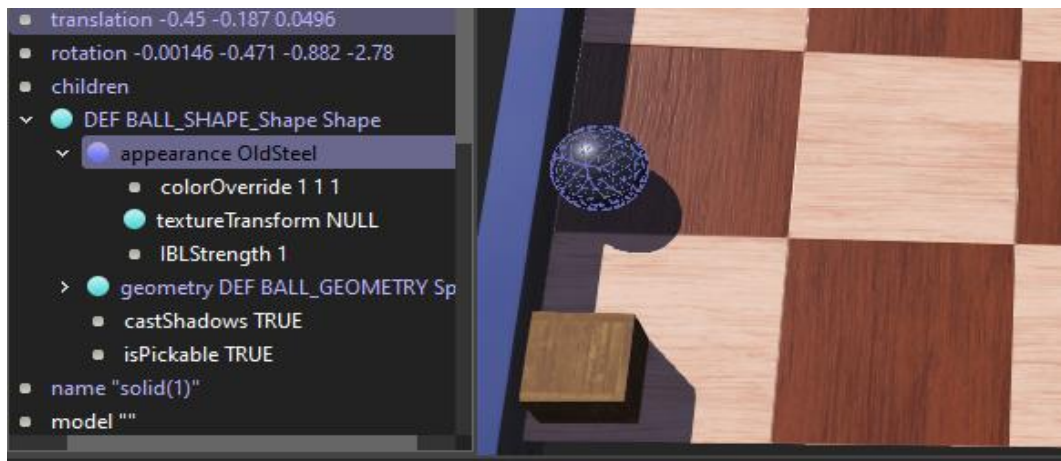
Hands-on 3:

Xóa appearance của quả bóng cũ



*Quả bóng cũ sau khi xóa **appearance***

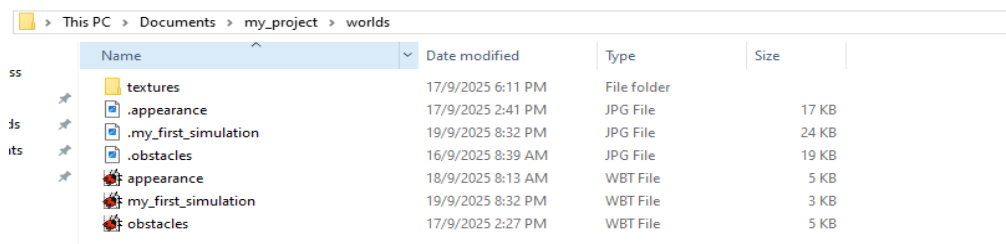
Thêm **OldSteel** cho ball



*Thay thế **OldSteel** cho quả bóng*

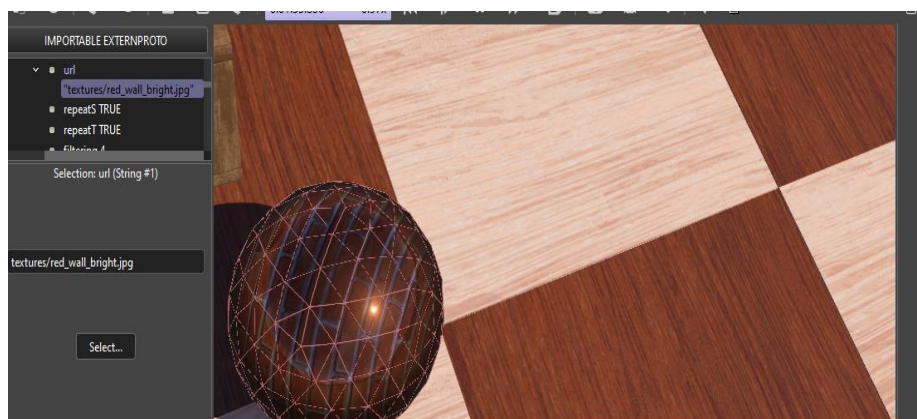
Hands-on 4:

Tạo tập textures → my_project\worlds\textures và add hình red_wall_bright.jpg



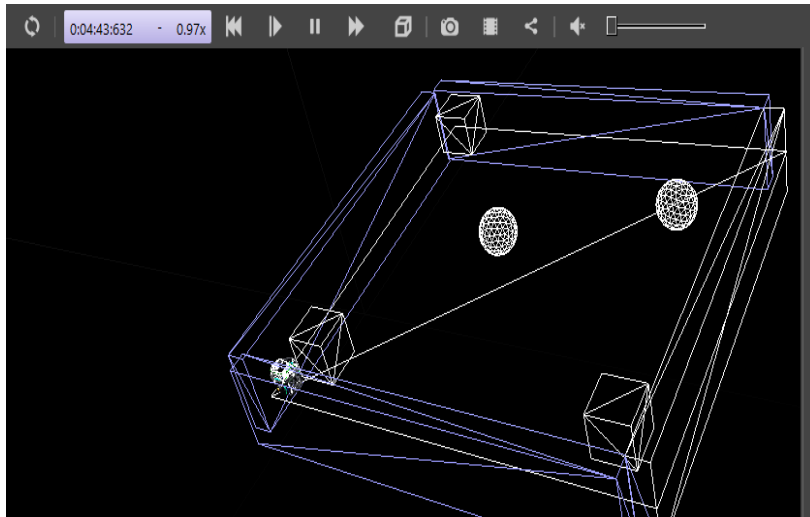
*Đường dẫn tạo file **textures** chứa ảnh cần thêm*

Mở rộng tệp url và thêm đường dẫn đến file ảnh .jpg cho quả bóng

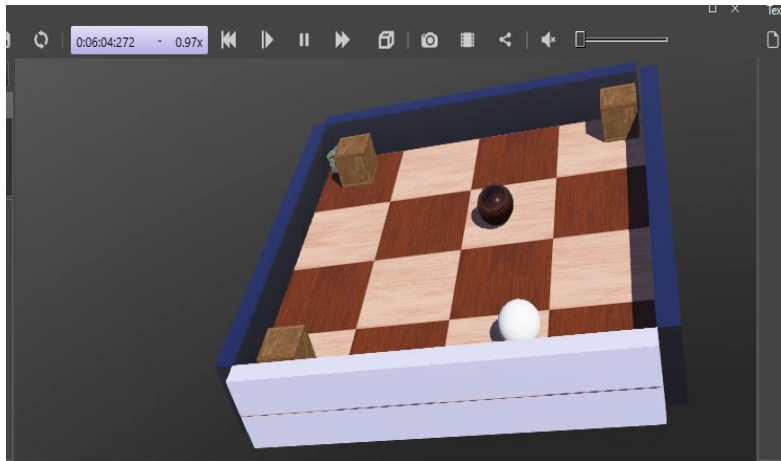


Hình ảnh quả bóng sau khi được thêm viên là một bức tường đỏ “red_wall_bright”

Hands-on 5: Xem các giả lập môi trường sử dụng View/Wirefram Rendering hoặc Plain Rendering



*Xem giả lập trong wireframe sử dụng **Wireframe Rendering***



*Khôi phục lại kết quả cuối cùng sử dụng **Plain Rendering***

KẾT LUẬN:

Trong hướng dẫn này, bạn đã học cách thiết lập môi trường đẹp mắt bằng cách sử dụng node **PBRAppearance** và các node ánh sáng.

[Tutorial 3.mp4](#)

- ➔ Video kết quả đã thêm đường viền “red_wall_bright” cho quả bóng bằng url và chỉnh sửa **metalness** và **roughness** cho bức tường.