

教程 2：修改環境

在本教程中，我們將教您如何在環境中創建簡單對象。第一步將是創建一個與環境互動的球。我們將處理與節點有關的幾個概念：它們的含義，如何創建它們，必須如何關聯它們等等。此外，我們還將了解如何設置物理學。

新的模擬

首先，我們根據教程 1 中創建的模擬創建一個新的模擬。

動手操作 1：確保 `my_first_simulation.wbt` 世界文件已打開，並且模擬已暫停並且處於虛擬時間 0。使用 `File / Save World As...` 菜單，將模擬另存為 `obstacles.wbt`。

Hands-on #1: Make sure the `my_first_simulation.wbt` world file is open, and that the simulation is paused and is at a virtual time of 0. Using the `File / Save World As...` menu, save the simulation as `obstacles.wbt`.

修改地板

默認的 `RectangleArenaPROTO` 定義固定在靜態環境上的簡單地板，即沒有 [Physics](#) 節點，並被牆壁包圍。`Webots` 對像庫中提供了其他預建樓層。現在，我們將刪除該 `RectangleArena` 節點，並添加一個簡單的地板，在本教程後面的部分中，我們將使用牆手動對其進行環繞。

動手 #2：要刪除 `RectangleArena`，請在 3D 視圖或場景樹視圖中單擊鼠標左鍵，然後按 `Delete` 鍵盤上的鍵。或者，您可以在 3D 視圖中右鍵單擊它，然後 `Delete` 在上下文菜單中選擇（也可以直接在場景樹視圖中使用上下文菜單）。選擇 `TexturedBackgroundLight` 節點，然後單擊 `Add` 按鈕。在打開的對話框中，然後選擇 `PROTO nodes (Webots Projects) / objects / floors / Floor (Solid)`。

Hands-on #2: To remove the `RectangleArena`, select it either in the 3D view or in the scene tree view with a left click and press the `Delete` key on your keyboard. Alternatively, you can right click on it in the 3D view and select `Delete` in the context menu (you can also use the context menu directly

in the scene tree view). Select the `TexturedBackgroundLight` node and click on the `Add` button. In the open dialog box, and choose `PROTO` nodes (Webots Projects) / objects / floors / Floor (Solid).

新添加的 `FloorPROTO` 的默認大小為 `10mx10m`，但是可以通過更改相應的字段來調整其大小，位置和紋理。

動手 #3：在場景樹視圖中，選擇並展開 `Floor`。修改 `size` 字段並將其設置為將 `{1, 1}` 其大小調整為 `1mx1m`。

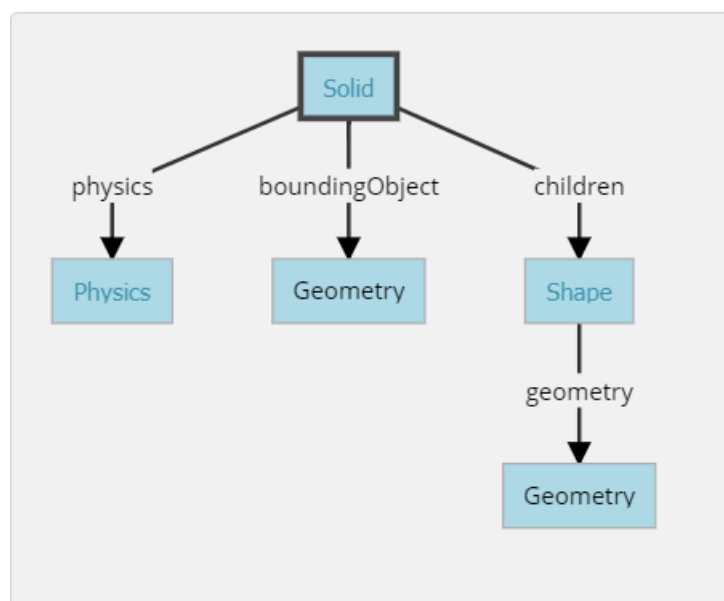
Hands-on #3: In the scene tree view select and expand the `Floor`. Modify the `size` field and set it to `{1, 1}` to resize it to `1mx1m`.

實體節點

本小節介紹了 `Webots` 中最重要基礎節點：`Solid` 節點，許多其他節點都來自 `Solid` 節點。

`Webots` 的物理引擎僅設計用於模擬剛體。設計仿真時，重要的一步是將各個實體分解為單獨的剛體。

要定義剛體，您將必須創建一個 `實體` 節點。在此節點內，您將根據剛體的特性設置不同的子節點。下圖描繪了剛體及其子節點。`實體` 節點的圖形表示由填充其列表的 `Shape` 節點定義 `children`。碰撞範圍在其 `boundingObject` 字段中定義。圖形表示和碰撞形狀通常但不一定相同。最後，該 `physics` 字段定義對像是屬於動態環境還是屬於靜態環境。所有這些子節點都是可選的，但是 `physics` 需要 `boundingObject` 定義該字段。



幾何框（在[此圖中](#)）代表任何種類的幾何圖元。實際上，它可以用 [Sphere](#)，[Box](#)，[Cylinder](#) 等代替。

創建一個球

現在，我們將球添加到模擬中。如本那個球將被建模為剛性體圖。一[球](#)節點將被用來定義我們的球的幾何形狀。

動手操作 4：在場景樹視圖中，選擇最後一個節點，然後按 [Add](#) 按鈕。在對話框中，打開該 [Bases nodes](#) 部分，然後選擇“[實體](#)”節點。在場景樹視圖中，展開“[實體](#)”節點並選擇其 [children](#) 字段。使用按鈕向其添加一個 [Shape](#) 節點 [Add](#)。選擇“[形狀](#)”節點的 [appearance](#) 字段，然後使用按鈕添加一個節點。AddPBRAppearance

1. 將[球體](#)節點添加為 [geometry](#) 新創建的[形狀](#)節點的字段。
2. 展開 [PBRAppearance](#) 節點並將其 [metalness](#) 字段更改為 0，並將其 [roughness](#) 字段更改為 1。
3. 將另一個 [Sphere](#) 節點添加到 [Solid](#) 的 [boundingObject](#) 字段。
4. 最後，將一個 [Physics](#) 節點添加到 [Solid](#) 的 [physics](#) 字段中。
5. 通過修改[實體](#)節點的 [translation](#) 字段，將球放在機器人的前面(例如)。{0, 0.2, -0.2}
6. 保存模擬。
7. 結果如圖[所示](#)。

Hands-on #4: In the scene tree view, select the last node and press the [Add](#) button. In the dialog, open the [Bases nodes](#) section and select the [Solid](#) node. In the scene tree view, expand the [Solid](#) node and select its [children](#) field. Add a [Shape](#) node to it by using the [Add](#) button. Select the [appearance](#) field of the [Shape](#) node and use the [Add](#) button to add a [PBRAppearance](#) node.

1. Add a [Sphere](#) node as the [geometry](#) field of the newly created [Shape](#) node.
2. Expand the [PBRAppearance](#) node and change its [metalness](#) field to 0 and its [roughness](#) field to 1.
3. Add another [Sphere](#) node to the [boundingObject](#) field of the [Solid](#).
4. Finally add a [Physics](#) node to the [physics](#) field of the [Solid](#).
5. By modifying the [translation](#) field of the [Solid](#) node, place the ball in front of the robot (at {0, 0.2, -0.2} for example).

該 **DEF-USE 機制** 允許在一個地方定義一個節點，並在場景樹在其他地方重複使用的定義。這對於避免在世界文件中復制相同節點很有用。此外，它還允許用戶同時修改多個對象。它是這樣工作的：首先用 **DEF** 字符串標記節點。然後，可以使用 **USE** 關鍵字將該節點的副本重用於其他地方。只能編輯 **DEF** 節點的字段，**USE** 的字段從 **DEF** 節點繼承，並且不能更改。該機制取決於 **world** 文件中節點的順序。應在任何相應的 **USE** 節點之前定義 **DEF** 節點。

我們之前用來定義球的兩個 [Sphere](#) 定義是多餘的。現在，我們將使用 **DEF-USE** 機制將這兩個 [Sphere](#) 合併為一次。

動手操作 6：在場景樹視圖中選擇第一個 [Sphere](#) 節點（[Shape](#) 的子級）。場景樹視圖的[字段編輯器](#)允許您輸入 **DEF** 字符串。

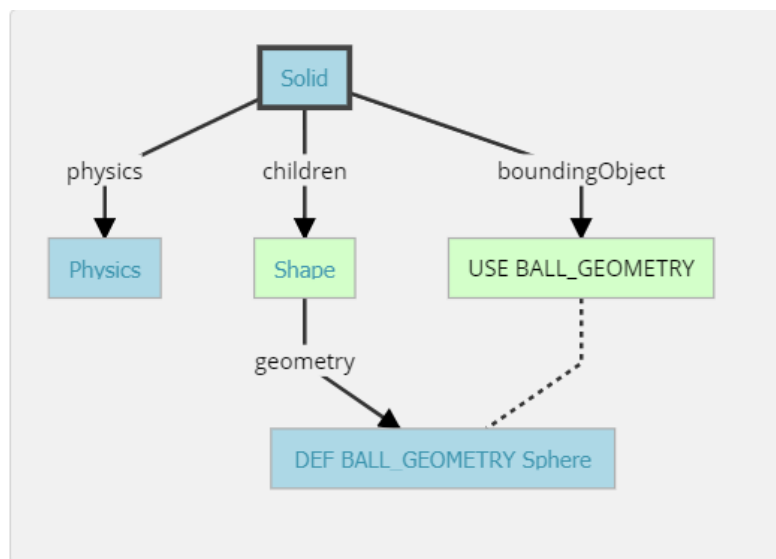
1. **BALL_GEOMETRY** 在此字段中輸入。
2. 選擇 **boundingObject** 字段（包含第二個 [Sphere](#) 節點），然後通過右鍵單擊場景樹中的字段並 **Delete** 在彈出的上下文菜單中選擇條目來將其清空。
3. 然後，選擇 **boundingObject** 字段並單擊 **Add** 按鈕，然後 **USE / BALL_GEOMETRY** 在對話框中選擇。
4. 結果如圖[所示](#)。

Hands-on #6: Select the first [Sphere](#) node (the child of the [Shape](#)) in the scene tree view. The [field editor](#) of the scene tree view allows you to enter the DEF string.

1. Enter **BALL_GEOMETRY** in this field.
2. Select the **boundingObject** field (containing the second [Sphere](#) node), and empty it by right clicking the field in the scene tree and choosing the **Delete** entry in the context menu that pops up.
3. Then, select the **boundingObject** field and click on the **Add** button, and select the **USE / BALL_GEOMETRY** in the dialog box.
4. The result is shown in [this figure](#).

現在，更改 **radius** 第一個 [Sphere](#) 節點的字段也會修改其 **boundingObject**。

為方便起見，該 **boundingObject** 字段還接受 [Shape](#) 節點（而不是直接接受 [Sphere](#) 節點）。也可以在 [Shape](#) 級別使用相同的 **DEF-USE** 機制，如圖[所示](#)。到目前為止，最大的好處就是也可以直接將此 [Shape](#) 用於圖形目的。後來，對於某些傳感器，該機制將變得非常有用。



動手練習 7：創建第二個球，使用相同的參數，但使用 [Shape](#) 節點（而不是直接使用 [Sphere](#) 節點）作為 DEF-USE 機制。

Hands-on #7: Create a second ball with the same parameters but using the [Shape](#) node (rather than the [Sphere](#) node directly) for the DEF-USE mechanism.

添加牆

為了驗證您的進度，請自己實施四堵牆以包圍環境。必鬚根據環境靜態定義牆壁。要了解靜態和動態之間的區別，我們將定義的物體（球）放在地面上方。如果“[物理](#)”節點為 NULL，則在仿真過程中它將保持凍結狀態（靜態情況）。如果該 `physics` 字段包含“[物理](#)”節點，則它將屬於重力作用（動態情況）。

在 `Shape` 級別而不是 `Geometry` 級別，盡可能使用 DEF-USE 機制。實際上，在 `Solid` 節點的 `boundingObject` 字段中添加中間 `Shape` 節點更為方便。實現

牆的最佳幾何原語是 **Box** 節點。所有牆壁僅需定義一個形狀。預期結果如圖所示。

動手 #8：添加四面牆，而不添加物理場，並且僅使用 [Shape](#) 節點的一個定義。

Hands-on #8: Add four walls without physics and using only one definition of the [Shape](#) node.

效率

剛體的模擬在計算上是昂貴的。可以通過最小化邊界對象的數量，最小化邊界對象之間的約束（在下一個教程中有關約束的更多信息）以及最大化 `WorldInfo.basicTimeStep` 字段來提高仿真速度。在每個模擬中，必須在模擬速度和真實性之間找到一個折衷。

結論

在本教程的最後，您將能夠基於剛體創建簡單的環境。您可以從場景樹視圖添加節點並修改其字段。您已經熟悉了 [Solid](#)，[Physics](#)，[Shape](#)，[Sphere](#) 和 [Box](#) 節點。您還看到了 DEF-USE 機制，該機制可減少場景樹的節點冗餘。