

協同產品設計實習

期末簡報

四設二乙_41223243_廖耕毅

作業網址

41223243:<https://mdec2025.github.io/hw-41223243/content/Final.html>

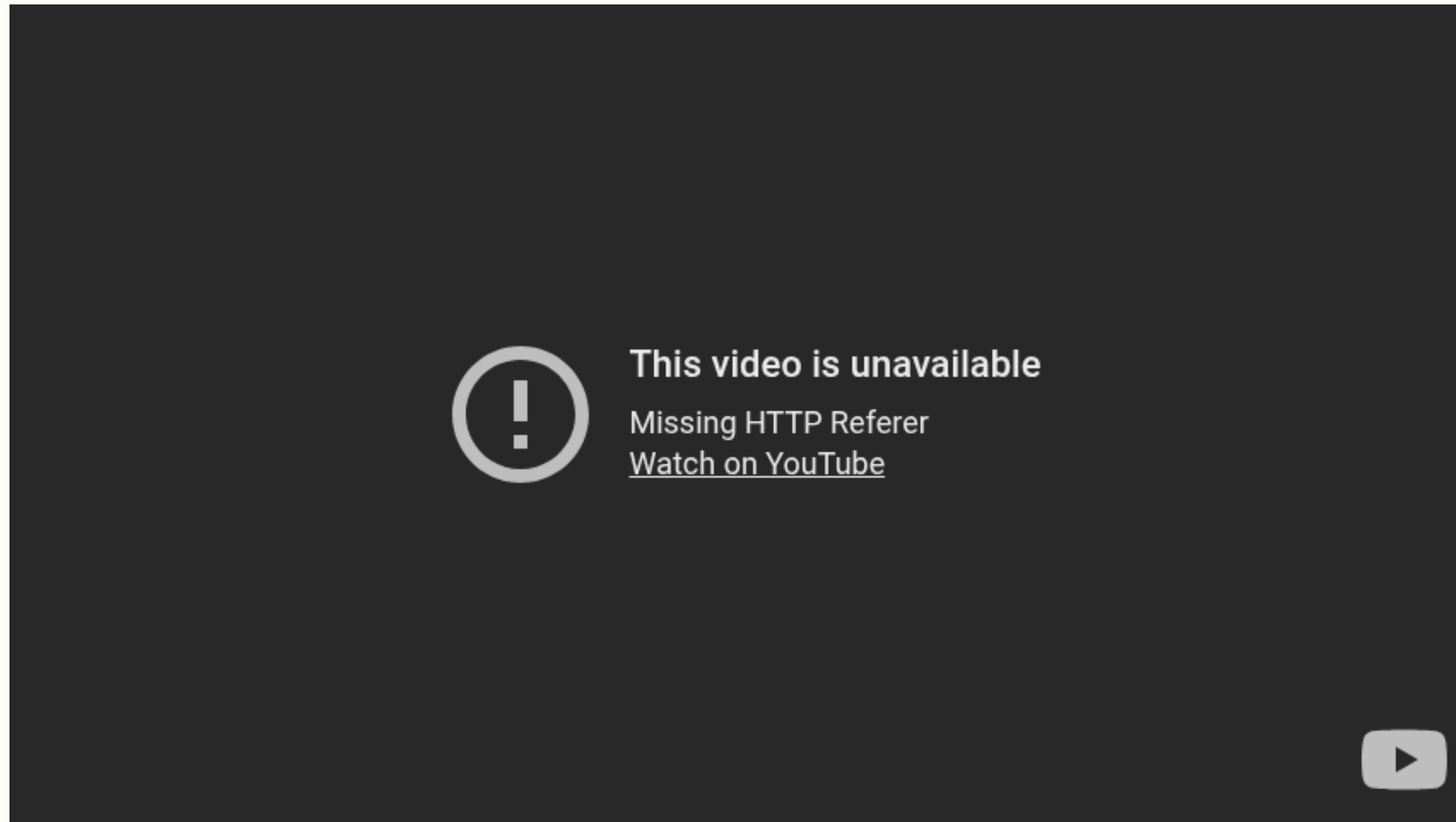
組裝影片

影片網址:<https://www.youtube.com/watch?v=12J9qm32pJQ>



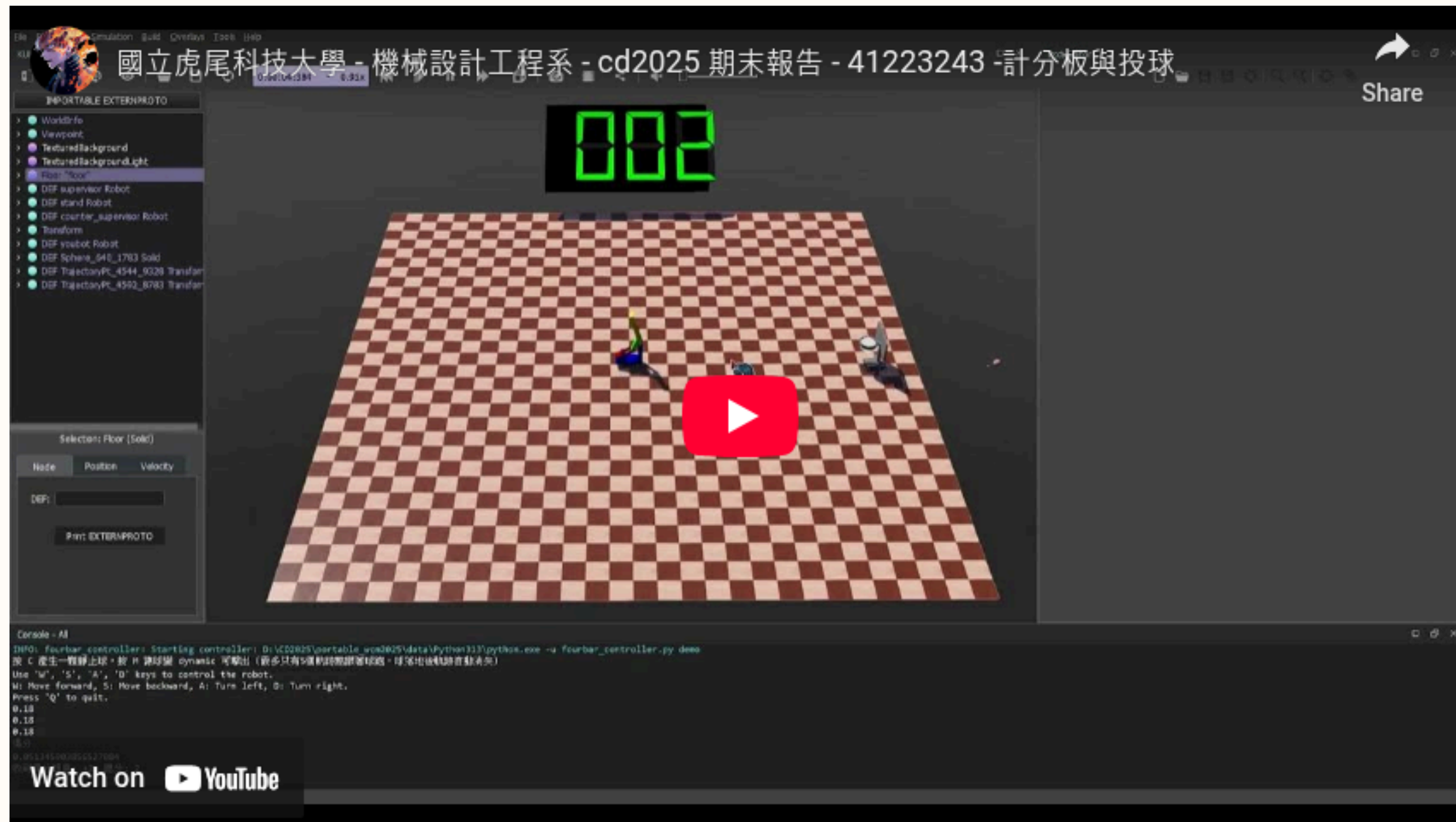
模擬影片(輪車移動)

影片網址:https://www.youtube.com/watch?v=nA7_NVboh0s



模擬影片(投籃與計分)

影片網址:<https://www.youtube.com/watch?v=24chXa1bfHo>

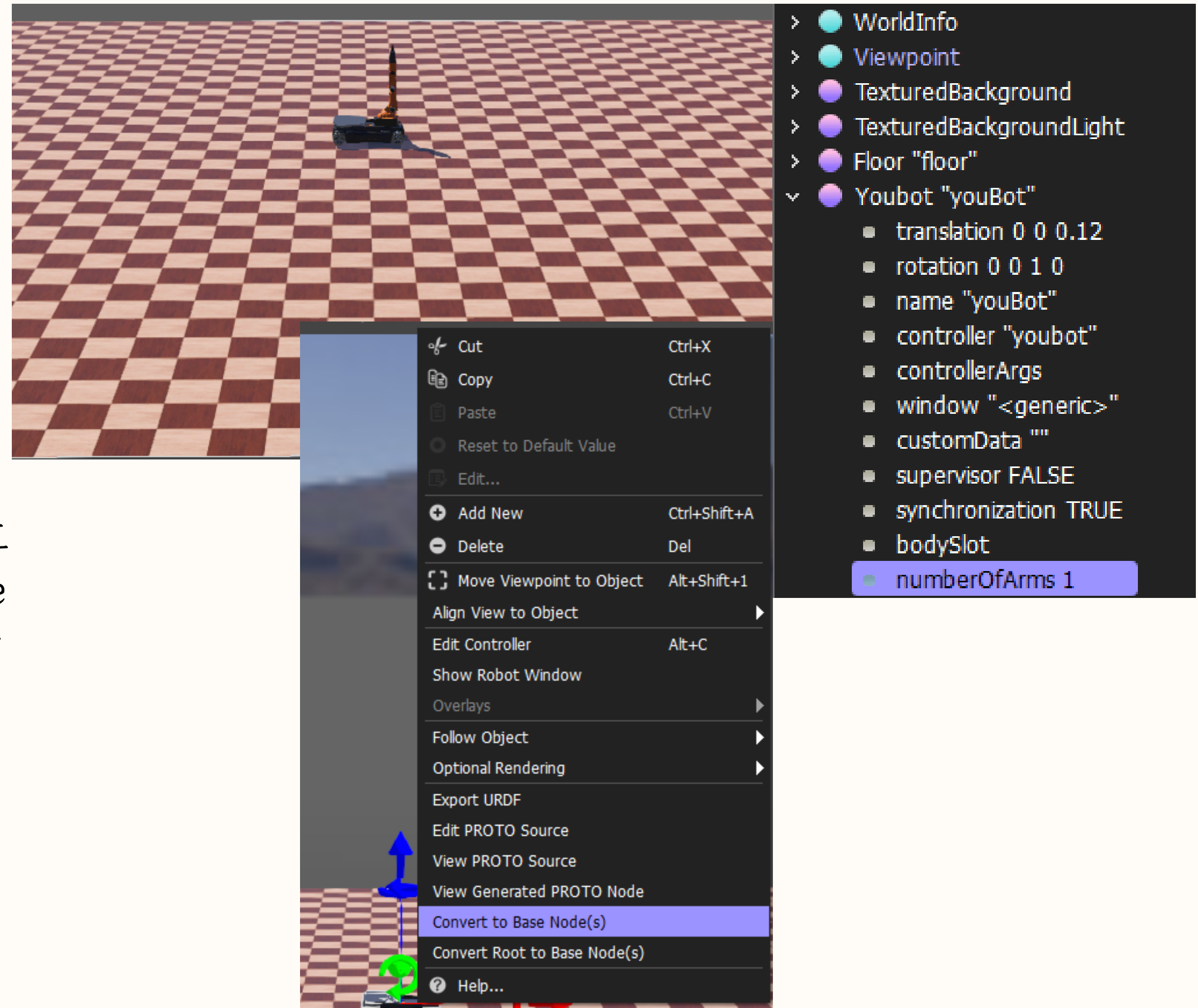


組裝步驟

這次組裝的部分，須先由輪車的部分開始處理。

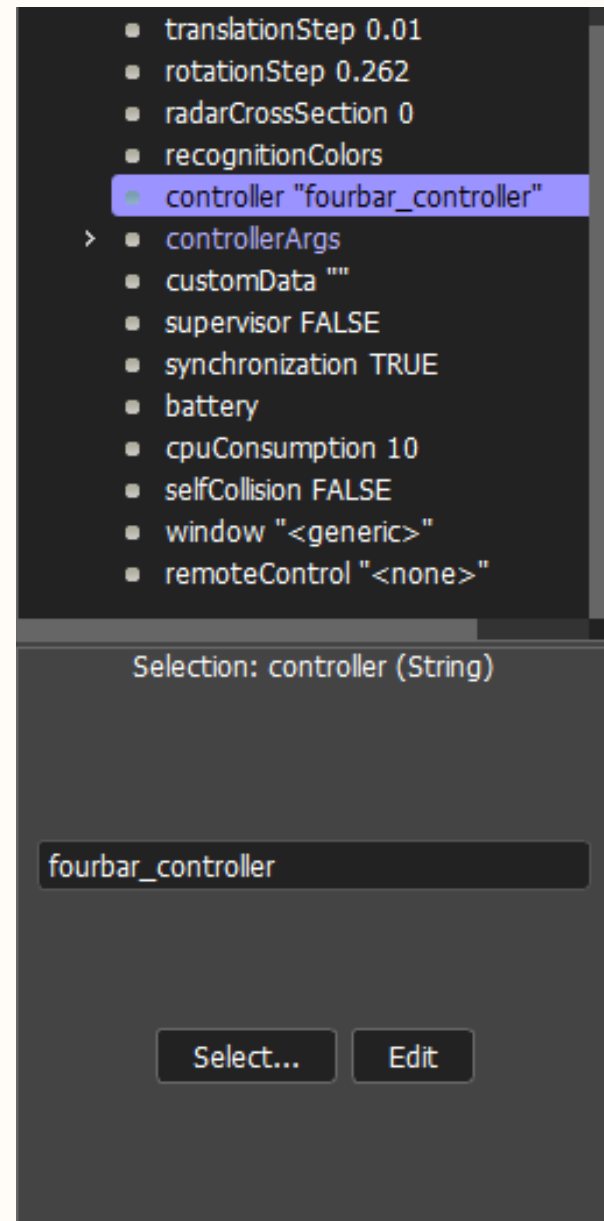
當新建好專案，並設置一個floor後就可以右鍵點擊地板，並選擇Add New這一選項，然後於Find部分搜尋Youbot，然後Add，此時就會像右方圖片一樣出現一台帶有機械臂的輪車，然而機械臂並非我們所需要，所以我們必須將其移除。

在右方的場景樹選取Youbot，於numberOfArms將1修改為零，然後再右鍵點擊車子，選擇Convert to Base Node就可以將車子拆解成可編輯的形式，然後就是再車身上安裝板子與發射器。



組裝步驟

發射器

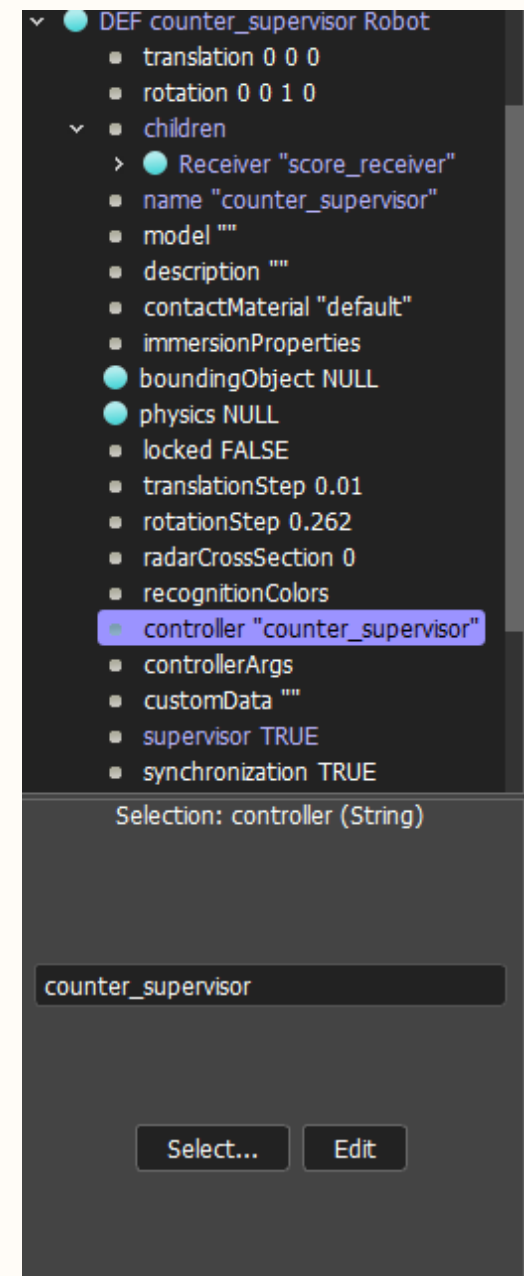
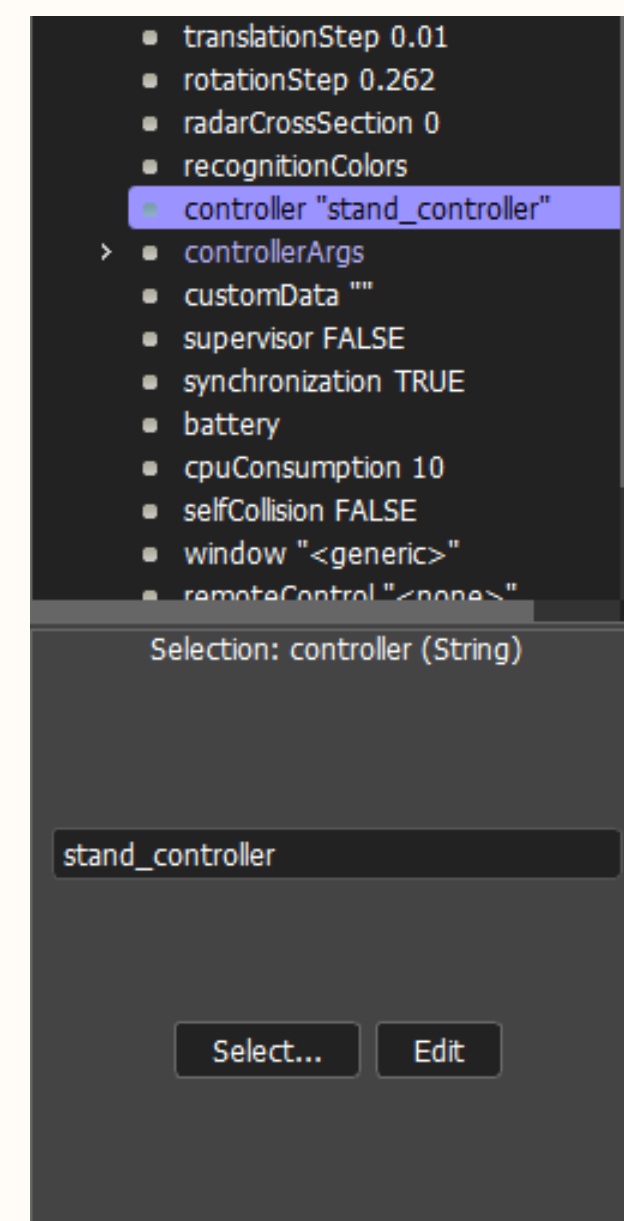


而籃框的部分與發射器差不多，童都是先拆輪車出來，然後安上籃框，與偵測後，並且將計分板丟上去、設置 counter_supervisor，分別匯入控制程式。

這個控制程式包含了輪車的移動與計分的功能。

而上一頁中，輪車與發射器的控制程式匯入與此步驟相同。

籃框



在這個設計中，我們使用一個名為 `counter_supervisor` 的 Robot 節點作為系統的計分管理模組。雖然它在模擬中並不具有外觀或行動能力，但 Webots 的架構規定所有控制器必須掛載在 Robot 節點上，因此即便該模組實質上只是用於監聽得分訊號與控制七段顯示器，它仍必須以 Robot 節點的形式存在。

控制器程式中透過 `supervisor = Supervisor()` 創建出一個具有監控能力的控制器物件，允許我們使用 Webots 的高階 API（如 `getFromDef`, `getDevice`, `getField`, `setSFColor` 等）來直接修改場景中的材質、顯示數字，或根據外部傳來的得分訊息進行即時更新。

此種設計模式充分利用了 Webots 對 Supervisor 的支援，讓我們能夠將視覺介面（七段顯示器）與模擬邏輯（得分處理）整合在一個獨立模組中，維持主控程式與顯示控制之間的低耦合性，提升系統可讀性與可維護性。

心得

本學期著重於教學webots的模擬與操作方式，這個過程可以說是困難重重，不單只是要繪製物件、還要去留意webots裡的每個部件的安排，然後還要匯入控制程式，有時候模以出來的結果與範例不同，又要一個一個東西檢查，非常麻煩，甚至有時候可能是物件的名稱與控制程式不同，可能差個一個字，就會導致無法模擬，然後第一時間想到的也不會是這問題。

這個期末報告光是重新建立然後刪除錯誤世界這一程序，我就走了4、5次以上後來真的是對著範例研究很久，才勉強弄出來。而且還常常因為python的某些套件沒有安裝，導致某些控制程式的失效，我的狀況更慘，webots甚至不會去讀我可攜裡的python套件，而是讀取我電腦裡很久之前下載的，導致套件東缺稀少，東西一開始模擬就閃退，後來問半天才找到指定webots使用指定python方式，才解決閃退問題。

總結來說webots雖然開源，但這個程式的操作門檻實在不算低，裡面的東西牽一髮動全身，少任何一個東西都能導致模擬失敗，但也讓我藉機學到做這種東西最重要的一點，就是耐心跟細心。

感謝觀看