Assignment 1:

Due March 25, 2020 for class 2a and March 26, 2020 for class 2b.

1. Describe how to do an efficient random grouping for this course or do the roll calling randomly?
2. Describe how to prepare a portable Python programming system for Windows 10 64bit system to allow one the maintain CMSiMDE website, Pelican blog and Reveal.js presentation on Github?
3. What do you need to know from http://www.coppeliarobotics.com/helpFiles/index.html to implement a four-wheeled robot?

亂數分組要注意下列事項

1. 注意資料來源是否為 https，否則將無法成功亂數分組
2. 所有資料來源都來自 http，也就是說輸出 URL 都在 http，記得更改不要只有更改 Dart 的 URL
3. 乙班名單

https://s1.mde.nfu.edu.tw:7443/?semester=1082&courseno=0780

可攜系統升級 python 3.8 注意事項

1. 安裝 python 時先不要安裝 pip 檔，若是將 pip 檔打勾那下載下來 python 3.8.exe 裡面就會有 pip 檔
2. 安裝好沒有 pip 檔的 python 之後再輸入 python -m pip install

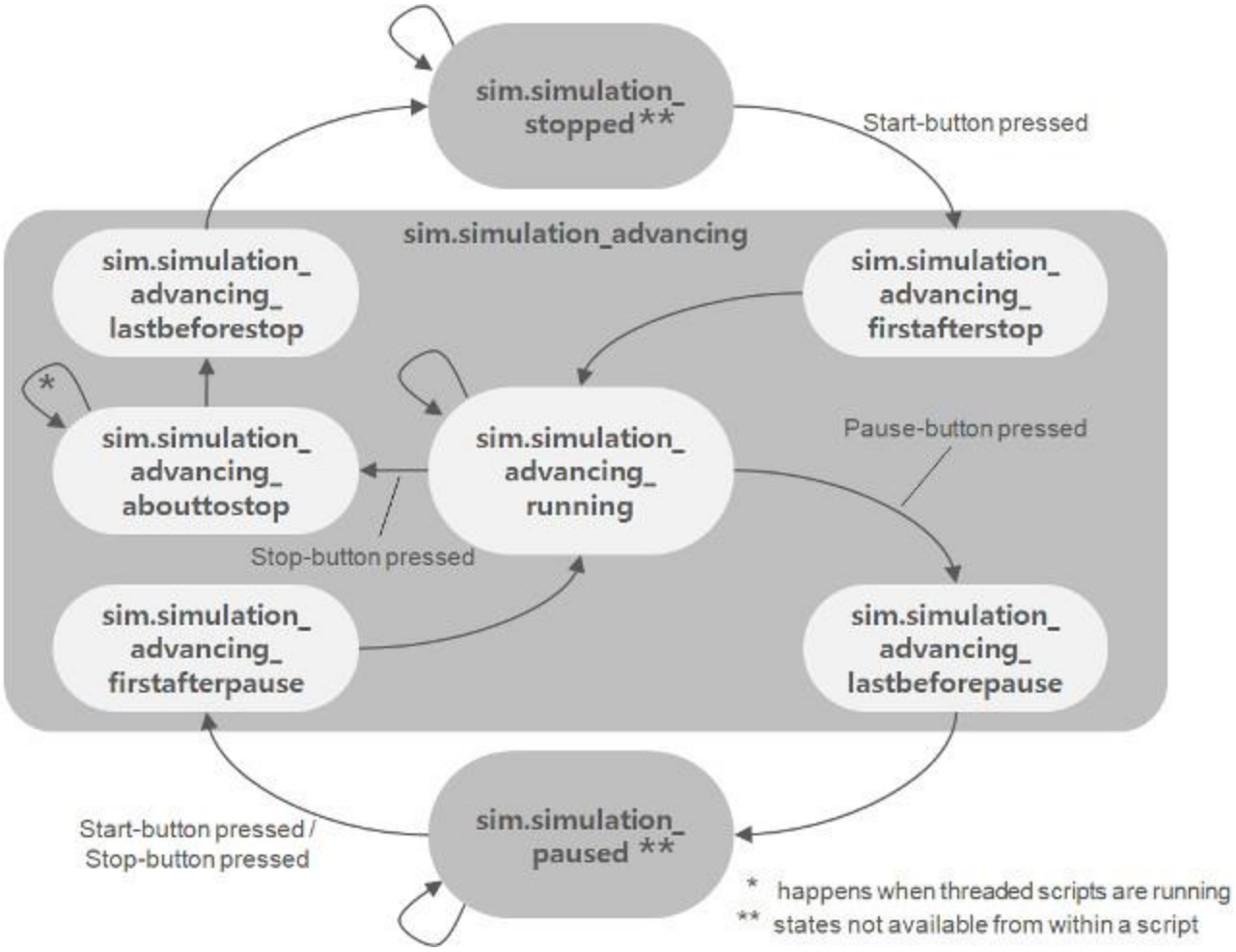
翻譯四輪車

模擬

在 coppeliaSim 中的模擬可透過[選單-->模擬-->開始/暫停/停止模擬]或相關工具欄按鈕來啟動

[Simulation start/pause/stop toolbar buttons]

為了能正確告知下一步的狀態，在程序中模擬器將會使用其他內部狀態，以下為模擬器內部狀態



[Simulation state diagram]

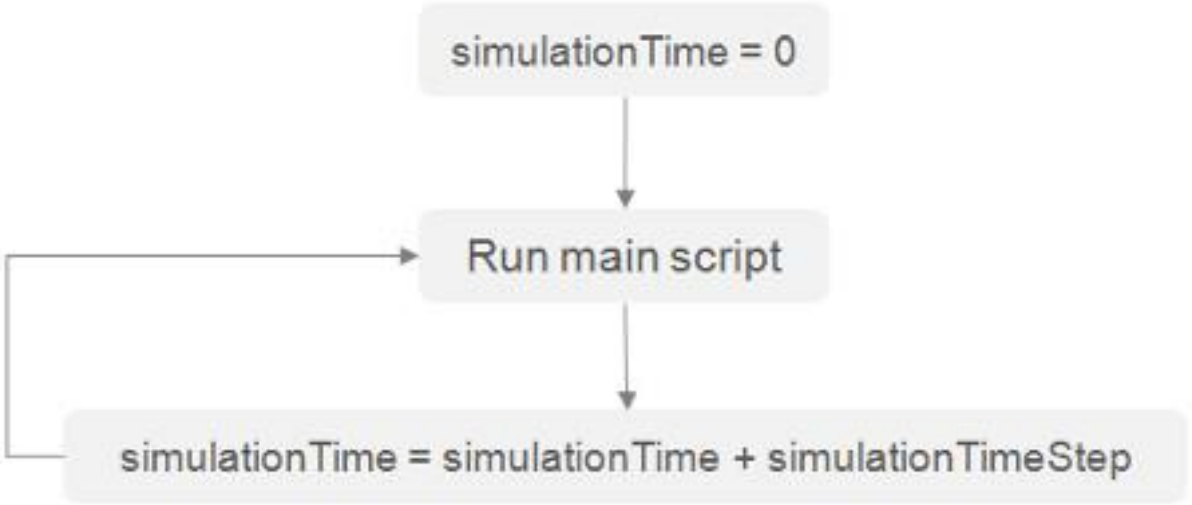
為了能正確執行，腳本與程序必須根據當前系統及模擬型態進行修正，較佳的作法為將每個控制代碼分成 4 個系統函數

- **初始函數：僅在腳本初始化使用**
- **激活函數：僅在激活時使用.**
- **感知函數：僅在感應發生使用.**
- **清除函數:僅在函數為初始化之前使用**

有關如何操作腳本的例子請參考 main script, child scripts 和 customization scripts

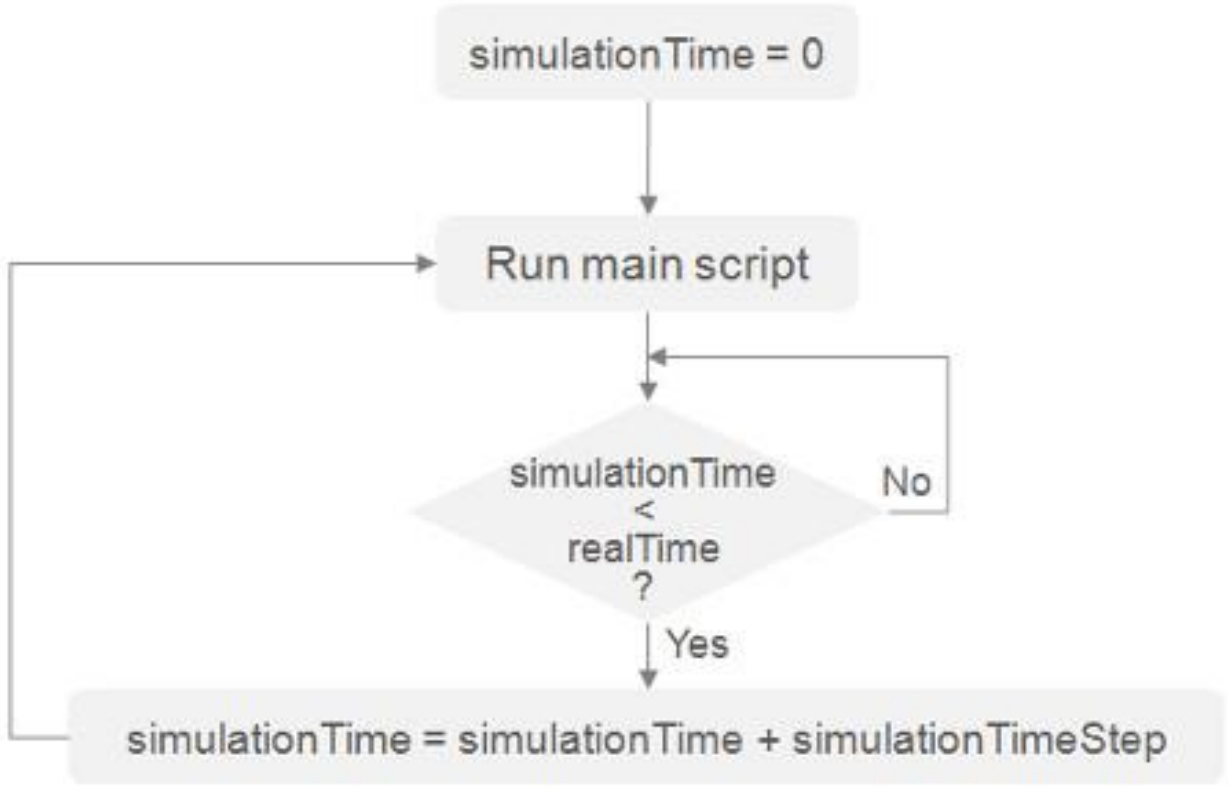
Simulation loop

模擬器透過縮短時間提高仿真的操作，下圖說明了主要仿真循環



[Main simulation loop]

Real-time 模擬支援模擬同步時間



[Real-time simulation loop]

下面是已經精簡過的客戶端應用程序（為了淺顯易懂，下列程式碼已經省略了訊息、插件處理及其他詳細信息）:

```
void initializationCallback
{
    // do some initialization here
}

void loopCallback
{
    if
( (simGetSimulationState()&sim_simulation_advancing)!=0 )
    {
        if
( (simGetRealTimeSimulation()!=1)||(simIsRealTimeSimulationSt
epNeeded()==1) )
        {
            if
((simHandleMainScript()&sim_script_main_script_not_called)==0
)
                simAdvanceSimulationByOneStep();
        }
    }
}

void deinitializationCallback
{
    // do some clean-up here
}
```

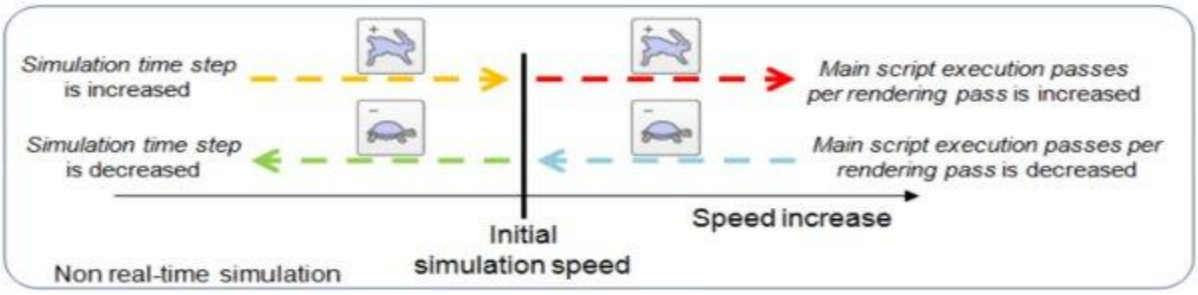
取決於仿真困難度，有時電腦不一定可以執行。

Simulation speed

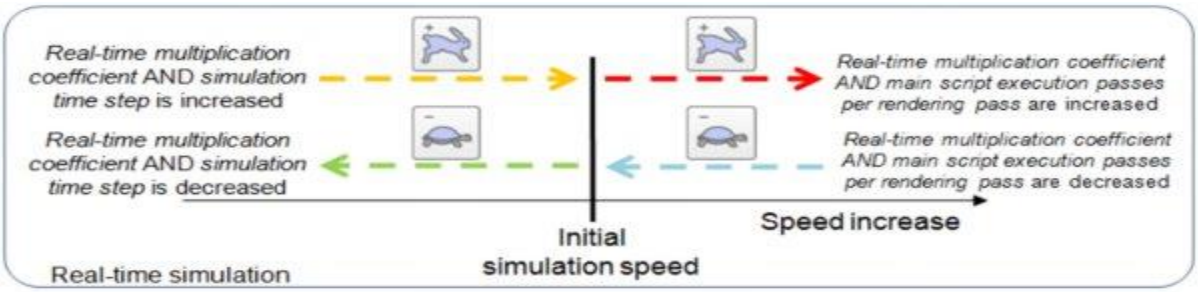
在 non real-time 模擬，模擬速度取決於兩個要素 ： 模擬時間長短和一個渲染的模擬遍數。 在 real-time 仿真的情況下，模擬速度主要取決於實時乘法係數，而且在一定程度上取決於模擬時間長短（太小的仿真時間可能與 real time 時間不兼容）。 由於電腦的計算能力有限，因此無法進行模擬。 在模擬過程中，可以使用以下工具欄按鈕來調整模擬速度

[Simulation speed adjustment toolbar buttons]

以特殊方式調整模擬速度使初始模擬時間永遠不會增加，下圖說明了模擬速度調整機制



[Simulation speed adjustment mechanism for **non real-time simulations**]



在默認情況下，每個周期由以下順序組成

- **執行 main script**
- **渲染場景**

Threaded rendering

渲染將會增加模擬時間，因此我們可以定義每個場景腳本執行次數，但有些特殊情形下這些步驟會不管用，因為渲染仍然會減慢每個第 x 個模擬週期的時間。 在這種情況下，可以通過用戶設置或以下工具欄按鈕激活線程渲染模式

[Threaded rendering toolbar button]

激活線程渲染模式後，模擬週期將包含在執行主腳本中因此模擬將以最大速度運行。渲染將通過不同的線程進行，並且不會減慢模擬任務的速度。然而必須考慮以下缺點

- **渲染將與模擬循環異步發生，並且可能會出現視覺故障**
- **錄像機無法以恆定速度運行**
- **應用程序的穩定性可能會降低**
- **某些操作需要等待渲染線程完成工作才能執行，反之亦然，在那些情況下，循環可能比順序渲染模式花費更多的時間。**