國立虎尾科技大學 機械設計工程系

課程:協同產品設計實習

指導老師:嚴家銘教授

組別:stage1-bg1

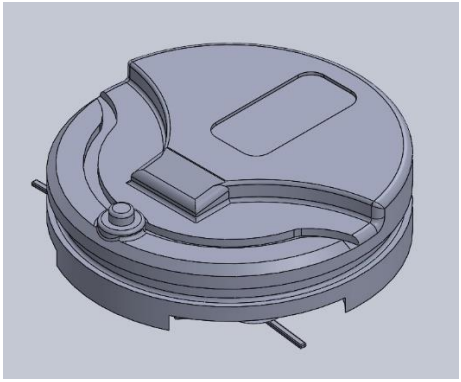
組員資料:

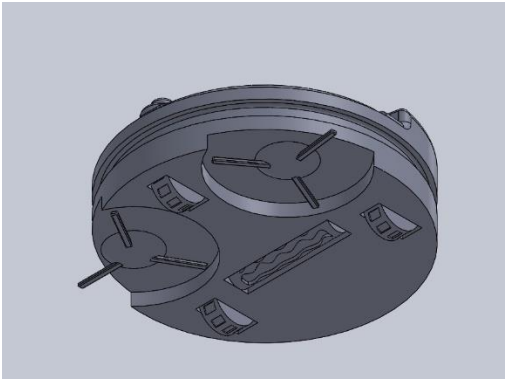
40823245	林浩瑋	組長
倉儲:https://github.com/40823245/cd2021		
網誌:https://40823245.github.io/cd2021/content/index.html		
40823251	林進益	組員
倉儲:https://github.com/40823251/cd2021		
網誌:https://40823251.github.io/cd2021/content/index.html		

小組資料:

倉儲:https://github.com/40823245/stage1-bg1 https://github.com/40823251/stage1-bg1
網誌:https://40823245.github.io/stage1-bg1/content/index.html https://40823251.github.io/stage1-bg1/content/index.html

主題:掃地機器人





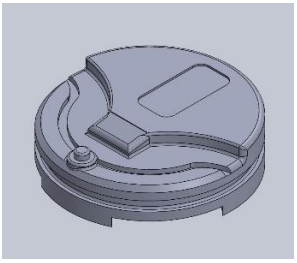
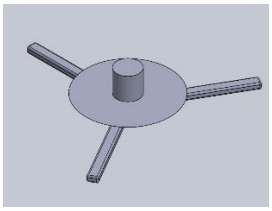
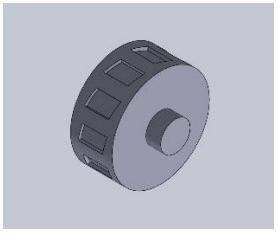
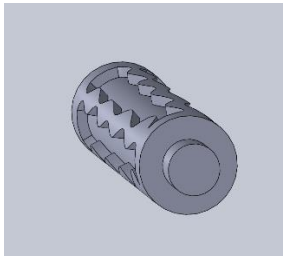
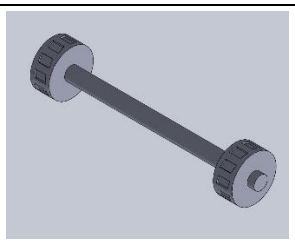
動機:

利用自動化機械結構來滿足靠人力打掃家中的需求，並實現節省時間及勞力的願景。

結構構思:

透過前方兩葉片將物品帶入後方的滾輪，滾輪則將要清理的物品吸入內部，並透過後方兩個連通輪軸帶動整台機械。

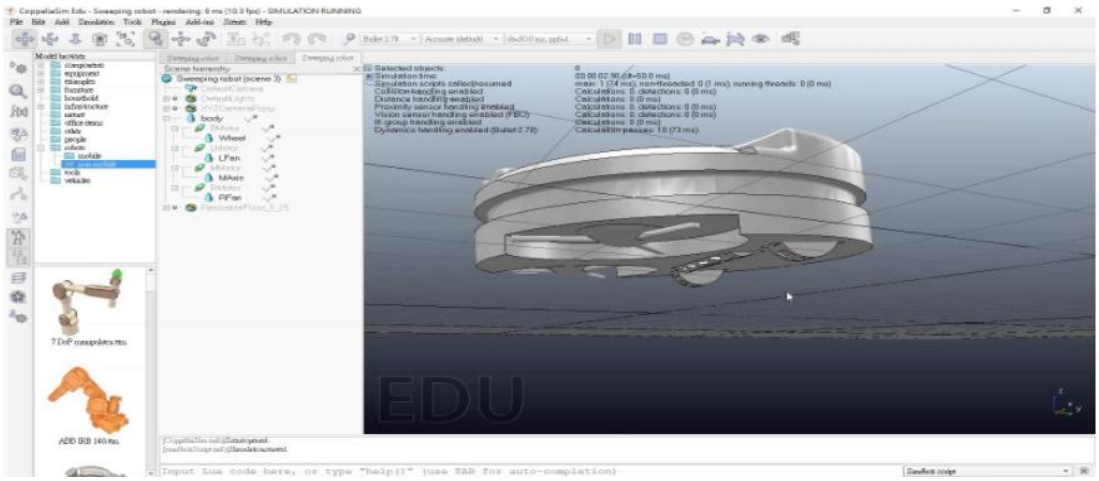
結構繪製:

零件圖	說明
	掃地機器人外部表殼 透過側邊凹槽判定有無撞到物品以改變行走方向。
	清掃地板及物品之葉片 透過三根刷子帶動物品進入內部。
	前輪 改良後新增於前方防止前傾之輪軸。
	內部滾輪 將物品進行吸入及排除之內刷。
	後輪 帶動整部機械的核心轉軸。

CoppeliaSim 模擬:

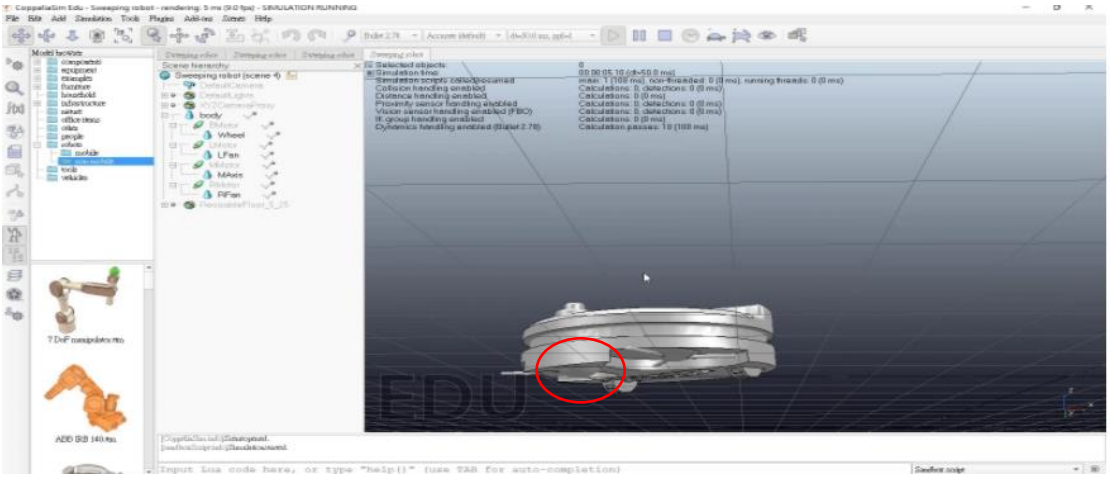
3/6 模擬測試

第一版 最初版本



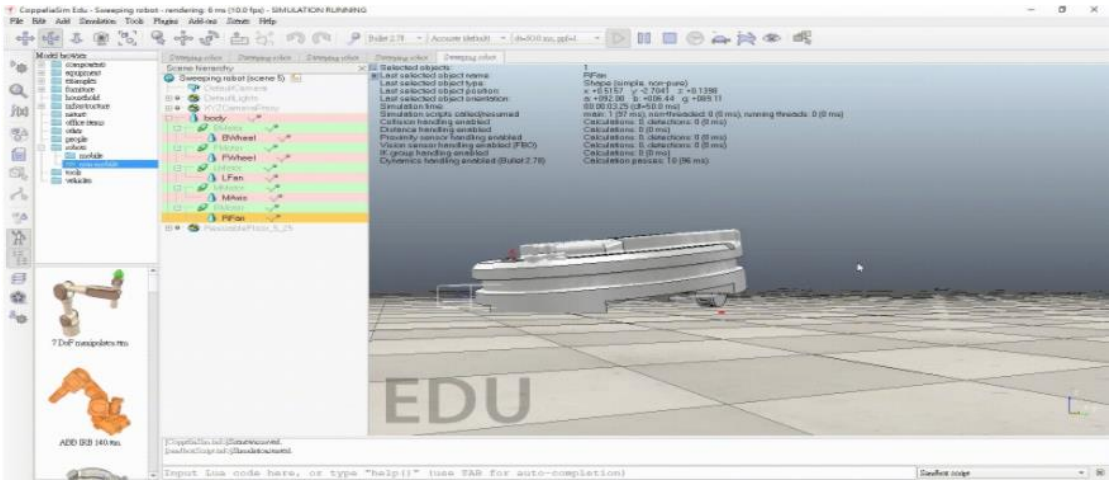
問題:由於前方無任何支撐，故模擬行走時會發生前傾的現象。

第二版 修正內容:添加圓弧物撐起前方



問題:由於前方為固定式，導致移動時仍不穩。

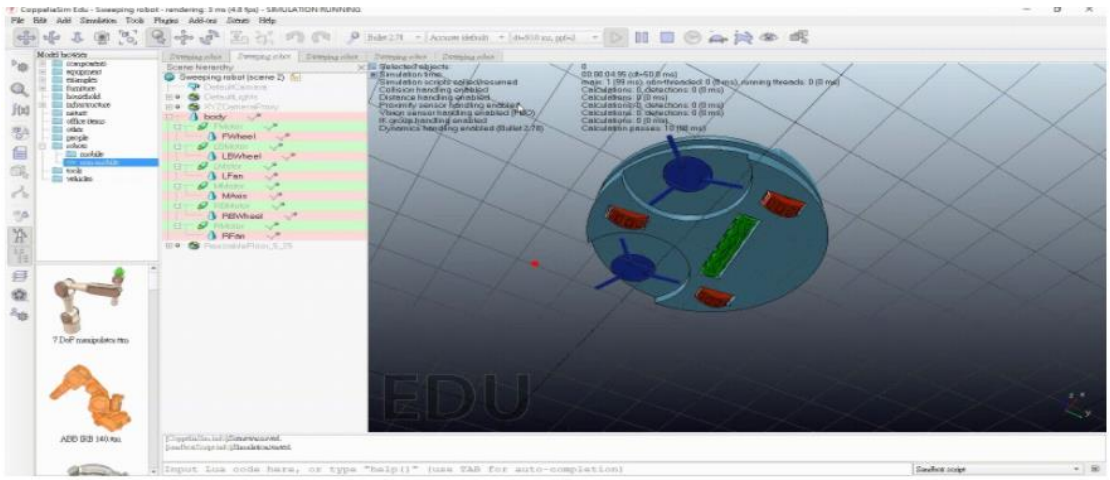
第三版 修正內容:添加前輪



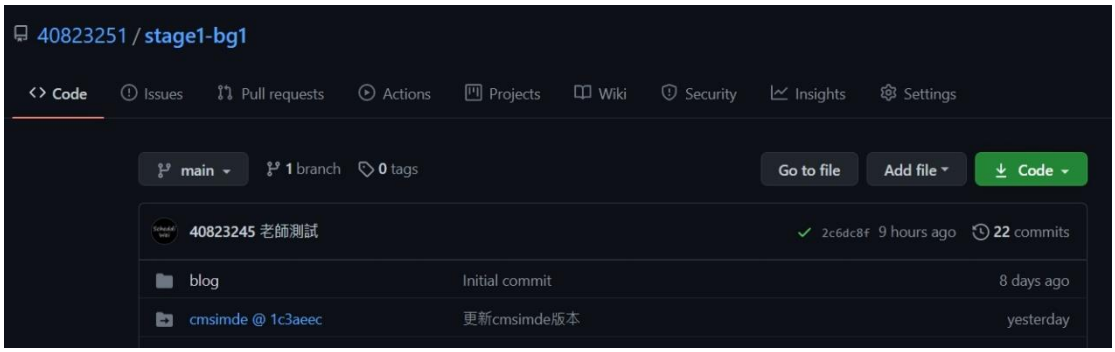
問題:由於前後馬達不一致，會有打轉的情形發生。

3/7 模擬測試

第四版 可行走(最終版)



小組網誌維護: 2021/3/13 撰寫



主旨:

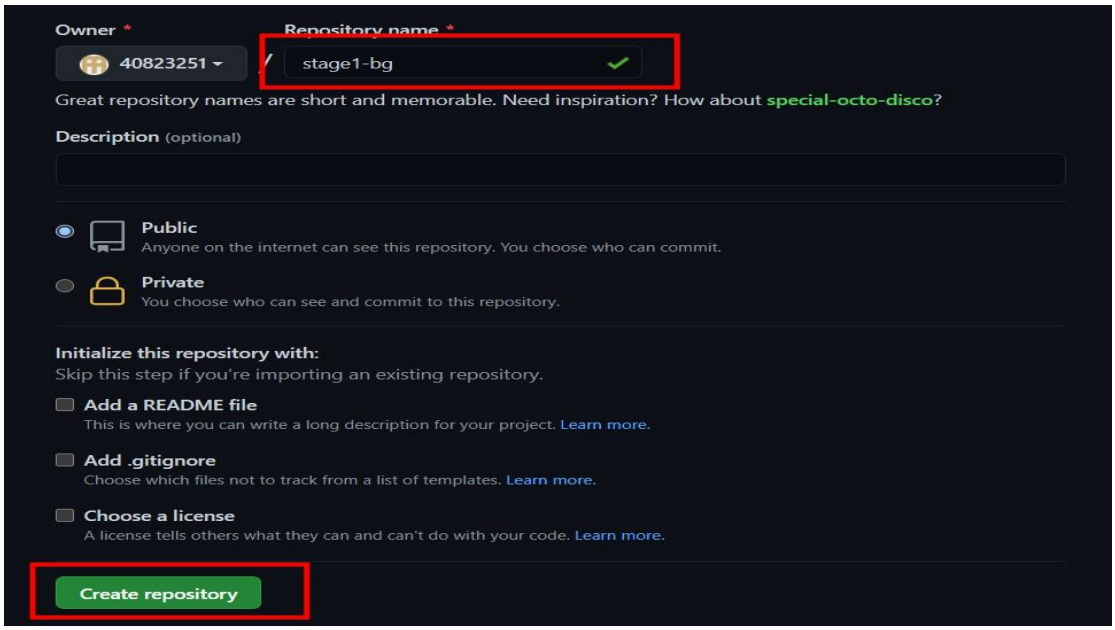
透過兩人共同維護分組倉儲，並達到互相推送及討論問題解決辦法之成效。

★本次分組遇到最艱難問題討論

時間:2021/03/12 上課時　　問題: Pull requests 時出現衝突

解決方法:

1.到自己 github 創一個空的資料夾。

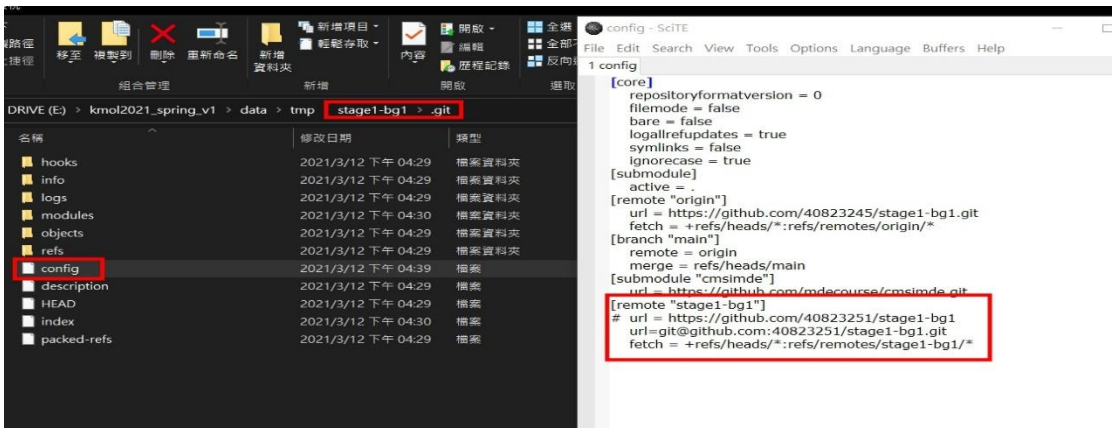


2.到小黑窗執行

git clone --recurse-submodules 你們小組的網址。

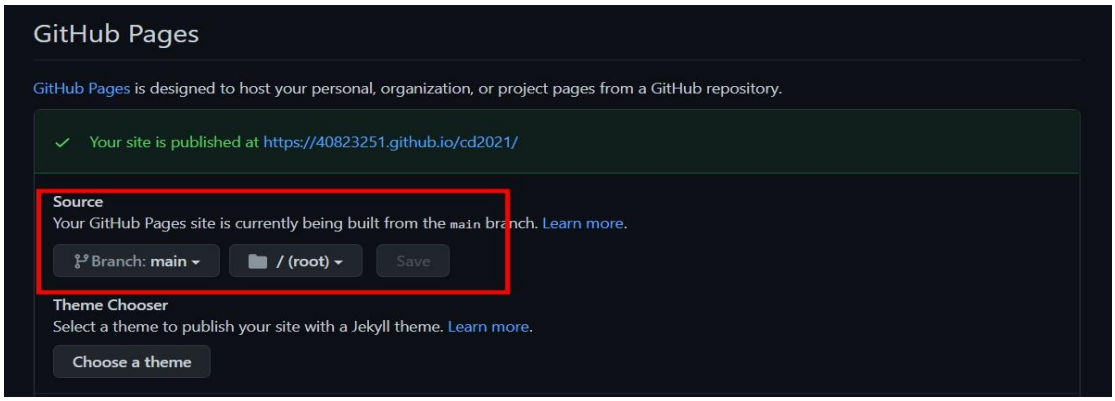
git remote add 資料夾名稱 個人創的 github 名稱.git

3.到.git 資料夾裡的 config 修改下方資訊



4. 在小黑窗中打 git push 資料夾名稱。

5. 到 github 設定將此 save



心得:

透過此次兩人協同產品讓我了解到分工的重要性，每一個人都有不同的想法透過更新網誌我們可以知道他人更改的內容及新增了些什麼，再經由互相的討論解決協同時產品發生的衝突；當遇到問題時也可以利用每次更新所產生不同的版本來找到問題的根源，接著就是要面臨 4 人小組，雖然問題一定會變多、協同也會變複雜，但只要能找出問題並寫成筆記，不僅能幫助自己也能協助其他同學。

分組的有效方案

心得:

個人認為自行選擇組員能把誤解最小化，畢竟雙方都是達成合作上的共識才會一起做協同；但就如所說的，這樣的分工效率實在太低，程式亂數分組就能立刻產生組別，希望真的能有更好的方案替代人工自動選組員。

附上 40823245 組長測試中的分組程式(未完成)

分組程式測試(待測試)

第一版本

```
#!/usr/bin/env python3
import ethercalc
import pprint
import os

proxy = 'http://192.168.0.139:8000'

os.environ['http_proxy'] = proxy
os.environ['HTTP_PROXY'] = proxy
os.environ['https_proxy'] = proxy
os.environ['HTTPS_PROXY'] = proxy

pp=pprint.PrettyPrinter(indent=4)
e = ethercalc.EtherCalc("http://192.168.0.139:8000")

pp.pprint(e.export("qv12b4lvl0hr"))
```

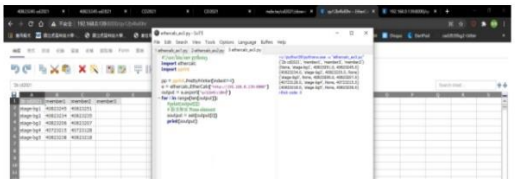
結果

第二版本

```
#!/usr/bin/env python3
import ethercalc
import pprint

pp = pprint.PrettyPrinter(indent=4)
e = ethercalc.EtherCalc("http://192.168.0.139:8000")
output = e.export("qv12b4lvl0hr")
for i in range(len(output)):
    #print(output[i])
    # 設法除掉 None element
    soutput = set(output[i])
    print(soutput)
```

結果



Pelican 採用或不採用 Leo Editor

1. 採用:

優點:分頁明確、編輯日期清楚、可當簡報使用

缺點:需每次開啟 pelican、編輯完都需透過指令請求合併且至小黑窗中 git push

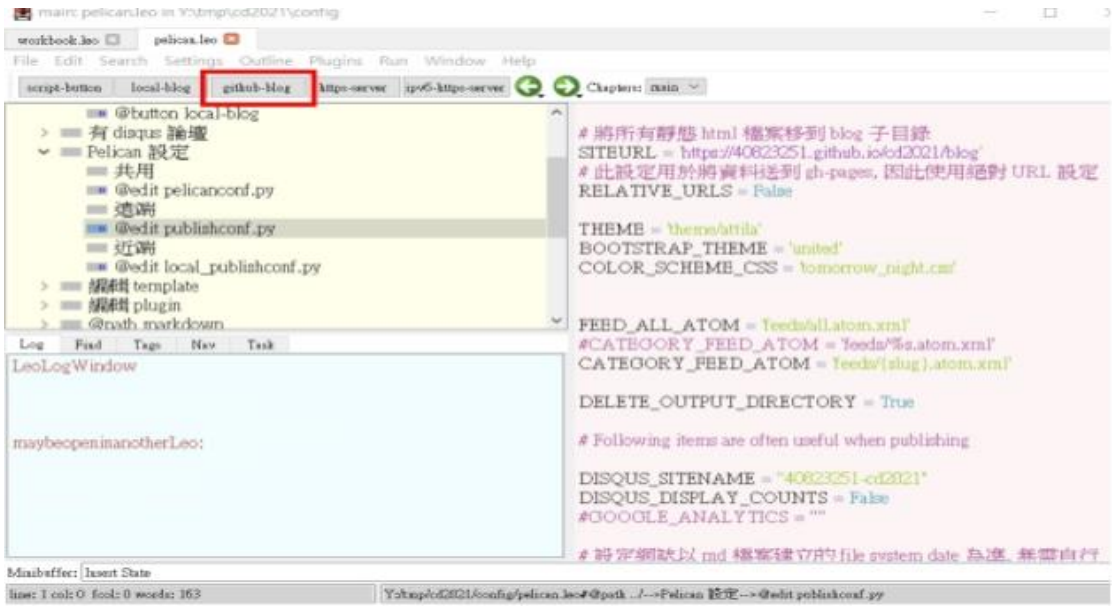
2. 不採用:

優點:直接在網誌上編輯且上傳省下兩邊都要 push 的時間

缺點:想上研究所時別人會你卻不會而降低錄取機會

心得:

就上面分析而言，我個人認為不採用的缺點是非常致命的，寧可比別人多學一點，也不希望少了些什麼；leo 上傳步驟確實多了個幾部，但操作久了自然成為你學習能力的一部份了。



參考資料

掃地機器人

1. https://i.epochtimes.com/assets/uploads/2019/05/sweep-robot_292501286-600x400.jpg

2. https://i1.kknews.cc/SIG=3fkkqlh/47120001r47oqr0p4p5q.jpg

CoppeliaSim 基本操作

1. https://www.bilibili.com/s/video/BV1yC4y1874J

版本衝突解決

1. https://www.youtube.com/watch?v=0fKyPmNI1xg