

國立虎尾科技大學

機械設計工程系

協同產品設計實習 40623224 期末報告

手足球系統模擬

Hand soccer system simulation

學生：

設計二乙 40623224 鐘偉哲

指導教授：嚴家銘

2019.06.09

摘要

我是負責繪圖與送球機構繪製的部分，繪畫了兩版的送球機構，其中第二版的送球機構進行的非常成功，以及拍攝影片及放下載連結，還有練習 vrep 與程式開發。

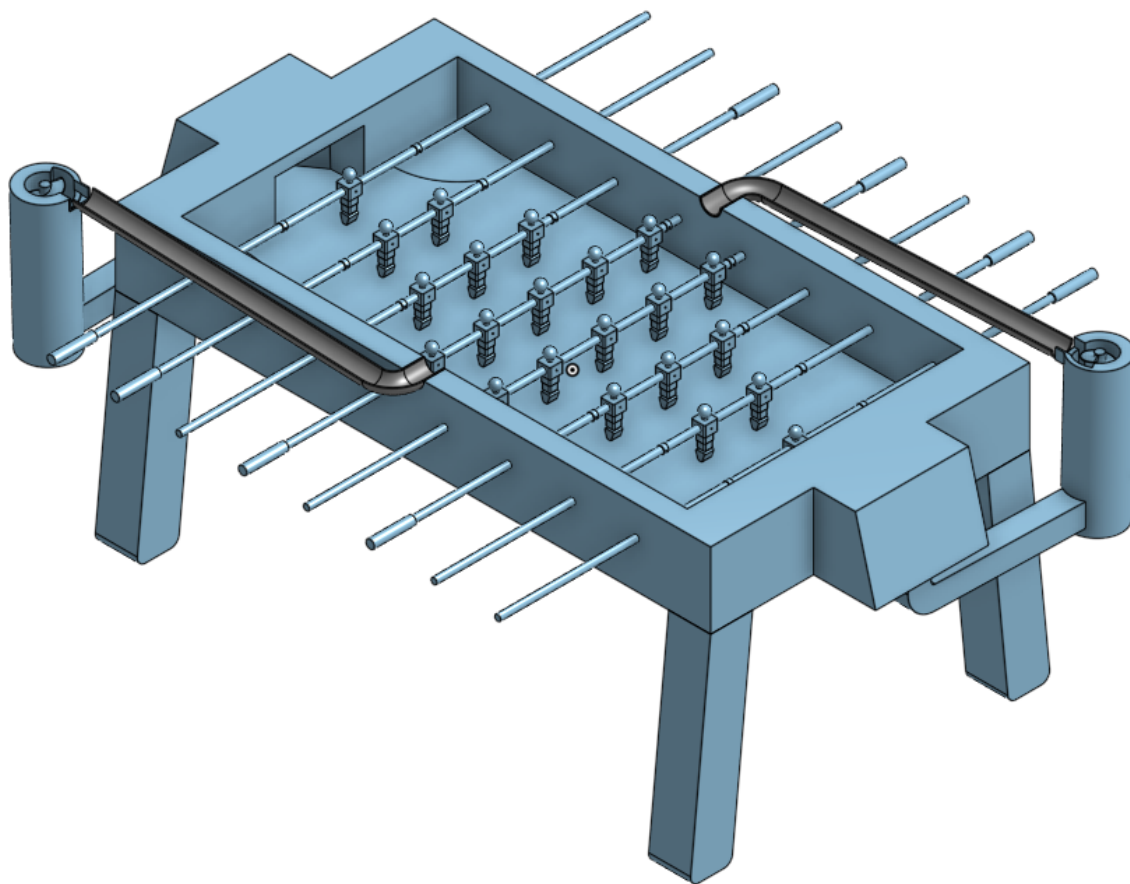


圖 0.1: Hand soccer table combination

目錄

摘要	i
目錄	ii
表目錄	iv
圖目錄	v
第一章 前言	1
第二章 Onshape	2
2.1 Hand soccer table 手足球台	2
2.2 Football player 手足球員	2
2.3 Hand football pole A-D 手足球桿 A-D	2
2.4 Collar 套筒	2
2.5 Football table corner 足球台桌角	2
2.6 Auger 螺旋桿	2
2.7 Ball sending mechanism 送球機構	3
第三章 Part_size_drawing	12
3.1 part_size_drawing 零件尺寸圖	12
第四章 Combination	21
4.1 Combination 組合圖	21
第五章 Explosion	28
5.1 Explosion 爆炸圖	28
第六章 Operation_Demo 操作演示	31
6.1 Operation_Demo	31
第七章 Vrep_simulation Vrep 模擬	32
7.1 Vrep_simulation Vrep 模擬	32
第八章 Lesson_notes 課程筆記	36
8.1 Lesson_notes	36

第九章	Download_link 下載連結	38
9.1	Download_link	38
第十章	影片連結	39

表目錄

圖目錄

圖 0.1	Hand soccer table combination	i
圖 1.1	Ball sending mechanism	1
圖 2.1	Hand soccer table	4
圖 2.2	Football player	5
圖 2.3	Hand football pole A	6
圖 2.4	Hand football pole B	6
圖 2.5	Hand football pole C	7
圖 2.6	Hand football pole D	7
圖 2.7	Collar	8
圖 2.8	Football table corner	9
圖 2.9	Auger	10
圖 2.10	ball sending mechanism	11
圖 3.1	Hand soccer table Drawings	13
圖 3.2	Football player Drawings	14
圖 3.3	Hand football pole A-B Drawings	15
圖 3.4	Hand football pole C-D Drawings	16
圖 3.5	Collar Drawings	17
圖 3.6	Football table corner Drawings	18
圖 3.7	Auger Drawings	19
圖 3.8	ball sending mechanism Drawings	20
圖 4.1	combination	22
圖 4.2	Partial combination A	23
圖 4.3	Partial combination B	24
圖 4.4	Partial combination C	25

圖 4.5	Partial combination D	26
圖 4.6	Combined engineering drawing	27
圖 5.1	explosion	29
圖 5.2	Explosion engineering drawing	30
圖 7.1	human to human Program1	33
圖 7.2	human to human Program2	34
圖 7.3	human to human Program3	35

第一章 前言

利用 onshape 這種開放式的繪圖軟體來繪製手足球機構，並且增加送球功能，使機台能再踢進球門後經由送球機構回到足球場上得以繼續踢球。

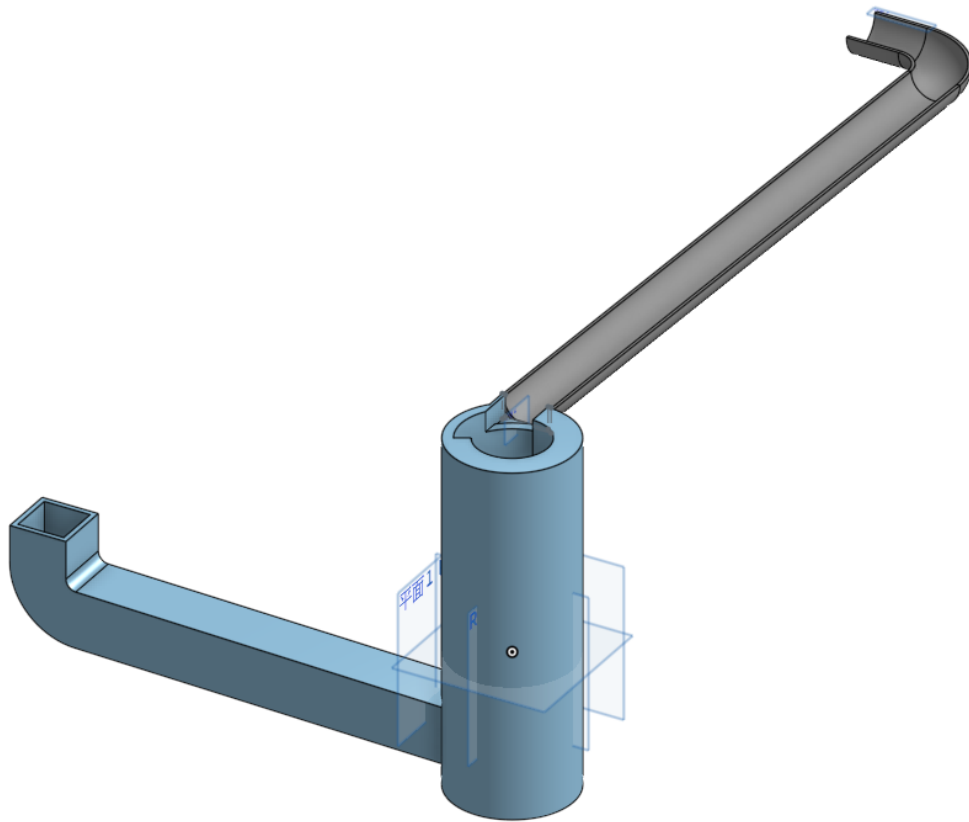


圖 1.1: Ball sending mechanism

第二章 Onshape

手足球台各部位介紹，並且下方可開啟連結進行下載動作。

2.1 Hand soccer table 手足球台

手足球台是與手足球桿、送球機構等等連接的主要零件，是用來進行足球比賽的場地。

2.2 Football player 手足球員

手足球員是與各種手足球桿結合並插入銷，以手足球桿旋轉以及移動讓手足球員可以踢球。

2.3 Hand football pole A-D 手足球桿 A-D

手足球桿結合手足球員以及套筒插入手足球台中利用插銷固定足球員以及套筒，套筒可以用來固定手足球桿移動的限制的距離，而手足球桿可轉動 + 移動。

2.4 Collar 套筒

套筒可以用來固定手足球桿移動的限制的距離，用插銷固定套筒。

2.5 Football table corner 足球台桌角

足球台桌角主要是用來支撐整個手足球台，並且讓手足球台可以進行送球機構。

2.6 Auger 螺旋桿

螺旋桿主要用來進行送球機構的模擬，利用自旋的功能讓手足球送回台面得以續行踢球。

2.7 Ball sending mechanism 送球機構

送球機構主要的目的是讓球射進球門後利用軌道滾至圓柱，再由螺旋桿使球體向上升至底端，再由滾道滾回至台面。

Hand soccer table 圖片連結

Football player 圖片連結

Hand football pole A 圖片連結

Hand football pole B 圖片連結

Hand football pole C 圖片連結

Hand football pole D 圖片連結

Collar 圖片連結

Football table corner 圖片連結

Auger 圖片連結

ball sending mechanism 圖片連結

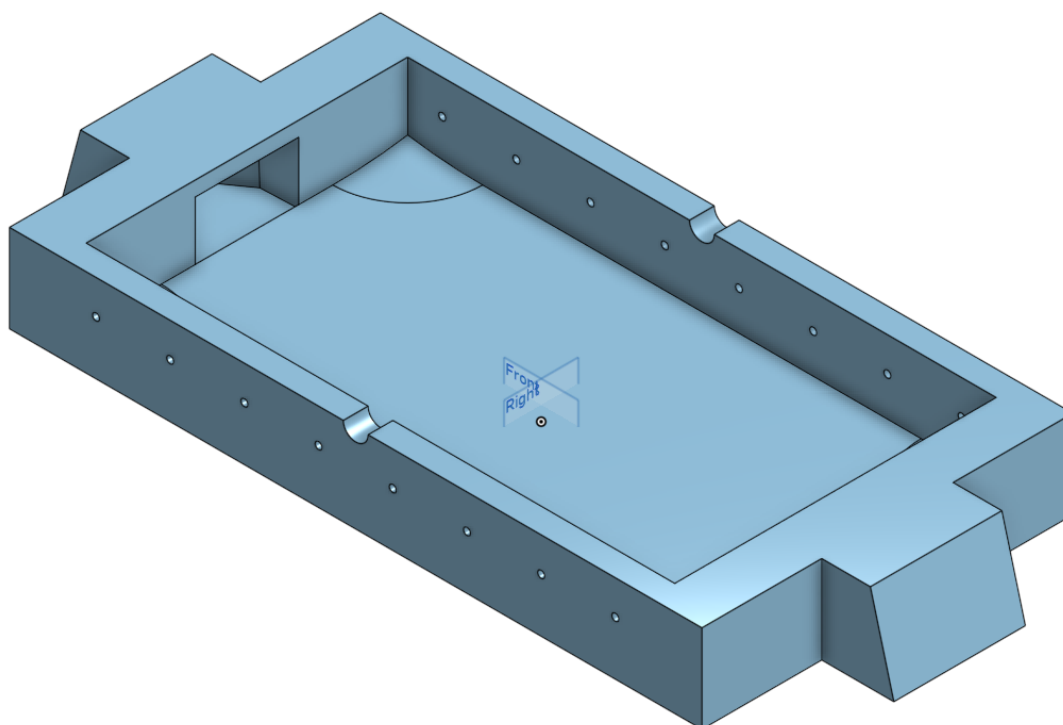


圖 2.1: Hand soccer table

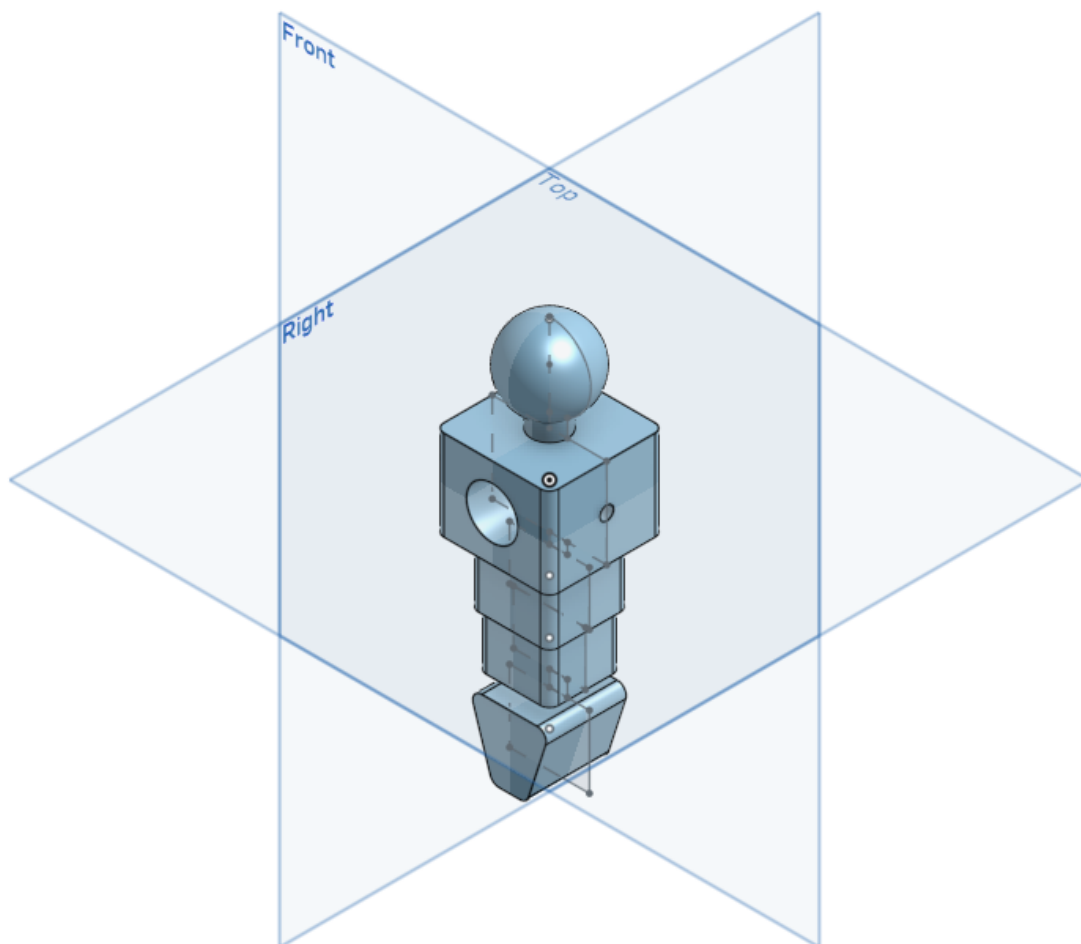


圖 2.2: Football player

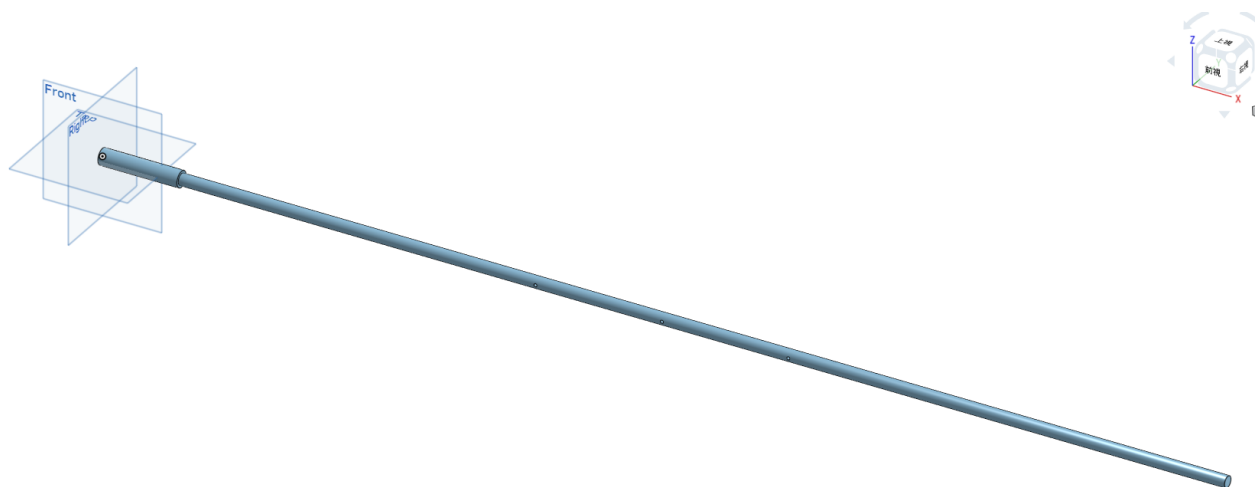


圖 2.3: Hand football pole A

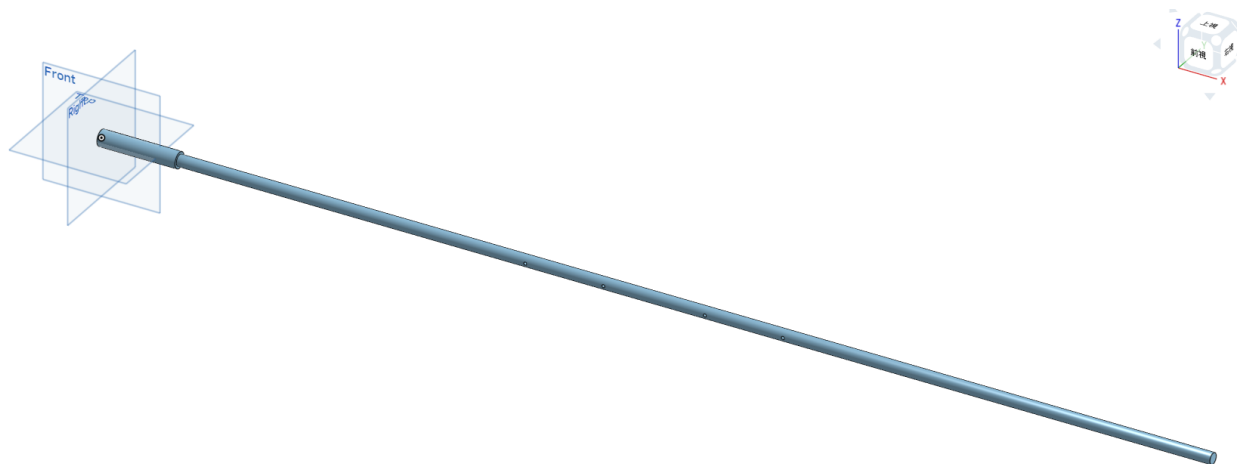


圖 2.4: Hand football pole B

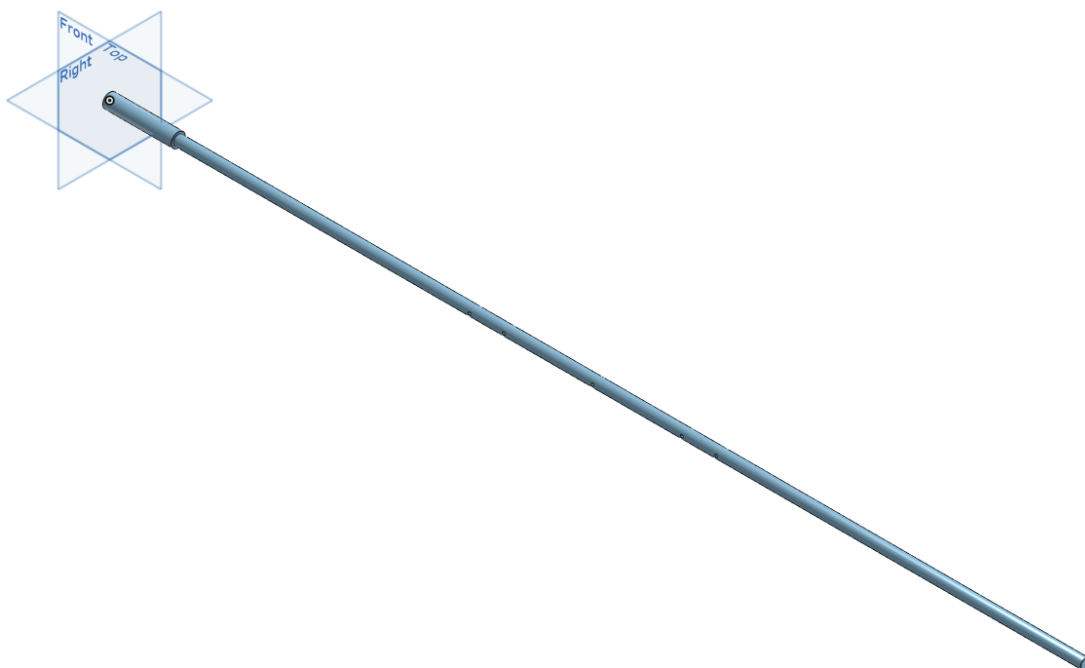


圖 2.5: Hand football pole C

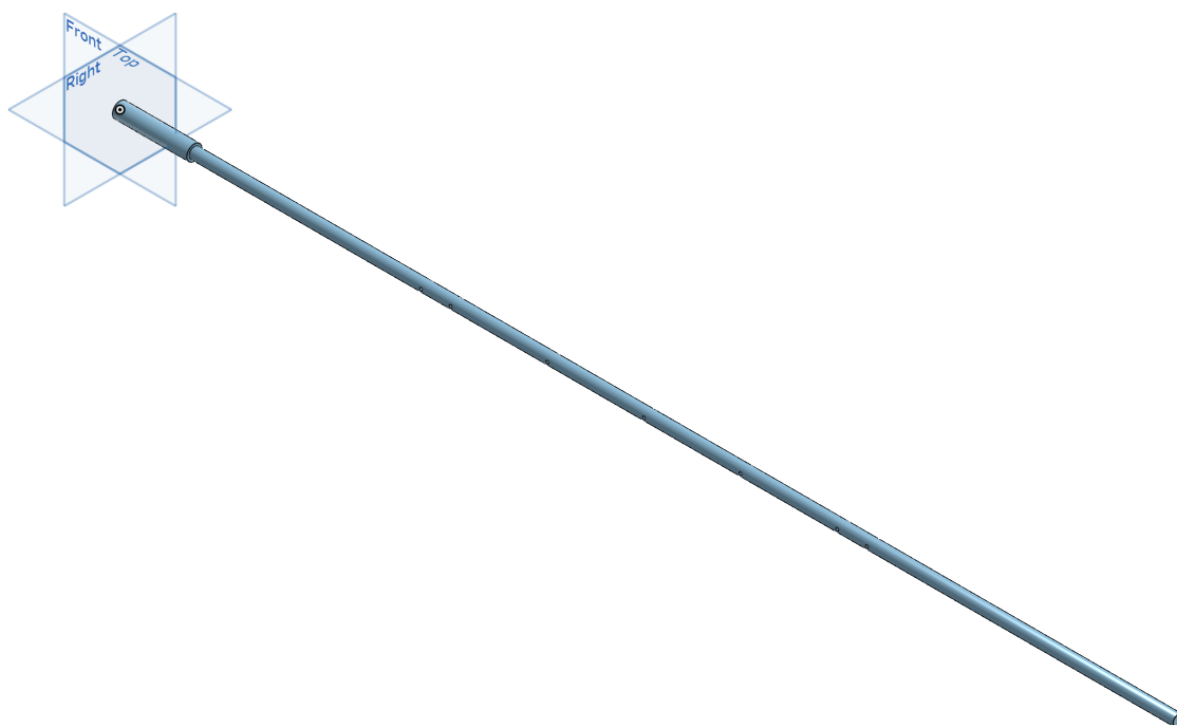


圖 2.6: Hand football pole D

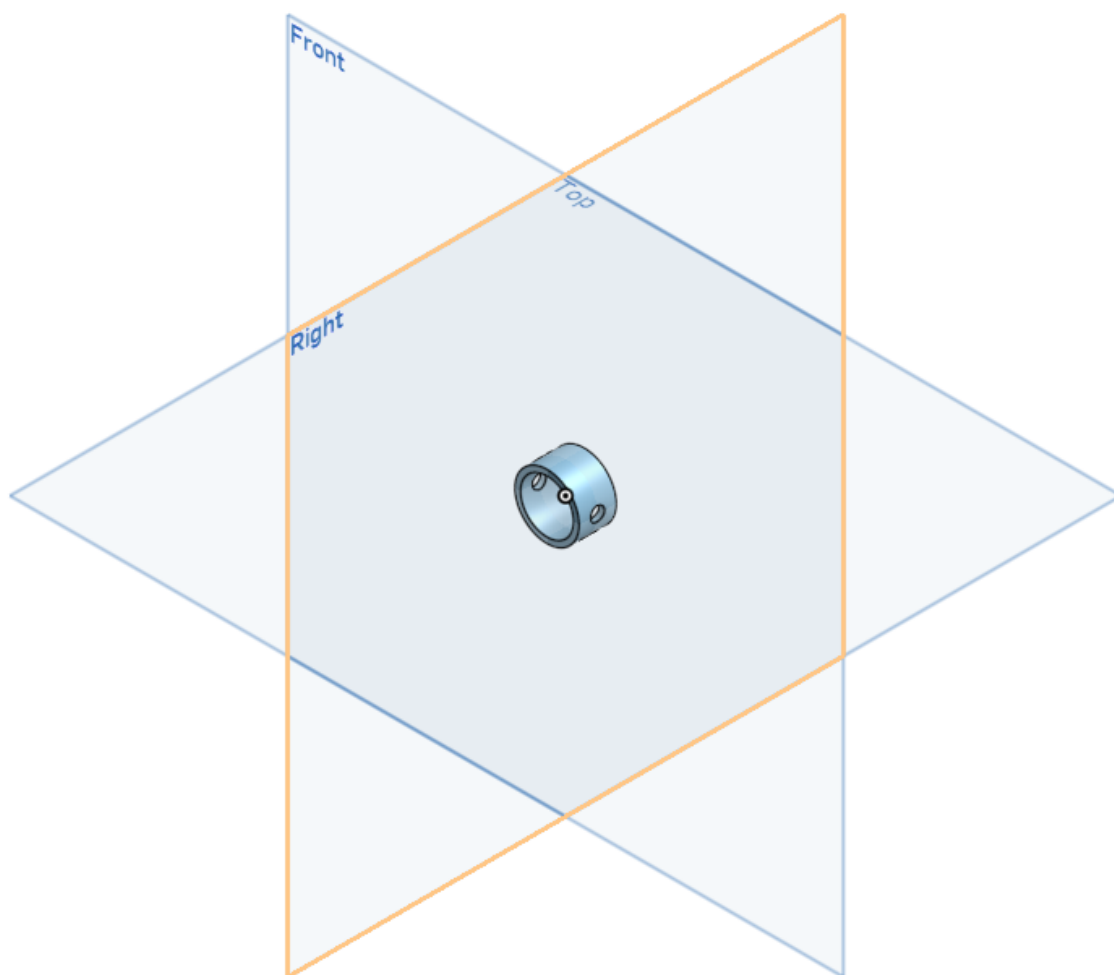


圖 2.7: Collar

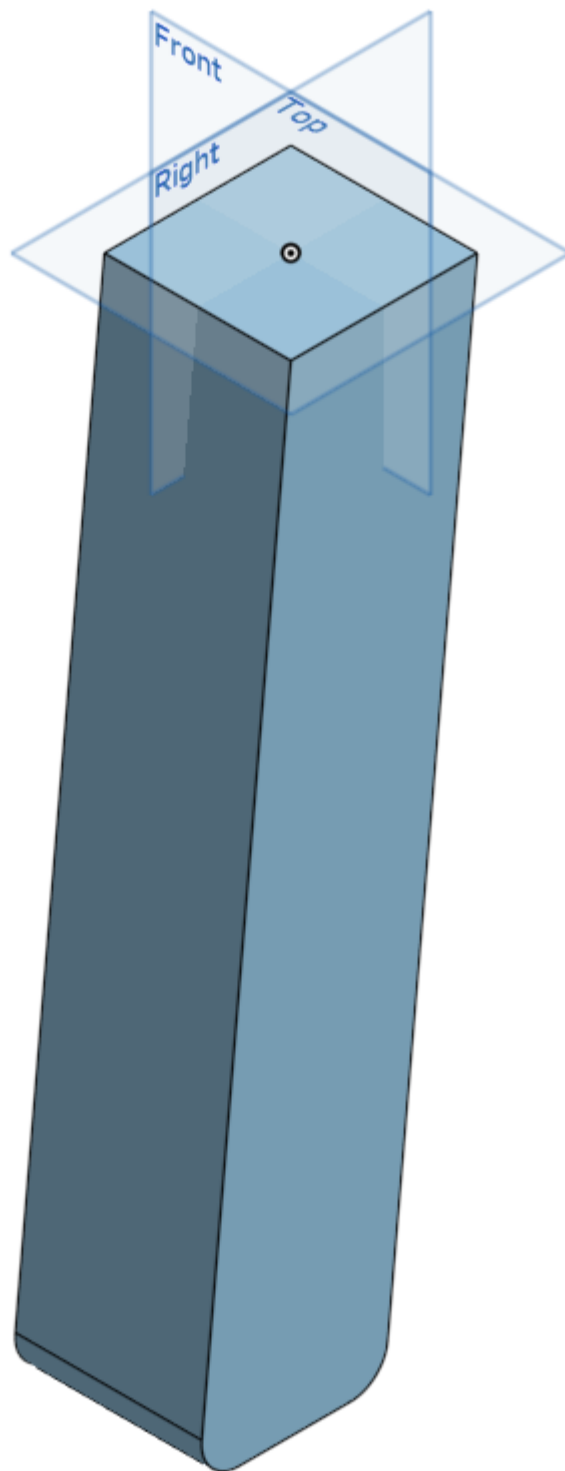


圖 2.8: Football table corner

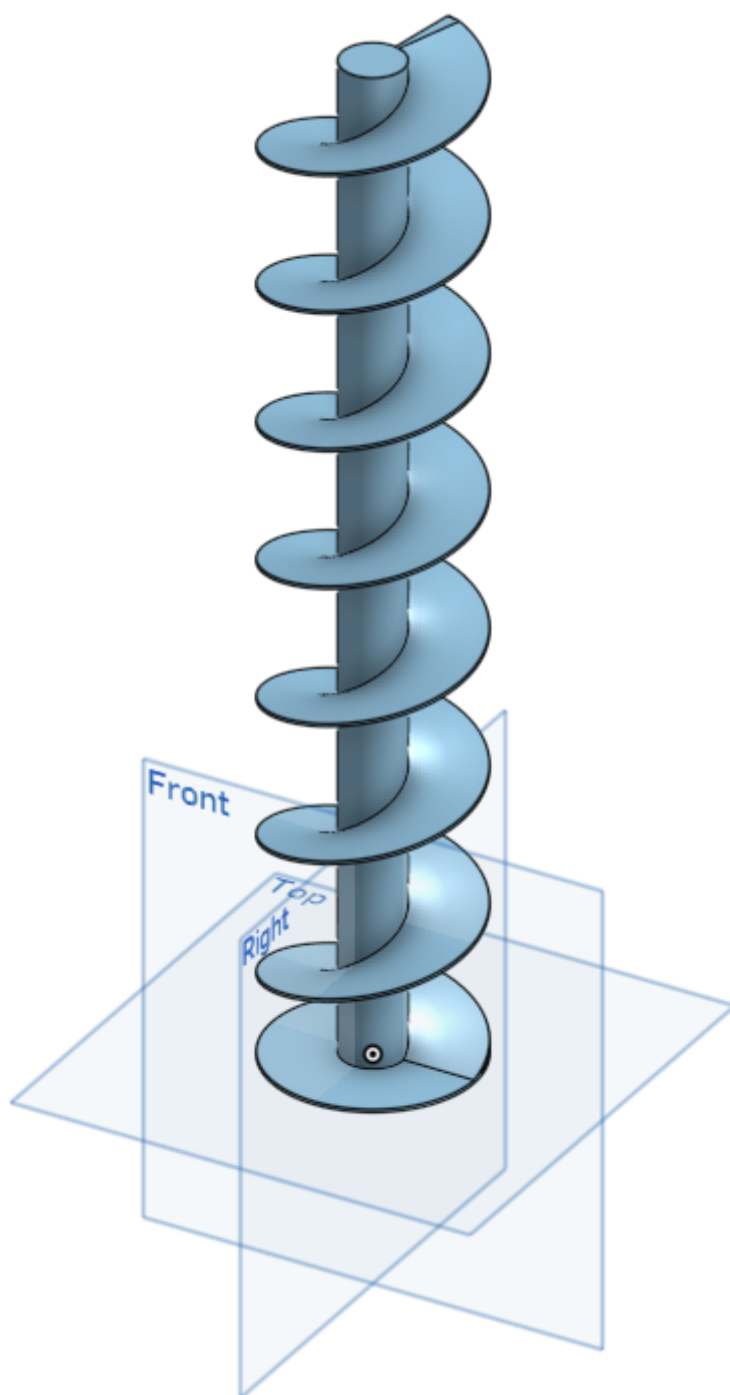


圖 2.9: Auger

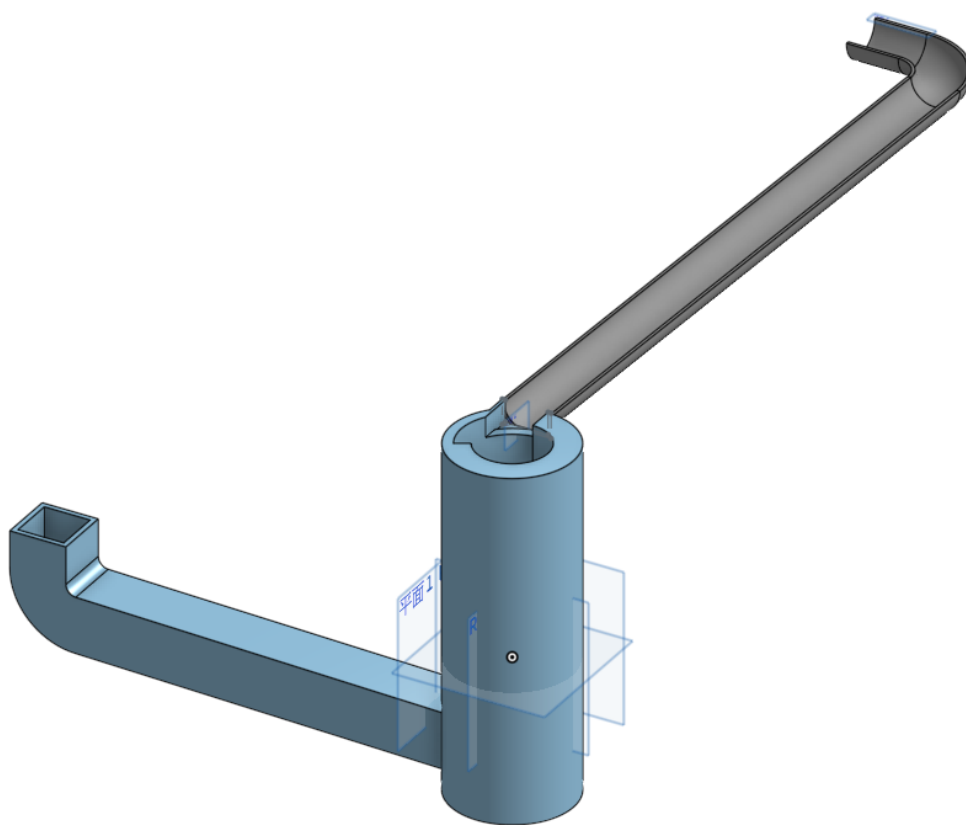


圖 2.10: ball sending mechanism

第三章 Part_size_drawing

3.1 part_size_drawing 零件尺寸圖

觀看各部位之零件尺寸，並且下方可開啟連結進行下載動作。

Hand soccer table Drawings [圖片連結](#)

Football player Drawings [圖片連結](#)

Hand football pole A-B Drawings [圖片連結](#)

Hand football pole C-D Drawings [圖片連結](#)

Collar Drawings [圖片連結](#)

Football table corner Drawings [圖片連結](#)

Auger Drawings [圖片連結](#)

ball sending mechanism Drawings [圖片連結](#)

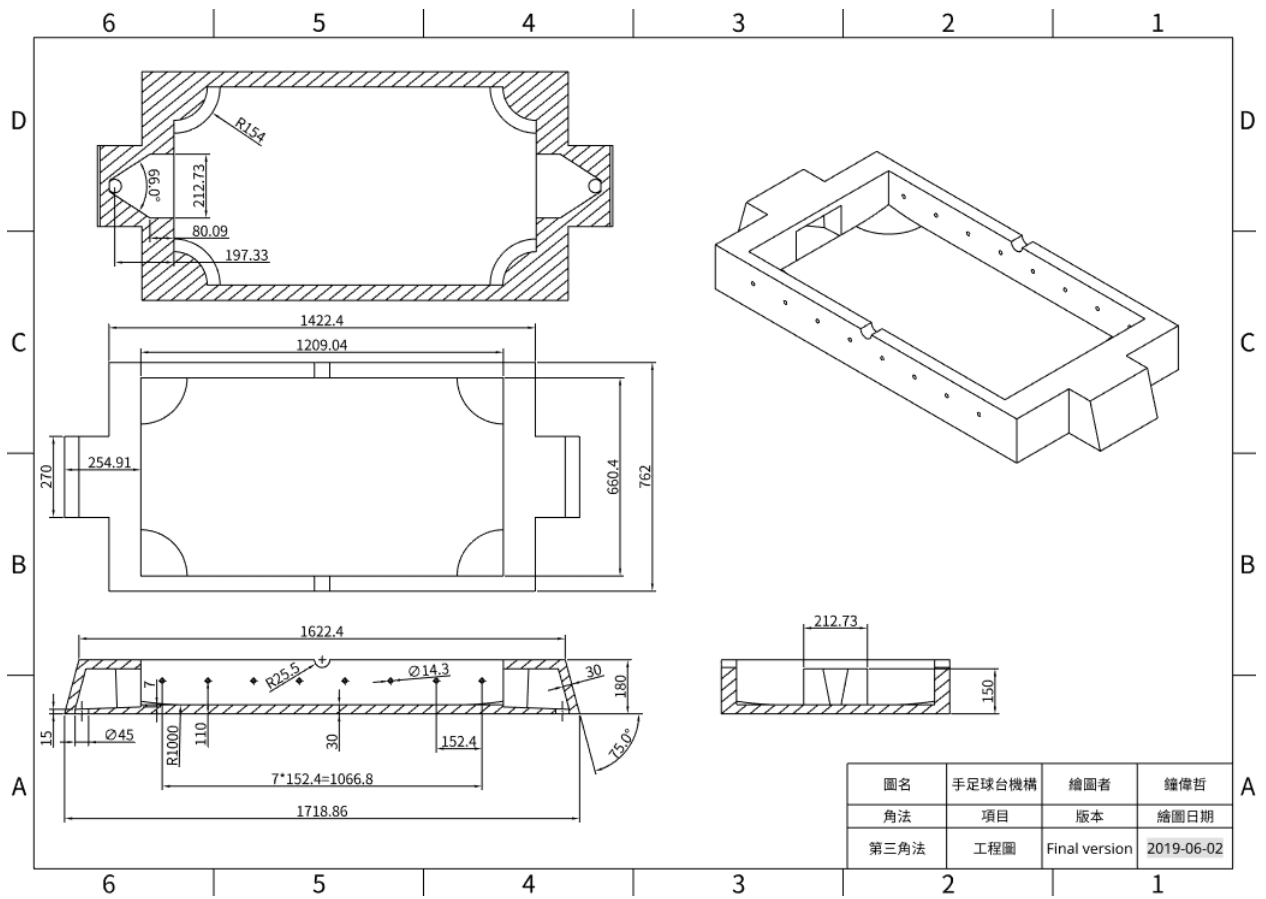


圖 3.1: Hand soccer table Drawings

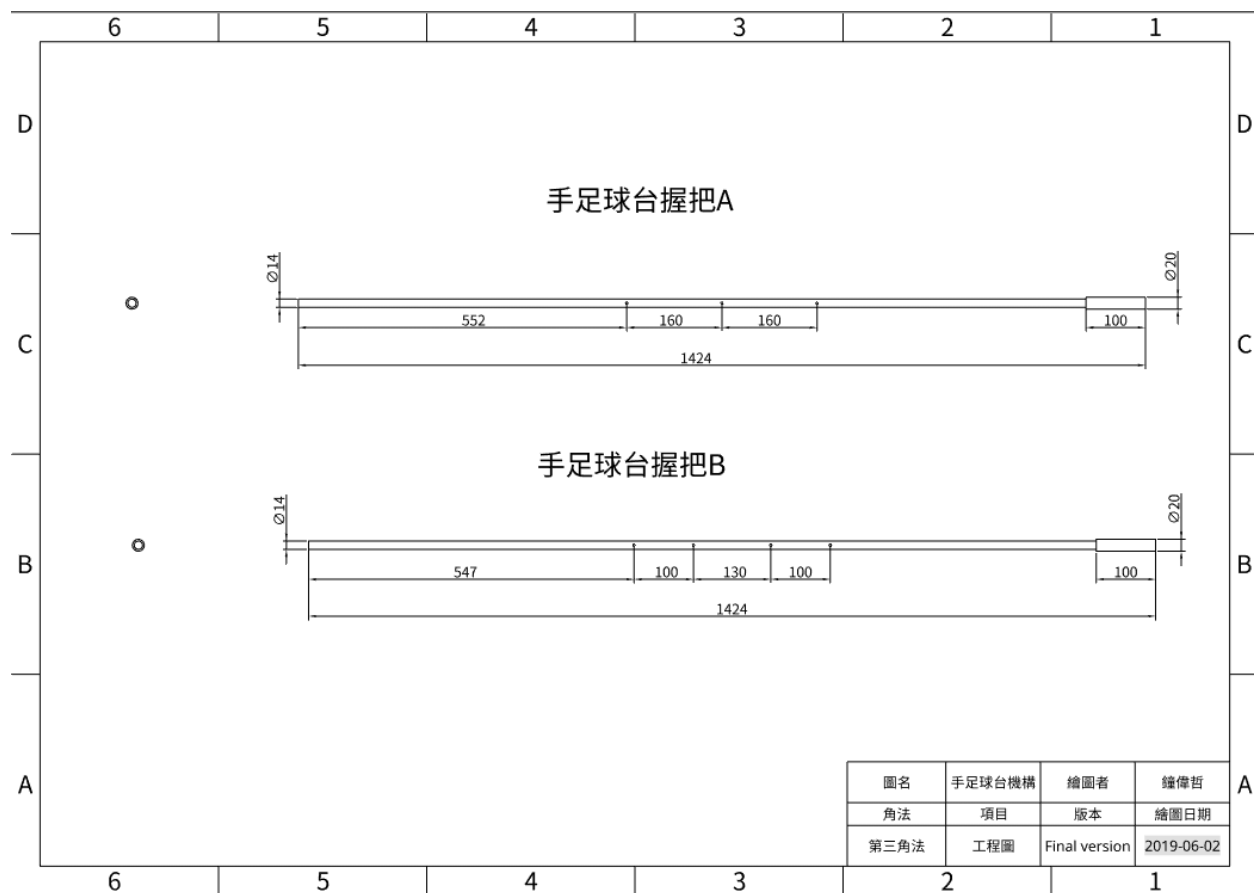


圖 3.3: Hand football pole A-B Drawings

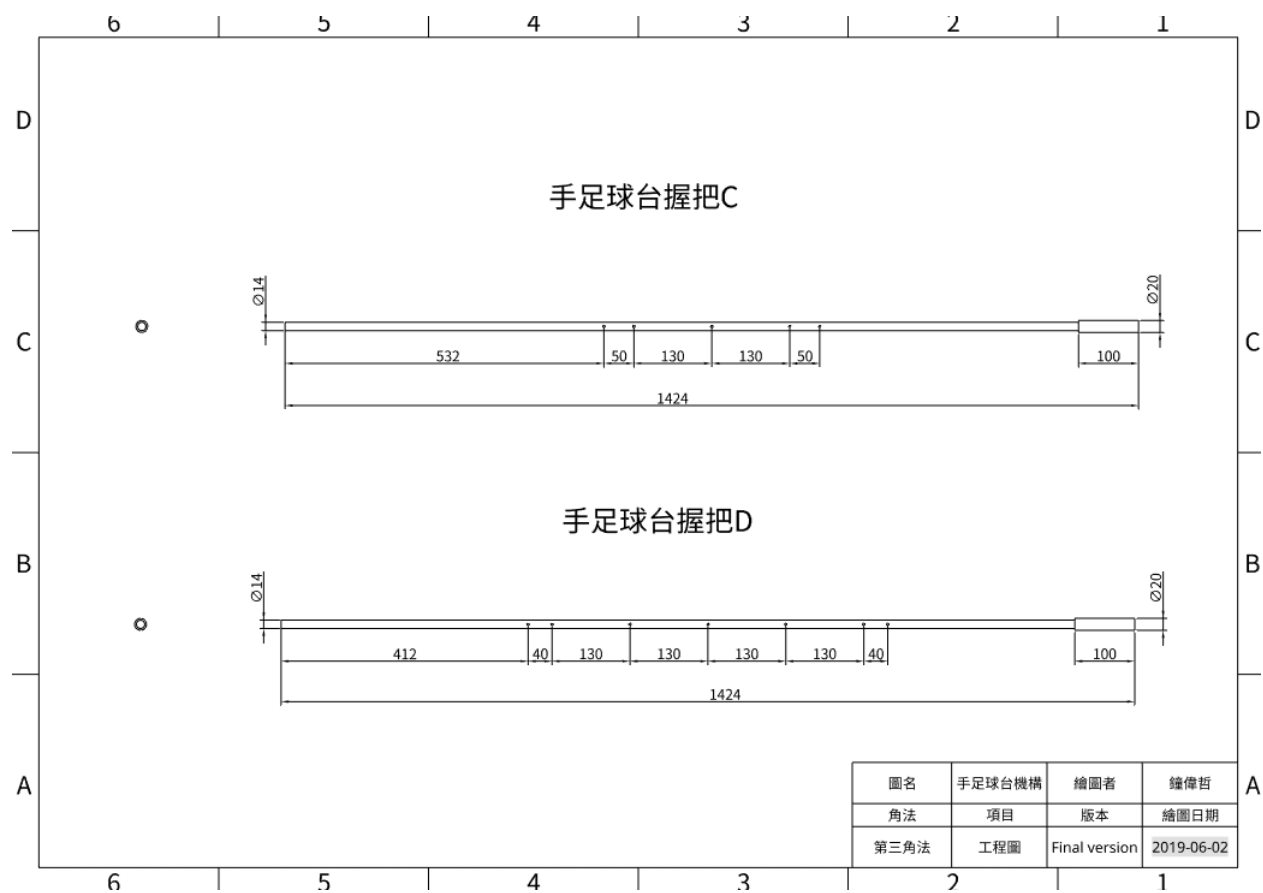


圖 3.4: Hand football pole C-D Drawings

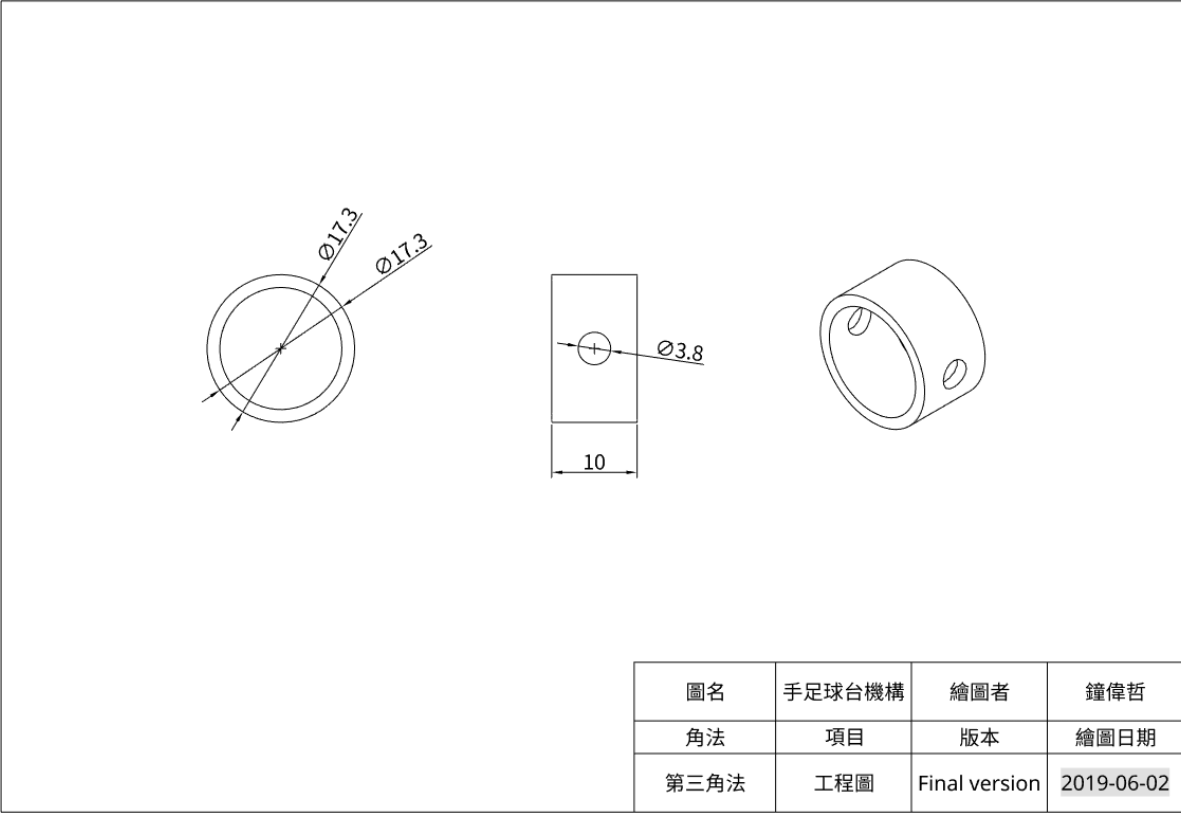


圖 3.5: Collar Drawings

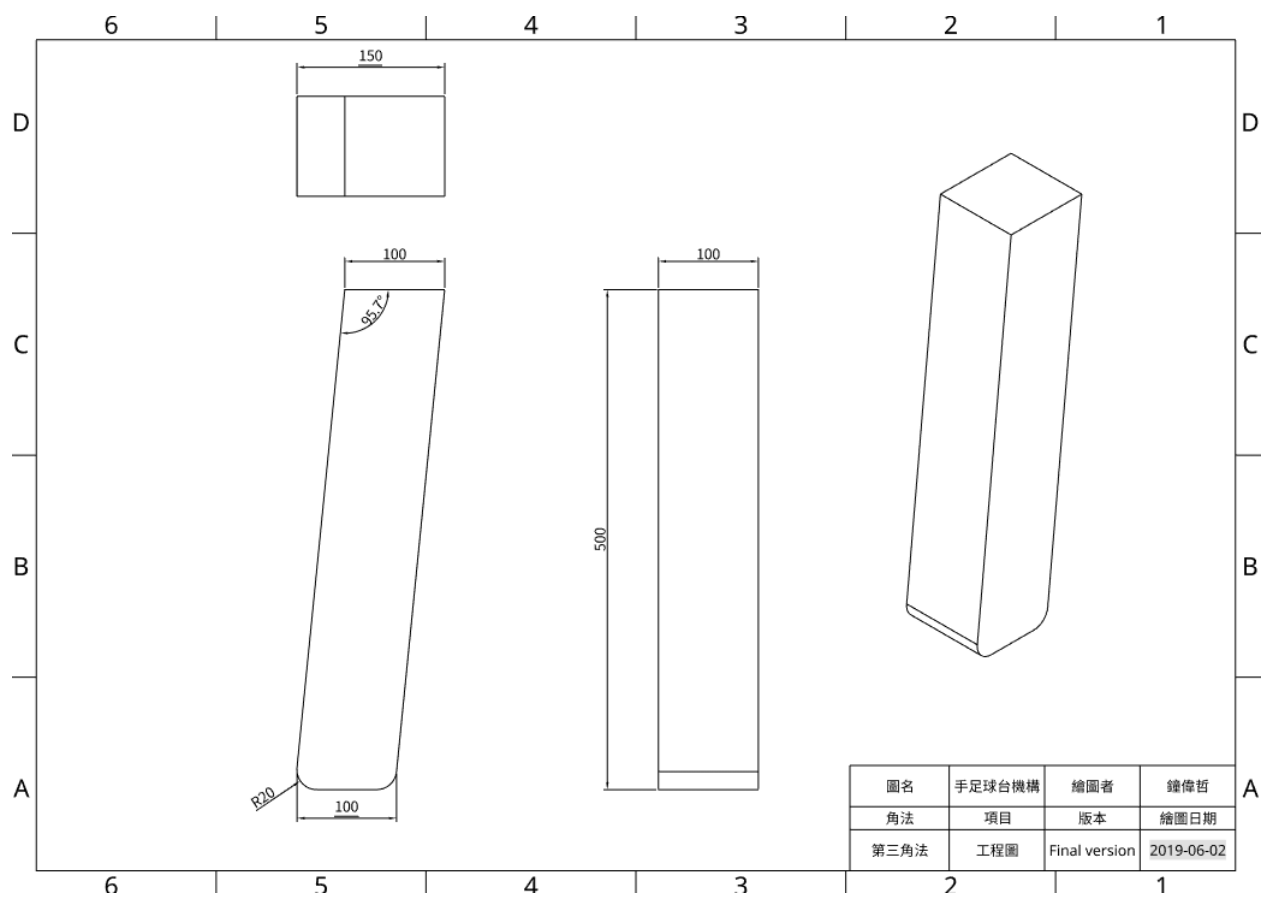


圖 3.6: Football table corner Drawings

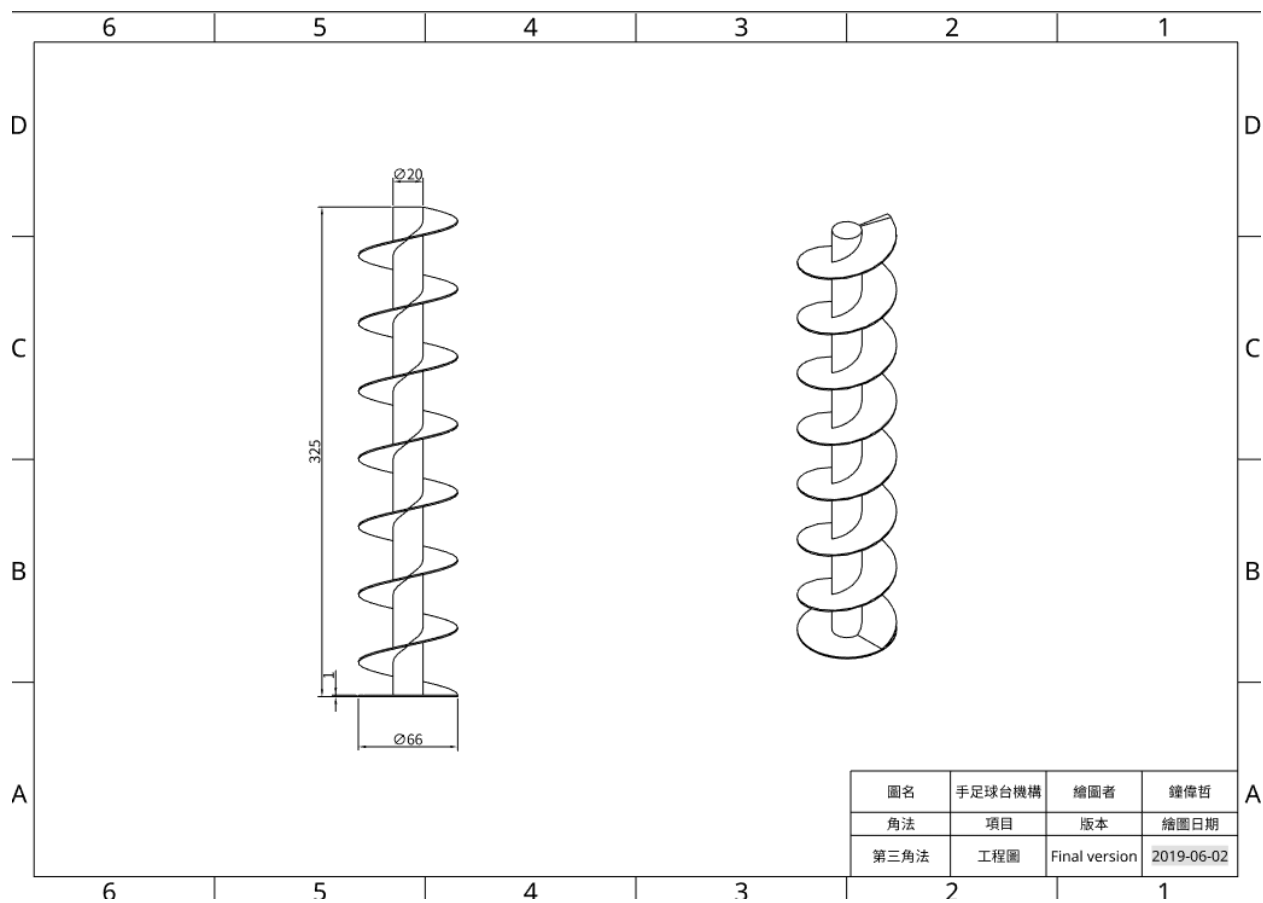


圖 3.7: Auger Drawings

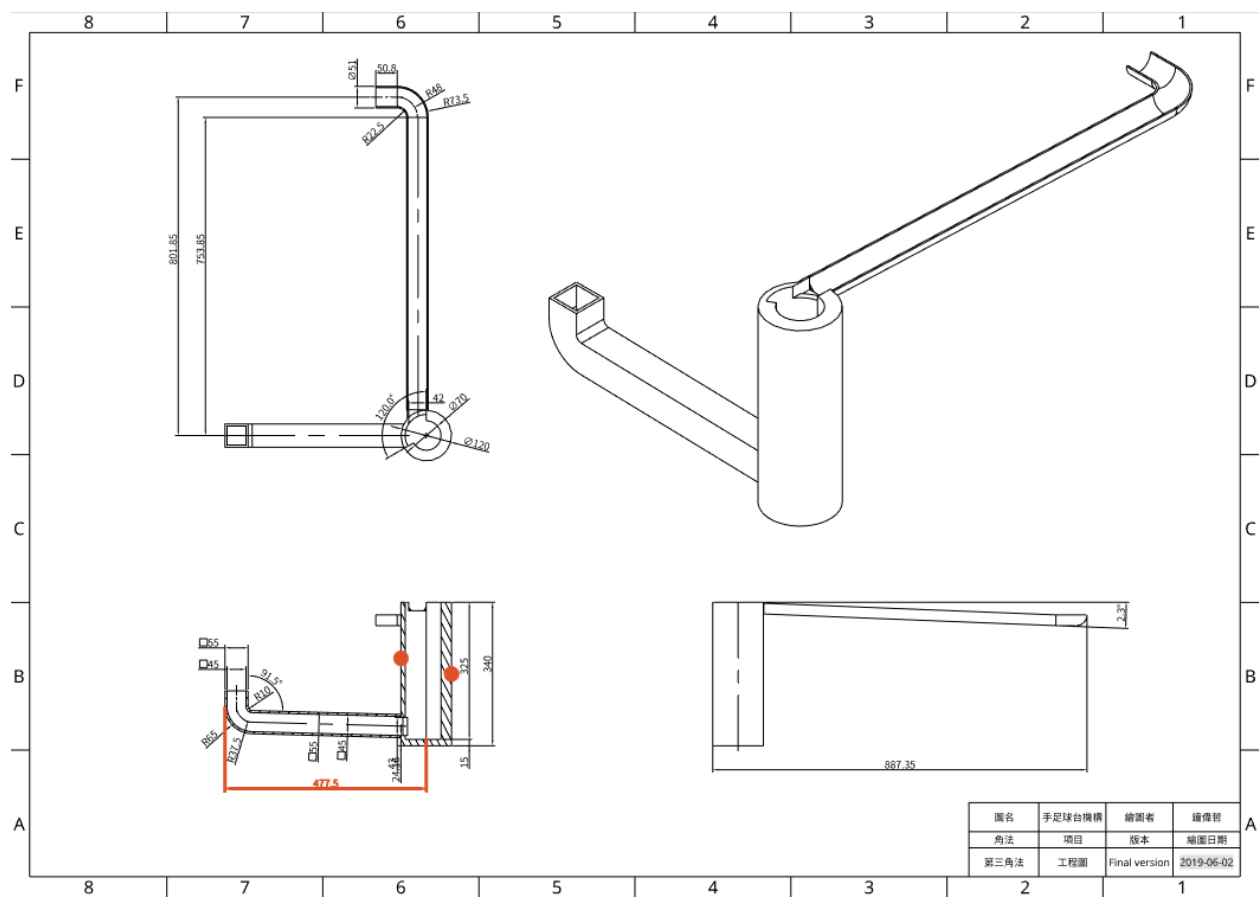


圖 3.8: ball sending mechanism Drawings

第四章 Combination

4.1 Combination 組合圖

觀看組合圖的模樣，並且下方可開啟連結進行下載動作。

combination 圖片連結

Partial combination A 圖片連結

Partial combination B 圖片連結

Partial combination C 圖片連結

Partial combination D 圖片連結

Combined engineering drawing 圖片連結

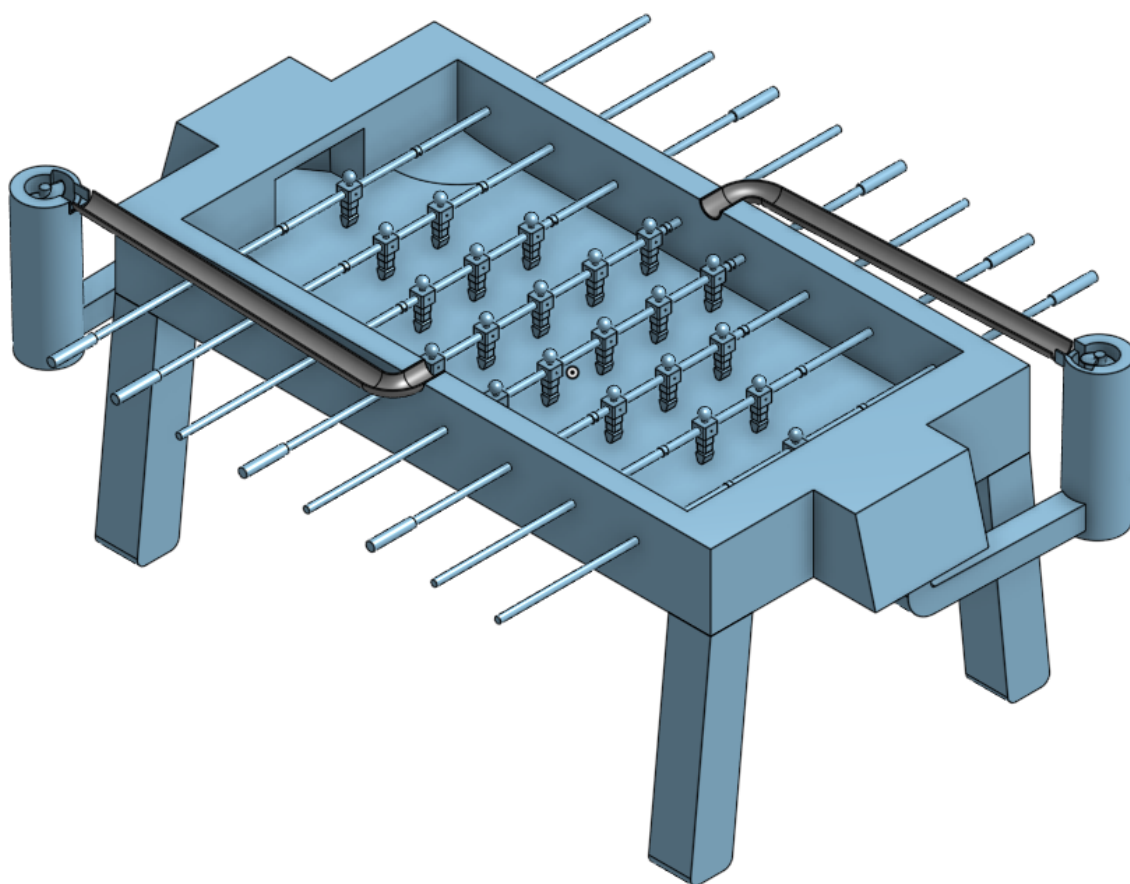


圖 4.1: combination

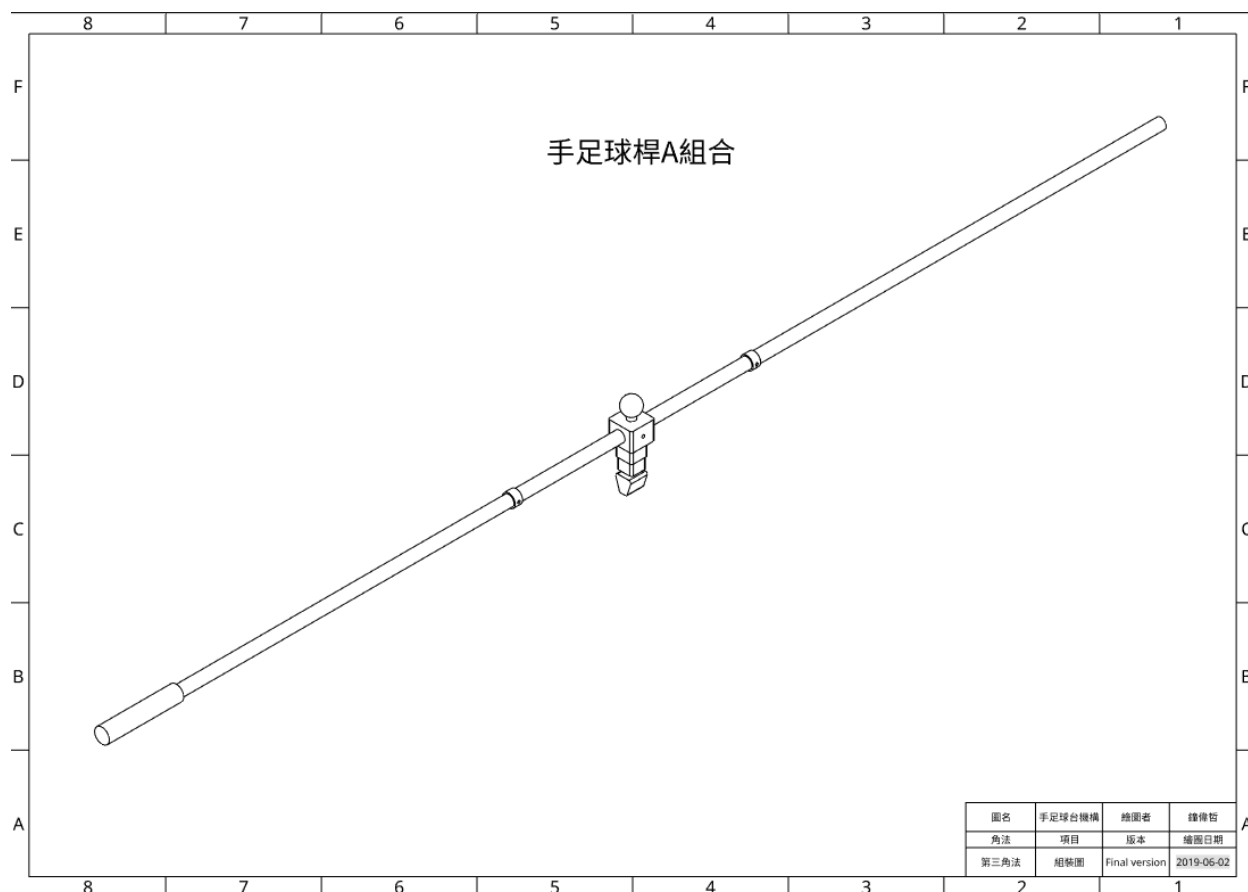


圖 4.2: Partial combination A

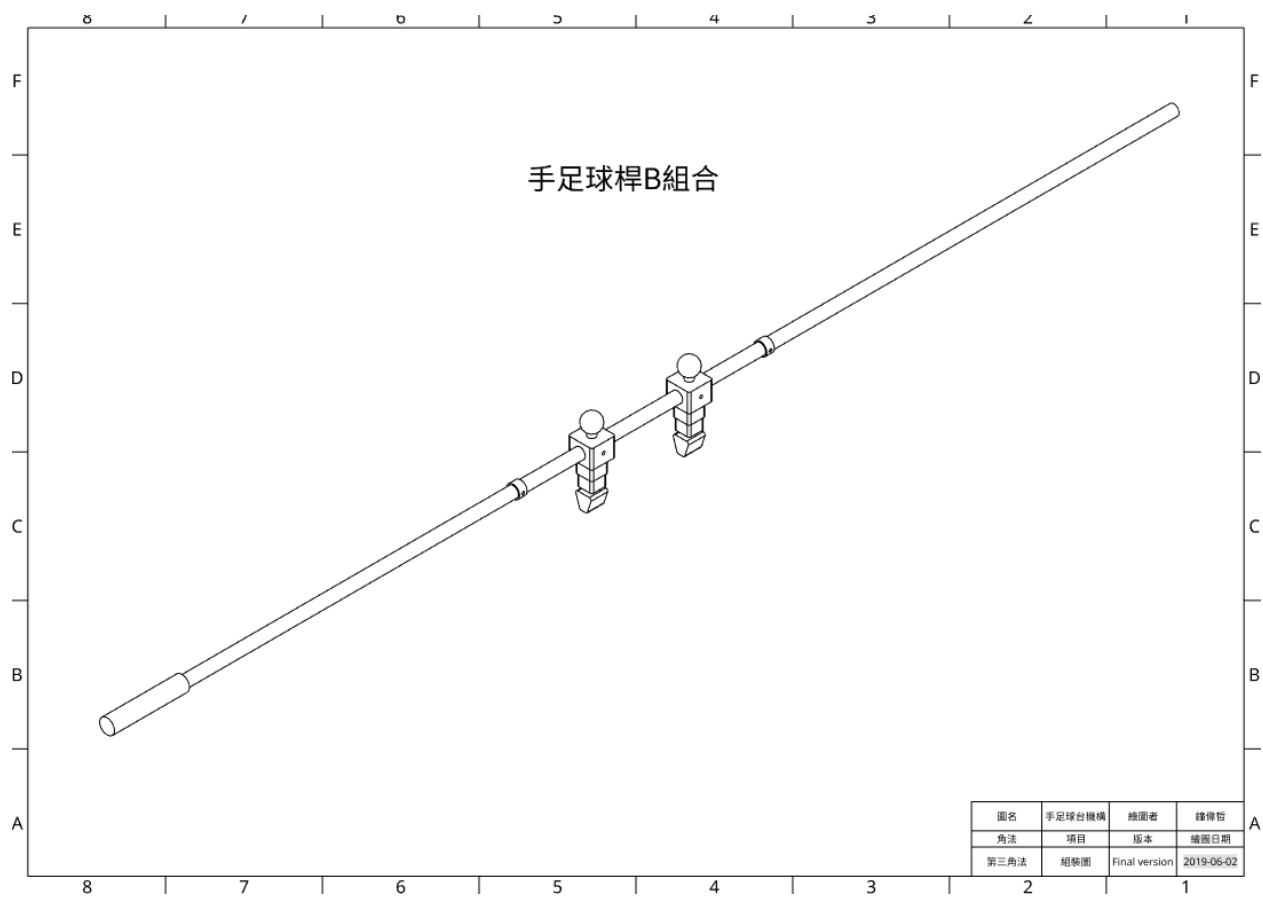


圖 4.3: Partial combination B

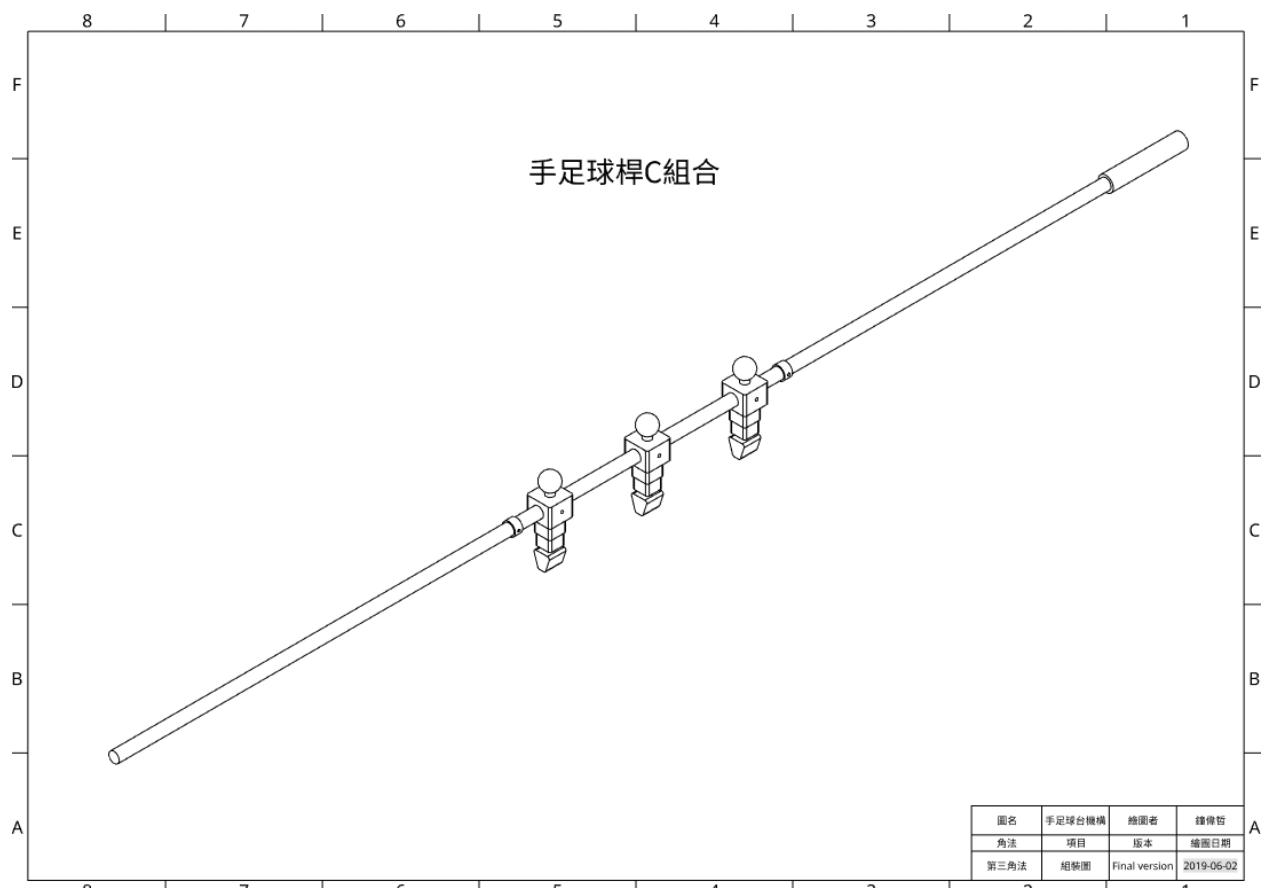


圖 4.4: Partial combination C

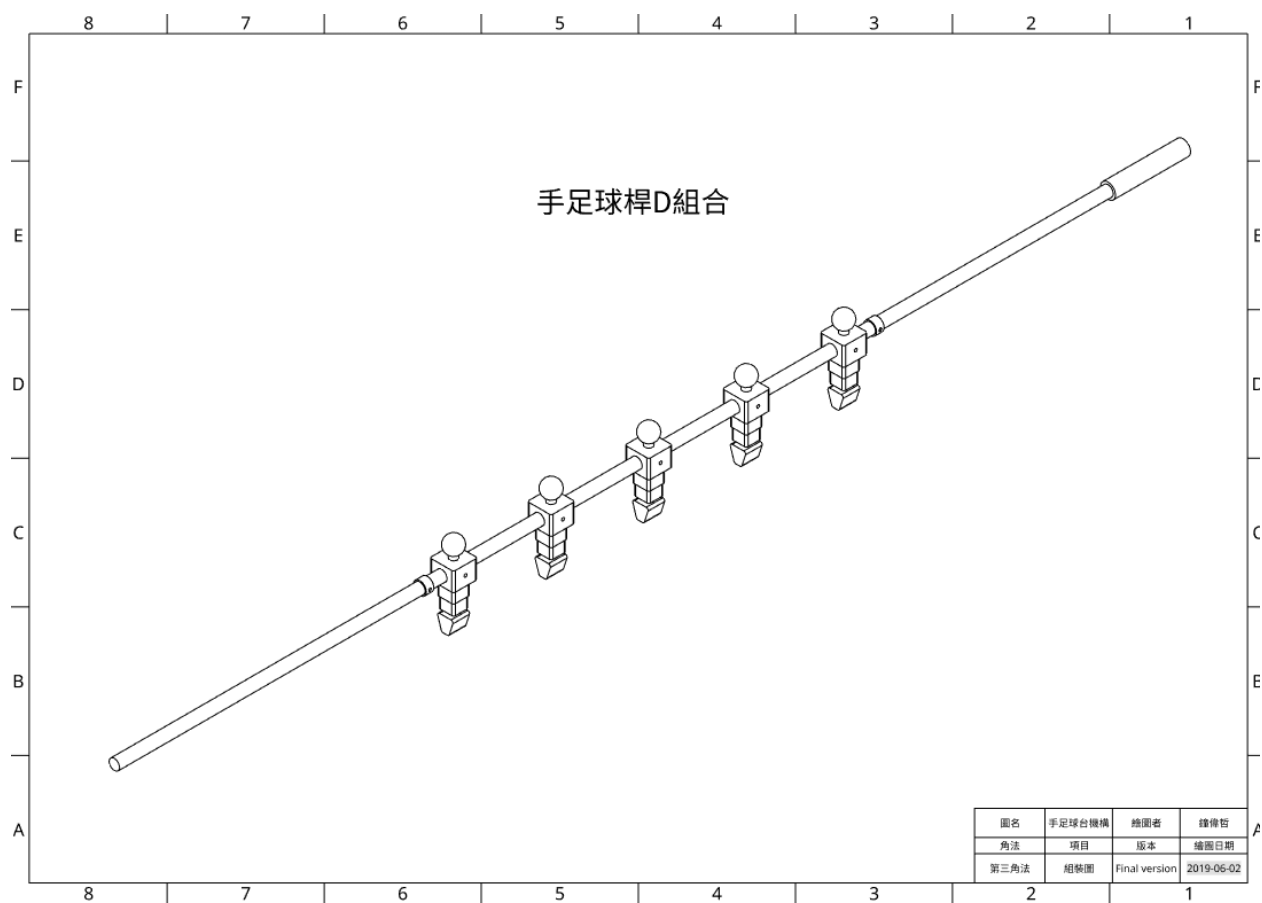


圖 4.5: Partial combination D

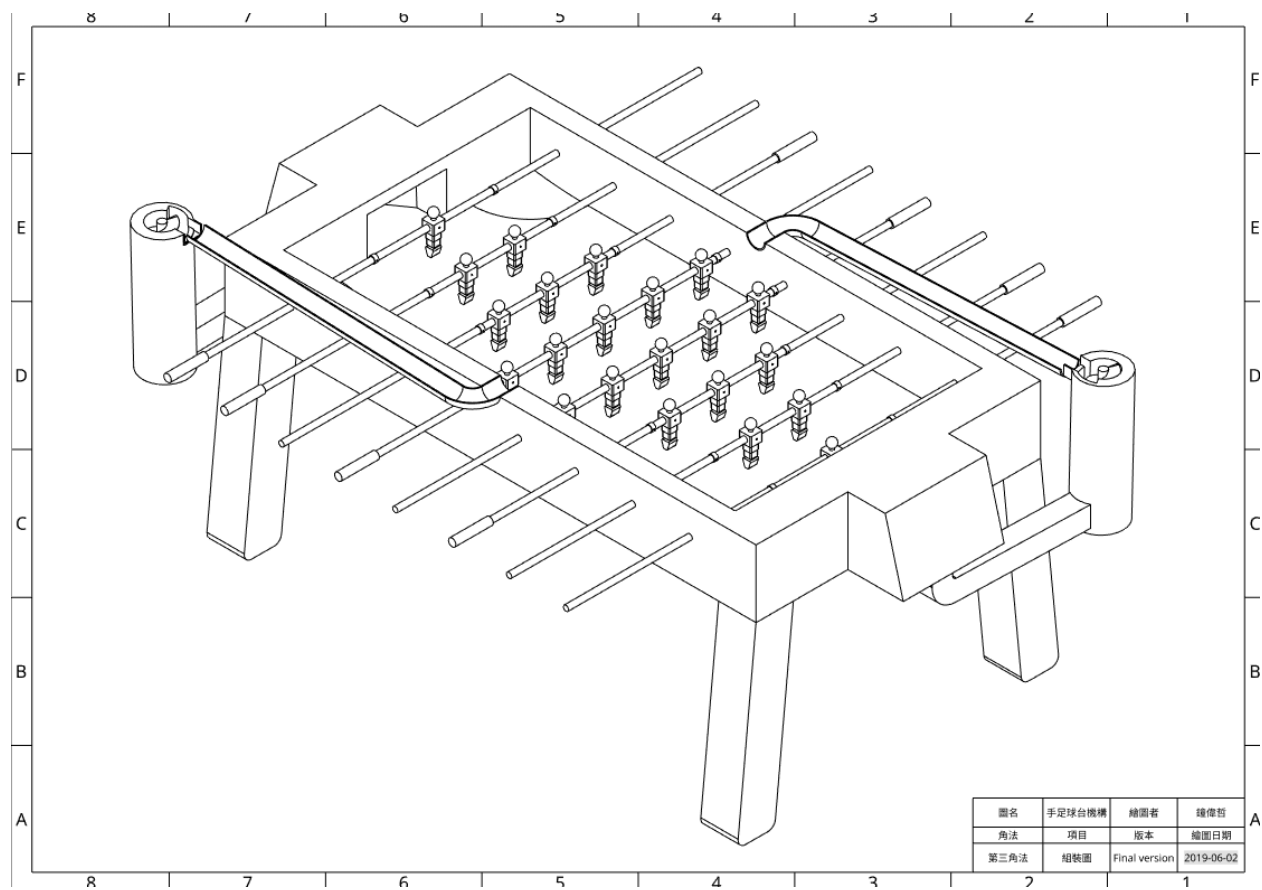


圖 4.6: Combined engineering drawing

第五章 Explosion

5.1 Explosion 爆炸圖

觀看爆炸圖的模樣，並且下方可開啟連結進行下載動作。

explosion 圖片連結

Explosion engineering drawing 圖片連結

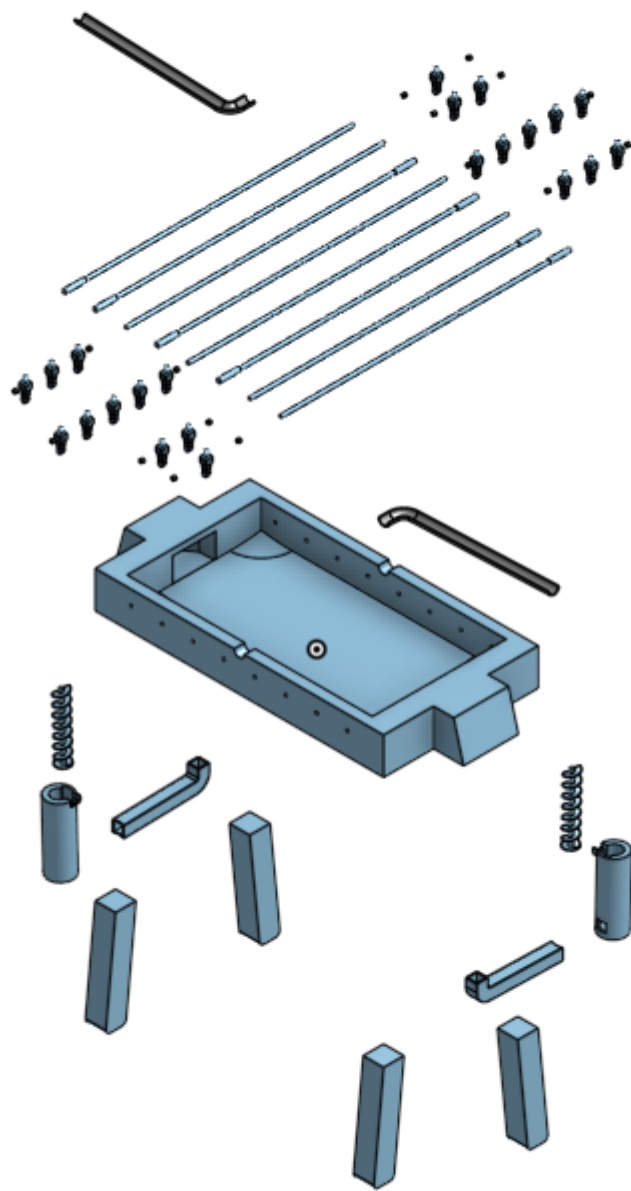


圖 5.1: explosion

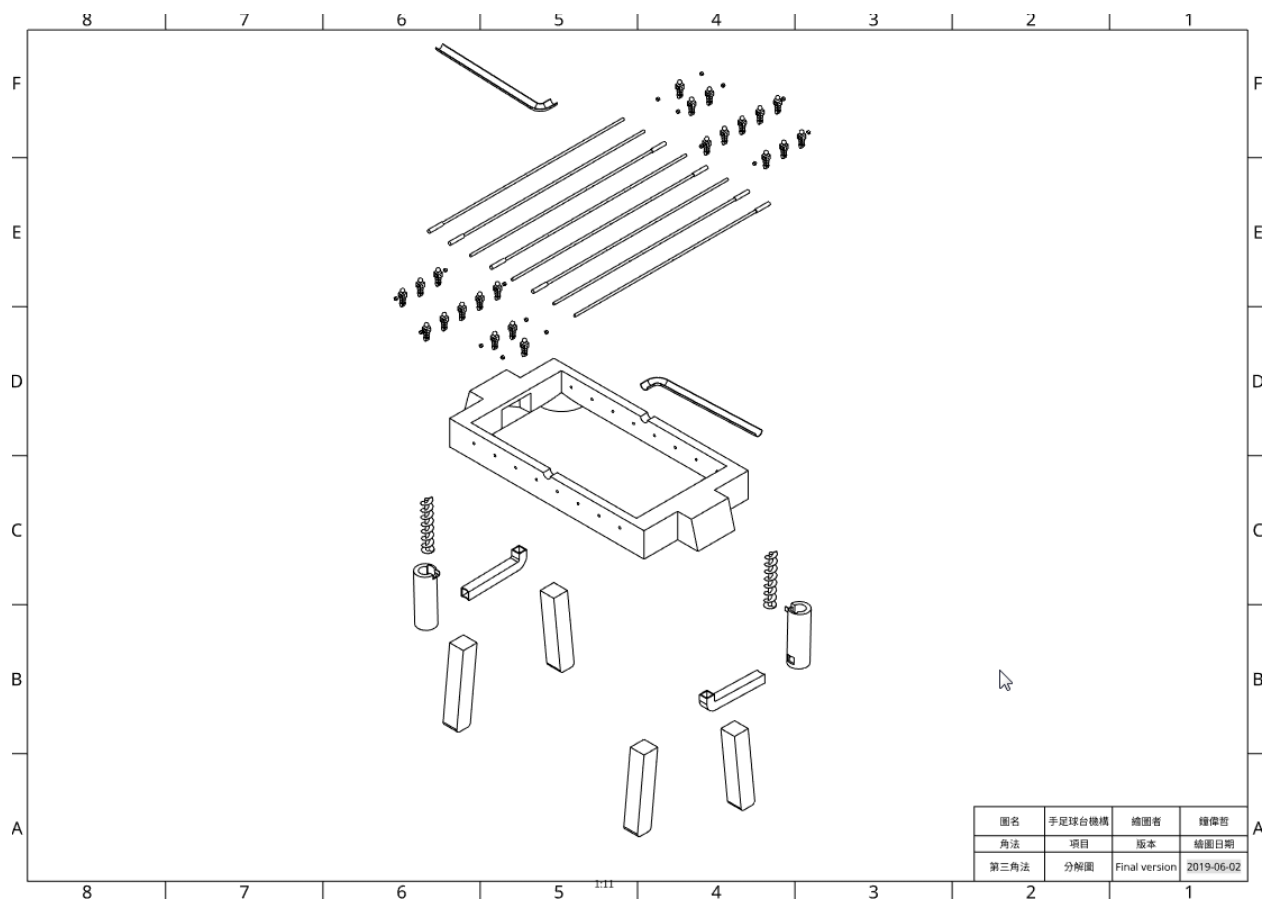


圖 5.2: Explosion engineering drawing

第六章 Operation_Demo 操作演示

6.1 Operation_Demo

把上課老師所教的東西整理並放於此處。

如何建立可攜程式系統

Start.bat 製作

Stop.bat 製作

Kungit 製作

Kungit 使用教學

GitExtensions 使用教學

第七章 Vrep_simulation Vrep 模擬

7.1 Vrep_simulation Vrep 模擬

把 vrep 的東西整理並放於此處。

送球機構模擬練習

V-rep 動態模擬 human to human

```

(Untitled) * ScilE
File Edit Search View Tools Options Language Buffers Help
1 Untitled *

function sysCall_init()
    velocity=0
    velocity2=0
    velocity3=0
    velocity4=0
    velocity5=0
    velocity6=0
    velocity7=0
    velocity8=0

    -----
    sliding_b = 0
    sliding_r = 0
    -----

    left_joint_b1=sim.getObjectHandle('left_R1')
    left_joint_b2=sim.getObjectHandle('left_R2')
    left_joint_b3=sim.getObjectHandle('left_R3')
    left_joint_b4=sim.getObjectHandle('left_R4')

    left_slider_b1=sim.getObjectHandle('left_S1')
    left_slider_b2=sim.getObjectHandle('left_S2')
    left_slider_b3=sim.getObjectHandle('left_S3')
    left_slider_b4=sim.getObjectHandle('left_S4')

    -----

    right_joint_r1=sim.getObjectHandle('right_R1')
    right_joint_r2=sim.getObjectHandle('right_R2')
    right_joint_r3=sim.getObjectHandle('right_R3')
    right_joint_r4=sim.getObjectHandle('right_R4')

    right_slider_r1=sim.getObjectHandle('right_S1')
    right_slider_r2=sim.getObjectHandle('right_S2')
    right_slider_r3=sim.getObjectHandle('right_S3')
    right_slider_r4=sim.getObjectHandle('right_S4')
end
function sysCall_actuation()
    sim.setJointTargetPosition(left_joint_b1,velocity)
    sim.setJointTargetPosition(left_joint_b2,velocity2)
    sim.setJointTargetPosition(left_joint_b3,velocity3)
    sim.setJointTargetPosition(left_joint_b4,velocity4)
    sim.setJointTargetPosition(right_joint_r1,velocity5)
    sim.setJointTargetPosition(right_joint_r2,velocity6)
    sim.setJointTargetPosition(right_joint_r3,velocity7)
    sim.setJointTargetPosition(right_joint_r4,velocity8)

    -----

    sim.setJointTargetPosition(left_slider_b1,sliding_b)
    sim.setJointTargetPosition(left_slider_b2,sliding_b)
    sim.setJointTargetPosition(left_slider_b3,sliding_b)
    sim.setJointTargetPosition(left_slider_b4,sliding_b)
    sim.setJointTargetPosition(right_slider_r1,sliding_r)
    sim.setJointTargetPosition(right_slider_r2,sliding_r)
    sim.setJointTargetPosition(right_slider_r3,sliding_r)
    sim.setJointTargetPosition(right_slider_r4,sliding_r)
    message,auxiliaryData=sim.getSimulatorMessage()
    while message ~= -1 do
        key=auxiliaryData[1]
        sim.addStatusbarMessage('key: '..key)
        if (message==sim.message_keypress) then

```

圖 7.1: human to human Program1


```

key=auxiliaryData[1]
sim.addStatusbarMessage('key: '..key)
if (message==sim.message_keypress) then
-----player_B
    if (auxiliaryData[1]==119)then --w
        velocity=100
        velocity2=100
        velocity3=100
        velocity4=100
    end
    if (auxiliaryData[1]==103) then --g
        -- goalkeeper kick keypress(g)
        velocity = -100
    end
    if (auxiliaryData[1]==104) then
        -- guard kick keypress(h)
        velocity2 = -100
    end
    if (auxiliaryData[1]==106) then
        -- midfield kick keypress(j)
        velocity3 = -100
    end
    if (auxiliaryData[1]==107) then
        -- forward kick keypress(k)
        velocity4 = -100
    end
    if (auxiliaryData[1]==97) then
        -- left key
        if (sliding_b >= 0.05) then
            else sliding_b = sliding_b + 0.01
        end
        sim.addStatusbarMessage('sliding_b: '..sliding_b)
    end
    if (auxiliaryData[1]==100) then
        -- right key
        if (sliding_b <= -0.05) then
            else sliding_b = sliding_b - 0.01
        end
        sim.addStatusbarMessage('sliding_b: '..sliding_b)
    end
-----player_R
    if (auxiliaryData[1]==2007)then
        velocity5=100
        velocity6=100
        velocity7=100
        velocity8=100
    end
    if (auxiliaryData[1]==49) then
        -- goalkeeper kick keypress(1)
        velocity5 = -100
    end
    if (auxiliaryData[1]==50) then
        -- guard kick keypress(2)
        velocity6 = -100
    end
    if (auxiliaryData[1]==51) then
        -- midfield kick keypress(3)

```

圖 7.2: human to human Program2

```

(Untitled) * ScilE
File Edit Search View Tools Options Language Buffers Help
1 Untitled *

    if (auxiliaryData[1]==97) then
        -- left key
        if (sliding_b >= 0.05) then
            else sliding_b = sliding_b + 0.01
        end
        sim.addStatusbarMessage('sliding_b: '..sliding_b)
    end
    if (auxiliaryData[1]==100) then
        -- right key
        if (sliding_b <= -0.05) then
            else sliding_b = sliding_b - 0.01
        end
        sim.addStatusbarMessage('sliding_b: '..sliding_b)
    end
end
-----player_R
    if (auxiliaryData[1]==2007)then
        velocity5=100
        velocity6=100
        velocity7=100
        velocity8=100
    end
    if (auxiliaryData[1]==49) then
        -- goalkeeper kick keypress(1)
        velocity5 = -100
    end
    if (auxiliaryData[1]==50) then
        -- guard kick keypress(2)
        velocity6 = -100
    end
    if (auxiliaryData[1]==51) then
        -- midfield kick keypress(3)
        velocity7 = -100
    end
    if (auxiliaryData[1]==2001) then
        -- forward kick keypress(enter)
        velocity8 = -100
    end
    if (auxiliaryData[1]==2009) then
        -- left key
        if (sliding_r >= 0.05) then
            else sliding_r = sliding_r + 0.01
        end
        sim.addStatusbarMessage('sliding_r: '..sliding_r)
    end
    if (auxiliaryData[1]==2010) then
        -- right key
        if (sliding_r <= -0.05) then
            else sliding_r = sliding_r - 0.01
        end
        sim.addStatusbarMessage('sliding_r: '..sliding_r)
    end
end
message,auxiliaryData=sim.getSimulatorMessage()
end
-----
end

```

圖 7.3: human to human Program3

第八章 Lesson_notes 課程筆記

8.1 Lesson_notes

圖片網址的 BUG:

原本的:

```
page = [w.replace('/images/', '../images/') for w in page]
```

```
page = [w.replace('/downloads/', '../downloads/') for w in page]
```

修改的:

```
page = [w.replace('src='../images/', 'src='../images/') for w in page]
```

```
page = [w.replace('href='../downloads/', 'href='../downloads/') for w in page]
```

對於頁面中的 `w` 進行更換，將 `/images/` 換為 `../images/`，也就是標籤的 `src` 屬性值是圖像文件的 URL，引用該圖像的文件的絕對路徑或相對路徑，而 URL 是指向站點內的文件。

對於頁面中的 `w` 進行更換，將 `/downloads/` 換為 `../downloads/`，也就是標籤的 `href` 是等號右邊用來放置要前往的連結網址，而 `download` 屬性規定被下載的超鏈接目標。

兩個都是以 `content` 為基準的相對目錄設定。

git 常用指令:

git add . 新增所有改版內容

git commit -m “提交訊息, 用來辨識版本” 在近端提交改版內容

git push 將近端改版內容推送到遠端

git pull 拉下遠端新版本內容, 在 fossil scm 使用 fossil update

git log 查驗 git 提交推送紀錄

git status 查驗狀態

git 新增提交推送的身分綁定

以 40623224@gm.nfu.edu.tw 註冊帳號為 40623224 為例:

```
git config --global user.name "40623224"
```

```
git config --global user.email "40623224@gm.nfu.edu.tw"
```

若網路直接連線過慢, 可以透過 proxy 執行 git

```
git config --global http.proxy http://proxy.kmol.info:3128
```

以上的 git config 會將設定存入 Y:/home/.gitconfig 檔案中

刪除近端與遠端的 Git 分支.

To delete a local branch

```
git branch -D the_local_branch
```

To remove a remote branch

```
git push origin :the_remote_branch
```

or

```
git push origin --delete the_remote_branch
```

第九章 Download_link 下載連結

9.1 Download_link

零組件下載連結 Final version: 螺旋桿運動

版本:Onshape(可以自行利用 Onshape 轉換成其他畫圖軟體的檔案, 例如:Solidworks)

檔案連結裡面包含所有零件的畫製過程以及工程圖與爆炸分解圖和組合圖, 使用模組化的製程讓觀閱者可以自行觀看內容並且自行下載。

零組件下載連結第一版: 活塞與連桿

版本:inventor2017(有 STL 檔可以自行用 Onshape、Solidworks 開啟)

其餘版本尺寸或過程可至下列網址點閱:

細部設計與 BOM

送球機構設計與模擬

第十章 影片連結

零組件零件部分組裝圖展示過程 Final version

零組件零件部分組裝圖展示過程第一版

零組件爆炸圖展示過程 Final version

零組件爆炸圖展示過程第一版

送球機構的初步構想 Final version

送球機構的初步構想第一版

送球機構的模擬運動展示第一版

期末示範網站電子書 leo editor