

中華民國第 59 屆中小學科學展覽會 作品說明書

高級中等學校組 工程學(一)科

團隊合作獎

052301

Dr·魔法手「識」

學校名稱：新北市立北大高級中學

作者： 高二 吳昆展 高二 范哲維 高二 詹閔安	指導老師： 李威霖
---	------------------

關鍵詞：復健、手套、Arduino

摘要

臺灣老年人口比率逐年攀升，老人照護逐受重視，而運動對於健康保健更是不可或缺。因此我們想製作一產品，結合時下流行的體感控制，也就是以穿戴手套的方式操控遊戲，讓復健同時兼具娛樂，活動筋骨，一舉多得。

手部與手指的運動分別會被我們所設置的三軸加速度感測器與彎曲感測器偵測，並以類比訊號的形式透過傳輸線傳送對應的鍵盤按鍵訊號至電腦，以此可操縱多種透過鍵盤遊玩的遊戲增進使用之樂趣與意願達到。

使用 Arduino IDE 編寫程式之體感手套，搭配 Scratch 與 KODU 設計之遊戲，邀請 8 位長者體驗，並以問卷回饋與分析。多數長者在活動與遊玩過程中認為，能增加其樂趣以及使用之意願。希望透過本次研究，使復健中的運動能夠翻轉成新世代的模樣！

壹、研究動機

根據內政部統計顯示，臺灣老年人口比率已達 13.33%，預估在 2025 年將攀升至 20%，且持續上升中，如圖 1-1 所示。現代中，樂活人生已成為一股風潮，但我們觀察家中的長輩們發現在日常生活中，常常艱困地控制手部、因無力而需攙扶或協助提起重物。查詢相關資料的過程中，發現運動復健過程往往沉悶又無趣。為解決此觀察到的問題，我們因而想做出一項產品，能結合復健、運動與遊戲的手套，讓過程中增添樂趣。

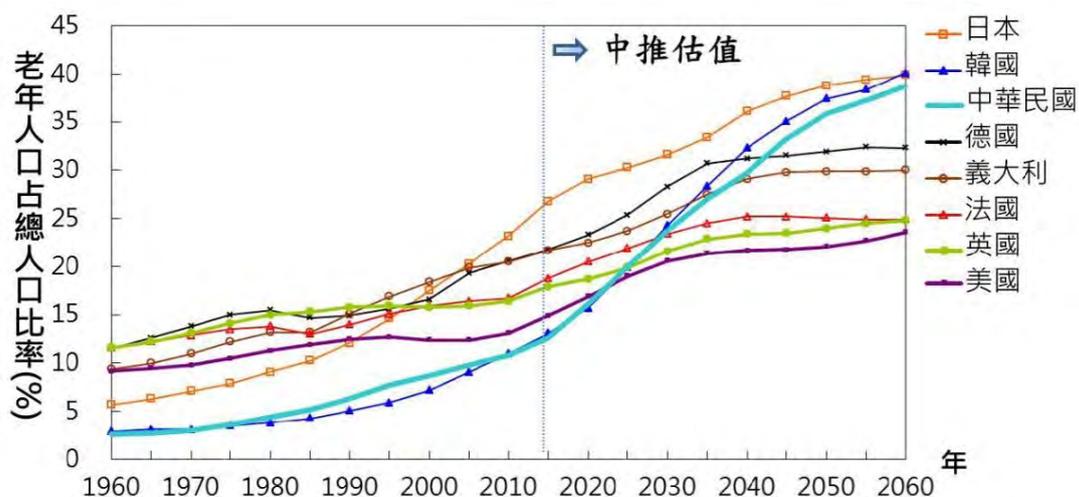


圖 1-1 內政部統計資料

貳、研究目的

復健的過程中大多單調，重複同樣的動作使人覺得乏味(圖 2-1)；「我討厭跌倒(I HATE FALLING)」一詞為長者們應注意的跌倒原因之縮寫(圖 2-2)，其中的 L 代表體能不足(Lack of conditioning)；根據 2007 年國民健康署「健康行為危險因子監測調查」統計顯示，過去兩個禮拜有運動習慣的老人僅佔 52.3%。許多研究皆表明對於長者們來說運動的重要性。

一、以 Arduino 製作可辨識手勢的手套

以手部為基礎開始構想，使用感測器與 Arduino 開發版的結合，透過回傳的數值進行判斷，分析出使用者的手部姿勢、動作。

二、以完成之輔具提高長者使用之樂趣與意願

使用輔具以遊戲式復健與長者照護兩者結合，使長者於運動與復健過程中感到樂趣，進而期望提高長者們或患者們活動身體之意願。

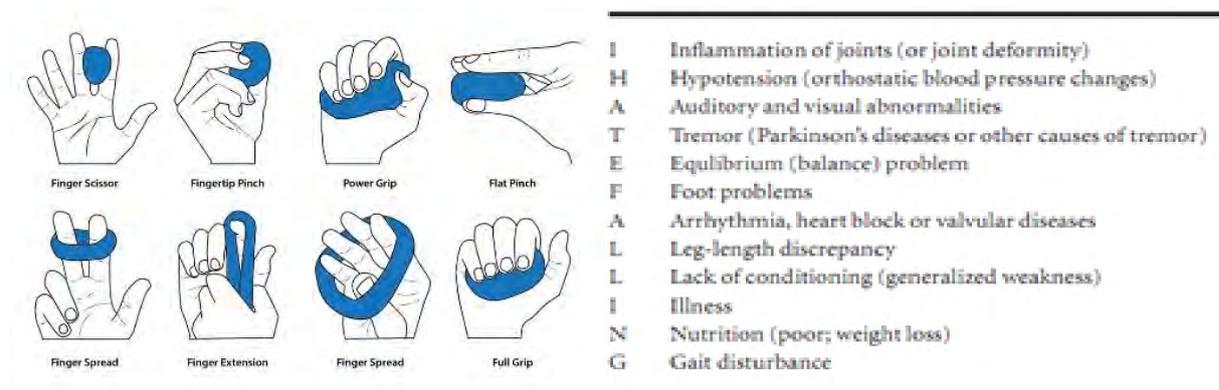
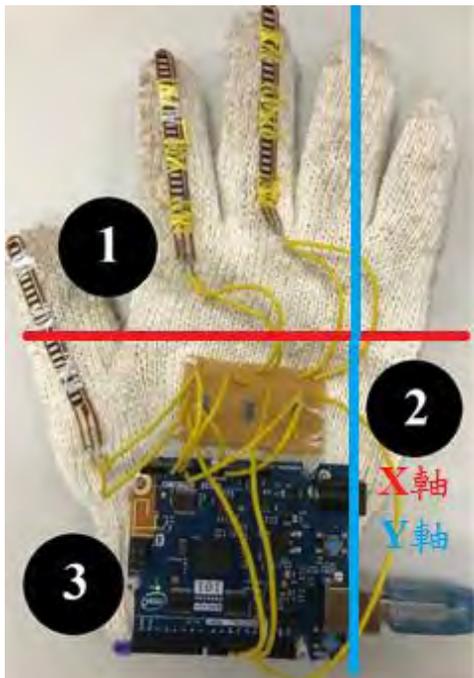


圖 2-1 Hand Therapy Putty Exercises

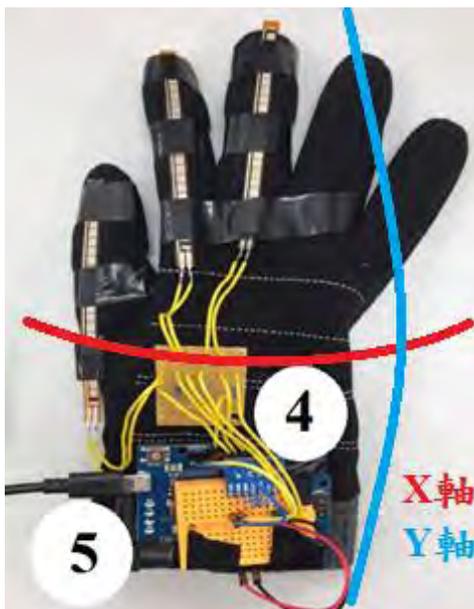
圖 2-1 I HATE FALLING

參、研究設備及器材

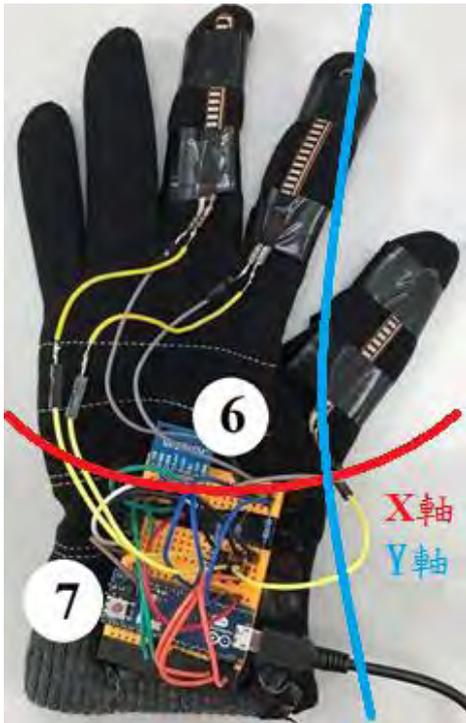
本研究收集文獻、參考技術文章，以及硬體規格表，與指導老師討論並嘗試解決問題後，先手工繪製草圖，再使用 Fritzing Fritzing 繪製電路圖。總共使用了三款 Arduino 開發版，並搭配三軸加速度感測器、彎曲感測器以及陀螺儀，透過 Arduino IDE 和 Processing IDE 來編寫程式。在遊戲設計過程前，嘗試多款遊戲後，對於年長且不熟遊戲的使用者頗為困難，因此決定使用 Scratch 2 與 Kodu Game 設計符合初學者簡單有趣且達運動復健效果的遊戲。



	名稱	規格	用途
	Flex/Bend Sensor	長：5.6 cm 平放電阻：10K	透過內部電阻值的改變來測量手指的彎曲程度
	萬能板	雛形：5×9 二代：7×6	作為感測器與 Arduino 的連接橋梁
	Genuino 101	晶片：Intel Curie 特色：內建六軸感測器、BLE	作為手套控制器，使用內建三軸加速度感測器



	ADXL345	型號：GY-291 通訊：I ² C/SPI 供電：3~5V	透過加速度變化測量手掌傾斜程度
	Arduino Leonardo	晶片：ATmega32u4 特色：獨立 USB-Serial 轉換芯片，實現 USB 通信及控制	手套控制器，使用 Keyboard 函式庫操縱鍵盤



6	MPU6050	型號：GY-521 通訊：I ² C 供電：3~5V	透過六軸感測器的變化測量手掌的傾斜與揮動
7	Arduino Micro	晶片：ATmega32u4 特色：獨立 USB-Serial 轉換芯片，實現 USB 通信及控制	手套的控制器，且同樣具有 Keyboard 函式庫

	名稱	用途
	Arduino IDE	在各代手套中皆使用此整合開發環境來編寫程式，並編譯以控制 Arduino 開發版
	Processing IDE	配合 Genuino 101 開發版，將回傳之數值轉換成鍵盤指令，用以操控電腦
	Fritzing Fritzing	在研究與測試各個感測器與開發版中，利用此軟體繪製電路草圖，方便編排線路的連接
	KODU	用以設計遊戲程式，並在各代手套中用來實測運行效率與效果。遊戲畫面具有立體感
	Scratch2	用以編寫遊戲程式，並在各代手套中用來實測運行效率與效果。遊戲畫面為平面效果

肆、研究過程或方式

研究過程中，首先參考許多硬體製作與程式設計之教學文件製作出初版雛形，並針對使用上所面臨的問題進行改善。透過多個 Arduino 版本與另外加裝感測器做測試，設計出能減低操作延遲感，輕量化、提升手勢辨識率與使用者體驗等。接著由 35~90 歲區間的長者進行試用，再以問卷方式取得使用者體驗回饋，最後進行分析資料提出結果與討論。

一、手套優化設計過程

多款產品在我們擁有初步的想法後，首先測試了彎曲感測器(圖 4-1)，成功使用因彎曲就會改變的電阻值之變化，透過設定數值變動範圍來發送指令。並使用 Genuino 101 內建的陀螺儀來測試能否測量手掌翻轉的變化，然而實作之後發現，無法順利的觀察數據的改變。於是，我們改成以三軸加速度感測器來判定、測量其值的變化，程式畫面測試畫面(圖 4-2)。

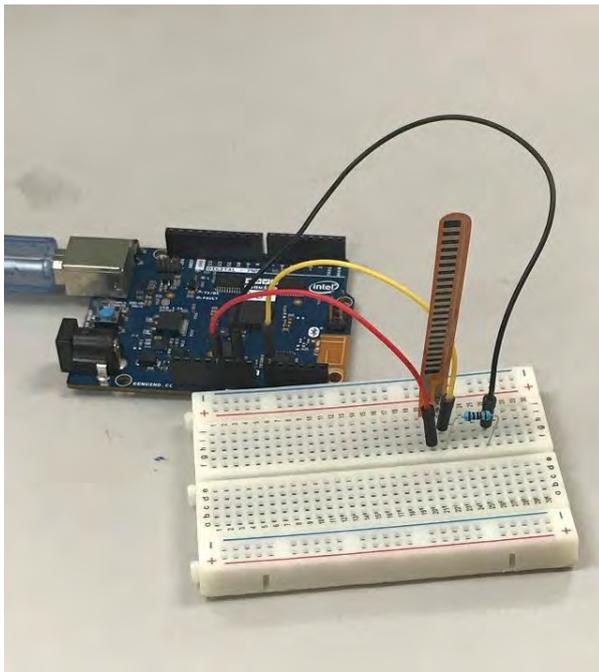


圖 4-1 彎曲感測器測試

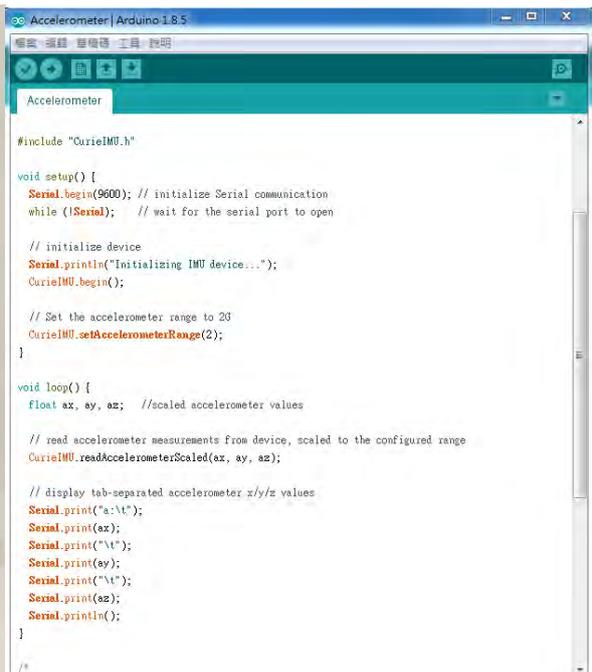


圖 4-2 Arduino 三軸加速度感測程式圖

而在我們成功使用感測器獲得數值後，試著將以上兩者結合，並嘗試裝置於手套上。以 Fritzing Fritzing 來構思電路的配置與繪製草圖(圖 4-3)，並利用烙鐵與錫線，將線路焊至萬能板上，試圖將整個裝置輕量化。接著我們用 Processing IDE 編寫程式碼，讓所偵測的手部動作的 Arduino 訊號數值，轉譯為鍵盤指令。將彎曲感測器分別安裝在手套的三根手指上，並將 Genuino 101 開發版放置在手背，並用棉線縫紉固定，最後完成能控制鍵盤指令的初代手套(圖 4-4)。

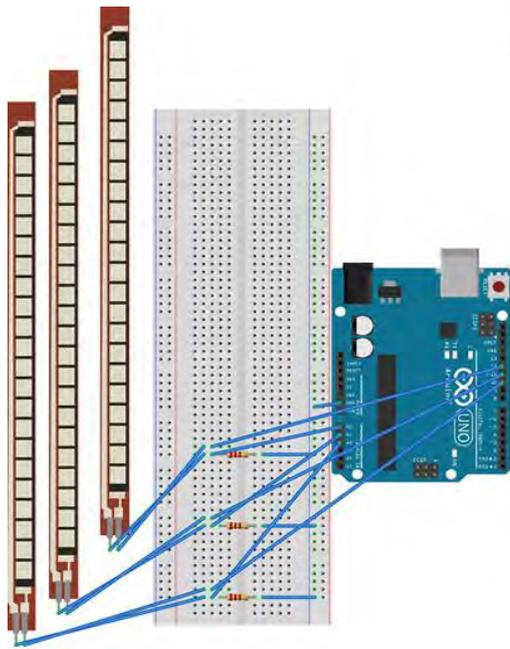


圖 4-3 電路草圖

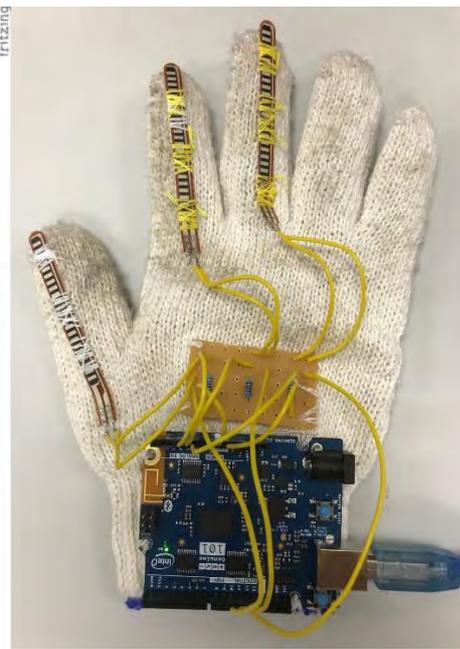


圖 4-4 初代手套

完成第一代的手套後，我們製作了兩款不同開發環境的遊戲，用以實測手套的靈敏度與功能性。分別是使用 Scratch2 製作的簡易打地鼠遊戲(圖 4-5)，與使用 KODU 所製作的賽車遊戲(圖 4-6)。以賽車遊戲來做舉例，設計成透過傾斜手掌可以使角色左、右移動，彎曲大拇指或食指可以操控角色的前進與煞車。我們試操作時發現，手套的靈敏度與反應度還尚待改進，延遲的數秒鐘容易導致撞車、卡牆而無法前進，因此我們開始尋找能夠改善的方法。



圖 4-5 Scratch 打地鼠遊戲



圖 4-6 KODU 賽車遊戲

針對問題改善的方式，透過 Arduino Leonardo 開發版(圖 4-7)Keyboard 函式庫，不需額外依賴 Processing IDE 轉譯程式，可直接對應鍵盤指令操作。但因為該開發板並無內建的三軸加速度感測器，因此我們必須另外加裝 ADXL345 三軸加速度感測器。這次的改良消除了 Processing

IDE 運行時的延遲狀況，使手套整體運行速率加快。而後為了增加使用者操作上的舒適感，我們使用了 Arduino Micro 開發板，同樣具有 Keyboard 函式庫，並成功製作出左手的新完成品(圖 4-8)，擴大可使用的族群。

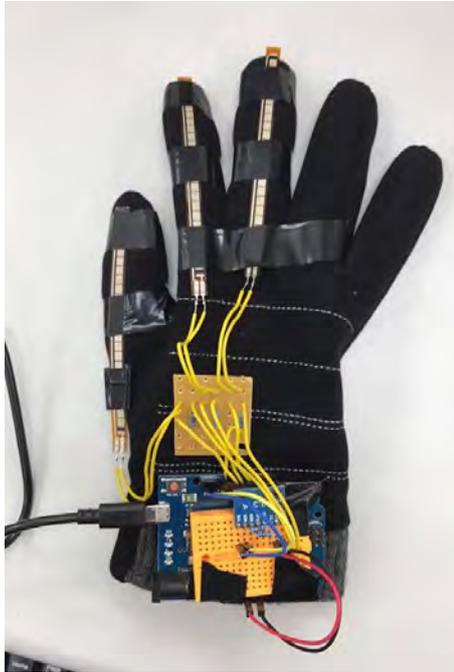


圖 4-7 Arduino Leonardo

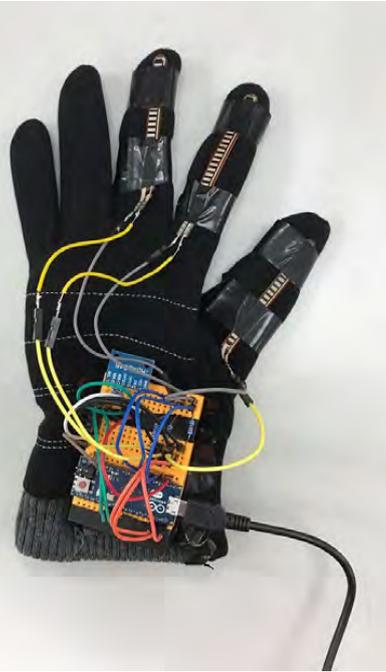


圖 4-8 Arduino Micro

二、手套程式功能操作優化

在新階段中，為使手套的運行更加方便，及避免戴上手套時，因為晃動或傾斜而產生不必要的指令，因此設計了開關系統。藉由彎曲感測器，當偵測到中指上的感測器彎曲，Delay 指令就會開始倒數(圖 4-9)。當達到所設的秒數後，便會執行開與關的程式(圖 4-10)，開啟後才能進行指令的傳送。並透過 LED 的亮與滅，讓使用者了解目前是開或關的狀態。藉此系統讓我們能在同個手套中任意地切換不同模式，分別是平放手掌且彎曲中指時可進入鍵盤操控模式，結合遊戲進行手部的活動與遊玩；直立手掌且彎曲中指時可進入手勢分析模式，識別手部的揮動。

為使長者們在復健的過程中更有趣，於鍵盤操控模式中，透過彎折手指時，彎曲感測器回傳的數值變化，觀察使用者是否握拳(圖 4-11)，可以達到同時彎曲多根手指的運動效果，模擬患者們的復健過程。除了彎曲大拇指、食指與傾斜、翻轉手掌外，新增抓、握時，計數且輸出握拳次數(圖 4-12)，並發送對應的鍵盤指令。透過多樣的按鍵，可以使遊玩豐富的遊戲時，充分的活動手部關節。

```

2019224 | Arduino 1.8.8
檔案 編輯 草稿碼 工具 說明
2019224 $
if(Y<150){
  Mode=1;//手部平放
}
else if(Y>150){
  Mode=2;//手部直立
}
if(valA2 < 510){
  t++;//倒數
  if(t>t1){
    key = !key;
    RealMode=Mode;
    t=0;
  }
}
else t=0;
delay(250);
}

```

圖 4-9 倒數程式碼

```

2019224 | Arduino 1.8.8
檔案 編輯 草稿碼 工具 說明
2019224 $
if(key==false){
  digitalWrite(LED_BUILTIN, LOW);
  Serial.println("==關機狀態==");// Mode 0
}
if(key==true && RealMode==1){
  digitalWrite(LED_BUILTIN, HIGH);
  Serial.println("==鍵盤操控中==");//切換成Mode 1
}
else if(key==true && RealMode==2){
  digitalWrite(LED_BUILTIN, HIGH);
  Serial.println("==手勢分析中==");//切換成Mode 2
  temp=0;
  if(temp==0){
    if(Y > 400){
      Serial.println("手勢向上");
      if(circle1==0){
        circle1++;
      }
    }
  }
}

```

圖 4-10 開關程式碼

```

2019224 | Arduino 1.8.8
檔案 編輯 草稿碼 工具 說明
2019224 $
if(key==true && RealMode==1){
  digitalWrite(LED_BUILTIN, HIGH);
  Serial.println("==鍵盤操控中==");//切換成Mode 1
  Serial.print("握拳次數:");
  Serial.println(grab);
  if(grab == 20){
    Serial.println("恭喜");
    grab=0;
    delay(3000);
  }
  if(valA0 < 510 && valA1 < 510 && valA2 < 510 && count == 0){
    tgrab++;//抓、握
    if(tgrab>t1grab){
      grab++;
      count=1;
    }
  }
  else if(valA0 > 510 && valA1 > 510 && valA2 > 510){
    count=0;
  }
  else{
    tgrab=0;
  }
}

```

圖 4-11 握拳程式碼

```

COM3 (Arduino/Genuino Micro)
傳送
Y= -26.00 Y= 51.00 Z= -274.00
==鍵盤操控中==
握拳次數: 1
452,461,438
X= -27.00 Y= 54.00 Z= -278.00
==鍵盤操控中==
握拳次數: 1
453,462,438
X= -27.00 Y= 53.00 Z= -274.00
==鍵盤操控中==
握拳次數: 1
454,462,439
X= -27.00 Y= 53.00 Z= -269.00
==鍵盤操控中==
握拳次數: 2
454,462,439
X= -20.00 Y= 51.00 Z= -272.00
==鍵盤操控中==
握拳次數: 2
519,535,515
X= -20.00 Y= 64.00 Z= -383.00
==鍵盤操控中==
握拳次數: 2
535,543,534
X= -2.00 Y= 63.00 Z= -379.00
==鍵盤操控中==
自動啟動 Show timestamp NL(newline) 9600 baud Clear output

```

圖 4-12 輸出握拳次數

在我們透過開關功能所增設第二個模式中，使用三軸加速度感測器偵測的數值，分析手部的全方位揮動(圖 4-13)。成功感應並使用 print 指令輸出手勢向上、下、左、右後，開始嘗試分析更進一步的手勢。設計了一套演算法，當連續偵測到固定順序的方位揮動時，能夠判斷並印出「手順時針轉動」及「手逆時針轉動」(圖 4-14)。

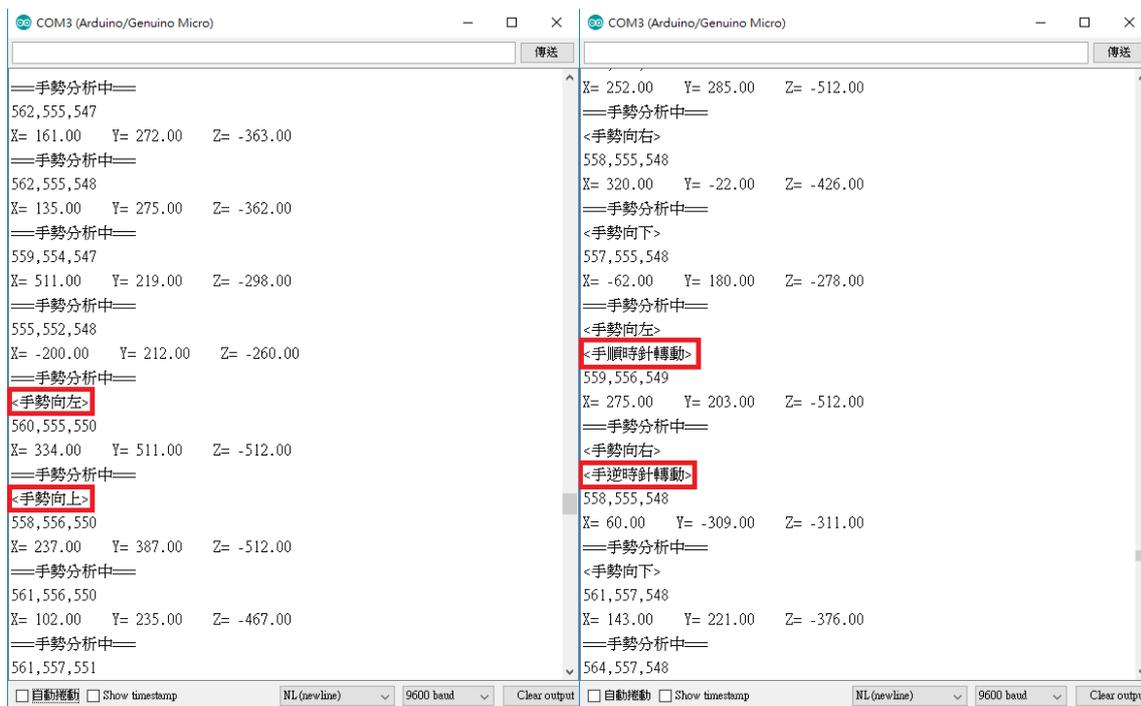


圖 4-13 手部上下左右移動

圖 4-14 順時針圓與逆時針圓

緊接著，使用了 MPU-6050 陀螺儀來進行改良，因為我們發現透過陀螺儀偵測出的數值(圖 4-15)，能夠使手勢的判斷更加精確。並且由於它同時擁有三軸陀螺儀與三軸加速計感測模組，能夠直接替換只有三軸加速度感測的 ADXL345，不會使手套整體的配置變得更複雜。

為了使手套功能更全面、完整，我們將五根手指皆裝上彎曲感測器，完成第三代手套(圖 4-16)。且將手套的兩種模式結合，使其不必透過開關系統切換即可一同使用。並參考各資料的復健姿勢與動作，新增了新的復健動作：將大拇指與其餘四指之一同時彎曲(圖 4-17、4-18)。配合利用 KODU 設計的遊戲，當做出此動作時，即可發射飛彈，在固定的時間內擊倒前來的敵人，保護我方不受傷害即可通關。

```

20190307 | Arduino 1.8.8
檔案 編輯 草稿碼 工具 說明
20190307 $
void setup(){
  Serial.begin(38400);
  Wire.begin();//初始化I2C
  #if I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
    Fastwire::setup(400, true);
  #endif
  accelgyro.initialize();
}
void loop(){
  accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  #ifdef OUTPUT_BINARY_ACCELYGRO
    Serial.write((uint8_t)(ax >> 8)); Serial.write((uint8_t)(ax & 0xFF));
    Serial.write((uint8_t)(ay >> 8)); Serial.write((uint8_t)(ay & 0xFF));
    Serial.write((uint8_t)(az >> 8)); Serial.write((uint8_t)(az & 0xFF));
    Serial.write((uint8_t)(gx >> 8)); Serial.write((uint8_t)(gx & 0xFF));
    Serial.write((uint8_t)(gy >> 8)); Serial.write((uint8_t)(gy & 0xFF));
    Serial.write((uint8_t)(gz >> 8)); Serial.write((uint8_t)(gz & 0xFF));
  #endif
}
儲存完畢
43 Arduino/Genuino Micro COM3

```

圖 4-15 陀螺儀程式碼

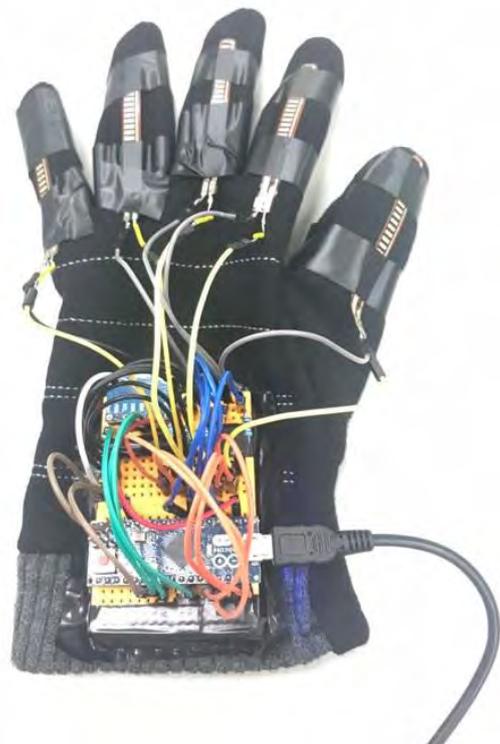


圖 4-16 裝上 MPU-6050 手套



圖 4-17 手指復健

```

20190529 | Arduino 1.8.9
檔案 編輯 草稿碼 工具 說明
20190529
if(valA0 < 120 && valA1 < 110 && valA2 > 110 && valA3 > 110 && valA4 > 160){
  Keyboard.press((char)49);
  Serial.println("彎曲食指和大拇指");
}
else if(valA0 > 120 && valA1 > 110 && valA2 > 110 && valA3 > 110 && valA4 > 160){
  Keyboard.release((char)49);
}
if(valA0 < 120 && valA1 > 110 && valA2 < 110 && valA3 > 110 && valA4 > 160){
  Keyboard.press((char)50);
  Serial.println("彎曲中指和大拇指");
}
else if(valA0 > 120 && valA1 > 110 && valA2 > 110 && valA3 > 110 && valA4 > 160){
  Keyboard.release((char)50);
}
if(valA0 < 120 && valA1 > 110 && valA2 > 110 && valA3 < 110 && valA4 > 160){
  Keyboard.press((char)51);
  Serial.println("彎曲無名指和大拇指");
}
else if(valA0 > 120 && valA1 > 110 && valA2 > 110 && valA3 > 110 && valA4 > 160){
  Keyboard.release((char)51);
}
if(valA0 < 130 && valA1 > 110 && valA2 > 110 && valA3 > 110 && valA4 < 170){
  Keyboard.press((char)52);
  Serial.println("彎曲小拇指和大拇指");
}
else if(valA0 > 130 && valA1 > 110 && valA2 > 110 && valA3 > 110 && valA4 > 170){
  Keyboard.release((char)52);
}
100 Arduino/Genuino Micro COM3

```

圖 4-18 手指復健程式碼

三、長者使用體驗與問卷回饋

最後為了使手套能與復健緊密結合，我們設計了一份共 14 道題目的問卷，包含了選擇題和簡答題(附錄二)。總共邀請 8 位 35 歲以上的人，其中有 5 位 65 歲以上的年長者(圖 4-19)，並有 1 位曾經接受過復健(圖 4-20)。戴上手套體驗遊戲，給予我們回饋，透過他們的建議，能讓我們了解目前成品的成效與不足。

你/妳的年齡介於?

8 則回應

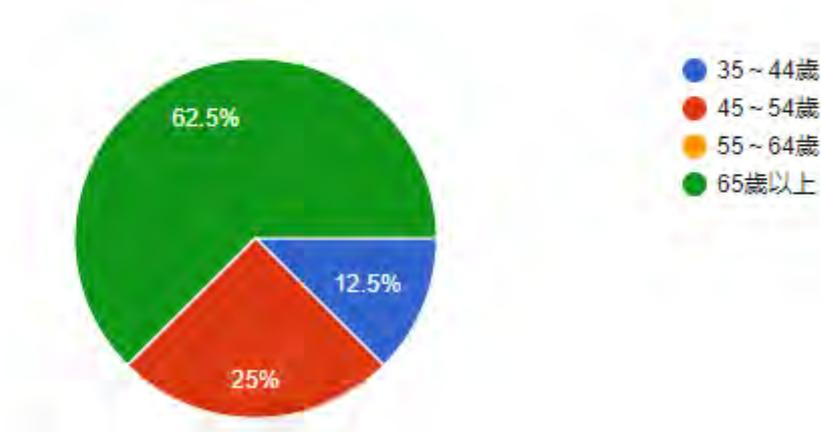


圖 4-19 受試者年齡

你/妳是否曾經或現在需要接受復健?

8 則回應

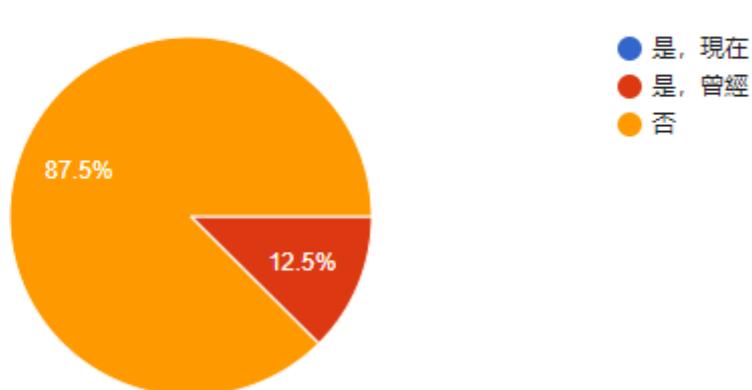


圖 4-20 受試者經歷

伍、研究結果

一、遊戲介紹



圖 5-1 KODU 遊戲第一層手掌傾斜活動

(一) 左右轉向

將手掌左、右傾斜，三軸加速度感測器偵測到數值變化後，執行鍵盤指令，並設計成於遊戲中，當接收到對應的鍵盤指令，可控制角色的左、右轉向(畫面 2)。

(二) 向前加速

而在角色靜止時，彎曲拇指，彎曲感測器偵測到變化，則是能夠使角色向前加速(畫面 2)。我們亦有設置較慢速度的加速指令於食指，讓進行方式更多元，可進行細微的操縱。

(三) 遊戲介面

遊戲剛開始時倒數 3 秒後(畫面 1)，沿地上的紅色箭頭前行，並在最後於終點撞擊蘋果即可通關進入下一階層(畫面 3)。右下角是縮小地圖後，用上帝視角從起點觀看，第一層地圖的全貌(畫面 4)。

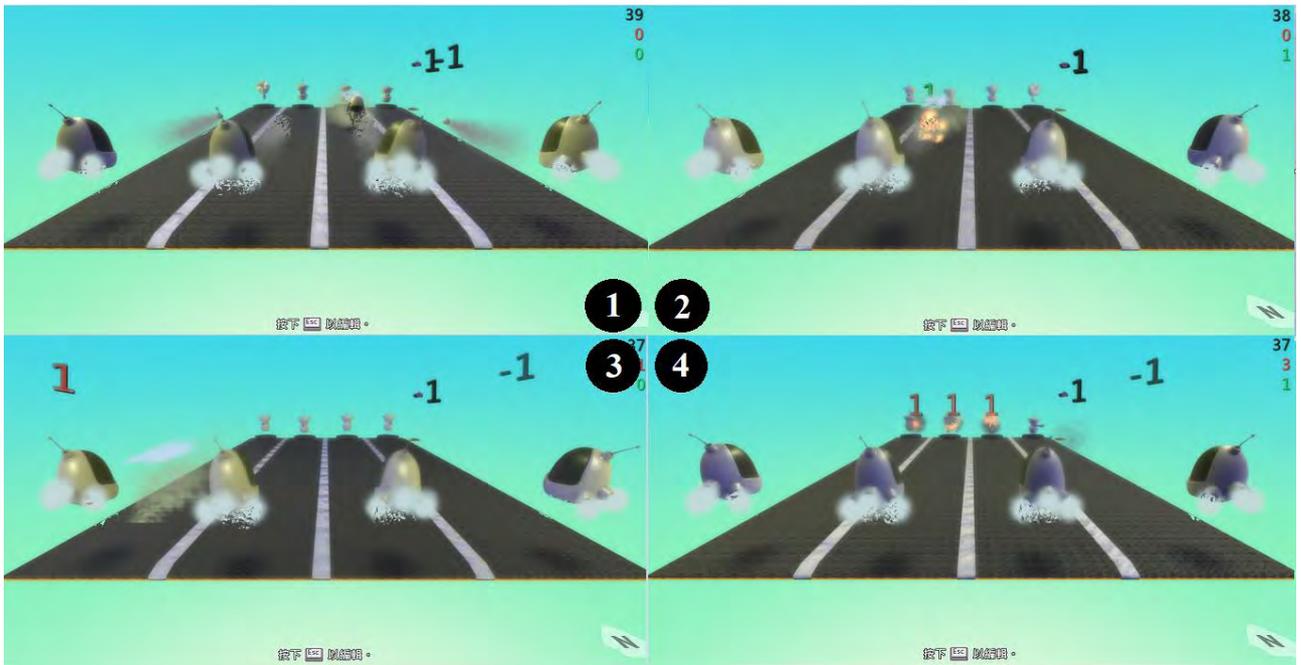


圖 5-3 KODU 遊戲第三層手指抓握活動

(一) 發射飛彈

在出現單輪車時，根據不同道來發射飛彈，從右至左，分別是將大拇指與食指、中指、無名指和小拇指之一同時彎曲(畫面 1)。

(二) 遊戲介面

在第二層獲取 10 分，進入第三層時，畫面右上角黑色數字會開始倒數計時，總共有 45 秒的時間(畫面 1)，在時間內，要發射飛彈保護 KODU 不被撞擊，完成任務即可通關獲勝！若不幸防禦失敗，導致 KODU 受到撞擊，或是飛彈射錯道，畫面右上紅色數字就會+1(畫面 3、4)，用來計數失敗次數；若成功擊中單輪車，畫面右上角的綠色數字便會+1，可用來統計成功次數(畫面 4)。

二、手套運行流程圖



圖 5-4 手套整體運行流程圖

(一) 輸出指令

1. 握拳

當偵測到使用者握拳時，會輸出鍵盤指令 c。

2. 兩根彎曲

當偵測到使用者將大拇指與其餘四指之一同時彎曲，食指到小拇指分別輸出鍵盤指令 1~4。

3. 個別彎曲

當偵測到使用者彎曲大拇指、食指時，分別會輸出鍵盤指令 a、b。

4. 傾斜

當偵測到使用者傾斜手掌時，會輸出鍵盤指令對應的方向鍵。

5. 揮動、畫圓

當偵測到使用者手心朝前揮動手掌時，會在 Arduino 序列埠監控視窗中印出對應的「手勢向上(下、左、右)」；當偵測到使用者手心朝前手掌畫圓時，同樣在序列埠監控視窗中，會印出「手順(逆)時針轉動」。

三、問卷調查

整體而言，你/妳覺得這項產品，對於復健過程是否有幫助？

8 則回應

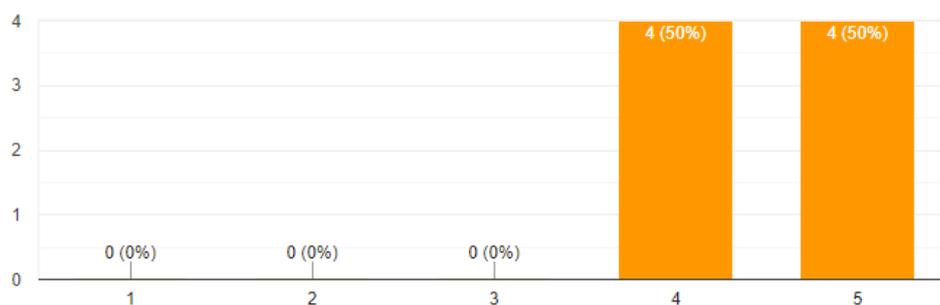


圖 5-5 受試者看法

倘若這項產品能夠在世面上普遍應用，你/妳對於此產品的支持度為何？

8 則回應

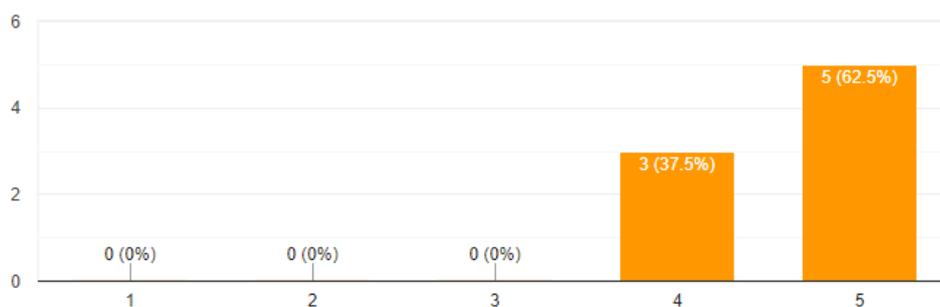


圖 5-6 受試者支持度

使用過程中，你/妳覺得遊戲有趣嗎？

8 則回應

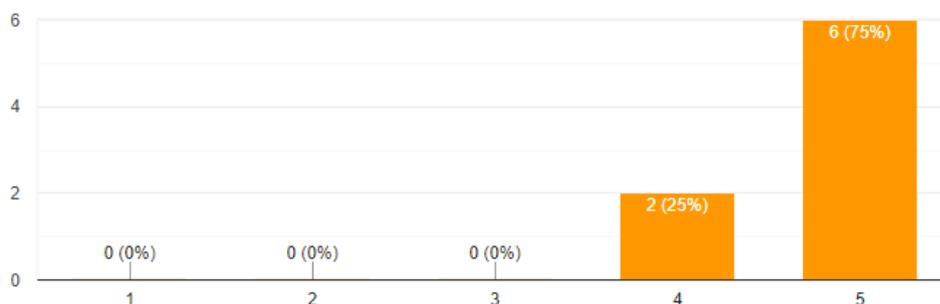


圖 5-7 受試者評價

依照使用後的感想，你/妳覺得自己多久會想使用一次？

8 則回應

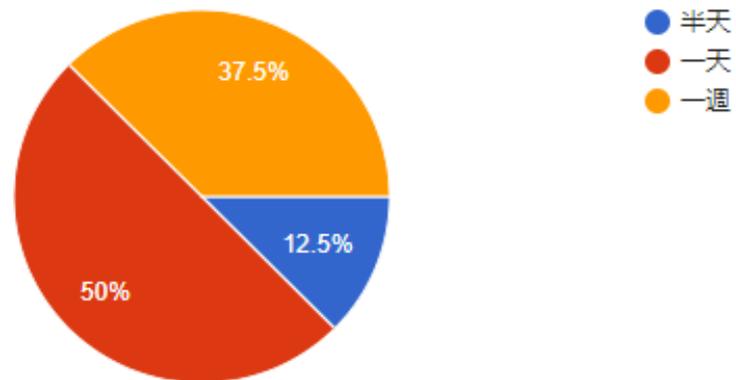


圖 5-8 受試者感想

在體驗完遊戲後，請受試者填寫我們設計的問卷，若選項為 1~5，1 為非常不贊成，5 為非常贊成。在整體而言，多數人認為此產品對於復健有幫助(圖 5-5)，並且支持未來能夠普及於世面上(圖 5-6)。其中有 6 位覺得遊戲很有趣(圖 5-7)，並且 5 位認為至少一天會想使用一次(圖 5-8)。

陸、討論

一、外觀改善

此次手套，主要是著重於功能性的展現，因此在外觀與舒適度上尚有許多的改進空間，如將萬能版和排線改為電路板焊接縮小體積，以及手套能適性針對手掌大小或是左、右手或是其他部位皆能調整大小舒適適用。

二、軟硬體更新

因為 Arduino 是一種為新手和專業人員，提供一種低成本且簡單的方法，以建立使用傳感器與環境相互作用的裝置執行器，因此可嘗試使用其他更加精簡的控制器取代 Arduino。更能搭配藍芽或 Wi-Fi，使手套不必依靠傳輸線連接電腦。且不同的遊戲，操作上常使用不同的按鍵，可將程式改寫成適應各種遊戲。

三、結合智慧家庭生活

在將長者們的手部運動結合科技、娛樂的同時，開發多元的手勢，希冀手勢與智慧家電能夠使長者們的生活更加便利、安全。

四、輔具的延伸發展

產品部位也不限於手，若能發想出穿戴在腳部等等的裝置，就能使遊戲、復健受到更廣泛的運用與結合。我們也期待著能夠將產品進行更多不同方面的實地測試，蒐集與分析使用者的回饋問卷，與專家討論、研究是否能夠取代傳統的復健過程，並能以此為基礎持續地改進，推展出更多的產品面世。

五、問卷回饋

根據問卷調查最後的簡答題，發現尚有許多改進的地方。可配合使用者的年齡，改變遊戲難度，例如：入門的轉向或加速速度減慢、蘋果設定改為固定不動、增加第二關八爪機器人的抓取距離等等，讓玩家慢慢適應；也有建議遊戲可增加記憶性等內容，讓長者們在訓練手部的同時激發腦力，邊玩邊動腦；再來則是可配合使用者的需求，提供左、右手或不同手套大小的選擇，因為尺寸不合導致手套太鬆時，彎曲手指、傾斜手掌時，感測器可能會無法精確感應。

六、產品比較

			
名稱	MusicGlove Hand Therapy for PC/Mac	Dr.魔法手「識」	
價格	定價：約\$11000	成本：約\$1820	
感測器	力量感測器	彎曲感測器 / MPU-6050	
外型	指套式（有左／右+大／中／小）	手套式（新版為左手）	
遊戲	音符下沉時，在適當的時機捏碎	總共有三層：賽車、抓蘋果、射擊	

在搜尋資料的過程中，我們發現了 FlintRehab 推出了好幾款復健相關產品，而現有市面上的產品，因研發成本較高，市價皆高達上萬元。左圖產品主要使用力量感測器，透過施力導致電阻產生改變，可偵測介面的受力大小。該產品的外型運用手環式固定，連接裝於指尖的感測器，且購買時可選左、右手，與大、中、小三種規格。將 MusicGlove 戴好，傳輸線插進筆電或 Flint 平板，按下開始即可進行遊戲。在遊戲畫面中會有音符掉落，根據不同軌道，將大拇指的感測器與其他指的感測器互觸，即可捏碎音符。

而作為比較，我們的手套成本將近 2000 元，主要是因使用了三種感測，分別是彎曲感測器，以及 MPU-6050 的陀螺儀及加速度計，可偵測手指的彎曲、手掌的傾斜或揮動。款式因各版本的改良，皆將前代的感測器拆解，並繼續使用，因此只剩最新的左手版本。遊戲則設計了三層，對應不同的復健手勢：第一層的賽車遊戲，使用者透過傾斜手掌與彎曲手指，讓角色抵達終點進入下一層關卡；第二層的抓蘋果，讓使用者練習握拳，抓取蘋果累積一定數量進入下一層；第三層的射擊遊戲，則是類似保衛戰的模式，玩家同時彎曲大拇指與其餘四指之一發射飛彈，射擊單輪車來保衛 KODU 角色。

柒、結論

我們總共製作了四個版本的手套，分別在各代演進過程中，改變了控制鍵盤的方式與速率、整體的大小與靈活性，以及最後變為使用六軸感測器(圖 7-1)。



圖 7-1 手套演變

而我們最後所製作出的成品，能夠在感測器彎曲而使電阻改變時，判斷出手指的彎曲，發送指令給電腦。並通過當偵測到翻轉手掌，造成加速度感測器數值之變化，傳送程式中設定好的鍵盤指令，藉此來操控鍵盤按鍵。透過以上兩者，可配合我們設計的遊戲，來達到邊進行遊玩邊活動關節的效果。

在最後的研究過程中，我們邀請了 8 位長者來協助實測。透過這次的調查結果，我們得知了目前產品的不足之處，與值得改進的地方。期待未來能夠進行更多的實地測試，並在長期追蹤下觀察成效，讓遊戲可以與復健完美結合。

除增進復健過程的趣味外，我們還嘗試發展手勢識別的功能，例如手部的揮動等等，未來如可成功連接智慧家電，將可使長者們的生活更加便利。現在的成品結合復健、運動與遊戲，成為樂活新指標。未來也有機會能夠發展更多的手勢識別，與生活更進一步融合。

捌、參考資料及其他

一、Arduino. (2018). Language Reference

<https://www.arduino.cc/reference/en/>

二、Dejan Nedelkovski. (2015). Arduino Game Controller

<https://howtomechatronics.com/projects/arduino-game-controller/>

三、Barczewska, K. & Drozd, A. (2013). Comparison of methods for hand gesture recognition based on Dynamic Time Warping algorithm

https://annals-csis.org/Volume_1/pliki/448.pdf

四、EVELO. (2019). EXERCISES for Seniors

<https://www.evelo.com/exercises-for-seniors/>

五、Flint Rehab. (2018). 37 Hand Therapy Exercises to Improve Strength and Dexterity

[https://www.flintrehab.com/2018/hand-therapy-](https://www.flintrehab.com/2018/hand-therapy-exercises/?fbclid=IwAR14lsyzXW88cJPn7Ik3KGIUcqwk9bn7v7AIvEjrKRX7sxZ9a4vveesE56w)

[exercises/?fbclid=IwAR14lsyzXW88cJPn7Ik3KGIUcqwk9bn7v7AIvEjrKRX7sxZ9a4vveesE56w](https://www.flintrehab.com/2018/hand-therapy-exercises/?fbclid=IwAR14lsyzXW88cJPn7Ik3KGIUcqwk9bn7v7AIvEjrKRX7sxZ9a4vveesE56w)

六、Flint Rehab. (2018). 9 Expert Hand Exercises for Stroke Patients with Pictures

<https://www.flintrehab.com/2018/hand-exercises-for-stroke-patients-with-pictures/>

七、CALIFORNIA MOBILITY. (2018). 21 Chair Exercises For Seniors: A Comprehensive Visual Guide

<https://californiamobility.com/21-chair-exercises-for-seniors-visual-guide/>

八、Processing. (2018). Reference. Processing was designed to be a flexible software sketchbook.

<https://processing.org/reference/>

九、jrowberg. (2019). github. jrowberg/i2cdevlib.

<https://github.com/jrowberg/i2cdevlib>

十、今周刊(民 107 年 5 月 28 日)。**【特教老師也能教老人運動？國外長照創新的祕密在這！】**

<https://www.businesstoday.com.tw/article/category/151139/post/201805280040/%E7%89%B9%E6%95%99%E8%80%81%E5%B8%AB%E4%B9%9F%E8%83%BD%E6%95%99%E8%80%81%E4%BA>

%BA%E9%81%8B%E5%8B%95%EF%BC%9F%E5%9C%8B%E5%A4%96%E9%95%B7%E7%85
%A7%E5%89%B5%E6%96%B0%E7%9A%84%E7%A5%95%E5%AF%86%E5%9C%A8%E9%80
%99%EF%BC%81

十一、 中華民國內政部(民 107 年 4 月 10 日)。**【老年人口突破 14% 內政部：臺灣正式邁入高齡社會】**

https://www.moi.gov.tw/chi/chi_news/news_detail.aspx?type_code=02&sn=13723

十二、 中華民國衛生福利部國民健康署(民 87 年 3 月 27 日)。**【老人健康促進計畫】**

https://www.hpa.gov.tw/File/Attach/953/File_969.pdf

十三、 台灣健康運動管理協會(民 107 年 6 月 6 日)。**【肌力訓練對中老年人的重要性】**

<http://www.thfma.org.tw/archives/1706>

十四、 自由時報(民 107 年 6 月 28 日)。**【這個長照站好好玩 爺奶迷上「打電玩」】**

<https://news.ltn.com.tw/news/life/breakingnews/2472258>

十五、 林雁飛(民 99 年 10 月 8 日)。**【生活處處是復健】**

<http://www.uho.com.tw/sick.asp?aid=9323>

十六、 黃彥霖(民 102 年 10 月 23 日)。**【Arduino 入門教學：讀取 ADXL345 三軸加速度計】**

<http://lolikitty.pixnet.net/blog/post/165475299-arduino->

%E5%85%A5%E9%96%80%E6%95%99%E5%AD%B8%EF%BC%9A%E8%AE%80%E5%8F%96-
adx1345-%E4%B8%89%E8%BB%B8%E5%8A%A0%E9%80%9F%E5%BA%A6%E8%A8%88

十七、 葉難(民 102 年 9 月 18 日)。**【Arduino 練習：彎曲感測器 (flex/bend sensor)】**

<http://yehnan.blogspot.com/2013/09/arduinooflexbend-sensor.html>

附錄一：完整程式碼

```
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
  #include "Wire.h"
#endif
#include<Keyboard.h>
#include "I2Cdev.h"
#include "MPU6050.h"
#define OUTPUT_READABLE_ACCELYYRO
MPU6050 accelgyro;
int16_t ax, ay, az;
int16_t gx, gy, gz;
int circle1=0;
int circle2=0;
int grab=0;
int count=0;
int pinA0 = A0, pinA1 = A1, pinA2 = A2, pinA3 = A3, pinA4 = A4;
boolean key=false;
int t=0, t1=10;
int tgrab=0, t1grab=11;
int RealMode=0, Mode=0;
int temp=1;
void setup(){
  Serial.begin(38400);
  Keyboard.begin();
  Serial.println("Start : \n-----");
  Wire.begin();//初始化 I2C
  #if I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
    Fastwire::setup(400, true);
  #endif
  accelgyro.initialize();
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);
}
void loop(){
  int a = 1;
```

```

digitalWrite(8, HIGH);
digitalWrite(9, HIGH);
digitalWrite(10, HIGH);
digitalWrite(11, HIGH);
digitalWrite(12, HIGH);
int valA0 = analogRead(pinA0); //大拇指
int valA1 = analogRead(pinA1); //食指
int valA2 = analogRead(pinA2); //中指
int valA3 = analogRead(pinA3); //無名指
int valA4 = analogRead(pinA4); //小拇指
accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
#ifdef OUTPUT_BINARY_ACCELYGYRO
    Serial.write((uint8_t)(ax >> 8)); Serial.write((uint8_t)(ax & 0xFF));
    Serial.write((uint8_t)(ay >> 8)); Serial.write((uint8_t)(ay & 0xFF));
    Serial.write((uint8_t)(az >> 8)); Serial.write((uint8_t)(az & 0xFF));
    Serial.write((uint8_t)(gx >> 8)); Serial.write((uint8_t)(gx & 0xFF));
    Serial.write((uint8_t)(gy >> 8)); Serial.write((uint8_t)(gy & 0xFF));
    Serial.write((uint8_t)(gz >> 8)); Serial.write((uint8_t)(gz & 0xFF));
#endif
if (a == 1){
    Serial.print("彎曲感測器：");
    Serial.print(valA0);
    Serial.print(",");
    Serial.print(valA1);
    Serial.print(",");
    Serial.print(valA2);
    Serial.print(",");
    Serial.print(valA3);
    Serial.print(",");
    Serial.println(valA4);
    Serial.print("六軸感測器：");
    Serial.print(ax); Serial.print("\t");
    Serial.print(ay); Serial.print("\t");
    Serial.print(az); Serial.print("\t");
    Serial.print(gx); Serial.print("\t");
    Serial.print(gy); Serial.print("\t");
    Serial.println(gz);
}
if(az <- 10000){
    Mode=1; //手部平放

```

```

}
else if(az >- 10000){
    Mode=2;//手部直立
}
RealMode=Mode;
if(valA2 < 110 && valA0 > 110 && valA1 > 110){
    t++;//彎曲中指後倒數
    if(t > t1){
        key = !key;
        t=0;
    }
}
else{ t=0; }
if(key == false){
    grab=0;
    digitalWrite(LED_BUILTIN, LOW);
    Serial.println("==關機狀態==");// Mode 0
}
if(key == true){
    Serial.print("握拳次數：");
    Serial.println(grab);
    if(valA0 < 125 && valA1 < 110 && valA2 < 110 && valA3<110 && valA4<140
&& count == 0){
        tgrab++;//抓、握
        Keyboard.press(char(99));
        if(tgrab > t1grab){
            grab++;
            count=1;
        }
    }
    else if(valA0 > 125 && valA1 > 110 && valA2 > 110 && valA3<110 &&
valA4<140){
        count=0;
        Keyboard.release(char(99));
    }
    else{
        tgrab=0;
        Keyboard.release(char(99));
    }
    digitalWrite(LED_BUILTIN, HIGH);
}

```

```

Serial.println("====開機中====");//切換成 Mode 1
if(valA0 < 120 && valA1 < 110 && valA2 > 110 && valA3 > 110 && valA4 >
140){
    Keyboard.press((char)49);
    Serial.println("彎曲食指和大拇指");
}
else if(valA0 > 120 && valA1 > 110 && valA2 > 110 && valA3 > 110 &&
valA4 > 140){
    Keyboard.release((char)49);
}
if(valA0 < 120 && valA1 > 110 && valA2 < 110 && valA3 > 110 && valA4 >
140){
    Keyboard.press((char)50);
    Serial.println("彎曲中指和大拇指");
}
else if(valA0 > 120 && valA1 > 110 && valA2 > 110 && valA3 > 110 &&
valA4 > 140){
    Keyboard.release((char)50);
}
if(valA0 < 120 && valA1 > 110 && valA2 > 110 && valA3 < 110 && valA4 >
140){
    Keyboard.press((char)51);
    Serial.println("彎曲無名指和大拇指");
}
else if(valA0 > 120 && valA1 > 110 && valA2 > 110 && valA3 > 110 &&
valA4 > 140){
    Keyboard.release((char)51);
}
if( valA0 < 130 && valA1 > 110 && valA2 > 110 && valA3 > 110 && valA4 <
170){
    Keyboard.press((char)52);
    Serial.println("彎曲小拇指和大拇指");
}
else if(valA0 > 130 && valA1 > 110 && valA2 > 110 && valA3 > 110 &&
valA4 > 170){
    Keyboard.release((char)52);
}
if(valA0 < 120 && valA1 > 110 && valA2 > 110 && valA3 > 110 && valA4 > 140){
    Keyboard.press((char)97);//彎曲大拇指 按鍵 a
}
}

```

```

else if(valA0 > 120 && valA1 > 110 && valA2 > 110 && valA3 > 110 &&
valA4 > 140){
    Keyboard.release((char)97); //鬆開大拇指
}
if(valA0 > 120 && valA1 < 110 && valA2 > 110 && valA3 > 110 && valA4 >
140){
    Keyboard.press((char)98); //彎曲食指 按鍵 b
}
else if(valA0 > 120 && valA1 > 110 && valA2 > 110 && valA3 > 110 &&
valA4 > 140){
    Keyboard.release((char)98); //鬆開食指
}
if(ax < -4000){
    Keyboard.press((char)218); //方向鍵下
}
else if(ax >= -4000){
    Keyboard.release((char)218);
}
if(ax > 10000){
    Keyboard.press((char)217); //方向鍵上
}
else if(ax <= 10000){
    Keyboard.release((char)217);
}
if(ay > 9000){
    Keyboard.press((char)216); //方向鍵左
}
else if(ay <= 9000){
    Keyboard.release((char)216);
}
if(ay < -9000){
    Keyboard.press((char)215); //方向鍵右
}
else if(ay >= -9000){
    Keyboard.release((char)215);
}
temp=0;
if(temp == 0){
    if(gy > 10000){
        Serial.println("<手勢向上>");
    }
}

```

```

        if(circle1 == 0){ circle1++; }
        if(circle2 == 0){ circle2++; }
        delay(250);
        temp=2;
    }
    else if(gy < -10000){
        Serial.println("<手勢向下>");
        if(circle1 == 2){ circle1++; }
        if(circle2 == 2){ circle2++; }
        delay(250);
        temp=2;
    }
    else{ temp=1; }
}
if(temp == 1){
    if(gz > 10000){
        Serial.println("<手勢向右>");
        if(circle1 == 1){ circle1++; }
        if(circle2 == 3){
            Serial.println("<手逆時針轉動>");
            circle2=0;
        }
        delay(250);
    }
    else if(gz < -10000 ){
        Serial.println("<手勢向左>");
        if(circle1 == 3){
            Serial.println("<手順時針轉動>");
            circle1=0;
        }
        if(circle2 == 1){ circle2++; }
        delay(250);
    }
}
}
delay(250);
}

```

附錄二：問卷題目

一、 年齡、經歷調查

(一)你的年齡介於？

- 35~44 歲
- 45~54 歲
- 55~64 歲
- 65 歲以上

(二)你/妳是否曾經或現在需要接受復健？

- 是，現在
- 是，曾經
- 否

二、 遊戲體驗

(三)使用過程中，你/妳覺得遊戲有趣嗎？

1. 非常無趣 ~ 5. 非常有趣

(四)在使用的過程中，你/妳覺得遊戲的難易度？

1. 非常困難 ~ 5. 非常簡單

(五)你/妳喜歡活動手腕與手指完成遊戲任務帶來的成就感嗎？

1. 非常不喜歡 ~ 5. 非常喜歡

(六)你/妳想再多嘗試幾次遊戲，讓遊戲成績更好嗎？

1. 非常不想 ~ 5. 非常想

(七)依照使用後的感想，你/妳覺得自己多久會想使用一次？

- 半天
- 一天
- 一週

三、 感測器靈敏度

(八)在使用的過程中，你/妳覺得手套在「彎曲手指抓握」動作的接收是否靈敏？

1. 非常不靈敏 ~ 5. 非常靈敏

(九)在使用的過程中，你/妳覺得手套在「手掌傾斜」動作的接收是否靈敏？

1. 非常不靈敏 ~ 5. 非常靈敏

四、 產品支持度

(十)整體而言，你/妳覺得這項產品，對於復健過程是否有幫助？

1. 完全沒有 ~ 5. 非常有幫助

(十一)倘若這項產品能夠在世面上普遍應用，你/妳對於此產品的支持度為何？

1. 非常不支持 ~ 5. 非常支持

五、 整體回饋

(十二)你/妳之前是否使用過相關手勢的器材？

- 是 (→ 十二之一)
- 否 (→ 十二之二)

(十二之一)與其他產品之間的差異，你/妳覺得這個手套比較好的地方是？

(十二之二)在體驗結束後，你/妳對於整個產品有什麼樣的回饋與建議？

【評語】 052301

此作品主要是製作穿戴手套之手勢辨識系統，應用情境設定於老人手指復健，以電動遊戲操控。作者以 Arduino IDE 編寫體感手套的相關程式，創意點在於搭配 Scratch 與 KODU 設計之遊戲，期望能讓長者在活動與遊玩過程中，達到復健運動的目的。應用於老人復健的電動遊戲操控，也有問卷調查，不過老人復健與操控遊戲的關聯性或必要性並未在作品中呈現，建議進一步結合復健醫學的觀點以及實際需求，對於系統效益做量化評估，並對於系統優化進一步量化說明。綜言之，本作品在技術上具有製作及設計能力，雖然在老人復健的實用性尚難評估，但可以發展手勢識別的功能，連接智慧家電，使長者們的生活更加便利。

作品海報

摘要

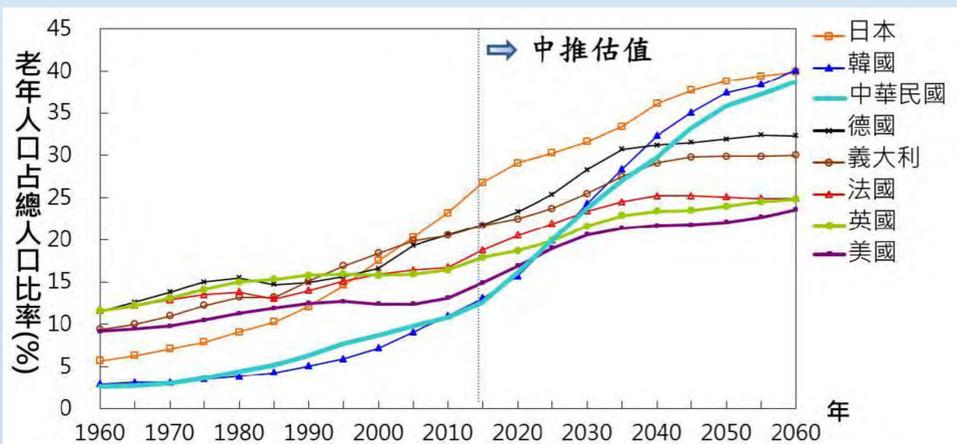
臺灣老年人口比率逐年攀升，老人照護逐受重視，而運動對於健康保健更是不可或缺。因此我們想製作一產品，結合時下流行的體感控制，也就是以穿戴手套的方式操控遊戲，讓復健同時兼具娛樂，活動筋骨，一舉多得。

手部與手指的運動分別會被我們所設置的三軸加速度感測器與彎曲感測器偵測，並以類比訊號的形式透過傳輸線傳送對應的鍵盤按鍵訊號至電腦，以此可操縱多種透過鍵盤遊玩的遊戲達到增進使用之樂趣與意願。

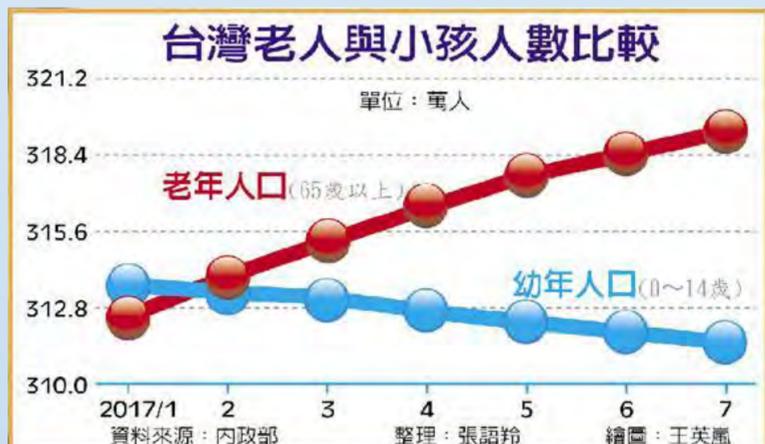
使用Arduino IDE編寫程式之手套，搭配Scratch與KODU設計之遊戲，邀請8位長者體驗，並以問卷方式進行回饋與分析。多數長者在活動與遊玩過程，認為能增加其樂趣以及使用之意願。希望透過本次研究，使復健中的運動能夠翻轉成新世代的模樣！

壹、研究動機

根據內政部統計顯示，臺灣老年人口比率已達13.33%，預估在2025年將攀升至20%，且持續上升中，如圖1-1所示。現代中，樂活人生已成為一股風潮，但我們觀察家中的長輩們發現在日常生活中，常常艱因地控制手部、因無力而需攙扶或協助提起重物。查詢相關資料的過程中，發現運動復健過程往往沉悶又無趣。為解決此觀察到的問題，我們因而想做出一項產品，能結合復健、運動與遊戲的手套，讓過程中增添樂趣。



各國未來老年人口比率逐年攀升



臺灣老人與小孩人數比

貳、研究目的

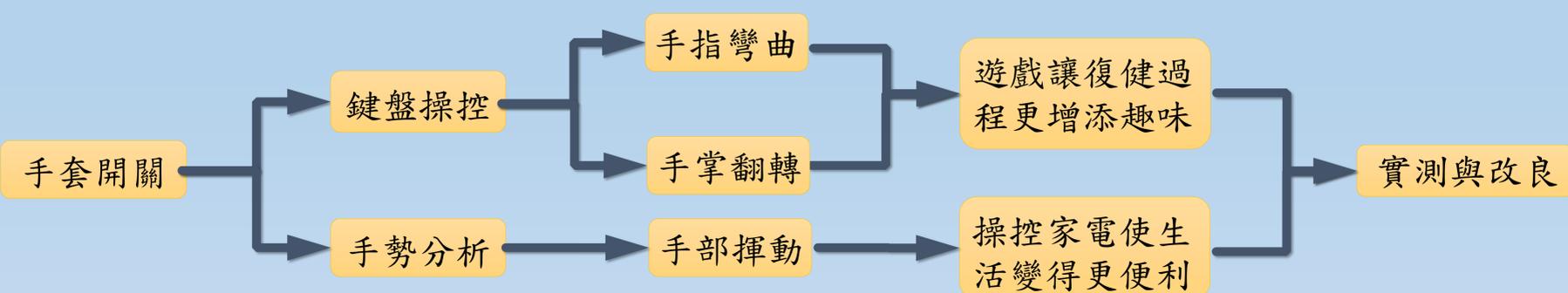
復健的過程大多單調，重複同樣的動作使人覺得乏味；「我討厭跌倒(I HATE FALLING)」一詞為長者們應注意的跌倒原因之縮寫，其中的L代表體能不足(Lack of conditioning)；根據2007年國民健康署「健康行為危險因子監測調查」統計顯示，過去兩個禮拜有運動習慣的老人僅佔52.3%。由於許多研究皆表明運動對於長者的重要性，因此我們研究目的如下：

一、以Arduino製作可辨識手勢的手套

以手部為基礎開始構想，使用感測器與Arduino開發版的結合，透過回傳的數值進行判斷，分析出使用者的手部姿勢、動作。

二、以完成之輔具提高長者使用之樂趣與意願

使用輔具以模擬復健手勢方式進行自行設計之電腦遊戲，使長者於運動與復健過程中感到樂趣，進而期望提高長者們或患者們活動身體之意願。



參、研究設備及器材

Genuino 101	Arduino Leonardo	Arduino Micro	ADXL 345	MPU-6050	Flex Sensor
Arduino	Processing IDE	Fritzing	Scratch 2.0	KODU遊戲軟體	

肆、研究過程及方式

手套演變／硬體設計



手套總共分成四個版本，每次更新中，嘗試不同開發版或感測器，讓手套更完善。每次改良後，使用KODU或Scratch 2設計遊戲，來進行功能實測，長者使用手套時，可以搭配使用，結合遊戲讓過程更有趣。



程式功能／軟體設計

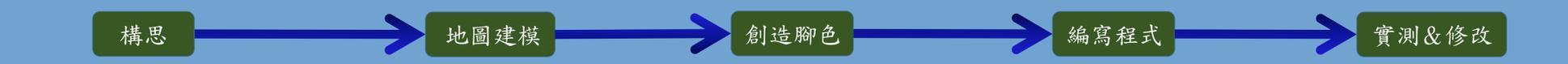


目前手套可辨識的手部動作可分為6種，分別是握拳、傾斜、畫圓等等（如上圖）。

- 一、加速度感測
偵測手掌的左、右傾斜。
- 二、陀螺儀感測
偵測手掌直立時，上、下、左、右的揮動或順、逆時針畫圓。
- 三、彎曲感測
偵測手指的彎曲、握拳等狀態。其中兩根彎曲代表大拇指與其餘四指之一同時彎曲；個別彎曲代表大拇指與食指的彎曲。
- 四、開關功能
為了避免使用者戴上手套時，因傾斜或晃動，產生不必要的指令，所以設計開關功能，當關機時彎曲中指2.5秒便會開機，開機時亦如此。在關機的時候停止輸出鍵盤指令。



遊戲設計／實作與評估



一、測試功能

第一層測試加速度感測的左、右傾斜，大拇指與食指的彎曲。

二、活動關節

第二層增加握拳手勢；第三層參考復健手勢。

一、選擇材質、刷具

有各種材質可以選，再透過選擇不同刷具便可做出地形。

一、選擇角色

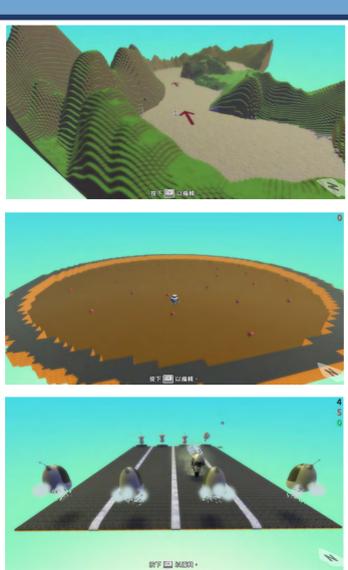
同樣遊戲中具有多樣的內建角色可選擇，每個都擁有不同特性，根據所需來選擇。

二、創造物件

除了角色外還有各種物件可選擇。

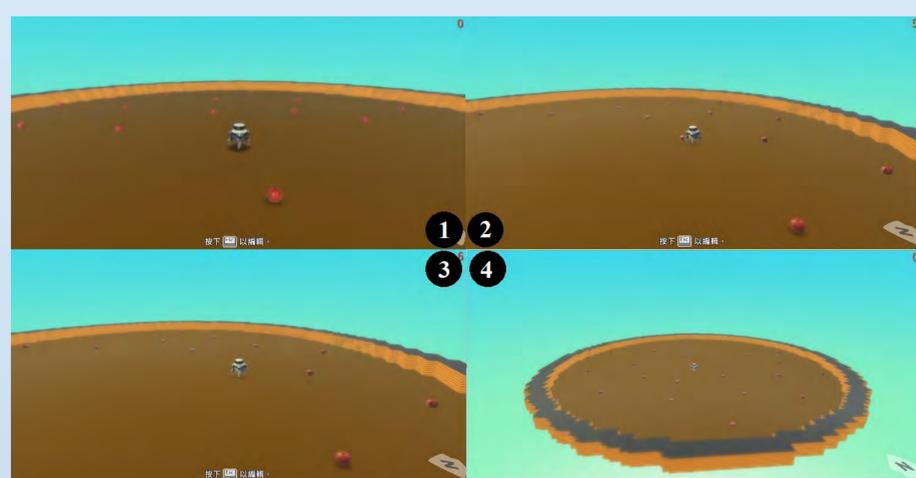
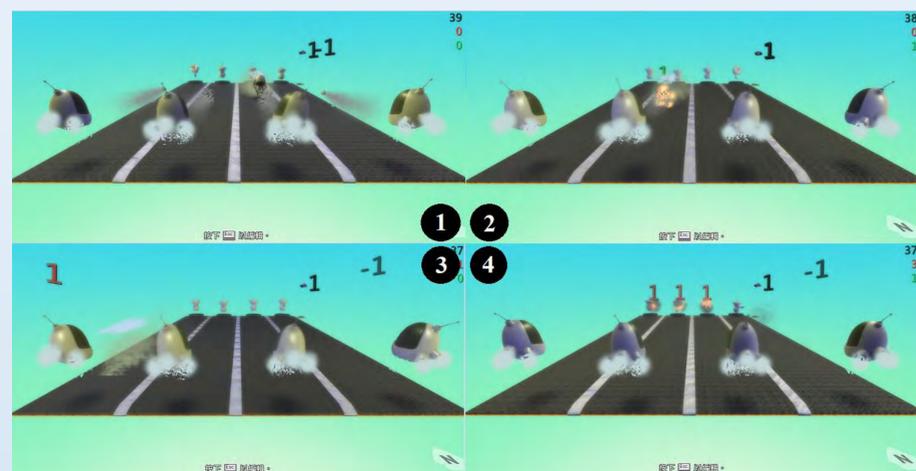
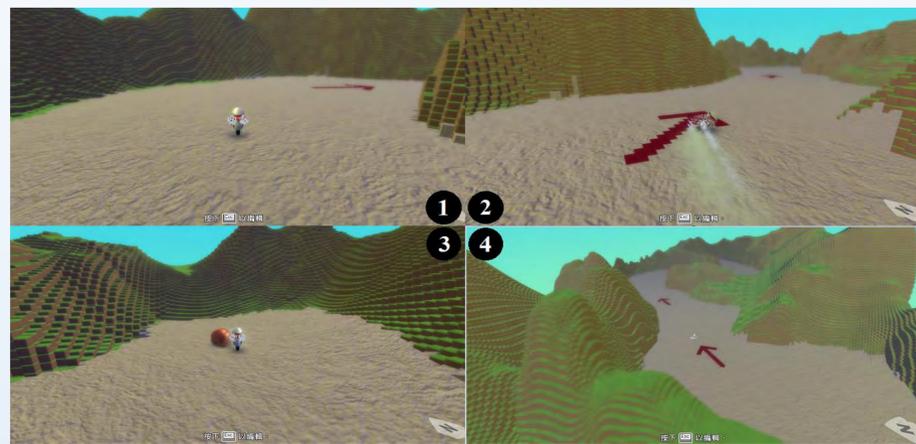
一、編排程式

根據需求，可對角色或物件編排程式，讓角色或物件做出反應。

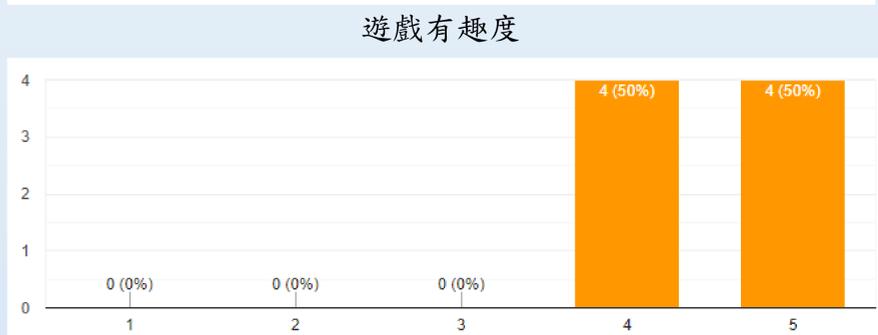
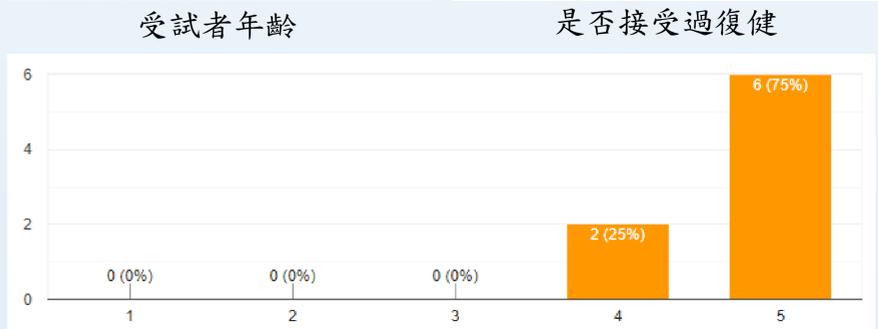
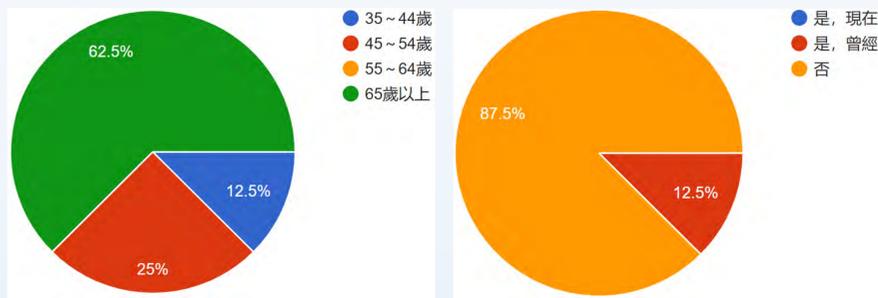


伍、研究結果

一、遊戲介紹



二、問卷調查



對復健是否有幫助

個案質性回饋

1. 使用後手很痠，眼睛花。
2. 手套輕量化，且有左、右手或不同大小規格給使用者選擇，如果不合手，彎曲時可能沒辦法偵測。
3. 遊戲依照不同年紀分成不同難度，或讓使用者依熟練度自由選擇。
4. 遊戲中，第一、二層角色轉向太快；第三層可以讓子彈連射。
5. 蘋果的滾動很有趣，能去追才有樂趣。
6. 手套與電腦的連接電線，可能纏繞發生危險。

陸、討論

一、外觀改善

繪製更精細的草圖，以方便製作，並自行焊接線路，取代杜邦線和麵包版。

二、軟硬體更新

增加藍芽，讓手套可以無線連接；或在Arduino的程式中使用不同的指令，使手套有更多功能，與生活更緊密地結合。

三、結合智慧家庭生活

透過物聯網的概念，讓行動不便的使用者可以藉由手套，進行遠端的操控。

四、輔具的延伸發展

可以結合復健和遊戲，使復健過程不再那麼單調乏味，增加復健對於患者的吸引力。



名稱	MusicGlove Hand Therapy	Dr.魔法手「識」
感測器	力量感測器	彎曲感測器 / 六軸感測器
外型	指套式	手套式
遊戲	音符下沉時，在適當的時機捏碎	自製遊戲 / 對應鍵盤操作遊戲
價格	定價：約\$11000	成本：約\$1820

柒、結論

一、製作出能準確辨識動作的手套

成品能在感測器彎曲使電阻改變時，判斷出手指的彎曲，發送指令給電腦。並通過當偵測到翻轉手掌，造成加速度感測器數值之變化，傳送程式中設定好的鍵盤指令，藉此來操控鍵盤按鍵。

二、遊戲式復健姿勢活動

讓手套搭配遊戲一起使用，設計復健手勢動作關卡，達到邊遊玩邊活動關節的效果。每層不同的遊戲方式，除增進活動過程中的趣味外，還發展手勢識別的功能。

三、進行測試並填寫問卷

總共了8位受測者進行測試，並填寫問卷，透過選擇與問答題，收集數據並給予我們回饋。結果對於在長者活動上有帶來初步效益。