

110年海青工商  
資訊科  
資訊三善專題報告

AB Game機器人

學生

組長: 林承龍

組員: 薛聿辰

組員: 楊博全

組員: 黃宥翔

指導老師: 漆慶福 老師

2022年 5月4日

# 分工表

**軟體:**

18黃宥翔

**硬體:**

04林承龍

28薛聿辰

**簡報:**

21楊博全

## 摘要

AB Game是數字遊戲, 玩法是出數字的人要先想好一個沒有重複的4位數, 不能讓猜的人知道, 猜的人就可以開始猜。每猜一個數, 出數者就要根據這個數字給出幾A幾B,

其中A前面的數字表示位置正確的數的個數, 而B前面的數字表示數字正確而位置不對的數的個數。

例如正確答案為 2017, 那麼如果猜的數字為 1056, 關係便是 1A1B。遊戲的主要目的是藉由最少的猜測次數求出正確的四碼數字。

關鍵字-----ABgame、語音辨識、AI

# 目錄

封面.....	I
分工表.....	II
目錄	
.....	
.....	III
第1章 概要	
.....	
.....	1
1-1研究動機	
.....	1
1-2製作背景及目的.....	
.....	1
1-3預期成果	
.....	
..	2
第二章	理論探討
.....	3
.....	2-1研究理論
.....	4
.....	2-2研究理論
.....	6
第三章	專題設計
.....	8

	3-1研究理論	
.....		9
	3-2研究理論	
.....		18
第四章	結論	
.....		
.25		
	4-1研究理論	
.....		25
	4-2研究理論	
.....		25

## 圖表目錄

表	
1-3.....	
.....	2
圖	
2-1-1.....	
.....	4
圖	
2-1-2.....	
.....	4
圖	
2-2-1.....	
.....	7



2-2-2.....  
.....8



3-1.....  
.....9



3-2-1.....  
.....10



3-2-2.....  
.....11



3-2-3.....  
.....11



3-2-4.....  
.....12



3-2-5.....  
.....12



3-2-6.....  
.....12



3-2-7.....  
.....13



3-2-8.....  
.....14



3-2-9.....  
.....15



3-2-10.....  
.....16



3-2-11.....  
.....17

# 第1章 概要

## 1-1研究動機:

小時候與朋友玩遊戲，有時候會玩到「猜數字」，這是一個既好玩又需要動一點腦筋的遊戲。只需要一張紙和一隻筆，無論身處何處，便可以和朋友們進行這個遊戲。

以前，這個遊戲是需要兩個人才可以玩，但現在是個資訊發達的時代，經過我們的改良，把此遊戲加入了現代科技，在們學了一點Python程式設計後，便結合在機器人裡。

最後將遊戲的程式配合機器人的語音功能，就能用遊戲的方式跟機器人互動，讓此遊戲更有科技感，且就算只有一個人獨處時，也能跟機器人進行遊戲。

## 1-2研究背景及目標:

先前有個很紅的網路遊戲叫神燈精靈，他就能很準確的猜到你在想的任何人或任何東西。

我們組做的AB Game篩選法也是差不多的概念，目的是將此遊戲結合語音機器人，做出一個能夠以遊戲互動的機器人。

### 軟體方面:

我們組共研究了

- 一、如何在最少的次數內找到正確的排列組合，並找出能夠必勝的方法，讓有興趣的人可以沉浸在勝利的喜悅之中。
- 二、如何讓電腦猜的次數保持最多的答案組合。
- 三、了解電腦程式解題過程。
- 四、比較多種不同情況的結果。

硬體方面：

在硬體我們組研究所花的時間比較少

需利用樹梅派結合麥克風跟喇叭，將聲音輸入到樹梅派後，經過google的語音程式處理，再由語音轉成文字到電腦上執行式，並將結果輸出在螢幕或喇叭上，就能達到互動的功能。

### 1-3專題設備、製作

軟體製作：

使用程式是python

並使用Tkinter函式庫做成gui介面

語音辨識則是用Speech\_recognition函式庫

藉由google語音來辨識。

硬體製作：

先畫設計圖，然後將木頭切成想要的大小，之後組裝，最後將其釘牢。

表(1-3)使用的主要軟體/材料設備清單

				
樹梅派3	喇叭	麥克風	木板	螢幕
				
HDMI轉接線	行動電源	Speech_recognition	Tkinter	Python

## 第二章 理論探討

設計一個猜數字遊戲，該遊戲可以由程式隨機產生整數。用戶隨機輸入數字，當有錯誤時就會提醒錯誤的個數。

例如正確答案為 1234，那麼如果猜的數字為1243，關係便是 2A2B。2A表示用戶數字跟位置都正確，2B表示用戶猜對2個數但數字位置是錯的。遊戲的主要目的是藉由最少的猜測次數求出正確的四碼數字。

當與電腦產生的亂數結果符合時，遊戲就會提醒用戶正確的訊息(4A0B)。

每次執行完程式後，程式會將結果轉成語音播出來。

## 2-1研究理論

### (一)出題數字為不重複的亂數:

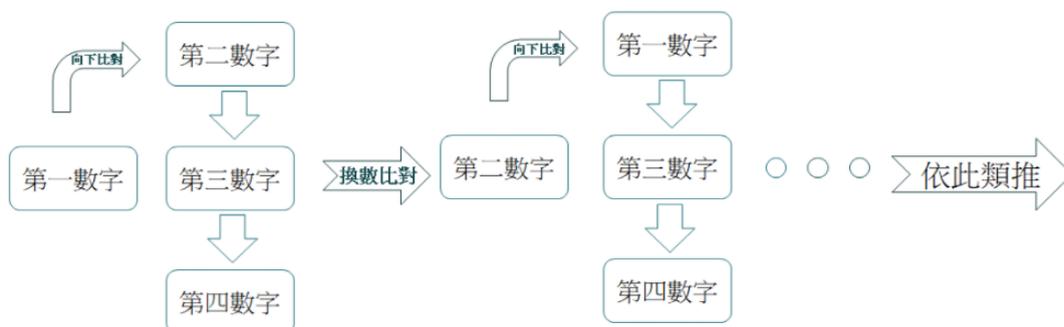
為了使出題數字為不重複得亂數，程式是以類似撲克牌洗牌的方式，來隨機對調 0~9 的數字，因為原先設定的每張牌內文字本就不重覆，因此以這種方式來出題，會重複的機率是零。

下圖2-1-1為此演算原理的示意圖



### (二)檢查玩家數字不重複:

為了檢查數字是否重複，我們採用的是比對的方式  
用第一格數字和第一、二、三、四個數字比對檢查是否重複，依此類推，如下圖2-1-2



### (三)AB 判斷法

當輸入的數字中，有數字跟提示中的數字一樣時，電腦會先判斷位置是否正確，若位置及數字都正確，A 就會加一，否則 B 會加一，最後將判別的 AB 顯示在猜測結果，以便使用者猜測正確答案。當 A 等於 4 的時候，就代表為正確答案。

以下為範例：

若正確答案為 **1234**

輸入**5678** 猜測結果**0A0B**

輸入**5143** 猜測結果 **0A2B**

輸入**5134** 猜測結果**2A0B**

輸入**3412** 猜測結果**0A4B**

輸入**1234** 猜測結果**4A0B**

(跳出提示“恭喜猜中”並顯示答案)

## 2-2程式理論

### (一)程式演算法:

#### 1.人猜電腦(電腦防守)

建立系統資料庫

**Step 1**→系統利用撲克牌原理組出一組四字隨機亂數。

**Step 2**→玩家猜測亂數並用語音輸入猜測數字。

**Step 3**→系統判斷字數總數是否為四。

\* (如為否, 回到**Step 2**, 並提醒字數不對)

**Step 4**→系統判斷數字是否有重複

\* (如為是, 同樣回到**Step 2**, 並提醒數字重複)

**Step 5**→系統判斷對應位置的數字是否相同。

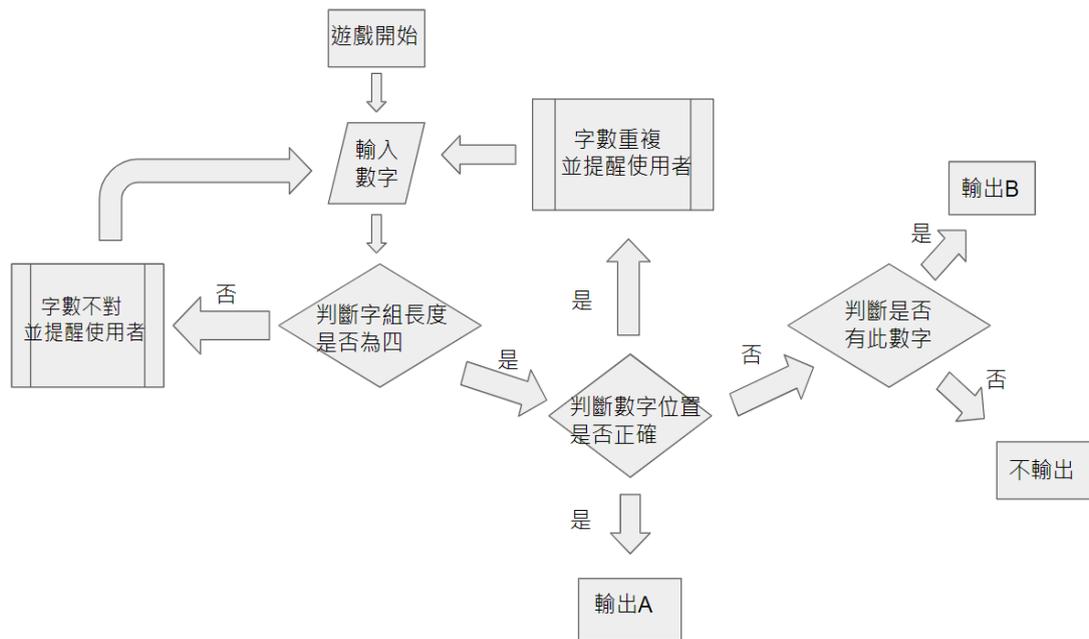
**Step 6**→系統判斷是否此數字。

\*每猜中正確數且在正確位置會輸出A且X A前面數字累加

相對猜中正確數但錯誤位置會輸出B且X B前面數字累加

如果都沒猜中輸出0A0B

**Step 7**→重複 **Step 3-7** 直到玩家猜到系統出的隨機亂碼才結束。



## 1. 電腦猜人(電腦攻擊)

**Step 1**→系統從已建立好的隨機抽出一組四字隨機亂數。

**Step 2**→用此亂數猜測並以玩家提供的資訊作篩檢。

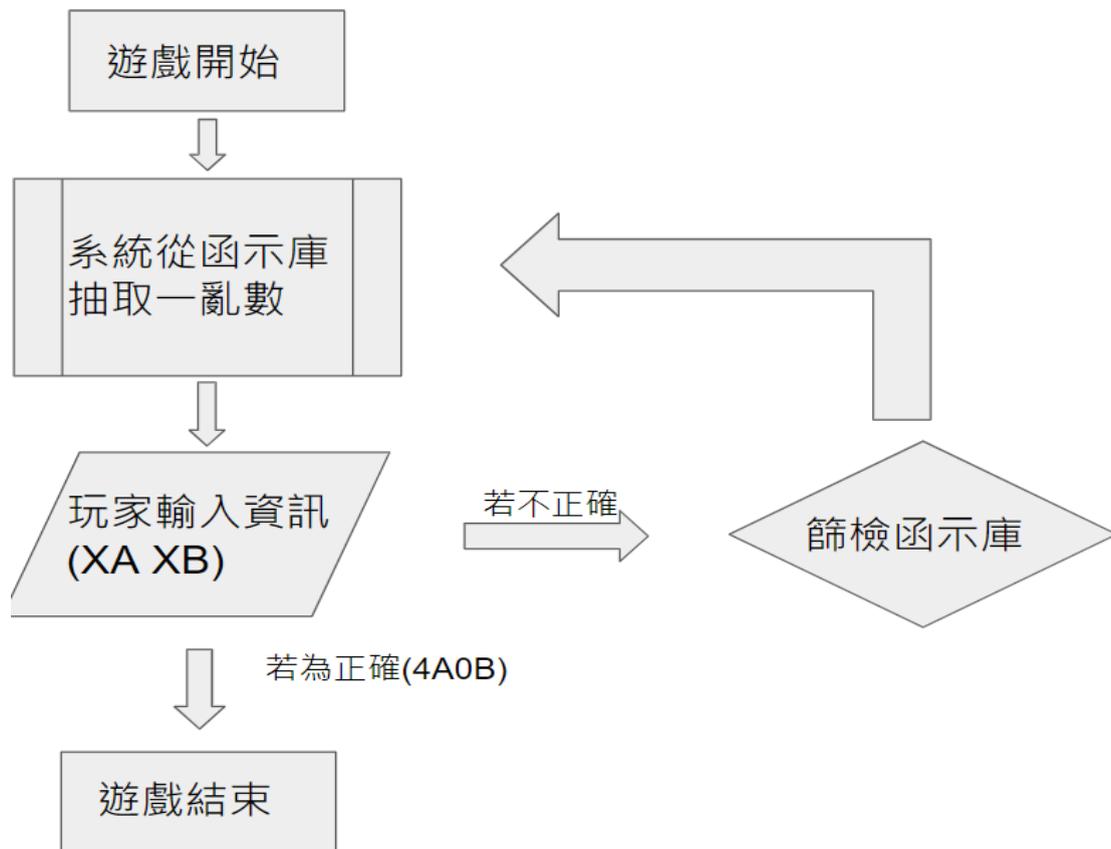
\*(系統從玩家提供資訊來刪去不可能的答案)

**Step 3**→系統篩檢完函示庫後重新提出新亂數。

\*(設計者能決定電腦猜測次數)

**Step 4**→系統重複Step1-3直到猜測次數用進或猜出答案才結束。

以下為程式流程圖2-2-2



# 第三章 專題設計

## 3-1 研究過程:

### (一)、研究過程

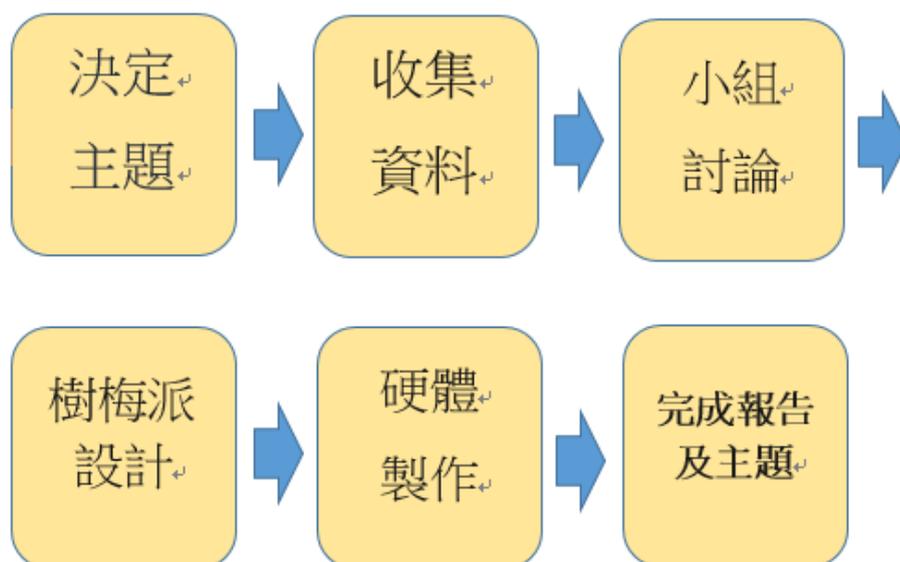
#### 1.研究過程

下圖是我們的研究方法，首先是做專題前討論，討論後決定的控制器是樹梅派。

再來是軟體設計，除了要設計出AB Game跟語音辨識外，因為樹梅派的程式是用之前沒學過的Python，所以在軟體方面花費了比較多的時間。

再來就是硬體設計，決定好硬體擺設之後再設計出外觀，當材料都準備齊了最後才做組裝，完成作品。

圖(3-1)流程圖



## 二、研究設備 本組以下列設備完成專題研究

### (一)、硬體配件:

#### 1.樹梅派:



圖3-2-1

樹莓派(英語:**Raspberry Pi**)是基於Linux的單晶片電腦。

樹莓派每一代均使用博通(Broadcom)出產的ARM架構處理器,如今生產的機型記憶體在2GB和8GB之間,主要使用SD卡或者TF卡作為儲存媒體,配備USB介面、HDMI的視訊輸出(支援聲音輸出)和RCA端子輸出,內建Ethernet/WLAN/Bluetooth網路鏈結的方式(依據型號決定),並且可使用多種操作系統。產品綫型號分為A型、B型、Zero型和ComputeModule計算卡。

Raspberry Pi OS是所有型號樹莓派的官方作業系統,樹莓派基金會網站也提供了Ubuntu MATE、Ubuntu Core、Ubuntu Server、OSMC等第三方系統供大眾下載。

#### 2.揚聲器:



圖3-2-2

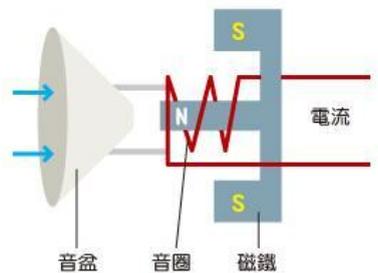


圖3-2-3

通常揚聲器就是指音響喇叭，由單體和箱體、防塵罩、分頻器等配件組成一個音響喇叭；單體則是揚聲器、喇叭中的一個重要零件，英文是 driver。

一個單體主要是由電磁鐵 (Magnitic)、線圈 (Voice Coil)、振膜 (Diaphragm, 通常是紙盆) 所組成。當電流 (從放大器出來的音頻訊號) 通過線圈產生電磁場，磁場的方向為右手法則。假設播放C調，其頻率為256Hz，即每秒振動256次，則輸出256Hz的交流電，每秒256次電流改變，發出C調頻率。當電流通過線圈，線圈即隨著電流的頻率振動，而和線圈相連的振膜當然也就跟著振動。推動周圍的空氣振動，由此產生聲音。



圖3-2-4

麥克風是一種將聲音轉換成電子訊號的換能器。正式中文名為傳聲器。

麥克風原理是利用聲波震動內部的振膜，振膜的振動會進而產生電流。進入錄音界面後將電流訊號轉為類比訊號，利用放大機放出聲音。目前常見麥克風可分電容式麥克風、動圈式麥克風等。因為收音原理不同，電容式較動圈式靈敏許多，也因此，電容式較適合用在非常安靜、只專注於同一音源的錄音室錄音，動圈式較適合使用在可能會有許多額外噪音的現場演出。

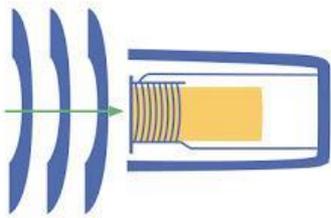


圖3-2-5:動圈式麥克風:包含線圈、振膜、永久磁鐵三部份,根據法拉第定律以及冷次定律,線圈會產生感應電流。

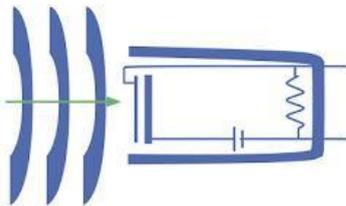


圖3-2-6:電容式麥克風:利用電容兩片隔板間距來改變電壓變化。

#### 4.顯示器:



圖3-2-7

液晶顯示器的原理比較複雜，主要是利用前後偏光片與前後導電璃之間的液晶分子改變極化光的極化方向，造成每一個畫素亮(白)與暗(黑)的變化，再利用彩色濾光片產生紅(R)、綠(G)、藍(B)三種不同顏色排列組成我們所看到的影像，我們先簡單說明液晶顯示器相關零組件的功能：

➤背光模組：背光模組包括光源、反射板、導光板等元件組成，使光源均勻分布在整個液晶顯示器的畫面上，看起來整個畫面亮度相同。

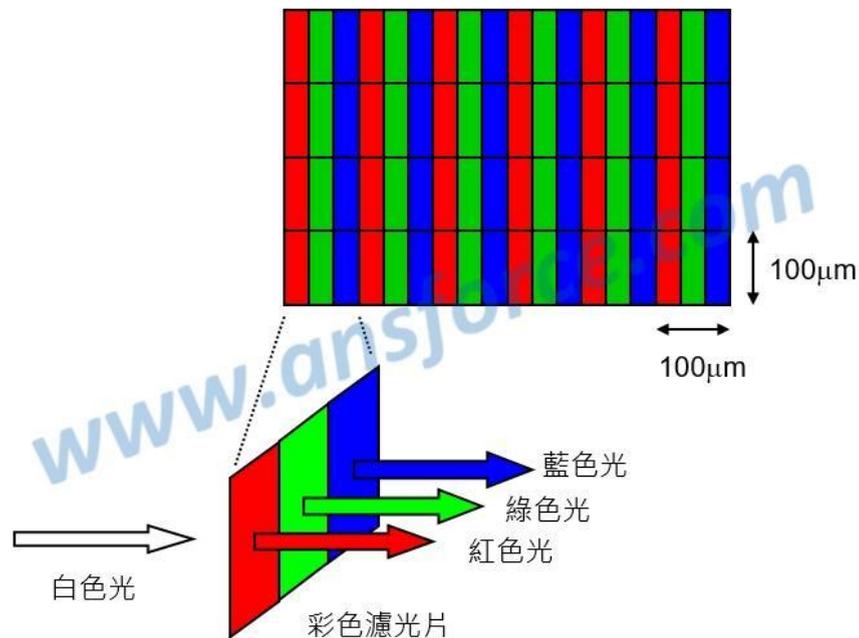
➤後偏光片：由於光源發出來的白光為「非極化光」，後偏光片主要的目的在使非極化光變成「極化光(Polarized light)」。

➤後導電玻璃：在玻璃基板上使用濺鍍法(Sputter)成長「氧化銦錫(ITO: Indium Tin Oxide)」形成可以導電的玻璃，稱為「導電玻璃」。氧化銦錫(ITO)是一種陶瓷(金屬氧化物)，幾乎所有的陶瓷都是絕緣體，但是科學家發現氧化銦錫(ITO)不但可以導電，而且在厚度很薄的時候，還是透明的，故稱為「透明電極」。

➤薄膜電晶體：在導電玻璃的上面使用半導體製程技術成長「開關元件」，最簡單的開關元件就是「CMOS」，但是CMOS必須具有金屬、氧化物、半導體的結構，必須成長在矽晶圓上才行，要在導電玻璃上成長開關元件沒辦法使用CMOS，因此必須另外設計一種開關元件，它的工作原理和「CMOS」很像，我們稱為「薄膜電晶體(TFT: Thin Film Transistor)」，使用薄膜電晶體(TFT)製作的

➤ 前導電玻璃：與後導電玻璃相同。

➤ 彩色濾光片：在塑膠薄片上塗佈紅(R)、綠(G)、藍(B)三種不同顏色的顏料，不停地反覆排列在顯示器的整個畫面上，稱為「彩色濾光片(Color filter)」。彩色濾光片的原理如〈圖一〉所示，當白光(紅、綠、藍的混合光)通過「紅色的濾光片」，則只有紅光可以通過，綠光、藍光被吸收，所以眼睛只看到紅光；當白光通過「綠色的濾光片」，則只有綠光可以通過，紅光、藍光被吸收，所以眼睛只看到綠光；當白光通過「藍色的濾光片」，則只有藍光可以通過，紅光、綠光被吸收，所以眼睛只看到藍光，「濾光片」其實就是濾掉我們不要的顏色，只讓我們想要的顏色通過



〈圖3-2-8〉

➤ 前偏光片：前偏光片主要的目的在決定是否要讓旋轉後的極化光通過，如果可以通過則眼睛看起來是「亮(白)」，如果無法通過則眼睛看起來是「暗(黑)」

(二)、軟體配置：

## 1. Python :



圖3-2-9

**Python**是一種廣泛使用的直譯式、進階程式、通用型程式語言。Python支援多種程式範式，包括物件導向、結構化、指令式、函數式和反射式程式。它擁有動態型別系統和垃圾回收功能，能夠自動管理記憶體使用，並且其本身擁有一個巨大而廣泛的標準庫。

### Python 的特色包含

1. **High level interaction** 直譯式高階語言，可立即呈現與試驗執行結果
2. **Readable and intuitive** 語法直覺簡潔易理解，規則強制需換行、縮排
3. **For General purpose** 研究單位用於學術研究、自動控制、金融等泛用場域
4. **Free and open-source library** 跨平台、開放完整、豐富的資料分析、AI函式庫生態系

## 2. Python Tkinter:



圖3-2-10

Tkinter 支援物件導向撰寫方式，可以將程式碼重構成物件導向撰寫方式。比起 CLI 來說 GUI 對於一般使用者在使用上更方便。

建立Tkinter視窗有幾個步驟：

1. 匯入 Tkinter 模組。
2. 建立主視窗。
3. 建立視窗控制項元件。
4. 進入事件處理迴圈以處理不同的視窗訊息。



圖3-2-11

## 語音辨認技術(Speech Recognition)

是一種經過讓機器經過辨認和了解的過程，把人類的語音信號轉變為相應文本的技術。

看起來強大上的語音辨認技術，原理並沒有多麼複雜。不管是微軟家的Cortana、三星家的S-voice蘋果家的Siri，還是國內一些獨立做語音辨認的比方訊飛、Rokid，在原理在實質上沒有幾差別：就是語音輸入後，停止特徵提取，將提取的特徵值放進模型庫里，再不時地停止鍛鍊和匹配，最終解碼得到結果。

所謂鍛鍊就是指依照一定的原則，從大量已知語音形式中獲取一個最具特徵的模型參數。而形式匹配則相反，是依據一定原則，將未知語音形式與模型庫中的某一個模型取得最佳匹配。

最後的解碼過程又能夠分紅動態解碼網絡和靜態解碼網絡兩種：動態網絡會編譯一個狀態網絡並構成搜索空間，把單詞轉換成一個個的音素後將其依照語序拆分成狀態序列，再依據音素上下文分歧性準繩將狀態序列停止銜接。

而靜態網絡普通是針對一些特殊詞(孤立詞)的辨認網絡，它的構造就簡單多了：先將每條特殊詞擴展成HMM狀態序列，然後再計算得分，選擇得分最大的作為辨認輸出結果。由於靜態網絡是依據聲學機率計算權重，不需求查詢言語模型機率，因而解碼速度很快。

# 1. 語音程式:

## (一) 接收音源

```
def Recognition(self) -> str:
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("請說話")
        audio = r.listen(source)
        /**錄音*/

    s=""
    try:
        s = r.recognize_google(audio,language='zh-CN')
        /**辨識哪種語言(註:因為沒有zh-TW版)*/

    except sr.UnknownValueError:
        /**如果使用者沒有說話*/

        print("Unknow Value")
        return 00

    except sr.RequestError as e:
        /**Google語音服務沒有回應時*/
        print("語音辨識服務錯誤; {0}".format(e))
        s = s.upper()

    return s
```

## (二) 語音辨識:

```
def PlayVoice(self,speech : str):
    if(speech==""):
        return
    tts = gTTS(speech, lang='zh-TW')
        /**語音轉文字*/
        path = 'Buff'+
str(datetime.now().strftime('%Y-%m-%d-%H-%M-%
S'))+'.mp3'
    tts.save(path)
    mixer.init()
    mixer.music.load(path)
    mixer.music.play()
    if(len(speech)>=4):
        time.sleep(0.5)
    else:
        time.sleep((5-len(speech))*0.5)
    while(mixer.get_busy()):
        continue
```

## 2.猜電腦(電腦防守):

(一)隨機產生一組不重複亂數

```
def number(lst):
```

```
    while len(lst) < 4:
```

```
        lst.append(randint(1,9))
```

```
    if len(lst) > 1:
```

```
        x = 0
```

```
        while x < len(lst)-1:
```

```
            /**去除重複數字*/
```

```
                if lst[x] == lst[len(lst)-1]:
```

```
                    del lst[len(lst)-1]
```

```
                else:
```

```
                    x += 1
```

```
    final_ans = ""
```

```
    for i in range(0,len(lst)):
```

```
        final_ans = final_ans + str(lst[i])
```

```
    return final_ans
```

(二) 比對玩家的答案是否正確並回答**AB**值

```
def player(answer):
```

```
    finans = number(ranlist)
```

```
    starter = True
```

```
    counter = 1
```

```
    while starter:
```

```
        while counter < 15:
```

```
            result = 0
```

```
            for i in range(0,len(finans)):
```

```
                for j in range(0,len(finans)):
```

```
                    if answer[i] == finans[j]:
```

```
                        if i == j:
```

```
                            result += 10
```

```
                        else:
```

```
                            result += 1
```

```
            return result
```

```
        counter += 1
```

```
    if result == 40:
```

```
        starter = False
```

```
    if starter == False:
```

```
        break
```

### 3.電腦猜人(電腦攻擊):

```
playerMaybe = '1256'
```

```
/**四字亂數初始值*/
```

```
abPlayer = "02"
```

```
/**回答初始值*/
```

```
createDB(abPlayer,playerMaybe)
```

```
/**初始化DataBase*/
```

```
if abPlayer==40:
```

```
    print("Win")
```

```
else:
```

```
    while(True):
```

```
        playerMaybe = DBget()
```

```
        abPlayer = QuestionPlayer(playerMaybe)
```

```
        playerAnsList.append(answerList(playerMaybe, abPlayer))
```

```
/**加入每次回答的值*/
```

```
        if(abPlayer=="40"):
```

```
            print("win")
```

```
            break
```

```
        else:
```

```
            if(deleteDB(abPlayer, playerMaybe, cpuTry)==False):
```

```
/**先認輸,然後請玩家輸入答案*/
```

```
            k=""
```

```
            while(True):
```

```
                k = input("error,please enter your answer: ").strip()
```

```
                if(len(k)!=0 and k.isdigit()):
```

```
                    break
```

```
            fresult = checkList(k)
```

```
            if(fresult==None):
```

```
                print("I am Loser.")
```

```
print('MyDataBase"s Data')
    for kk in playerMaybe:
        print(kk)
        /**電腦從上面Anslist中尋找玩家每次回復的值是否有誤*/
    else:
        print("Your answer is false")
        break
```

總結以上，我們的程式具有以下功能：

1. 利用程式將運算結果、以文字方式傳到Google，再轉成語音傳至喇叭出來。
2. 遊戲AI在攻擊方面，使用資料庫篩選方法，將玩家提供的資訊，去除資料庫中不可能的答案後，再從可能的答案隨機選出數字作為猜測答案。
3. 遊戲AI在防守方面，電腦會使用資料庫，將玩家猜測的答案，提供出一組AB答案，該組AB答案，會使答案資料庫刪除不可能的答案，並留下可能的答案。然而該組AB答案的可能性為最多。

#### 4-1研究心得:

不論是從前,亦或是現在,學生一直都熱愛著電玩遊戲。

基於上述原因,在選擇專題研究報告的主題時,我們毫不考慮地將遊戲製作列入選項。

對高職學生來說,在學校之學術課程裡並沒有直接接觸到遊戲製作之相關課程,雖然為此感到遺憾,但也為能夠在專題研究課程中製作出一款以自身所學所愛的遊戲而感到高興。

因為研究這個專題,讓我們了解到自己的見識淺薄,還有很多要學習和不足的地方,這次的專題讓我們從中學習到不少東西。我們從甚麼都不懂的狀態下,經過漫長的訓練,到最後完成這個專題。

#### 4-2 未來展望:

本遊戲的設計理念還尚需加強,機器人互動方式目前也較為單調,希望未來能利用所學習到的專業,加強機器人互動聲調且加強美術部分。

另外程式部分也須在更加深入研究,雖然AB Game能用推理的方式找出答案,但玩家可能會覺得太簡單或困難,所以我們可以再把遊戲的程式稍作改進,透過互動的方式來調整難易度等等…

期望未來能夠以此專題為基,並研發出更多小遊戲,做出更生動之機器人。