

1A2B遊戲實作

指導老師：張傑帆

組員：法學二，劉冠甫；財法二，許殷齊



動機與目的

- 動機
- 目的

動機

- 暑假
 - 吃飯
 - 雛形
- 計算機概論課程
 - 視窗程式運用
 - 期末報告機會

```
        m != i):
            range(0, 10):
                != m) and (n != j) and (n != i):
                    num = (i*1000 + j*100 + m*10 + n)
                    num = str(num)
                    universalSet.append(num)

i in range(1,10):
for j in range(1, 10):
    if j != i:
        for m in range(1, 10):
            if (m != j) and (m != i):
                num = (i*100 + j*10 + m)
                num = str(num).zfill(4)
                universalSet.append(num)

# 執行猜測
global cnt
cnt = 0
while True:
    if cnt == 0:
        guess = "1234"
    elif cnt == 1:
        guess = "5678"
```

目的

- 演算法
 - 追求最低次數
- 視窗程式應用
 - 貼近使用者之遊戲介面
 - 介面優化

```
me, text=f'您欲給電腦猜測的數字為'

# 進度條 running motion
for i in range(5):
    pb['value'] += 20
    root.update_idletasks()

root.after(1000, result_config_method)
root.after(1000, lambda: root.destroy())

def progress_bar_widget():

    global pb
    global root
    global code

    # 毀滅機制
    code = input_1_entry.get()
    window_4.destroy()
```

```
1 import tkinter as tk
2 import random as re
3 from tkinter import ttk
4 from tkinter.constants import END
5
6
7 > def check_winner_method(): ...
16
17 > def guess_algorithm(guess, var): ...
31
32 > def ai_algorithm(code): ...
86
87
88
89
90 > def result_config_method(): ...
102
103
104 > def progress_method(): ...
113
114
115 > def progress_bar_widget(): ...
140
141
142 > def input_tkinter_method(): ...
168
169
170
171 > def end_game_method(): ...
176
177
178 > def restart_tkinter_method(): ...
200
201 > def gotcha_method(): ...
220
```

素材與方法

- 素材
- 方法

素材

- 程式
 - 開發軟體Python
- 套件
 - Random
 - 亂數表隨機生成數字
 - Tkinter
 - 建立介面模組
 - Ttk
 - 模擬電腦計算動畫

```
---560x1600
)

tk.Label(window, text='WELCOME TO 1A2B GAME!', font=
).pack()

# 模擬啟動
loading_label = tk.Label(window, text='GAME LOADING...', font=('',
loading_label.pack()
loading_label.after(3000, lambda: loading_label.destroy()) # 時間戳記

# 製作使用者輸入猜測數字欄位--上半部
input_frame = tk.Frame(window)
input_frame.after(3000, lambda: input_frame.pack(side=tk.TOP))
input_label = tk.Label(input_frame, text='請輸入您的猜測: ')
input_label.after(3000, lambda: input_label.pack(side=tk.LEFT))

global input_entry

input_entry = tk.Entry(input_frame)
input_entry.after(3000, lambda: input_entry.pack(side=tk.LEFT))

# 建立按鈕區域--下半部
```

方法

- 分為二部分分述之
 - 演算法
 - 遊戲

```
        label_3.place(x=10, y=60)

    input_1_entry = tk.Entry(window_4)
    input_1_entry.place(x=200, y=60)

    submit_btn = tk.Button(window_4, text='Submit', height=2, width=10)
    submit_btn.place(x=155, y=100)

    window_4.mainloop()

def end_game_method():

    # 整個程式結束
    window.destroy()
    window_3.destroy()
```

方法-演算法

- 1A2B遊戲實為四不同數字的排列組合
 - 組合數有限、電腦計算容易
 - 創造字集儲存所有可能性
- 切割字集
 - 先行猜測1234、5678二組合大範圍切割字集
 - 於所剩可能中依亂數模組任選一數字繼續切割字集
 - 同理直至猜測正確為止

方法-遊戲

- 建立主要畫面，以欄位供使用者輸入參數
- 使用者猜對時，呈現類似彈跳視窗詢問使用者是否挑戰演算法
 - 若不同意，則在作一個類似彈跳視窗詢問使用者是否重新開始
 - 若使用者同意挑戰演算法，則遊戲進行如下頁所述

方法-遊戲

- 若使用者同意挑戰電腦，則呈現類似彈跳視窗之欄位供使用者輸入題目
- 跑出進度條營造較自然之遊戲感
- 將雙方猜測之次數顯示於遊戲之主畫面中，並判斷由演算法或使用者獲勝

221
222
233
234
235
258
259
260
261
275
276
277
284
285
286
298
299
300
304
305
306
359
360
361

```
> def configure_hint_method(Str): ...
```

```
> def game_mainloop():|...
```

```
> def check_machine(check, code): ...
```

```
> def answer_random_maker(): ...
```

```
> def refresh_method(): ...
```

```
> def config_answer_method(): ...
```

```
> def tkinter_mainloop(): ...
```

```
tkinter_mainloop()
```

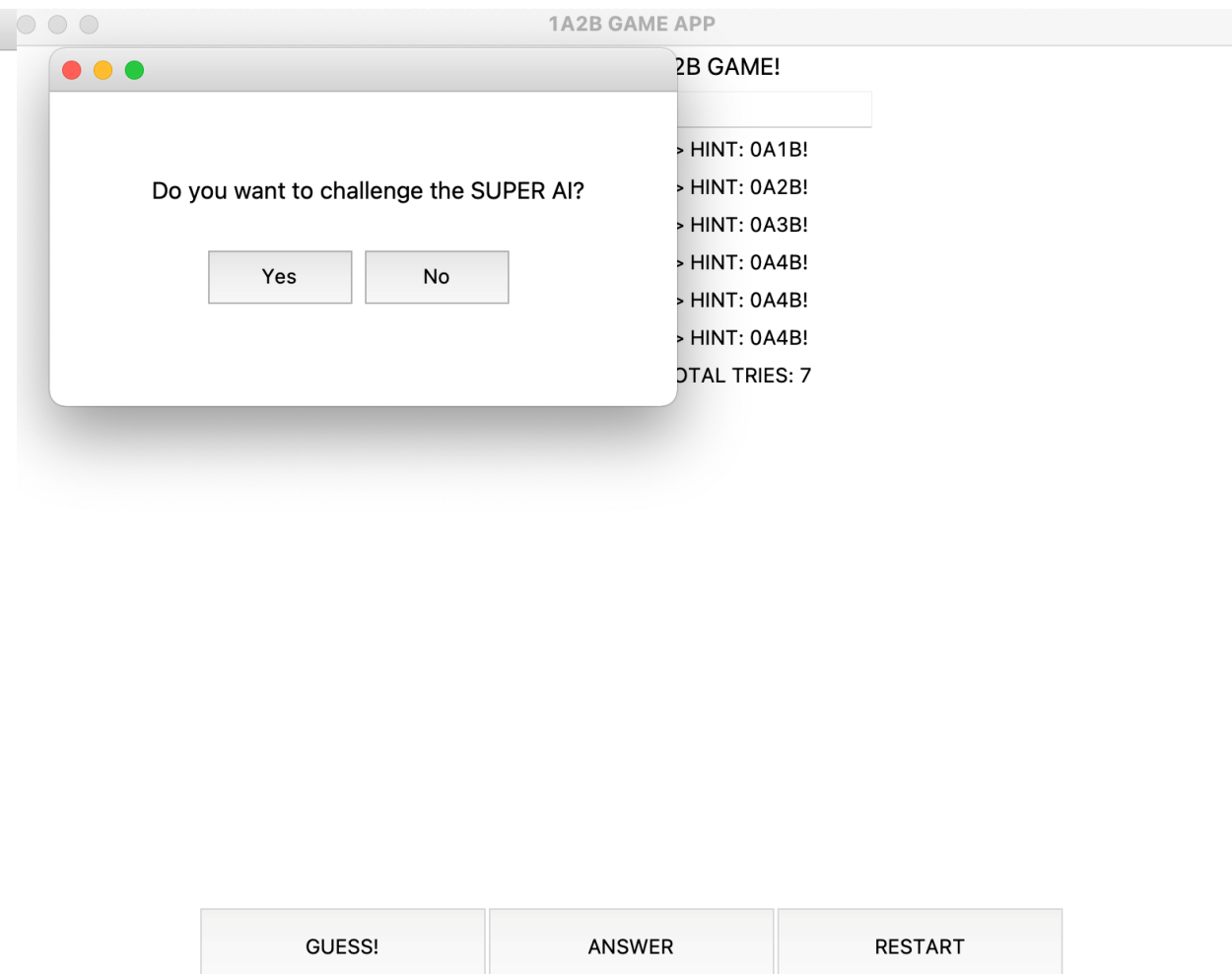
結果

- 遊戲畫面展現

遊戲畫面展現

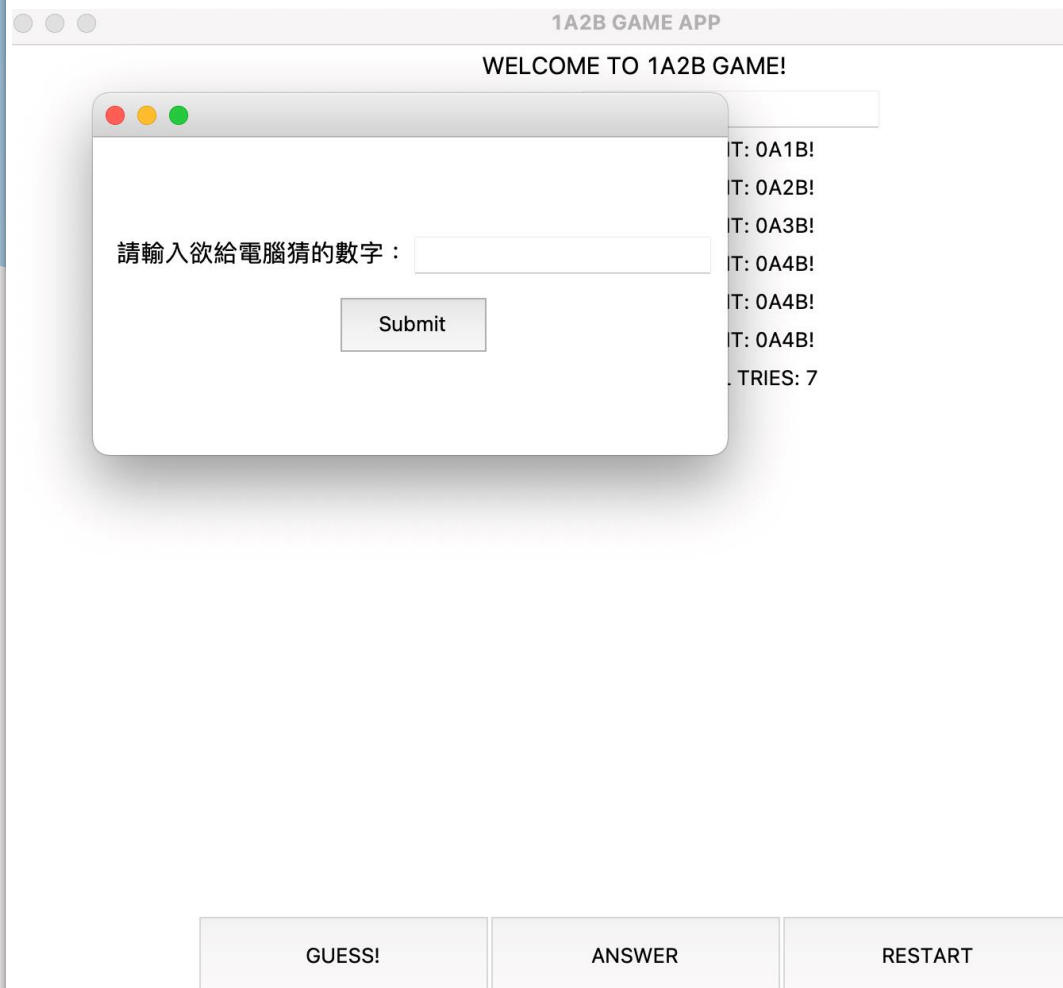


↑ 遊戲主要畫面

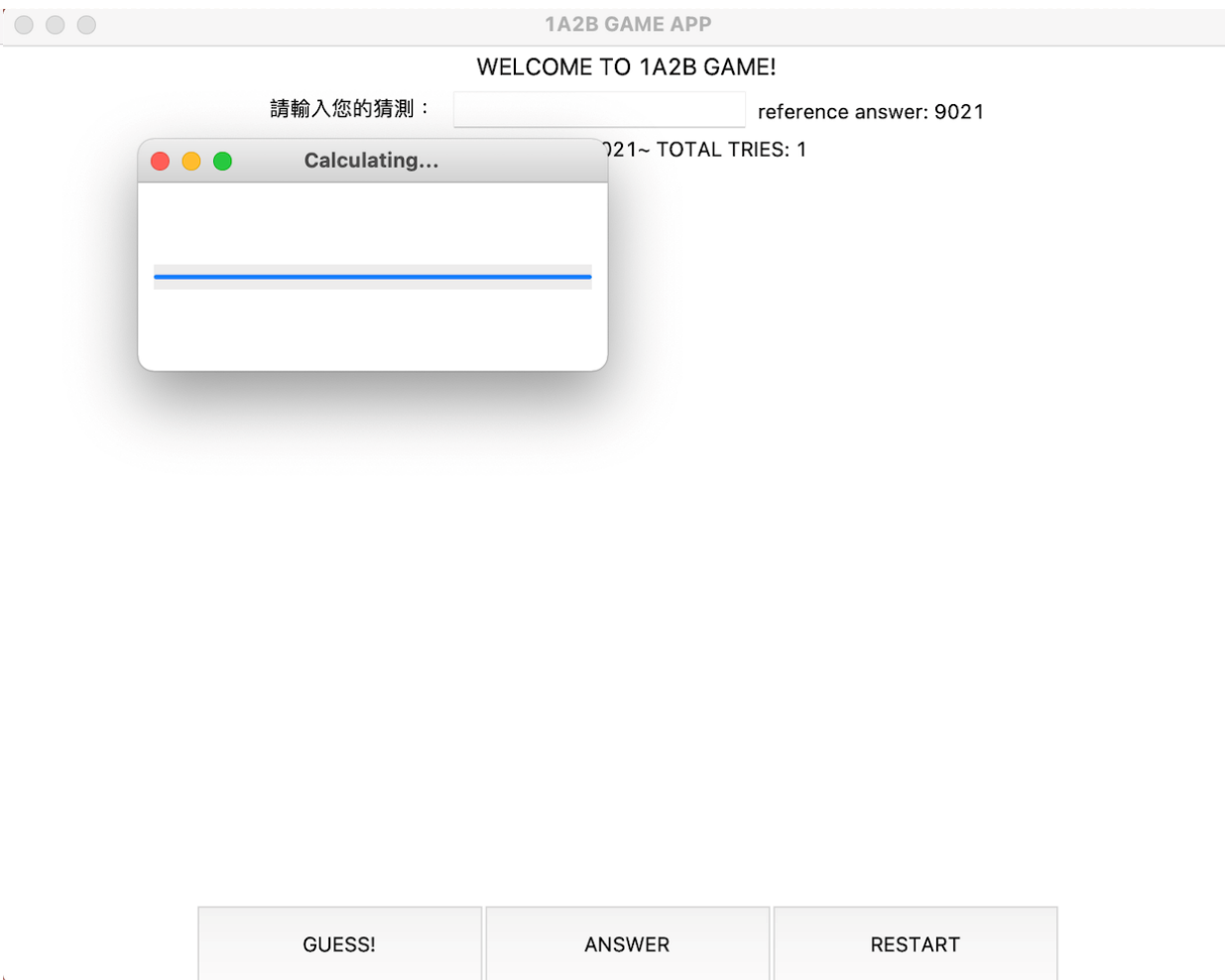


↑ 彈跳視窗-是否挑戰演算法？

遊戲畫面展現



↑ 彈跳視窗-使用者輸入題目



↑ 彈跳視窗-進度條

遊戲畫面展現



WELCOME TO 1A2B GAME!

請輸入您的猜測： 您欲給電腦猜測的數字為： 5467

YOUR GUESS: 1234 --> HINT: 0A1B!

YOUR GUESS: 5678 --> HINT: 0A2B!

YOUR GUESS: 1278 --> HINT: 0A3B!

YOUR GUESS: 0178 --> HINT: 0A4B!

YOUR GUESS: 8710 --> HINT: 0A4B!

YOUR GUESS: 1807 --> HINT: 0A4B!

GOTCHA! IT'S 7081~ TOTAL TRIES: 7

AI GUESS: 1234 --> HINT: 0A1B!

AI GUESS: 5678 --> HINT: 1A2B!

AI GUESS: 6953 --> HINT: 0A2B!

AI GUESS: 9681 --> HINT: 0A1B!

AI GUESS: 7490 --> HINT: 1A1B!

AI GUESS: 2560 --> HINT: 1A1B!

AI GUESS: 7528 --> HINT: 0A2B!

AI GUESS: 5840 --> HINT: 1A1B!

GOTCHA! IT'S 5467~ TOTAL COUNTS: 9

You Win!

GUESS!

ANSWER

RESTART

← 遊戲最終畫面



結論與討論

- 結論
- 討論與展望

結論

- 成功製作
- 演算法
 - 若基數夠大，演算法猜測平均應為七次
 - 根據測試，演算法猜測幾乎皆於九次內
 - 無法確切統計，上述僅為約略之估計值

```
---560x1600
)

tk.Label(window, text='WELCOME TO 1A2B GAME!', font=
).pack()

# 模擬啟動
loading_label = tk.Label(window, text='GAME LOADING...', font=('',
loading_label.pack()
loading_label.after(3000, lambda: loading_label.destroy()) # 時間戳記

# 製作使用者輸入猜測數字欄位--上半部
input_frame = tk.Frame(window)
input_frame.after(3000, lambda: input_frame.pack(side=tk.TOP))
input_label = tk.Label(input_frame, text='請輸入您的猜測: ')
input_label.after(3000, lambda: input_label.pack(side=tk.LEFT))

global input_entry

input_entry = tk.Entry(input_frame)
input_entry.after(3000, lambda: input_entry.pack(side=tk.LEFT))

# 建立按鈕區域--下半部
```


討論與展望-演算法

- 是否更能精進演算法使其更加有邏輯
 - 人類思考邏輯的加入
- 其他演算法
 - 可能涉及到機率分配與其他數學相關問題
 - 並非數學專業，僅能依最傳統之方式土法煉鋼

討論與展望-遊戲

- 介面優化
 - Hint部分若超過介面大小則無法顯示
 - 是否可以清除或以其他方式繼續保存
- 遊戲
 - 遊戲中無特別設定除錯系統，無法阻擋使用者的錯誤鍵入
 - 程式碼可寫成一種class，在交換使用或供他人參考時可以更加方便

主要參考資料

- Tkinter

- <https://blog.techbridge.cc/2019/09/21/how-to-use-python-tkinter-to-make-gui-app-tutorial/>
- <https://www.delftstack.com/zh-tw/tutorial/tkinter-tutorial/tkinter-geometry-managers/>

- ProgressBar Widget

- <https://www.geeksforgeeks.org/progressbar-widget-in-tkinter-python/>
- <https://www.pythontutorial.net/tkinter/tkinter-progressbar/>



感謝聆聽

煩請諸位先進不吝賜教