

สรุปองค์ความรู้ที่ได้

1. หลักการทำงานของ GUI ของ tkinter แบบคร่าวๆ (How render loop work)
2. การออกแบบโปรแกรมที่เป็น Event-Driven Programming (Non-blocking code)
3. ได้พัฒนา Program ให้ใช้งานได้หลากหลาย Platform (MacOS, Windows)
4. ได้ศึกษา mode ต่างๆ ในการเปิดไฟล์ (r, r+, w, w+, rw, a, a+)
5. การทำงานกันเป็นทีมผ่านการใช้ git และ github
 - 5.1 โดยการสร้าง branch ของเพื่อน และของเราโดยไม่ให้คนที่ม่ประสบการณ์น้อยใช้ branch
6. Method การ เขียนไฟล์ csv โดยใช้ read
7. ในช่อง command ของ widget entry หากต้องการใช้ฟังก์ชันที่ต้องใช้ argument ติดไปด้วยให้เราใช้ lambda : ตามด้วย call back function
8. โดยปกติ grid ของ tkinter จะเป็น relative หากเราใส่ grid(row = 20) แต่ก่อนหน้านั้นไม่ม row 1 2 3 ... 19 มันจะไปอยู่ที่ row 1 แต่การจะจองที่ให้ใช้
rowconfigure,columnconfigure(tuple(range(ช่องที่อยากจอง)))
9. การทำ Flowchart หากว่าเป็นการเริ่มต้นใหม่ของแต่ละฟังก์ชัน เราจะเริ่มด้วยชื่อฟังก์ชัน
10. การบริหารเวลา ควบคุมความท้อแท้ และการสื่อสารกับเพื่อนร่วมทีม
11. การบังคับไม่ให้ resize หน้าจอได้ด้วย window.resizable(false, false)
12. การให้ขึ้นหน้าต่างแจ้งเตือนด้วย show info และการดักจับต่าง ๆ ให้ครอบคลุม
13. ได้รู้แนวทางในการใช้ dictionary และ list มากขึ้น เช่น ใช้ dictionary สำหรับเก็บแต่และตัวอักษร
ใน คำศัพท์ที่เรากรอก จากนั้นนำไปลบในฟังก์ชัน checkWord
14. สไตล์การเขียน code ให้ clean และความรอบคอบของเพื่อนผู้ที่มีประสบการณ์มาก่อน

Project Com Pro 1

หัวข้อ “Wordle”

จัดทำโดย

รหัส 6504062620159 นายโมกซ์ มาอาจ ลำดับที่ 16

รหัส 6504062620175 นายอชิชา เล็กสรรเสริญ ลำดับที่ 17

รายวิชา 040613201 COMPUTER PROGRAMMING I

ปีการศึกษา 2565

บทที่ 1

1. วัตถุประสงค์ของโปรแกรม

- 1.1 ใช้เป็นเกมแก้ปริศนาคำว่างได้
- 1.2 เป็นเกมบริหารสมอง เนื่องด้วยต้องใช้ทักษะ วิเคราะห์ และ recall
- 1.3 ฝึกคำศัพท์ภาษาอังกฤษ
- 1.4 เป็น project สำหรับการนำความรู้ด้าน coding มาประยุกต์

2. หลักการทำงานของโปรแกรม

- 2.1 โปรแกรมจะถูกทำงาน เมื่อเรารัน execute
- 2.2 ทำการสร้างหน้า GUI
 - 2.2.1 หน้าแสดงผล
 - 2.2.2 กล่องตอบ
 - 2.2.3 กราฟิกคีย์บอร์ด ที่สามารถคลิก แล้วแสดงผลแบบคีย์บอร์ด
 - 2.2.4 ปุ่มไปหน้า stats window
- 2.3 สุ่มคำจากไฟล์ คำศัพท์
- 2.4 เมื่อหน้าแสดงผลขึ้นมาแล้วจึงกรอก คำศัพท์ที่มีความหมาย และมี 5 ตัวอักษรลงไป
 - 2.4.1 หากคำศัพท์ไม่มีความหมายให้สั่งให้ กรอกใหม่ แต่คงคำศัพท์ไว้
- 2.5 ทำการเช็คตัวอักษรในคำศัพท์ ตรงในเฉลยไหม
 - 2.5.1 หากตัวอักษรตรงกันแบบ ตำแหน่งเดียวกันให้ หน้าแสดงผล เป็นสีเขียว
 - 2.5.2 หากตัวอักษรอยู่ในเฉลย แต่ไม่ได้อยู่ตรงตำแหน่ง หน้าแสดงผล เป็นสีเหลือง
 - 2.5.3 ทำการเปลี่ยนสีคีย์บอร์ด ตามอักษรของคำตอบ
- 2.5 ทำแบบนี้ 6 รอบ
- 2.6 อัปเดต history ไปสู่ไฟล์ csv
- 2.7 หากกดหน้า stat window ให้ใช้สถิติการเล่น

3. การเล่น

เราจะสุ่มคำมา 5 ตัวอักษรซ่อนไว้ จากนั้นให้ผู้เล่นกรอกคำศัพท์ที่มีความหมาย และมี 5 ตัวอักษร ให้ถูกภายใน 6 ครั้ง มาโดยโปรแกรมจะตรวจสอบว่า ในคำนั้นมีตัวอักษรที่ตรงกับคำที่ซ่อนไว้ไหม และตรงกับตำแหน่งนั้น ๆ เลยหรือเปล่า เป็น 3 สีได้แก่



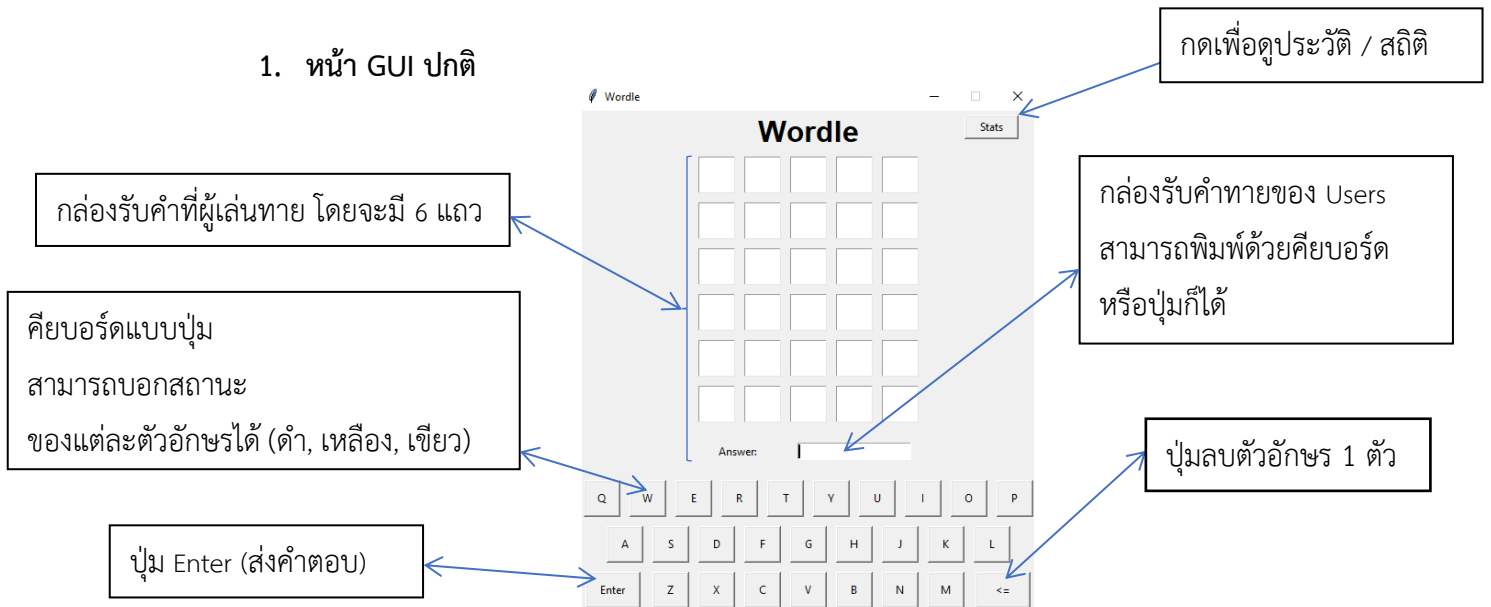
รูป 1.1 เปรียบเทียบสัญลักษณ์สีในเกม “wordle”

- สีเขียว : อักษรที่ทายลงไปนั้น ถูกทั้งตัวและตำแหน่ง
- สีเหลือง : อักษรที่ทายลงไปนั้น ถูกตัวแล้ว แต่ไม่ถูกตำแหน่ง
- สีดำ : อักษรที่ทายลงไปนั้น ไม่ถูกทั้งตัวอักษร และตำแหน่ง

บทที่ 2

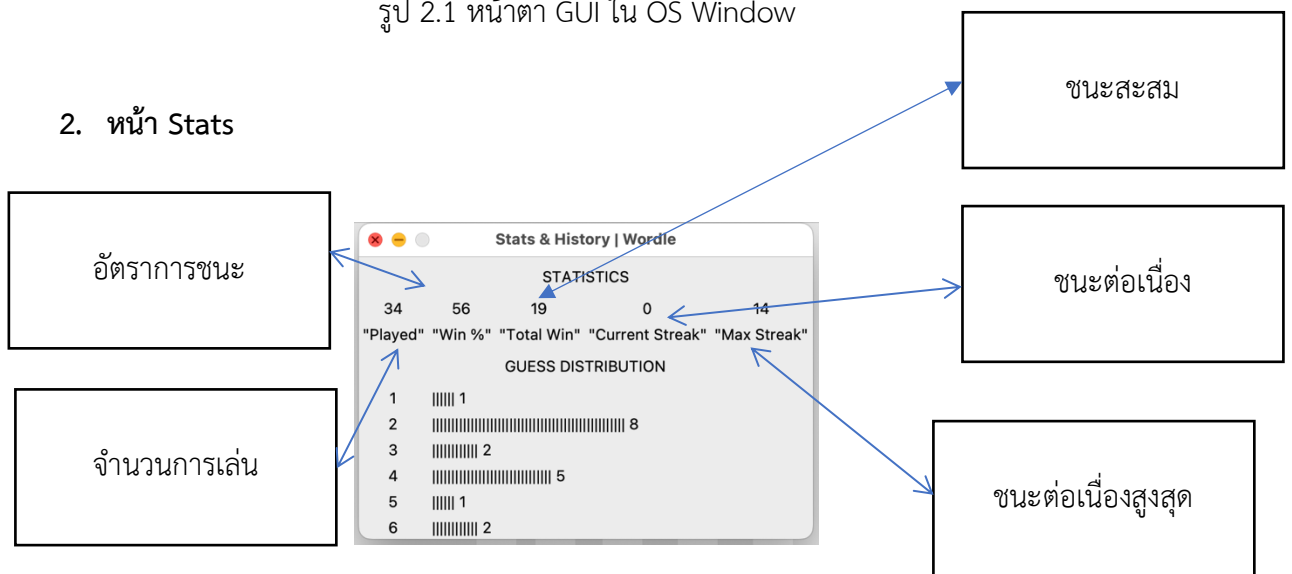
User interface

1. หน้า GUI ปกติ



รูป 2.1 หน้าตา GUI ใน OS Window

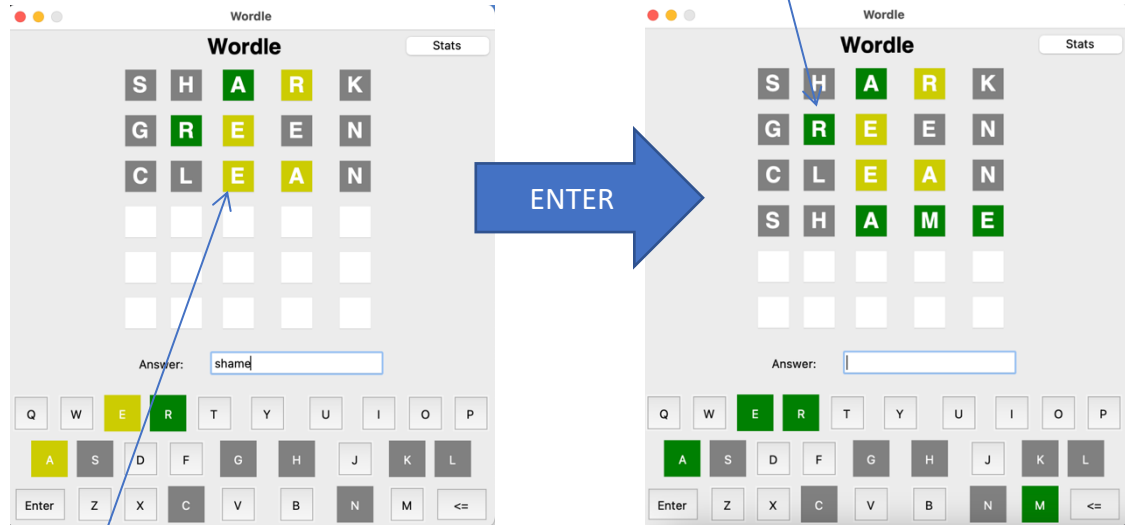
2. หน้า Stats



รูป 2.2 หน้าตาของหน้าต่าง Stats & History | Wordle

3. เมื่อกรอกคำศัพท์ไปแล้ว

หากเรากด ตัวอักษรของคำตอบเรา ตรงกับ
เฉลยที่ตำแหน่งนั้น ๆ จะเป็นคำใบ้สีเขียว

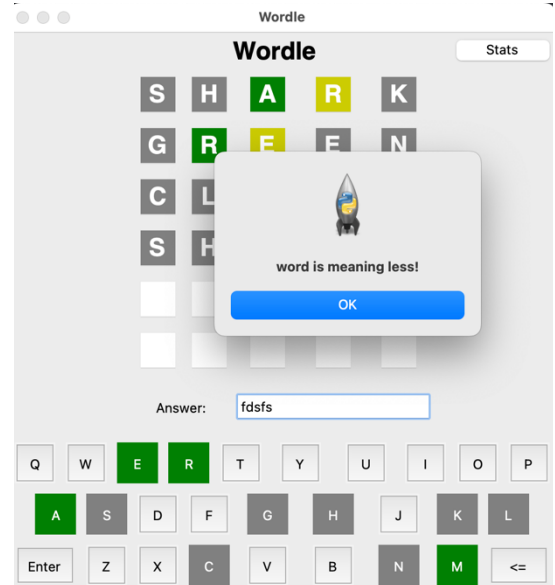
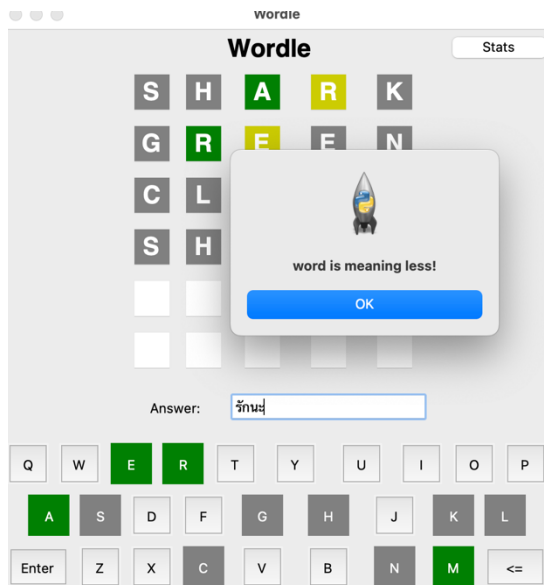


รูป 2.3 รูปเปรียบเทียบการเปลี่ยนสีของ ตาราง และคีย์บอร์ด

หากในคำตอบ ตัวอักษรของคำตอบเรา อยู่ในเฉลย
แต่ไม่ตรงตำแหน่งจะเป็นคำใบ้สีเหลือง

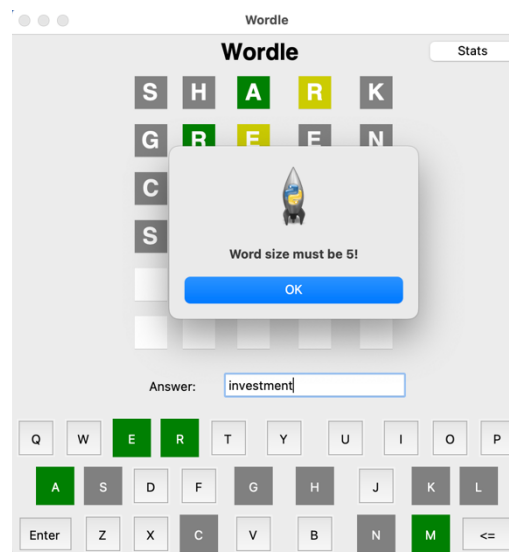
โดยเฉลยคือ “ frame ” ที่ “ AME ” ของ “ shame ” เป็นสีเขียว เพราะตัวอักษรตรงตำแหน่งกันทั้งเฉลย และ คำตอบ

4. เมื่อกรอกคำศัพท์ที่ไร้ความหมายหรือ ไม่ใช่ภาษาอังกฤษ จะขึ้นแจ้งเตือนว่า “word is meaning less!”



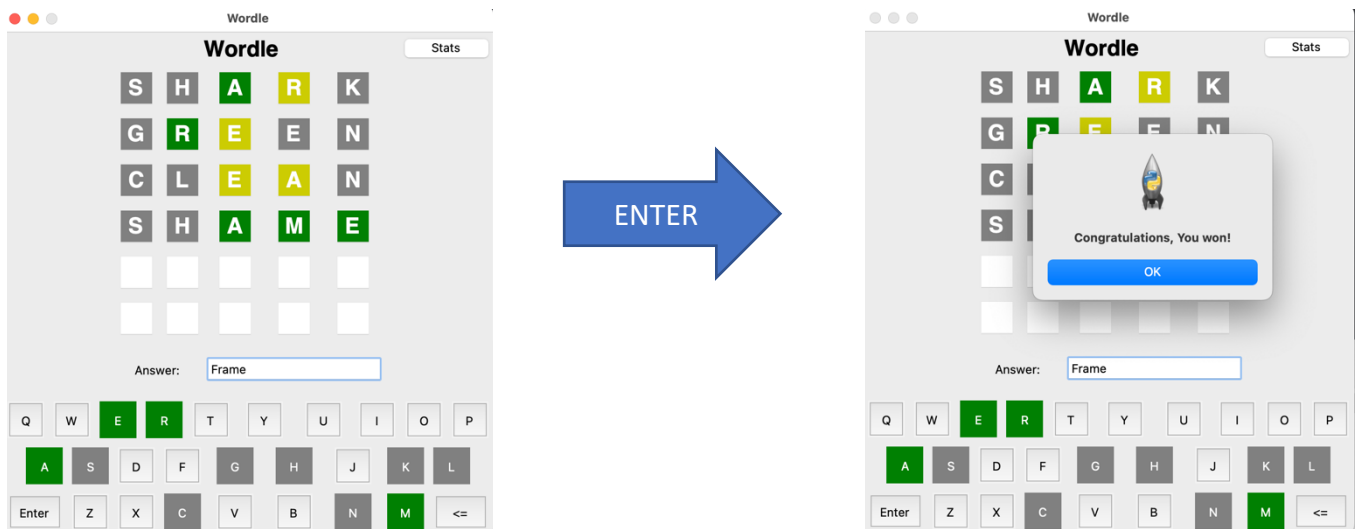
รูป 2.4 การแสดงผล เมื่อกรอกคำศัพท์ที่ไร้ความหมาย และไม่ใช่ภาษาอังกฤษ

5. เมื่อกรอกคำศัพท์ตัวอักษรไม่ถึง 5 และ น้อยกว่า 5 จะขึ้นว่า “word size must be 5!”



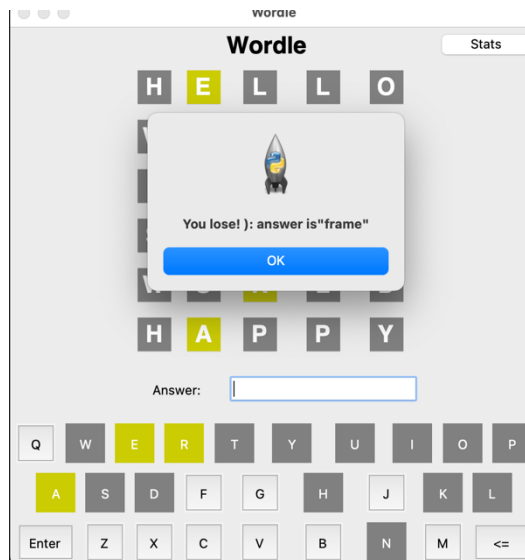
รูป 2.5 การแสดงผล เมื่อกรอกคำศัพท์ที่มีอักขรมากกว่า หรือ น้อยกว่า 5

6. เมื่อชนะ



รูป 2.5 การแสดงผล เมื่อคำศัพท์ที่กรอก ตรงกับเฉลย

7. เมื่อกรอกคำศัพท์ครบ 6 แถวแต่ยังไม่ถูก หรือแพ้



รูป 2.6 การแสดงผล เมื่อแพ้

บทที่ 3

Input and Output

1. Input

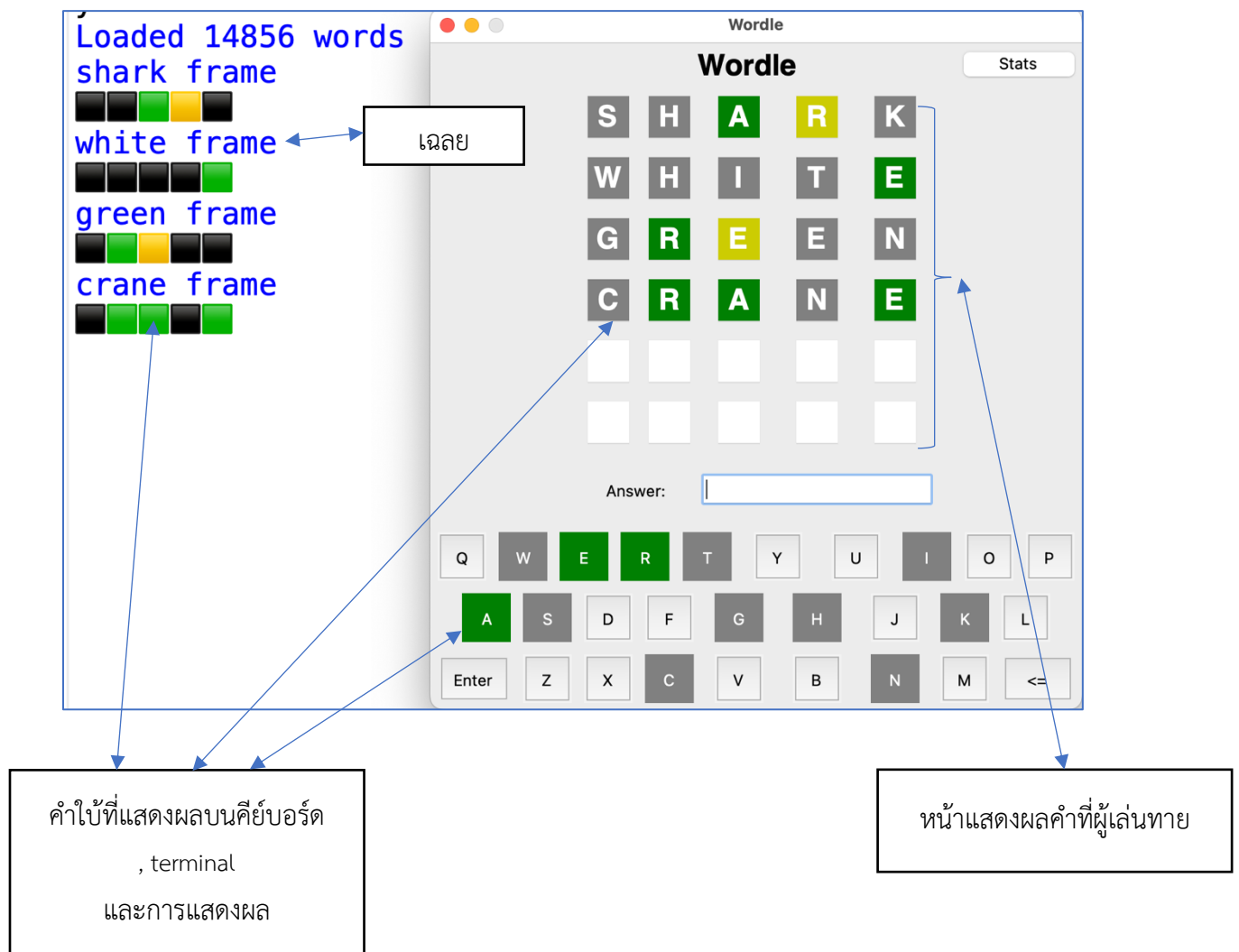
Input จะเป็นคำศัพท์ที่มี 5 ตัวอักษรสามารถส่งได้ 6 ครั้งในแต่ละเกม

2. Output

2.1 หน้าจอแสดงผลคำศัพท์ที่ผู้เล่นทาย

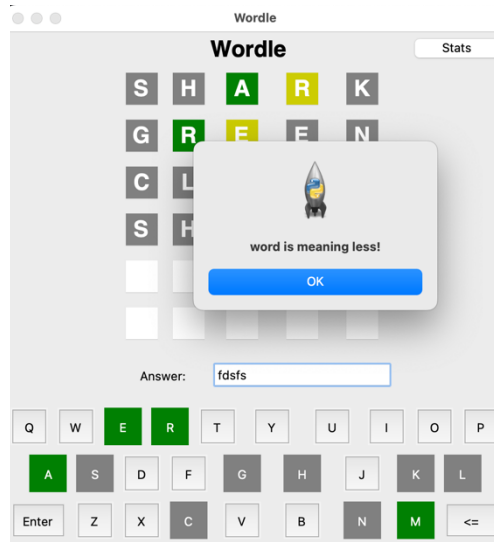
2.2 การแสดงคำใบ้ เหลือง หรือเขียว บนช่องแสดงผล และ คีย์บอร์ด

2.3 เฉลย, คำที่เรากรอก และ คำใบ้ในช่อง Terminal



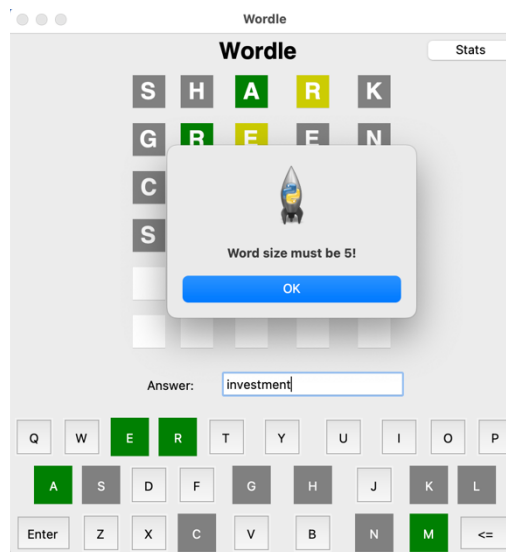
รูป 3.1 การแสดงผล

2.4 หน้าจอเตือน เมื่อกรอกคำศัพท์ที่ไม่มีความหมาย หรือภาษาอื่น



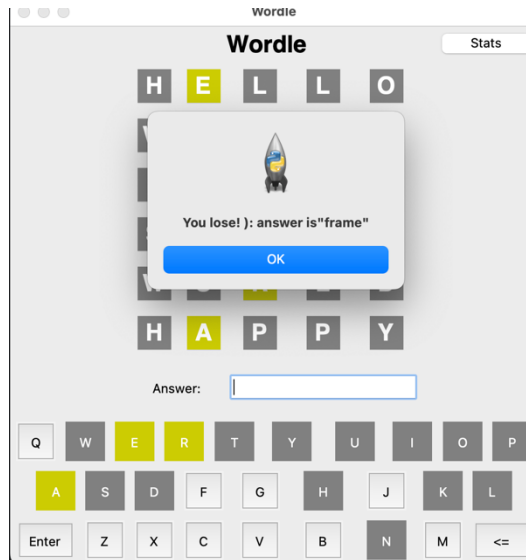
รูป 3.2 กรอกศัพท์ที่ไม่มีความหมาย

2.5 หน้าจอเตือน เมื่อกรอกคำศัพท์อักขรที่มาก หรือน้อยกว่า 5



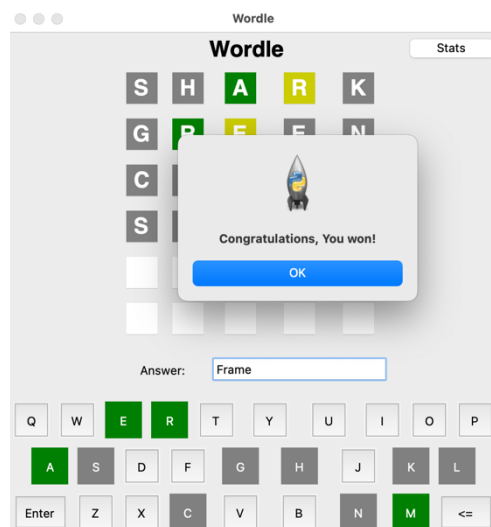
รูป 3.3 กรอกคำศัพท์ตัวอักษรน้อยกว่า หรือมากกว่า 5

2.6 หน้าจอเตือน เมื่อแพ้



รูป 3.4 หน้าจอเตือนว่าแพ้

2.7 หน้าจอเตือน เมื่อชนะ

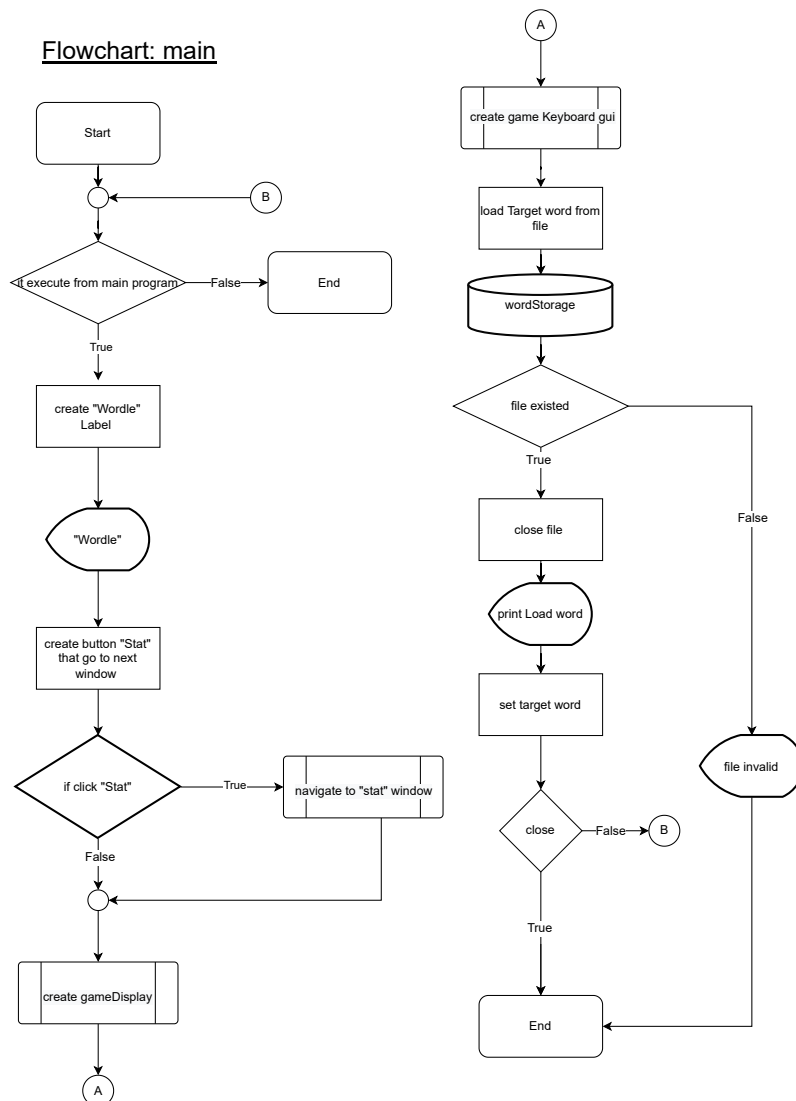


รูป 3.5 หน้าจอเตือนว่าชนะ

บทที่ 4

Flow chart

1. Flowchart : การทำงานรวม

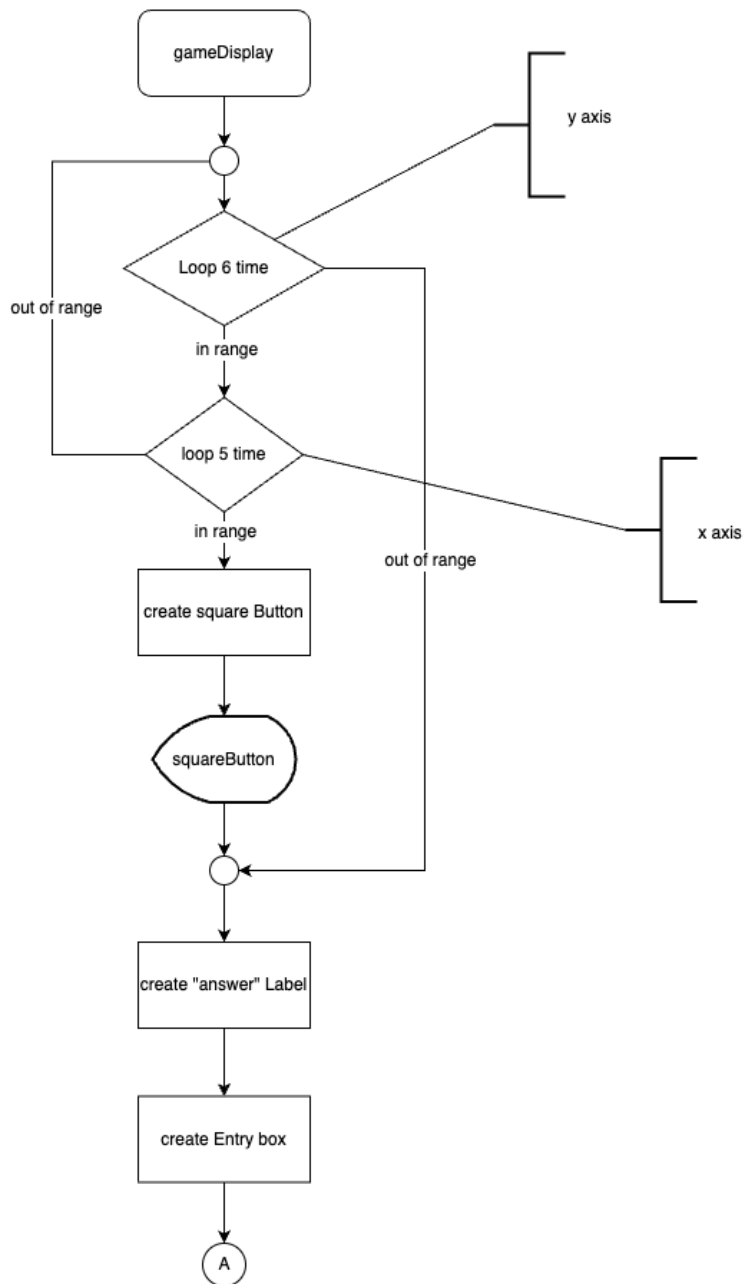


รูปที่ 4.1 Flowchart ของการทำงานรวม

2. Flowchart : การทำงานของส่วนหลัก

2.1 Flowchart : การสร้าง Graphic ส่วนกล่องแสดงผล และกล่องรับคำศัพท์

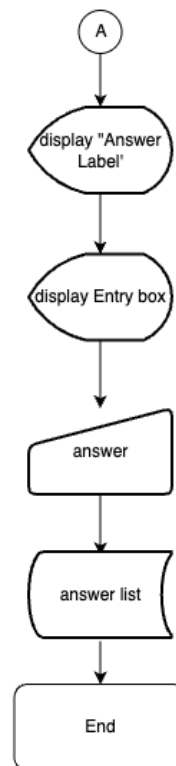
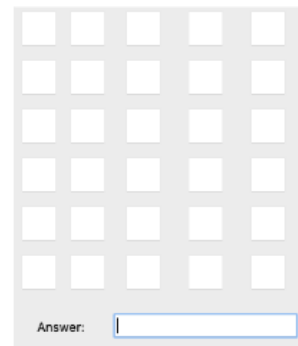
Flowchart: gameDisplay



Description

gameDisplay

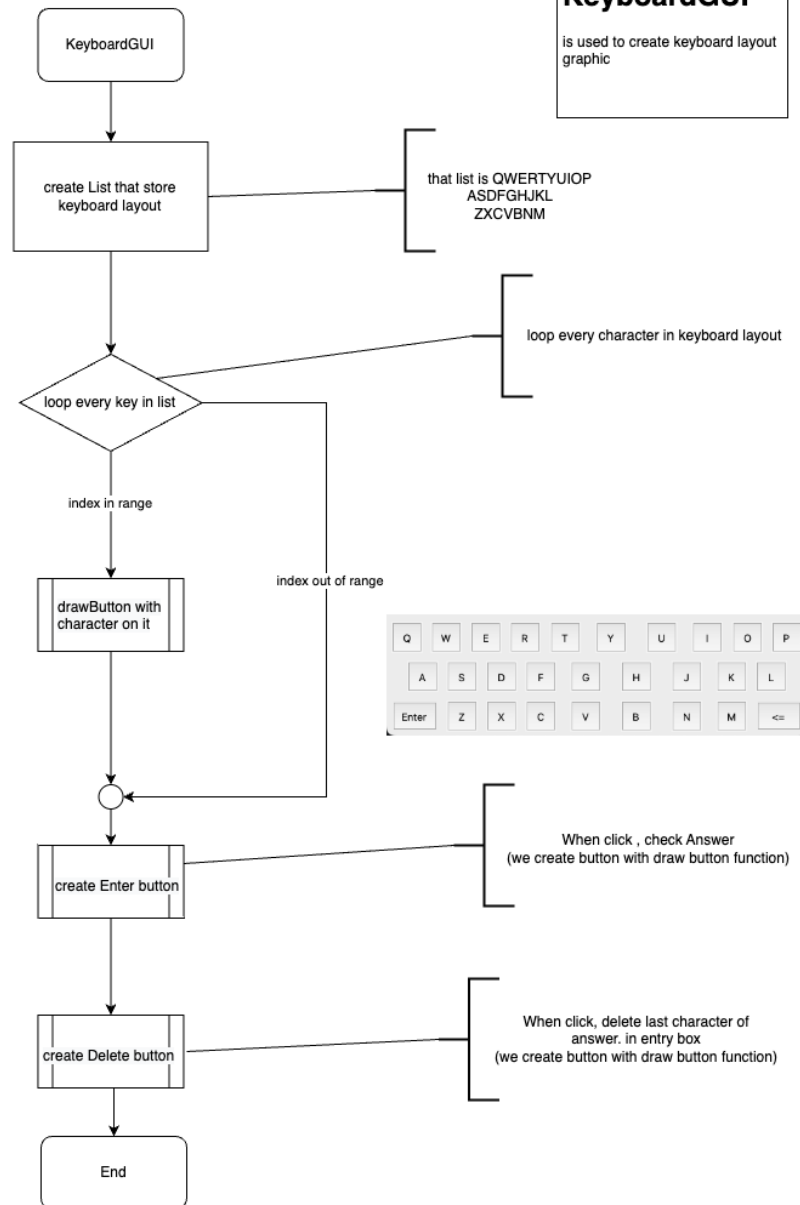
is used to create graphic box for answer to display.



รูป 4.2 Flow chart ของการสร้าง GUI สำหรับแสดงคำใบ้ และกล่องคำตอบ

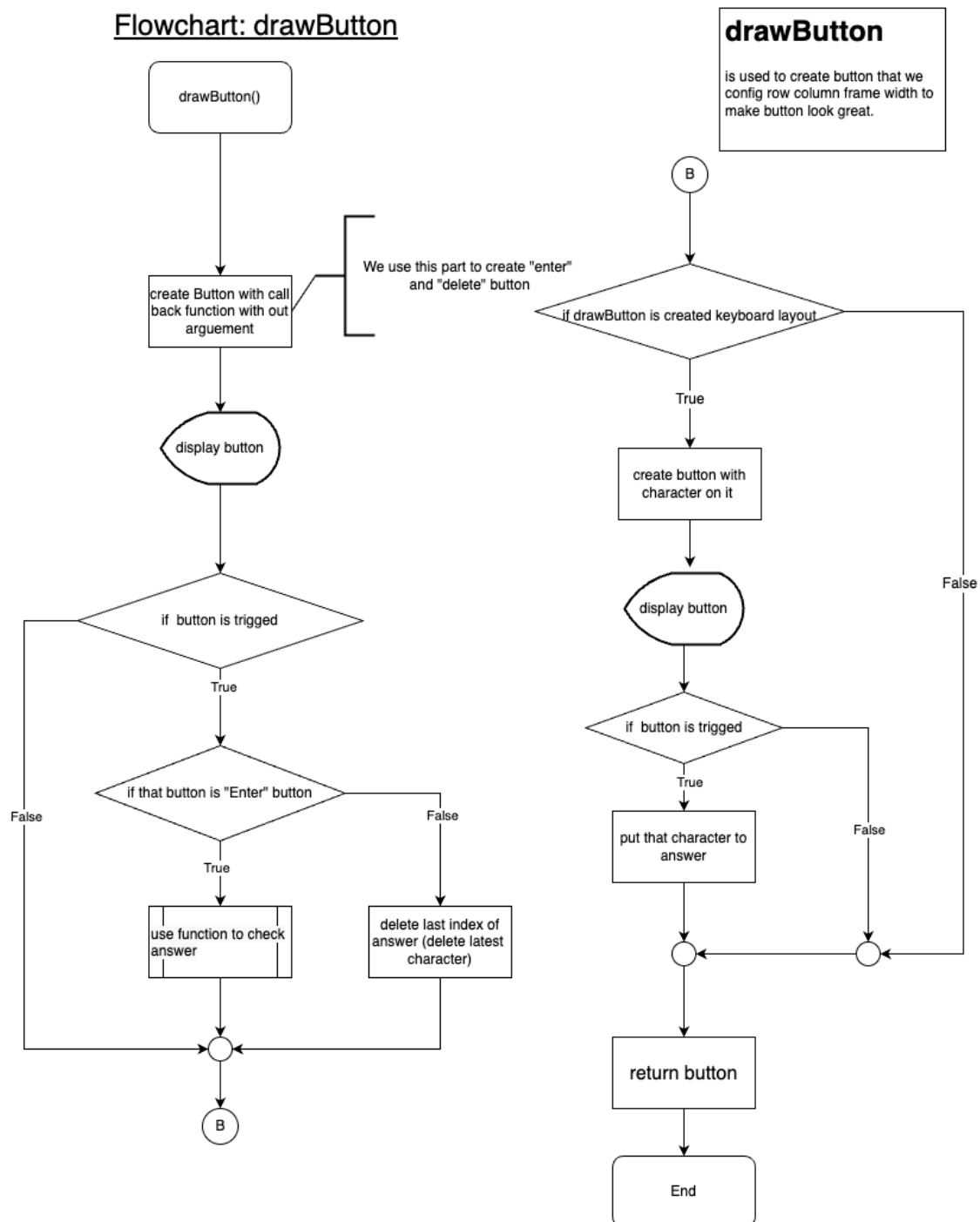
2.2 Flowchart : การสร้าง Graphic ส่วนของคีย์บอร์ด

Flowchart: KeyboardGUI



รูป 4.3 Flowchart ของการสร้าง keyboard graphic

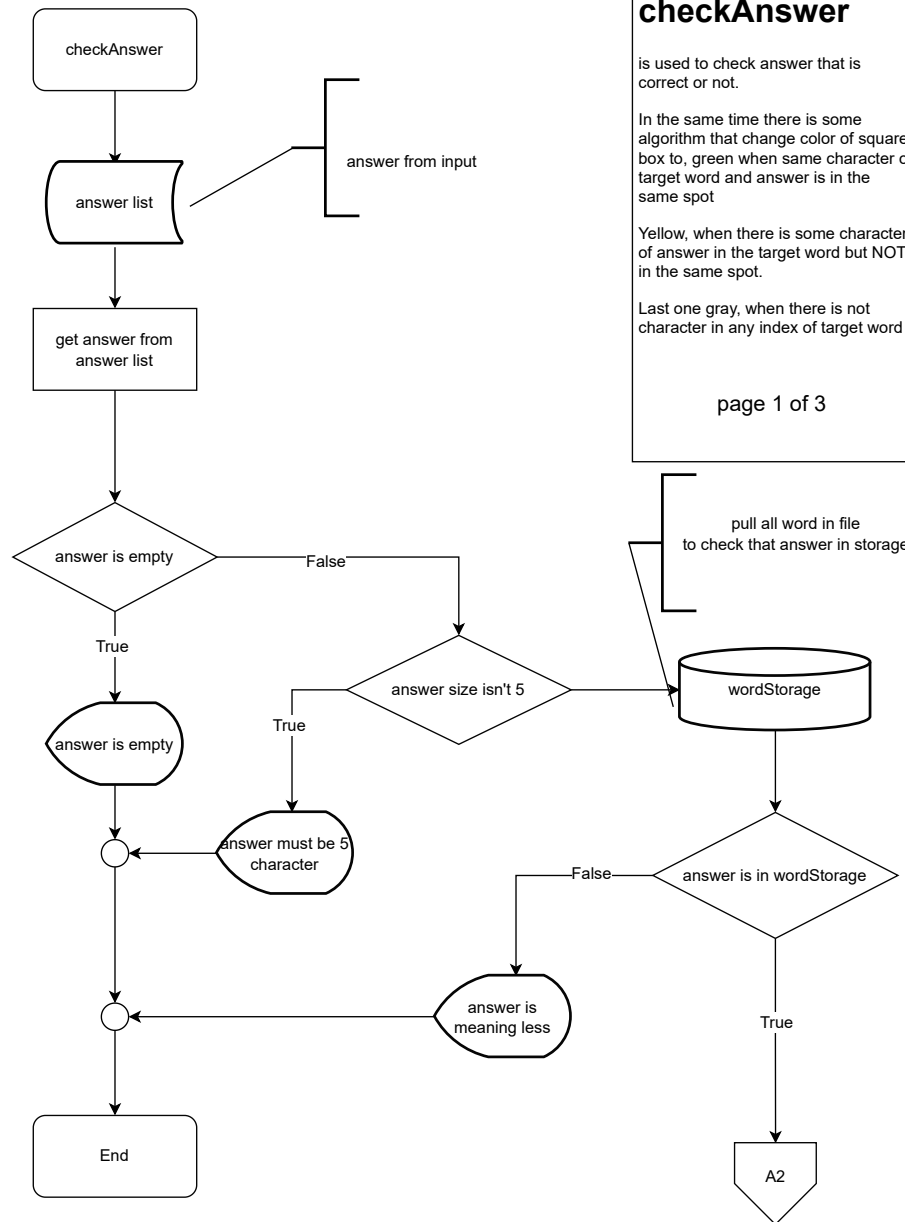
2.3 Flowchart : ฟังก์ชัน drawButton เป็นฟังก์ชันการสร้าง button ที่มีคุณสมบัติที่จะสามารถเปลี่ยนสีคำไปได้ ถูกใช้ใน graphic keyboard เพื่อสร้างปุ่มต่าง ๆ ที่มีตัวอักษรบนนั้น



รูป 4.4 Flowchart drawButton

2.4 Flowchart : ฟังก์ชัน checkAnswer ใช้ในการตรวจสอบคำศัพท์ เปลี่ยนสีคำใบ้ และใช้งานฟังก์ชัน บันทึกข้อมูล ถูกใช้ในฟังก์ชัน drawButton

Flowchart: checkAnswer



Description

checkAnswer

is used to check answer that is correct or not.

In the same time there is some algorithm that change color of square box to, green when same character of target word and answer is in the same spot

Yellow, when there is some character of answer in the target word but NOT in the same spot.

Last one gray, when there is not character in any index of target word

page 1 of 3

รูป 4.5 Flowchart ฟังก์ชัน checkAnswer page 1

Flowchart: checkAnswer

Description

checkAnswer

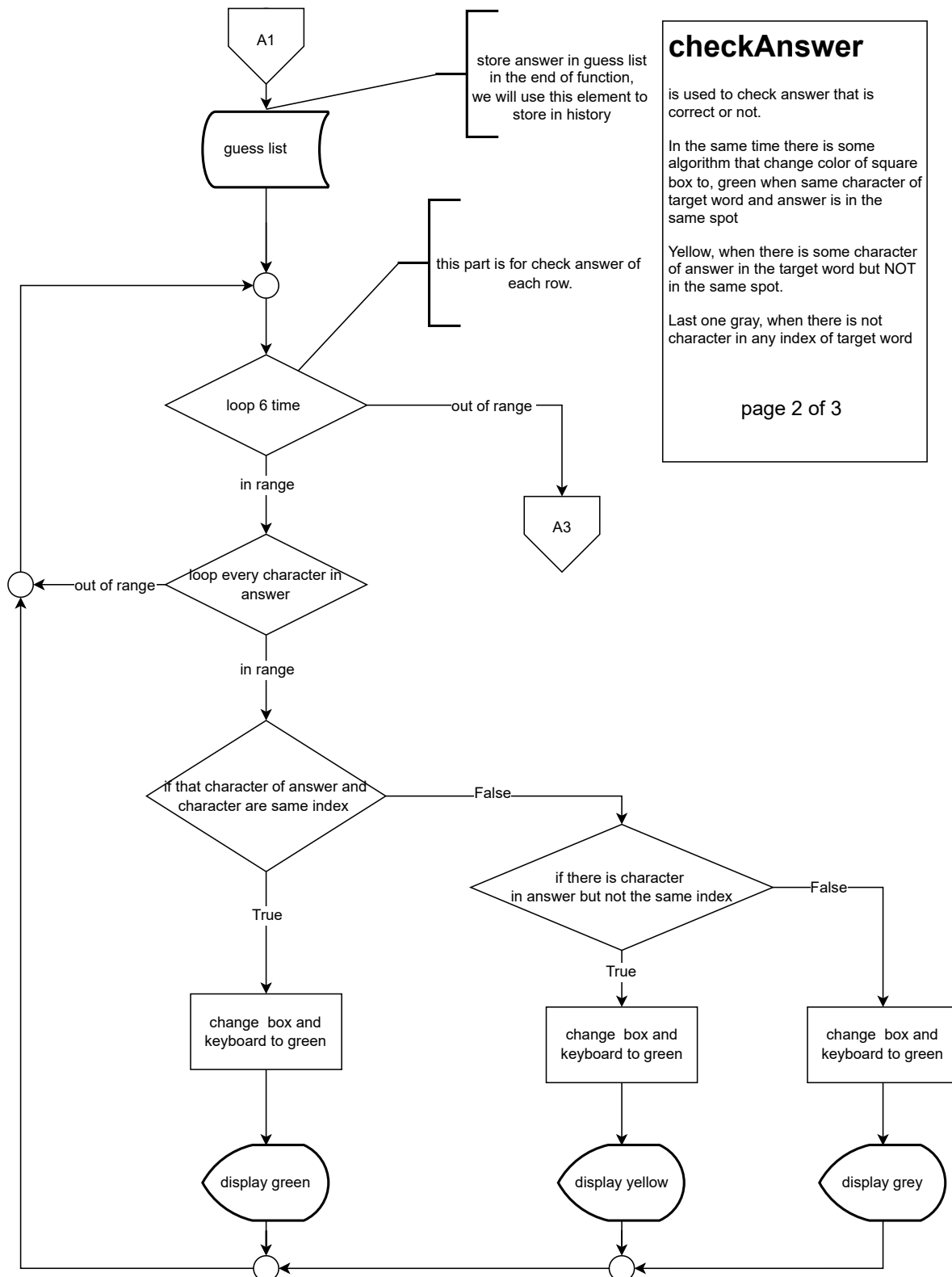
is used to check answer that is correct or not.

In the same time there is some algorithm that change color of square box to, green when same character of target word and answer is in the same spot

Yellow, when there is some character of answer in the target word but NOT in the same spot.

Last one gray, when there is not character in any index of target word

page 2 of 3



รูป 4.6 Flowchart ฟังก์ชัน checkAnswer page 2

Flowchart: checkAnswer

Description

checkAnswer

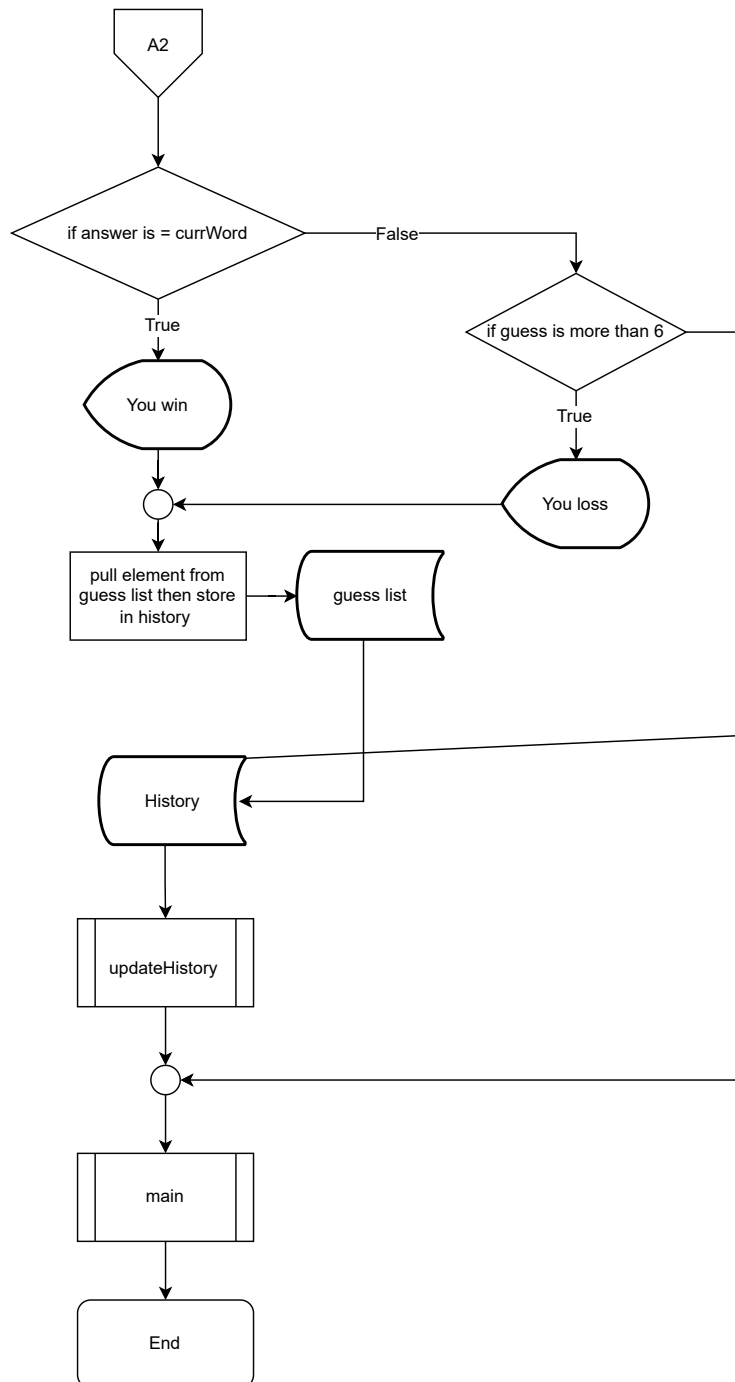
is used to check answer that is correct or not.

In the same time there is some algorithm that change color of square box to, green when same character of target word and answer is in the same spot

Yellow, when there is some character of answer in the target word but NOT in the same spot.

Last one gray, when there is not character in any index of target word

page 3 of 3



store history in data such as,
time targetWord guess word

รูป 4.7 Flowchart ฟังก์ชัน checkAnswer page 3

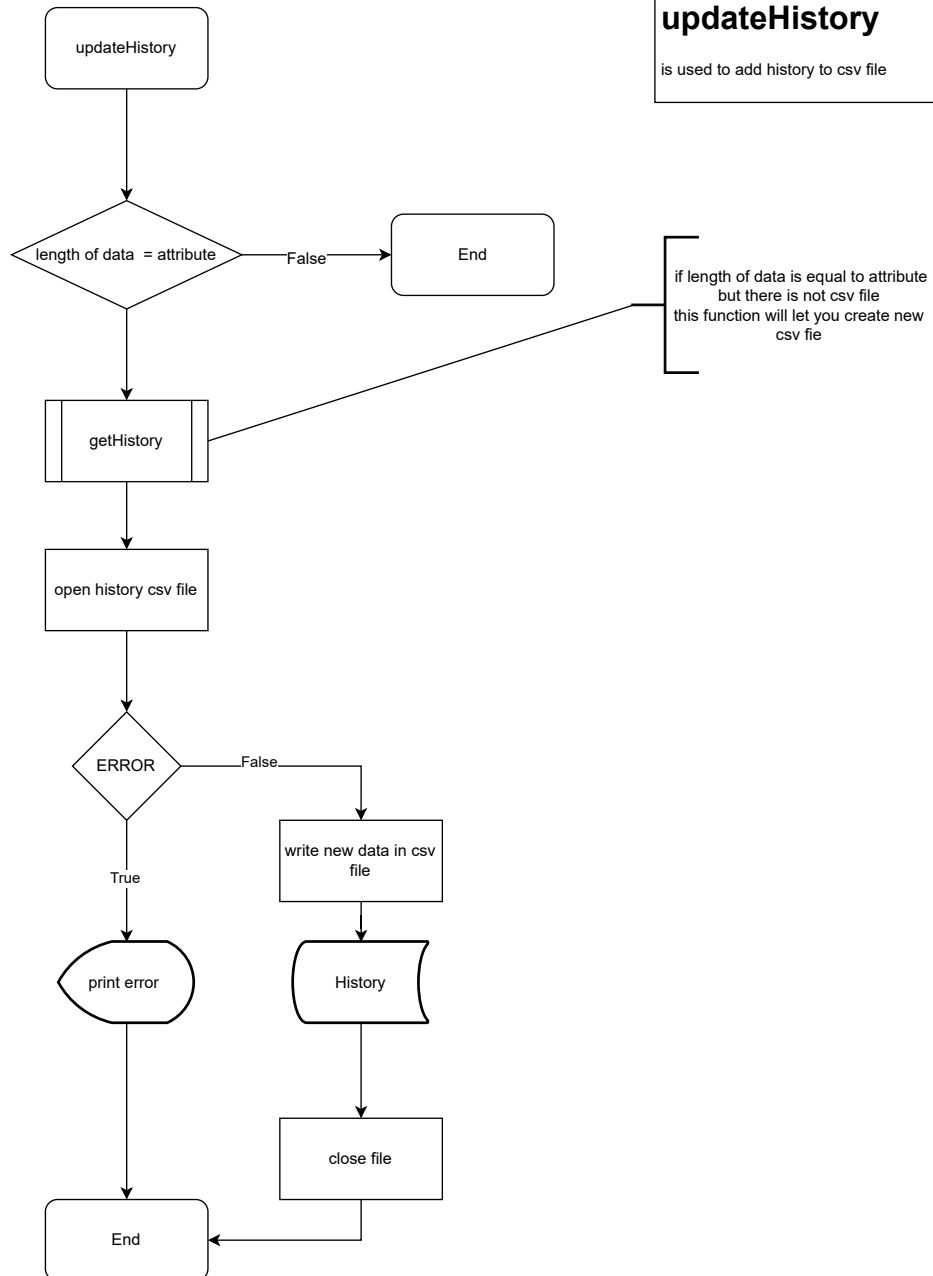
2.5 Flowchart : ฟังก์ชันในการอัปเดตข้อมูลไปยังไฟล์ csv ถูกใช้ใน ฟังก์ชัน check word

Flowchart: updateHistory

Description

updateHistory

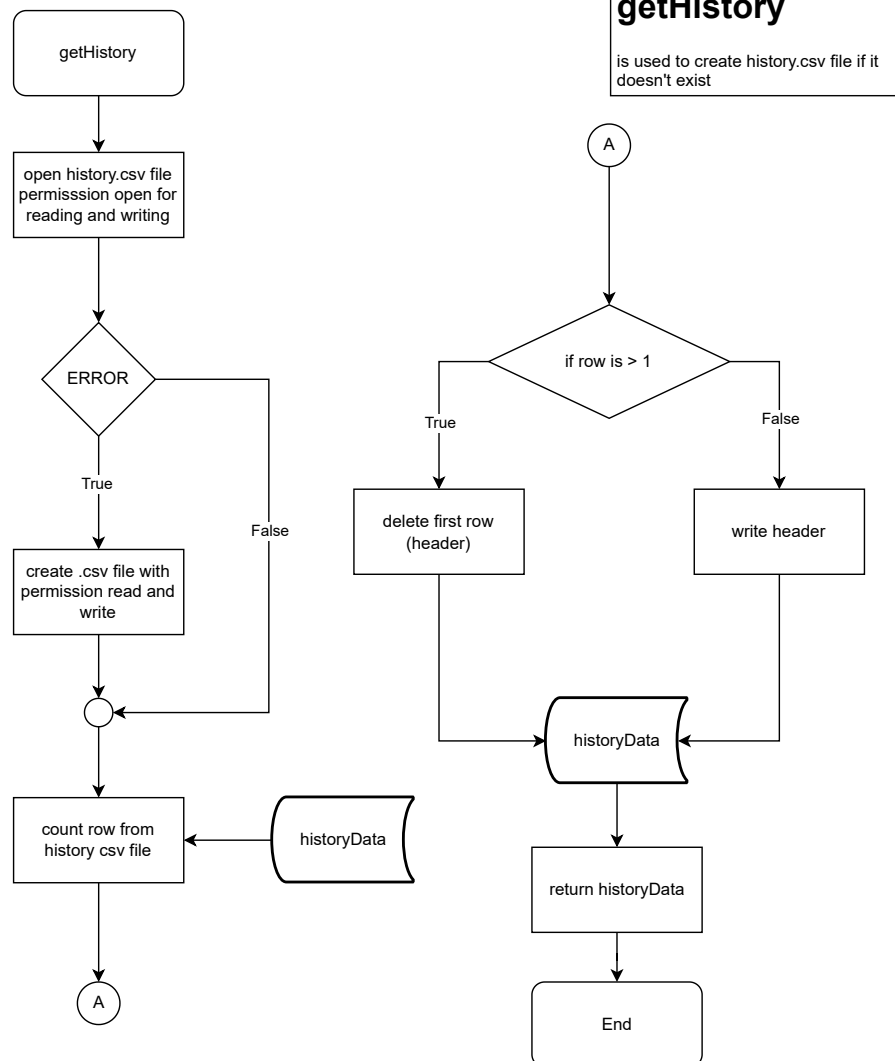
is used to add history to csv file



รูป 4.8 Flowchart ฟังก์ชัน updateHistory

2.6 Flowchart : ฟังก์ชัน getHistory เป็นฟังก์ชันที่เอาไว้สร้างไฟล์ csv หากไม่มี และเพิ่มส่วน header หากไฟล์ csv มีอยู่แล้ว เราจะลบ header ออก

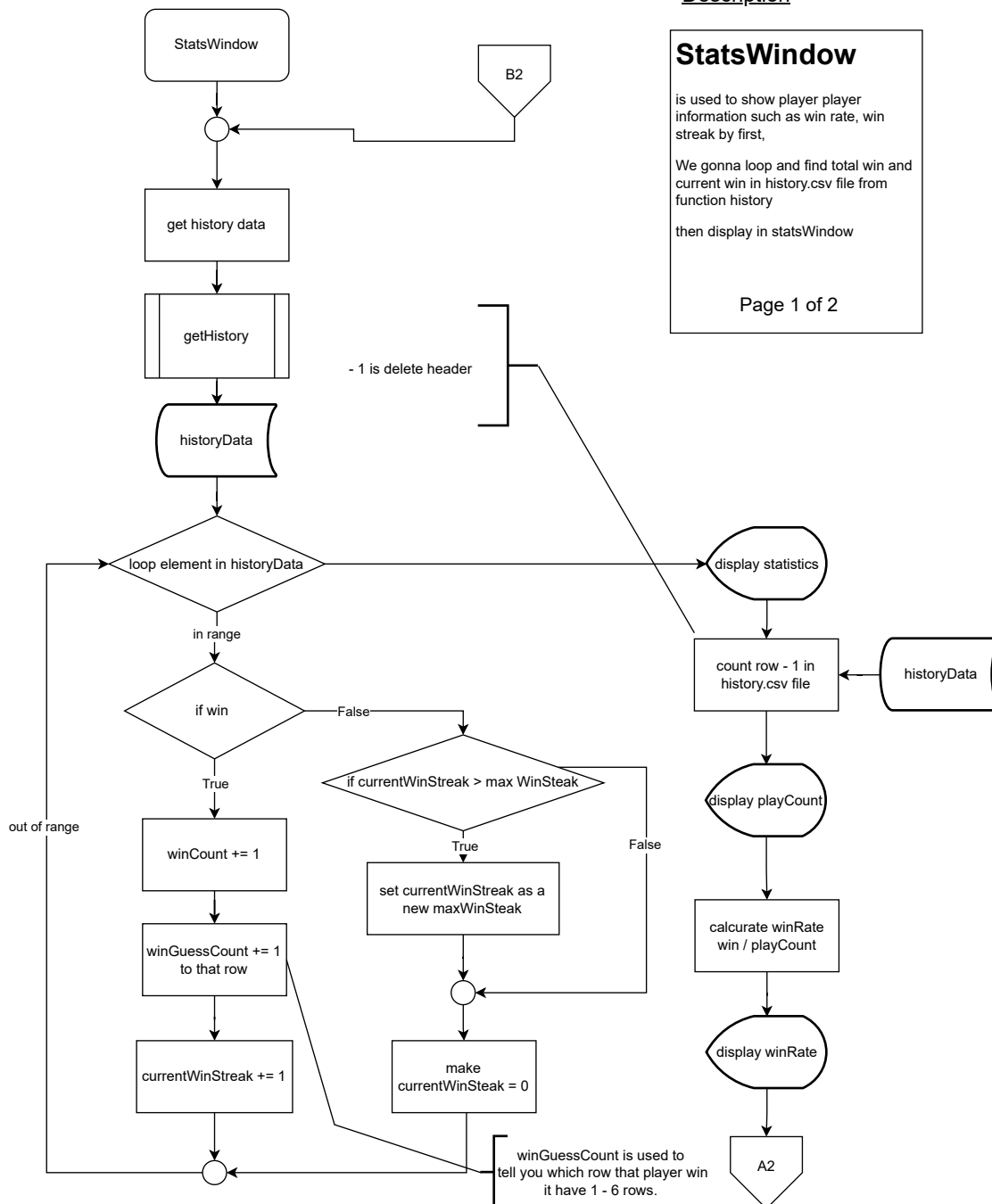
Flowchart: getHistory.



รูป 4.9 Flowchart ฟังก์ชัน getHistory

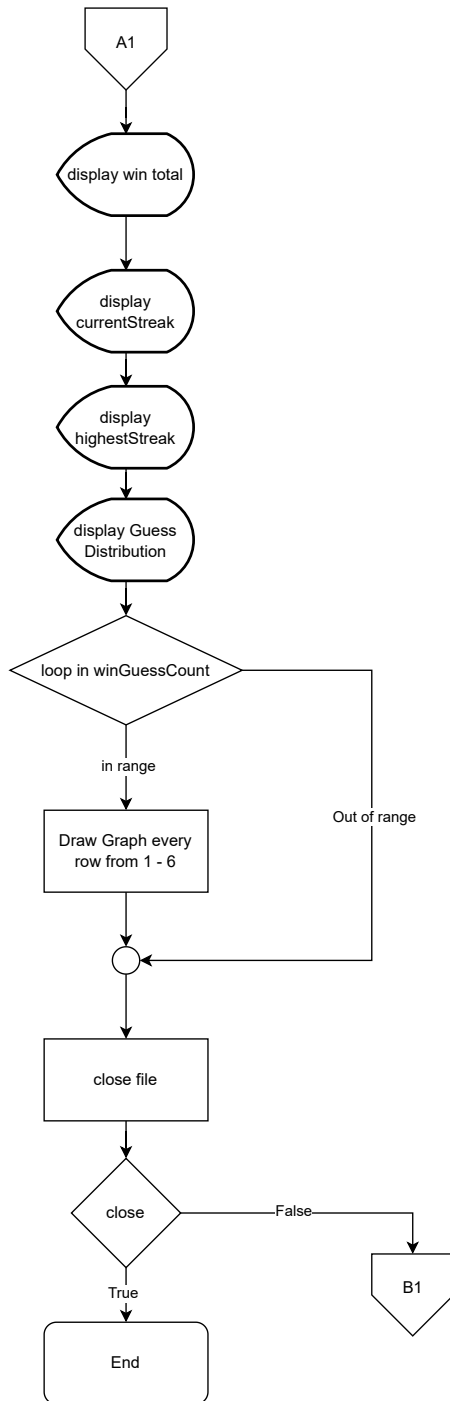
2.7 Flowchart : การสร้าง graphic หน้าต่าง สถิติข้อมูลของผู้เล่น

Flowchart: StatsWindow



รูป 4.10 Flowchart หน้าต่างสถิติ page 1

Flowchart: StatsWindow



Description

StatsWindow

is used to show player player information such as win rate, win streak by first,

We gonna loop and find total win and current win in history.csv file from function history

then display in statsWindow

Page 2 of 2

รูป 4.11 Flowchart หน้าต่างสถิติ page 2

บทที่ 5

Data file

1. ไฟล์ words สำหรับเป็นคลังศัพท์

ComPro1Project > words_			
1	shark		
2	green	14848	femal
3	white	14849	frame
4	world		
5	annoy	14850	abuse
6	space		
7	squad	14851	penny
8	crane		
9	apply	14852	stirp
10	aware	14853	input
11	angel		
12	light	14854	queen
13	speed		
14	apple	14855	change
15	array		
16	polio		

รูป 5.1 ตัวอย่างคำในคลังศัพท์

2. ไฟล์ csv สำหรับเก็บประวัติการเล่น โดยมี header ดังนี้

- 1.1 date : เวลาที่เกมนั้นจบลง (Epoch time)
- 1.2 target_word : คำตอบของเกมนั้น
- 1.3 guess_count : จำนวนที่ผู้เล่นตอบ (-1 ถ้าแพ้)
- 1.4 guess_word1 - guess_word6 : คำที่ผู้เล่นเดา

ComPro1Project > history.csv

```
1 date,target_word,guess_count,guess_word1,guess_word2,guess_word3,guess_word4,guess_word5,guess_word6
2 1665935033.9171999,frame,-1,hello,world,green,white,abuse,fetal
3 1665935210.8469172,adore,2,adorn,adore,,,,
4 1665935274.549983,bloom,-1,carry,catch,world,shark,bunch,blood
5 1665935335.549576,cache,6,input,cliff,candy,cheat,camel,cache
6 1665935378.6963441,email,5,world,erase,equip,embed,email,
7 1665935424.504321,grade,6,white,green,great,gloat,grace,grade
8 1665935509.846447,first,6,white,ghost,green,world,final,first
9 1665935566.3636632,latte,-1,least,leapt,label,layer,lapse
10 |
```

history

date	target_word	guess_count	guess_word1	guess_word2	guess_word3	guess_word4	guess_word5	guess_word6
1665935033.9172000	frame	-1	hello	world	green	white	abuse	fetal
1665935210.8469200	adore	2	adorn	adore				
1665935274.549980	bloom	-1	carry	catch	world	shark	bunch	blood
1665935335.549580	cache	6	input	cliff	candy	cheat	camel	cache
1665935378.6963400	email	5	world	erase	equip	embed	email	
1665935424.504320	grade	6	white	green	great	gloat	grace	grade
1665935509.846450	first	6	white	ghost	green	world	final	first
1665935566.3636600	latte	-1	least	leapt	label	later	layer	lapse

รูป 5.2 ตัวอย่างข้อมูลสถิติจากไฟล์ history.csv

บทที่ 6

1. **entryList**: ไว้เก็บ Entry ที่เป็นตารางการเล่น (5 x 6) สำหรับแก้ไขสี

ข้อมูลตัวอย่าง: entryList = [
 [<tkinter.Entry object .!frame29.!entry>, ...], # ขนาด 5
 [<tkinter.Entry object .!frame29.!entry>, ...], # ขนาด 5
 ...
] # ขนาด 6

2. **buttonList**: ไว้เก็บ Button ที่เป็นคีย์บอร์ดสำหรับแก้ไขสี (เก็บตามตัวอักษร)

ข้อมูลตัวอย่าง: buttonList = {
 'q': <tkinter.Button object .!frame.!button2>,
 'w': <tkinter.Button object .!frame2.!button2>,
 ...
}

3. **textVariableList**: เหมือนกับ entryList แต่เก็บ textVariable แทนไว้ใช้เปลี่ยนค่าที่แสดงอยู่

4. **wordsList**: ไว้เก็บคำศัพท์ของ Wordle ทั้งหมด (เปิดมาจากไฟล์)

5. **guessList**: ไว้เก็บคำที่ผู้เล่นเดา ในเกมนั้นๆ (รีเซ็ตทุกๆ เกม)

ข้อมูลตัวอย่าง: guessList = ['crane', 'plaza', 'quash', 'staff', 'toast', 'staid']

6. **targetWordCount**: จำนวนของแต่ละตัวอักษรของคำตอบ (รีเซ็ตทุกๆ เกม)

ข้อมูลตัวอย่าง: ถ้าคำตอบ = 'fells', targetWordCount = {
 'f': 1, 'e': 1, 'l': 2, 's': 1
}

7. **currWordState**: ไว้เก็บข้อมูล (ว่าเป็นสีเหลือง หรือเขียว) ของคำที่ผู้เล่นเดา (รีเซ็ตทุกๆ ครั้งที่ผู้เล่นพิมพ์)

ข้อมูลตัวอย่าง: ถ้าคำตอบ = 'wonts', ผู้เล่นเดาว่า 'pains', จะได้ currWordState = {
 0: {'char': 'p', 'color': 'gray'},
 1: {'char': 'a', 'color': 'gray'},
 2: {'char': 'l', 'color': 'gray'},
 3: {'char': 'n', 'color': 'yellow'},
 4: {'char': 's', 'color': 'green'}
}
}