

**Word Guessing Game**

จัดทำโดย

นาย ณัฐชนนท์ ศิรมลพิวัฒน์ 1670700044 **Section 127A**  
นางสาว ณิชาภรณ์ ช้อยเพ็ง 1670700333 **Section 127A**

นาย เมฆินทร์ ขันธวิชัย 1670703675 **Section 127C**

อาจารย์ผู้สอน

ผู้ช่วยศาสตราจารย์สิรินธร จียาศักดิ์

**หลักสูตรวิทยาศาสตรบัณฑิต**

**ภาคการศึกษาที่ 1 ปีการศึกษา 2567**

**ภาควิชาวิทยาการคอมพิวเตอร์คณะเทคโนโลยีสารสนเทศและนวัตกรรม**

**มหาวิทยาลัยกรุงเทพ**

**สารบัญ**

หน้า

บทที่

1. บทนำ 3

1.1 ชื่อโครงงาน 3

1.2 วัตถุประสงค์ของโครงงาน 3

1.3 ขอบเขตของโครงงาน 3-4

2. การทำงานของแอปพลิเคชันหรือโปรแกรม 5

2.1 แผนภาพแสดงกระบวนการทำงานของแอปพลิเคชัน (Flowchart) 5-6

1. ฟังก์ชันRegister Login 7-10

2. ฟังกชัน Play 11

3. ฟังก์ชัน Profile 12

4. ฟังก์ชัน Ranking 13

5. ฟังก์ชัน Logout 14

6. else Register Login 15

7. ฟังก์ชัน Score Unlock mode 16

8. else Play mode 17

9. ฟังก์ชัน Questions You Want to Play 18

10. ฟังก์ชัน Hint Category 19

11. ฟังก์ชัน < Go Back 20

12. ฟังก์ชัน Save Data Player in data\_user.txt 21

ภาคผนวก ก Source Code (.py) 22-31

**บทที่ 1**

**บทนำ**

## ชื่อโครงงาน

**Word Guessing Game *(เกมส์ทายคำศัพท์)***

## วัตถุประสงค์ของโครงงาน

1. เพื่อการพัฒนาทักษะการเขียนโปรแกรม Python เป็น แอพพลิเคชั่น Console และได้นำความรู้ที่เรียนมามาทำเป็น โครงงาน เพื่อต่อยอดได้

2. เป็นการส่งเสริมการเรียนรู้ภาษาอังกฤษ ที่เป็นภาษาหลักของโลก เพื่อนำไปใช้ในชีวิตประจำวัน

3. เพื่อสร้างความสนุกสนานแก่ผู้ใช้งานที่ได้เล่นเกมทายคำศัพท์

4. มีการใช้ความคิด ต่อยอดเชื่อมโยง ระหว่างภาษา ฝึกความคิด ความจำ

5. มีการฝึกฝนการใช้งานแนวคิดจาก AI มาต่อยอดเพื่อให้งานดียิ่งขึ้น

* 1. **ขอบเขตของโครงงาน**

1. เกมจะเป็นแอปพลิเคชันแบบคอนโซลที่เขียนด้วยภาษา Python และรองรับการใช้งานบนระบบปฏิบัติการ Windows และ macOS และ อื่นๆ ถ้ามีการพัฒนาต่อในอนาคต

2. มีเนื้อหา เกี่ยวกับ สิ่งที่เรียนมาทั้งหมด เช่น Variable , input , output , if else , loop while for , textfile , string และเนื้อหาอื่นๆ ที่ศึกษาเพิ่มเติม

3. โปรแกรมมีฟังก์ชันในการทำงาน ได้แก่ 1 Register + Login , 2 Play Mode < Easy , Normal , Hard , Random > , 3 Show Profile , 4 Show Ranking , 5 Logout , 6 Score Unlock Mode , 7 Questions You Want To Play , 8 Hint , Category , 9 Go Back , 10 Save Data Player in data\_user.txt

4.เกมมีการใช้ไฟล์ file.txt เพื่อเก็บคำศัพท์และคำใบ้ และ data\_user.txt เพื่อบันทึกข้อมูลคะแนนและสถิติการเล่นของผู้เล่น

5. ระบบ Register + Login โดยผู้เล่น สามารถ ลงทะเบียน ผ่านระบบ Register และ สามารถ Login ได้ เพื่อเข้าสู่ระบบ ไป เล่นเกม **Word Guessing Game *ต่อไป***

6. **ระบบ** Logout ผู้เล่นสามารถ กด Logout เพื่อออก ไปหน้า Login/Register ของโปรแกรม ที่ผู้ใช้สามาร Register , Login และ Exit ได้

7. ระบบ Play Mode ผู้เล่นสามารถทำการทายคำศัพท์โดยระบบจะให้คำใบ้และหมวดหมู่สำหรับคำที่เลือกแบบสุ่มจากไฟล์คำศัพท์ ระดับ ง่าย ธรรมดา ยาก หรือ สุ่ม ( Easy , Normal , Hard , Random )

6. ระบบ Show Profile ผู้เล่นสามารดูข้อมูล โปรไฟล์ของตัวเองได้ โดยมีเนื้อหา ดังนี้ ชื่อ คะแนนปัจจุบัน จำนวนครั้งตอบถูก จำนวนครั้งตอบผิด จำนวนคำถามทั้งหมดที่เล่น

8. ระบบ Show Ranking ผู้เล่นสามารถดูคะแนนอันดับของตัวเองได้ โดยอ้างอิงจาก Score จากจำนวนผู้เล่นทั้งหมด

( ระบบ Ranking )

9. ระบบ Score Unlock Mode ผู้เล่นสามารถเล่นโหมดต่างๆ ได้เมื่อ Score ถึงที่กำหนด

( [ Easy >= 0 Score ] , [ Normal >= 25 Score ] , [ Hard >= 75 Score ] ,

[ Random >= 100 Score ] , )

10. ระบบ Questions You Want To Play ผู้เล่นสามารถกำหนด จำนวนข้อที่อยากเล่นเองได้

11. ระบบ Hint , Category เมื่อผู้เล่นเริ่มทายคำศัพท์ จะมี คำใบ้ และ หมวดของคำใบ้ให้ ผู้เล่นสามารถเดาคำศัพท์ได้

12. ระบบ Count + Calculate Score เมื่อผู้เล่น ตอบถูก Score จะ +1 เมื่อ ผู้เล่นตอบผิด Score จะ -1 และ มี การเก็บ Score จำนวนครั้งตอบถูก จำนวนครั้งตอบผิด จำนวนคำถามทั้งหมดที่เล่น

13. ฟังก์ชัน Go Back เมื่อผู้เล่นกด ในหน้า Play Mode จะกลับไปหน้าเมนูหลัก( Main Program )

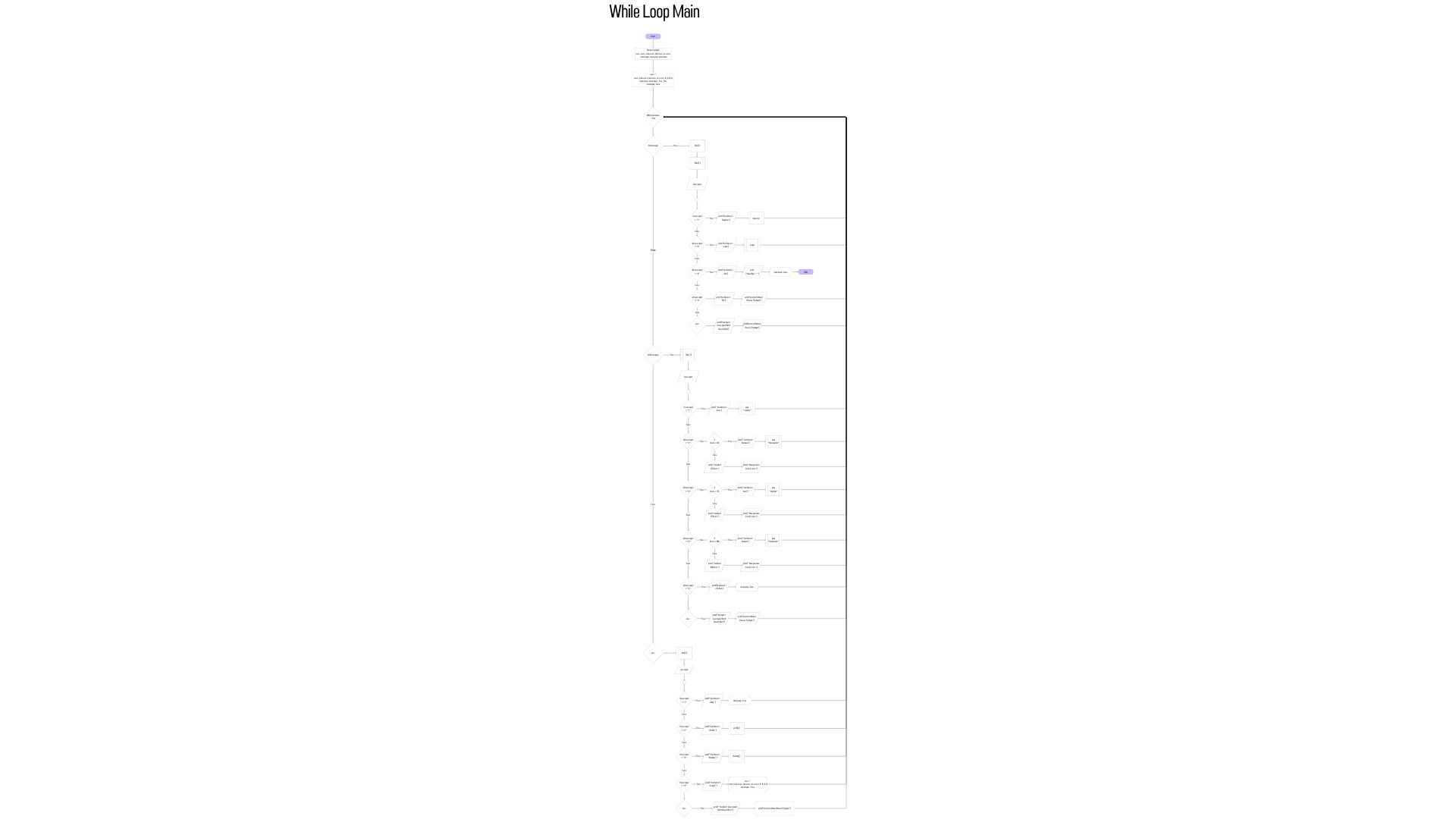
14. ฟังก์ชัน Save Data Player เป็นฟังก์ชันที่ใช้สำหรับ Save ข้อมูลผู้เล่นลงใน data\_user.txt

โดยผู้เล่นสามารถกลับมาเล่นใหม่ได้โดยใช้ข้อมูลเดิม

**บทที่ 2**

**การทำงานของแอปพลิเคชัน**

**2.1 ฟังก์ชันการทำงานหลักของ โปรแกรม Word Guessing Game**



**ภาพที่ 2.1** ภาพแสดงการทำงานรวมของโปรแกรม Word Guessing Game

ไฟล์ PDF Flowchart ของโปรแกรม <https://drive.google.com/file/d/1BvqAbWJxRtZU4E6CfKKK7UALIMPk4EXl/view?usp=sharing>

**โหลดไฟล์ดูเพื่อความชัดเจนของภาพ**

**การทำงานของ Flowchart โปรแกรม Word Guessing Game**

2.1.1 การทำงาน ของ FlowChart นี้ เริ่มต้น สร้างตัวแปร และ กำหนดค่าตัวแปร เมื่อ While main\_menu = true ลูปจะทำงาน และ เมื่อ check\_login = true จะเข้าหน้าเมนู Regester Login เมื่อ check\_login = false จะลงไปเช็ค check\_play และ เมื่อ check\_play = true จะเข้าหน้าเมนูสำหรับเลือกโหมดในการเล่น และ เมื่อ check\_play = false จะลงไปทำ else ซึ่งเป็นหน้าเมนูหลัก

2.1.2 ในหน้า Regester Login จะรับ input แล้วถ้า input = 1 จะเข้าหน้า Regester ถ้า input = 2 จะเข้าหน้า Login ถ้า input = 3 จะออกจากโปรแกรม

2.1.3 ในหน้า Main program จะรับ input แล้วถ้า input = 1 จะเข้าหน้า Paly mode ถ้า input = 2 จะเข้าหน้า Profile ถ้า input = 3 จะเข้าหน้า Ranking ถ้า

input = 4 จะทำการ Logout และจะกลับไปหน้า Regester Login

2.1.4 ในหน้า Paly mode จะรับ input แล้วถ้า input = 1 จะเป็นการเลือกโหมด Easy ถ้า input = 2 จะเป็นการเลือกโหมด Normal โดยจะต้องมี Score 25 ขึ้นไป ถ้า input = 3 จะเป็นการเลือกโหมด Hard โดยจะต้องมี Score 75 ขึ้นไป ถ้า input = 4 จะเป็นการเลือกโหมด Random โดยจะต้องมี Score 100 ขึ้นไป ถ้า input = 5 จะเป็นการกลับไปหน้า Main program

**2.2 เมนู Register Login**

**2.2.1 ฟังก์ชัน (1) Register + Login**

A screen shot of a game

Description automatically generated

รูปที่ 2.2.1 หน้าRegister Login

คำอธิบาย

เป็นฟังก์ชันที่ผู้เล่นจะเจอเมื่อรันโปรแกรมโดยจะแสดง หัวข้อมาให้ ดังนี้

Register , Login และ Exit โดยให้ ผู้เล่น Input ตัวเลข เพื่อไปยังหัวข้อต่อไป

**เมื่อผู้เล่น Input เลข 1 เข้า เมนู Register**

A screenshot of a computer screen

Description automatically generated

รูปที่ 2.2.2 หน้าRegister

คำอธิบาย หน้า Register

เป็นเมนูสำหรับลงทะเบียนในเกม โดย จะให้ผู้ใช้ Input Username และ Password

โดยมีระบบตรวจสอบความถูกต้องของ Username และ Password ที่ผู้เล่นกรอกมา ถ้า Usernameมีอยู่แล้วจะให้ผู้ใช้กรอกใหม่ และ ถ้า Password ไม่เหมือนกันจะให้ผู้ใช้กรอกใหม่ เมื่อลงทะเบียนเสร็จแล้วจะขึ้น ว่า ลงทะเบียนสำเร็จ โดยข้อมูลผู้ใช้ และ Password จะถูกเก็บในไฟล์ data\_user.txt เพื่อดึงข้อมูลมาตรวจสอบ ตอน ผู้ใช้ Login อีกที

**เมื่อผู้เล่น Input เลข 2 เข้า เมนู Login**

A screenshot of a computer screen

Description automatically generated

รูปที่ 2.2.3 หน้า Login

คำอธิบาย หน้า Login

เป็นเมนูสำหรับล็อกอินเกม โดย จะให้ผู้เล่น Input Username และ Password

ที่ผู้ใช้ลงทะเบียนไว้ ในหน้า Register เพื่อเข้าสู่ระบบโดยเมื่อผู้เล่นกรอกข้อมูลเสร็จแล้วจะแสดงข้อความว่า ล็อกอินสำเร็จ และ จะเข้าหน้า Main Program

A screen shot of a computer screen

Description automatically generated

รูปที่ 2.2.4 หน้า Login Fail

คำอธิบาย หน้า Login Fail

\* เมื่อผู้เล่นกรอกข้อมูลผิด หรือ ไม่ตรง จะแสดงข้อความว่า Username หรือ Password ผิดให้ผู้เล่นกรอกข้อมูลใหม่อีกรอบ

**เมื่อผู้เล่น Input เลข 3 เข้า เมนู Exit**

A screenshot of a computer screen

Description automatically generated

รูปที่ 2.2.5 หน้า Exit

คำอธิบาย หน้า Exit

เป็นเมนูสำหรับออกจากเกม โดยจะแสดงข้อความ Good Bye และ จะปิดโปรแกรมลง

**เมื่อผู้เล่น Input นอกเหลือจาก 1 2 3**

A screen shot of a black screen

Description automatically generated

รูปที่ 2.2.6 หน้า Other เมนูRegister Login Input

คำอธิบาย หน้า Other Register LoginInput

เป็นการแจ้งเตือนกันข้อผิดพลาดของการกรอกข้อมูลของผู้เล่นโดยจะแสดงสิ่งที่ผู้เล่นกรอกมาและบอกว่า นอกเหนือจากเมนูที่กรอกได้ ให้ลองใหม่อีกครั้ง

**2.3 เมนู**  **Main Program**

**2.3.1 ฟังก์ชัน (2) Play**

**เมื่อผู้เล่น Input เลข 1 เข้า ฟังก์ชัน Play**

A screen shot of a game menu

Description automatically generated

รูปที่ 2.3.1 หน้าPlay

คำอธิบาย หน้าPlay

เป็นเมนูสำหรับเข้าเล่นเกม โดยจะแสดงโหมดต่างๆ ให้ผู้เล่นได้เล่น เช่น

Easy , Normal , Hard , Random โดยจะมีการดึงข้อมูลคำศัพท์ แบบสุ่มจากไฟล์ txt ในแต่ละข้อที่เลือก

**2.3.2 ฟังก์ชัน (3) Profile**

**เมื่อผู้เล่น Input เลข 2 เข้า ฟังก์ชัน Profile**

A screen shot of a computer menu

Description automatically generated

รูปที่ 2.3.2 หน้าProfile

คำอธิบาย หน้าProfile

เป็นเมนูสำหรับแสดงข้อมูลผู้เล่น โดยจะแสดง ชื่อผู้ใช้ , Score ปัจจุบัน , คำถามที่ตอบถูก , คำถามที่ตอบผิด และ จำนวนข้อทั้งหมดที่เล่น ตามลำดับ

**2.3.3 ฟังก์ชัน (4) Ranking**

**เมื่อผู้เล่น Input เลข 3 เข้า ฟังก์ชัน Ranking**

A screen shot of a game menu

Description automatically generated

รูปที่ 2.3.3 หน้าRanking

คำอธิบาย หน้าRanking

ฟังก์ชันแสดงอันดับของผู้เล่นทั้งหมดในเกม โดยอิงจาก Score ของผู้เล่น โดยเรียงลำดับจากScore มากสุด เป็นอันดับ 1 และ ไล่ลงมาตามลำดับของ Score ไปจนลำดับสุดท้าย ของ ผู้เล่นใน data\_user.txt ซึ่งเป็นไฟล์เก็บข้อมูล

**2.3.4 ฟังก์ชัน (5) Logout**

**เมื่อผู้เล่น Input เลข 4 เข้า ฟังก์ชัน Logout**

A screen shot of a computer

Description automatically generated

รูปที่ 2.3.4 หน้าLogout

คำอธิบาย หน้าLogout

เป็นฟังก์ชันสำหรับการออกจากระบบการลงชื่อเข้าใช้ เพื่อเข้าสู่หน้า เมนู Register Login

**เมื่อผู้เล่น Input นอกเหนือจาก 1 2 3 4**

A screen shot of a computer menu

Description automatically generated

รูปที่ 2.3.5 หน้าOther Main Program

คำอธิบาย หน้าOther Main Program

เป็นการแจ้งเตือนกันข้อผิดพลาดของการกรอกข้อมูลของผู้เล่นโดยจะแสดงสิ่งที่ผู้เล่นกรอกมาและบอกว่า นอกเหนือจากเมนูที่กรอกได้ ให้ลองใหม่อีกครั้ง

**2.4 เมนู**  **Play mode**

**2.4.1 ฟังก์ชัน (6) Score Unlock Mode**

**เมื่อผู้เล่น Input ค่า แล้ว Score ไม่เท่ากับที่โหมดกำหนด**

A screenshot of a game

Description automatically generated

รูปที่ 2.4.1 หน้าScore Unlock Mode

คำอธิบาย หน้าScore Unlock Mode

เป็นโหมดต่างๆในการการเล่น โดยจะปลดล็อคตาม Score ที่กำหนดไว้ ดังนี้

( [ Easy >= 0 score ] , [ Normal >= 25 score ] , [ Hard >= 75 score ]

, [ Random >= 100 score] ) ถ้า ผู้เล่น มี Score ไม่ถึง โปรแกรมจะขึ้นแสดง Score ตอนนี้ที่มีอยู่

และ เมื่อ ผู้เล่นมี Score ครบตามที่กำหนด ผู้เล่นก็จะเข้าเล่นได้

**เมื่อผู้เล่น Input นอกเหนือจาก 1 2 3 4 5**

A screen shot of a black screen

Description automatically generated

รูปที่ 2.4.2 หน้าOther Main Program

คำอธิบาย หน้าOther Main Program

เป็นการแจ้งเตือนกันข้อผิดพลาดของการกรอกข้อมูลของผู้เล่นโดยจะแสดงสิ่งที่ผู้เล่นกรอกมาและบอกว่า นอกเหนือจากเมนูที่กรอกได้ ให้ลองใหม่อีกครั้ง

**2.4.2 ฟังก์ชัน (7) Questions You Want To Play**

**เมื่อผู้เล่น กรอก จำนวนข้อที่ผู้เล่นอยากจะเล่น**

A screenshot of a computer game

Description automatically generated

รูปที่ 2.4.3 หน้าScore Unlock Mode

คำอธิบาย หน้าQuestions You Want To Play

เป็นฟังก์ชัน ที่ จะให้ผู้เล่น สามารถ กรอก จำนวนข้อที่อยากจะเล่นเองได้โดยโปรแกรม จะ นำค่าที่ผู้เล่นกรอก มาสุ่มคำใบ้ ตามจำนวนข้อ

**2.4.3 ฟังก์ชัน (8) Hint Category**

**เมื่อผู้เล่น กรอก จำนวนข้อที่ผู้เล่นอยากจะเล่น**

A screenshot of a computer game

Description automatically generated

รูปที่ 2.4.4 หน้าHint Category

คำอธิบาย หน้าHint Category

เป็นฟังก์ชันสุ่มคำใบ้ และ หมวดหมู่คำใบ้ เพื่อให้ผู้ใช้ ทายคำศัพท์ และ เมื่อผู้ใช้ทายถูก จะขึ้นข้อความแสดงว่า ผู้ใช้ตอบถูก และ ขึ้น Score +1 พร้อม แสดง Score ปัจจุบัน และ เมื่อผู้ใช้ทายผิด จะขึ้นข้อความแสดงว่า ผู้ใช้ตอบผิด และจะเฉลยคำที่ตอบคำที่ตอบผิด และ ขึ้น Score -1 พร้อม แสดง Score ปัจจุบัน

**2.4.4 ฟังก์ชัน (9) < Go Back**

**เมื่อผู้เล่น กรอก 5 จะกลับไปหน้า Main Program**

A screen shot of a game menu

Description automatically generated

รูปที่ 2.4.5 หน้า< Go Back

คำอธิบาย หน้า< Go Back

เป็นฟังก์ชันย้อนกลับ เพื่อ ย้อนกลับไปหน้า Main Program

**2.4.5 ฟังก์ชัน (10) Save Data Player in data\_user.txt ( Inside Function )**

**เมื่อผู้เล่นทายคำศัพท์เสร็จหมดทุกข้อจะ Save Data User ทั้งหมด**

A screenshot of a computer game

Description automatically generated

รูปที่ 2.4.6 หน้าSave Data

คำอธิบาย หน้าSave Data

เมื่อผู้เล่นทายคำศัพท์เสร็จหมดทุกข้อจะ Save Data User ทั้งหมดไว้ใน data\_user.txt และ เมื่อผู้เล่นกลับมา Login ใหม่ ก็จะสามารถดึงค่าจากไฟล์ data\_user.txt มาใช้ใหม่ได้อีกรอบ

**ภาคผนวก**

import random

# Random Word

def title(title\_num) :

if title\_num == 1 : # Start Game Word Guessing Game MENU

print("-"\*40)

print(f"- Word Guessing Game - ".center(40))

elif title\_num == 2 : # Menu login Register Exit

print(f"- SELECT MENU -".center(40))

print("-"\*40)

print(f"1 Register\n2 Login\n3 Exit")

print("-"\*40)

elif title\_num == 3 : # REGESTER MAIN

print("-"\*40)

print(f"REGESTER".center(40))

print("-"\*40)

elif title\_num == 4 : # LOGIN MAIN

print("-"\*40)

print(f"LOGIN".center(40))

print("-"\*40)

elif title\_num == 5 : # Game Menu

print("-"\*40)

print(f"- GAME MENU -".center(40))

print("-"\*40)

print(f"1 Play\n2 Profile\n3 Ranking\n4 Logout")

print("-"\*40)

elif title\_num == 6 : # Play Menu

print("-"\*40)

print(f"- SELECT PLAY MENU -".center(40))

print("-"\*40)

print(f"1 Easy\n2 Normal\n3 Hard \n4 Random \n5 < Go back")

print("-"\*40)

def play(filename) :

global score

global total\_score

global fail\_score

global nc\_score

words = []

with open(filename, "r") as file:

for line in file:

word, hint, category = line.strip().split(":")

words.append((word, hint, category))

print("-"\*40)

user\_input = int(input("Choose How Many Questions You want to play : "))

for i in range(user\_input) :

print("-"\*40)

word, hint, category = random.choice(words)

print(f"{i+1}.Category : {category}")

print(f"Hint : {hint}")

guess = input("Input Guess The Word : ")

if guess.lower() == word.lower():

score += 1

total\_score += 1

nc\_score += 1

print("That's right ! Now You Have Guessed The Word Correctly")

print("Score +1")

else:

score -= 1

total\_score += 1

fail\_score += 1

print(f"The correct answer is {word}")

print("You guessed wrong ! Please Try Again")

print("Score -1")

print(f"Now you have {score} score")

save\_user()

def regester() :

title(3)

check\_username\_found = True

while check\_username\_found :

with open("data\_user.txt","r") as file :

username = input("Please Enter Username : ")

data = file.readlines()

check\_username = False

for i in data :

item = i.split()

if username == item[0] :

check\_username = True

if check\_username :

print("This Username Already Exists !")

else :

check\_username\_found = False

password = input("Please Enter Password : ")

check\_password\_found = True

while check\_password\_found :

confirm\_password = input("Please Confirm Password : ")

if password != confirm\_password :

print("Invalid Confirm Password Please Try Again.")

else :

print("Regester Succeed !")

check\_password\_found = False

with open("data\_user.txt","a") as file :

file.write(f"{username} {password} 0 0 0 0\n")

def login() :

title(4)

username = input("Please Enter Username : ")

password = input("Please Enter Password : ")

with open("data\_user.txt","r") as file :

data = file.readlines()

check\_fail = True

for i in data :

item = i.split()

if username == item[0] and password == item[1] :

print("Login Succeed !")

check\_fail = False

global check\_login

global user

global score

global total\_score

global fail\_score

global nc\_score

check\_login = False

user = item[0]

score = int(item[2])

total\_score = int(item[3])

fail\_score = int(item[4])

nc\_score = int(item[5])

if check\_fail == True:

print("Username Or Password Is Incorrect Please Try Again.")

def save\_user() :

info = []

with open("data\_user.txt","r") as file :

data = file.readlines()

for i in data :

item = i.split()

if user == item[0] :

info.append([item[0],item[1],score,total\_score,fail\_score,nc\_score])

else :

info.append([item[0],item[1],item[2],item[3],item[4],item[5]])

with open("data\_user.txt","w") as file :

for item in info :

file.write(f"{item[0]} {item[1]} {item[2]} {item[3]} {item[4]} {item[5]}\n")

def profile() :

print("-"\*40)

print('Username : ',user)

print('Score : ',score)

print('Succeed : ',nc\_score)

print('Fail : ',fail\_score)

print('Total : ',total\_score)

def ranking() :

info = []

with open("data\_user.txt","r") as file :

data = file.readlines()

for i in data :

item = i.split()

info.append([item[0],int(item[2])])

for i in range(len(info)):

for j in range(i + 1, len(info)):

if info[i][1] < info[j][1]:

info[i], info[j] = info[j], info[i]

print("-"\*40)

index = 1

for item in info:

print(f"{index}.{item[0]} Score = {item[1]}")

index += 1

user = ''

score = 0

total\_score = 0

fail\_score = 0

nc\_score = 0

check\_login = True

check\_play = False

main\_menu = True

while main\_menu :

if check\_login :

title(1)

title(2)

user\_input = input("Please Choich Number : ")

if user\_input == "1" :

print('You Selcect > Regester')

regester()

elif user\_input == "2" :

print('You Selcect > Login')

login()

elif user\_input == "3" :

print('You Selcect > Exit')

print("Good Bye ^-^")

main\_menu = False

else :

print(f"You Input > {user\_input} But It Doesn't Exist")

print("Incorrect Menu !! Please Try Again")

elif check\_play :

title(6)

user\_input = input("Please Choich Number : ")

if user\_input == "1" :

print('You Selcect > Easy')

play("easy.txt")

elif user\_input == "2" :

if score >= 25 :

print('You Selcect > Normal')

play("normal.txt")

else :

print('You need 25 score')

print(f"Now you have {score} score")

elif user\_input == "3" :

if score >= 75 :

print('You Selcect > Hard')

play("hard.txt")

else :

print('You need 75 score')

print(f"Now you have {score} score")

elif user\_input == "4" :

if score >= 100 :

print('You Selcect > Random')

play("random.txt")

else :

print('You need 100 score')

print(f"Now you have {score} score")

elif user\_input == "5" :

print('You Selcect < Goback')

check\_play = False

else :

print(f"You Input > {user\_input} But It Doesn't Exist")

print("Incorrect Menu !! Please Try Again ")

else :

title(5)

user\_input = input("Please Choich Number : ")

if user\_input == "1" :

print('You Selcect > Play')

check\_play = True

elif user\_input == "2" :

print('You Selcect > Profile')

profile()

elif user\_input == "3" :

print('You Selcect > Ranking')

ranking()

elif user\_input == "4" :

print('You Selcect > Logout')

user = ''

score = 0

total\_score = 0

fail\_score = 0

nc\_score = 0

check\_login = True

else :

print(f"You Input > {user\_input} But It Doesn't Exist")

print("Incorrect Menu !! Please Try Again ")