

Term Project Group 9: พ่อหนุ่มมมม ไปไหนมาจ๊ะ

66070501022 ธนภัทร นามสมบูรณ์

66070501033 ปิณณวิชญ์ หวานแท้

66070501042 ภูริณัฐ พลอาสา

66070501053 ศักซัซ เห็นสุข

66070501056 สันติภาพ ทองจันทร์

66070501058 สุวิจักขณ์ เรืองอินตา

ชื่อโปรแกรม: Blackjacks

บทนำ

เกม blackjacks มีเป้าหมายสูงสุดของเกม คือ การมีแต้มที่ใกล้เคียงกับ 21 หรือ มีแต้มสูงสุดจากผู้เล่นทั้งหมด โดยที่เริ่มเกมผู้เล่นจะได้รับการ์ดไพ่สุ่มจากระบบจำนวน 2 ใบ หลังจากนั้นระบบจะให้ผู้เล่นเลือกว่าจะจั่วไพ่เพิ่มอีกหรือไม่ โดยผู้เล่นจะพิจารณาจากแต้มของไพ่ในมือ โดยที่ แต้มทั้งหมดจะนับจากเลขไพ่ (2-9), A = 1 หรือ 11, J Q K = 10 เกมจะจบเมื่อผ่านการตัดสินใจจั่วหรือไม่จั่วของผู้เล่นทั้งหมด หลังจากนั้นระบบจะนับแต้มที่อยู่บนมือของผู้เล่น โดยที่ผู้เล่นจะต้องมีแต้มไม่เกิน 21 และมีค่าใกล้เคียง 21 มากที่สุด

ดังนั้น การพัฒนาเกม blackjacks นี้จะใช้เนื้อหาในรายวิชา CPE100- Computer Programming for Engineers ดังนี้ Function, Array, Pointer, Decision, Input/ Output, etc. ในการสร้างฟังก์ชัน shuffle, random, score, display, pop_card, add_card, display, seed_generator และฟังก์ชัน main

วิธีการดำเนินงาน

i. Flowchart

ii. Code Program

ในส่วน code ของโปรแกรมนี้นี้ ถูก upload ไว้ที่

<https://github.com/pannawittt/CPE100/blob/main/project/blackjack.c>

และสามารถเข้าถึงได้โดยสาธารณะ

iii. วิธีการใช้งานและการทดสอบการทำงานของโปรแกรม

กฎการเล่น Blackjack เบื้องต้น

- วัตถุประสงค์: ผู้เล่นต้องพยายามรวบรวมนับแต้มในมือให้เท่ากับ 21 หรือใกล้เคียงมากที่สุด โดยที่ไม่เกิน
- ค่าของไพ่:
 - ไพ่ตัวเลขมีค่าตามค่านับ (2-9)

- ไพ่หน้า (K, Q, J) มีค่าแต่ละใบเป็น 10 คะแนน
- A สามารถมีค่าเป็น 1 หรือ 11 คะแนน ขึ้นอยู่กับความเหมาะสมของผู้เล่น
- การเล่น:
 - ผู้เล่นจะได้รับไพ่ 2 ใบในตอนเริ่มต้น
 - ผู้เล่นสามารถเลือกที่จะ "ต่อ" (รับไพ่เพิ่ม) หรือ "ยืน" (ไม่รับไพ่เพิ่ม)
 - ผู้แจกจะเล่นตามกฎที่กำหนดไว้
- การชนะ:
 - ผู้เล่นที่บรรลุค่ามือที่ใกล้เคียงกับ 21 มากที่สุดโดยไม่เกิน จะเป็นผู้ชนะ
 - "Blackjack" (A, K, Q, J) เป็นมือที่สูงสุด
- "แต้มมือ" หมายถึง ผลรวมของค่าไพ่ทั้งหมดที่อยู่ในมือของผู้เล่น
- "Blackjack" หมายถึง มือที่ประกอบไปด้วย A, J, Q, K

การทำงานของโปรแกรม C ในการเล่น แบล็คแจ็ค

● shuffle

ฟังก์ชัน shuffle ทำหน้าที่สับไพ่ในสสารับไพ่ ฟังก์ชันนี้จะสุ่มเลือกไพ่ในสสารับไพ่และวางไพ่นั้นไว้ในตำแหน่งใหม่ ฟังก์ชันนี้เทียบเท่ากับการแจกไพ่โดยเจ้ามือก่อนเริ่มเกม

● random

ฟังก์ชัน random ทำหน้าที่สร้างตัวเลขสุ่ม ฟังก์ชันนี้จะสร้างตัวเลขสุ่มระหว่าง 0 ถึงค่าที่กำหนด ฟังก์ชันนี้ใช้สำหรับการสับไพ่และตัดสินใจอื่นๆ ของเกมที่ต้องมีการสุ่ม เช่น การตัดสินใจของผู้เล่นว่าจะตีไพ่หรือไม่

● score

ฟังก์ชัน score ทำหน้าที่คำนวณแต้มของผู้เล่นหรือเจ้ามือ ฟังก์ชันนี้จะนับจำนวนไพ่ที่มีค่า 10 ขึ้นไปในมือของผู้เล่นหรือเจ้ามือ ฟังก์ชันนี้เทียบเท่ากับผู้เล่นหรือเจ้ามือ นับแต้มไพ่ของตนเอง

● display

ฟังก์ชัน display ทำหน้าที่แสดงไพ่ของผู้เล่นและเจ้ามือ รวมถึงแต้มของไพ่เหล่านั้น ฟังก์ชันนี้จะแสดงไพ่ของผู้เล่นและเจ้ามือ โดยหงายไพ่ขึ้น ฟังก์ชันนี้เทียบเท่ากับเจ้ามือแจกไพ่ให้ผู้เล่นและหงายไพ่ขึ้น

● pop_card

ฟังก์ชัน pop_card ทำหน้าที่ดึงไพ่ออกจากสสารับไพ่ ฟังก์ชันนี้จะลบไพ่ออกจากสสารับไพ่และส่งไพ่นั้นคืน ฟังก์ชันนี้เทียบเท่ากับเจ้ามือแจกไพ่จากสสารับไพ่ให้ผู้เล่นหรือเจ้ามือ

- **add_card**

ฟังก์ชัน `add_card` ทำหน้าที่เพิ่มไพ่ให้กับมือของผู้เล่นหรือเจ้ามือ ฟังก์ชันนี้จะเพิ่มไพ่เข้าไปในมือของผู้เล่นหรือเจ้ามือ ฟังก์ชันนี้เทียบเท่ากับผู้เล่นหรือเจ้ามือตีไพ่และได้รับไพ่อีกใบจากสำรับไพ่

- **seed_generator**

ฟังก์ชัน `seed_generator` ทำหน้าที่สร้าง seed สำหรับตัวสร้างตัวเลขสุ่ม ฟังก์ชันนี้จะสร้าง seed จากชื่อของผู้เล่น ฟังก์ชันนี้ช่วยให้แน่ใจว่าแต่ละเกมจะแตกต่างกัน

- **main**

ฟังก์ชัน `main` เป็นฟังก์ชันหลักของโปรแกรม ฟังก์ชันนี้จะเรียกใช้ฟังก์ชันอื่นๆ และควบคุมลำดับของเกม

ฟังก์ชันต่อไปนี้เหมือนกันกับการเล่นของจริง:

- `shuffle`
- `random`
- `score`
- `display`
- `pop_card`
- `add_card`

ฟังก์ชันเหล่านี้เป็นฟังก์ชันที่จำเป็นในการจำลองเกมแบล็คแจ็ค ฟังก์ชัน `shuffle` ทำหน้าที่สับไพ่ในสำรับไพ่ ฟังก์ชัน `random` ทำหน้าที่สร้างตัวเลขสุ่ม ฟังก์ชัน `score` ทำหน้าที่คำนวณแต้มของผู้เล่นหรือเจ้ามือ ฟังก์ชัน `display` ทำหน้าที่แสดงไพ่ของผู้เล่นและเจ้ามือ ฟังก์ชัน `pop_card` ทำหน้าที่ดึงไพ่ออกจากสำรับไพ่ และฟังก์ชัน `add_card` ทำหน้าที่เพิ่มไพ่ให้กับมือของผู้เล่นหรือเจ้ามือ

วิธีการเล่นของแบล็คแจ็ค

1. เริ่มต้นด้วยการป้อนชื่อของคุณ

```
printf("Enter your name: ");  
scanf("%s", name);
```

2. เจ้ามือจะทำการแจกไพ่สองใบให้ผู้เล่นและไพ่หนึ่งใบให้ตัวเอง โดยไพ่ของผู้เล่นจะหงายขึ้น ส่วนไพ่ของเจ้ามือจะหงายลง

```
for (int i = 0; i < 2; i++) {  
    player_deck[i] = pop_card(deck);  
}  
  
computer_deck[0] = pop_card(deck);
```

3. ผู้เล่นสามารถเลือกที่จะตีไพ่อีกใบหรือหยุดได้ ไพ่จะถูกตีจนกว่าผู้เล่นจะหยุดหรือแต้มเกิน 21

```
while (true) {  
    printf("Hit (H) or stand (S)? ");  
    char input;  
    scanf(" %c", &input);  
  
    if (input == 'H') {  
        player_deck[player_size] = pop_card(deck);  
    } else if (input == 'S') {  
        break;  
    }  
}
```

4. เมื่อผู้เล่นหยุด เจ้ามือจะทำการตีไพ่จนกว่าแต้มจะเท่ากับหรือมากกว่า 17

```
while (score(computer_deck, computer_size) < 17) {  
    computer_deck[computer_size] = pop_card(deck);  
}
```

5. ผู้เล่นที่มีแต้มสูงกว่าจะเป็นผู้ชนะ

```
if (score(player_deck, player_size) > score(computer_deck, computer_size)) {  
    printf("You win! %d vs %d\n", score(player_deck, player_size), score(computer_deck, computer_size));  
} else if (score(player_deck, player_size) < score(computer_deck, computer_size)) {  
    printf("You lose! %d vs %d\n", score(player_deck, player_size), score(computer_deck, computer_size));  
} else {  
    printf("Tie! %d vs %d\n", score(player_deck, player_size), score(computer_deck, computer_size));  
}
```

คำอธิบายเพิ่มเติม

ฟังก์ชัน score ใช้เพื่อคำนวณแต้มของผู้เล่นหรือเจ้ามือ ฟังก์ชันนี้จะวนซ้ำจำนวนครั้งเท่ากับจำนวนไพ่ในมือของผู้เล่นหรือเจ้ามือ และในแต่ละรอบจะเพิ่มค่าของไพ่นั้นลงในแต้มรวม

ไพ่แต่ละใบมีคะแนนดังนี้

- ไพ่ตัวเลขมีค่าเท่ากับตัวเลขนั้น (2-9)
- ไพ่หน้ามีค่าเท่ากับ 10
- ไพ่เอซมีค่าเท่ากับ 1 หรือ 11

โดยที่

- หากผู้เล่นแต้มเกิน 21 ผู้เล่นจะแพ้ทันที
- หากผู้เล่นแต้มเท่ากับ 21 ผู้เล่นจะชนะทันที
- หากผู้เล่นแตมน้อยกว่าเจ้ามือ ผู้เล่นจะแพ้
- หากผู้เล่นแต้มมากกว่าเจ้ามือ ผู้เล่นจะชนะ
- หากผู้เล่นแต้มเท่ากับเจ้ามือ เกมจะเสมอ

ตัวอย่างการเล่น

สมมติว่าชื่อของคุณคือ "John"

```
Enter your name: John

John's hand:
A-\3 (11)
2-\3 (2)

Computer's hand:
J-\3 (10)

Hit (H) or stand (S)? H

John's hand:
A-\3 (11)
2-\3 (2)
Q-\3 (12)

Hit (H) or stand (S)? H

John's hand:
A-\3 (11)
2-\3 (2)
Q-\3 (12)
7-\3 (7)

Hit (H) or stand (S)? S

Computer's hand:
J-\3 (10)
A-\4 (11)

You lose! 20 vs 21
```

ในตัวอย่างนี้ John แต้มเกิน 21 จึงแพ้

สรุปผลการทำงาน Term Project