2022 2학기

프로그래밍과 문제해결

Assignment #1

담당교수: 윤은영

학과: 무은재학부

학번: 20220778

이름: 표승현

POVIS ID: hyeony312

“명예서약”

“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”

Problem: 인디언 홀덤

1. 프로그램 기능 개요

- Structure chart

Print card, chips, winner

Get select, betting

Find winner

Calculate chips

Calculate hand

Select cards

Calculate

Game (assn1)

- 입력: 1. 유저의 행동값을 입력받는다.

2. Raise를 선택했을 시 얼마를 베팅할 것인지 입력받는다.

- 처리: 1. 랜덤함수를 이용하여 각 라운드마다 4장의 무작위 카드를 생성한다.

2. 컴퓨터와 유저의 hand를 각각 계산한다.

3. Hand와 플레이어의 행동값을 계산하여 winner를 판단한다.

4. 베팅 결과 플레이어의 남은 칩 수를 계산한다.

- 출력: 1. 게임 시작 시 카드를 출력한다.

2. 게임 종료 시 남은 칩과 승자를 출력한다.

2. 알고리즘

-Psedocode

1 while(1) {

2 card\_shuffle()을 이용해 유저, 컴퓨터, 공유1, 공유2 카드를 생성한다.

3 calc\_hand()함수를 이용하여 유저와 컴퓨터의 hand를 각각 판별한다.

4 do{ // 유저가 선인 경우 turn값은 0, 컴퓨터가 선인 경우에는 1의 값으로 시작한다.

5 user\_turn()을 이용하여 행동값, 베팅할 칩 수를 입력 받아 user\_betting\_chips에 반영한다. //turn이 짝수일 때만 실행

6 com\_turn()을 이용하여 유저의 hand에 따라 행동값, 베팅할 칩 수를 계산하고 com\_betting\_chips에 반영한다. //turn이 홀수일 때만 실행

7 turn++

8 } while (플레이어 중 하나가 Call 또는 Fold를 선택할 때까지)

9 calc\_winner()를 이용해 hand를 비교하여 승자를 판별한다.

10 승자와 각 플레이어의 남은 chips를 출력한다.

11 게임 계속 진행 여부를 묻는다.

12 if (유저가 게임 종료 원함 or 10번째 라운드가 종료 or 플레이어의 칩이 바닥난 경우){

13 최종 결과를 출력하고 break;}

14 else{

15 다음 게임을 진행한다.}

16 }

유저가 게임 진행을 선택, 10 미만의 라운드 진행, 양 측 플레이어의 칩이 남았는가?

turn값에 따라 user\_turn, com\_turn중 하나 실행

Calc\_hand를 이용해 플레이어의 조합 판별

card\_shuffle을 이용해 카드 생성

베팅이 종료되었는가?

(call이나 fold가 선택되었는가?)

게임 진행 여부 질문

승자, 각 플레이어의 남은 칩 출력

Calc\_winner를 이용해 승자 판별

No

Yes

3. 프로그램 구조 및 설명

Yes

최종 결과 출력 및 프로그램 종료

No

a) 게임 세팅

-while(1)을 이용해 break를 만나기 전까지 아래 내용들을 계속 반복한다.

-card\_shuffle(&shared\_card1, &shared\_card2, &user\_card, &computer\_card): 1에서 10 사이의 난수를 생성하는 함수를 이용하여 매 라운드 무작위 4장의 카드를 생성한다.

-print\_card\_info(shared\_card1, shared\_card2, user\_card, computer\_card): 유저의 카드를 제외한 카드를 시각적으로 출력한다.

-calc\_hand(card, shard\_card1, shard\_card2): for문을 이용하여 카드를 숫자 크기 순으로 나열한 수 해당되는 hand 조합이 있는지 조사한다.

b) 베팅

-is\_valid\_num(a, b, num): a와 b 사이의 숫자라는 범위를 이용하여, 유저에게 입력 받은 입력값이 유효한지 판단한다.

-user\_turn(user\_chips, &user\_betting\_chips, betted\_chips, turn): turn값이 짝수일 때 실행한다. 상대가 베팅한 칩 수와 보유한 칩 수를 전달받고, 유저의 입력에 따라 베팅할 칩 수를 수정한다.

-computer\_turn(user\_hand, com\_chips, &com\_betting\_chips, betted\_chips, turn): turn값이 홀수일 때 실행한다. 상대가 베팅한 칩 수와 보유한 칩 수를 전달받고, 유저의 hand에 따라 확률적으로 다음에 할 행동을 결정한다.

-turn값은 user\_turn 또는 computer\_turn이 실행될 때마다 1씩 증가한다.

-플레이어 중 한 쪽이 call이나 fold를 선택할 때까지 반복한다.

c) 베팅 종료

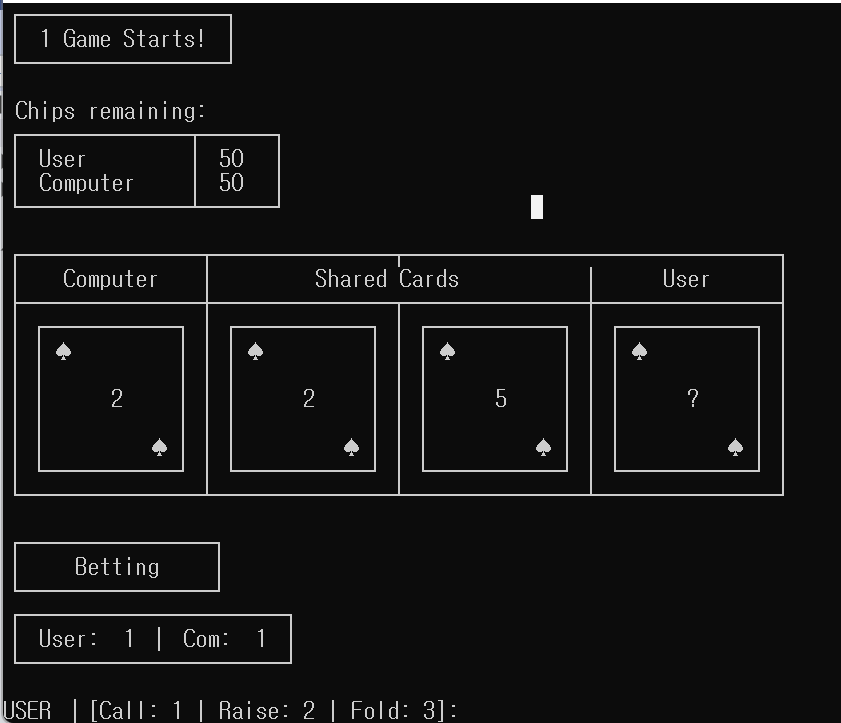
-calc\_winner(shared\_card1, shared\_card2, user\_card, computer\_card): 플레이어의 hand를 비교하고, 필요에 따라 개인 카드의 숫자를 비교하여 각 라운드의 승자를 결정한다.

-게임을 계속 진행할 것인지 묻는다. 계속 할 것이라면 1을, 종료를 원한다면 -1을 입력 받는다.

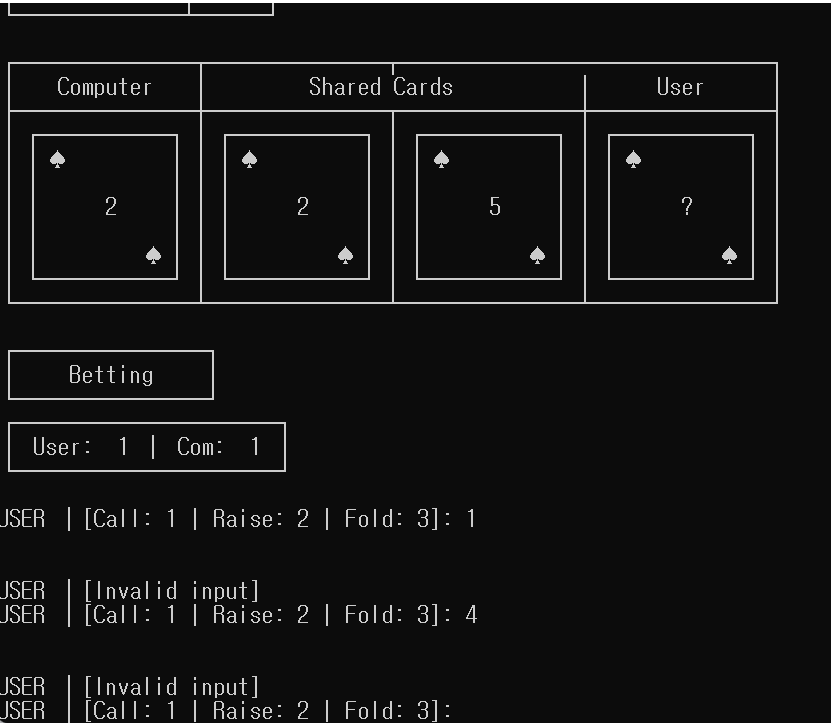
-유저가 -1을 입력하거나, 10라운드 게임이 끝났거나, 한 측의 칩이 모두 소진되었다면 최종 스코어와 승자를 출력하고 break로 프로그램을 종료한다.

4. 프로그램 실행방법 및 예제

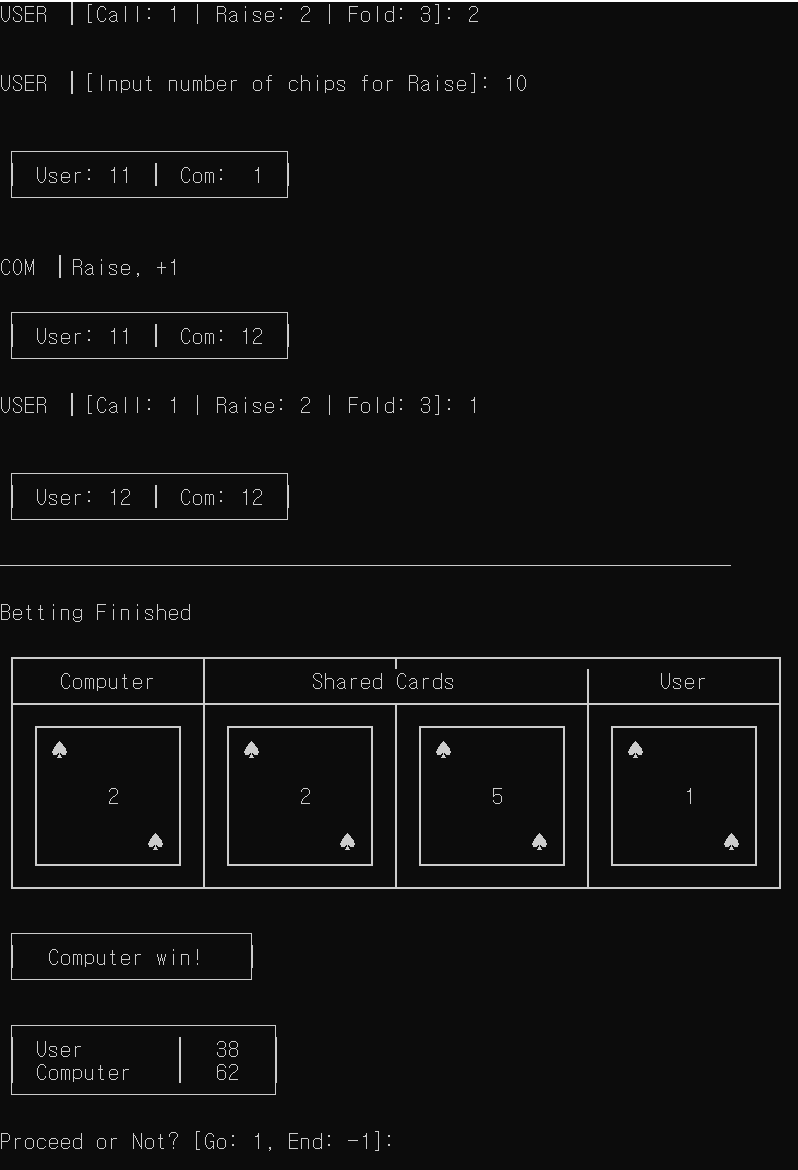
-visual studio에서 assn1을 컴파일한다.



시작 화면이다. 라운드 수와 남은 칩 수, 생성된 카드패를 출력한다. 첫 라운드는 유저가 먼저 베팅을 시작한다.



첫 베팅에서 call을 선택한 경우, 1,2,3 외의 숫자를 선택한 경우에는 유효하지 않은 입력이라는 문구를 출력하고 다시 입력값을 받는다.



Raise를 선택하면 베팅할 칩 수를 입력 받는다. 이후 컴퓨터와 번갈아가며 베팅이 종료될 때까지 베팅을 진행한다. 베팅이 끝났다면 모든 카드를 공개하고 결과를 출력한다.



프로그램 종료 조건 중 하나이다. 10번째 라운드가 종료되어 최종 결과를 출력하고 프로그램이 종료된다.

5. 토론

- hand를 판별하는 함수를 구현할 때 정돈되지 않는 일련의 숫자를 바로 분석하고자 하니 어려움이 있었다. 때문에 for문을 이용하여 카드를 크기 순으로 나열하는 과정을 거쳤더니 간단한 대소비교를 통해 hand를 쉽게 구분할 수 있었다.

- user\_turn과 com\_turn이 번갈아가면서 실행되도록 하기 위해 turn이라는 정수형 변수를 선언하고, 이 값이 짝수인지 홀수인지에 따라 둘 중 하나를 수행하도록 하였다. 이를 이용하여 전 판의 승자가 누구인지에 따른 선 결정도 구현할 수 있었다.

6. 결론

이번 과제에서 포인터의 기본적인 기능을 직접 활용해보며 익힐 수 있었다. 특히 포인터 변수, 주소, 역참조의 개념을 보다 정확하게 이해하게 되고, 포인터를 활용함으로써 얻을 수 있는 이점이 무엇인지 고민하고 알아볼 수 있었다.

7. 개선방향

원작 게임에서는 똑같이 총 40장의 카드를 이용하지만 각 라운드에서 사용된 카드는 버려진다는 차이가 있다. 중복된 카드를 제외하고 카드를 생성하는 기능을 추가한다면 카드카운팅을 필요로 하는 원작 게임에서의 재미도 느낄 수 있을 것이다.

또한 유저와 컴퓨터의 턴에서 call, raise, fold를 함에 따라 베팅 칩을 수정하고 이를 출력하는 기능을 구현하였는데, 이 과정에서 비슷한 형태의 코드가 많이 반복되었다는 느낌이 들었다. 때문에 각각의 행동을 사용자정의함수로 선언하여 사용한다면, 코드의 가독성을 높이고 복잡함도 줄일 수 있을 것이다.