**2021년 2학기**

**프로그래밍과 문제해결**

**Assignment #1**

**담당교수: 윤은영**

**학번: 20217016**

**학과: 무은재학부**

**이름: 최대현**

**POVIS ID: daehyeonchoi**

**명예서약(Honor Code)**

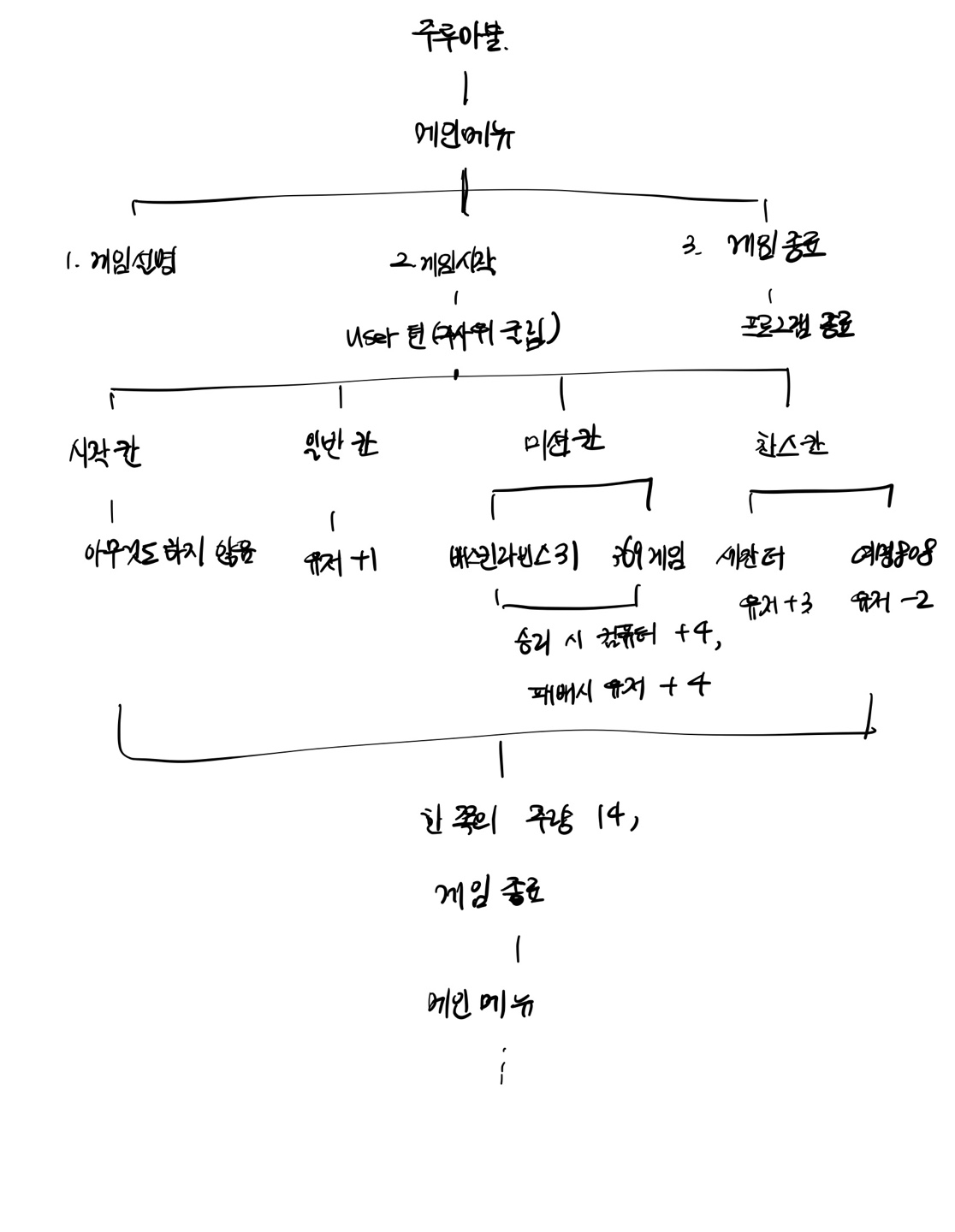
**“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”**

**프로그램 실행 방법**

(제작할 때 사용했던 Dev-C++ 환경 기준으로 설명) assn1.c 파일을 다운로드하고 실행한 뒤 F11키를 누르면 프로그램을 실행할 수 있다.

**Structure Chart**

이번 주루마불 프로그램의 대략적인 구조도는 다음과 같다.



**알고리즘**

* **Psuedocode**

1. 메인 메뉴 출력, 1~3까지 입력받는다.
   1. 1번을 선택하면 게임의 규칙이 출력된다.
   2. 2번을 선택하면 게임이 시작된다.
   3. 3번을 선택하면 프로그램이 종료된다.
2. <게임 시작> 시작 칸에서 출발하고, 유저가 주사위를 굴린다. 이 때 -1을 입력하면 1로 돌아간다.

주사위를 굴리면 맵에 user의 위치가 표시된다.

Play\_juru 함수에 저장된 내용대로 칸에 도착할 때마다 그 칸에 맞는 행동들(1잔 마시기, 미션, 찬스, 휴식)을 수행한다.

미션의 경우 2가지 게임 중 랜덤으로 실행되고, 진 사람이 4잔을 마신다.

시작과 휴식 칸에서는 아무것도 하지 않은 채 턴이 넘어간다.

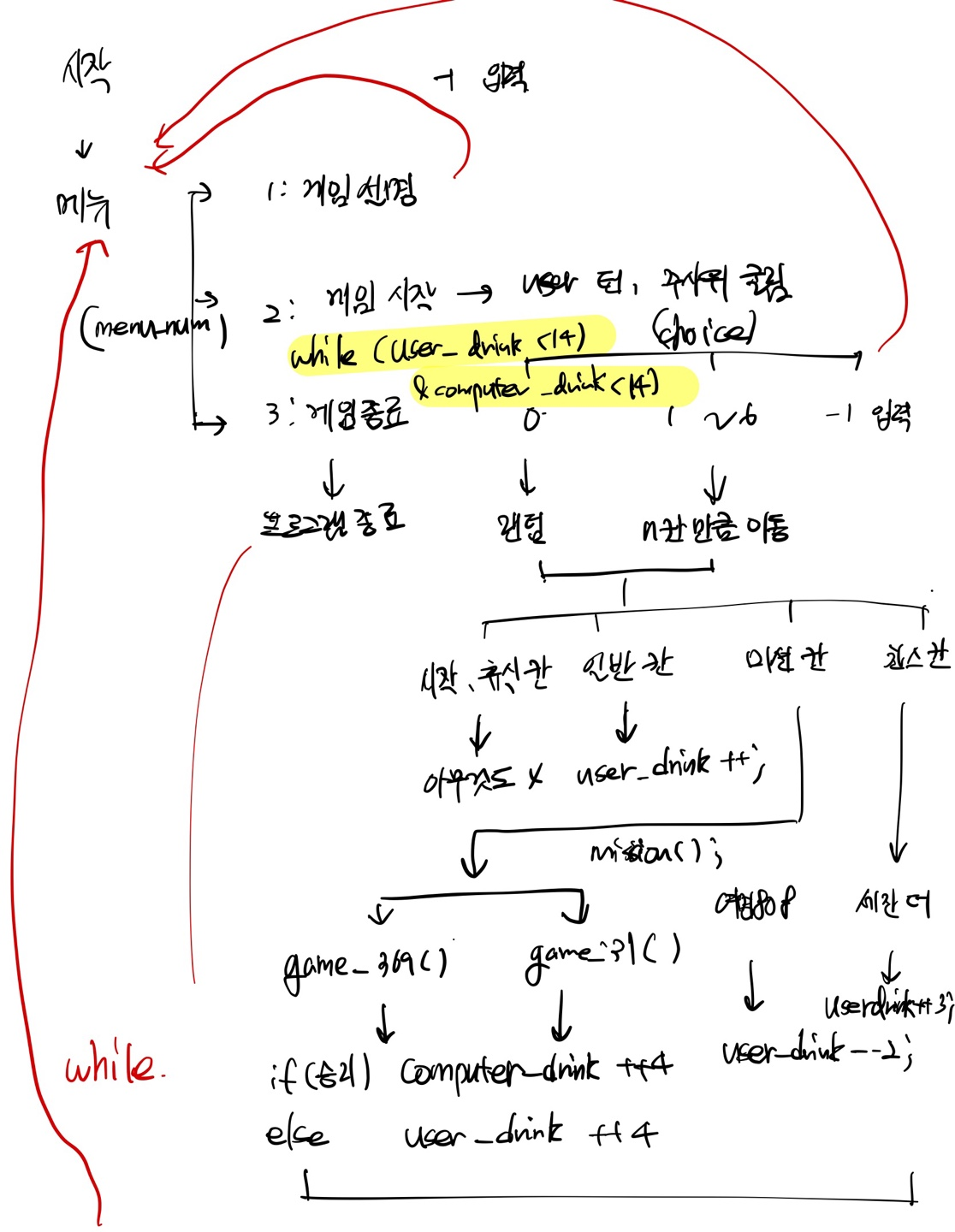
컴퓨터는 미션을 제외하고 매 차례 1잔을 마신다.

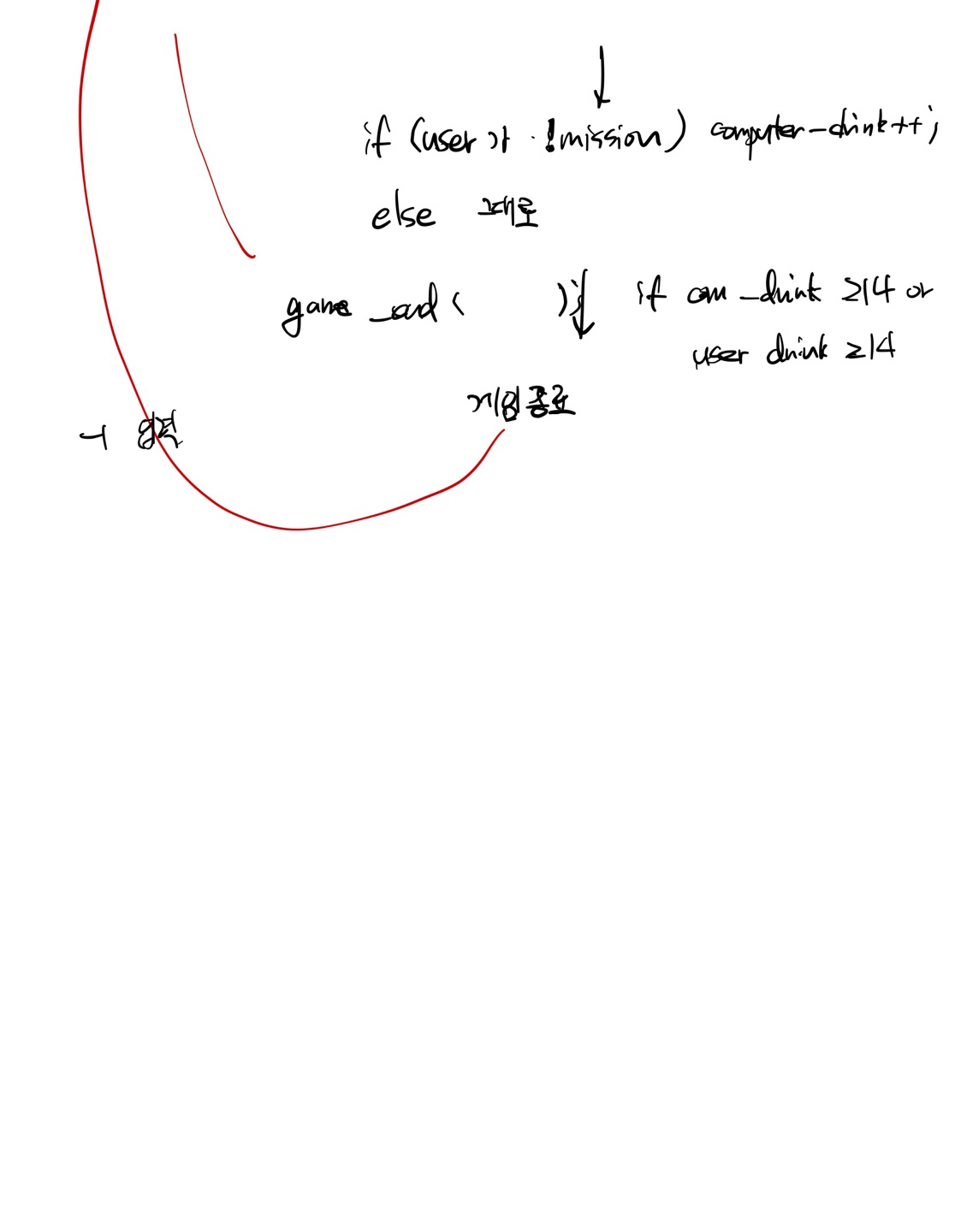
컴퓨터의 턴이 끝날 때 마다 추가된 주량을 반영한 현재 주량이 표시된다.

1. User나 컴퓨터 둘 중 한 명의 주량이 14잔이 되는 순간 게임이 끝난다. 이 때 -1을 입력하면 1로 돌아간다.

* **Flowchart**

이번 주루마불 프로그램의 대략적인 flowchart는 다음과 같다.





**함수 설명**

1. **int print\_menu()**

* Menu를 출력해주는 함수이다. 정수를 입력받아서 다른 함수로 넘겨주기 위해 int형 함수로 제작하였다.

1. **int rescan\_menu()**

* 받아온 메뉴 번호 값이 1~3 범위가 아닐 때 메뉴 번호를 다시 입력받기 위해서 만든 함수이다. 다시 입력받았는데도 범위 내의 숫자가 아닌 경우를 고려하여 함수 내에 while문을 넣었다.

1. **Int game\_info()**

* 메인함수에서 print\_menu()함수를 호출한 뒤 메뉴에서 1번을 선택할 경우 호출되는 함수로, 게임의 규칙을 설명해준다. -1을 입력했을 때 메인메뉴로 돌아가도록 프로그램을 설계해야 하기 때문에 int형으로 함수를 제작하였다.

1. **int game\_start()**

* 메인함수에서 print\_menu()함수를 호출한 뒤 메뉴에서 2번을 선택할 경우 호출되는 함수로, 게임을 시작하는 함수이다. 가장 복잡한 함수로, 안에 print\_map, print\_drink, play\_juru 등의 다양한 함수들을 호출하도록 설계되어 있다.

1. **void print\_drink(int user\_pos, int user\_count, int comp\_count) 함수**

* game\_start 함수에서 호출되어 user의 위치와 유저의 주량, 컴퓨터의 주량을 출력해주는 함수로, 이런 정보들을 출력해야 하기 때문에 3개의 매개변수를 받고 있다.

1. **void print\_map(int user\_pos)**

* 맵을 출력하는 함수이다. 맵을 출력하기 위해서는 user의 위치를 알아야 하기 때문에 user\_pos 변수를 매개변수로 가져왔고, 함수 내부에서 어싸인 공지에 포함되어있는 set\_position 함수를 0번 자리부터 11번 자리까지 호출하였다.

1. **char set\_position(int user\_pos, int pos)**

* 어싸인 공지 그대로의 내용을 구현한 간단한 함수이다. 스페이스바 또는 ‘0’을 리턴해야 하기 때문에 char형으로 구현하였다.

1. **int game\_369()**

* 369 게임의 기능을 구현한 함수이다. 턴을 먼저 시작하는 것을 랜덤으로 하기 위해 seed가 time(NULL)인 rand값을 2로 나눈 나머지를 사용하였다. 0일 때는 컴퓨터가 턴을 시작하고, 1일 때는 user가 턴을 시작한다. cnt라는 변수가 핵심인데, 21까지 정상적으로 출력하는 것이 user가 이기는 것이기 때문에 cnt 변수가 21 이하일 조건으로 while문을 돌리되 user가 잘못된 수를 입력하는 경우 cnt값을 23으로 만들어서 while 문을 깨고, cnt값이 23인 경우에는 user가 패배한 경우로 처리하였다. Int형 함수로 제작하여 승리 시 1을 반환하고 패배 시 0을 반환했다.

1. **int game\_31()**

* 배스킨라빈스 31 게임의 기능을 구현한 함수이다. 같은 방법으로 턴 시작을 랜덤으로 정하였고, while문을 사용하여 user가 입력하는 값이 1~3 범위가 아닌 동안 계속하여 입력받도록 설계하였다. whos\_turn이라는 변수를 사용하여 짝수인 경우, 홀수인 경우로 나눠서 user와 컴퓨터의 턴을 구분하였다. 369 게임과 마찬가지로 cnt라는 변수를 설정했는데, 이것은 현재 게임이 몇까지 진행되었는지를 표현하는 변수이다. cnt값이 32가 되는 순간 while문을 break하도록 설계하였다.

1. **int play\_juru(int user\_pos)**

* 포인터를 사용하지 않고 구현하기 위해 void로 주어졌던 함수의 형을 int로 변환해서 사용하였다. 어느 칸에 있는지에 따라 다른 기능을 구현해야 하므로 user\_pos를 가져와서 switch-case문으로 user\_pos에 따라, 즉 현재 유저의 위치에 따라 다른 case들을 구현하였다. 각각의 케이스들에 return값을 넣어줌에 따라 game\_start 함수에서 play\_juru함수의 리턴값에 따라 각각 주량을 다르게 넣어주도록 설계하였다.

1. **Int mission()**

* user가 미션 칸에 도착할 경우 369게임이나 배스킨라빈스 게임 중에 하나를 랜덤으로 호출하는 함수이다. 이 함수 내에서 직접 game\_31()과 game\_369()를 호출한다.

1. **int game\_end(int user\_pos, int user\_count, int comp\_count)**

* game\_start() 함수에서 한 쪽 주량이 14 이상이 되면 바로 호출되는 함수로, 이 함수 내에서 주량이 14를 초과할 수 없기 때문에 주량을 14로 만들어준다. 그리고 user가 승리했을 경우, computer가 승리했을 경우 각각 다른 게임 종료 안내 메시지를 호출하고, -1을 입력받으면 -1을 리턴하며 함수를 종료한다.

**프로그램 실행 방법 및 예제**

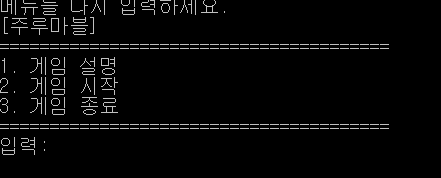
1. **초기 선택 메뉴**

초기 선택 메뉴에서 1,2,3번을 선택할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

1,2,3번이 아닌 다른 수를 입력하면 다시 입력하라는 메시지와 함께 다시 입력받는다.



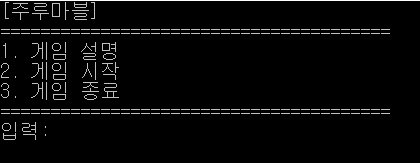
1. **게임 설명**

메뉴에서 1번을 누르면 화면이 지워지고 메뉴를 설명하는 창이 띄워지게 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**-**-1을 입력할 경우, 메인 화면으로 돌아간다.



1. **게임 종료**

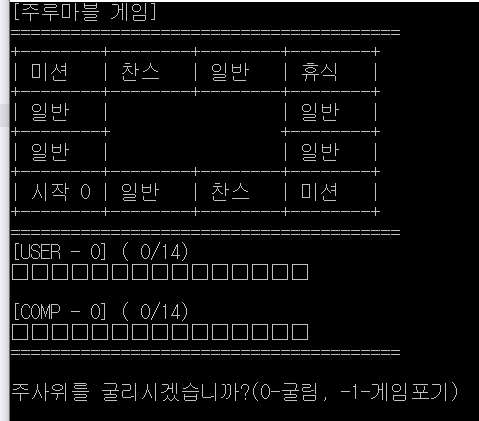
* 메뉴에서 3번을 누르면 화면이 지워지고 게임을 종료하는 메시지와 함께 프로그램이 종료된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

1. **게임 시작**

* 메뉴에서 2번을 누르면 화면이 지워지고 게임이 시작된다.



- 가장 첫 화면으로, user의 위치가 0이고 시작 칸에 위치한다. 메시지와 함께 -1~6 범위 내의 수를 입력받으며, 다른 수를 입력하면 다시 입력받는다. –

- -1을 누르면 메인화면으로 돌아간다.



- 0을 누르면 주사위 값이 1~6 중 랜덤으로 선택되고, 라운드값이 1 늘어나며 출력된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

- 0을 입력했을 때 주사위가 4로 설정된 경우이다. 컴퓨터는 주사위를 굴리지 않고 미션 칸을 제외하고는 항상 1잔씩 마신다. 그 뒤 다시 주사위값을 입력받는다.

이번에는 2를 입력하여 휴식칸에 도달해보았다.

텍스트이(가) 표시된 사진

자동 생성된 설명

- 휴식 칸에 도착하여 아무것도 하지 않고 컴퓨터에게 턴이 넘어간다.

이번에는 3을 입력하여 미션 칸에 도달해보았다.

텍스트이(가) 표시된 사진

자동 생성된 설명

- 두 가지 게임 중 배스킨라빈스 게임이 선택되었고, 이번에는 컴퓨터의 선제 턴으로 게임이 이뤄진다.

텍스트이(가) 표시된 사진

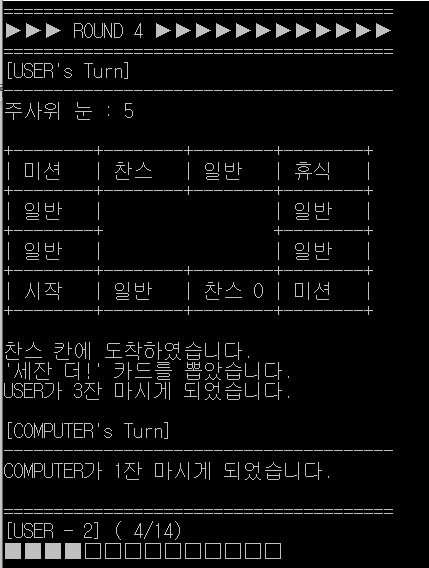
자동 생성된 설명

마지막에 1을 입력하여 user가 승리하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

미션 칸에 도달하면 컴퓨터는 자신의 턴을 넘긴다는 규칙에 따라 게임 패배에 해당하는 패널티인 4잔만 마신 채 게임이 진행된다.



- 5를 입력하여 찬스 칸에 도달해보았다. 두 가지 찬스 중 세 잔 더! 카드가 랜덤으로 선택되었고, user의 주량이 1에서 4로 증가하였다.

- 게임을 계속 진행하던 중 라운드 8에서 369 게임을 진행하였고, 컴퓨터의 주량이 14를 초과하며 게임이 그대로 종료되었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**토론**

play\_juru 함수가 처음에 void형으로 주어져서 void형으로 시도를 해 보았으나, 각 칸마다 무엇을 해야하는지 game\_start 함수에 정보를 전달하는 과정에서 void형으로 함수를 제작하면 포인터를 사용해야 하는 복잡한 문제가 발생해서 int형 함수를 사용하여 간단하게 제작하였다.

**결론**

이번 assignment를 통하여 다양한 반환 형태를 가진 사용자 정의 함수를 제작해볼 수 있었고, 반복문(for, while)과 조건문(if, switch-case)을 사용하는 방법을 익힐 수 있었다. 또한 seed를 time(NULL)로 설정하는 기법을 통해 매초 달라지는 난수를 생성하는 난수 생성 알고리즘을 탐구할 수 있었다.

**개선 방향**

컴퓨터는 항상 한 칸씩 움직임에도 불구하고 항상 컴퓨터의 위치는 0으로 기록되어 있는데, 컴퓨터의 위치도 주사위를 굴려서 표현하고 맵에 추가하여 더 생동감 있는 게임 진행 방식을 고려해보면 좋을 것 같다.