

C++ 프로그래밍 및 실습

술게임 구현하기

진척 보고서 #1

제출일자: 2024 / 11 / 17

제출자명: 박태은

제출자학번: 213182

1. 프로젝트 목표 (16 pt)

1) 배경 및 필요성 (14 pt)

최근 로제의 노래 "APT"가 인기를 끌면서, 아파트 게임 및 다양한 술게임이 해외에 SNS에 유행하고 있습니다. 이러한 유행에 맞춰 친구들과 즐길 수 있는 술게임 프로그램의 필요성을 느꼈습니다. 비대면 또는 대면 모임에서도 손쉽게 할 수 있는 K-술게임을 프로그래밍 함으로써 사람들에게 더 편리하고 재미있는 경험을 제공하고 싶었습니다. 이 프로젝트를 통해 술자리에서의 즐거움을 극대화하고, 술자리 문화를 다양화할 수 있습니다.

2) 프로젝트 목표

이 프로젝트의 목표는 술자리에서 즐길 수 있는 다양한 게임들을 하나의 C++ 프로그램으로 통합해, 사용자들이 편리하게 접근할 수 있도록 하는 것을 목표로 합니다. 아파트 게임(APT)', '369 게임', '베스킨라빈스31 게임' 등 3가지 게임을 제공하며, 각 게임의 룰을 영어로 설명해 국제적으로도 사용할 수 있는 프로그램을 만듭니다. 게임에서의 실패는 '술' 카운트로 누적되며, 주량을 넘길 경우 프로그램이 자동으로 만취 상태를 알리고 종료합니다. 이 프로그램은 참여 인원 수를 설정하고, 각 참여자의 주량을 설정할 수 있어서 사용자들이 쉽게 적응하고 편하게 사용할 수 있도록 한다.

3) 차별점

이 프로그램은 여러 술게임을 하나의 프로그램에서 손쉽게 이용할 수 있도록 제공한다. 또한 APT노래 유행에 따라 해외에서도 사용 가능하도록 영어로 진행되도록 하고, 누구나 간편하게 이해하고 즐길 수 있는 점에서 차별화됩니다. 또한, 각 게임에서 사용자가 자율적으로 참여자 자리 배정 순서와 주량을 설정하고, 규칙을 어긴 참여자의 술 카운트를 자동으로 누적하며, 주량을 초과할 경우 프로그램이 참여자의 만취 상태를 출력하여 게임을 종료합니다. 이러한 기능을 통해 현실감을 더해줘서 게임에 더 몰입하게 할 수 있습니다.

2. 기능 계획

1) 기능 1: 게임 시작 및 선택

- 설명: 사용자가 원하는 술게임을 선택할 수 있는 메뉴 기능

(1) 세부 기능 1: 참여자 대학생 클래스 생성

- 설명: 게임 참여자(대학생으로 가정) 클래스를 생성 student.h, student.cpp, main.cpp로 세가지 파일로 분류

(2) 세부 기능 2: 참여자 대학생 객체 생성

- 설명: 학생 객체를 만들어서 자리배정을 설정해놓고 게임 시작, 각 참여자 주량도 상수 값으로설정.

(3) 세부 기능 3: 게임 리스트 표시 및 설명 출력

- 설명: 사용 가능한 게임 목록(APT 게임, 369 게임, 베스킨라빈스 31 게임)을 처음에 출력해 준다.

(4) 세부 기능 4: 게임 선택 입력 처리

- 설명: 매판이 끝날 때마다 게임에 저서 술을 마신 사람이 선택한 게임에 따라 게임을 시작할 수 있도록 사용자 입력을 받음.

2) 기능 2: 아파트 게임(APT)

- 설명: 특정 숫자를 외치는 순서를 랜덤으로 지정하여 진행하는 게임

(1) 세부 기능 1: 게임 시작 및 숫자 랜덤화

- 설명: 게임 인트로 출력 후, 층 입력 받는다. 각 참여자가 정해진 숫자를 랜덤배정 시키도록 설정

(2) 세부 기능 2: 순서대로 층 출력 후 탈락

- 설명: 순서대로 층 출력하다가 해당 층에 걸린 사람이 탈락시킨다.

(3) 세부 기능 3: 탈락자가 다음 게임 차례로 설정 및 술 카운트 증가

- 설명: 해당 탈락자의 인덱스로 Turn 변수가 바뀌게 만들고, 술 카운트를 증가시킨다.

3) 기능 3: 369 게임

- 설명: 숫자에 3, 6, 9가 포함된 경우 박수를 치는 게임

(1) 세부 기능 1: 순서 설정

- 설명: 각 자리에서 오른쪽, 혹은 왼쪽으로 순서를 진행할지 선정

(2) 세부 기능 2: 숫자 입력 및 규칙 판정

- 설명: 3, 6, 9가 포함된 숫자일 때 박수를 치도록하고, 잘못된 숫자나, 박수를 치지 않은 경우 탈락하도록 설정.

(3) 세부 기능 3: 실패 시 술 누적

- 설명: 탈락한 참여자의 '술' 카운트를 증가시킴.

4) 기능 4: 베스킨라빈스 31게임

- 설명: 참가자들이 1~3개의 숫자를 순서대로 부르다가 31을 외치게 되는 사람이 탈락하게 되는 게임

(1) 세부 기능 1: 순서 설정

- 설명: 각 자리에서 오른쪽, 혹은 왼쪽으로 순서를 진행할지 선정

(2) 세부 기능 2: 게임 규칙에 따른 진행

- 설명: 각 참여자가 1~3개의 숫자를 순서대로 부르다가 31을 외치게 되면 탈락하게 설정.

(3) 세부 기능 3: 실패 시 술 누적

- 설명: 규칙에 어긋난 숫자를 외치거나 31을 외친 사람, 해당 참여자의 '술' 카운트를 증

가시킴.

5) 기능 5: 술 누적 시스템

- 설명: 각 게임에서의 실패로 인해 누적된 술의 양을 기록하고, 게임 종료시킴.

(1) 세부 기능 1: 술카운트 기록

- 설명: 각 참여자 별로 술이 누적되는 양을 기록할 변수 생성.

(2) 세부 기능 2: 누적 결과 표시

- 설명: 각 게임 종료 후, 해당 게임에서의 누적된 술 양을 보여줌.

(3) 세부 기능 3: 종료 및 전체 결과 표시

- 설명: 1명이 주량(설정된 상수값)을 넘길 경우 만취했다는 문구를 출력하고 게임을 종료함. 그 후 전체 얼마나 많이 마셨는지 결과 표시

3. 진척사항

1) 게임 시작 및 선택

(1) 참여자 대학생 클래스 생성

- 입출력

생성자 Student() : string name, int tolerance

Set_apNum()함수 : int n1, n2

- 설명 **student.h와 student.cpp, main.cpp로 구분.**

1. student.h : Student 클래스 선언부

2. student.cpp: Student 클래스 내 함수

멤버변수로는 이름, 주량, 술잔 개수, APT게임에서 배정받을 숫자 vector 4가지

멤버함수로는 생성자(파라미터X), 생성자(이름, 주량 파라미터), 이름출력 함수, 주량 출력 함수, 한 잔씩 마시게 하는 함수, APT 번호 설정 함수, 6가지

Int Tolerance(주량) 변수를 const로 설정

vector<int> aptNum 벡터로 aptNum을 받도록 멤버변수 설정

Student(string name, int tolerance)생성자 만들 때 상수 존재해서 초기화 시켜준다.

String getName()함수는 name을 리턴

int getTolerance()함수는 glasses을 리턴

void inc_glasses()함수는 glasses+=1하고, 1잔 마시고, 현재 얼마나 마셨는지 출력

void set_apNum(int n1, int n2)는 vector인 aptNum에 push_back(n1,n2)해준다.

Vector<int> getAptNum()함수는 벡터인 aptNum을 리턴

- 적용된 배운 내용 : 6-7주차 함수와 매개변수, 9주차 클래스, 10주차 생성자, Vector

- 코드 스크린샷(student.h)

```
src > C student.h > Student
1  #include <iostream>
2  #include <string>
3  #include <vector>
4  using namespace std;
5
6  class Student{
7  private:
8      // 이름
9      string name;
10     // 주량
11     const int tolerance;
12     // 마신 술잔 개수
13     int glasses;
14     // APT게임에서 배정받을 숫자vector
15     vector<int> aptNum;
16
17 public:
18     Student();
19     Student(string name, int tolerance);
20     string getName();
21     int getTolerance();
22     void inc_glasses();
23
24     void set_aptNum(int n1, int n2);
25     vector<int> getAptNum();
26
27 };
```

코드 스크린샷(student.cpp)

```
src > student.cpp X test.txt main.cpp tasks.json student.h
1  #include "student.h"
2
3  // 생성자 : 파라미터 없을 때, 주량이 상수라 초기화 시켜줘야한다.
4  Student::Student()
5  |   // 초기화
6  |   : name(""), glasses(0), tolerance(0) {}
7
8  // 생성자 : 파라미터 없을 때, 주량이 상수라 초기화 시켜줘야한다.
9  Student::Student(string name, int tolerance)
10 |   // 초기화
11 |   : name(name), glasses(0), tolerance(tolerance) {}
12
13 // 이름 출력하는 함수
14 string Student::getName(){
15 |   return name;
16 | }
17 // 주량 출력하는 함수
18 int Student::getTolerance(){
19 |   return glasses;
20 | }
21 // 한잔씩 마시게 하고, 현재까지 마신 잔 수 출력.
22 void Student::inc_glasses(){
23 |   glasses+=1;
24 |   cout << "have a shot" << endl;
25 |   cout << name << " has drunk " << glasses << " glasses\n" << endl;
26 | }
27
28 void Student::set_apNum(int n1, int n2) {
29 |   aptNum.push_back(n1);
30 |   aptNum.push_back(n2);
31 | }
32
33 vector<int> Student::getAptNum() {
34 |   return aptNum;
35 | }
```


(2) 참여자 대학생 객체 생성

- 입출력 : 없음

- 설명

students 벡터를 Student클래스 타입으로 생성해서 참여자들 6명 넣었다. 이제 students 인덱스에 따라 자리 배정.

Name, tolerance로 이름과 주량을 각각 입력받는다.

- 적용된 배운 내용 : , 9주차 클래스, 10주차 생성자, 10주차 Vector

- 코드 스크린샷

```
15  int main(){
16      // 벡터로 학생 객체를 저장
17      // 벡터 인덱스를 자리 배정으로 선정
18      vector<Student> students={
19          Student("Jenny",3), // 제니
20          Student("Lisa",5), // 리사
21          Student("Jisoo",7), // 지수
22          Student("Rose",2), // 로제
23          Student("Bruno",3), // 브루노 마스
24          Student("Taeun",4) // 태은
25      };
// 학생 객체 생성 및 벡터에 저장
```

(3) 게임 리스트 표시 및 설명 출력

- 출력 : 게임 목록 및 설명

- 설명 : 게임 시작할 때 game select문구와 함께 이 게임에서 사용 가능한 APT 게임, 369게임, 베스킨라빈스31게임 3가지를 모두 영어로 출력해서 설명서를 읽고 게임을 시작할 수 있도록 해준다.

- 적용된 배운 내용 : 2주차 입출력

- 코드 스크린샷

```
// 게임 설명 출력
cout << "\n<Game Select>" << endl;
// APT게임 설명 출력
cout << "1. APT game: In this game, everyone placing both hands on top of each other simultaneously,\n "
    "and the host announces a floor number. As players count up from the bottom, they remove \n "
    "their hands one at a time and place them on top. The player whose hand lands on the announced \n "
    "floor number is the \"winner\" (or chosen player) for that round.\n" << endl;
// 369게임 설명 출력
cout << "2. 369 game: In this game involves counting up from 1, and whenever a number contains the digits 3, 6, or 9\n "
    "players must clap instead of saying the number.\n "
    "For example:\n "
    "Number 13: Clap once\n "
    "Number 33: Clap twice\n " << endl;
// BR21게임 설명 출력
cout << "3. BR21 game: In this game, participants take turns and count sequentially from 1 to 31. On each turn, \n "
    "a player can call out between 1 to 3 consecutive numbers. The player who has to say 31 loses the game.\n" << endl;
```

(4) 게임 선택 입력 처리

- 입력 : string choice,
- 출력 : OO가 좋아하는 랜덤 게임! 무슨 게임! 게임 스타트!, OO의 게임 선택(영어로), APT, 369, BR21 외의 입력을 하면 병신샷~ 병신샷~ 다시!" (영어로) 출력
- 설명

게임에 진사람이 게임을 선택하도록 다시 게임을 시작하도록 반복문 설정.

Int Turn변수 설정해서 처음은 students벡터 0번 인덱스 사람부터 시작하도록 한다.

While문 안에서 매번 게임 선택 문구 출력

if문으로 choice가 APT일경우 APT()함수 실행,

369일 경우 Three_Six_Nine()함수 실행

BR21일 경우 Baskin_Robbins_31()함수 실행

그 밖의 다른 게임을 부를 시 "Fool shot! Fool shot! One more time!"출력 후

해당 Turn의 학생 inc_glasses()로 술 마시게 하고 다시 게임 반복

- 적용된 배운 내용 : 4주차 if조건문, 5주차 while반복문, 9주차 클래스, 10주차 벡터

- 코드 스크린샷

```

43     int Turn = 0;
44     bool game_Running=true;
45     while(game_Running){
46         string choice;
47         // oo가 좋아하는 랜덤 게임! 무슨 게임! 게임 스타트!
48         cout << students[Turn].getName() << "'s favorite random game! Which game! Game start!" << endl;
49         // oo의 게임 선택
50         cout << students[Turn].getName() << "'s choice: ";
51         cin >> choice;
52
53         if (choice == "APT") {
54             APT(students, Turn); // 아파트 게임 시작
55         } else if (choice == "369") {
56             Three_Six_Nine(); // 369게임 시작
57         } else if (choice == "BR31") {
58             Baskin_Robbins_31(); // 베스킨 라빈스 시작
59         } else {
60             // "병신샷~ 병신샷~ 다시!"라고 하면서 말할수 한 사람이 술 한잔 마시고 다시 게임 골라서 시작.
61             cout << "Fool shot! Fool shot! One more time!" << endl;
62             students[Turn].inc_glasses();
63             continue;
64         }
65     }
66
67     return 0;
68 }

```

2) APT 게임 구현

(1) 게임 시작 및 숫자 랜덤화

- 입력 : int floor

- 출력 : APT 인트로("APT~ APT~ APT~ APT~")

층 입력 문구 : ("Which floor?(1 to 20 floors only):")

- 설명 : **APT()함수 정의**

아파트~ 아파트~ 아파트~ 아파트~인트로를 영어로 출력후

Cin >> floor로 걸리는 층 수 설정. 그런데 너무 게임이 오래 걸릴 것을 생각해서 1층부터 20층까지만 받기로 했다. Do while문으로 $1 \leq \text{floor} \leq 20$ 만 받는다.

Vector numbers에 1부터 12까지 배열을 만들고

random라이브러리를 통해 난수 생성 엔진 g를 초기화 한다.

random_device rd;

mt19937 g(rd());

algorithm 라이브러리 함수 shuffle()로 numbers 벡터 숫자들 섞어준다.

for문으로 students벡터의 각 학생들에게 2개씩 입력해준다.

Printfloor_out()함수로 1층부터 차례로 진행시키고 탈락자 Turn으로 설정

Students[Turn].inc_glasses()로 그 해당 Turn 술 마시게 한다.

- 적용된 배운 내용 : 4주차 Do while문, 10주차 vector, algorithm 라이브러리 11주차 난수 생성

- 코드 스크린샷

```
70 void APT(vector<Student>& students, int &Turn){
71     int floor;
72     cout << "APT~ APT~ APT~ APT~" << endl;    // 인트로 : 아파트~ 아파트~ 아파트~ 아파트~
73     do {
74         cout << "Which floor?(1 to 20 floors only): "; // 몇층?(1~20층까지)
75         cin >> floor;
76
77         // 입력값 검증
78         if (floor < 1 || floor > 20) {
79             cout << "Invalid input! Please enter a floor between 1 and 20." << endl;
80         }
81     } while (floor < 1 || floor > 20);
82
83     // 미리 숫자들 생성
84     vector<int> numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
85     // 난수 생성 엔진 초기화
86     // 난수 생성 엔진 객체 g
87     random_device rd;
88     mt19937 g(rd());
89     // algorithm의 shuffle함수로 숫자들 섞어놓는다.
90     shuffle(numbers.begin(), numbers.end(), g);
91
92     // for문과 set_apNum()함수로 숫자 두개 저장
93     // ex) Lisa(2,4), Jenny(1,5) ...
94     for (int i = 0; i < students.size(); ++i) {
95         students[i].set_apNum(numbers[i * 2], numbers[i * 2 + 1]);
96     }
97     printfloor_out(students, floor, Turn);
98     students[Turn].inc_glasses();
99
100
101 }
```

(2) 순서대로 층 출력 후 탈락

- 입력 : `vector<Student>& students, int floor, int & floor`

- 설명 : **Printfloor_out** 함수 정의

`vector<pair<int, string>> floorAssignments;`로 층번호와 이름 넣을 벡터 생성

for문으로 이름, 번호 1개씩 pair로 floorAssignments 벡터에 push한다.

Algorithm 라이브러리의 `sort()`로 층 번호 기준 순서대로 정렬한다.

`int currentFloor = 1`로 설정

while문으로 1층부터 순서대로 출력하도록 설정한다. 그런데 층 인덱스를

`int index = (currentFloor-1)%12`로 해서 12층을 넘어가더라도 1층부터 다시 13층으로 올라가도록 설정한다.

if문으로 목표 층수에 도달하면 탈락처리 후 `find_nameIndex()`함수로 Turn을 그 차례로 바꾼 뒤 종료시킨다.

- 적용된 배운 내용 : 4주차 Do while문, while문, 7단원 함수 정의, 10주차 vector, algorithm 11주차 참조

- 코드 스크린샷

```

103 // 층 순서대로 출력하며 탈락시키는 함수
104 void printfloor_out(vector<Student>& students, int floor, int &Turn) {
105     // 층 번호와 학생 이름을 저장할 벡터
106     vector<pair<int, string>> floorAssignments;
107
108     // 학생들의 aptNum을 기반으로 floorAssignments를 채움
109     for (auto& student : students) {
110         // 그 객체의 aptNum을 순회로 각각 push한다.
111         for (int apt : student.getAptNum()) {
112             floorAssignments.push_back({apt, student.getName()});
113         }
114     }
115     // 층 번호를 기준으로 정렬 (출력을 순서대로 하기 위해)
116     // ex) {(1, "Jenny"), (2, "Jisoo"), (3, "Jenny"), (4, "Rose"), (5, "Taeun")}...}
117     sort(floorAssignments.begin(), floorAssignments.end());
118
119     // 현재 층수 설정
120     int currentFloor = 1;
121     while (true) {
122         // 12층을 초과하더라도 1층에 있던 학생이 13층으로 올라가는 방식
123         int index = (currentFloor - 1) % 12; // 0부터 시작하는 인덱스
124         auto [assignedFloor, name] = floorAssignments[index];
125
126         cout << name << ": " << currentFloor << " floor" << endl;
127
128         // 목표 층수에 도달했으면 탈락 처리 후 종료
129         if (currentFloor == floor) {
130             cout << name << " is out of the game" << endl;
131             Turn = find_nameIndex(students, name);
132             return; // 함수 종료
133         }
134
135         currentFloor++; // 다음 층으로 진행
136     }
137 }

```

(3) 탈락자가 다음 게임 차례로 설정 및 술 카운트 증가

- 입력 : vector<Student>& students, const string& name

- 설명 : find_nameIndex() 함수 정의

for문과 if문으로 name에 해당하는 students[i].getName()을 찾게 되면 return하는 함수를 생성했다.

- 적용된 배운 내용 : 4주차 if문, for문, 7단원 함수 정의, 9주차 클래스 함수, 10주차 vector

- 코드 스크린샷

```

138 // name에 해당하는 students벡터의 인덱스 값 반환하는 함수
139 int find_nameIndex(vector<Student>& students, const string& name) {
140     for (int i = 0; i < students.size(); i++) {
141         if (students[i].getName() == name) {
142             return i; // 찾은 경우 인덱스 반환
143         }
144     }
145     return -1; // 찾지 못한 경우 -1 반환
146 }

```

3) 369 게임 구현

(1) 아직 미구현 출력

```

// 아래 두 게임은 구현 아직 못했다.
void Three_Six_Nine(){
    cout << "Three_Six_Nine game is not yet implemented." << endl;
}

```

4) BR21 게임 구현

(1) 아직 미구현 출력

```

void Baskin_Robbins_31(){
    cout << "Baskin_Robbins_31 game is not yet implemented." << endl;
}

```

5) 술 누적 기능 시스템

(1) 술 카운트 기록

- 입력 : 없음
- 설명 : Student 클래스 내에 마실 술잔 glasses변수 설정

객체마다 각자 glasses변수에 마신 술 양이 누적되게 설정한다.

- 적용된 배운 내용 : 9주차 클래스
- 코드 스크린샷

```
class Student{
private:
    // 이름
    string name;
    // 주량
    const int tolerance;
    // 마신 술잔 개수
    int glasses;
    // APT게임에서 배정받을 숫자vector
    vector<int> aptNum;
```

(2) 누적 결과 표시

- 입출력 : 없음
- 설명 : Student 멤버함수 inc_glasses()를 student.cpp에 정의
함수 실행시 그 객체의 glasses+=1하고

Have a shot 출력 후

현재까지 누적된 glasses양 출력한다.

- 적용된 배운 내용 : 7단원 함수 정의, 9주차 클래스 함수
- 코드 스크린샷

```
// 한잔씩 마시게 하고, 현재까지 마신 잔 수 출력.
void Student::inc_glasses(){
    glasses+=1;
    cout << "have a shot" << endl;
    cout << name << " has drunk "<< glasses << " glasses\n" << endl;
}
```


2) 테스트 결과

(1) 게임 리스트 표시 및 설명 출력

- 설명 : 실행시키면 게임 리스트와 설명이 출력되고

랜덤게임 인트로와 함께 Turn=0에 해당하는 객체의 name에게 게임을 고르라고 출력

- 테스트 결과 스크린샷

```
C:\TaeunPark\CPP2409-P> cmd /C "c:\Users\user\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe --stdin=Microsoft-MIEngine-In-uq0ui5m4.e3j --stdout=Microsoft-MIEngine-Out-ioltiggh.23p --stderr=Microsoft-MIEngine-Error-rvzj3eoc.ucl --pid=Microsoft-MIEngine-Pid-do2mrfiv.owu --dbgExe=C:\msys64\ucrt64\bin\gdb.exe --interpreter=mi "
```

<Game Select>

1. APT game: In this game, everyone placing both hands on top of each other simultaneously, and the host announces a floor number. As players count up from the bottom, they remove their hands one at a time and place them on top. The player whose hand lands on the announced floor number is the "winner" (or chosen player) for that round.

2. 369 game: In this game involves counting up from 1, and whenever a number contains the digits 3, 6, or 9 players must clap instead of saying the number.
For example:
Number 13: Clap once
Number 33: Clap twice

3. BR21 game: In this game, participants take turns and count sequentially from 1 to 31. On each turn, a player can call out between 1 to 3 consecutive numbers. The player who has to say 31 loses the game.

Jenny's favorite random game! Which game! Game start!
Jenny's choice:

(2) 게임 선택 입력 처리

- 설명 : APT, 369, BR31을 입력하면 해당 게임 실행,

그 외 단어 입력 시 술 한잔 더 마시고 다시 게임 이름 입력 받음

- 테스트 결과 스크린샷

APT경우

```
Jenny's favorite random game! Which game! Game start!  
Jenny's choice: APT  
APT~ APT~ APT~ APT~  
Which floor?(1 to 20 floors only): █
```

369경우

```
Jenny's favorite random game! Which game! Game start!  
Jenny's choice: 369  
Three_Six_Nine game is not yet implemented.  
Jenny's favorite random game! Which game! Game start!  
Jenny's choice: █
```

BR21 경우

```
Jenny's favorite random game! Which game! Game start!  
Jenny's choice: BR31  
Baskin_Robbins_31 game is not yet implemented.  
Jenny's favorite random game! Which game! Game start!  
Jenny's choice: █
```

그 외 입력 경우

```
Jenny's favorite random game! Which game! Game start!  
Jenny's choice: BunnyBunny  
Fool shot! Fool shot! One more time!  
have a shot  
Jenny has drunk 1 glasses  
  
Jenny's favorite random game! Which game! Game start!  
Jenny's choice: █
```

(3) APT실행시 층 범위 외 입력 오류

- 설명 : 너무 코드 실행이 길어질 것을 염두에 두어 1층부터 20층 이외의 층을 입력받을 경우 Invalid input!을 출력하도록 했다.

- 테스트 결과 스크린샷

```
Jenny's favorite random game! Which game! Game start!  
Jenny's choice: APT  
APT~ APT~ APT~ APT~  
Which floor?(1 to 20 floors only): 100  
Invalid input! Please enter a floor between 1 and 20.  
Which floor?(1 to 20 floors only): █
```

(4) APT 정상 층 입력 실행 시 순서대로 층 출력 후 종료되는지 확인

- 설명 : 8층 입력시 1층부터 순서대로 8층까지 출력한 후 8층에 해당하는 Taeun이 탈락했다고 출력한다.
- 테스트 결과 스크린샷

```
Jenny's choice: APT
APT~ APT~ APT~ APT~
Which floor?(1 to 20 floors only): 8
Taeun: 1 floor
Lisa: 2 floor
Jenny: 3 floor
Jisoo: 4 floor
Jisoo: 5 floor
Jenny: 6 floor
Rose: 7 floor
Taeun: 8 floor
Taeun is out of the game
have a shot
Taeun has drunk 1 glasses

Taeun's favorite random game! Which game! Game start!
Taeun's choice: █
```

(5) APT 매번 랜덤 배정 되는지 확인

- 설명 : APT를 2번 이상 실행시켜서 매번 랜덤하게 학생들에게 층 숫자가 배정되는지 확인 테스트
- 테스트 결과 스크린샷

```
APT~ APT~ APT~ APT~
Which floor?(1 to 20 floors only): 8
Taeun: 1 floor
Lisa: 2 floor
Jenny: 3 floor
Jisoo: 4 floor
Jisoo: 5 floor
Jenny: 6 floor
Rose: 7 floor
Taeun: 8 floor
Taeun is out of the game
have a shot
Taeun has drunk 1 glasses

Taeun's favorite random game! Which game! Game start!
Taeun's choice: APT
APT~ APT~ APT~ APT~
Which floor?(1 to 20 floors only): 6
Jisoo: 1 floor
Taeun: 2 floor
Lisa: 3 floor
Lisa: 4 floor
Jenny: 5 floor
Rose: 6 floor
Rose is out of the game
have a shot
Rose has drunk 1 glasses

Rose's favorite random game! Which game! Game start!
Rose's choice: 
```

(6) APT 탈락자가 다음 게임 차례로 설정 및 술 카운트 증가

- 설명 : 탈락자가 다음 게임 차례로 설정되고 술 카운트가 증가 되는지 확인

아래에서 APT탈락자가 다음 Turn이 되고 다음 게임 입력처리때 없는 게임인 bunny를 입력해서 술이 증가할 때 이미 증가되어 있는 것을 바탕으로 2잔을 has drunk했다고 출력되는 것을 볼 수 있다.

- 테스트 결과 스크린샷

```

Jenny's favorite random game! Which game! Game start!
Jenny's choice: APT
APT~ APT~ APT~ APT~
Which floor?(1 to 20 floors only): 4
Jenny: 1 floor
Rose: 2 floor
Rose: 3 floor
Taeun: 4 floor
Taeun is out of the game
have a shot
Taeun has drunk 1 glasses

Taeun's favorite random game! Which game! Game start!
Taeun's choice: bunny
Fool shot! Fool shot! One more time!
have a shot
Taeun has drunk 2 glasses

Taeun's favorite random game! Which game! Game start!
Taeun's choice: 

```

(7) 술 누적 결과 표시

- 설명 : 술 누적이 정상적으로 이루어지고 출력되는지 누적 시켜본다.
- 테스트 결과 스크린샷

```

Jenny's favorite random game! Which game! Game start!
Jenny's choice: bunny
Fool shot! Fool shot! One more time!
have a shot
Jenny has drunk 1 glasses

Jenny's favorite random game! Which game! Game start!
Jenny's choice: bunny
Fool shot! Fool shot! One more time!
have a shot
Jenny has drunk 2 glasses

Jenny's favorite random game! Which game! Game start!
Jenny's choice: bunny
Fool shot! Fool shot! One more time!
have a shot
Jenny has drunk 3 glasses

Jenny's favorite random game! Which game! Game start!
Jenny's choice: bunny
Fool shot! Fool shot! One more time!
have a shot
Jenny has drunk 4 glasses

Jenny's favorite random game! Which game! Game start!
Jenny's choice: 

```

4. 계획 대비 변경 사항

1) 기능1(게임시작 및 선택)의 세부기능 구체화

- 이전: 세부기능 3가지

- (1) 세부 기능 1: 참여자 설정
- (2) 세부 기능 2: 게임 리스트 표시 및 설명 출력
- (3) 세부 기능 3: 게임 선택 입력 처리

- 이후: 세부기능 4가지

- (1) 세부 기능 1: 참여자 대학생 클래스 생성
- (2) 세부 기능 2: 참여자 대학생 객체 생성
- (3) 세부 기능 3: 게임 리스트 표시 및 설명 출력
- (4) 세부 기능 4: 게임 선택 입력 처리

- 사유

참여자들을 클래스 생성 후 벡터 안에 객체로 만들다 보니까 기능 구현 부분에서 클래스에 대한 기능 설명을 위한 파트가 필요하다고 느꼈다. 그래서 세부기능 1번을 1과 2번으로 나누어서 1부분에 클래스 부분 설명을 추가했다.

2) 기능2(APT게임)의 세부기능 구체화

- 이전: 세부기능 2가지

- (1) 세부 기능 1: 숫자 랜덤화 및 진행
- (2) 세부 기능 2: 실패 시 술 누적

- 이후: 세부기능 3가지

- (1) 세부 기능 1: 게임 시작 및 숫자 랜덤화
- (2) 세부 기능 2: 순서대로 총 출력 후 탈락

(3) 세부 기능 3: 탈락자가 다음 게임 차례로 설정 및 술 카운트 증가

- 사유

처음에 제안서 작성시에는 아파트 게임은 간단히 구현할 수 있을 것이라고 생각했는데, 랜덤 배정 이후 탈락자 결정시에 1층부터 순서대로 출력하다가 탈락자를 결정하려고 하다 보니 함수가 추가로 더 들어가게 되어서 좀 더 구체화 해서 작성했다.

2) 기능5(술 누적 시스템)의 구현 일정보다 앞당김

- 이전: 12/15-12/22 기간동안 구현 예정

- 이후: 세부기능 1, 2번을 먼저 구현

- 사유

Student클래스를 생성 후 객체를 사용하다 보니 자연스럽게 클래스 내에 술 누적 관련 변수와 함수를 넣게 되었고, 게임 종료시마다 출력하는 것을 미리 구현해두고 세부기능 3번인 종료 조건을 나중에 구현하는 것이 좋다고 생각해서 미리 구현하게 되었다.

5. 프로젝트 일정

업무		11/3	11/10	11/17	12/1	12/15	12/22
제안서 작성		완료					
기능1	세부기능1		완료				
	세부기능2		완료				
	세부기능3		완료				
	세부기능4		완료				
기능2	세부기능1			완료			
	세부기능2			완료			
	세부기능3			완료			
중간보고서1				완료			
기능3	세부기능1				----->		
	세부기능2				----->		
	세부기능3				----->		
중간보고서2					----->		
기능4	세부기능1					----->	
	세부기능2					----->	
	세부기능3					----->	
중간보고서3						----->	
기능5	세부기능1			완료			
	세부기능2			완료			
	세부기능3						
최종보고서							----->