2022 고급C++프로그래밍 Project 1: 윷놀이 구현하기 1

due: 4월 24일(일) 23:59:59

1. Problem Statement

- 플레이어(사람)은 컴퓨터 1대(총 참가자 수가 둘), 또는 2대와 윷놀이를 한다(총 참가자 수가 셋).
- 모든 판은 한 명의 플레이어라도 도착점에 다다르면 게임을 종료한다.
- 각 참가자에게 지급된 말은 하나다.
- 윷놀이에서는 상대편 말을 잡으면, 한 번 더 윷을 던진다.
- 말이 잡히면 처음부터 다시 시작한다.
- 윷, 모가 나오면 한 번 더 윷을 던진다.
- 한번 더 던지는 기회가 왔을 때, 결과의 순서를 바꿔 이동할 수 없다(e.g. 윷 + 걸이 나왔을 때, 걸로 먼저 이동하여 분기점에 도착한 뒤, 윷으로 이동할 수 없다.)
- 뒷도는 없다.
- 그 외 규칙은 일반적인 윷놀이 규칙을 따른다.

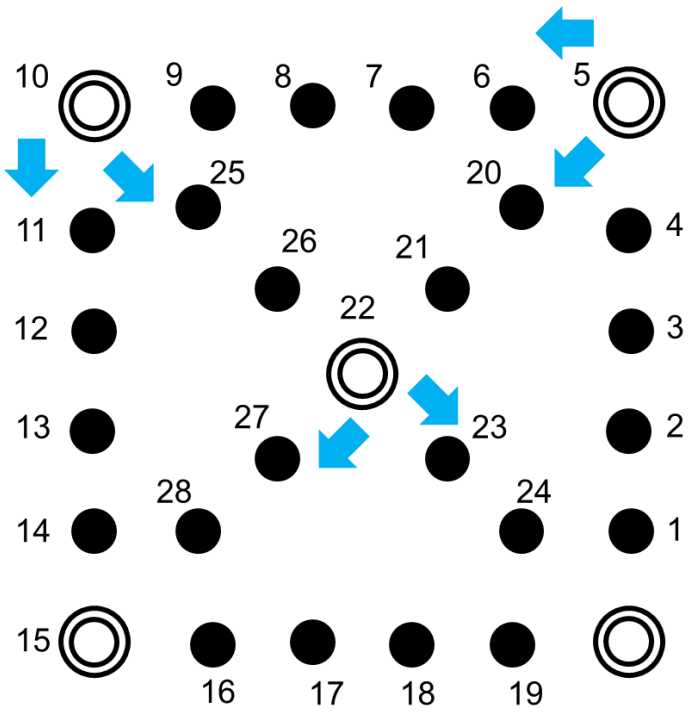


그림 1. 윷놀이 판과 분기점.

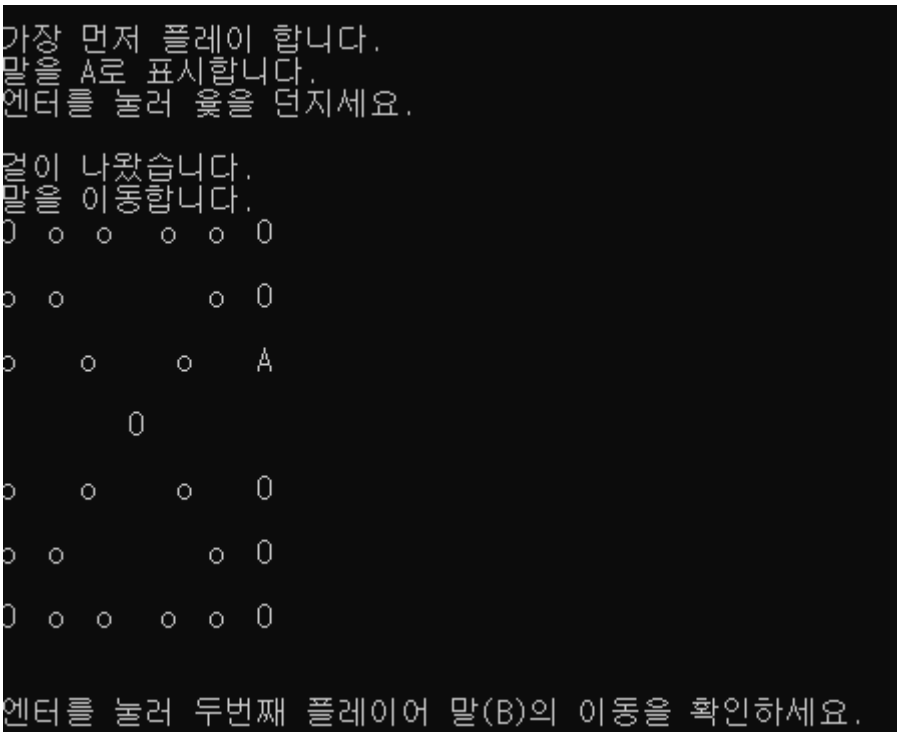
■ 구현 규칙

프로그램은 가장 먼저 play.in의 입력 파일에 따라 모드1과 모드2로 구별하여 아래의 내용을 별개로 실행한다.

1. 모드 1.

1. 모드1 시작 시 아무 말도 없는 윷놀이 판을 콘솔창에 출력한다.
2. 플레이어의 순서를 랜덤하게 결정한다.
3. 엔터를 누를 때마다 참가자들(플레이어와 컴퓨터)는 윷을 던지고, 판의 변화를 콘솔창과 로그 파일에 출력한다.
4. 로그 파일에는 모든 참가자들이 던진 윷의 결과, 판의 변화를 순서에 따라 기록한다.
5. 윷놀이 판의 표시 형식과 로그 파일 출력 양식의 세부 사항은 학생 자율에 맡긴다.

· 게임 구현 예시 1. 콘솔창



· 게임 구현 예시 2. play.log

play.log

```
1  플레이어의 플레이 순서는 3명의 참가자 중 두번째입니다.
2  컴퓨터1이 A로 표시됩니다.
3  플레이어는 B로 표시됩니다.
4  컴퓨터2는 C로 표시됩니다.
5  모드1을 실행합니다.
6
7  컴퓨터1이 윷을 던집니다.
8  걸이 나왔습니다.
9  O  o  o  o  o  O
10 o  o        o  o
11 o     o     o     A
12          O
13 o     o     o     o
14 o  o        o  o
15 O  o  o  o  o  O
16 플레이어가 윷을 던집니다.
17 개가 나왔습니다.
18 O  o  o  o  o  O
19 o  o        o  o
20 o     o     o     A
21          O
22 o     o     o     B
23 o  o        o  o
24 O  o  o  o  o  O
25 컴퓨터2가 윷을 던집니다.
```

모드 2.

플레이 과정의 콘솔창 입출력 없이 실행 즉시 플레이어는 컴퓨터 1대와 10만번, 또는 1분 이내로 윷놀이를 반복한다(총 참가자 수는 2인, 플레이 횟수가 10만번이 되거나 실행 후 1분이 지나면 프로그램 종료). 그리고 play.out 에 결과를 출력한다.

1. 플레이하는 순서에 따라 달라지는 승률(소수점 한자리까지, 반올림하여 %단위).
2. 플레이하는 순서에 따라 윷놀이 판 위치에 방문하는 횟수(한번 더 기회가 주어진 경우, 방문한 모든 곳을 횟수에 포함한다. e.g. 윷과 걸이 나온 경우, 윷으로 이동한 위치와 걸로 이동한 위치 모두 해당 위치의 방문 횟수에 포함).

모드 2에서는 로그파일을 출력하지 않는다.

입력파일과 .exe 파일은 한 폴더 내에 위치하며, 출력파일도 같은 폴더 안에 생성한다.

[입력]

입력 파일의 이름은 play.in이다. 입력 파일의 각 항목은 공백과 줄바꿈으로 구분되며, 정수다.

[출력]

출력 파일의 이름은 모드1의 출력인 play.log와 모드2의 출력인 play.out이다.

[입출력 예]

입력. play.in

1 2 2

(모드값) (플레이어 숫자) (각 플레이어가 사용하는 말의 숫자)

출력1. play.log

모드값이 1일 때 플레이 로그 파일로 최소 출력 내용은 다음과 같다.

1행: 플레이어의 플레이 순서 표기.

2행 – 4행: 학생이 사용한 플레이어 표기 방법.

5행: 모드1, 모드2 여부.

6행: 빈칸.

7행 이하: 플레이 이력

플레이 이력은 플레이어의 플레이 회차마다 다음 순서로 표기한다.

1. 누가 윷을 던졌는지
2. 던진 윷의 결과
3. 판의 변화

예시.

play.log

```
1  플레이어의 플레이 순서는 3명의 참가자 중 두번째입니다.
2  컴퓨터1이 A로 표시됩니다.
3  플레이어는 B로 표시됩니다.
4  컴퓨터2는 C로 표시됩니다.
5  모드1을 실행합니다.
6
7  컴퓨터1이 윷을 던집니다.
8  걸이 나왔습니다.
9  O  o  o  o  o  O
10 o  o        o  o
11 o     o     o     A
12          O
13 o     o     o     o
14 o  o        o  o
15 O  o  o  o  o  O
16 플레이어가 윷을 던집니다.
17 개가 나왔습니다.
18 O  o  o  o  o  O
19 o  o        o  o
20 o     o     o     A
21          O
22 o     o     o     B
23 o  o        o  o
24 O  o  o  o  o  O
25 컴퓨터2가 윷을 던집니다.
```

출력.2 play.out

모드값이 2일 때 결과 파일로 최소 출력 내용은 다음과 같다.

1. 선공 시 승률(%)
2. 후공 시 승률(%)
3. 선공 시 위치 별 방문 횟수(위치, 횟수)
4. 후공 시 위치 별 방문 횟수(위치, 횟수)

예시.

```
1.
선공 시 승률: 100.0%
후공 시 승률:   0.0%
2.
선공 시 방문 횟수(위치, 횟수)
1, 21404
2, 62309
3, 63453
.
.
.
후공 시 방문 횟수(위치, 횟수)
1, 35123
2, 59120
3, 60983
.
.
```

2. Restriction

이번 과제에서는 클래스를 반드시 사용하도록 한다. 구조체을 반드시 사용할 필요 없다. 이번 과제를 포함, 이 강좌의 모든 과제에서 제출할 프로그램은 반드시 CPU 상에서 single thread로 수행되어야 한다. 멀티코어 혹은 멀티쓰레드 컴퓨팅은 허용하지 않으며, GPU 활용 역시 허용하지 않는다. 프로그램은 외부와 통신 없이 단독으로 수행되어야 한다. 앞의 제약을 위배한 경우 큰 페널티를 받는다.

과제 마감 1주일 전부터 과제와 관련된 어떤 질문도 허용하지 않는다.

3. Language

프로그램 언어는 C++을 사용한다. 다른 언어를 사용하는 경우 미리 교수와 상의하여 승인된 경우에 한해 허용한다. 기본적으로 제공하는 standard library 이외의 다른 라이브러리를 사용하는 경우에는 미리 교수와 상의해야한다. 교수와 상의 없이 다른 언어나 라이브러리를 사용하는 경우, 큰 페널티를 받을 수 있다.

4. Compilation and Execution

채점 컴퓨터는 Intel(R) Core(TM) i7-11700 2.50GHz이며 OS는 Ubuntu 12.04 LTS이다. (컴퓨터의 사양 차이를 고려하여 프로그램을 작성하는 것을 권장한다.)

5. Report

프로젝트에 대해 보고서를 작성하여 제출한다. 분량은 A4 4페이지 이하이며, 겉표지 없이 바로 제목과 소속, 학번, 이름을 첫 페이지 상단에 작성한 뒤 곧바로 내용을 작성한다. 보고서 내에 소스코드를 그대로 포함해서는 안 된다. 서식은 자유형식이다. 보고서 본문에는 최소한 다음 내용을 포함시킨다.

1. 프로그램 플로우차트

2. 사용한 클래스에 대한 설명

3. 플레이어의 플레이 순서에 따른 기대 승률

4. 플레이 순서가 승률에 영향을 미치지 않는다고 판단한다면 그 이유를 논의. 플레이 순서가 승률에 크게 영향을 미친다고 판단한다면, 승률을 비슷하게 만들기 위한 방법.

5. Discussion (느낀 점, 잘 안 되는 점, 의외의 현상, 예상대로 된 점 등)

6. Submission

"PROJ1" 파일명으로 하여 zip 또는 tar(+gz) 형식으로 압축한다(ex. PROJ1.zip or PROJ1.tar). 압축 파일에는 소스코드(.cpp), 실행파일(.exe), 그리고 보고서(pdf)가 포함되어야 한다. 첨부파일은 압축 파일 하나만 있어야하며, 첨부파일이 없거나 둘 이상 있으면 제출로 인정하지 않는다. 압축을 풀었을 때 상기 파일을 바로 사용할 수 있어야 하며, 폴더 등을 압축하지 않는다. .exe 파일을 만들기 어려운 경우 MAKEFILE과 함께 제출하는 것을 허가한다. 제출형식이 다른 경우 제출로 인정하지 않는다.

7. Grading

- 보고서와 프로그램이 각각 1:1의 비율로 반영된다. 둘 모두 상대평가에 의해 점수가 부여된다.

- 각 보고서는 위의 최소 내용을 모두 포함할 경우 기본 점수가 부여되며 나머지 성적은 보고서 내용이 충실한 정도에 따라 평가한다. 보고서에 불필요한 내용(표지, 소스코드, 별도의 분석 없는 raw data 등)을 포함하는 경우 감점요인이 된다.

- 타인의 source code는 절대 보지 말 것. 자동 Copy detection program에 의하여 카피를 판별한 후 원본과 복사본 공히 0점 처리한 후 최종 학점에서 한 grade 강등함. 일부든 전부든 모두 카피로 처리함. Source code copy 후 변경도 detection 됨.

- 수강생이 많은 관계로 프로그램 채점의 많은 부분이 자동적으로 이루어질 예정이다. 수강생의 실수로 인해 예외적인 상황이 발생하여 전체 채점이 지연될 경우 큰 감점 요인이 될 수 있다.

- 과제물을 늦게 제출한 경우, 교수에게 최종적으로 도착한 시간을 기준으로 매 24시간마다 20% 씩 감점된다. 채점 및 규정 적용은 제일 마지막 제출물에 따라 처리된다. 보고서와 프로그램을 따로 제출할 수는 없다.