|  |
| --- |
| [**C++프로그래밍및실습**](https://sel.jnu.ac.kr/course/view.php?id=55996)  **C++ 미니게임**  **진척 보고서 #1** |

모든 기능은 main(임시)에서 확인할 수 있습니다!

|  |  |
| --- | --- |
|  | 제출일자: 2024-11-15  제출자명: 김휘승  제출자학번: 214958 |
|  |  |

**1. 프로젝트 목표 (16 pt)**

**1) 배경 및 필요성 (14 pt)**

수업시간에 terminal 환경에서 수행하는 기초적인 게임들을 배워봤습니다. 흥미로 운 게임들이지만, 2인 이상의 플레이어가 필요하거나 볼륨이 작은 게임이 많았습니다. 과제 주제를 고민하던 중, 어렸을 때 폴더폰으로 즐겼던 ‘미니게임 천국’이 떠올랐고, 이를 참고해 나만의 ‘미니게임 천국’을 만들면 컴퓨터와 함께 혼자서도 충분히 즐길 수 있는 환경을 제공할 수 있을 것이라 생각했습니다. 또한, 여러 작 은 게임을 하나의 프로그램으로 묶음으로써 볼륨 문제도 해결할 수 있다고 판단했습니다. 이와 같은 이유로 나만의 ‘미니게임 천국’을 만들어 보기로 했습니다.

**2) 프로젝트 목표**

터미널 환경에서 컴퓨터와 함께 혼자서 즐길 수 있고, 여러가지 게임을 담은 나 만의 ‘미니게임 천국’ 만들기

**3) 차별점**

1. 컴퓨터와의 상호작용: 모든 미니게임에서 컴퓨터가 플레이어의 행동에 반응하며 특정 결과를 제공합니다. 이를 통해 혼자서도 재미있게 즐길 수 있습니다.

2. 점수와 보너스 시스템: 각 미니게임에서 승리할 때마다 점수가 주어지며, 누적된 점수가 일정 기준에 도달하면 다음 게임에서 보너스를 받을 수 있습니다.

3. 랜덤 플레이 모드: 플레이할 게임이 무작위로 결정되는 랜덤 플레이 모드를 통 해 게임에 신선함을 더했습니다.

**2. 기능 계획**

**1) 기능 1 : 게임 구현하기**

**- 설명: 게임은 총 4개 (묵찌빠, 틱택토, 숫자 배열하기, 단어 맞추기)로 구성한다.**

(1) 세부 기능 1: 묵찌빠 & 틱택토 게임&컴퓨터 행동 알고리즘

- 설명: 플레이어의 행동 후 다음 행동은 컴퓨터가 수행합니다. 컴퓨터의 행동은 랜덤으로 결정되나, 룰을 준수해야 합니다. (예: 컴퓨터가 돌이 놓여있는 칸에 돌을 놓지 않는다.)

(2) 세부 기능 2: 단어 맞추기 게임& 단어 불러오기

- 설명: 컴퓨터가 제공할 단어는 텍스트 파일에서 불러오고, 텍스트 파일은 단어 주제에 따라 분류합니다. (동물, 과일, 나라이름)

(3) 세부 기능 3 숫자 배열하기 게임& 숫자 배열 알고리즘

- 설명: 컴퓨터가 무작위 숫자를 제공하고, 컴퓨터는 merge sort 알고리즘을 통해 숫자를 크기순으로 정렬합니다. 유저는 특정한 시간 안에 숫자를 크기순으로 정렬해야 합니다.

(4) 세부 기능 4: 누적 점수 기능

- 설명: 게임에서 이기면 점수가 주어지고 누적 점수가 일정 점수를 넘으면 다음 게임에 서 보너스를 제공합니다. 묵찌빠는 패배 방어 1회, 틱택토는 추가 행동 1회, 숫자 배열하 기는 추가 시간 5초, 단어 맞추기는 질문권을 3회 더 제공합니다.

**2) 기능 2 : 메뉴 구현하기**

**- 설명: 메뉴에서 플레이할 게임을 선택합니다.**

(1) 세부 기능 1: 랜덤 플레이

- 설명: 플레이할 게임이 랜덤으로 결정합니다.

(2) 세부 기능 2: 설명 & 점수 표시

- 설명: 메인 메뉴에서 누적 점수를 확인하고, 각 게임의 설명을 제공합니다.

**3. 진척사항**

**1) 게임 구현하기**

**(1-1) 틱택토 컴퓨터 행동 알고리즘**

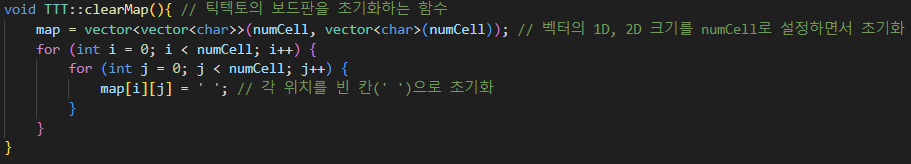
**- 입출력 :** 보드판(map), 좌표(x, y), 보드 한 줄의 칸 수(numCell), 플레이어의 돌(currentPlayer)

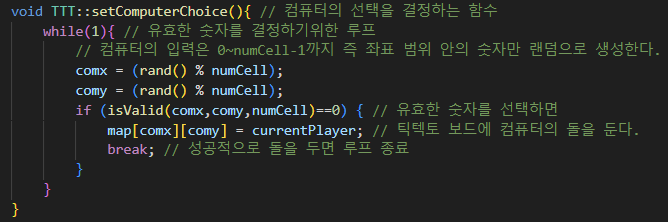
**-** **설명 :** 틱택토 보드판은 vector로 구현, 게임 시작 시 clearMap()함수를 호출, 보드판의 크기를 설정 후, 각 칸을 빈칸(‘ ‘)로 초기화한다.

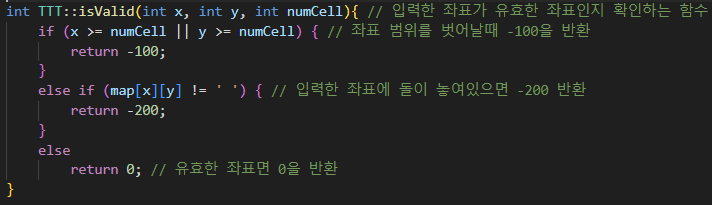
컴퓨터의 행동은 setComputerChoice()함수를 통해 결정된다. 컴퓨터는 보드의 좌표 범위 안의 숫자만 랜덤으로 선택하고, 선택한 수를 isValid()함수를 통해 유효한 좌표를 선택했는지 확인한 후 유효한 좌표면 해당 좌표에 돌을 둔다.

**- 적용된 배운 내용 :** vector, 조건문&반복문, 원하는 범위 내의 무작위 수 생성(rand() % numCell, 게임 실행 함수에 srand(time(0))를 추가해 무작위 난수를 생성한다.)

**- 코드 스크린샷**







**(1-2) 묵찌빠 컴퓨터 행동 알고리즘**

**- 입출력 :** 사용자의 선택(UserChoice), 컴퓨터의 선택(ComputerChoice), 공격권(turn)

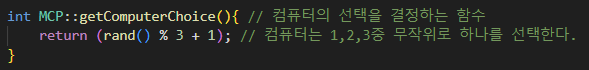
**-** **설명 :** 1은 묵, 2는 찌, 3은 빠를 나타낸다. 컴퓨터는 1~3까지 무작위의 수를 선택한다.

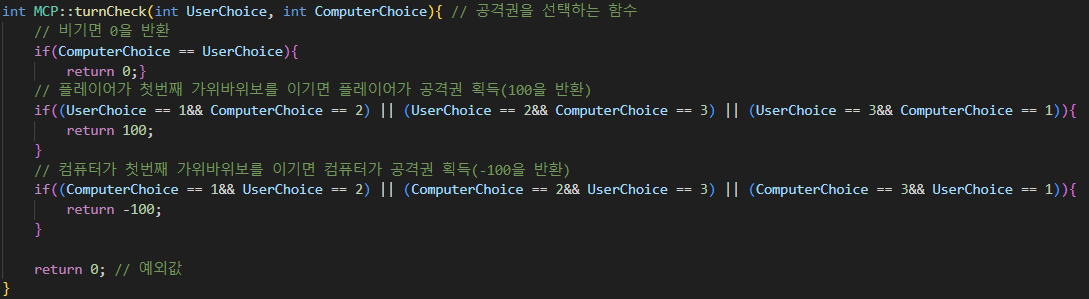
공격권은 첫번째 가위바위보를 이긴 사람이 가지며 이 상태에서 가위바위보를 비기면 승리, 이기면 공격권 유지, 지면 공격권을 상대에게 넘긴다.

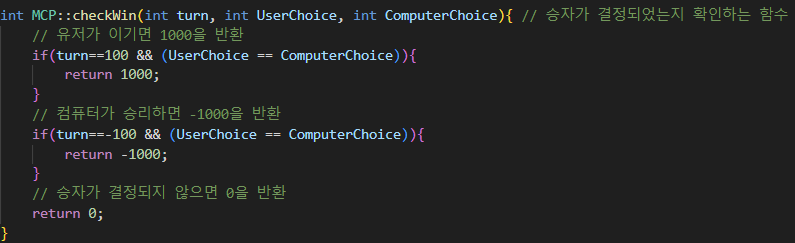
공격권이 설정된 상태에서 두번째 가위바위보를 비기면(사용자의 선택과 컴퓨터의 선택이 같으면) 공격권이 있는 플레이어가 승리한다.

**- 적용된 배운 내용 :** 조건문, 원하는 범위 내의 무작위 수 생성(rand() % numCell, 게임 실행 함수에 srand(time(0))를 추가해 무작위 난수를 생성한다.)

**- 코드 스크린샷**

****

****

****

**(1-3) 게임 구현의 전체적인 특징**

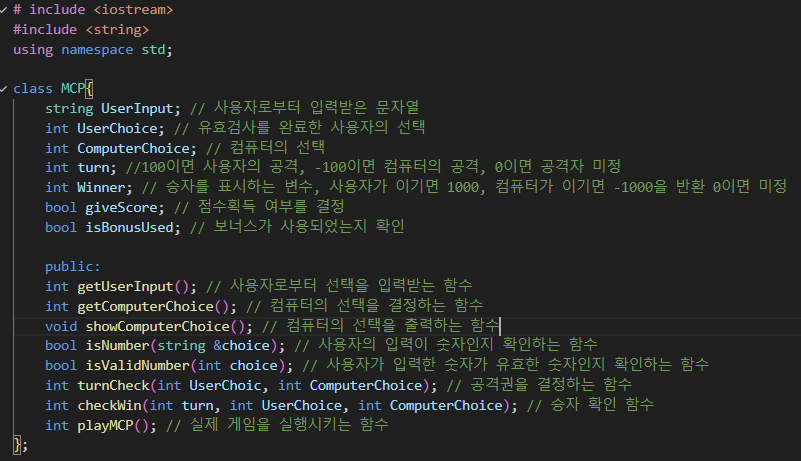
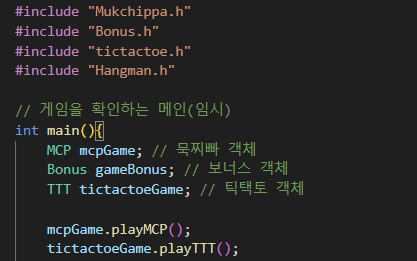
**- 설명 : 게임 구현이라는 추상적인 내용을 보충 설명하기 위한 내용이다.**

**- 특징 1 :** 모든 게임은 헤더파일(.h)과 실행파일(.cpp)로 구성되어 있고 게임 실행은 메뉴화면(현재 : main(임시))에서 각 게임의 객체를 생성한 후 객체의 게임실행 메소드를 호출하는 방식으로 이뤄진다.

- **특징 2 :** 사용자로부터 입력을 받을 경우 모든 입력은 유효성검사를 거치게 된다. 가령 특정 범위의 숫자(1,2,3)를 입력해야 하는 경우 사용자의 입력한 문자가 숫자인지(isNumber()), 사용자의 입력이 유효한 숫자인지(isValidNumber()) 확인한다.

- **특징 3** **:** 각 게임 실행파일(.cpp)에서 보너스 객체를 생성한 후 보너스 사용 여부를 물어보고 사용 여부를 isBonusUsed(bool)에 저장 후 보너스 기믹 사용여부를 결정한다.

**- 코드 스크린샷**

****

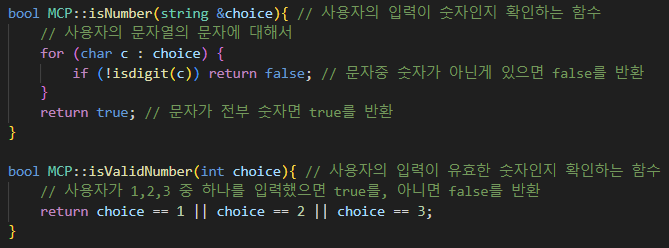
**묵찌빠 헤더파일**

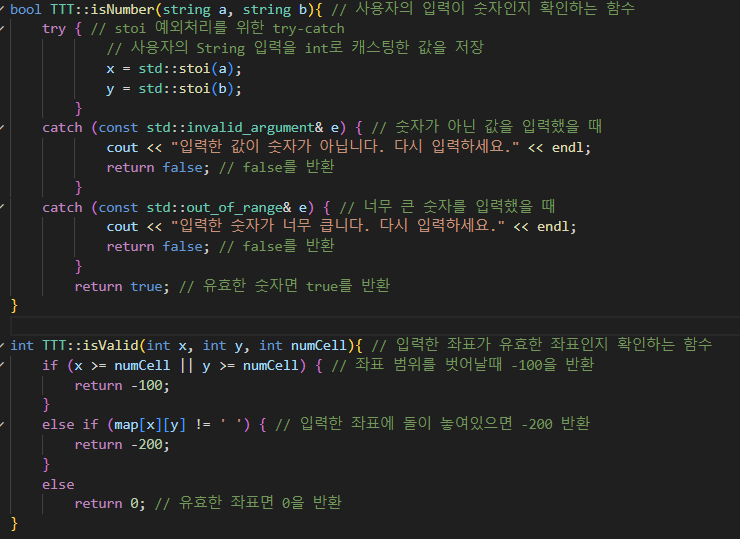
**메인 함수**

**각 게임의 객체를 생성 후**

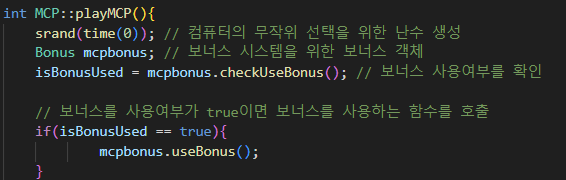
**객체의 게임 실행 함수를 호출한다.**

**2) 테스트 결과**

****

****

**보너스 객체 생성 & 보너스 사용 여부 확인**

****

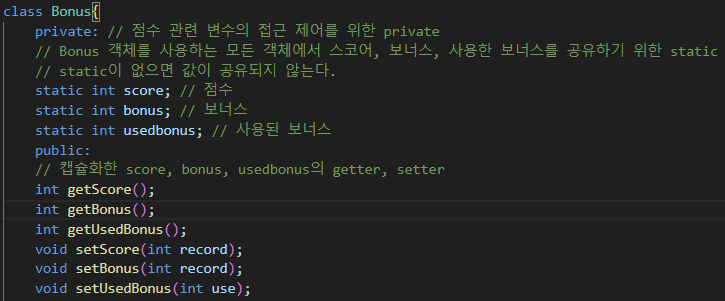
**(4-1) 누적 점수 기능**

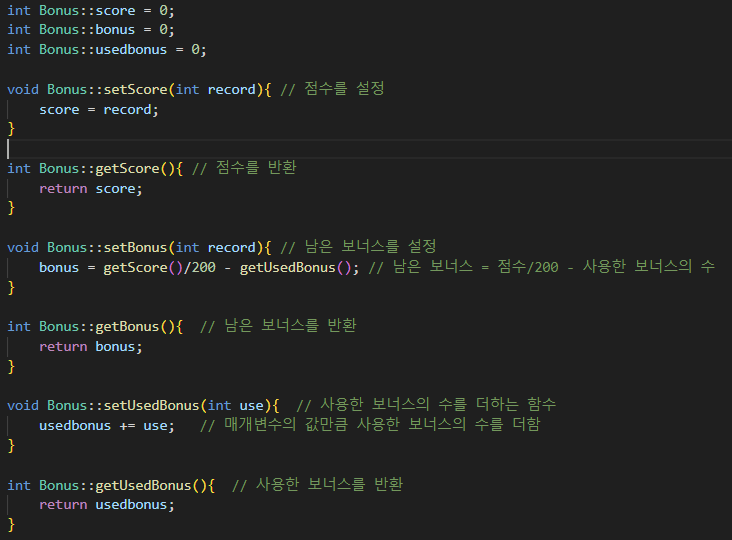
**- 입출력 :** 누적점수(score), 남은 보너스(bonus), 사용된 보너스(usedbonus)

**-** **설명 :** Bonus 객체를 사용하는 모든 객체에서 스코어, 보너스, 사용된 보너스를 공유하기 위해 변수를 static으로 설정했으며, 변수의 값을 실수로라도 바꾸는 상황을 줄이기 위해 접근 제어자를 private로 하고 getter, setter로 접근하도록 구성했다.

**- 적용된 배운 내용 :** 접근제어자, 캡슐화, 정적변수(static)

**- 코드 스크린샷**

****

****

**(4-2) 보너스 사용 기능**

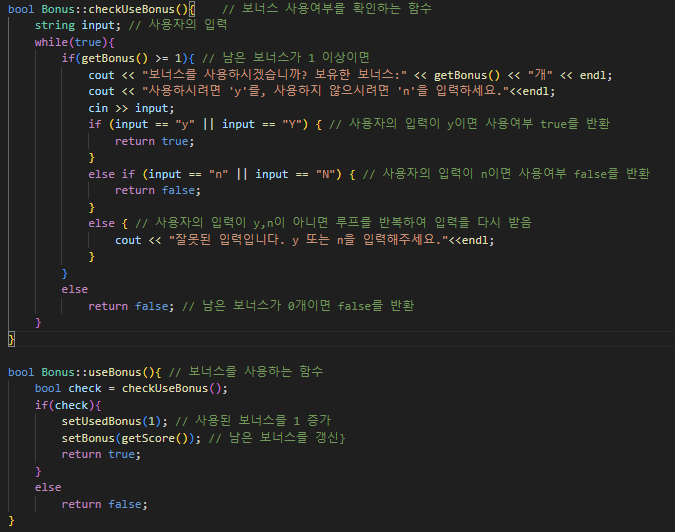
**- 입출력 :** 변수의 getter, setter

**-** **설명 :** 사용자로부터 유효한 입력(y,n 대문자 가능)을 받은 후 사용 여부를 결정한다.

보너스가 사용되면 사용된 보너스를 1 올린 후 남은 보너스를 갱신한다. 그 후 각 게임마다 보너스 기믹이 발동된다.

**- 적용된 배운 내용 :** 함수화, 캡슐화, 정적변수(static)

**- 코드 스크린샷**

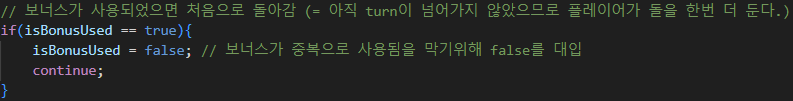
****

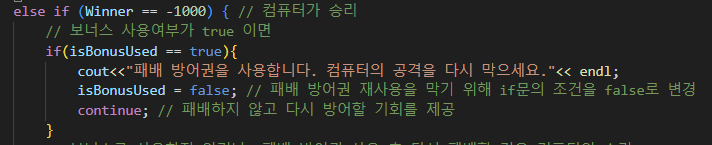
**묵찌빠, 틱택톡 객체에서 보너스 객체 생성**

****

****

**묵찌빠, 틱택토의 보너스 기믹 (패배 방어권, 돌 한번 더 두기)**

****

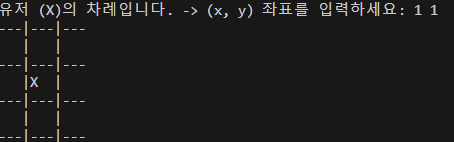
****

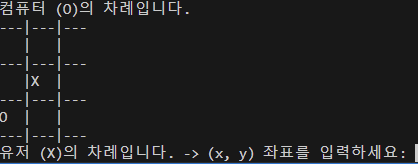
**2) 테스트 결과**

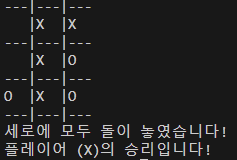
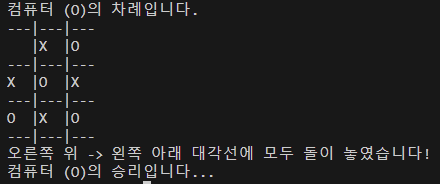
**(1-1) 틱택토 컴퓨터 행동 알고리즘**

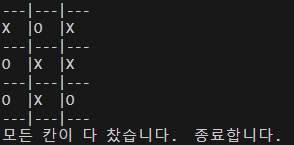
- 설명 : 유저가 돌을 둔 후 컴퓨터도 랜덤으로 돌을 둔다.

- 테스트 결과 스크린샷

****

****

****

****

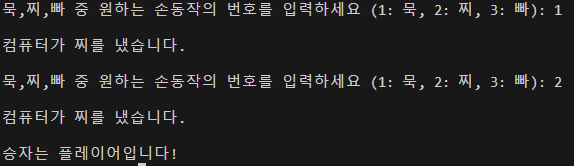
틱택토의 3가지 상황 승리, 패배, 무승부

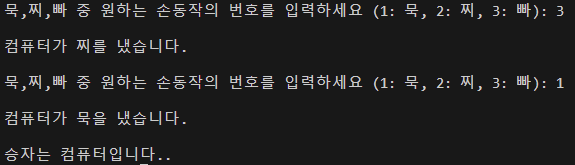
**(1-2) 묵찌빠 컴퓨터 행동 알고리즘**

- 설명 : 컴퓨터가 무작위로 가위바위보를 한다.

- 테스트 결과 스크린샷

묵찌빠의 2가지 상황 승리, 패배

****

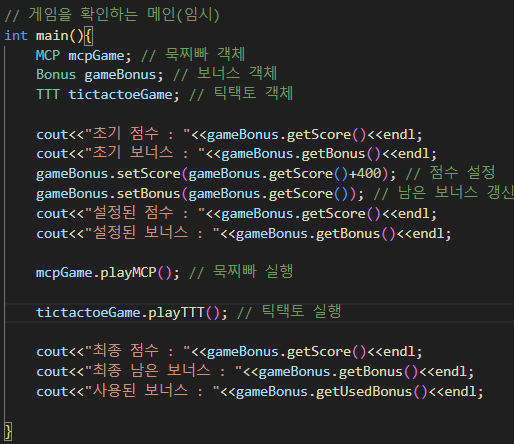
****

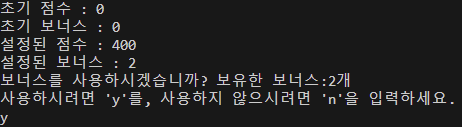
**(4-1), (4-2) 누적 점수기능 & 보너스 사용 기능**

- 설명 : 승리 시 점수 획득, 패배 또는 무승부 시 점수 획득 불가, 보너스 사용

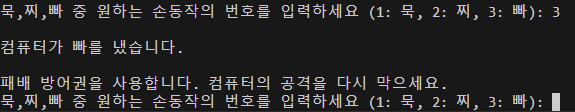
- 테스트 결과 스크린샷

점수, 보너스 설정 후 묵찌빠, 틱택토 연속으로 실행

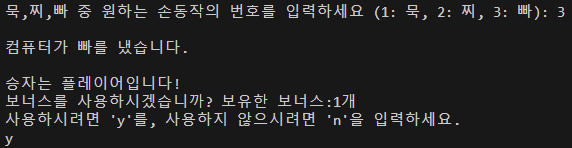
****

****

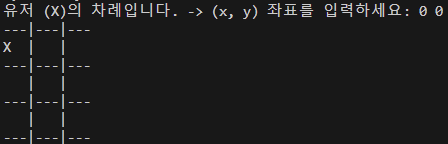
**보너스 사용 여부 확인 (y)**

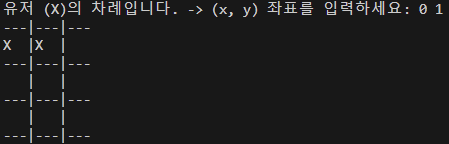
****

**묵찌빠 보너스 기믹 패배 방어권**

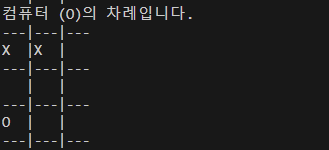
****

**보너스 사용 여부 확인 (y)**





**틱택토 보너스 기믹 추가 행동권**

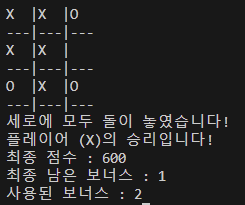


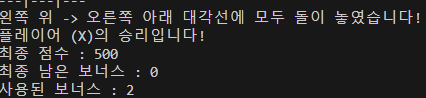
**승리시 +100**

**2승 했으므로 남은 점수는 400+200**

**200점당 보너스 = 남은 보너스 1개**

**사용된 보너스 = 2개**





**보너스 2번 사용, 묵찌빠 패배, 틱택토 승리**

**승리시 +100**

**1승 했으므로 남은 점수는 400+100**

**200점당 보너스 = 남은 보너스 0개**

**사용된 보너스 = 2개**

**5. 프로젝트 일정**

(진행한 작업과 진행 중인 작업 등을 표기)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **업무** | | **11/3** | **11/10** | **11/17** | **11/24** | **12/1** | **12/8** |
| **제안서 작성** | | **완료** |  |  |  |  |  |
| **기능1** | **세부기능1** |  | **완료** | |  |  |  |
| **세부기능2** |  |  | **진행중** |  |  |  |
| **세부기능3** |  |  |  | **------->** |  |  |
| **세부기능4** |  | **진행중** | | |  |  |
| **기능2** | **세부기능1** |  |  |  |  | **------->** |  |
| **세부기능2** |  |  |  |  | **-------------------->** | |