문제 정의

1. 첫 번째 선수의 이름을 입력받고 두 번째 선수의 이름을 입력받는다.

2. 입력을 받으면, "첫 번째 선수의 이름: <Enter>"가 출력되어 엔터키를 입력하면 난수가 나오는 게임을 시작할 수 있게 플레이어에게 알린다.

3. 난수를 생성해야 하고, 난수는 0, 1, 2 이 3개로 정의한다.

4. 선수는 Player 클래스, 게임은 GamblingGame 클래스로 작성한다.

5. 랜덤의 난수 3개가 나온 뒤, 전부 같으면 "<User>님 승리!!" 메세지가 나오고, 실패 시 "아쉽군요!" 메세지가 난수가 나온 줄 오른쪽에 출력한다.

문제 해결 방법

1. 랜덤 난수는 어떤 방법으로 만들 수 있을까? 파이썬 등 다른 언어랑 비슷하게 다른 함수가 있을까? - 정재현

2. 승과 패는 어떻게 구별지을 수 있을까? - 정민수

아이디어 평가

1. 난수를 생성하기 위한 방법을 검색해 본 결과, 다른 언어와 마찬가지로 난수생성을 사용, #include <cstdlib>, #include <ctime> 활용.

```
2  #include <cstdlib> // 난수 생성을 위한 헤더파일 rand() 등
3  #include <ctime> // 현재의 시간을 가져와서 랜덤 난수를 생성하기 위함
```

또한, 플레이 함수 내에서 랜덤 난수를 뽑기위한 과정.

```
22      for (int i = 0; i < 3; i++) {
23          numbers[i] = rand() % 3;
24      }
```

난수의 범위 지정은 여태 배워왔던 것들 중 하나인 나머지 활용법이 있다.

rand()로 난수를 생성 후, 3으로 나누면 그 나머지는 0, 1, 2 중 하나이기에 %를 활용한다.

```
26          // 다음 줄은 생성된 난수를 화면에 출력, " "를 활용하여 칸
27          for (int i = 0; i < 3; i++) {
28              cout << "     " << numbers[i] << "     ";
29              // 위의 줄에 endl을 넣으면 문제대로 난수가 오른쪽에 
30          }
```

그 후에, 이렇게 난수를 화면에 출력한다. 반복문을 사용하고, 공백도 넣어줌으로써 문제랑 똑같이 보기에 편리하게 바꾸어 준다. 또한, 문제랑 똑같이 만들기 위하여 endl;을 넣지 않는다. 마지막 키 아이디어 또는 알고리즘 설명에 첨부하였다.

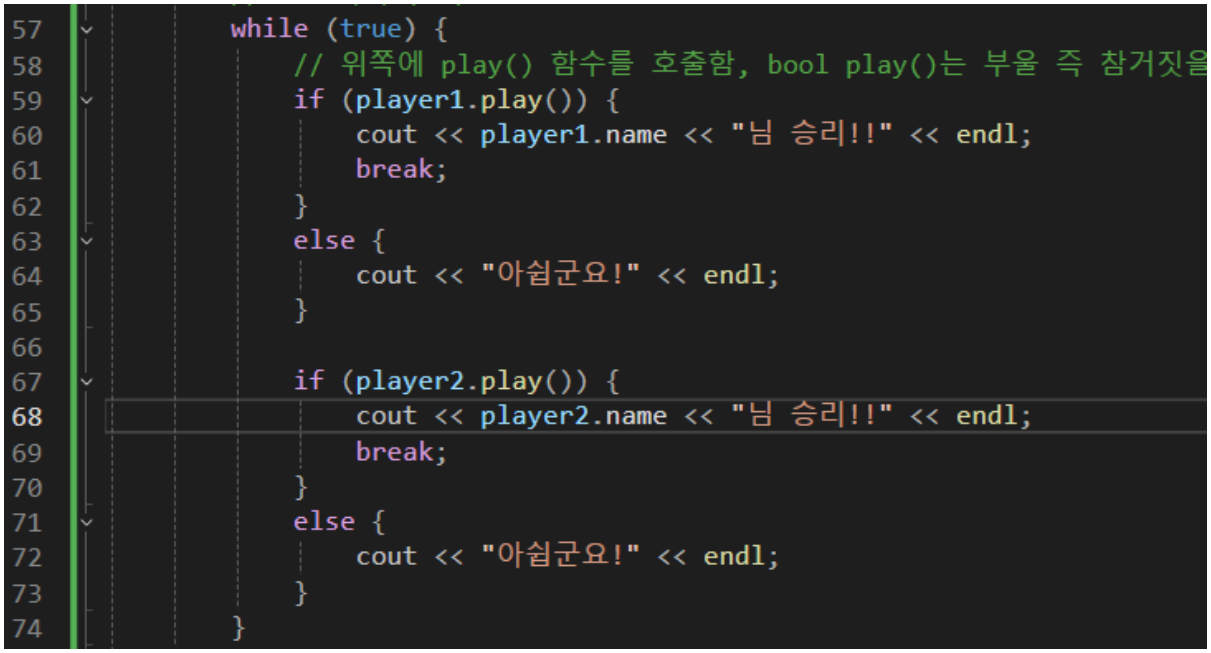
2. 승과 패는 어떻게 구별지을 수 있을까? 사실 if문과 while문을 결합하거나, switch~case를 활용할 수도 있다. 하지만 우리는 bool, 논리회로나 다른 언어에도 자주 등장하는 참거짓 판별 자료형 bool(항상 값은 2개, 참 또는 거짓<0or1>) 로 결정하였다.

```
33          if (numbers[0] == numbers[1] && numbers[1] == numbers[2]) {
34              return true;  // 승리
35          }
36          return false;  // 실패
37      }
```

게 &&(and)를 사용하면 numbers[0] == numbers[1] == numbers[2] 가 성립한다는 뜻이 되므로 3개의 숫자가 모두 같음을 판별할 수 있고, 그 외에는 전부 실패로 처리할 수 있다.

```
16      bool play() {
```

이렇게 play 함수에 bool 자료형을 사용하였다.



```
57          while (true) {
58              // 위쪽에 play() 함수를 호출함, bool play()는 부울 즉 참거짓을
59              if (player1.play()) {
60                  cout << player1.name << "님 승리!!" << endl;
61                  break;
62              }
63              else {
64                  cout << "아쉽군요!" << endl;
65              }
66
67              if (player2.play()) {
68                  cout << player2.name << "님 승리!!" << endl;
69                  break;
70              }
71              else {
72                  cout << "아쉽군요!" << endl;
73              }
74          }
```

while(true) 를 사용하여 무한루프로 만들어 놓고, bool play()에서 true(1)가 출력 될 경우, 플레이어별로 참이 될 경우 그 플레이어가 승리하는 형식으로, 무한루프가 break;를 통하여 끝나도록 만들었다.

문제를 해결한 키 아이디어 또는 알고리즘

여태까지 위에 정리한 방법들을 사용하여 큰 문제 없이 프로그램을 만들 수 있었다.

하지만, 문제랑 똑같이 만들기에는 여러 시행착오가 존재하였다.

첫 번째로, 선수2의 이름을 입력하자마자, 선수1의 게임이 이미 진행되어 버리고 선수2의 게임 차례가 찾아온다.

```
***** 갬블링 게임을 시작합니다. *****
첫번째 선수 이름>>정민수
두번째 선수 이름>>정재현
정민수: <Enter>
정민수:
        1               1               2
아쉽군요!
정재현: <Enter>
```

또한 문제에는 난수의 오른쪽에 "아쉽군요!"가 출력되는데 이 프로그램에서는 아랫줄에 출력된다. 처음에는 그래도 출력되니까 그대로 두었는데, 뭔가 가독성이 떨어져 보여서 수정하기로 결정했다. 이는 endl;을 없애고 쉽게 해결되었다.

```
***** 갬블링 게임을 시작합니다. *****
첫번째 선수 이름>>
정민수
두번째 선수 이름>>정재현
정민수: <Enter>
     0          2          2     아쉽군요!
정재현: <Enter>
```

하지만 가장 큰 문제는 두 번째 선수의 이름을 입력하면 첫 번째 선수는 이미 끝나고 두 번째 선수의 턴으로 넘어가는 것이었다. 솔직히 큰 문제는 없었지만 처음 보면 당황스러울 정도였다. 이를 수정하고자 방법을 찾아보니, 입력 버퍼에 남아있던 엔터키가 문제를 유발하는 것을 알았다.

그래서 두 번째 선수의 이름을 입력하고, cin.ignore(); 을 활용하여 개행 문자(엔터키)를 제거하여 주니 문제가 해결되었다.