

1. 문제 정의

두명의 플레이어의 각자의 이름을 입력받고 각 플레이어는 자신의 차례에 엔터키를 입력한다. 키를 입력받은 프로그램은 0부터 2까지의 범위에서 랜덤으로 숫자를 뽑아 3개의 숫자로 이루어진 배열을 생성한다. 만약 생성된 숫자가 모두 같으면 해당 플레이어가 승리한다. 승자가 결정될 때까지 게임은 계속 실행되며, 승자가 결정되면 승자를 알려주는 메시지를 출력함과 동시에 프로그램을 종료한다.

2. 문제 해결 방법

헤더에서 입력받을 두 선수의 이름을 넣을 배열을 정의하고 구현부(cpp)에서 0에서 2까지의 랜덤 숫자를 생성할 생성자를 만든다. 랜덤으로 나온 숫자들을 비교하는 조건문을 만들고 게임시작 생성자를 만들어 승자가 나올때까지 게임을 진행해야 하기 때문에 게임은 무한루프로 돌리고 종료되는 조건은 입력한 3자리 수와 랜덤 숫자가 모두 일치할때 승리하도록 설정한다. 동시에 게임을 끝내면서 이름님 승리 !! 라는 메시지가 나오도록 출력한다. 졌을 경우에는 아쉽군요 ! 메시지를 출력하고 순서를 바꿔 계속해서 프로그램을 진행한다.

아이디어 평가

선수 이름을 설정하는데 this 를 사용하여 name 값이 Player 객체의 멤버 변수임을 구분시켜주었다.

두 선수가 이름을 입력하고 게임을 시작하기 전에 순서를 정해야하는데 이 순서를 어떻게 정할지 고민하였다. 이를 해결하기 위해 순서인 i를 선수가 2명이기 때문에 2로 나누면 나머지가 0, 1 둘밖에 없다. 따라서 이 두 수를 통해 순서를 정하였다.

처음에 숫자를 cin으로 받았을때 공백을 인식을 못하는 것 같아 getline으로 입력을 받았더니 공백 포함 입력한 모든 것을 받아 해결하였다.

ranNum 메소드에서 숫자를 비교할 때 코드를 어떻게 짜야 최대한 단순하고 정확하게 할까 고민한 결과 맨 첫 숫자와 두번째 숫자를 비교하고, 다시 첫 숫자와 마지막 숫자를 비교하는 것이 빠르기 때문에 and 연산으로 조건문을 완성하였다.

무한루프를 수행하는 중, 승리시 출력할 입력받은 두 선수의 이름과 승리 여부를 가져와야 했다. 따라서 문자열 m이라는 변수에 해당 순서에 입력받은 선수의 이름을 가져오고, 숫자를 비교하여 결과를 평가한 문자열 n을 조건문에 넣어 승리하였는지 검사한다. 승리하였을 경우, m과 n을 붙여 출력하였다.

4. 키 아이디어와 알고리즘

클래스와 메서드

Player

- setName: 문자열 name으로 플레이어 이름을 설정
- getName: 플레이어의 이름을 반환
- Player* p = new Player[2] : 두 선수를 받기 위한 동적 배열을 생성

GamblingGame

- GamblingGame : 게임 시작 메시지를 출력하고 변수 생성을 위한 처리를 함
- nameSet : 두명의 플레이어 이름을 입력받음
- ranNum : 0에서 2까지의 범위에서 랜덤 변수 3개를 저장하는 배열을 생성 후 해당 숫자들의 동일여부를 확인 후 일치하면 승리 메시지 출력, 아니면 '아쉽군요' 메시지 출력
- startGame() : 배열의 수가 모두 일치하면 해당 턴의 플레이어에게 승리 처리를 하고 게임을 종료.
아니면 다음 플레이어에게 턴을 넘김

게임 알고리즘

Gambling으로 객체 생성 후

생성자에서 난수 생성을 위한 처리를 진행

게임 시작 메시지를 출력 후 두 플레이어의 이름을 입력받아 setName()으로 p배열 값을 설정함

이때 턴은 해당 배열에 값이 설정된 순서대로 진행된다

이름을 입력하고 Enter 키를 누르면 반복문이 실행된다.

ranNum으로 3개의 난수를 출력 후 생성된 3개의 숫자가 모두 동일하면 승리 처리를 통해 (플레이어 이름)님 승리!! 메시지 출력 후 게임 종료. 아니면 '아쉽군요' 메시지 출력 후 배열 순서(i)에 1을 더해 해당 순서의 플레이어에게 턴을 넘긴다.

승리자가 나올 때까지 이 반복문은 무한 실행된다.