

2021-1 C++ 프로그래밍 실습과제 04

학번	2020136129	이름	최수연
----	------------	----	-----

(1) 각 문제에 대한 분석과 및 해결 방법

1. 교재 190~196쪽의 코드를 참고하여 퍼즐 게임을 구현하라.

- (1) 게임은 교재와 같이 4x4로 구현해도 되고, 3x3이나 5x5등 다른 크기로 구현해도 됨

(2) 랭킹은 10위까지 관리되도록 구현하며, 랭킹 파일은 가급적 이진 파일로 저장할 것(이진 파일이 어려운 경우 텍스트 파일로 저장해도 됨)

(3) 점수 계산은 이동 횟수가 아니라 시간 기준으로 처리하도록 함. 즉, 조금이라도 빨리 해결한 사람이 더 높은 랭킹을 가짐

[문제분석 및 해결방법]

1. 헤더 파일

헤더 파일의 경우, 구조체 하나를 선언하여 선수 이름, 퍼즐 조각을 움직인 횟수, 경기 소요 시간을 선언한다. extern을 사용하여 랭킹 함수에서 loadRanking(), storeRanking(), printRanking(), addRanking()을 가져와, 본 헤더 파일을 이용하는 외부 파일에서 사용할 수 있도록 한다.

2. 랭킹 함수

랭킹은 10위까지 관리되어야 하므로 랭킹 배열의 크기를 10으로 설정한다. 함수는 총 5개 initRanking(), loadRanking(), storeRanking(), printRanking(), addRanking()로 구분한다. initRanking()은 랭킹을 초기화 하는 함수이고, loadRanking()에서 파일을 불러와 읽고, 만약 파일이 비어있다면 initRanking() 함수를 호출해 랭킹 초기값을 대입한다. 그렇지 않으면 파일 내용을 해당 배열 항목에 저장한다. storeRanking()에서는 현재 랭킹을 파일에 저장한다. 이때, 파일 저장은 이진 모드로 저장한다. printRanking() 함수를 사용하여 화면에 출력하고, addRanking() 함수를 통해 각 경기마다 소요 시간을 비교하여 소요 시간이 적을수록 높은 순위를 가지도록 처리한다.

3. 퍼즐 함수

퍼즐 함수에서는 퍼즐 게임 과정을 담고 있는 함수이다. 먼저 모든 함수와 데이터를 static으로 처리하여 외부에 보이지 않도록 하여 본 파일 내에서만 사용할 수 있도록 하였다. init() 함수를 통해 3x3퍼즐에 맞게 퍼즐 배열을 1~8까지 순서대로 정렬하고, display() 함수를 통해 퍼즐을 화면에 출력한다. 이동에 필요한 move() 함수는 자료형 bool을 사용하여 누르는 방향키에 따라 움직일 수 있으면 참, 불가능하면 거짓으로 반환한다. 이때 숫자가 움직일 때마다 이동 횟수를 증가시킨다. shuffle() 함수를 통해 퍼즐을 섞고 화면에 출력한다. 이때 Sleep()을 사용하여 퍼즐이 섞이는 과정이 보이도록 한다. isDone() 함수에서는 숫자가 1~8까지 모두 순서대로 들어가 있는지 확인하고 종료하는 함수이다. getDirKey() 함수에서 _getch()로 방향키가 입력되면 반응하도록 한다. 마지막 playPuzzle() 함수에서는 위의 모든 함수를 퍼즐 게임 알고리즘의 순서에 맞게 호출하여 3x3 퍼즐 게임이 실행되도록 한다. playPuzzle()을 통해 게임이 끝나면, 현재 게임의 이동 횟수와 소요 시간을 addRanking() 으로 보낸다.

4. 메인 함수

메인 함수에서는 랭킹 함수와 퍼즐 함수를 사용하여 전체적인 3x3 퍼즐 게임을 실행한다. loadRanking()와 storeRanking()에서는 "ranking.bin" 파일을 매개변수를 통해 전달한다. 먼저 loadRanking()을 호출하여 파일을 읽고 랭킹을 배열 각 항목에 맞춰 저장한다. 그리고 playPuzzle()을 실행하여 게임을 진행한다. 게임이 끝나면 printfRanking()을 통해 게임 결과를 화면에 출력하고, storeRanking()을 사용하여 랭킹을 저장한다.

(2) 자신이 구현한 주요 코드

```
if (tElap < MVP[nMVP - 1].tElapsed) {
    int pos = nMVP - 1;
    for (; pos > 0; pos--) {
        if (tElap >= MVP[pos - 1].tElapsed) break;
        MVP[pos] = MVP[pos - 1];
    }
    MVP[pos].nMove = nMove;
    MVP[pos].tElapsed = tElap;
    printf(" %d위입니다. 이름을 입력하세요: ", pos + 1);
    scanf("%s", MVP[pos].name);
}
```

playPuzzle()을 통해 전달된 소요 시간과 기존 랭킹에 있는 각 순위마다의 소요 시간을 각각 비교하여 랭킹을 갱신하는 코드이다. 만약 현재 게임의 소요 시간 tElap가 기존 랭킹 맨 마지막 순위의 소요 시간 MVP[nMVP - 1].tElapsed보다 작으면 마지막 순위에 현재 게임 데이터를 넣는다. 그리고 맨 끝 순위 pos부터 pos - 1과 하나씩 비교하여 소요 시간이 더 적은 것을 앞 순위로 둔다. 반복문이 종료되고 현재 게임의 순위를 출력한다. 이때 현재 게임을 나타낼 수 있는 이름을 입력받는다.

(3) 다양한 입력에 대한 테스트 결과

1. 교재 190~196쪽의 코드를 참고하여 퍼즐 게임을 구현하라.

Puzzle	Puzzle	Puzzle
<pre> 1 2 3 4 5 6 7 8 ----- 이동 횟수: 0 소요 시간: 0.1 [1위] 4 정구 2.9 [2위] 10 헬로키티 7.9 [3위] 16 비실이 11.6 [4위] 28 도라에몽 16.9 [5위] 36 뽀로로 24.1 [6위] 98 빵식이 58.3 [7위] 78 어린왕자 65.6 [8위] 138 펭수 76.8 [9위] 1000 C++좋아요 1000.0 [10위] 1000 C++좋아요 1000.0 퍼즐을 섞어주세요(엔터) ...</pre>	<pre> Puzzle ----- 2 6 8 1 3 7 4 5 ----- 이동 횟수: 50 소요 시간: 43.3 게이미 시작됩니다...</pre>	<pre> Puzzle ----- 1 2 3 4 5 6 7 8 ----- 이동 횟수: 52 소요 시간: 27.0 6위입니다. 이름을 입력하세요: 라이언 [1위] 4 정구 2.9 [2위] 10 헬로키티 7.9 [3위] 16 비실이 11.6 [4위] 28 도라에몽 16.9 [5위] 36 뽀로로 24.1 [6위] 52 라이언 27.0 [7위] 98 빵식이 58.3 [8위] 78 어린왕자 65.6 [9위] 138 펭수 76.8 [10위] 1000 C++좋아요 1000.0 C:\#2020136129 최수연#2020136129최수연_4장 이 창을 닫으려면 아무 키나 누르세요...</pre>

빵식사와 어린왕자를 살펴보면 빵식사가 이동 횟수가 더 많음에도 불구하고 소요 시간이 빵식사가 더 적으므로 어린왕자보다 높은 순위에 있다.

(4) 코드에 대한 설명 및 해당 문제에 대한 고찰

없음.

(5) 이번 과제에 대한 느낀점

이번 강의와 과제를 통해 파일 입출력에 대한 내용을 제대로 정리할 수 있어서 좋았다. 사실 이 퍼즐 게임은 살면서 거의 한 번도 제대로 해본 적 없는 게임이라 4x4 퍼즐을 푸는 데 이동 횟수가 500회가 넘어가도 계속 풀지 못했다. 그래서 게임 난이도를 줄이기 위해 퍼즐을 3x3으로 바꾸었고, shuffle 횟수도 100회가 아닌 50회만 돌리고 실행되도록 설정하였다. 이번 과제를 하면서 처음에는 퍼즐 게임을 못하다보니 재미도 없고 힘들었다. 그러나 과제 때문에 게임을 계속 하다 보니 퍼즐 게임 실력이 늘어서 퍼즐 게임에 대한 흥미를 재대

로 느낄 수 있었다. 이러한 게임을 C언어를 통해 직접 구현하여 해보니 게임이 더 재미있다고 느껴졌다. 나중에 졸업했을 때 코딩 실력이 많이 향상되어 내가 원하는 프로그램을 마음껏 구현해낼 수 있었으면 좋겠다.

(6) 궁금한 점이나 건의사항

딱히 없습니다.