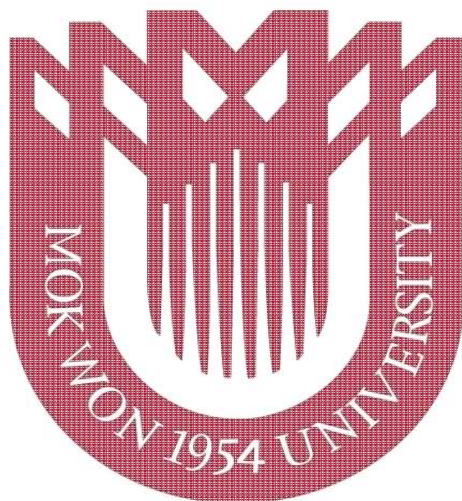


객체지향기초

【중간고사 보고서】



「 4조 」

1760031

1960025

2160009

2260006

서장일

양승현

명승훈

김상주

팀장

■ 프로젝트 목적

- # 퍼즐 게임 기능 확장
- # 기존 4 x 4 퍼즐 게임을 이용하여 제시된 6가지의 기능을 확장한다.

■ 프로젝트 개발

- # 4조
- # 김상주(팀장)
- # 서장일
- # 양승현
- # 명승훈

■ 제시된 4 x 4 퍼즐 게임 알고리즘

- # 4x4 퍼즐에는 16개의 자리가 있는데, 1부터 15까지의 퍼즐 조각들을 채우고, 나머지 한자리는 비워둔다.
- # 빈자리를 이용해 퍼즐 조각들을 움직여 모든 조각이 제자리를 찾도록 맞추는 게임
- # 이 게임을 15 퍼즐(Fifteen Puzzle) 게임이라고도 함
- # 최초에는 퍼즐이 모두 제 위치에 있어야 함
- # 엔터를 누르면 퍼즐 조각을 섞는다. (난수 발생)
- # 퍼즐이 섞이는 과정을 화면에서 볼 수 있도록 출력에 시간 지연
- # 힌트 : <Windows.h>에 선언된 Sleep() 함수 사용
- # 퍼즐이 섞이면 엔터를 눌러 게임을 시작
- # 퍼즐 조각의 이동에는 화살표 키 사용
- # 키를 누르면 바로 반응하도록 함
- # 모든 퍼즐 조각들이 제자리를 찾으면 게임이 종료
- # 퍼즐 조각을 움직인 전체 횟수, 걸린 시간을 계산
- # 이동 횟수를 바탕으로 랭킹을 갱신
- # 현재 랭킹을 화면에 출력
- # 랭킹은 파일에 저장되어야 함

■ 확장 기능

1. 퍼즐 맵의 크기 확장, 게임을 3 x 3과 5 x 5로 확장하라.
2. 리플레이(Replay) 기능, 경기가 끝나고 나면 게이머가 움직인 내용들을 다시 순서대로 보여준다. 이를 위해, 최초의 섞인 퍼즐 맵을 저장해두어야 하며, 게이머의 움직임들도 모두 저장해야 한다.
3. 게임 저장 기능, 경기 중에 현재 게임 상태를 파일에 저장하고 경기를 종료하는 기능을 구현하라. 물론 현재까지 퍼즐 조각을 움직인 횟수나 시각 정보도 저장되어야 한다.
4. 저장된 게임 이어가기 기능, 저장된 경기를 계속 이어서 진행할 수 있는 기능을 구현하라. 물론 시작 시각과 중간에 종료한 시각도 저장되어야 한다.
5. 그림 퍼즐 기능, 퍼즐 조각에 숫자가 아니라 문자가 출력되도록 해 보라. 이를 위해 1장을 참조해 간단한 아스키 아트로 4 x 4 그림을 만들어야 할 것이다.

■ 프로젝트 팀원별 역할 분담

1.	양승현	<pre>static void f_GameInit(); # 게임 초기화 함수 static void f_GameDisplay(); # 맵과 이동 횟수, 소요시간 등 게임 상황을 화면으로 출력 함수 { # 알파벳 퍼즐(5번 문항) 출력 }</pre>
2.	서장일	<pre>static int f_ArrowGetKeyDirection(); # 방향키를 받아들이는 함수 { # 입력된 키가 「Enter」라면, break # 입력된 키가 「A」라면, 이어 하기(확장 기능) }</pre>
3.	양승현	<pre>static void f_PuzzleShuffle(int nShuffleCount); # 퍼즐 조각을 nShuffleCount번 이동해 섞는 함수 { static bool f_PuzzleArrowMove(int nArrowDirection); # 퍼즐 이동 함수 }</pre>
4.	명승훈	<pre>static bool f_PuzzelsDone(); # 퍼즐이 다 맞춰졌는지 검사하는 함수 { static int f_ArrowGetKeyDirection(); # 방향키를 받아들이는 함수 { # 입력된 키가 「방향키」라면, 이동 함수 # 입력된 키가 「S」라면, 저장하고 게임 종료 } static bool f_PuzzleArrowMove(int nArrowDirection); # 퍼즐 이동 함수 + # 저장 함수(확장 기능) }</pre>
5.	김상주	<pre># 프로젝트 리드(역할 분배 및 진행도 확인) # # 저장 방법에 대한 알고리즘</pre>

■ 브레인 스토밍 (제시된 아이디어)

1. 단순 파일 처리로 프로그램의 모든 데이터를 저장한다. (퍼즐 맵, 게이머의 움직임, 퍼즐 조각을 움직인 횟수, 시각 정보 등을 모두 저장하여 불러오기)
2. 3차원 배열[PAGE][ROW][COL] 을 사용하여 배열을 저장하고 PAGE에 따라 플레이어의 움직임을 저장하고, 게임 저장, 이어가기 기능을 구현한다.
3. 기본적으로 파일 처리를 이용하여 데이터를 저장하나, 1번의 방식과 다르게 퍼즐의 크기(DIM)를 기준으로 데이터를 나누고 추가하거나 출력을 가능하게 한다. 이를 이용하여 제시된 확장기능을 수행한다.
4. 제시된 문제에서 공통점인 플레이어의 움직임을 기록하여 기존 프로그램을 응용 확장기능을 수행한다.
5. 최초 섞인 퍼즐 파일을 저장하고, 플레이어의 움직임(방향키)을 기록하여 리플레이 기능을 확장한다. 이를 기반으로 게임 저장, 불러오기 & 이어가기 기능을 확장할 수 있다.

■ 문제 해결을 위한 알고리즘

1. 퍼즐 맵의 크기 확장 : 게임을 3 x 3과 5 x 5로 확장하라.

define 되어 있는 DIM의 숫자를 변경함에 따라 3 x 3 혹은 5 x 5로 확장할 수 있다.

2. 리플레이 기능 : 경기가 끝나고 나면 게이머가 움직인 내용들을 다시 순서대로 보여준다. 이를 위해, 최초의 섞인 퍼즐 맵을 저장해두어야 하며, 게이머의 움직임도 모두 저장해야 한다.

최초의 섞인 퍼즐을 저장하려면 파일 처리를 통하여 퍼즐이 섞이고 난후 "퍼즐의 배열을 저장하는 함수 (f.ShufflePuzzleMapBackUp)"를 이용하여 .txt 파일로 저장한다. 플레이어의 움직임을 저장하려면, 방향키가 입력될때마다 숫자로 치환된 키값을 정수형태로 파일에 저장하는 함수를 이용하여 파일로 저장한다.

프로그램의 시작이 SHUFFLECOUNT의 횟수만큼 퍼즐이 움직인 후 nMoveCount를 0으로 초기화된다. 이후 nMoveCount가 증가할 때마다 nMoveCount(퍼즐을 움직인 횟수)와 2차원 배열로 되어있는 퍼즐을 저장하여 게임이 끝난 이후 nMoveCount의 첫 번째부터 마지막 수 까지 대응되는 배열을 출력하여 리플레이 기능을 제작한다. 이후 정상적으로 게임이 종료된다면(저장 및 종료가 아닌 경우) 저장되어 있는 파일을 삭제하여 다음 퍼즐 게임의 데이터 오류를 보완한다.

3. 게임 저장 기능 : 경기 중에 현재 게임 상태를 파일에 저장하고 경기를 종료하는 기능을 구현하라. 물론 현재까지 퍼즐 조각을 움직인 횟수나 시각 정보도 저장되어야 한다.

SHUFFLECOUNT의 횟수만큼 퍼즐이 움직인 후 nMoveCount를 0으로 초기화, 이후 nMoveCount가 증가할 때마다 nMoveCount와 2차원 배열을 저장한다. 저장된 파일은 다음 게임 시작시 이어가기 기능을 사용하지 않으면 초기화가 된다.

4. 저장된 게임 이어가기 : 저장된 경기를 계속 이어서 진행할 수 있는 기능을 구현하라. 물론 시작 시각과 중간에 종료한 시각도 저장 되어야한다.

만약 저장되어있는 파일이 있다면, 만약 저장된 파일이 남아있다면 이어가기를 진행할 것인지 물어본 후 만약 진행하겠다고 하면 가장 최근 2차원 배열을 불러오기 한 후 게임을 진행한다, 이후 리플레이 기능까지 수행, 만약 진행하지 않겠다고 하면 이전 파일을 삭제한 후 처음부터 시작

5. 그림 퍼즐 기능 : 퍼즐 조각에 숫자가 아니라 문자가 출력되도록 해보라. 이를 위해 1장을 참조해 간단한 아스키 아트로 4x4 그림을 만들어야 할 것이다.

그림퍼즐(GameDisplay 에서 수정) 배열에 +64를 추가하여 대응되는 아스키코드값을 출력하면 알파벳 순서대로 출력된다.

이중 반복문으로 정수형 데이터가 들어있는 배열을 정수 데이터가 아닌 문자형 알파벳으로 출력한다면 간단히 알파벳의 순서를 맞추는 그림 퍼즐을 구현할 수 있다.

6. 4.7절의 랭킹 보드 프로그램을 참고하여 상위 10개의 게임을 랭킹 파일에 저장하라.

define 되어 있는 NUMBER_MVP를 5에서 10으로 변경하면 1위부터 10위까지의 랭킹을 저장할 수 있다.

■ 분석 및 설계

주어진 기본 함수

f_PlayFifteenPuzzle()
4x4 퍼즐 게임 주기능 함수
프로그램이 시작되면 랭킹 파일을 읽어 현재 랭킹을 초기화
프로그램이 종료되기 직전에 랭킹을 파일에 저장
엔터를 누르면 퍼즐이 100번 섞임
종료 시간이 될 때까지 게임을 진행하고, 종료되면 랭킹을 등록함
f_GameInit()
퍼즐 맵의 1 ~ 15번 위치에 해당 숫자를 저장
마지막 위치에는 빈칸을 나타내는 0을 저장
빈자리 (x, y)의 초기값은 (3, 3)
시작 시각과 이동 횟수를 초기화
f_GameDisplay()
맵과 이동 횟수, 소요시간 등 게임 상황을 화면으로 출력
f_PuzzleArrowMove(int nArrowDirection)
퍼즐 이동 함수
상하좌우 방향키를 입력받고, 그 방향으로 퍼즐 조각을 움직일 수 있는지를 검사
(맵 왼쪽 열인지 검사, 맵 오른쪽 열인지 검사, 맵 아래행인지 검사, 맵 윗행인지 검사)
움직일수 있는 방향이면 퍼즐 조각을 움직이고, 빈칸의 위치를 갱신하며, 이동 횟수를 증가
성공하면 true를 반환하고, 이동 불가능 하면 false를 반환
f_PuzzleShuffle(int nShuffleCount)
퍼즐 조각을 섞는 횟수를 입력받아 무작위로 빈칸을 움직여 퍼즐을 섞는 함수
난수(0~3)를 발생하여 상하좌우 임의의 방향으로 빈칸을 이동
이동을 위해서는 f_PuzzleArrowMove(int nArrowDirection) 함수를 사용
퍼즐을 섞는 횟수는 매개변수로 전달받고, 이 횟수만큼 빈칸을 이동해 퍼즐을 섞는다.
Sleep() 함수를 이용해 시간을 지연시켜 퍼즐이 섞이는 과정을 볼 수 있도록 한다.
f_PuzzlesDone()
퍼즐이 다 맞추어졌는지를 검사하는 함수
모든 조각이 제자리에 있고 마지막에 공백이 있으면 게임 종료
f_ArrowGetKeyDirection()
방향키를 받아들이는 함수
getch() 함수를 사용하여 키가 입력되면 바로 반응하는 함수
up : 224, 72
down : 224, 80
left : 224, 75
right : 224, 77

■ 프로젝트 프로그램 명칭

프로그램 명칭 : FifteenPuzzleExpansion(외 다수1)

헤더 파일 : h_FifteenPuzzle.h / h_Ranking.h

메인 함수 : MainFifteenPuzzle.cpp
기능 함수 : f_FifteenPuzzle.cpp / f_RankingProcess.cpp

■ 소스 코드

별도 첨부

■ 프로젝트 결과

1 ~ 6번중 4번을 제외하고는 구현했으나 4번 문제는 부분적으로 해결되었으나 완성하지 못하였습니다.

김상주(팀장)

프로젝트 목적인 4 x 4 퍼즐게임 기능 확장을 완벽히 수행하지 못하였습니다. 이러한 점은 팀장인 저에게 프로젝트를 올바른 길로 리드하지 못한 책임이 있는 것 같습니다. 팀장으로서 프로젝트를 위해 모인 초기에 문제 해결을 위한 접근 방법을 어렵게 생각하고 설정하여 많은 시간을 코드 분석과 프로그래밍에 소비하게 하였고 이는 많은 버전의 코드들이 탄생하게 된 배경이 되었습니다. 그로 인하여 팀원들은 혼란스러워하였고 이러한 결과가 모여 프로젝트가 제대로 수행되지 못하였다고 생각합니다. 비록 이번 프로젝트는 실패하였으나 저희 팀은 이 프로젝트로 다양한 코드 기법들을 배우고 또 많은 오류와 싸우며 디버깅을 통하여 성장하는 기회를 가질 수 있었다고 생각합니다.

명승훈

이번 프로젝트를 하면서 이거 내가 할 수 있을까? 라는 생각이 먼저 들었지만 팀장님이 우리를 잘 이끌어주었기에 할 수 있었으며 내가 직접 코딩을 하면서 프로그램이 어떤 알고리즘으로 돌아가는지와 내 프로그램과 다른 사람의 프로그램을 합칠 때 주의 할 점도 배웠던거 같다. 또 팀원들 모두 프로젝트 참여에 우호적이어서 이상 없이 프로젝트를 할 수 있었던거 같다.

서장일

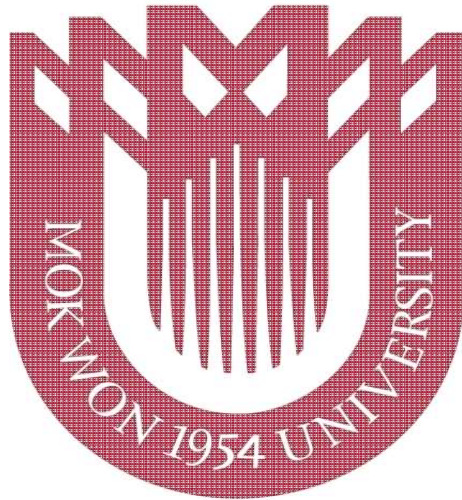
길다면 길고 짧다면 짧은 2주간의 중간 프로젝트를 진행하면서 지치고 힘든 부분이 많았습니다. 처음 1주차에는 주어진 형식의 코드를 이해하는데 많은 시간을 써서 시간이 부족했다면 2주차에는 간간히 발생하는 오류부터 컴파일 오류가 발견되지 않더라도 정확한 값을 받아오지 못하는 오류가 발생해서 골머리를 앓았습니다. 저희조는 디스코드로 새벽까지 코딩 작업을 하곤했는데 사기가 떨어지고 힘들어하는 모습이 보일때마다 팀장인 김상주가 사기를 북돋아주었고, 팀원들이 해결되지 않는 부분은 힌트를 줘서 팀원들 스스로 해결할 수 있게끔 도움을 줘서 아직 부족한 부분이 많지만 그래도 어느 정도 얻어가는 부분이 있다고 생각해서 좋았습니다. 팀장이 각자의 역량에 맞춰 역할분담을 해준 덕분에 여기까지 올 수 있었다고 생각합니다. 끝으로 팀 프로젝트를 진행함으로써 교수님께서 분업을 강조하신 이유를 알 것 같습니다

양승현

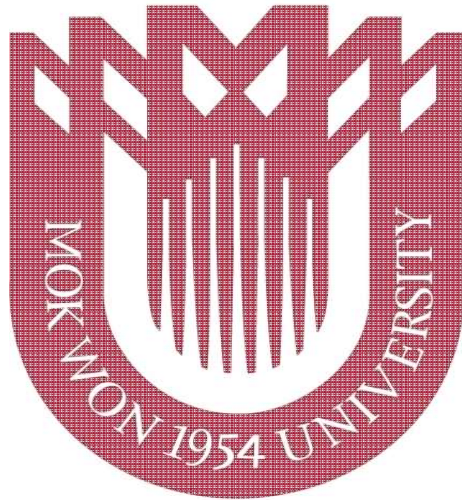
1) 다양한 방법으로 접근하여 문제 해결을 시도하여 접근한 방법의 모든 소스코드를 보고서에 첨부하였습니다.



"FifteenPuzzleExpansion" 소스 코드



"Project_FifteenPuzzle" 소스 코드



"Project_FifteenPuzzle2" 소스 코드