**STL 보고서**

**STL 월화 23/56**

**학번:** **2014182011**

**이름:**  **김태화**

**목 차**

1. **프로그램에 대한 설명....................................... 1**

1) 조작 방법

2) 게임 목표

3) 메뉴

1. **과제 해결방법................................................... 3**

1) 리플레이를 만들 게임 수정

2) 컨테이너 선택과 클래스 설계

3) 파일 입출력

**3. 과제를 마치고 느낀 점 .................................... 4**

1) Unordered\_Multimap은 Hash Table로 구성되어 저장하는

순서가 정해져 있지 않다.

2) Iterator를 사용하면서 루프를 돌릴 때 조건을 잘못 주거나,

삽입 삭제 동작으로 반복자를 무효화 시킬 수가 있다.

1. **프로그램에 대한 설명**

1) 조작 방법

① 마우스 클릭 후 드래그를 하여 블록들이 움직일 방향을 결정한다.

② 키보드 방향키를 사용해 블록들이 움직일 방향을 결정한다.

③ 메뉴를 사용하여 새 게임, 게임 종료, 목표점수 변경, 리플레이 저장,

리플레이 시작을 할 수 있다.

2) 게임 목표

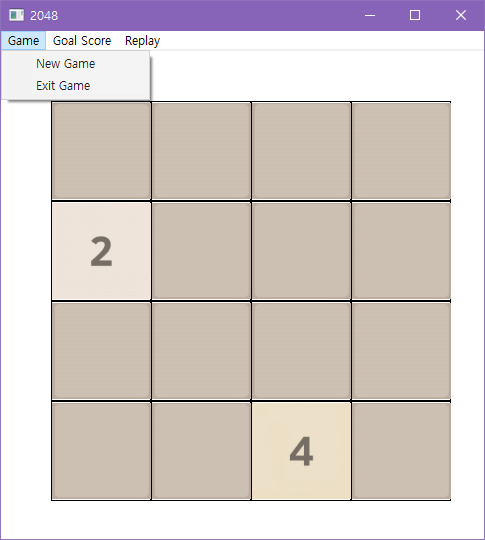
- 상하좌우로 블록들을 움직여 2048이라는 블록을 만드는 것이 승리조건이다.

- 목표 블록은 바꿀 수 있고 더 이상 블록이 움직일 수 없을 때 게임을

패배하게 된다.

3) 메뉴

① Game

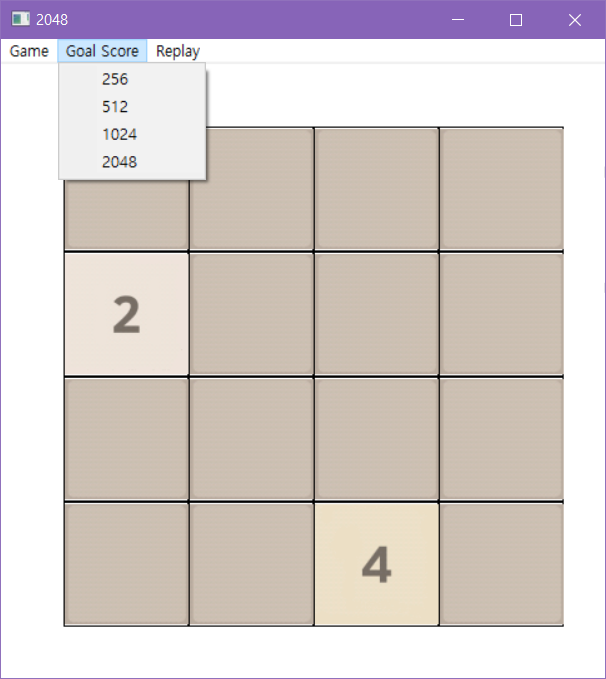


- New Game: 게임을 다시 시작한다.

- Exit Game: 게임을 종료한다.

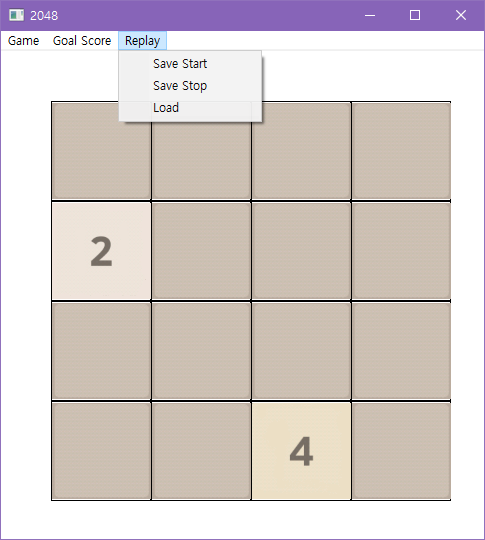
-1-

② Goal Score



- 도달해야 하는 목표 블록(게임 승리 조건)을 각 숫자로 변경할 수 있다.

③ Replay



- Save Start: 플레이어의 입력들을 저장해 놓는다.

- Save Stop: 저장한 입력 값들을 리플레이 파일에 저장한다.

- Load: 리플레이 파일을 읽어와 리플레이를 실행한다.

-2-

**2. 과제 해결방법**

1) 리플레이를 만들 게임 수정

- 리플레이를 제작하기에 앞서 일단 무슨 게임을 선택할 것인지 고민을

시작했다. 그러던 중 윈도우프로그래밍 시간에 만들었던 2048이라는

게임을 찾아서 사용하기로 하였다. 조작방법이 마우스를 이용한 방법밖에

없어서 방향키를 이용한 조작을 추가하였다. 그리고 리플레이 메뉴를

추가하여 리플레이의 저장시작과 종료, 리플레이 시작을 할 수 있게

하였다.

2) 컨테이너 선택과 클래스 설계

- 과제를 시작하기에 앞서 무엇을 저장해야 하는지 생각을 하다가

게임판의 상태, 키 ∙ 마우스 입력, 입력과 입력사이의 시간을 저장하면

될 것이라 생각해서 중복 값을 허용하고 키, 값을 가지고, 정렬을

하지 않는 Unordered multimap을 사용하려고 했다.

- 하지만 이 컨테이너는 해쉬 테이블로 구성되어 저장되어 있는 순서가

삽입한 순서대로 접근을 못하고, 랜덤으로 생성되는 블록들도 저장을

해야 해서 컨테이너를 list로 변경하고, 저장해야 할 값들을 담을

Replay클래스를 만들었다.

- list를 사용한 이유는 list는 데이터를 삽입 및 삭제할 때의 속도가

빠르고, vector는 크기가 필요한 것보다 더 많이 할당하게 되어 필요하지

않지만 할당 되어있는 빈 공간들이 생길 수 있어서 list를 사용하기로

하였다.

3) 파일 입출력

- 리플레이 저장을 시작하게 되면 그 시점의 게임판의 정보를 데이터

파일에 저장하게 된다. 그러고 난 후엔 시간을 재기 시작해서 다음

입력이 있을 경우에 입력과 입력 사이의 시간을 재고 어떤 입력인지,

생성된 블록의 정보를 담아 데이터 파일에 저장하였다.

-3-

**3. 과제를 마치고 느낀 점**

1) Unordered Multimap은 Hash Table로 구성되어

저장하는 순서가 정해져 있지 않다.

- 처음 리플레이 데이터를 저장할 때 키(방향 입력), 값(입력 간의 시간)만

함께 기록해두면 될 줄 알고 Unordered Multimap에 같이 삽입을

하였다. 그리고 나선 데이터가 의도한대로 들어가 있는지 확인을 해보려

했는데 저장한 순서대로 데이터가 나오지도 않고, 중복된 값들끼리 같이

모여 있는 것을 확인하게 되었다.

2) Iterator를 사용하면서 루프를 돌릴 때 조건을 잘못

주거나, 삽입 삭제 동작으로 반복자를 무효화 시킬 수가

있다.

- 리플레이 기능을 다 구현하고 나서 디버깅을 하는 과정에서 리플레이를

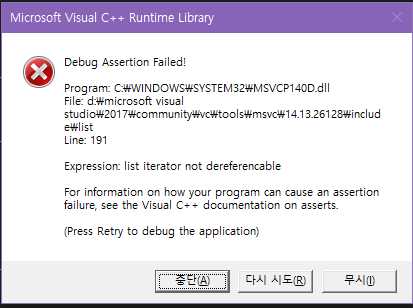
하고 있는 걸 시각적으로 좀 더 표현하고 싶어서 점수 판 옆에 리플레이

데이터를 출력하도록 하였다. 이를 수정하는 과정에서 iterator의 값을

변경을 했었다. 그러고 나서 디버그 모드를 사용해 빌드를 하였더니

아래의 사진 같은 오류가 발생하였다. 이를 검색을 통해 찾아보니

Iterator를 잘 못 사용하게 되면 이러한 문제들이 발생을 한다고 한다.



-4-