**Land-mine Maze Game**

2021.06.21

컴퓨터공학설계 및 실험1

20190018 김유이

**Project 목표**

Land-mine Maze Game을 구현하고, 게임 결과에 따른 랭킹을 부여하기 위한 ranking system을 구현한다.

수업 시간에 구현한 미로 파일을 읽는 함수와 이를 바탕으로 미로를 그리는 함수를 이용할 수 있는 게임을 구현한다. Land-mine Maze 게임은 미로 내부에 랜덤으로 지뢰가 생성되고 제한 시간 내에 미로의 출구를 찾는 게임이다. 만약, 출구에 도달하거나 절반이상의 지뢰를 밟거나 시간이 초과되면 게임은 종료된다.

모서리는 + 가로벽은 - 세로벽은 |로 그려진 불완전 미로가 있는 maze.maz 파일을 입력 파일로 한다. openframeworks에서 새로운 프로젝트를 생성한 다음에, src에 src 폴더 내에 있는 소스코드들을 추가한다. 프로그램을 실행한 후에 상단의 File을 눌러 bin/data 경로에 있는 maze0.maz 파일을 선택하면 불완전 미로가 그려진다.

또한, 랭킹 시스템을 구현하여, 게임이 종료된 후에는 밟은 지뢰의 개수와 실패, 성공 여부를 합산하여 점수를 계산하고 이를 토대로 사용자들을 순위에 따라 볼 수 있다.

**자료구조 및 알고리즘**

1. Land-mine

:Pair<int, int>로 이루어진 vector를 사용하였다. 불완전 미로 내부의 좌표 중에서 랜덤으로 지뢰의 좌표를 선택한다. 이 정수 좌표들을 pair로 묶어 vector에 넣는다. 이후, draw 함수 내부에서 지뢰를 그린다.

1. Ranking system

Linked list를 이용하여 각 노드에 사용자 이름과 점수를 저장하는 방식으로 구현하였다.

미로 내부에 지뢰를 만들 때, 미로 내부 중에서 랜덤으로 지뢰의 위치를 정하고 Direction array를 만들어서 상하좌우 방향 탐색을 하며 해당 위치가 벽인지 아닌지 구분하는 데에 이용하였다. ranking system 정렬하면서 삽입했다. linkedlist의 맨 앞에서부터, 제일 큰 배열 값을 찾아간다. 만약 더 큰 값을 찾은 경우에는 멈춰서 이전 값과 큰 값 사이에 새로운 랭킹을 삽입한다. 타이머 구현 시에 clock() 함수와 difftime() 함수를 사용하여 구현하였다.

**각 함수에 대한 설명**

Makemine()

불완전 미로 내부에 랜덤으로 지뢰의 좌표를 만드는 함수이다. 이 함수 내부에서는 게임의 초기 셋팅이 이루어진다. 따라서, 제한 시간도 이 함수 내부에서 MIN macro를 이용해서 셋팅되고, 밟을 수 있는 지뢰 개수의 한계, 즉 게임에서 ‘life’개수도 정해진다.

Draw()

S 키 입력시, 제한 시간 내일 경우 동안, 좌표가 지뢰를 밟는지 아닌지 계산하여 사용자의 위치를 옮겨 그린다.

사용자가 상하좌우 키를 입력함에 따라 사용자의 위치를 점으로 그린다. 만약, 사용자가 이동한 위치가 지뢰일 경우, 밟은 지뢰 계수(trie)를 카운트한다. 만약, 밟은 지뢰 개수가 life 보다 작으면 처음부터 다시 시작하고, life를 초과이면 게임 오버된다. 출구를 찾은 경우 역시 게임이 끝나는데 이 역시 draw 함수 내부에서 계산한다.

Result()

점수를 계산하고 결과를 출력하는 함수이다. 만약, success flag가 세워진 경우에는 score를 0으로 계산한다. 그렇지 않은 경우에는 실패한 것이므로 밟은 지뢰 개수\*5로 점수를 계산한다. 실패했지만, 밟은 지뢰개수가 0일 경우에는 3점을 부여한다.

Rank()

랭킹대로 사용자 정렬이 되어 있는 Linked list의 사용자 이름과 점수를 순위에 따라 출력한다.

newRank()

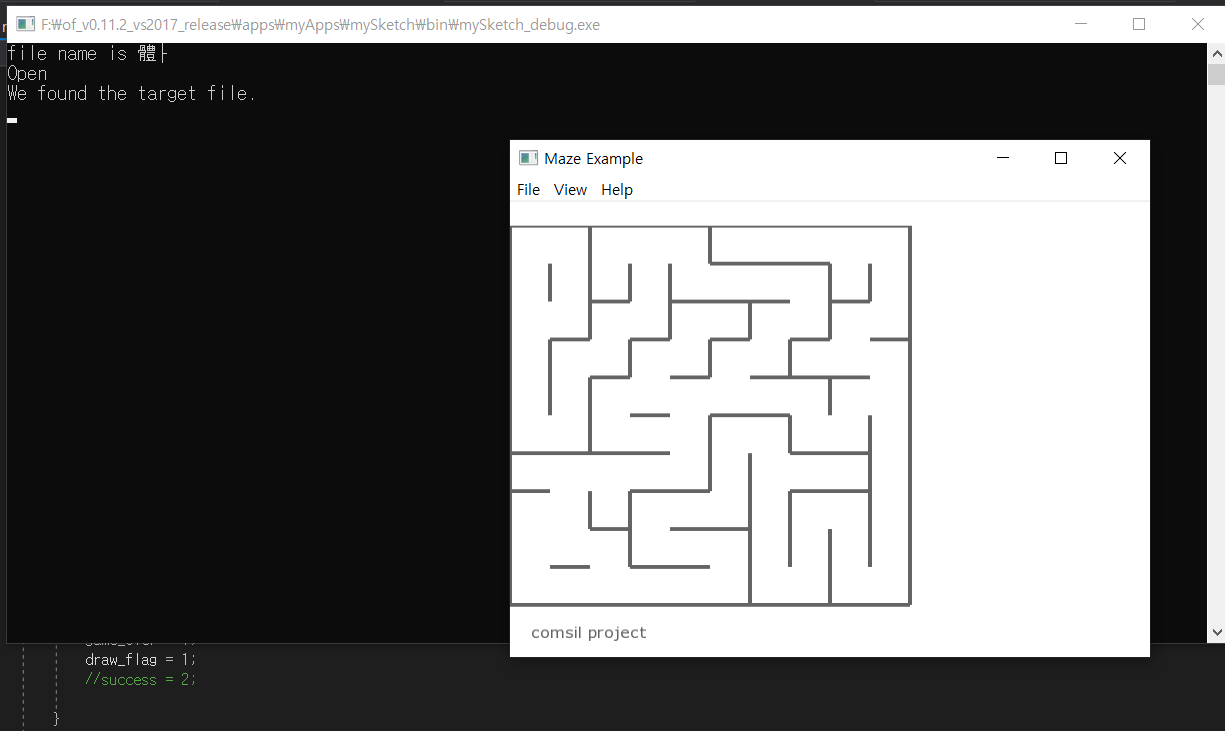
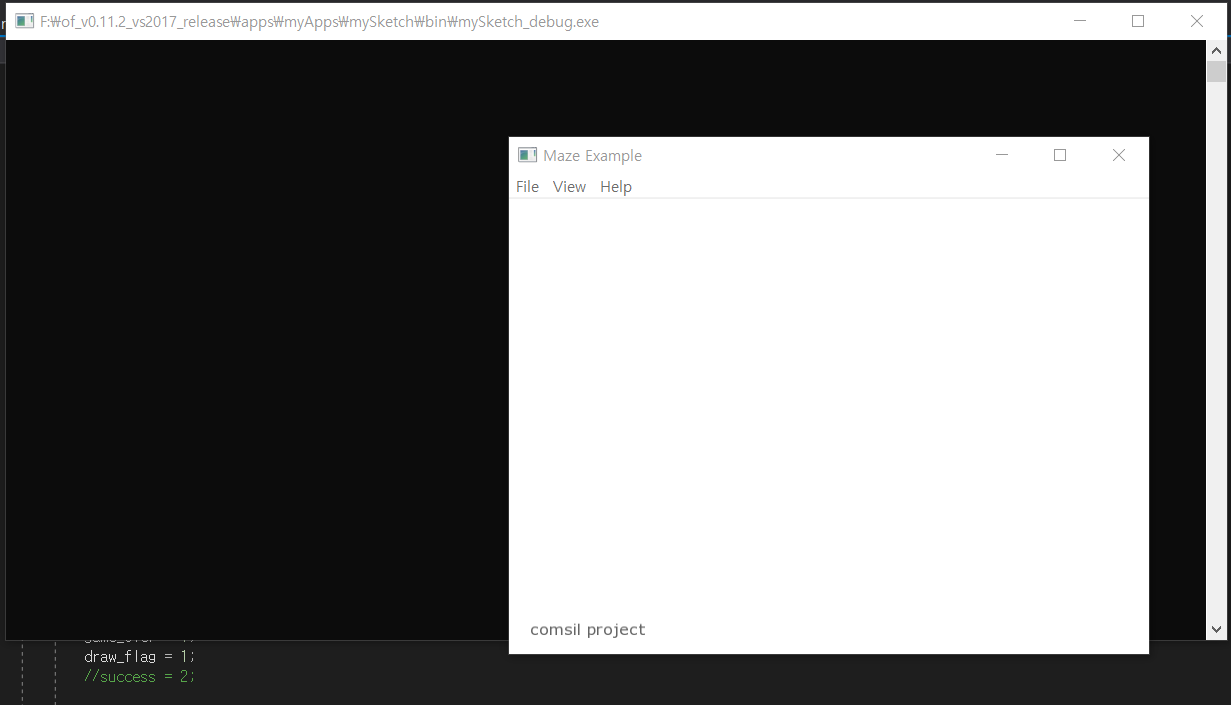
새로운 사용자의 점수를 linked list 에 있는 사용자의 점수와 비교하여 오름차순으로 정렬하여 새로운 사용자의 이름, 점수를 linked list에 삽입한다.

**창의적으로 구현한 부분**

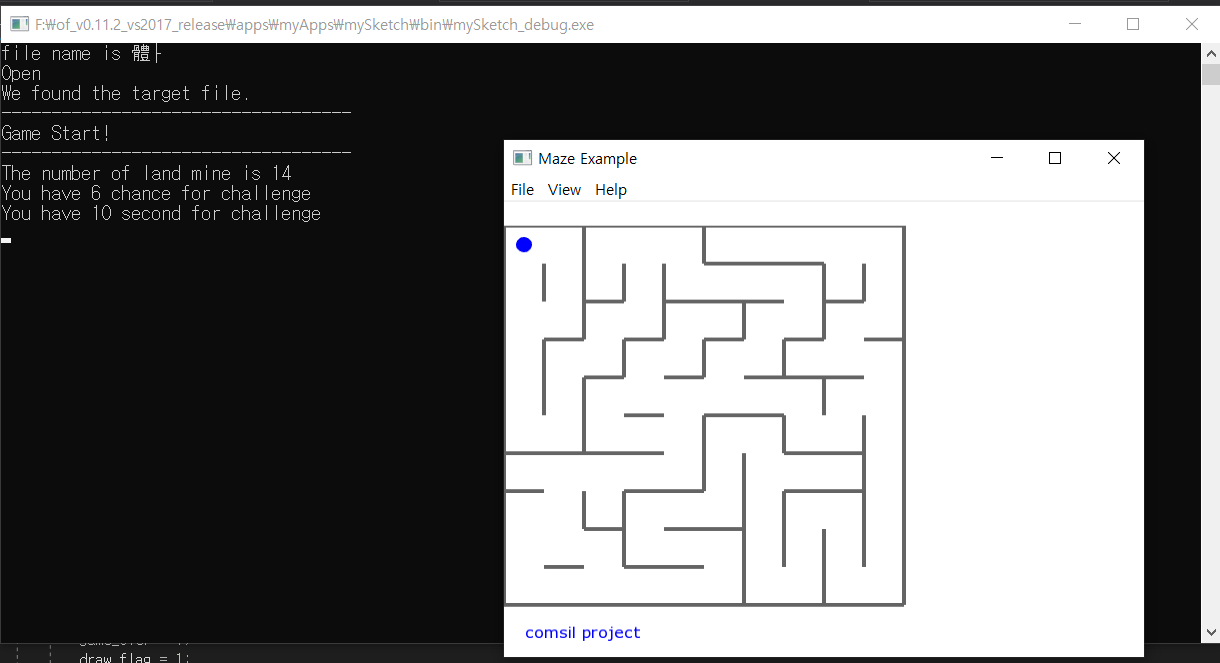
1. 정하고 Direction array를 만들어서 상하좌우 방향 탐색을 하며 0미로내에서 사용자를 움직일 수 있게 한 것
2. Vector와 direction array, clock() 을 이용해 랜덤으로 지뢰를 만들고 제한시간을 주는 게임을 구현한 것
3. 이 지뢰게임의 점수를 계산하기 위해서 링크드 리스트와 삽입정렬을 이용한 랭킹 시스템을 더한 것

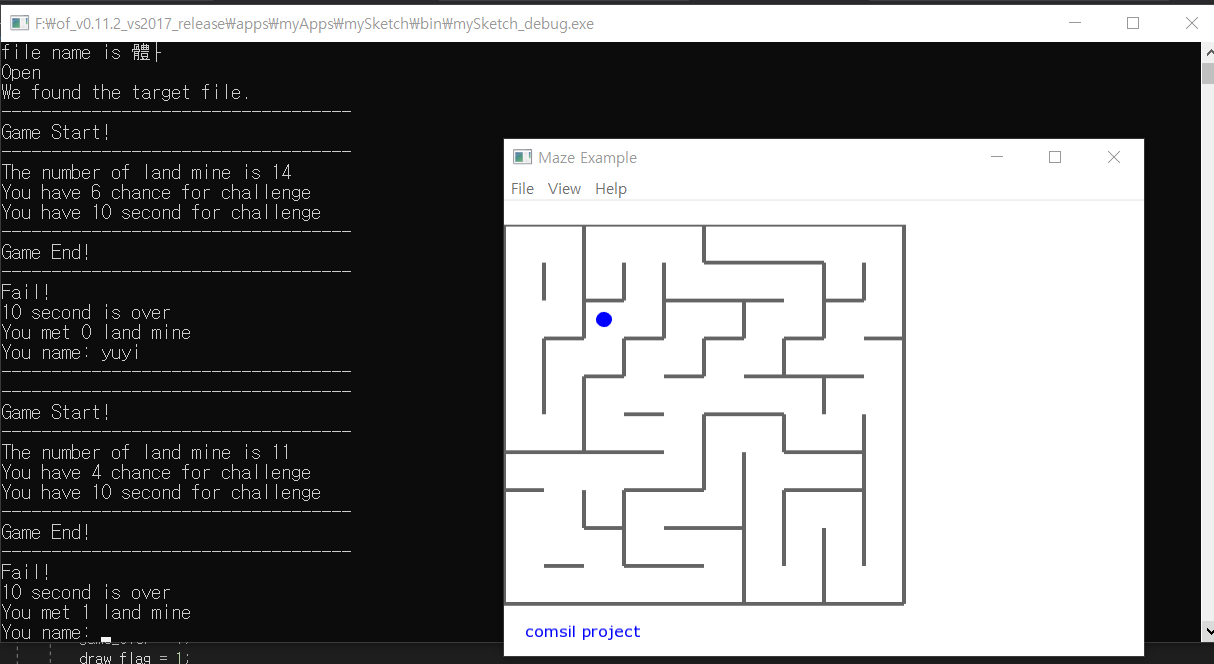
**프로젝트 실행 결과**

미로 그리기: File을 누르고 입력 파일을 선택한다.

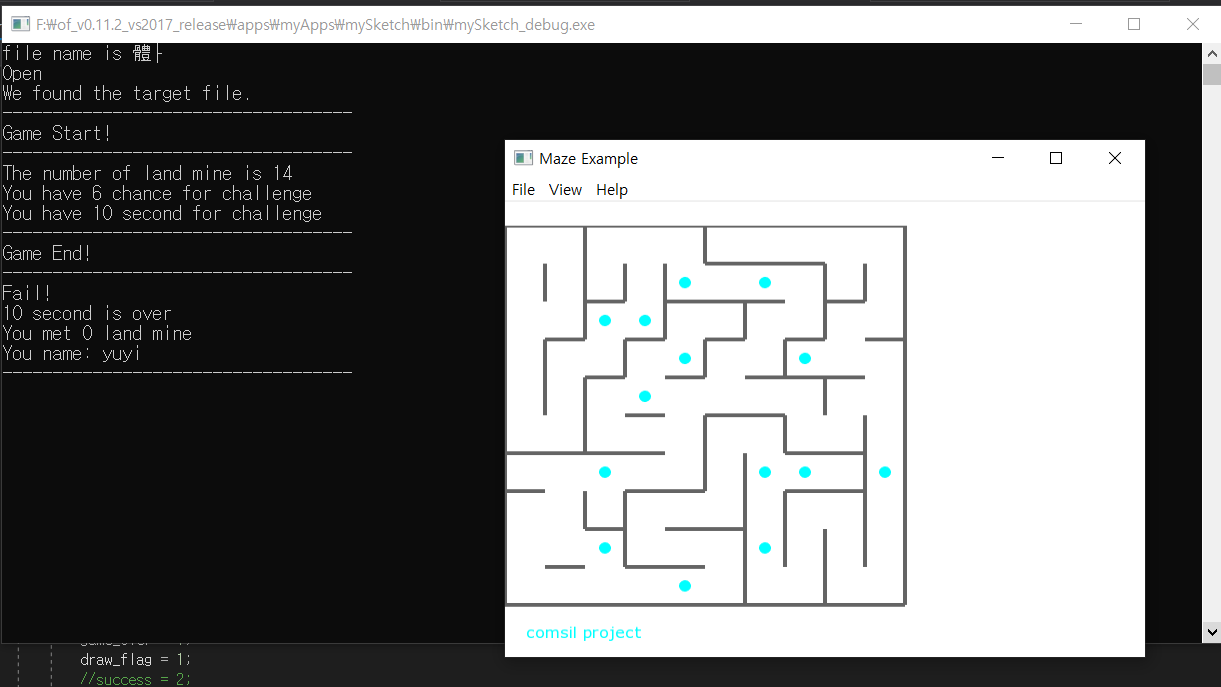


‘s’키를 눌러 게임을 시작한다. 지뢰와 타이머가 셋팅된다. 상하좌우 키를 눌러서 사용자 위치를 이동할 수 있다.





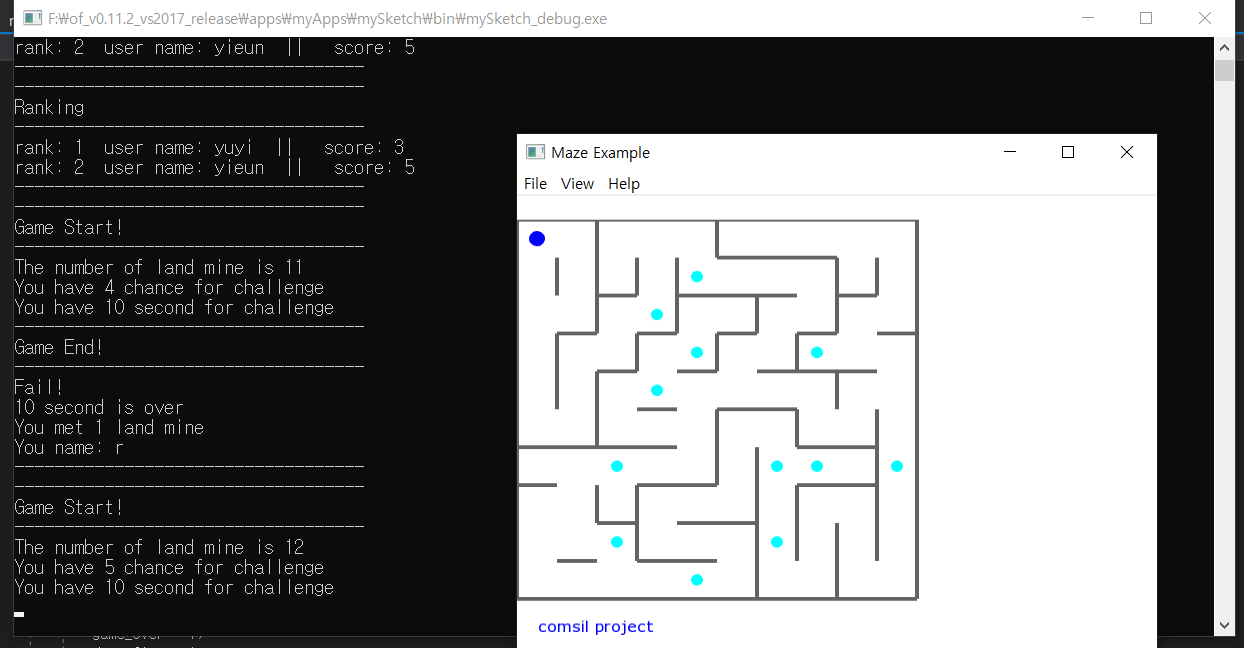
게임 오버된 경우에는 사용자 이름을 입력하라는 문구가 뜨고, 소멸되기 이전 처음 지뢰의 위치가 그려져서 사용자가 확인할 수 있도록 한다.



‘r’키를 눌러서 랭킹 시스템을 출력할 수 있다.

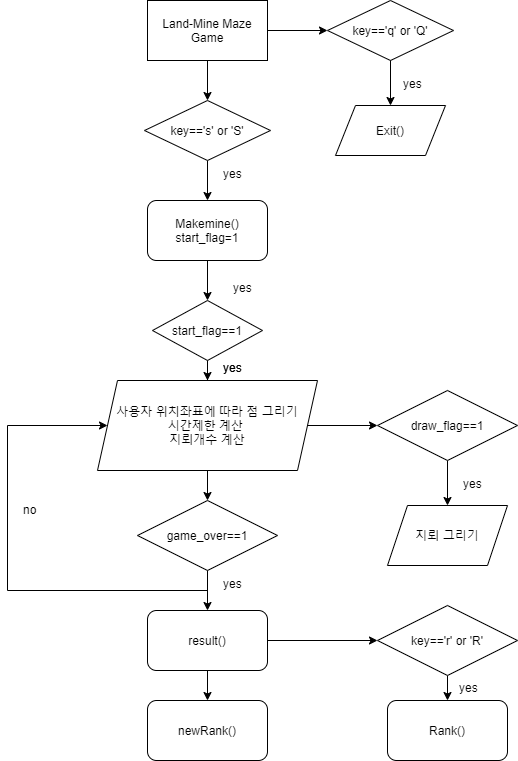


게임 진행시에는 사용하지 않아야 하는 키지만, 임시로 ‘d’ 키를 눌러서 지뢰의 위치를 확인할 수 있도록 해 두었다.



**추가자료**

**-Alogrithm diagram**

****

**느낀 점 및 개선 사항**

사용자 위치를 이동시키고 타이머와 지뢰를 구현해서 직접 간단한 게임을 구현해보고, 수업 시간에 배운 랭킹 시스템 역시 활용했다. 지뢰 찾기에서 아이디어를 얻어 미로 게임과 합쳐서 구현하였다. 테트리스 때 매뉴얼 대로 게임을 일부 구현해봤는데, 비록 엄청 정밀하지는 않지만, 이번에는 배운 것을 이용해 스스로 게임 프로그램을 구상하고 구현해보았다는 점에서 의의를 느꼈다.

시간이 부족으로 인해 불완전 미로를 구현할 때에 직접 가로 세로 길이를 입력 받아서 그려지도록 하지 못해서 아쉬웠다. 또한, 이 게임에서 타이머를 설정해야 했던 이유는 지뢰를 랜덤으로 생성하였기 때문에 어떠한 경우에도 출구를 찾을 수 없는 경우가 존재했기 때문이다. 이를 정밀하게 계산하기에는 시간이 부족하여 타이머를 설정하여 게임을 끝내는 방식으로 해결했다. 이 때문에 사실상 시간을 매우 타이트하게 잡았고, 성공하는 경우는 매우 운이 좋아야 한다. 대부분 실패하고 밟은 지뢰 개수를 이용하여 점수를 매기도록 하였다. 또한, 랭킹 시스템을 구현할 때 테트리스 게임에서 사용했던 랭킹 시스템을 이용한 것이 아쉬웠다. 찾아보니 바이너리 트리나 세그 트리를 이용해서 구현하는 방법도 있는 것 같은데, 기회가 될 때 구현해보고 싶다.