Code Review - Minefind B

YJH ver

1. 구성

Define 게임 구성에 필요한 정보들이 정의 되어있다.

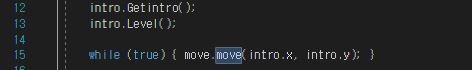
Intro 게임의 시작과 설정을 책임진다.  
Control 게임의 동작과 키 입력을 책임진다.  
Search 모든 연산이 이루어진다. GameManager 와 닮은꼴

2. 작동

Uml 참조

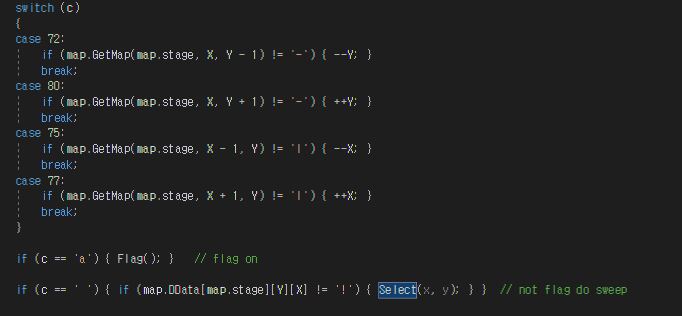
3. 아쉬운점

- 전반적으로 if , switch 등의 단순한 로직이 주로 이뤄 순서와 로직을 알아보기 힘듬  
class Search 에 모든 연산로직이 있는데,  
상태값의 변수를 선언하기도 하고 함수로서도 좀더 분활시켜 깔끔한 interface 를 유지할 수도 있었을거라 생각함.  
  
- Main()



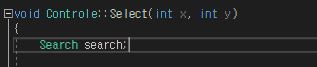
Application 의 정제된 코드에 익숙해져서인지 초반 도입부의 나눠진 루프(메뉴선택)는 좋지않은 출력 및 전환 방법이라고 생각되며, 식별자의 형태로만 동작이 어떻게 이루어지는지 짐작할 수가 없었다.

- Control -> move( int , int )



if 분기점들을 case 에 넣었으면 좀 더 가독성에 도움이 되었을 것 같다.

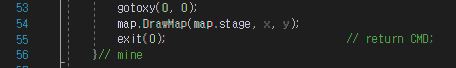
-Control -> Select ( int , int )



함수가 호출될 때마다 클래스가 선언되고 있다.

윗 단계에서 선언하여 참조하는 형태가 컴파일러 연산건강에 도움이 될 것 같다.

-Control -> Select ( int , int )



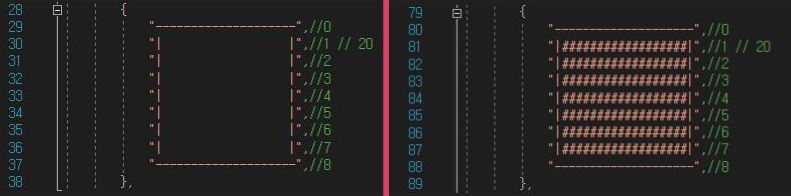
지뢰를 밟았을 때 강제로 종료가 되어버린다.

메모리 처리가 걱정되므로 좀더 안전한 방식이 좋을 것 같다. 의도한 것이라면

“지뢰를 밟게되면 메모리 릭과 강종을 선사한다.”라는 문구가 필요해보인다.

- Define::MapData[][][]

비어있는 맵과 꽉차있는맵 두가지가 있는데   
이런 닮은꼴의 배열을 비교하며 연산시 상태 값을 추가로 저장할 수 있는 struct 같은 타입의 배열을 사용하여 좀더 무난한 코딩이 가능할 것 같다.



- 가장 큰 문제점은 식별자와 interface 의 형태라고 생각한다.

짧은 코드에 코딩 따라하기도 쉬웠지만 사람의 생각방향과 어긋나있는 형식에 추론이 어려웠다.

4. 좋은 점