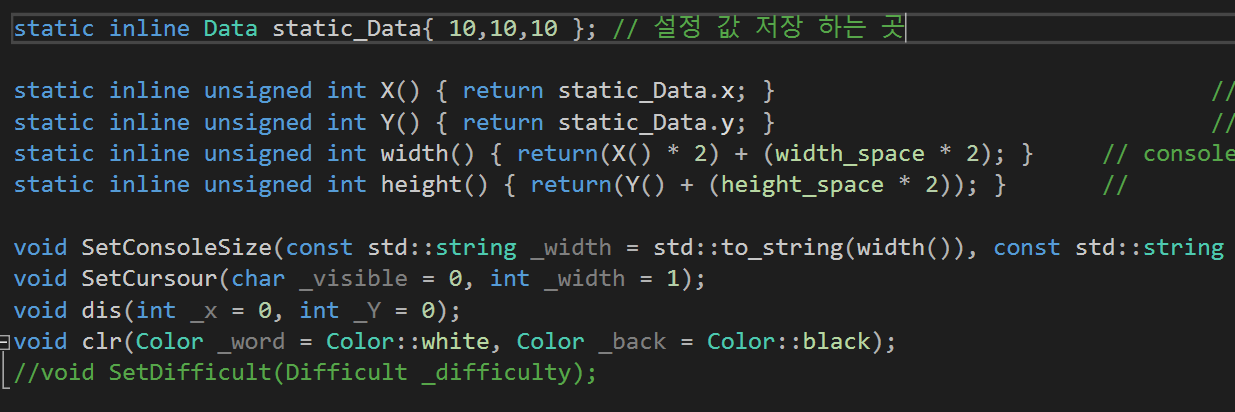
**지뢰찾기 코드리뷰 재현님**

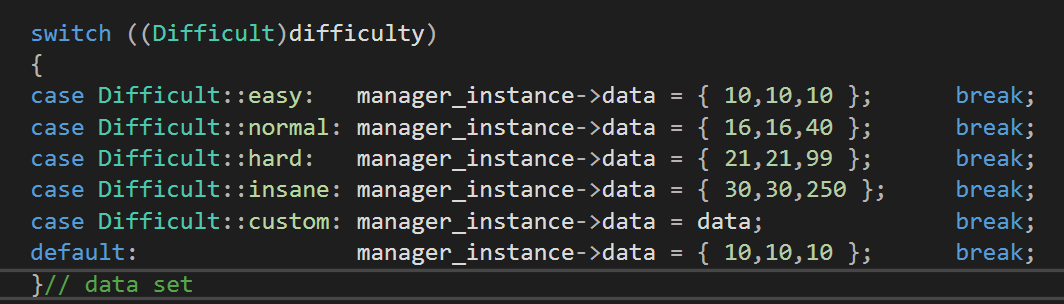
**written by 이호준**



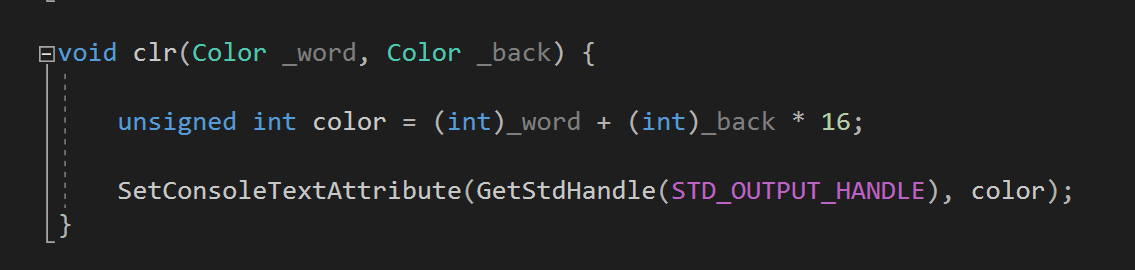
**난이도에 따른 맵데이터와 전역적으로 쓰일만한 함수들을 하나의 헤더에**

**정의하구 여러 클래스들이 사용하는데**

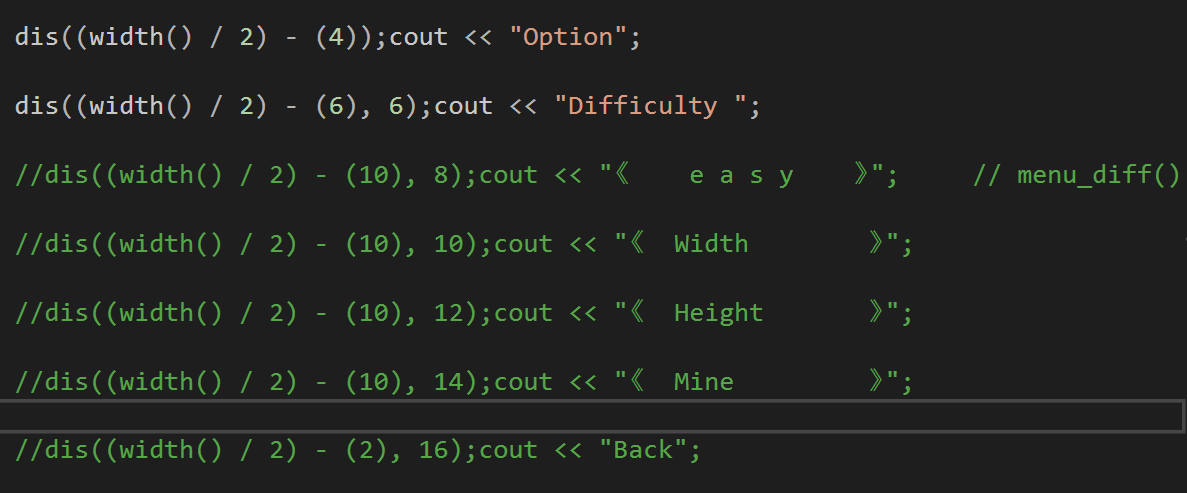
**맵데이터만 조심스럽게 사용한다면 좋은 방법이라구 생각합니다.**

****

**Enum Class 를 적극적으로 활용하여서 코드가 직관적이고 파악하기 용이한 도와주는부분이 많아서 좋은것 같습니다.**

****

**게임을 잘 만든것도 있지만 콘솔창에서 색깔까지 입히며 인터페이스를 깔끔하게 잘만들었다고 생각합니다.**

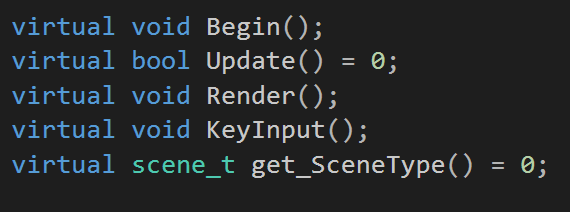
****

**코드설명 코멘트가 아니라면 깔끔히 지우는게 좋은것 같습니다.**

**코드내용 수정이후 컴파일 안될경우를 대비해 임시로 남겨놓는 경우는**

**작업하는 중에는 괜찮은것 같습니다.**

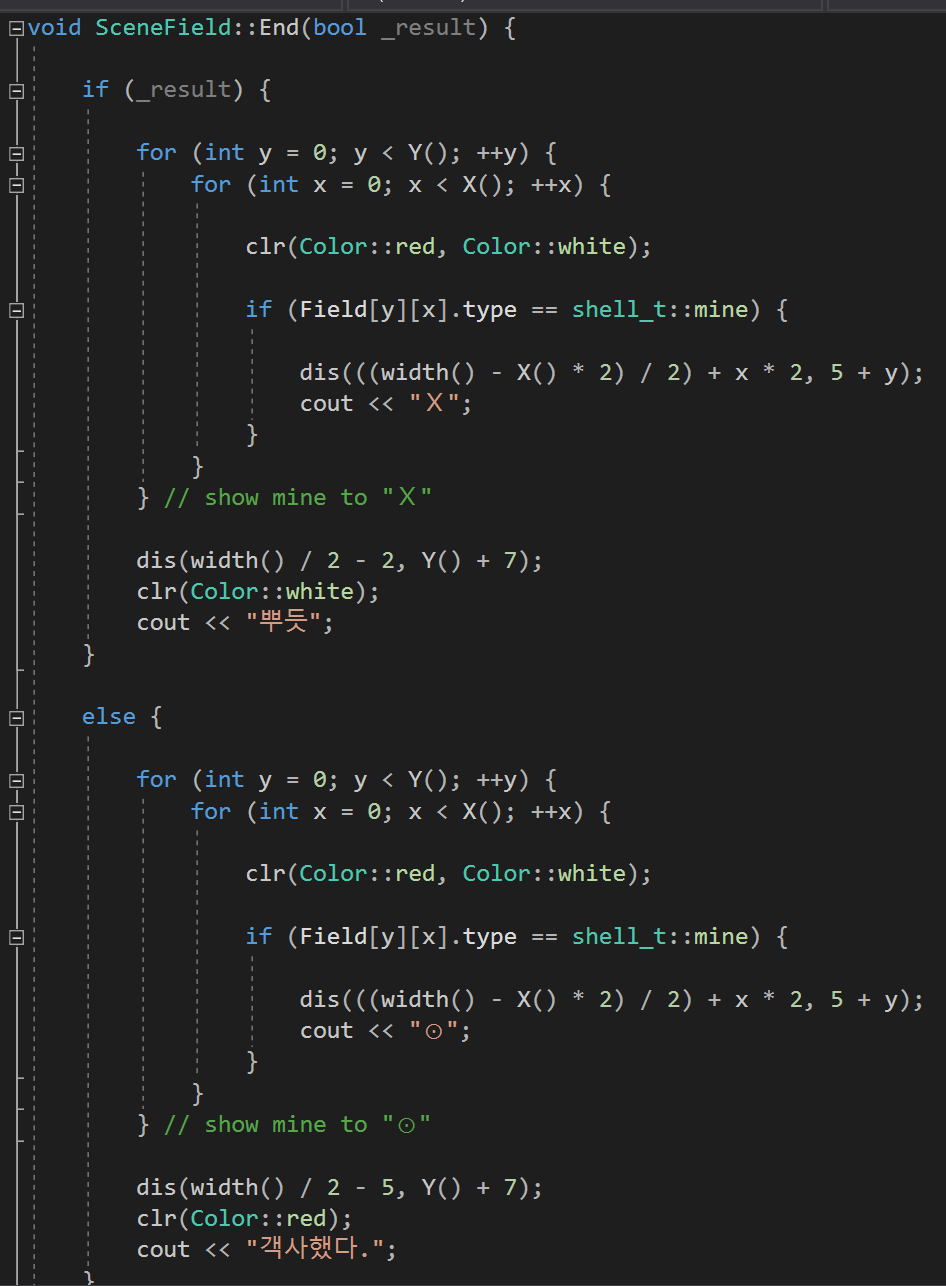
**(이 경우도 수정후 정상작동이 된다면 지우는게 좋은것 같습니다.)**

****

**FScene 클래스의 헤더인데 구현된건 생성자밖에 없고 멤버변수도 딱히 없고 자식클래스로 가보면 바로**

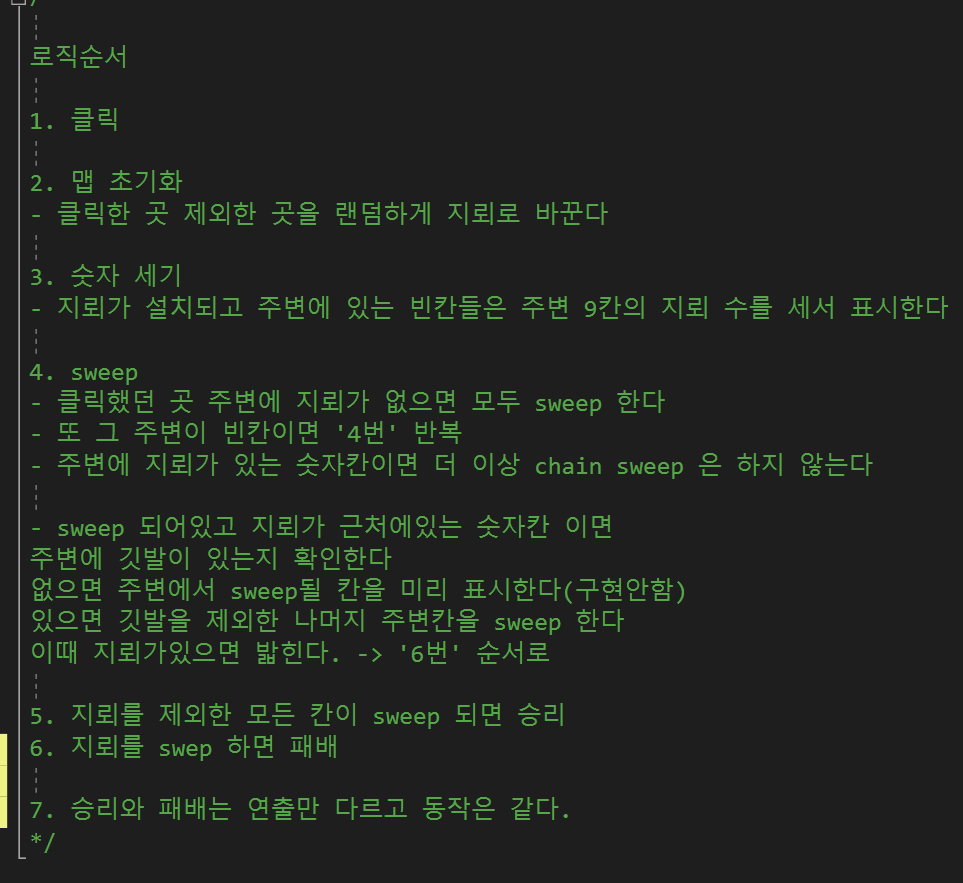
**추상클래스의 부모라는건 알수있지만 시그니처에 =0 으로 명시적으로**

**선언해주면 분석하는데에 조금 더 좋아지는것 같습니다.**

****

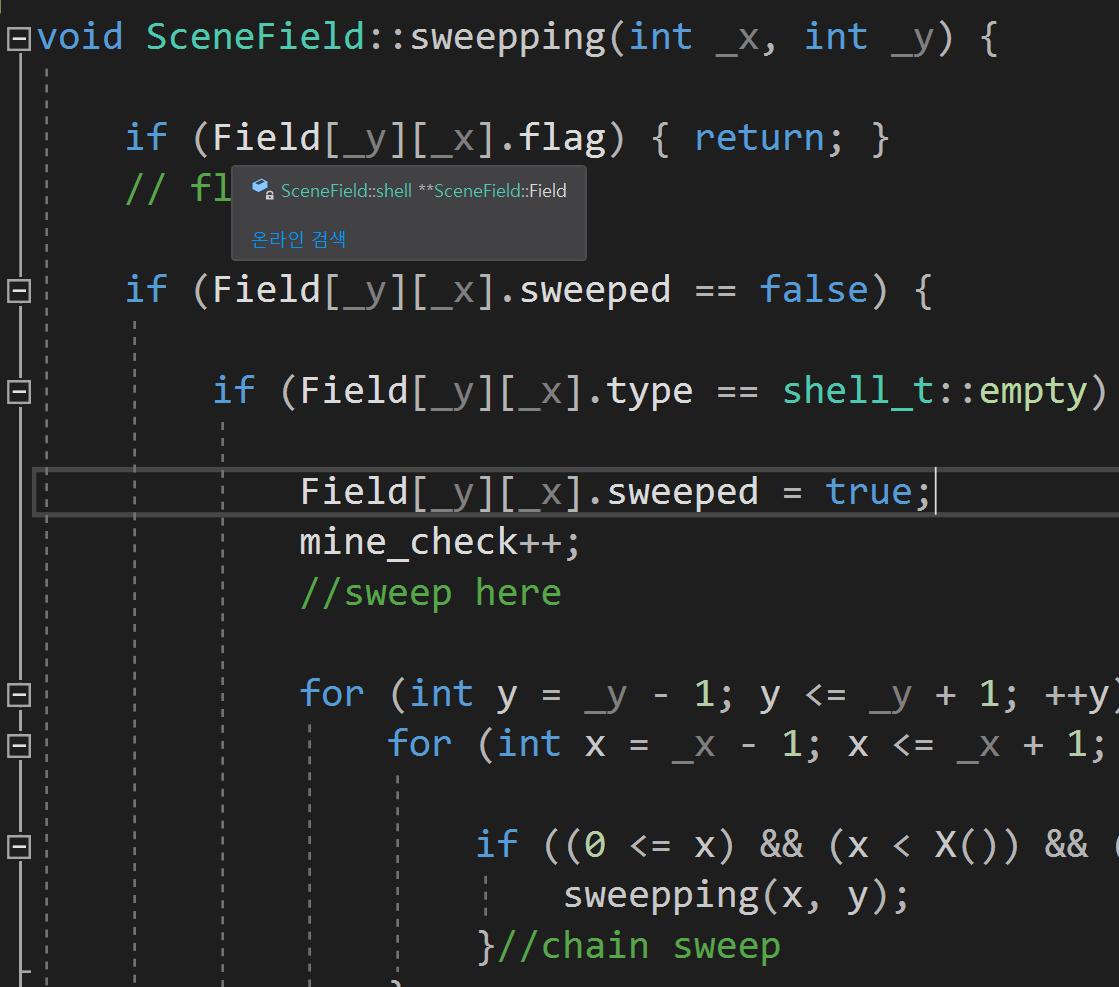
**게임오버씬 없이 종료플래그나 승리플래그시 호출되면 분기문으로 승리와 패배를 처리합니다. 지뢰찾기정도 사이즈의 게임에선 게임오버씬**

**클래스 안만들고 간단하게 처리하는게 맞다고 생각합니다.**

****

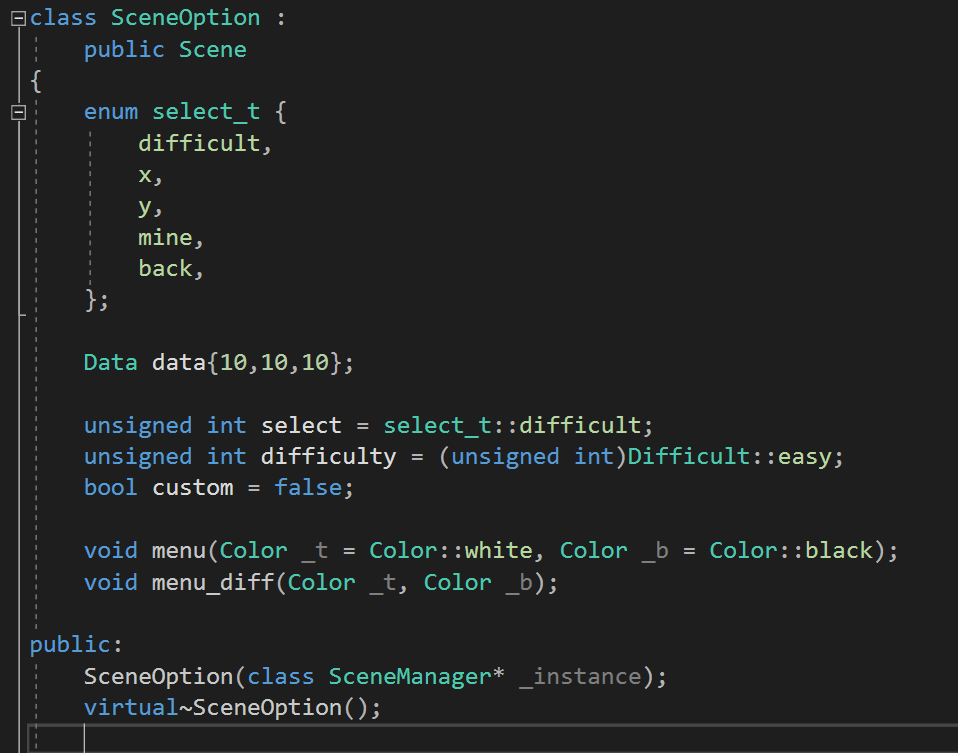
**구체적인 로직설명이 있어서 분석하기에 좋았던 점도 있지만**

**기간을 정해두고 프로그램을 다듬어 나갈때에 도움이 많이 될것 같습니다. 저는 간단히 한줄로만 짧게 적고 마는데 중요하거나 조금 복잡한 로직이 구현된 곳이라면 한곳에 이런식으로 적어두는것도 좋아보입니다.**

****

**플래그에 따라서 분기로 자기 자신을 계속 호출하는 식으로 지뢰찾기의**

**로직을 구현하였는데 지뢰찾기처럼 기준이 확실히 정해져있다면 분기를 정해두고 재귀호출로 구현하는 활용방안도 알았습니다.**

****

**해당 객체만 쓸 enum 은 해당 객체 private 안에 정의합니다**

**어떻게보면 기본적인건데 저같으면 전역적으로 참조하는 헤더도 있겠다**

**거기다가 일단 enum class 로 만들어놓고 시작할것 같은데**

**클래스안에 있으면 네임스페이스 처럼 구분도 되고 좋은것 같습니다.**

**멤버함수들도 private 안에 잘 숨겨놓았는데**

**저는 멤버함수는 거의 무조건 퍼블릭으로 만들고 enum 도 전역적인 헤더에 일단 만들고 봤는데 이 코드를 보고 많이 느꼈습니다.**