



코드 컨벤션

일단 정리 없이 생각나는거 적어봄 ㅎㅎ,, -산요나-

- extern 전역 변수를 사용하지 말자.
왜냐하면, 전역 변수를 사용해야 하는 곳마다, extern 전방 선언을 해주어야 하기 때문이다.
하나의 전역 변수이지만, 선언하는 곳이 여러군데 흩어져 있게 되는 것이다.
→ 대신 전역 변수만을 가지는 싱글톤 인스턴스를 만들고, 그 인스턴스로 가져오도록 하자.
예를 들어, Global이라는 클래스를 만들고, 사용할 전역 변수들을 이 클래스의 static 멤버로 선언해 사용하는 것이다.
이 경우에는 전역 변수에 접근할 코드 파일에서 Global 클래스가 선언된 헤더를 include해야 하지만, 멤버변수의 선언이 중복되지는 않는다.
- 어떤 함수가 매개변수로 포인터를 받고, 그 함수 내에서 포인터가 가리키는 실제 인스턴스를 변경할 때, 그 포인터 매개변수의 이름에 _out을 prefix로 달아주자.
→ 매개변수로 포인터를 건네줄 때에는 여러가지 경우가 있다.
크기가 큰 배열과 같이, 값을 복사하여 매개변수로 건네주기에는 부담스러울 때 주소값만을 건네주기 위해 사용하기도 한다.
그와는 다른 케이스로, 함수 바깥에서 사용하는 인스턴스가 함수를 호출하여 무언가의 변화가 있기를 기대할 때, 함수 내에서 그 인스턴스에 직접 접근하기 위하여 사용하기도 한다.
그렇기 때문에, 매개변수로 포인터를 건네줄 때 이 안에서 변경이 있을지 한 눈에 알아볼 수 있는 방법으로, 포인터가 가리키는 인스턴스가 수정될 수 있는 함수에서는 매개변수 이름 앞에 _out을 달아주자.

메모장에 적어둔거 적어두기 - 재현

- 멤버 변수는 m_ 를 앞에 붙인다.
- 전역 변수는 g_ 를 앞에 붙인다.

- 추가로 가능하면 전역으로 사용되는 변수를 멤버 변수로 갖는 싱글톤 객체를 사용한다.
- 함수는 무조건 대문자로 시작한다.
+ 동작이 앞에 온다.
(ex. SceneLoad(X) => LoadScene)
- 함수의 매개변수로 전달 되는 지역 변수는 _를 앞에 붙인다.
(ex. void Sum(int a, int b) (X) => void Sum(int _a, int _b))
이후 필요한 가이드라인이 생기면 추가한다.