코딩 컨벤션

// ############################# 함수네이밍 예시 #########################################

// 1. 상수는 대문자를 사용하고 단어 구분은 \_을 이용한다. ex) enum class, #define

// enum class EXAMPLE\_ENUM\_CLASS\_NUMBER

// {

// ONE,

// TWO,

// THREE,

//

// END =32,

// };

//

// #define SMAPLE\_DEFINE\_NUMBER 0

// 2. 변수명은 카멜 케이스 - 첫 문자는 소문자로 표기하고 그 이후에 연결되는 문자들의 첫자는 대문자로 표기

// ex) int exampleNumber = 10;

// 3. 클래스 내부의 변수(멤버 변수)는 앞에 m\_ 을 붙이고 이후에는 카멜케이스

// ex) float m\_playerSpeed = 100.f;

// 4. 함수 네이밍 파스칼 케이스 - 모든 문자의 첫 자를 대문자로 표기

// a.함수 이름은 해당 함수가 수행하는 작업을 나타내는 동사로 시작

// b.함수 이름에는 가능한 한 축약어 사용하지 않기

// c.함수 이름은 가능한 한 간결하고 명확하게 작성하기

// d. C++ 표준 라이브러리에서 제공되는 함수 이름과 겹치지 않도록 주의

// ex) int AddNumber();

// 5. 함수의 매개변수(인자) \_를 사용하고 소문자로 작성한다

// ex) void AddNumber(int \_num1, int \_num2, int\* \_result);

// 6. 클래스명 파스칼 케이스 사용

// ex) class KeyMgr

// 7. 오브젝트, 씬의 경우에는 자신의 소속을 표시한다. Object\_ + (이름)

// ex) Object\_Player, Object\_Pizza, Scene\_Main

// ##################################################################################

// ############################# 클래스 규칙 #########################################

// 1. 멤버 변수는 private으로 캡슐화

// 2. 멤버 변수는 함수를 사용해서 조정

// 3. 생성자와 소멸자는 필수적으로 구현

// 4. 매니저의 경우에는 싱글톤 패턴 사용

// 5. 함수의 선언은 헤더파일 구현은 cpp 파일에 구현

// 간단한 한줄짜리 코드는 헤더파일에 구현가능 하지만 두줄 이상의 경우에는 무조건 cpp파일에 구현해주세요

// ##################################################################################

// ############################# 습관 #########################################

// 1. assert 자주 사용하기

// 2. const 사용할 곳 생각해보기

// 3. 주석을 활용해서 코드를 작성하자

// 4. 유지 보수를 위해서 매직넘버는 피해주세요

// 5. 막히는 문제에 대해서 1시간 이상 고민X 같이하는 프로젝트이므로 물어보면서 진행

// 6. 전역변수 사용은 최대한 피하기

// ############################################################################