

코딩 컨벤션

작명 규칙

- 전역 변수: g_name
- 지역 변수 : camel case
ex) posX
- 함수 : 파스칼 pascal
ex) LoadFile
- 클래스 : 파스칼
ex) SceneManager
- 클래스 멤버 변수 : 언더바만 붙인다
ex) _pos
- 구조체 : 파스칼
ex) Vector
- 구조체 멤버 변수 : camel case
- Define : 대문자, 띄어쓰기는 언더바로 구분.
ex) DECLARE_SINGLE
- enum class : 파스칼, 내부 값도 파스칼

```
enum class SceneType
{
    None,
    DevScene,
    GameScene,
};
```

- enum : 대문자, 띄어쓰기는 언더바로 구분. 내부 값 동일

```
enum LAYER_TYPE
{
```

```

    LAYER_BACKGROUND,
    LAYER_OBJECT,
    LAYER_UI,

    LAYER_MAXCOUNT,
};

```

- int 사용하지 말고 아래와 같이 크기를 명시할 것

```

using int8      = __int8;
using int16     = __int16;
using int32     = __int32;
using int64     = __int64;
using uint8     = unsigned __int8;
using uint16    = unsigned __int16;
using uint32    = unsigned __int32;
using uint64    = unsigned __int64;

```

줄임말

가능하다면 풀네임으로 사용

- pos : position

괄호 사용 방식 : BSD

```

if(true)
{
    int a = 3;
}

```

클래스 규칙

- 멤버 변수는 private으로 캡슐화 - 멤버 변수는 함수를 사용해서 조정

- 멤버 변수 멤버 이니셜라이저로 초기화
- class 내에서는 멤버 변수는 하단에 작성
- Get, Set을 제외한 함수의 정의 부분은 cpp에 구현할 것.

ex)

```
// Test.h
class Test
{
public:
    Test(int32 a);
    void SetNumber(int32 a) {_a = a;}
    int32 GetNumber() { return _a; }

    void Function();
private:
    int32 _a;
};
```

```
// Test.cpp
Test::Test(int32 a) : _a(a) {}

void Test::Function()
{
    // abc
}
```

기타

- 함수의 주석은 아래와 같이 ///을 사용할 것.

```
/// <summary>
///
/// </summary>
/// <param name="component"></param>
```

- assert 자주 사용하기
- 전역변수 사용 시 문서에 따로 작성할 것.

- 주석 열심히 달기
- const는 될 수 있으면 자주 사용.
- 좀 오래 걸릴 것 같은 (2 ~3 시간) 문제는 공유할 것.

데일리 스크럼을 할 것

- <https://gmlwjd9405.github.io/2018/06/01/agile-dailyscrum.html>

1. 1. 어제 한 것 2. 오늘 할 것 3. 어려운 점