

1단계_02. 언리얼 C++ 코딩 표준

📅 수강일	@2023/03/14
➦ 이름	🐰 <u>전영재</u>
🔍 멘토	Min-Kang Song, 현웅 최
👥 멘티	
⚡ 작성 상태	In progress
📌 단계	1단계
☑ 강의 시청 여부	☑
☑ 이수 여부	☐

Contents



언리얼 C++ 코딩 표준

코딩 표준

프로그래밍 언어의 명명규칙

상속 접두사

Const 정확도

최신 C++언어 문법 (C++17) 권장

디폴트 멤버 이니셜라이저 (초기화)

물리적 종속성

0n. ~~~ 강의 과제

언리얼 C++ 코딩 표준



강의 목표

- 언리얼 C++ 코딩 표준을 이해한다.

- 언리얼 C++ 표준 코딩 중 주의해야 할 점 확인한다.
-

코딩 표준

(= 코딩 스타일, 코딩 컨벤션 ...)

해당 코드를 이해하기 위한 코드를 작성하는 방법을 의미한다.

마치 한사람이 작성한 것처럼 '규칙'을 준수해야 하며, 이를 통해 코드이해화 유지보수 등의 효율성이 올라간다.

프로그래밍 언어의 명명규칙

1. 파스칼 케이싱 : 합성어의 첫글자를 대문자로 사용 : UnrealEngine
2. 카멜 케이싱 : 첫 합성어는 소문자, 나머지는 대문자를 사용 : unrealEngine
3. 스네이크 케이싱 : 합성어 사이에 언더바를 사용 : unreal_engine

언리얼 엔진은 위 규칙 중 '1.파스칼 케이싱'을 사용한다. UnrealEngine

상속 접두사

또한 `Object` 를 상속받을 경우 접두사 U를 포함하고

`Actor` 를 상속받을 경우 접두사 A를,

`Widget` 을 상속받을 경우 접두사 S,

추상적 인터페이스 클래스의 경우 접두사 I를 포함한다.

위를 제외하고 대부분의 경우에는 접두사 F를 사용한다. // `FMath`

// `boolean` 값을 반환하는 경우는 소문자 b를 사용한다.

Const 정확도

불변성이 필요한 코드에는 가능한 const를 사용하여 코드의 정확도 향상을 노려야한다.

최신 C++언어 문법 (C++17) 권장

1. `override` 및 `final` 적극권장
2. `nullptr` 사용
3. `auto` 키워드의 제한

변수에 람다를 바인딩하는 경우, 이터레이터 변수를 사용하는 경우, 템플릿 코드에서 타입을 쉽게 식별할 수 없는 경우를 제외한 경우에는 사용을 자제한다.

4. 범위기반 `for` 문

처음부터 끝까지 모두 순회하는 경우에 사용하기 좋다.

5. 람다

6. `Enum`

디폴트 멤버 이니셜라이저 (초기화)

헤더에서 초기화를 하는 것은 가능하지만, `UObject`의 특성을 파악하고 생성자 구문에서 초기화하는 것을 권장한다.

물리적 종속성

파일이름에는 가급적 접두사(U,A,S) 등을 붙이지 않는다.

또한 모든 헤더에는 `#pragma once` 지시어를 통해 복수의 `#include`를 방지해야 한다. 나아가 `include`대신 '전방선언(forward declaration)' 사용을 권장한다.

Summary

1. `public` - `private`로 이어지는 클래스 체계 준수
2. 명명규칙: 파스칼 케이싱 사용과 각각 클래스와 구조체에 할당된 고유한 접두사를 사용할 것
3. 코드의 명확성을 위해 `In`, `Out` 파라미터접두사, `const`지시자, 레퍼런스 복사방지, `auto` 자제
4. Find In Files 적극활용

0n. ~~~ 강의 과제

? Q1. 왜 코딩 표준을 지켜야 하는가?

자신 이외의 프로그래머가 해당 코드를 봤을 때 가독성을 향상 시킬 수 있으며, 이를 통해 유지보수 및 발전에 드는 비용을 낮출 수 있다.

? Q2. 공식 웹의 코딩 규칙을 보면서 느낀점

? Q3. const 정확성이란? (선택)



Reference