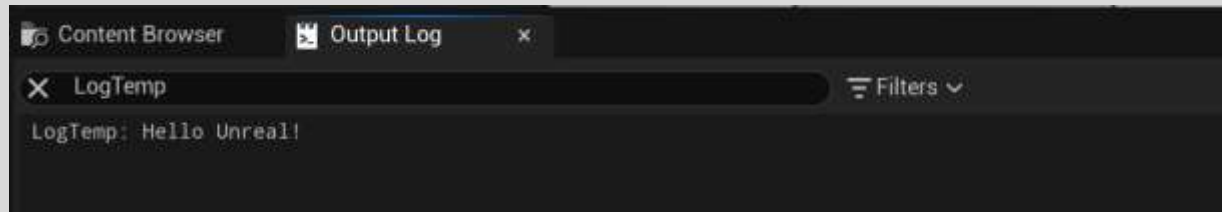


헬로 언리얼!

(Hello Unreal!)

강의 내용

언리얼 엔진 에디터에서 **Hello Unreal!** 문자열을 출력하기

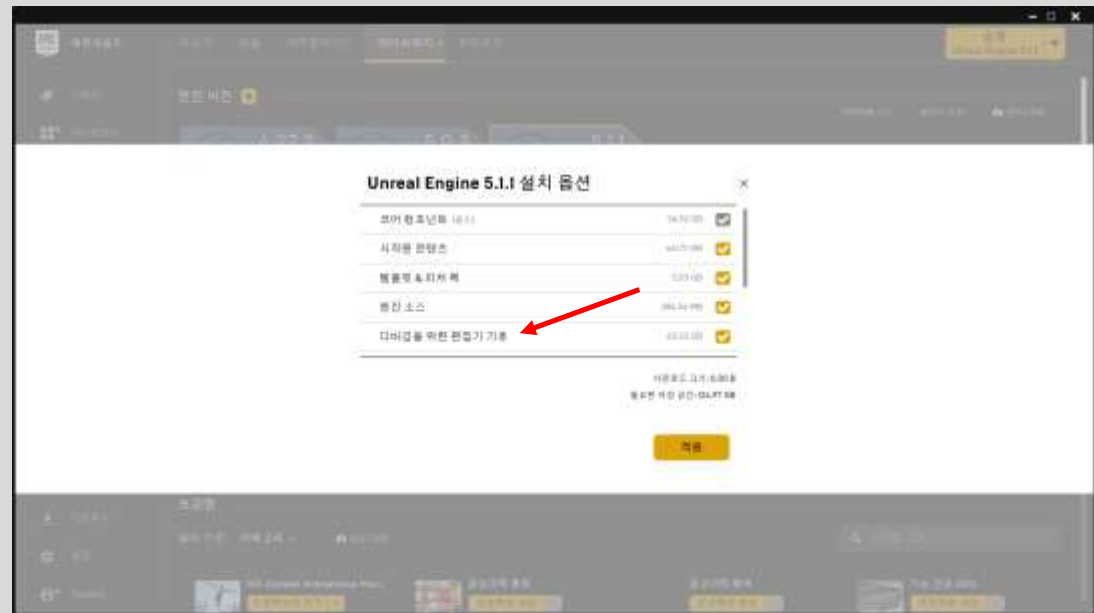


강의 목표

- 언리얼 C++ 프로그래밍을 위한 개발 환경 설정
- 앞으로 강의에 사용할 프로그래밍 환경 구성 학습

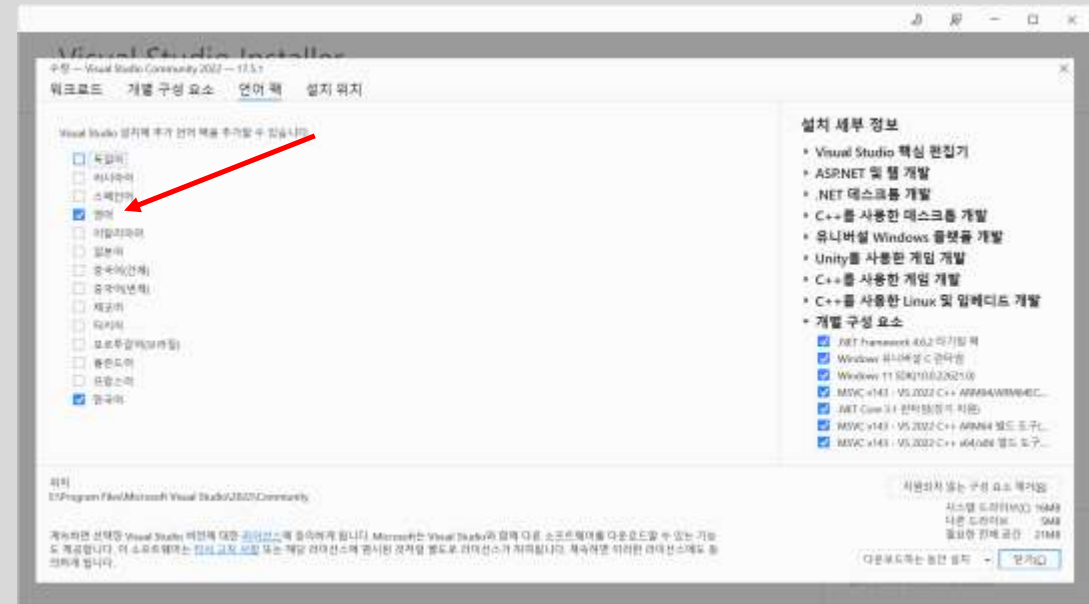
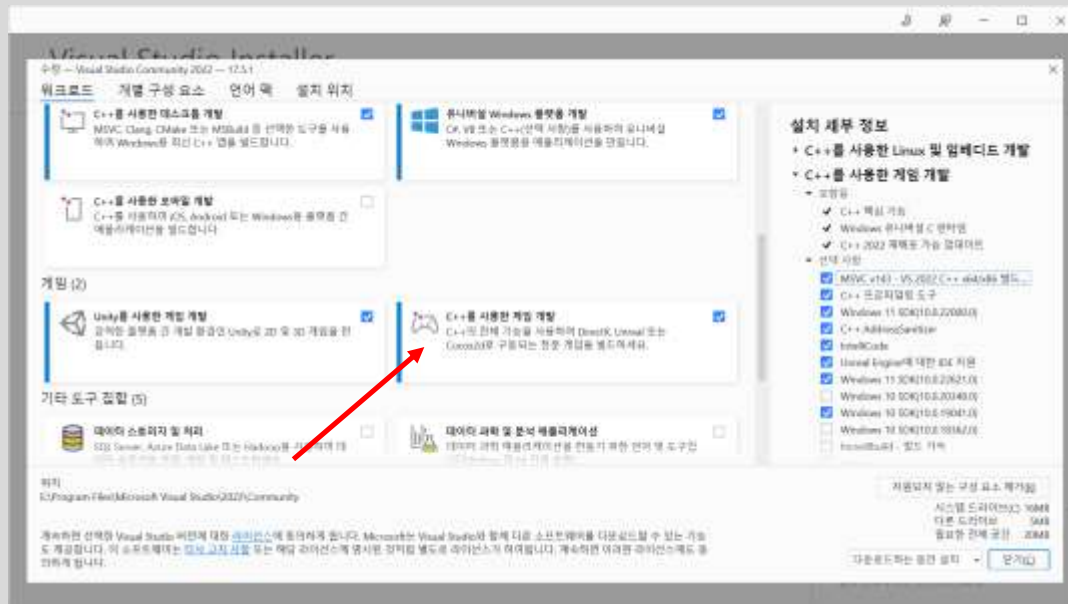
언리얼 엔진 설치 및 개발 환경 설정

- 언리얼 엔진 버전 : 5.1.1
- 디버깅을 위한 편집기 기호(디버깅 심볼) 체크

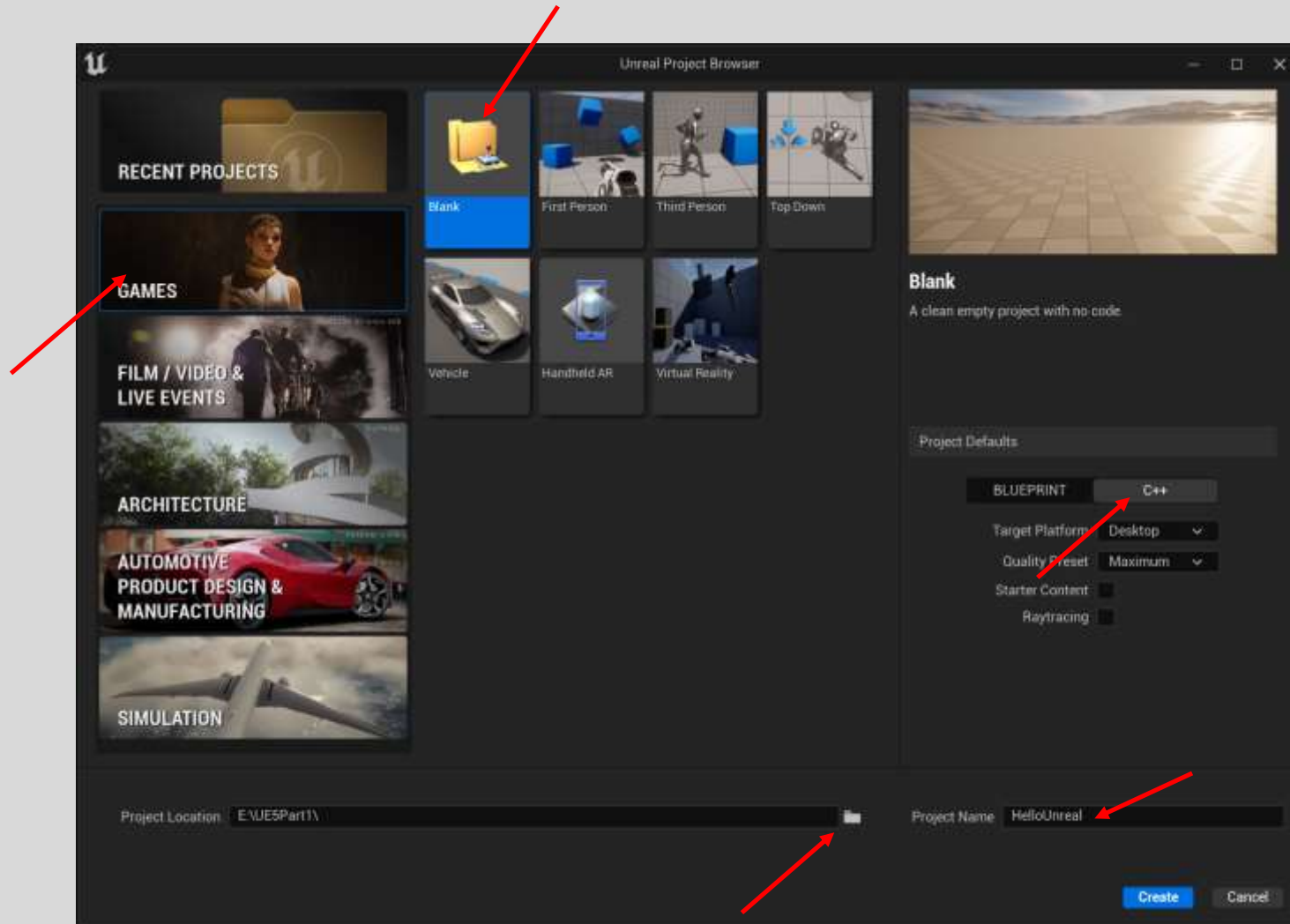


비주얼 스튜디오 2022 설치

- 비주얼 스튜디오 2022
- C++을 사용한 게임 개발
- 영어 언어팩 설치

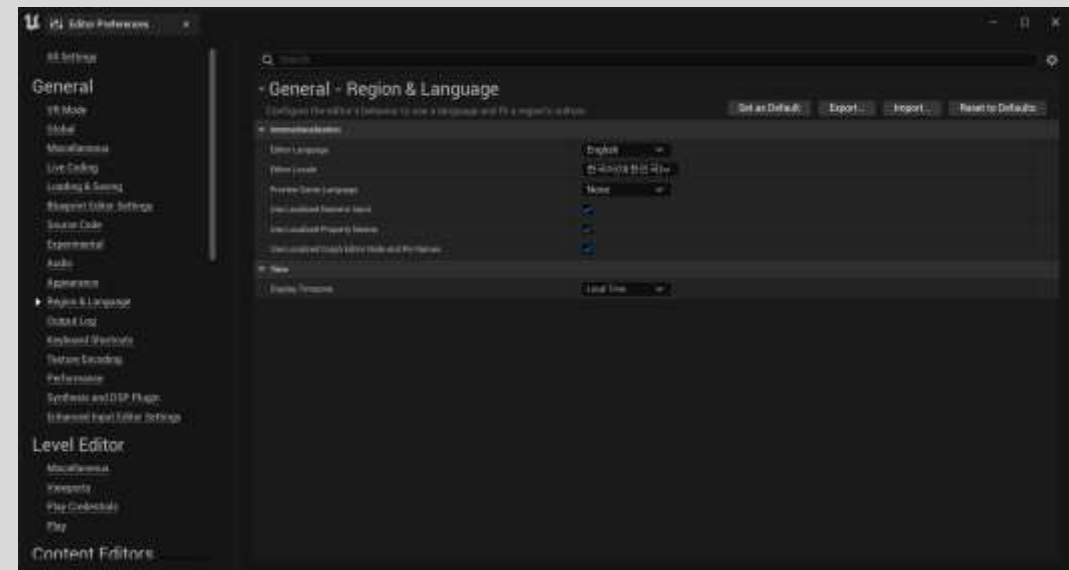
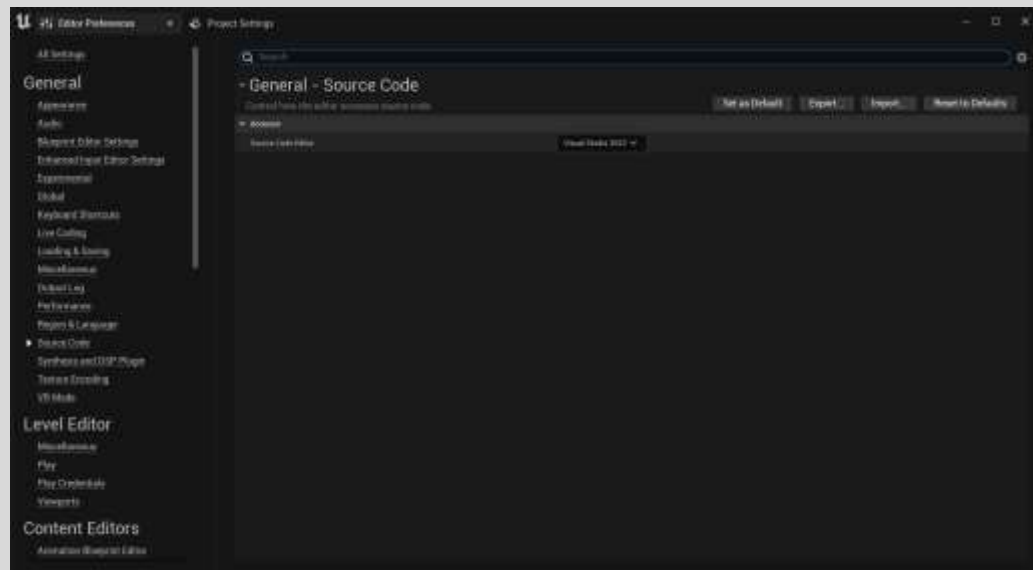


언리얼 C++ 프로젝트 생성



언리얼 엔진 프로젝트 설정

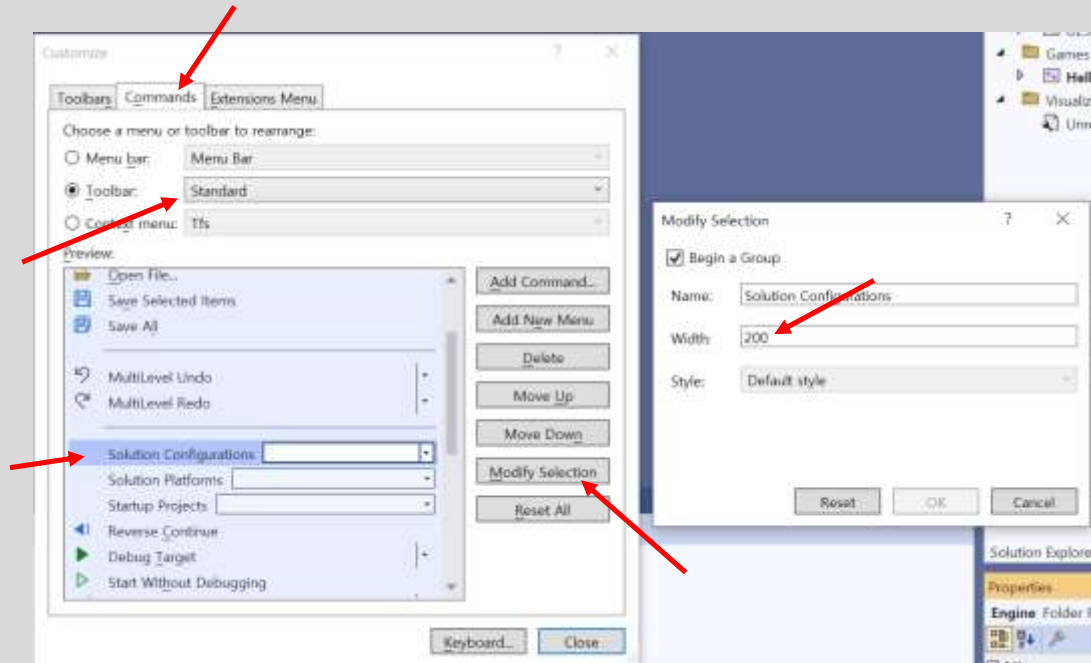
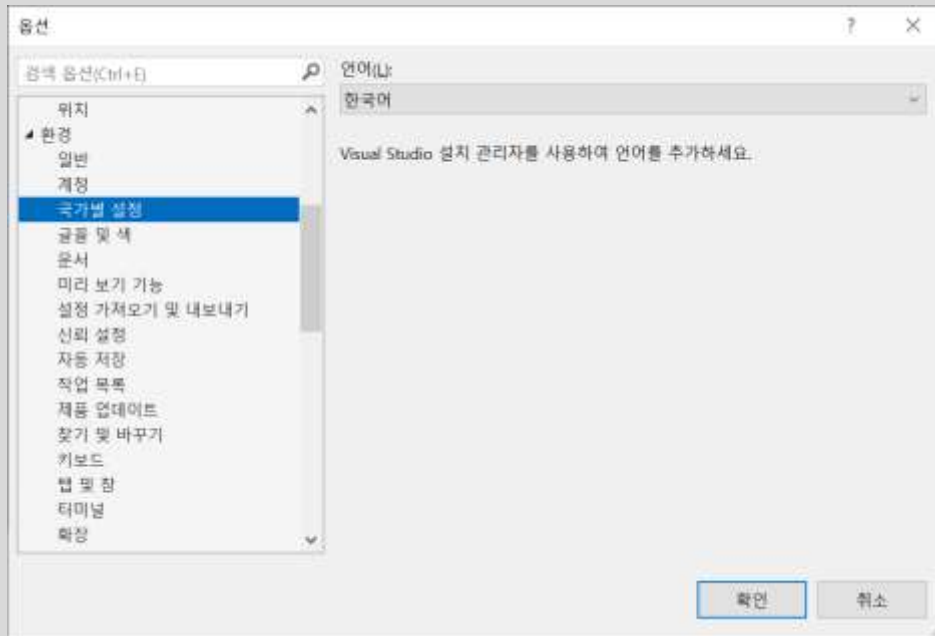
- 에디터 개인설정(Editor Preference) 점검하기
 - 지역 & 언어(Region & Language) 설정 : English로 변경
 - 소스코드(Source Code) 설정 : 비주얼 스튜디오 2022로 지정



언리얼 C++ 프로그래밍을 위한 설정

비주얼 스튜디오 개발 환경 설정

- 유용한 비주얼 스튜디오 설정
 - [도구 > 옵션] 메뉴에서 국가별 설정을 English로 변경
 - 비주얼 스튜디오 재시작
 - [Project & Solutions]의 General에서 Always show Error List 옵션 끄기
 - 툴바 : [Customize] 후 명령 > 도구모음(표준) > Solution Configuration 너비를 200으로



비주얼 스튜디오 추가 개발 환경 설정

- 비주얼 커맨더 설치
- 비주얼 커맨더내에서 언리얼 인덴트 매크로 확장 도구 설치



<https://bit.ly/visualcommander>

<https://vlasovstudio.com/visual-commander/>



<https://bit.ly/ue4vstool>

<https://github.com/hackalyze/ue4-vs-extensions>

언리얼 C++의 코딩과 컴파일

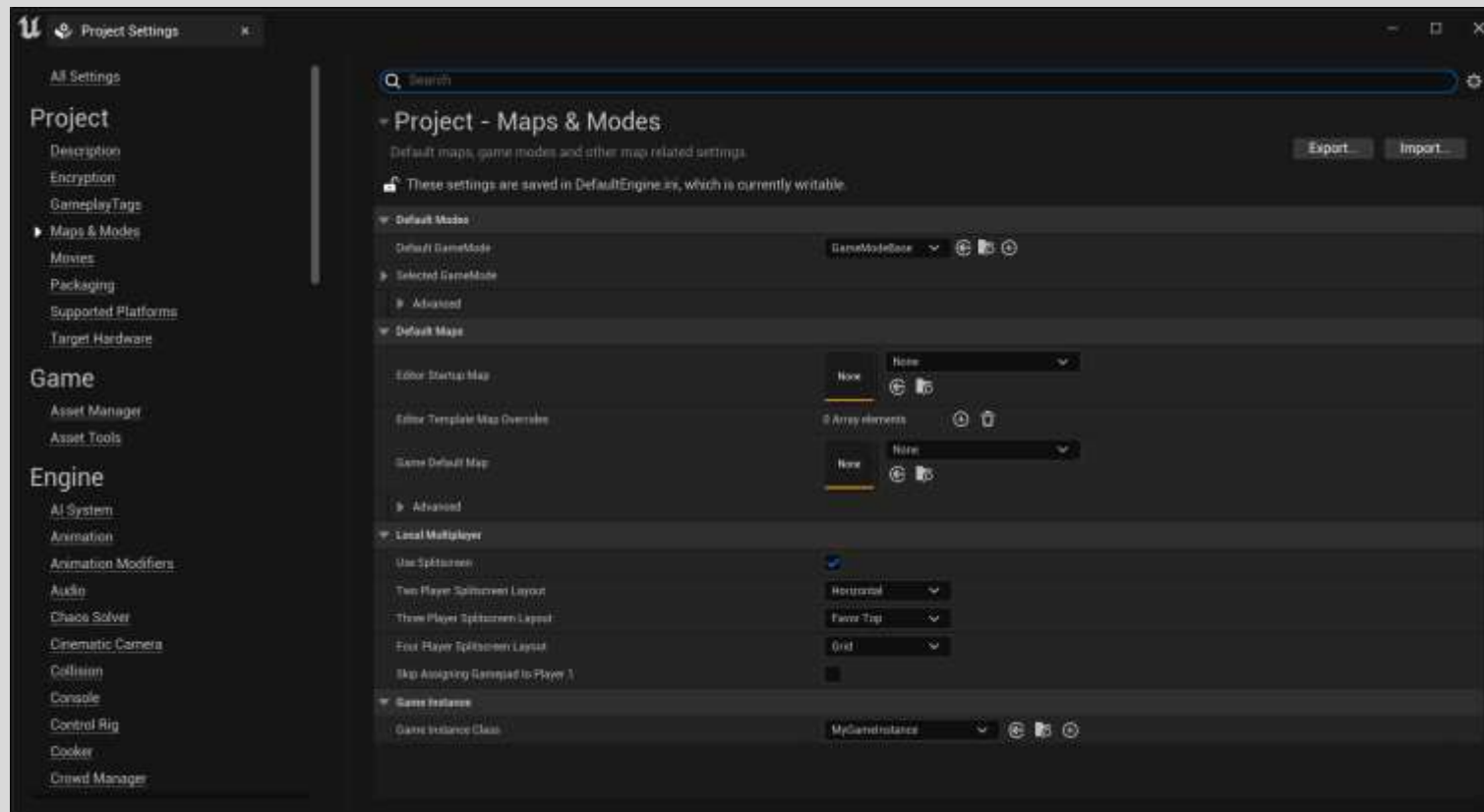
언리얼 코드 컴파일 방법

- 헤더 파일에 변경이 발생하면 - 에디터를 끄고 비주얼 스튜디오에서 컴파일한다.
- 소스 파일에만 변경이 발생하면 - 라이브 코딩으로 컴파일한다. (Ctrl+Alt+F11 단축키)

비주얼 스튜디오에서 수동으로 클래스를 추가하지 말 것!

새로운 게임 인스턴스의 적용

- 프로젝트 세팅(Project Settings)의 맵 & 모드(Map & Modes)에서 지정
- 추가 권장 설정 : 에디터와 게임 시작맵 OpenWorld를 휴지통(Clear) 버튼을 눌러 None으로 변경



정리

헬로 언리얼을 출력하기까지의 여정

1. 언리얼 엔진 설치와 프로그래밍 환경 구축
2. 언리얼 에디터에서의 클래스 추가
3. 언리얼 C++의 클래스 상속 및 오버라이딩 구현
4. 언리얼 엔진의 문자열 처리의 이해
5. 게임인스턴스 클래스의 적용과 카테고리를 활용한 로그 확인