

[Ludico] KNOCKturne

언리얼 프로젝트 관리 문서

1. 서론

언리얼 프로젝트는 용량이 매우 커 Git 단일 저장소로는 모든 파일을 관리하기 적합하지 않다. 기본적으로 100MB 이상의 파일 업로드를 허용하지 않기 때문에, 대용량 파일, 애셋을 관리하는 별도의 저장소를 두고 사용하는 것이 일반적이다.

a. 구글 드라이브를 활용

구글 드라이브는 15GB의 무료 저장소를 제공한다. 녹턴의 데모 버전이 아무리 커도 15GB면 모든 에셋을 수용할 수 있을 것으로 보인다. 애셋 저장용 계정을 따로 만들고, 그 곳에 에셋을 저장하는 식으로 진행하는 것이 적합할 것이다.

2. Git 저장소

Github의 원격 저장소(Repository)에는 필수적인 파일만 저장한다. 저장소에 있는 다른 파일을 이용해 생성할 수 있는 파일(ex. sln 파일 등)은 저장하지 않는다.

어떤 파일을 원격 저장소에 저장할 것인지는 .gitignore 파일을 이용해 관리한다.

a. .gitignore

원격 저장소 생성 시 추가할 수 있는 파일로, 현재 KNOCKturne repository에는 .gitignore 파일을 생성해 둔 상태이다. 해당 파일은 Git이 개발자가 원격 저장소를 Clone한 로컬 저장소에서 Change가 발생했을 때 특정 파일의 Change를 무시하는 역할을 한다.

다시 말해, 로컬 저장소에는 애셋을 포함하여 무수히 많은 파일이 프로젝트 폴더 안에 저장돼 있지만, .gitignore 파일을 활용하면 일일이 특정 파일에 대한 Change를 Discard하지 않아도 원하는 파일만 원격 저장소에 저장할 수 있다.

언리얼 프로젝트에서 .gitignore 파일로 관리해야 할 대상은 대표적으로 엔진에 의해 생성되는 파일 / 애셋 파일 / 스크립트 편집기(VS 등)에 의해 생성되는 파일이 해당된다. 이때 애셋 파일은 1.a절에서 언급한 별도의 저장소를 활용한다.

b. 언리얼 프로젝트 디렉토리 구조

언리얼 프로젝트가 생성되었을 때 접근할 수 있는 디렉토리 구조와 각 요소가 무엇인지 간략하게 알아볼 것이다.

C++ 기반 소스 프로젝트는 다음과 같은 디렉토리 구조를 갖는다. KNOCKturne 역시 블루프린트와 C++을 혼합하여 사용할 것이므로 해당 구조에 대해 알아 둘 필요가 있다. 항목 i의 요소들을 먼저 파악하고, 나머지는 후에 해도 괜찮다.

i . 깃허브에 반드시 저장해야 하는 파일 및 폴더

프로젝트.uproject	언리얼 에디터 구동 시 해당 프로젝트에 대한 정보를 포함하고 있는 JSON 파일. 언리얼 에디터를 구동시키는 연결 파일.
Config	<ul style="list-style-type: none"> - 프로젝트의 언리얼 환경설정 파일을 담고 있는 폴더 - 지워도 에디터 실행에는 문제가 되지 않지만 만약 프로젝트를 작업한 상태에서 지웠다면 설정해놓은 설정값들은 없어지고 언리얼 엔진에서 제공하는 기본 세팅으로 적용 - 지우고 나서 프로젝트 실행해도 Config 폴더가 생기지는 않음. Default 설정 파일들은 언리얼 엔진 폴더에 있기 때문.
Content	언리얼 에디터에서 사용하는 uasset 들이 들어있는 폴더.
Source	C++ 프로젝트 전용. 프로젝트의 모듈 소스와 빌드 파이프라인 C# 소스파일 등이 들어있는 폴더.
Plugins(옵션)	<ul style="list-style-type: none"> - 해당 프로젝트에서 외부 플러그인 혹은 자체 제작 플러그인을 사용하는 경우 존재하는 폴더. - 엔진에서 제공해주는 플러그인은 해당 프로젝트 폴더에 생성되지 않음.

ii . 깃허브에 있으면 좋은 파일 및 폴더

Binaries	<ul style="list-style-type: none"> - C++ 컴파일 과정을 거쳐 생성된 프로젝트에 대한 에디터 전용 DLL 파일들이 있는 폴더. - 그렇기 때문에 이 폴더가 없으면 .uproject를 실행시킬 때마다 에디터 실행 시 'Missing <프로젝트명> Modules' 메시지 팝업을 띄움. Yes를 누르면 컴파일 과정을 거쳐 다시 파일과 폴더를 생성하고, 에디터가 실행됨.
Saved	<ul style="list-style-type: none"> - 지워도 문제가 되지는 않지만 작업 시 귀찮은 부분들이 존재할 수 있음. - 프로젝트 작업 시 임시로 저장되는 파일들이 보관되는 장소로, 자동 저장, 스크린샷, 빌드 파일, 백업 임시 파일, 로그 등등 많은 데이터가 축적되는 장소. - 언리얼 에디터 사용 시 작업의 효율성을 높이거나 복구작업, 문제 확인 등의 활용성이 높은 폴더. - 다만 작업이 길어지고 패키징까지 하면 용량을 꽤 많이 잡아먹음. - Saved 폴더에 저장된 파일과 폴더 들 중에서 개인적으로 Config 폴더를 제외한 나머지 파일 및 폴더를 지우고 백업하는 편. 이유는 아래에서 설명함.

iii. 깃허브에 올리지 않아도 되는 파일 및 폴더

Intermediate	<ul style="list-style-type: none">- 언리얼 엔진 라이브러리 및 해당 프로젝트 소스 코드에 대한 빌드 과정을 거치면 생성되는 파일들이 들어있는 폴더.- [언리얼 문서 - 디렉터리 구조 설명] 엔진이나 게임 빌드 도중 생성된 임시 파일이 들어 있으며, 게임 디렉터리에서 셰이더는 Intermediate 디렉터리에 저장됨.- 임시 파일이기 때문에 지워도 에디터 실행 시 자동으로 생성됨.
Build	<ul style="list-style-type: none">- 언리얼 프로젝트 패키징 과정을 거치면 생성되는 폴더.- [언리얼 문서 - 디렉터리 구조 설명] 엔진이나 게임을 빌드하는데 필요한 파일은 물론, 플랫폼별 빌드를 만드는데 필요한 파일도 들어 있음.- 예를 들면, 안드로이드 플랫폼의 경우 빌드 환경 설정(프로젝트 설정 - Android) 시 자동으로 생성됨.
.vs	<ul style="list-style-type: none">- C++ 프로젝트에서 .sln 파일과 마찬가지로 Generate 과정에서 생성되는 폴더.- 폴더 자체는 숨김 설정되어 있으므로 윈도우 탐색기의 폴더 옵션에서 숨김 파일 보기를 설정해야만 볼 수 있음.- 해당 프로젝트를 Visual Studio에서 실행할 때 초기화 및 데이터 구조 등을 기록하여 추후 솔루션 파일을 실행할 때 불러오기 과정을 단축하기 위한 용도.- 지워도 솔루션 파일을 실행하거나 Generate 과정을 거치면 다시 생성.
프로젝트.sln	<ul style="list-style-type: none">- Visual Studio Solution 파일.- .uproject 파일 선택 후 마우스 우클릭 → 팝업 메뉴에서 'Generate Visual Studio project files'를 선택하면 생성되는 C++ 프로젝트에서만 볼 수 있는 파일.- Visual Studio에서 작업하기 위해서는 필요함.- 지워도 위에 적은 방식으로 다시 생성할 수 있음.

작성 중

3. 참고

- [\[UE Document\] 디렉터리 구조](#)
- [깃허브를 통한 언리얼 엔진 프로젝트 공유 방법](#)
- [\[Tip\] 협업시 관리해야할 언리얼 엔진 4 디렉토리에 대하여](#)
- [\[언리얼 엔진\] 프로젝트 폴더 구조 및 용량 관리](#)