언리얼 엔진을 활용한 효과적인 게임 개발 학습 방법

청강문화산업대학교 이득우

2024년도 게임 산업 트렌드

2024년 언더독 타이틀의 흥행

PC/Console 2500만명

Mobile 매출 1위

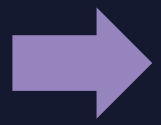
팰월드 게임 레벨 디자인

팰월드의 그래픽적인 분석. 과연 2024년도의 게임 기술인가?
그럼에도 흥행한 이유는?

언더독 타이틀 성공 요인 분석

- 우수한 고품질 그래픽
- 웅장한 서사와 반전 스토리
- 방대한 게임 레벨 디자인
- 게임 미캐닉의 조합과 조작감
- 깊이있는 스킬 시스템
- 멀티플레이 경험

방대한 자본과 인력 투입



- 우수한 고품질 그래픽
- 웅장한 서사와 반전 스토리
- 방대한 게임 레벨 디자인
- 게임 미캐닉의 조합과 조작감
- 깊이있는 스킬 시스템
- 멀티플레이 경험

중소규모 자본과 인력 투입

버섯커 키우기 분석 기사

https://www.thisisgame.com/webzine/special/nboard/11/?n=184842

기획/특집

MMORPG 잡아먹고 매출 1위, '버섯커 키우기' 왜 잘 나가나?

양산형? 방치형? 본질은 조금 다르다

17,654 view 2,787 view 에 유통된 기사입니다.

김승준(음주도치) 2024-02-19 18:47:15

처음에는 차 떼고 포 떼고 '업 앤 다운'

<버섯커 키우기>는 게임의 이름에서 유추할 수 있듯 '버섯'을 키우는 방치형 RPG다. 게임의 후반부까지 일관되게 요구되는 사항은 결국 캐릭터 '스펙 세팅'인데, 특정 성장 루트까지 도달하는 과정이 다른 게임들과 미세하게 다르다.

일단 메인 캐릭터가 하나다. <세븐나이츠 키우기>, <포트리스 사가> 등 최근 방치형 RPG들이 여러 캐릭터를 수집해 '파티'를 꾸리고 '전략'을 구성하는 것을 내세웠던 반면, <버섯커 키우기>에서는 스킬, 동료(펫) 뽑기는 존재해도 메인 캐릭터인 버섯은 단 하나 뿐이다. 해당 버섯의 전직 루트를 다르게 선택하며 성장시키는 다소 고전적인 구조다.

<버섯커 키우기>는 신규 유저의 진입장벽과 기존 유저들의 불필요한 부담을 없애기 위해 일반적인 RPG의 전형적인 요소들을 상당수 배제했다. 악룡으로부터 세계를 구한다는 것 외엔 스토리도 없고, 게임플레이 초반부에서는 눈에 띄는 장비나 캐릭터도 없다. 그러나 등급의 세분화, 장비의 다양화, 도전 과제 및 이벤트의 양을 통해 성장 경험을 굉장히 촘촘히 배치한 것이 눈에 띈다.

성장을 위한 깊이 있는 게임 시스템이 핵심

왜 언리얼 엔진인가?

언리얼 엔진의 강점

최첨단 실사 그래픽 기술

디즈니, 에픽게임즈에 2조원 투자…
“포트나이트, 디즈니 캐릭터 콜라보”

캐릭터 IP와 게임 서비스

언리얼 엔진의 구조

콘텐츠를 표현하는
각종 시각적 기능

네트웍 멀티플레이 기반의
게임플레이 프레임웍

언리얼 엔진이 제공하는 핵심 기능

스태틱메시와
스켈레탈 메시

머티리얼과
텍스처

캐릭터
애니메이션
시스템

충돌처리와
물리 시뮬레이션

네트웍 멀티플레이 기반의
게임플레이 프레임웍

언리얼 프로그래밍을 위한 언씬 커리큘럼

콘텐츠를 표현하는
각종 시각적 기능

언리얼 C++의 이해

게임플레이
프레임웍의 활용

네트웍 멀티플레이어
프로그래밍

게임플레이
어빌리티 시스템

언리얼 vs 유니티

언리얼 C++ vs 유니티 C#

게임플레이 프레임워의 활용

네트워 멀티플레이어 프로그래밍

게임플레이 어빌리티 시스템

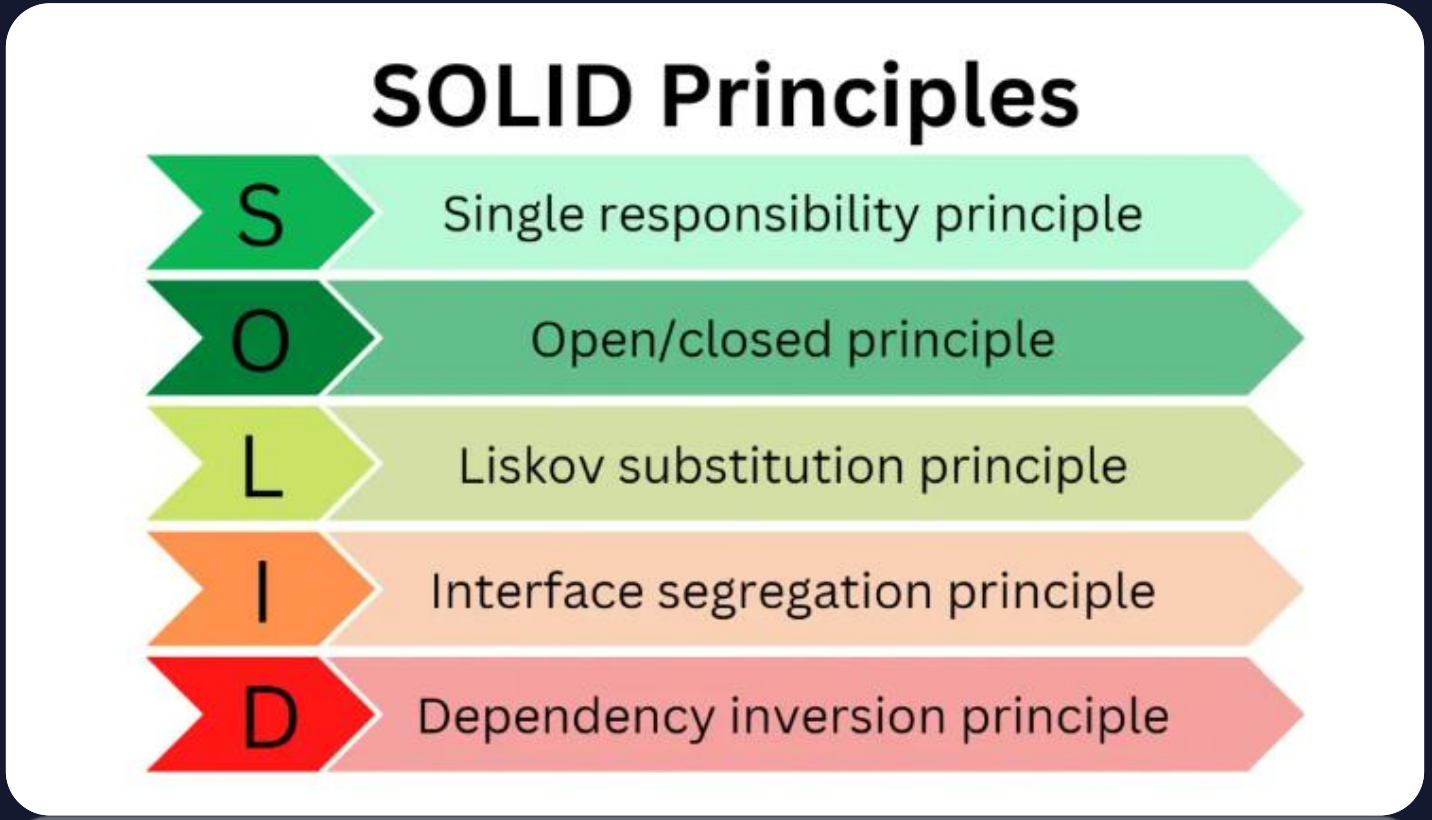
언리얼 엔진이 제공하는 장점

언리얼 엔진의 프로그래밍 공부 방법

- 언리얼 엔진의 기본 기능 숙지
- 언리얼 엔진 프로그래밍 익히기
  1. 언리얼 C++ 기초
  2. 언리얼 게임플레이 프레임웍
  3. 네트웍 멀티플레이어 프로그래밍
  4. 게임플레이 어빌리티 시스템 (GAS) 프레임웍

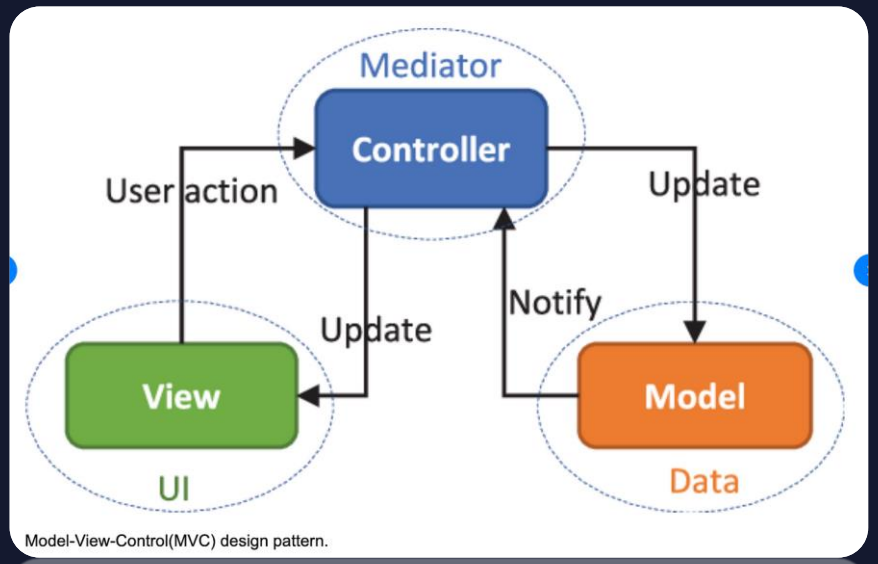
목표 : 방대한 게임 시스템을 어떻게 효과적으로 설계할 것인가?

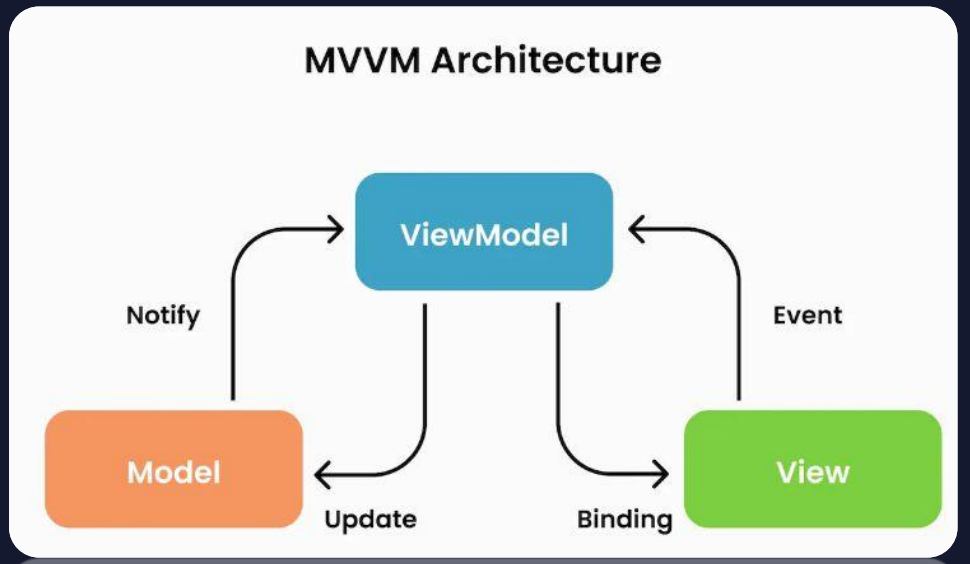
객체 지향 소프트웨어 원칙



프로그래밍 개발 방법론

- 객체 지향 프로그래밍 디자인 패턴 (GoF)
- 소프트웨어 아키텍처 모델의 등장
  - MVC
  - MVVM





하지만 게임 제작은?

게임에 특화된 개발 방법론

- 게임 프로그래밍 패턴 (게임에 특화된 디자인 기법)

https://gameprogrammingpatterns.com/contents.html

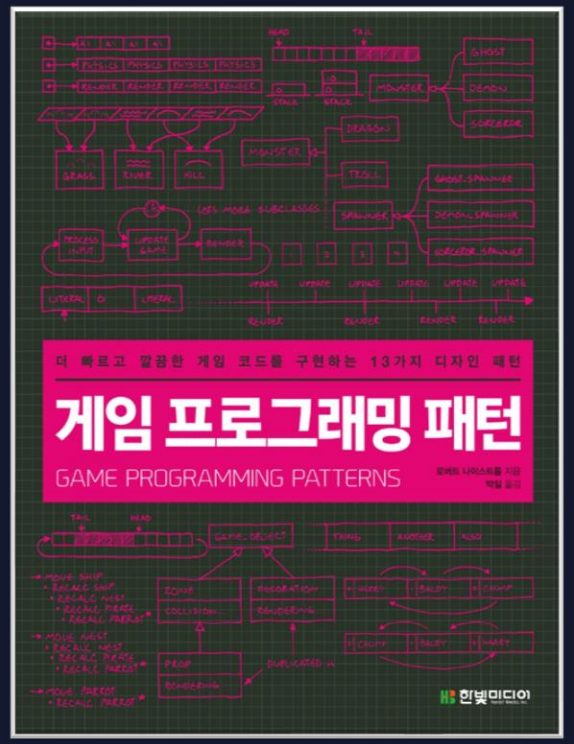
How can decoupling help?

While it isn't obvious, I think much of software architecture is about that learning phase. Loading code into neurons is so painfully slow that it pays to find strategies to reduce the volume of it. This book has an entire section on *decoupling* patterns, and a large chunk of *Design Patterns* is about the same idea.

You can define "decoupling" a bunch of ways, but I think if two pieces of code are coupled, it means you can't understand one without understanding the other. If you *de*-couple them, you can reason about either side independently. That's great because if only one of those pieces is relevant to your problem, you just need to load *it* into your monkey brain and not the other half too.

To me, this is a key goal of software architecture: **minimize the amount of knowledge you need to have in-cranium before you can make progress.**

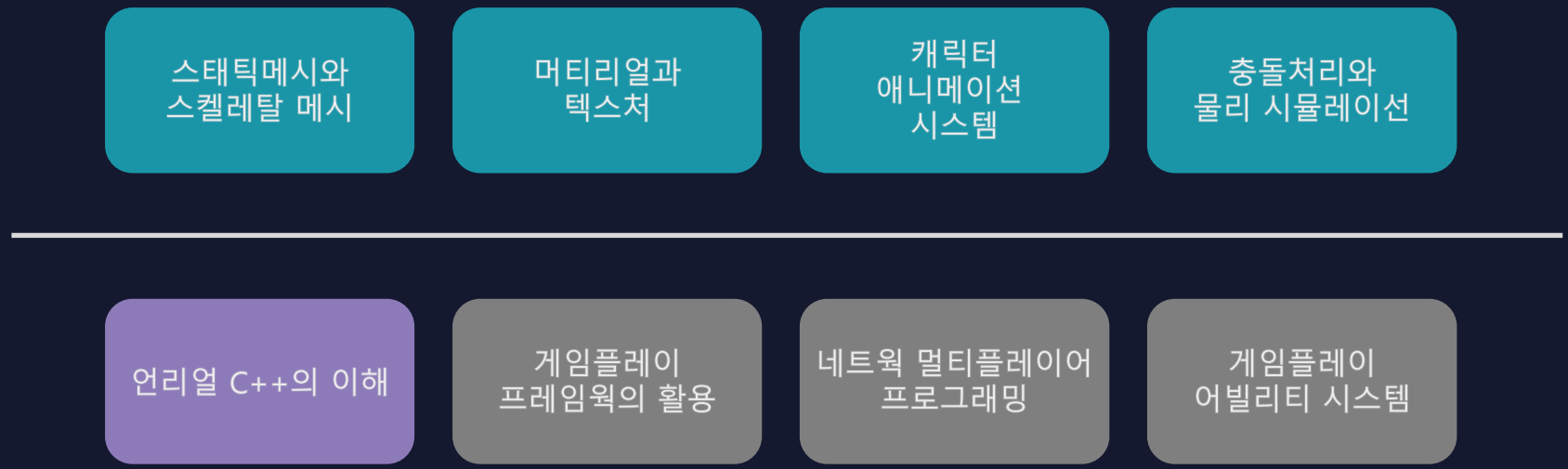
The later stages come into play too, of course. Another definition of decoupling is that a *change* to one piece of code doesn't necessitate a change to another. We obviously need to change *something*, but the less coupling we have, the less that change ripples throughout the rest of the game.



언리얼 엔진에서 게임 로직을 어떻게 효과적으로 설계할 것인가?

언리얼 엔진의 기초 학습

- 언리얼 엔진의 기본 기능을 익혀야 한다.
- 언리얼 C++을 제대로 익히고 활용하는 것만으로도 게임 코드를 효과적으로 관리할 수 있다.



게임플레이 프레임웍의 활용

- 게임 프로그래밍 패턴은 사실 게임 엔진 개발에 특화된 방법론.
- 게임 로직을 보다 효과적으로 개발하는 방법은 없을까?
  - MVC, MVVM은 화면 렌더링에 특화된 아키텍처
  - 언리얼의 게임플레이 프레임웍을 활용하기
  - 3단계 레이어 계층으로 게임 프레임웍 설계하기

스태틱메시와 스켈레탈 메시

머티리얼과 텍스처

캐릭터 애니메이션 시스템

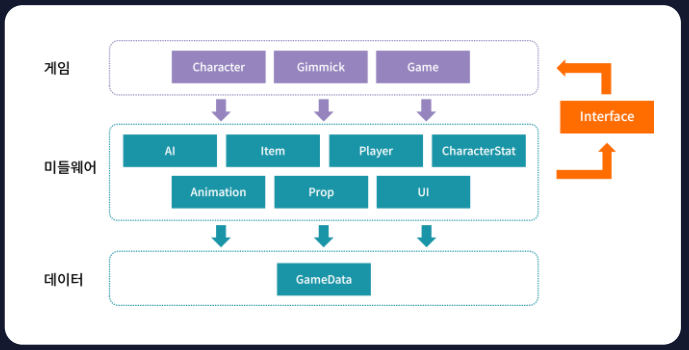
충돌처리와 물리 시뮬레이션

언리얼 C++의 이해

게임플레이 프레임웍의 활용

네트웍 멀티플레이어 프로그래밍

게임플레이 어빌리티 시스템



네트웍 멀티플레이어 프로그래밍

- 네트웍 멀티플레이어 게임은 거스를 수 없는 대세 흐름
- 네트웍 멀티플레이어 게임은 어떻게 만들어지는가?
- 소켓 프로그래밍이 아닌 클라이언트 – 서버 모델의 이해
- 언리얼 엔진의 캐릭터 무브먼트 컴포넌트의 분석

스태틱메시와 스켈레탈 메시

머티리얼과 텍스처

캐릭터 애니메이션 시스템

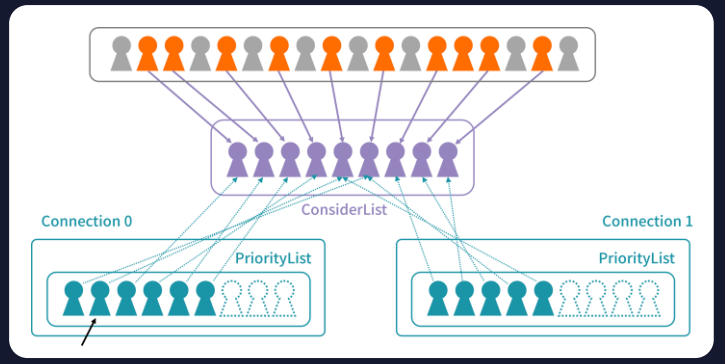
충돌처리와 물리 시뮬레이션

언리얼 C++의 이해

게임플레이 프레임웍의 활용

네트웍 멀티플레이어 프로그래밍

게임플레이 어빌리티 시스템



게임플레이 어빌리티 시스템은?

- 유연하고 확장성 높은 게임 시스템은 어떻게 만들어지는가?
- 기존 게임 개발 모델
  - FSM / HFSM (유한상태기계)
  - BT (행동 트리)
  - MVVM (UI 개발)
- RPG/MOBA/슈팅 등 대형 게임 개발에 필요한 요소를 디커플링하는 범용 방법론

스태틱메시와 스켈레탈 메시

머티리얼과 텍스처

캐릭터 애니메이션 시스템

충돌처리와 물리 시뮬레이션

언리얼 C++의 이해

게임플레이 프레임웍의 활용

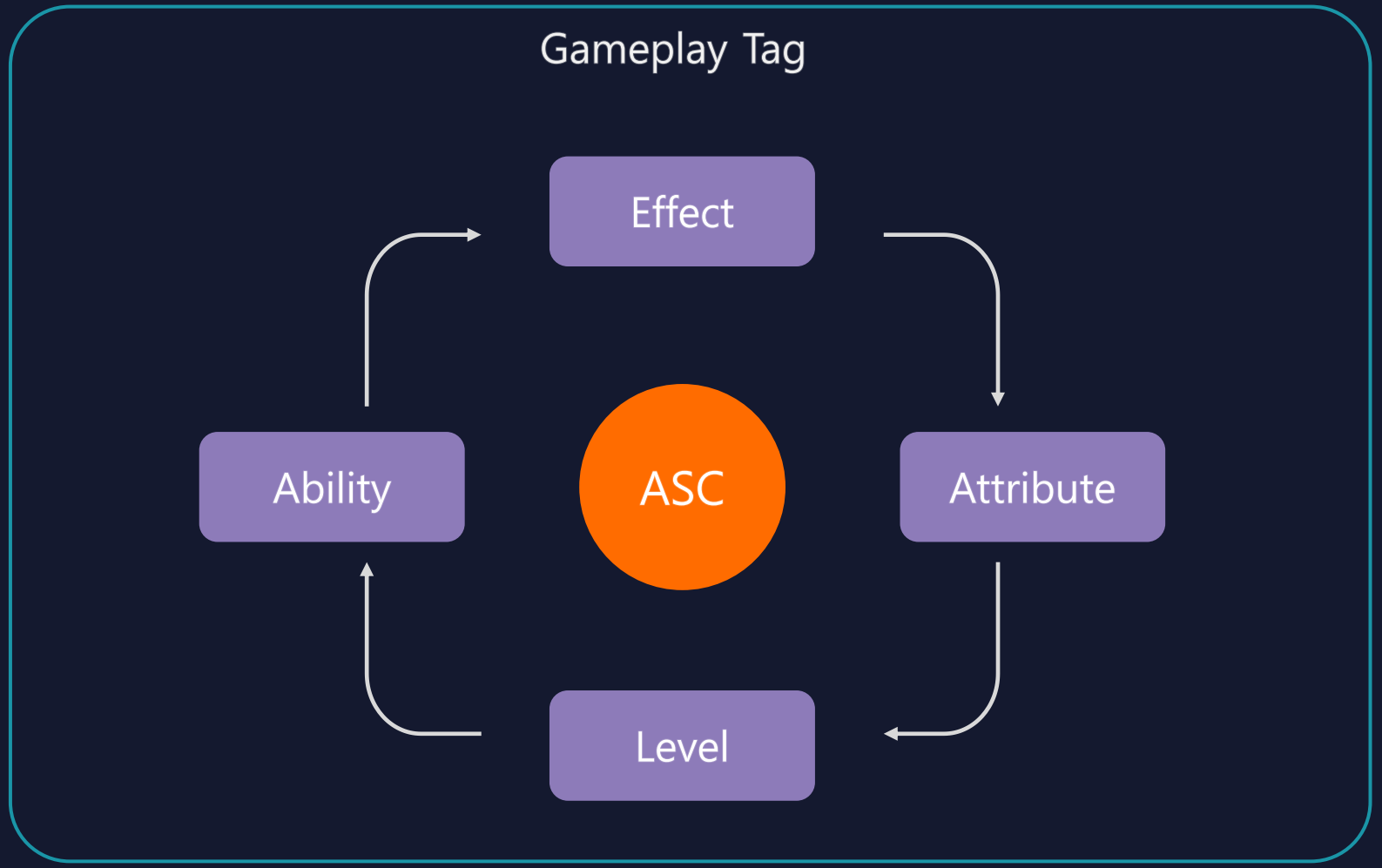
네트웍 멀티플레이어 프로그래밍

게임플레이 어빌리티 시스템

검증된 최대 동접 1200만 크리에이터 생태계

게임플레이 어빌리티 시스템의 아키텍처

- 앱 개발 아키텍처가 MVVM이라면 게임 개발은 GAS 아키텍쳐



정리

- 게임 개발에 입문한다면 잘 만든 롤모델을 참고하는 것이 효과적인 방법
- 최신의 소스코드까지 무료로 제공하는 언리얼 엔진을 최대한 활용
- 프로그래머로서 공부해야 할 내용
  - 효율적인 모던 객체 지향 기법 (C# , 언리얼 C++)
  - 기본 게임플레이 프레임웍
  - 네트웍 멀티플레이 게임프레임웍
  - 게임플레이 어빌리티 시스템