전투 시스템은 특정 플랫폼에서만 사용하지 않도록 범용으로 설계합니다. 서버와 클라이언트 모두 작동되어야 하며 언리얼 엔진뿐만 아니라 유니티와 같은 다른 엔진에서도 사용할 수 있어야 합니다. 이를 위해 C++ 기반 독립형 플러그인으로 구현하며, C# 과 같은 다른 언어 기반에서는 포팅(Interop) 을 하여 기능을 제공할 수 있어야 합니다.

**턴(Turn), 태스크(Tasks), 결정론적 처리(Deterministic Processing)**

턴제 전략 게임은 플레이어가 각자의 차례를 가지며, 턴을 획득하고 행동을 수행하고 턴을 반환하면 다음에 턴을 획득한 플레이어가 차례로 행동을 수행하는 방식을 취하는 게임입니다. 실시간 기반 게임이 아니므로 하나의 턴에서 수행하는 행동들은 모두 태스크화 하여 최소 1틱 내에 결정론적으로 처리될 수 있습니다. 태스크는 스케줄링 정책에 의해 큐에 담겨지게 되며 이 태스크들이 모두 처리되어야 1턴이 종료되는 것으로 간주될 수 있습니다.

**클라이언트 사이드 지연 처리(Deferred Processing)**

큐에 담기는 태스크는 애니메이션 또는 이펙트와 같은 렌더링 요소를 처리하지 않음에 유의해야 합니다. 순전히 소프트웨어적인 데이터 처리이며 렌더링 요소는 태스크 디스패처(Task Dispatcher) 를 통해 이벤트 형식으로 전달 받아 언리얼 클라이언트 단에서 후처리만 진행 할 뿐입니다.

**전투 시뮬레이팅, 랜덤 시드**

전투 프로그램은 플러그인으로 작성되고 전투 데이터만 처리하는 시뮬레이팅 기능을 하며 서버와 클라이언트 양쪽 모두 동일한 코드를 사용합니다. 또한 랜덤시드를 사용하여 무작위 수에 대한 동일한 결과값을 보장합니다. 다른 점이 있다면 사용 방법에 있어 클라이언트는 게임 진행간 실시간 사용인 반면 서버는 전투 종료 후 검증을 위해서만 사용합니다.

|  |
| --- |
|  |