TIH 메타버스 설계

# 용어

station == platform

pak == module

# 설명

## station 역할

* 서버접속 및 구체적인 기능

## station 기능

특징

* 라이프사이클 컨트롤
* 서버접속
* 비동기
* tickable 을 활용한 tick Base 점진적 실행
* Runable 을 활용한 멀티쓰레딩

로깅 및 기능 일반화

* UReplicationGraph 와 비슷한 매니지드 노드로 만들어진 오브젝트 풀링

특징

* + 해당 오브젝트 풀은 actor 나 uobject 를 카테고리화 시켜서 관리
  + tag 베이스

목적

* + - * Replication 관리와 커맨드 참조관리
* 커맨더를 통한 기능 일반화

컨셉

* + - max 나 maya 의 커맨드
    - 네트워크 protocol 설계

목적

* + - 기능 일반화를 통한 네트워크 동기화
    - 로깅
    - dataByte 시에도 문제 없이 변환 가능
    - Json 변화시에도 ustruct 기능 활용가능
    - 작업을 tick 베이스나 async 시에 사용가능하게 기능별로 묶는 커맨더패턴의 기능

커맨더의 기능

* + - * CreateLoad :
        + path 로드기능 async
      * CreateAssign :
        + 오브젝트 풀 기반 actor 제작기능 async
      * CreateAlloc :
        + 진짜 새로운 액터 제작기능 sync
      * ModifyTransform :
        + 움직임 sync
      * ModifyUpdate :
        + 다른 데이터 업데이트 sync
        + 범위로 업데이트시 async
      * DeleteCheck :
        + 삭제시킬 데이터를 확인하는 기능 sync
      * DeleteRange :
        + 범위 삭제기능 sync
      * DeleteList :
        + 리스트별 삭제기능 sync
      * ServerConnect :
        + 서버접속 async
      * ServerReConnect :
        + 재접속 및 서버 초기화 기능 async
      * ServerReques :
        + 서버에 데이터를 요청할때 사용하는 기능 async
      * ServerSend :
        + 서버에 데이터를 보낼때 기능 async || sync
      * ServerReceive :
        + 서버에서 데이터를 받을때 기능 async
      * ServerDisconnect :
        + 서버접속을 끊는기능 async || sync
      * ServerReplaction :
        + 서버에 send 뿐만아니라 데이터를 동기화 시킬때 사용하는 기능 sync

## pak 역할

* station 의 기능을 사용한 내부 개별 컨텐츠 구현
* 어디서든 접근 가능하고 유지되어야함
* level 간의 상호 컨트롤

## housingPak 기능

특징

* xml 파서
* 메타데이터 계층

목적

* + xml 로 만드는 맵

내부나 외부에서 컨트롤이 쉽게

데이터를 계층화 시켜서 구조가 변경되더라도 문제없게

데이터 파서는 어차피 커맨더의 기능 중 ModifyTransform 와 ModifyUpdate 를 사용한것

* managed확장 트리구조

목적

* + actor 와는 분리하여 actor
    - scene 계층화를 통한 statick batch 기능활용 및 자동 계층 transform
  + acotr 를 managed 로 컨트롤