# Cleania's system

* In-Unity (MonoBehavior)
  + [UI\_ItemContainer](file:///C:\Users\Admin\source\repos\CLEANIA\#UI_ItemContainer)
    - 옵저버(ItemStroage 구독)
    - 프록시(ItemStorage로)
  + [UI\_ItemController](file:///C:\Users\Admin\source\repos\CLEANIA\#UI_ItemController)
    - 프록시(UI\_ItemContainer로)
  + [AbilityStatus](file:///C:\Users\Admin\source\repos\CLEANIA\#AbilityStatus)
    - 퍼사드? (옵저버!)
  + [Status](file:///C:\Users\Admin\source\repos\CLEANIA\#Status)
    - 어댑터(StatusSO => AbilityStatus)
  + [Equipable](file:///C:\Users\Admin\source\repos\CLEANIA\#Equipable)
    - 옵저버(ItemStorage 구독)
    - 프록시(ItemStorage로)
    - 어댑터(ItemInstance\_Equipment들 => AbilityStatus)
* Out-Unity
  + [SavedData](file:///C:\Users\Admin\source\repos\CLEANIA\#SavedData)
    - 싱글톤
  + [ItemStorage](file:///C:\Users\Admin\source\repos\CLEANIA\#ItemStorage)
    - 브릿지, 옵저버(발행자)
  + [ItemInstance](file:///C:\Users\Admin\source\repos\CLEANIA\#ItemInstance)
    - 팩토리, 프로토타입
  + [ItemInstance\_Equipment](file:///C:\Users\Admin\source\repos\CLEANIA\#ItemInstance_Equipment)
    - .

## In-Unity

씬과 함께 생성  
씬과 함께 제거

* 게임 전체의 시스템을 맡을 수는 없음

#### UI\_ItemContainer

플레이어에게 노출되는 인벤토리

* 옵저버 (ItemStorage 구독)
  + ItemStorage가 변경되면 UI\_ItemController를 생성/제거
* 프록시(ItemStorage로)
  + this에 대한 추가/삭제는 직접 실행하지 않고 구독한 ItemStorage로 전달
    - UI\_ItemController가 주로 사용

#### UI\_ItemController

UI\_ItemContainer의 아이템 슬롯으로 나타남

* 프록시(UI\_ItemContainer로)
  + 유니티 이벤트에 따라 UI\_ItemContainer의 함수를 콜백

#### AbilityStatus

공격받고 공격할 수 있는 객체의 스탯 수치를 산출  
HP와 MP의 저장, 관리  
this.AttackedBy()를 통한 데미지 계산

* 퍼사드?
  + Equipable, Status, Buffable의 값을 모아 표면적인 스탯을 저장, 반영
    - get마다 매번 계산하기보단 **옵저버**로 변경하여 성능 향상 가능해보임

#### Status

레벨에 따른 기초 스텟 산출

* 어댑터
  + 레벨이 오름에 따른 스텟 변화 규칙이 다르기에 적합한 방식으로 테이블을 읽어줌
    - Status\_ArithmeticProgress, Status\_Table

#### Equipable

유닛의 착용중인 장비를 저장  
착용중인 장비의 스탯을 산출

* 옵저버 (ItemStorage 구독)
  + ItemStorage가 변경되면 ItemInstance\_Equipment를 가져와서 수치 반영
* 프록시 (ItemStorage로)
  + this에 대한 추가/삭제는 직접 실행하지 않고 구독한 ItemStorage로 전달
* 어댑터 (ItemInstance\_Equipment들 => AbilityStatus)
  + 여러 ItemInstance\_Equipment의 스탯 반영을 한차례 계산하여 반영할 수 있도록 산출, 저장

## Out-Unity

어플리케이션과 함께 생성  
어플리케이션과 함께 제거

#### SavedData

#### ItemStorage

#### ItemInstance

#### ItemInstance\_Equipment