(플레이 화면에 구현되어야 할 내용만을 담았습니다)

GameManager.cs

* 게임의 전반적인 수치를 관리하는 매니저 역할입니다.
* 게임 수치에 맞춰 UI 요소를 관리합니다(Private 함수)
* 플레이어가 버튼을 누르면 그걸 감지해서 HeadQuarter.cs의 Unit 소환 함수를 호출해야 합니다.
* 게임 내에서 GameManager Object의 Component로 작용합니다.
* 다음과 같은 기능을 제공하는 public 함수를 구현해야 합니다.

Public:

**Public Vector3 Get\_RGBValue()**

* Unit이 호출하며, 현재 환경의 RGB 값을 Vector3형으로 반환합니다.
* Input Parameter: None
* Output Parameter
  + Vector3(int Red, int Green, int Blue): 현재 환경값 수치를 반환합니다.

**Public void Destroyed(int Team)**

* 호출되면 Team 번호의 HQ가 파괴되었다고 간주하고 승리 판정을 내립니다.
* Input Parameter
  + int Team: HQ가 파괴된 Team의 번호입니다.
* Output Parameter: None

HeadQuarter.cs

* 각 HQ를 관리하는 서브 매니저 역할입니다.
* HQ의 체력, 오브젝트 소환, 피격시 체력 감소, 파괴되었을 때 GameManager에게 파괴되었음을 알려야 합니다. (GameManager의 Destroyed 함수를 호출)
* 모든 종류의 Unit을 Prefab화 하여 HeadQuarter.cs에서 정해진 종류의 Prefab을 소환할 수 있게 합니다.

Public

* **Public void Summon\_Unit(int ID)**
* GameManager에 의해 호출되며, 호출되면 ID 번호에 맞는 Unit GameObject를 생성합니다. 생성한 후 생성된 GameObject를 내부 변수(List)에 저장합니다.
* Input Parameter:
  + Int ID: 소환할 Unit의 미리 지정된 ID를 의미합니다. ID에 해당하는 종류의 Unit을 소환합니다.
* **Public void Hit(float damage)**
* **Public float Health()**

**Public void Pause\_Toggle(bool Is\_Pause)**

* GameManager에 의해 호출되며. 호출되면 게임이 일시정지됩니다. 모든 Unit의 동작을 일시정지합니다.(Unit.cs의 Pause\_Toggle을 호출)
* Input Parameter:
  + Bool Is\_Pause: 일시정지 여부를 나타냅니다, True면 모든 Unit을 일시정지하며, False면 일시정지를 해제합니다.
* Output Parameter: None
* Public void Finish\_Game()
* Public int Unit\_Count()
* Public void Unit\_Dead(GameObject Dead\_Target)

Unit.cs

* 모든 유닛의 AI 역할을 하는 코드입니다.
* 이동 / 공격 / 피격을 관장합니다.
* 각 Unit별로 고유의 공격Collider/ 피격Collider가 있으며, 공격Collider와 충돌한 Unit 중 특정 Unit의 Hit 함수를 호출합니다.

Public

**Public void Init(int Team)**

* Unit이 생성된 직후 HeadQuarter에 의해 호출되며, Unit이 속한 팀의 번호를 할당합니다.
* Input Parameter
  + Int Team: Unit이 소속된 팀의 번호입니다. 피아식별 할때 사용됩니다.
* Output Parameter: None

**Public void Hit(float damage)**

* 호출되면 피격되었다고 간주하고 체력을 damage만큼 경감합니다.
* Input Parameter
  + Float damage: 얼마의 데미지를 입었는가를 나타냅니다, 체력이 이 수치만큼 감소합니다.
* Output Parameter: None