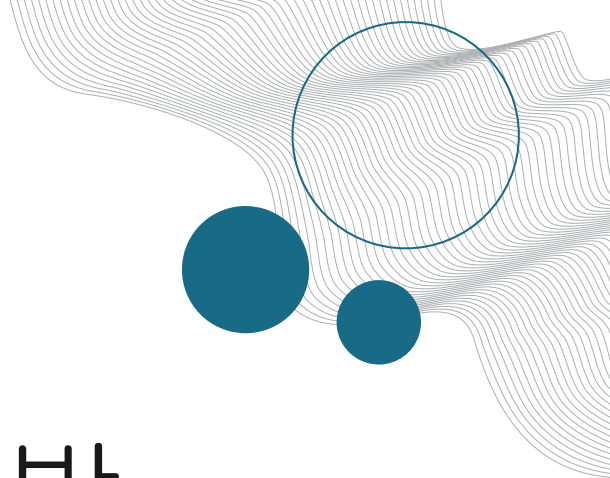


UE GAS 모방

Unity Action System



Player

HP : 100

MaxHP : 100

Q

E

Enemy

HP : 100

MaxHP : 100

"Character의 현재 상태를 기반으로
행동과 Action의 효과를 적용하는
상태 패턴 기반 시스템"

Status_Aiming



Attack



Defence



Dash



Skill1
Self harm

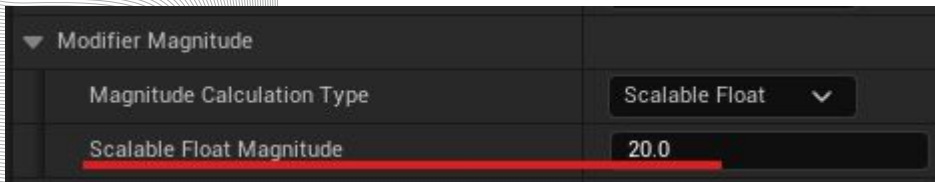
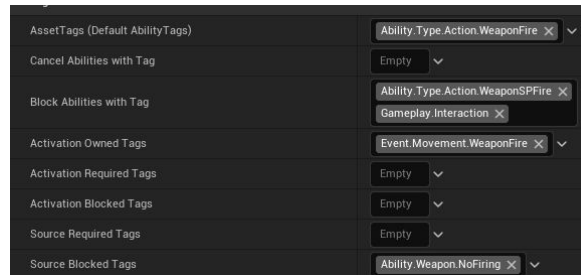


Skill2
Invincibility



UE GAS 분석

- UE의 Gameplay Ability System(GAS)의 여러 기능 중, 게임 내 능력(Ability)과 효과(Effect)를 관리하는 시스템을 분석하여 적용 함.
- 주로 RPG 및 액션 게임에서 캐릭터의 스킬, 상태 이상 등을 처리.
- 상태를 기반으로 Ability나 Effect들을 처리하는 **상태 기반 패턴** 사용.
- Action과 Effect의 수치 값을 변경하여 손쉽게 여러 기능을 구현하는 **데이터 중심적 패턴** 사용.



주요 기능요약



Player의 현재 Active 된 GameplayTag들을 기반으로 상호작용 및 Action을 관리

<u>Character</u>	World에서 행동하는 GameObject 관리 Class
<u>Action System</u>	Action을 관리하는 Main System, (GAS의 GameplayAbilitySystemComponent 역할)
<u>ActionData</u>	ScriptableObject로 ActionInstance 생성에 필요한 데이터를 관리 각 종 상세 Action 별 세분화
<u>ActionInstance</u>	Tag를 기반으로 다양한 상호 작용 및 관리 되는 기본 Class 다양한 기능을 간편하게 추가 (GAS의 GameplayAbility역할)
<u>ActionEffectInstance</u>	대상에게 특정 효과를 부여하는 기본 Class (GAS의 GameplayEffect)
<u>Health System</u>	Character의 체력 관리 Class



작업 환경



작업 기간

2025/02/25 ~ 2025/03/02 (6일)



Unity 6



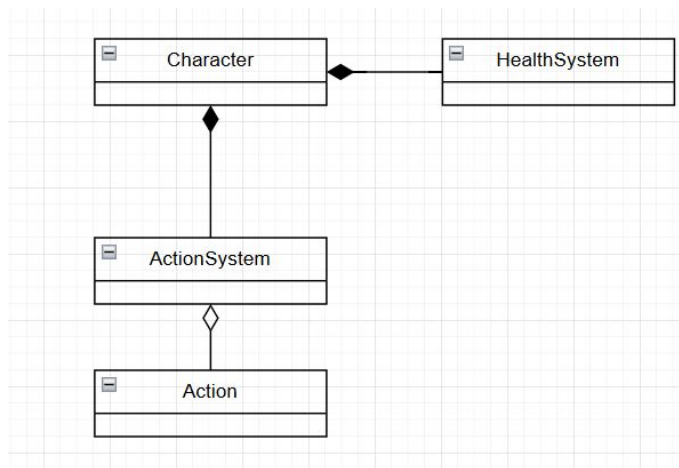
Visual Studio 2022

Github



Character

- CBD 기반으로 Character 가 ActionSystem 을
Action 들은 적용 할 ActionEffect 들을 각 각 필요에 따라 적용
- 이동 및 Aiming
 - Player Controller를 이용해 이동 제어
 - Aim Controller를 이용해 Aiming 제어
 - Free Camera와 Aiming Camera를 이용해 Aiming Zoom 적용
- Health System
 - Character의 체력 관리.
 - 체력 변동에 따른 Event 호출.
- Action System
 - Character가 행동 하거나 다양한 영향을 받을 Action관리.



Action System의 Action 처리 로직

// Action의 Unique성을 알 수 있는 Tag로 Action 실행

- StartActionByTag

// Start 하려는 Tag의 Action 실행 가능 여부 확인

- if isCanStart == false

- return

// Start 하려는 action의 tag에서 취소하도록 정의 한 tag들의 action 취소 처리

- Stop actions by start action cancel tags

// 새로 들어온 action 실행

- Start action

// 실행 후 바로 해제가 필요한 action은 해제 처리

- if AutoStopAfterOnce == true

StopAction

Action Instance 생성 로직

기본 동작들을 정의 하고 초기에 미리 세팅

- **List<ActionData>**를 미리 **SO**를 통해 **Load**

// List 내에 있는 ActionData를 Add Action

- **ActionSystem.AddAction**

// ActionData를 이용해 Instance 생성

- **ActionData.CreateInstance**

// 기존 보유 여부 확인 후 없을 시

- **Add New ActionInstance To ActionDic;**

// Auto 실행 및 Can Start Action 확인 Action Running 설정

- **Start Action**

Action Instance 로직

- Start Action

// 실행 Action에 정의 된 실행 주체에 부여해야 될 Tag 부여

- Set GrantsTags by Action Grants Tags

// Action Running 설정

- Set IsRunning Action

// 실행으로 인한 Cooltime 적용

- Set Cooltime

// Action 실행 시 설정정 해 줘야 되는 Animation Data 설정

- Apply Start animation Datas

// 부여해줘야 되는 ActionEffect가 존재 시

- if Action Implement Apply Action Effects

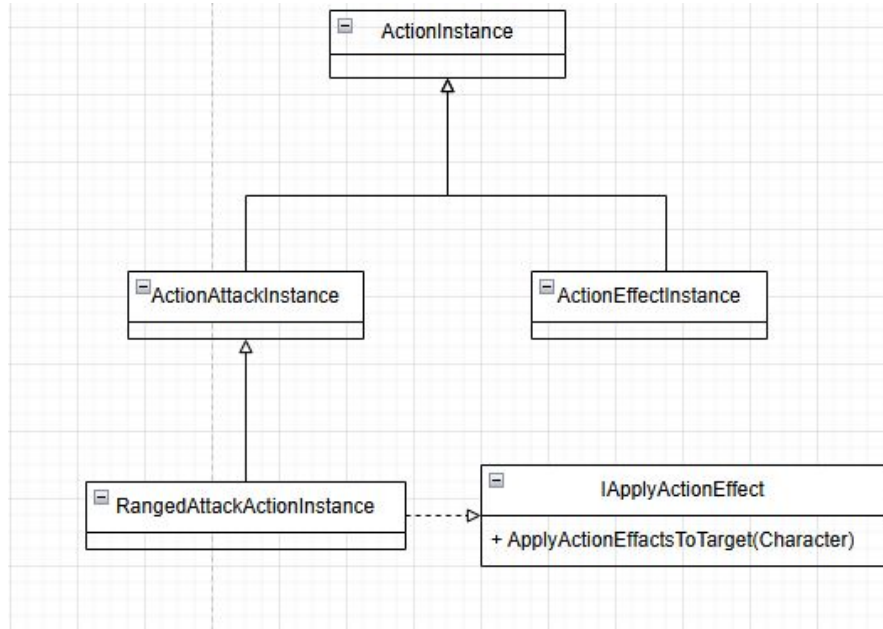
// 부여돼 있는 ActionEffect를 Target에게 부여

- Interface.ApplyActionEffects

// 실행 후 바로 중지 필요 시

- if Auto Stop == true

- Stop Action



Action Instance 로직

- Stop Action

// Active에서 부여 했 던 Tag 회수

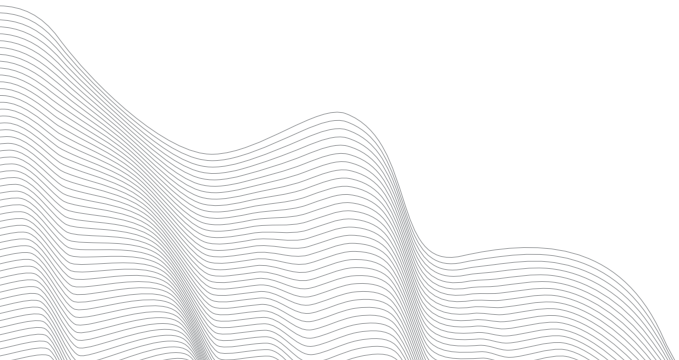
- Unset GrantsTags by Action Grants Tags

// Action Not Running 설정

- Set Is Not Running

// Stop 시 설정 해 줘야 되는 Animation Data 설정

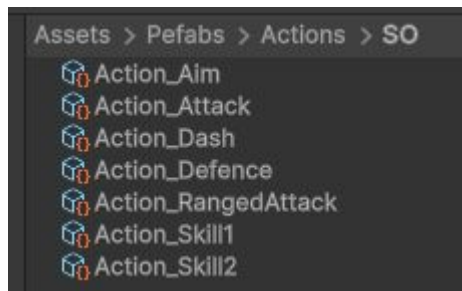
- Apply Stop animation Datas



Action Values

- **Action Tag**
 - Action의 Unique용 Tag. 해당 Tag를 이용해 Action 적용 가능
- **Grants Tags**
 - 해당 Action 소유주에게 부여 할 Tags 정의
- **Cancel Tags**
 - Action 수행 시 활성화 돼 있는 Action을 저지 시킬 Tags 정의
- **Blocked Tags**
 - 해당 Action 실행 전 Blocked Tags로 정의된 Tag를 Action 소유주가 보유 시 해당 Action 실행 불가할 Tag 정의
- **Auto Start**
 - Action Add 시 자동 실행 여부
- **AutoStopAfterOnce**
 - 1회 실행 직후 해제 여부
- **ApplyActionEffects**
 - 특정 Target에 ActionEffects를 적용할 List.
적용 이 필요한 Action은 IApplyActionEffectsInterface를 상속 필요
- **Cool Time**
 - 각 Action 별 쿨타임 설정. 쿨타임 진행 중일 시 스킬 재사용 불가
- **Start, Stop Animation Datas**
 - 각 상황 별 실행, 중지 할 애니메이션 데이터 세팅을 정의

생성한 Actions



<u>Action Aim</u>	조준을 위한 Action 부여할 Status 정의
<u>Action Attack</u>	공격력 시 필요한 데미지 및 애니메이션 등 정의
<u>Action Defence</u>	방어를 통해 데미지 경감. Input Action 설정을 통해 Hold 가능하게 적용
<u>Action Skill1</u>	자신에게 데미지를 주는 Case 확인을 위한 자기 피해형 스킬. 특정 시간 동안 주기 별 데미지 적용
<u>Action Skill2</u>	지속 시간 동안 모든 데미지 무효화 Status 적용
<u>Action Dash</u>	Dash 용 애니메이션 데이터 및 무적 상태 적용.



Action Effect 로직

- ApplyGameEffect

// 지속시간 설정 시 지정 시간 후 action stop 호출

- if duration > 0

- Start call execute stop action Coroutine

// 최소 1회 Action Effect 적용

- execute period Effect

// 설정 한 주기 시간이 지날 시 Action Effect 적용

- if period > 0

- Start call execute period Effect Coroutine

-Stop ActionEffect

// Duration 코루틴 중지

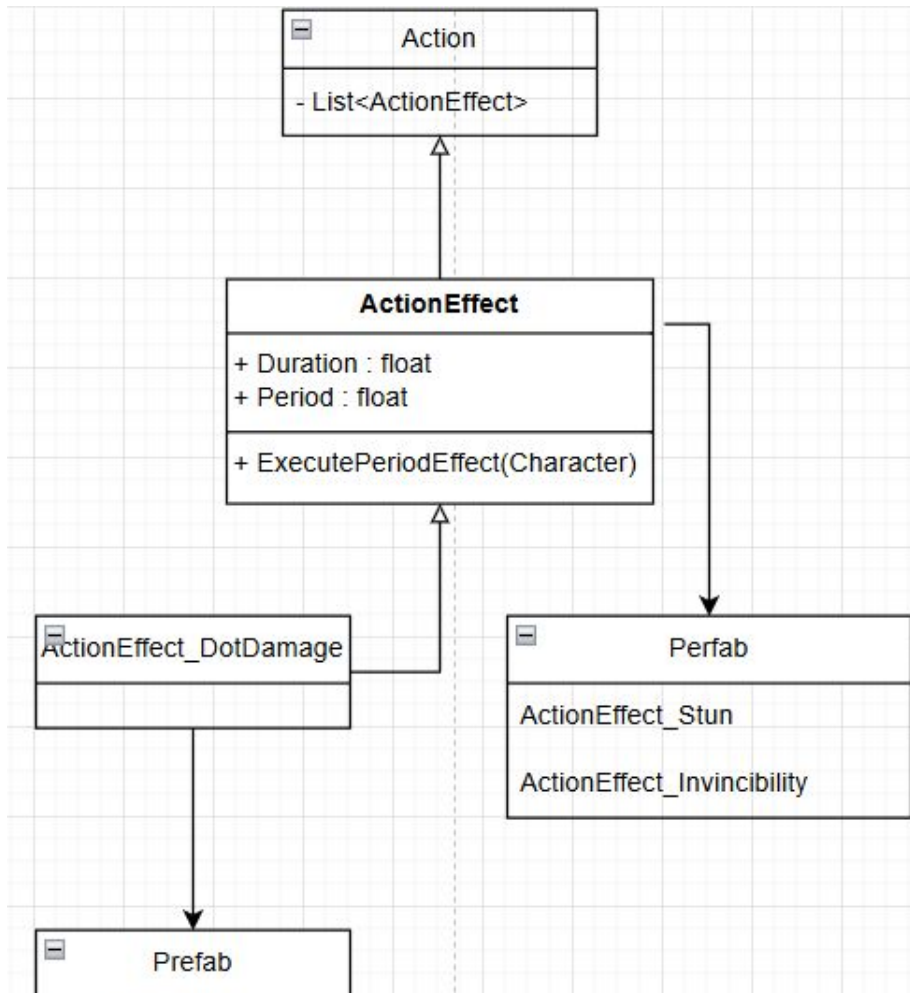
- Stop Duration Coroutine

// Period 코루틴 중지

- Stop Period Coroutine

// ActionEffect 제거

- RemoveGameEffect



Action Effect Values

- **Duration**
 - ActionEffect를 적용할 시간
- **Period**
 - 해당 ActionEffect를 적용할 주기
- **Total 적용 횟수**
 - $\text{Duration} / \text{Period} + 1$

생성한 Action Effects



<u>ActionEffect_Bleeding</u>	출혈 효과. Target의 체력을 dot damage로 특정 시간동안 특정 주기의 회수로 감소.
<u>ActionEffect_Invincibility</u>	무적 효과. 적용 Target의 체력 감소효과를 무시



UI

Game Input Detail

- Input 설명용 UI

Player Gameplay Tag Print Area

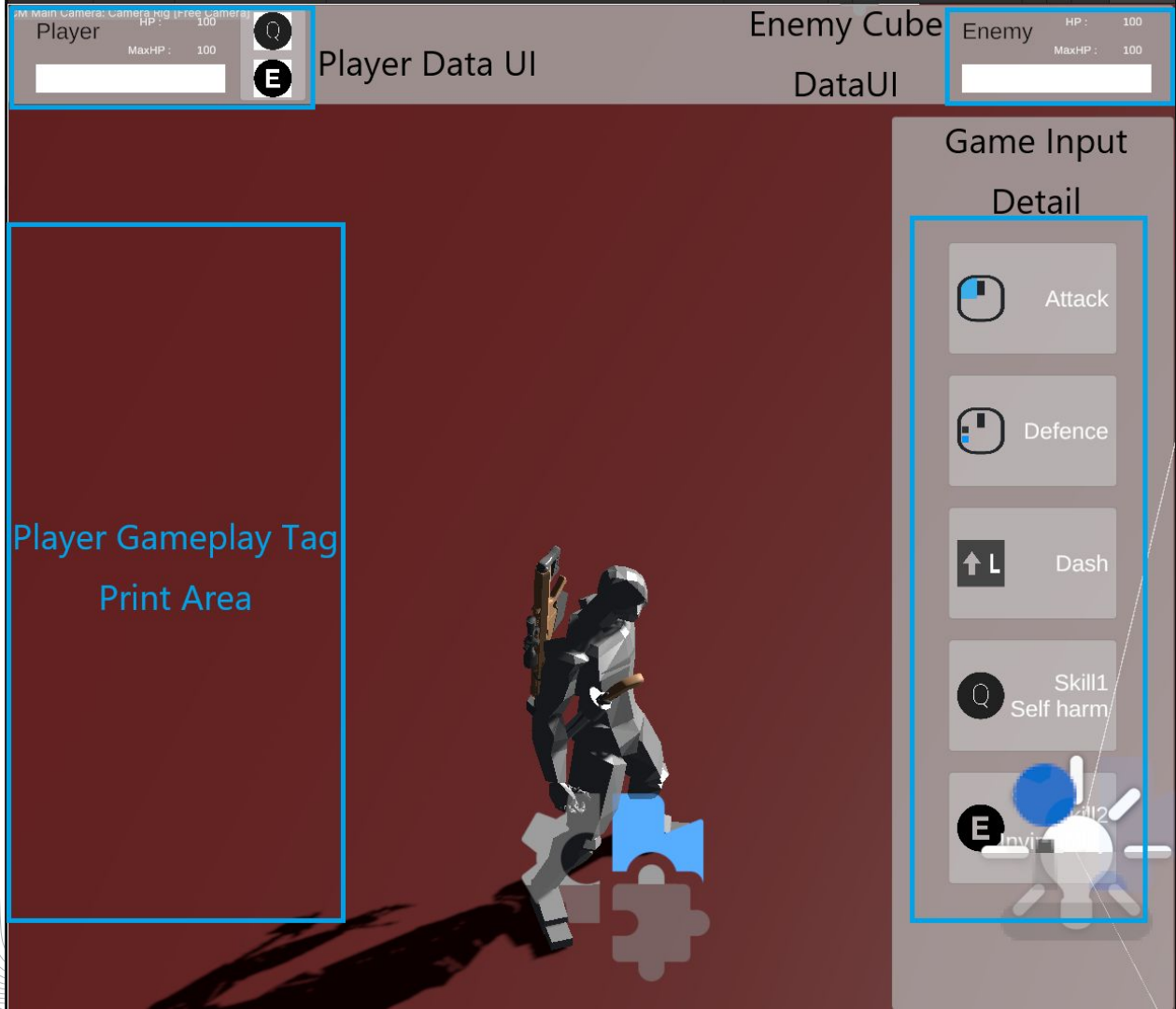
- 현재 Player가 보유중인 Tag Print

Enemy Cube Data UI

- Character 기반 HP 관리

Player Data UI

- Character 기반 HP 관리
- Skill Cooltime에 따른 UI Update



UI

HP UI 관리

- Character의 HP를 관리하는 HealthSystem에서 HP가 변경 될 때 bind 된 System.Aciton을 Invoke
- 실제 변경이 있을 때만 UI 갱신이 되는 Event 기반 UI 갱신 적용

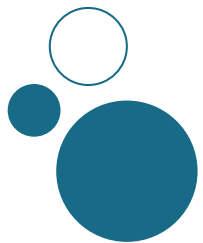
```
public void ApplyHealthChange(Character damageCauser, float damageAmount)
{
    if (blsInvincibility == true && damageAmount > 0)
    {
        return;
    }

    if (IsAlive() == true)
    {
        float curHP = hp;
        float newHP = Mathf.Clamp(curHP - damageAmount, 0, maxHP);
        hp = newHP;

        if (OnHealthChanged != null)
        {
            OnHealthChanged.Invoke(curHP, newHP);
        }

        if (OnDeath != null && IsAlive() == false)
        {
            OnDeath.Invoke();
        }

        Debug.LogFormat("Damaged!! Damage Causer : {0} Amount : {1}, Cur HP
    }
}
```



감사합니다.



For more info:
Github | BLOG

