OHL Technical Document

# Combat System Manager

Description: This is based on Unity Event system. This manager will treat all kind of actions (combat) in our project. The basic system is in CombatSysMgr class. And all Actions are in Action class. And there is a helper class for calculate for actions.

# Details

**CombatSysMgr.cs**

Description: This script treats basic combat system which is based on Unity Event.

|  |  |
| --- | --- |
| Class | Description |
| ActionEvent | Base actions |
| Action | Actions which take base actions |
| CombatSysMgr | Manage Actions and ActionEvents |

**class CombatSysMgr**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Value Name | Description |  |
| Dictionary | actionDic | Dictionary of actions |  |
| Dictionary | actionEventDic | Dictionary of base actions |  |
| CombatSysMgr | instance | Treat all components |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Function Name | Parameter | Description |  |
| Init | X | Initialize dictionaries |  |
| GetActionEvent | String | Get Action event which has same name. |  |
| TriggerAction | String, GameObject | Trigger action which has same name. |  |
| TriggerActionEvent | String, GameObject | Trigger action event which has same name. |  |

**Action.cs**

Description: All actions which our project needs are in this script.

**class Action**

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Parameter | Description |  |
| AddActionsToDictionary | X | Add new actions and action events to dictionary. | It should be updated when new action or new action event is added. |
| DealPhysicalDamage | Entity | This function will calculate physical damage and trigger events |  |
| DealPhysicalComboDamage | Entity | This function will calculate elemental damage and trigger events |  |
| SelfHeal | Entity | This function will calculate heal point and trigger events |  |
| TargetHPCheck | Entity | This function will check target and player is alive. |  |
| GetPhysicalDamage | X | Take damage |  |
| GetRecoveryValue | X | Take heal point |  |
| GetOGCDDuration | X | Take duration of OGCD |  |
| GetOGCD1Timer | X | Take first skill’s OGCD timer |  |
| GetOGCD2Timer | X | Take second skill’s OGCD timer |  |
| GetGCDDuration | X | Take duration of GCD |  |
| GetGCDTimer | X | Take GCD timer |  |
| More Functions will be added. Calculate OGCD, calculate effects, etc. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Value Name | Description |  |
| Int | comboCount | The number of combo attack |  |
| Float | GCD | GCD duration |  |
| Float | OGCD | OGCD duration |  |
| Float | EnemyAttackCooltime | Duration of enemy’s attack |  |
| GameObject[] | Player | Player |  |
| GameObject[] | Enemy | Enemy |  |

**ActionHelper.cs**

Description: All calculate functions for actions are in this script.

**class ActionHelper**

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Parameter | Description |  |
| DamageMod | Entity, enum | Damage modify based on entities of user and target |  |
| IsCriticalHit | Entity | Check critical hit is occurred based on user’s entity. |  |
| These functions are not used at this time. | | | |

# Event Names

**Actions**

|  |  |  |  |
| --- | --- | --- | --- |
| String | Description |  |  |
| DaggerStrike | User attack target with physical value | | |
| AutoAttack | User attack target automatically | | |
| SelfRecovery | User heal itself | | |

Actually, auto attack is called in Actions.cs.

**Specific behavior**

|  |  |  |  |
| --- | --- | --- | --- |
| String | Description |  |  |
| TakePhysicalDamage | Calculate damage with user’s and target’s physical entities | | |
| OGCD1\_Init | When first skill timer is initialized, this event will be called | | |
| OGCD2\_Init | When second skill timer is initialized, this event will be called | | |
| GDCInitialize | When GCD is initialized, this event will be called | | |
| SelfHeal | This event will be called when player use heal skill | | |
| PlayerIsAlive | This event will be called if player is alive | | |
| PlayerIsDead | This event will be called if player is dead | | |
| EnemyIsAlive | This event will be called if enemy is alive | | |
| EnemyIsDead | This event will be called if player is dead | | |

# Sample how to use

public void FireAction()

{

CombatSysMgr.TriggerAction(actionName, owner);

}

This will be used in Button. Find action which has same name and trigger that event.

ActionEvent Physical = new ActionEvent();

if (CombatSysMgr.actionEventDic.TryGetValue("TakePhysicalDamage", out Physical))

{

Physical.AddListener(GetDamage);

}

void GetDamage(Entity e)

{

if (owner.GetComponent<Entity>() == e)

{

m\_text.gameObject.SetActive(true);

m\_text.text = Actions.GetPhysicalDamage().ToString();

}

}

TryGetValue is find the event which has same name. And you could add listenr. This is sample of damage spawn.