ACS

ACTIVE COMPOSABLE SKILL

Acs Spirits

ACS Spirits as a game, aims to be a **generic base project** for other games to re-skin

ACS Spirits will always be the most updated version out of the other products. will include extra features like **Monster AI** and **Nest spawn system**, **DialogueSystem integration** and **server-wide quests**, **vending,** amongst many other small yet important features that add up into a fully fledged game.

ou can simply plug and play, change our pre-existing assets, add your models, fidget with some numbers, create a storyline, add NPC, Monsters, your own terrain, your own skill trees and skill sets and you'll be able to get your next idea into as soon as possible.

Acs SkillsOnly

Currently Unavailable.
An add-on built for uMmorpg and Mirror without the additional sprinkle of paid 3rd party assets. Using default Unity Navmesh for Pathfinding and ScriptableBrain from uM. Our ACS Essentials Character animator and Skill Model Pipeline intact to create the essence within your game.

Acs Single Player

For singleplayer games, not for uM/Mirror. This one will go to Unity Asset Store, containing only the barebones of the composability and very basic skill effect implementation, that can be customized for 2D or 3D

Introduction:

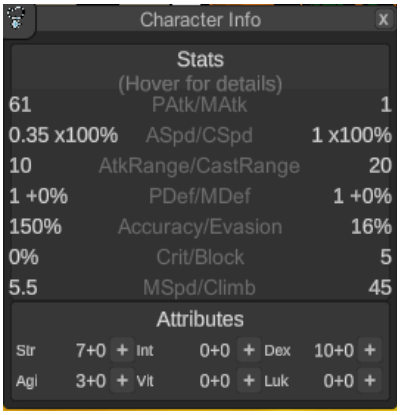
We've built an overarching add-on for uMMORPG Remastered, concerning the composition of skills with intricate effects, that can be applied to **Players**, **Monsters**, **NPC**, and **more**!

We've added and linked **Complex Secondary Attributes, Hybrid Player Movement, Nest Spawner & Minion Controller, Cinemachine & DialogueSystem Integration** together to improve gameplay, visuals and experience.

With no-code Unity enthusiasts in mind we've built **an Editor for Skill, Monster AND Nest Spawn Management.** Simply Add VFX, Add Status Effect, Choose Action type, Voila! using ACS Addon for uMMORPG can't be easier. It's about time to give your game a superboost.

Product Overview: ACS Spirits

We'll start from the very beginning, character.
The option of choosing the 6 different elements.
Fire, Water, Air, Earth, Light & Dark



Primary Attributes:

STR: Strength
AGI: Agility
INT: Intelligence
DEX: Dexterity

ACS
ACTIVE COMPOSABLE SKILL

Secondary Stats:

Physical Attack - PAtk | Physical Damage
Magical Attack - MAtk | Magical Damage
Attack Speed - ASpd | reduce cast time for physical skill. Also reduces cooldown
Cast Speed - CSpd | reduce cast time for magic skill. Also reduces reload

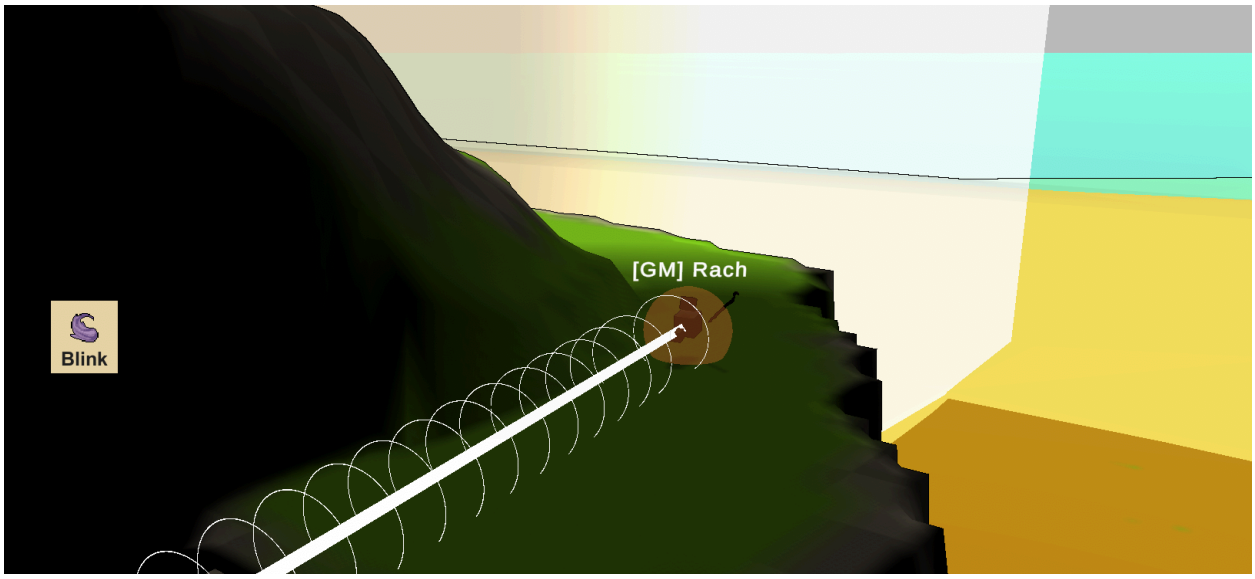
Attack Range - AtkRange | Attack Range for Physical Attacks
Cast Range - CastRange | Cast Range for Physical / Magical Skills
Physical Defense - PDef | Defense against Physical Attacks / Skills
Magical Defense - MDef | Defense against Magical Skills
Accuracy - Accuracy | Chance to pierce opponent's evasion
Evasion - Evasion | Chance to evade incoming damage (Physical / Magical, Attack / Skill)
Critical - Crit | Chance to land an attack that ignores the target's defense
Block - Block | The amount of damage nullified when an attack is parried
Movement Speed - MSpd | Distance traveled over time
Climb - Climb | Power to climb slopes in terrain

Bounces.mp4

Various types of bounce for projectile, mainly with these 2 distinct behavior: Can bounce to same target, can bounce to each target multiple times

Omnislash, built from several actions including ActionTeleport on the caster

ACS
ACTIVE COMPOSABLE SKILL

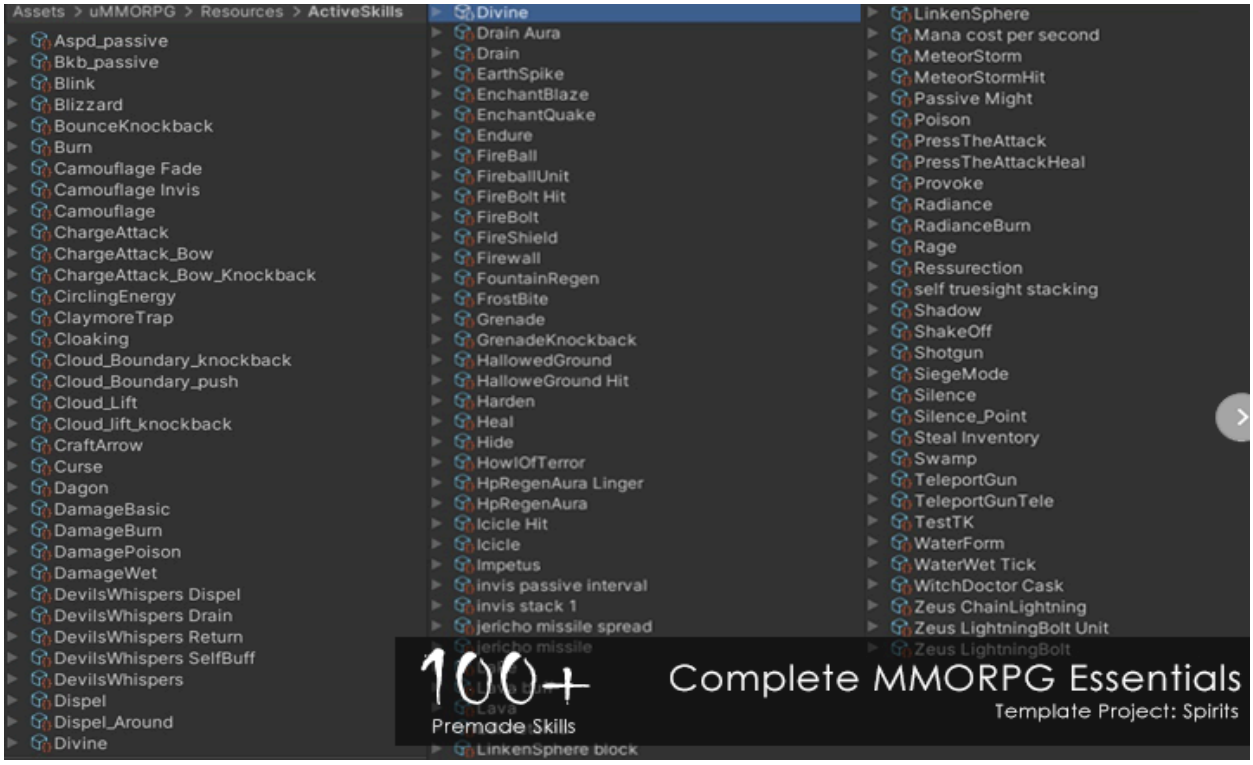


Blink Skill, Using the 'Teleport' Action Definition

Skill prerequisites:
Depending on activation context (passive/single/toggle on off/hold) and combo replenish mode (Manual/Auto/NoTick), these values may be used differently in the casting process

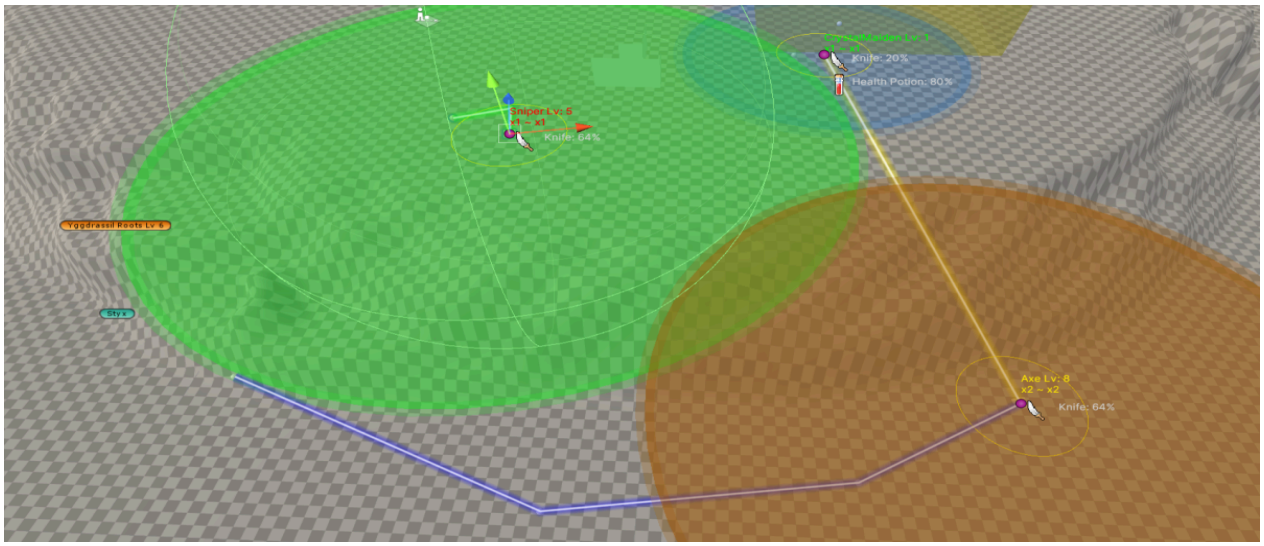
Cast Time | Time to cast a spell
Charges | One charge is necessary to cast skill, Skills may have multiple charges.
Cooldown | For Auto replenish, time to gain 1 combo for the next cast.
| For Manual, time before the player can reload the combo to max
Reload | Only for Manual replenish, the time to complete a reload to gain max combo back

Action Definition:
As the name states, we use this to define what a skill will do, stack them as you see fit and you've got a spell. Here are Action Definitions we have so far; Projectile, Damage, Melee, Parry, Buff, Shield, Steal, Particle, Modifier, Aoe, Knockback, Teleport, Dispel, Craft.



Hybrid Player Movement

The custom uMMORPG movement was a bit blocky in our opinion, so we have integrated our version of player movement. What it allows you to do is to move using the Keyboard, using the WASD keys or using the Mouse: point and click. This includes Swapping camera mode with [TAB], from Cursor mode -having to drag right click to rotate camera-, to Shooter mode where cursor is locked in center and camera follows mouse movement.



Monster AI & Nest Spawn System

Technical Overview:

ACS involves a set of custom editors for the composition of the data-oriented skill authoring, and the implementation of that data as skill effects in a networked game environment

The goal of Acs is to be able to handle the authoring of complex skill/gameplay mechanics inspired by RagnarokOnline, Dota, GuildWars, OnePiece, and typical Isekai anime genre where the skill effect significantly matters for the narrative like SAO, Konosuba, ShieldHero, Spider/Slime, and so on.

Index:

ACS
ACTIVE COMPOSABLE SKILL

Structure:

`ActiveSkill` : ScriptableSkill is the base class you'll deal with, that gets passed around, as a data container
[Pic of Acs inspector SO in project]

Developer notes: *An `Acs` holds some values of its own, and also sub-`SO` to further define the SkillTarget and list of `ActionDefinition`*
When casting an `Acs`, some other `Acs` overrides may be provided to select which one will actually get casted during certain situation, such as the weapon equipped, directional, whether it casts with the default targeting or specifically casted on a unit, etc

In an `Acs` , it determines the amount and therefore the amount of successive casts of this skill. It also can determine the power level or animations to this part of the "combo sequence"

Developer notes: *An `Acs` can invoke several `AcDef` , and some `AcDef` can chain invoke another `'Acs'`. The complete number formula may keep track of the source caster or source `'Acs'`. This is especially important to support say, a projectile being shot by a caster with +20% damage to fire elements, who happens to have a buff from an accessory to give further +10 damage, entering a target's shield area and applies the damage, breaks the shield, and dealing the remaining damage to the shield's owner who happens to be a fire element*

`NumberFormula` : The class that handles how much a number is, taking influence from the caster's stats, element, skill/passives, charge, previous number result from previous `Acs` chain.

Developer notes: *Whenever there's a number involved, most of the time there's a `'NF'` to define it. `'NF'` also handles the Order of Operation in the inspector. Simply define the formula on one of the list index, define another part in a second entry, and on the third entry, reference the result of first and second entry in any order/times necessary*

`Thinker` :
a Thinker is a networked object with its own target detection, movement, and other behavior. Thinkers are also necessary for server persistence. ThinkerBuff, only spawned if the buff deals with external targets
ThinkerAoe, defines an area with shape (Box, Circle, composition of Capsule for rivers), on which an `'Acs'` gets applied during its lifetime. Interval, OnEnd, OnStart, etc. Supports "

- ThinkerProjectile,
- ThinkerParticle,
- ThinkerShield

Developer notes: *An `'Acs'` can have multiple effects. Some `AcDef` may spawn a thinker. ersistence necessary in case the caster logs out behind 5 hour long duration StormGust Everything is definable with multiple AcDef Self `buff` while stealing a potion from all targets in an area? Area `buff` with different effects depending on the targets, and also different on self?*

Full list:

- *Damage*, if it involves changing the target's energy, then it's this. Hp/mana/stamina, and also supports ActionModifier from a previous 'Acs' chain for further defining the formula.

- *Projectile*, shoots a ThinkerProjectile. Supports more sequential logic such as bounces, OnEnd, OnDistance. Can have hitCount, hits once per target or allow multiple hit instances, interval between able to be hitted again, and option for homing.

  Developer notes: *TargetAim (shooting to cursor), shoots projectile to mouse direction*
  TargetPoint (spawns along ground), shoots either forward, world upwards, or ground normal upwards(like EarthSpike)
  For 'Acs' that shoots more than 1 projectile per target, consider using ActionParticle
  Some option of movement include custom physics, tween, or homing

- *Melee*, targetless swing/stab based target acquisition. Option for animation-collider based or ray sweep-cast based.

  Developer notes: *Specially interacts with Parry, or an option to be undeflectable by Parry by defining it in the AllowedTarget. A GhostSword might be unblockable*

- *Parry*, placing a collider to block incoming melee or projectile, at the cost of the caster's stamina.

- *Buff*, applies the effects on the target, optionally for the area around the target too (aura effect).

  Developer notes: *Put an 'Acs' with Damage in interval and you get Dota Pudge's Rot.*
  All Buffs consist of Charge and Duration. If the effect power is constant, then just keep the maxCharge at 1
  Upon applying a buff, certain type of stacking behavior will be considered.
  RefreshDurationCharge sets the duration and charge at that number
  AddDuration stacks the duration. Charge must stay at 1 for this one
  RefreshDurationAddCharge sets the duration to the max duration, but adds the charge by the new buff instance's charge value
  FullStack means each buff instance have their own duration, and the charge for this is accumulated to make the total strength
  A Buff can have multiple preset tags that is StatusEffect, which is used for interactions with dispels, included in NumberFormula value, screen visual effects, and some hard coded effects like silence disabling the casting of spells regardless the total charge of this 'SE'

- *Shield*, the "charge" of a Shield is really the remaining health of the shield. An option to spawn a ThinkerShield for a force field effect

- *Steal*, attempts to take an item from the target(s).

  Developer notes: *Steal chance formula is currently somewhat hard coded but really, everyone got their own take on this*

- *Particle*, spawns a ThinkerParticle that continues the handling of spawning particles.

  Developer notes: *The particle simulation is authored in Unity's ParticleSystem, and therefore deterministic enough in different machine across the network. An option to spawn particles that homes to target(s), but doing this means definitely not as deterministic across network*
  Spawning hundreds of particles from a single ActionParticle is pretty much 100x much cheaper than spawning 100 ActionProjectile. Movements of each particle are simulated per client and on the server, the only thing sync'd is the alive/dead of each particle, which is synced in bits, 8 particle per byte
  Some options are similar to ActionProjectile

(IceBloom, spawns 64 particles, authored using Shuriken particle system. Each adding a buff charge instance, with VFX gradually adding to a shape. The charges clears gradually showing the visual's depletion)

- *Modifier*, is not an Action by itself. Most of the time this is part of a passive. Modifier, depending on where it's placed (as a passive skill, buff, or passive effect from equipment) will be included in the chain of effects when an 'Acs' is casted.

  Developer notes: *A passive with a Damage(heal self) then Modifier, that's called from an OnDamage event, can give a lifesteal or return damage effect. Useful to cap the amount of lifesteal based on the actual damage number delivered from the previous 'Acs'*
  Modifier only contains condition filters, a NumberFormula, which can make use of the

previousTotal and damage.bakedNumber in the calculation

Enchant Blaze adds fire element to hits, also applies burn

- *Aoe*, The size can be determined by duration or health. The strength of the effect to the target can also be affected by the distance to center, duration, or health.

  Developer notes: *An explosion on OnEnd will make a timed bomb. A single hit counter with Rooted debuff in the OnEnter will make an ankle snare trap. Options for Aoe are similar to Projectile. Difference being, it doesn't move, and it has the option to have health and be attackable.*

- *Knockback*, adds a knockback direction + duration to the target.

  Developer notes: *Knockback value for a target is accumulated and averaged. The resulting new value will always have the total distance added, which may change the magnitude or total duration. This directional value overrides the target's movement direction for a duration, with various directional options like use self floor normal, use target floor normal, world up,*

away/towards caster, etc. Best to keep the duration value low, as knockback is a form of debuff that disables the target's movement control

- *Teleport*, The teleport location is searched by a series of raycast, bending as it hits an obstructing objects. So, a teleport is always an instantaneous, multi-linear dash.

  Developer notes: *comes with several rules to find the final destination, such as obstructed by list of layers defined in the Acs.targeting. Also depending on the targeting, the direction option changes. World up, forward/backward, outward/inward, closest on target or behind/in front of target.*

- *Dispel*, removes a Buff instance from the target.

  Developer notes: *Which buff gets removed is determined by several filters, such as comparing the Dispel charge against the Buff's charge, valid 'SE' tags, and/or very specific Buff. The amount dispelled depends on the charge of the 'Acs' casted. If there's no filter or amount of dispelled, then it may dispel all Buff completely*

- *Craft*, Adds to what the target can craft.

  Developer notes: *Must be a passive.*

Examples of Implementable Skills with ACS

DOTA

- Rikimaru's BlinkStrike - Teleports behind the target unit and deals bonus damage
- Radiance - Toggleable equipment. Burns enemies in an area per second and also causes chance to miss
- DarkWillow's Bedlam - Single ActionParticle circling the caster, damaging rapidly to any target touching it, particle accurate hitbox
- ArcWarden's SparkWraith - Ground target Aoe that triggers a homing projectile to the first target that comes close to it
- Zeus LightningBolt - Unit target, or ground target override. Damages the target enemy or split the damage in an area if casted on ground
- ArcLightning - bounces to different targets once
- Medusa's ManaShield - Reduces incoming damage at the cost of mana. Toggleable
- SplitArrow, auto attack projectile hits multiple targets. All triggering OnHit/Damage
- LinkenSphere, LotusOrb, AM's SpellShield - Spell block, spell reflect, spell immunity, spell immunity piercing. All charge based
- EmberSpirit's FireShield - Shield with its own health that absorbs only certain damage type

- Juggernaut's OmniSlash - Pretty much bouncing projectile that can bounce to current target too, applies BlinkStrike effect, with invulnerability and phased within the duration per bounce.
- Huskar's BerserkersBlood - Passive. Gains a charge for every percent missing health. Each charge increases attack speed and health regen * strength
- Orb effects - Desolator, Feast, choose which order whether damage or apply debuff first

GW2

- GW2 long bow - projectile number value based on distance
- GW2 combo - define attack, damage or effect multiplier, and animation for each sequence

RAGNAROK ONLINE

- Ragnarok damage formula - Elemental, lifesteal based on damage dealt,
- Ragnarok Mage's FrostDiver - Single unit target, chance to freeze. Frozen enemies have their armor element changed to water, cant move, and breaks on first damaged
- Ragnarok Wizard's StormGust - Aoe ground target. Causes water element damage per second, and targets staying inside for too long will also be frozen. Already frozen units becomes invalid for target

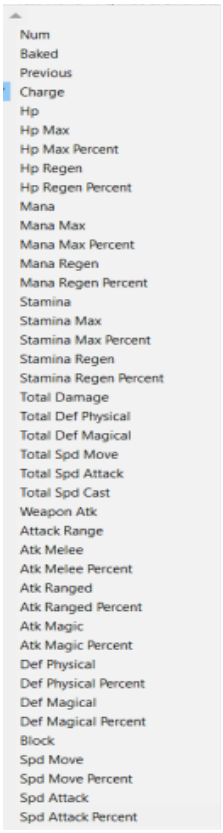
ANIME / OTHERS

- Konosuba power level - Goddess level Aoe ridiculous duration persistence
- WeaponAttack - supports weapon variance for different weapons currently equipped. A single skill hotkey that handles many weapon/directional overrides
- Melee - Directional swings, root motion
- Melee - block is back + click. Animated collider or raycast sweep option
- Melee - Block is reduce incoming physical based on direction. Amount blocked is stat based, and reduces stamina for blocked damage
- Melee - Stance for different weapon or "special moves" skills. Simulates those sword fight samurai games like GhostOfTsushima
  Guns - Reload restores x amount of charge, which enables x times successive cast.
- Shield - its own value for remaining shield points. Can define what damage type gets blocked
- Shield - Blocked projectile's shield damage considers the charge of the projectile (in case it's a shrapnel type projectile)
- Buff - Supports charge, duration, immunity piercing
- Buff - Supports energy based charge. ChargeIsEnergy such as based on remaining hp/mana/stamina
- Buff - Supports "damage instances" as charge before the buff ends
- Buff - StatusEffect (SE) is just tag. A buff can have multiple tags. Such as Burn, Hot, Wet, Cold, Airy, Dusty, Poison, Curse, Frozen, etc.
- Buff - ChargeIsDuration means jumping into a lake will cap the Wet SE to say 30. Leaving the lake will start drying up the Wet SE. Could be in a non linear reduction.

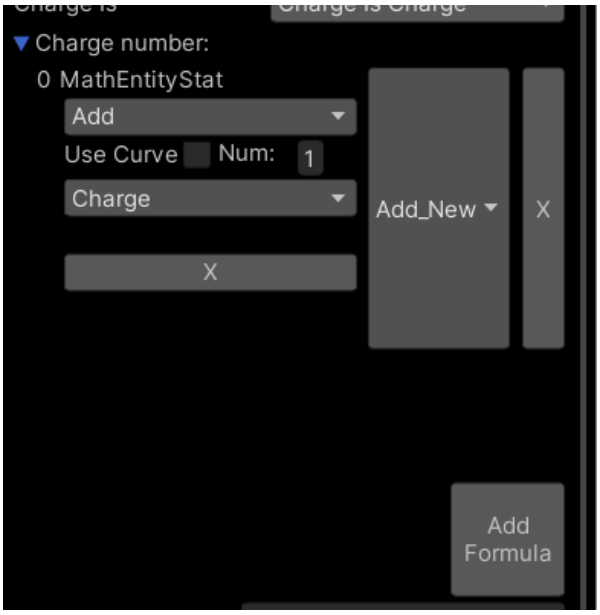
- Buff - Optional to spawn a Thinker, that handles targets entering the radius, interval applying times, event based (OnStart, OnEnd) applying. Can apply any other Acs on any of the event, on any of the Acs's targeting
  Dispel - Also supports tags. Interacts with buffs with whatever tag is relevant to it. Can also have override for different interaction with different tags
- Buff graphics - SE also have screen effects, also intensity based on charge. Common games have the "wet screen" depending on the Wet SE on the character
  Buff/Shield graphics - Visuals can follow the remaining charge or duration of the buff
- Charge based buffs, status effect is just tag, dispel, charge/duration

Terminologies:

SO : ScriptableObject. Unity's way to store data-oriented assets
Acs : ActiveSkill. The parent skill Scriptable object holds together the composition of the skills, which may include multiple sub-SO
AcDef : ActionDefinition, the abstract sub-SO that defines an action, triggered on events by Acs. Concrete examples: ActionAoe, ActionBuff, ActionMelee, ActionProjectile, ActionCraft, etc
Buff : Buff or Debuff is considered a Buff. Shield is pretty much a Buff with its own list
SE : StatusEffect. A tag that can help define a Buff. A Buff may or may not be tagged with any SE. The effect of an SE Buff are still directly defined in the Buff data, but some exceptions have special handling: Flinch, Silence, Rooted, Invisible, Detected, etc
NF : NumberFormula, a class to define how much a number is



Features Highlight:



(Screenshot on the right) Math entity options

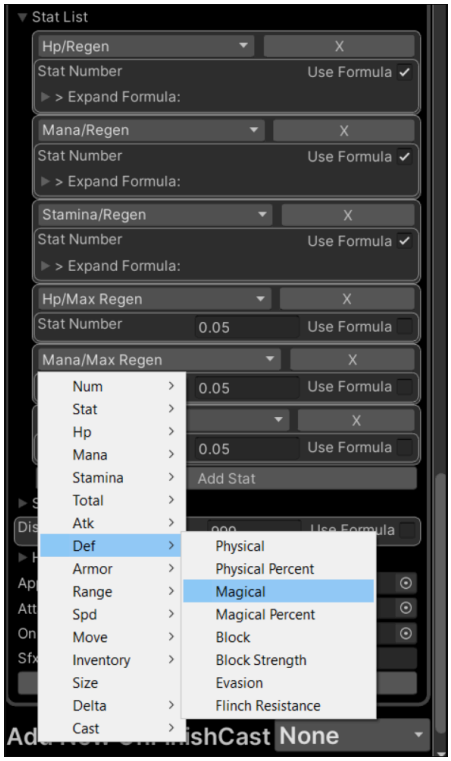
Math Entity Stat

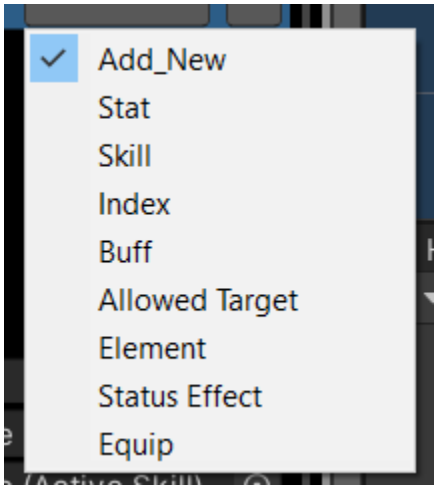
A math operator in the inspector to define any kind of formula, supports brackets/ordered math operations/pull result from previous part of calculation

These are used either to generate a number used for damage/projectile speed/aoe size/any part of skill's parameters, or used in special "TargetCondition" part in Acs or individual action parts, to further filter a valid action to apply.
TargetCondition returns true if result is >=1, otherwise false
For a "Except for" condition, do a (1 - theCondition)

A formula consists of 1+ NF, and an NF contains 1+ MathEntityAbc. This structure enables the ordered math operation
Most MathEntity can return a number or a curve based on the pulled result. Say Equip knife num 5 means returns 5 if equipping knife, else return 0.
Or Stat str num 2 means return total str x 2
Or Stat charge curve (time 1-10, value 0-1) returns 1 when at 10 charge, or 0 at 1 charge
Example of a full formula:
NF1 (Stat str num 1, + Equip knife num 5), NF2 (Stat skill FireMastery num 5, x Index 0 num 2)
Returns FireMastery level x5 x (Str + 5)
(The last NF part is always the calculated part)
Generally you only need multiple NF for a very complex formula

The most common formula to use. The entity stat contributing to the final number Special types: Num is flat number Baked is the resulting number from a precalculated value from this Acs instance Previous is accumulating number from previous action process in this Acs instance Charge can be used in different ways. The charge of the buff, or the casting charge when casting skills for skills with more than 1 casting charge. Useful for combo effect SourceDamage is specifically used for AcModifier during OnAttacked/OnHitted/OnDamaged/etc events, to modify the damage and returning the value SkillLevel is the level of the learned skill when casting





Type of Number Formula

Stat: the most common formula. The entity stat contributing to the final number
Skill: checks if the entity has a certain skill learned. Useful for skill mastery
type Index: a special type that only pulls number from other part of the formula. This type is necessary to solve more complicated math order operation Buff: checks if the entity has a certain buff AllowedTarget: checks if the target is a certain object type Element: checks if entity is of a certain element. Can check for only base element or final element after armor element enchant StatusEffect: checks if entity has a certain status effect. OnlyHas bool means return 1 if it has. Otherwise it returns the total number of charge of the SE Equip: checks if entity is equipping a certain equipment(edited)

Define Data Aspect and Visual aspect of ACS

Status Effect

This deserves it’s own section
Status Effect (SE) are tag based, means, any buff can have none/one/multiple tags

Tags determine dispel interaction. Like "green herb" cures poison. Then any skill that has poison tag will be affected
Buffs are also charge based, so the tags of the poison of a certain debuff can have say 5 charges, and green herb cures below 3 charge therefore doesn't cure this poison
A few also got hardcoded effects described below
They're also linked to screen visual status effect
Like the common "Wet" also turns the camera wet "trickling down water drops" on the camera screen

```
None
Npc // talking to npc, or npc handling players
Busy
SpellBlock // Linkens
SpellReflect // Lotus Orb. Combine with SpellBlock means AM spell shield
SpellImmune = 1 << 4, // cant be casted on or receive debuff
Hot //free
Cold //free
Burn // free
Wet // free
Airy // free
Dusted // free
Freeze // cant act, -1 charge OnHit
Petrify // cant act, -1 charge OnHit
Sleep // cant act, dispel completely OnHit
Stun // cant act
Grounded // cant jump
Rooted // cant move
Poison // free
Blind //free
Curse // negates positive buff
Silence // cant magic
Disarm // cant weapon/melee
Clumsy // cant use item
Fade // used by intermediary time towards invisible
Invisible
Detected // unit specific. Like revealed by dust
Truesight // observer specific
//Underwater // same as wet ? maybe for breathing
Phased // no move collision (not implemented)
BreakSpirit // disables passives stats and passive buff events
BreakArmor // disables equipment stats and eq buff events
```

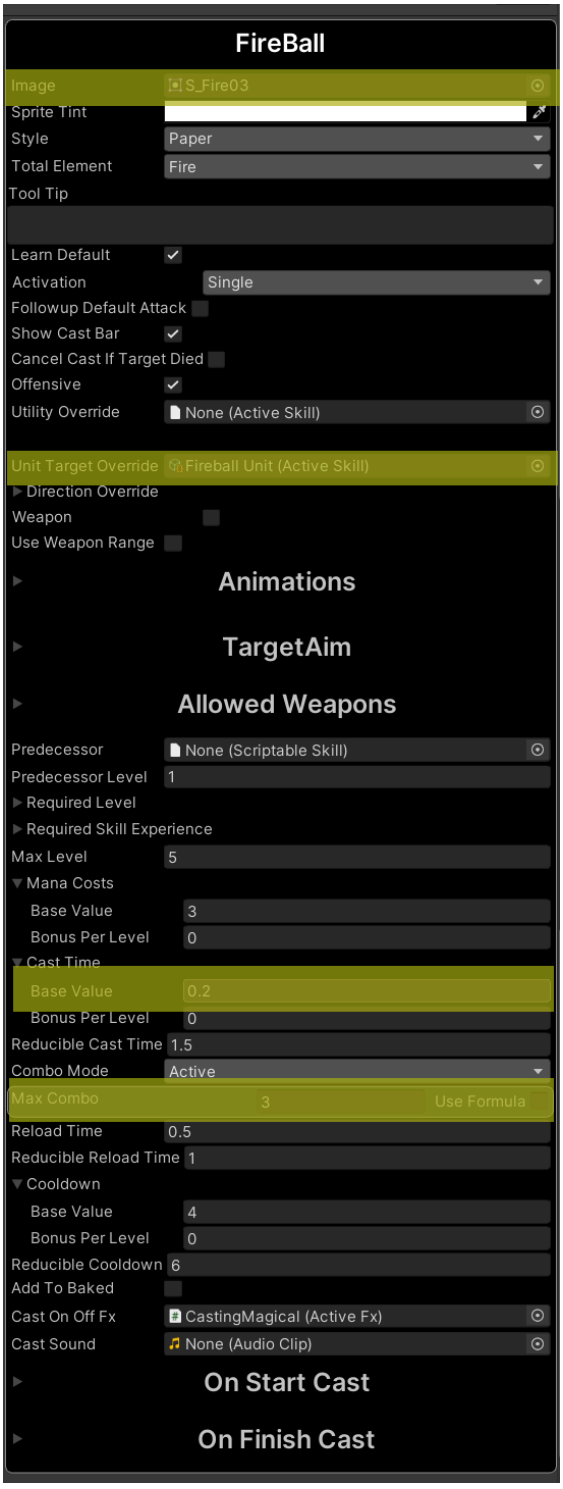
How To

Create a Projectile Skill.

E.G. FIREBALL

First things first - Intro

In this case we will be using Fireball as an example. To create a Projectile skill We would need you to become accustomed to ACS' Unity Inspector within the Skill Editor. There's 5 sets of tabs you will need to grasp before making sense to any skill within your imagination.



- **General**
- **Animation**
- **Targeting** Tab
- **Allowed Weapons** Tab
- **On Start/Finish Cast** .
- **Actions** Tab

Next up-

To create fireball, we'll want to make sure we are using a "Fire" icon/image as that will be the skill's hotkey icon/image within your game, you may upload any images / assets created by you or your design team and update this when necessary.

Designer notes: Recommended resolution 350x350 for retina, 175x175 for Standard or Mobile.

Understanding Style: Paper, Scissors and Rock.

+ Elements: Light, Dark, Earth, Fire, Water, Air, Neutral.

Since the targeting is "Aim", it doesn't need a unit target. But "UnitTargetOverride" can cast a

ACS
ACTIVE COMPOSABLE SKILL

unit targeting version of the skill if this is set
In this case, we want a projectile skill that casts a Fireball Unit in a direction directed by the player.
We can specify how quickly the player can cast Fireball, ***(General Tab)*** what's the max amount of times a player can spam it before having to cooldown ***(General Tab)***, and also specify the range of the fireball itself. ***(Target Aim Tab)***

In this case we have already set the **Range at 5**,
Base Value (Cast Time) at 0.2 with a **Max Combo at 3.**

This simply means it'll take 0.2 second to cast that we can spam 3 times. Before the cooldown triggers.
Understanding On finish cast and On start cast tabs:
On Finish Cast literally means what will occur after the player proceeds with the cast time of the skill.

Whereas, On Start Cast is what occurs when the cast just started. Or when you've hotkeyed a spell, and left clicked.

Understanding the Range and Damage:
These things can be given a flat amount or can be given a formula to create an ability / skill that can scale into higher level gameplay.

Last steps for Creating a Projectile Skill-
FIREBALL

We'll be apprehending all values and formulas of the fireball here,
On Finish Cast, We'll be sending an ActionProjectile *(reference can be seen on page 7/25)*
With a speed of 15, and a duration of 2, that can collide with Target types: Monster, Player and Enemy. The size of the Fireball itself is, 0.2 and it will create 1 hit.

In this case, We have already created an FX for Fireball that you can individually see within our library of FX.

The most important piece of information here however is,
On Destroy: Fireball Hit (Active Skill)

What this actually means, is that the fireball we were creating was only the projectile, the speed, the range, cast time, the cooldown and how it collides with the world, the explosion however will be calculated **ONLY when** the fireball is destroyed.

The Active skill, Fireball Hit will be representing how much damage you'll do and how the skill will inflict on other players. Tadaa! You've made your first Projectile Skill.

Let's dive into Active Skills, and have a more intense read on it on Page 6,
Structure of ACS

So just write the stuff in general tab without pics, and the following pics shows the step by step of setting the formula

How To

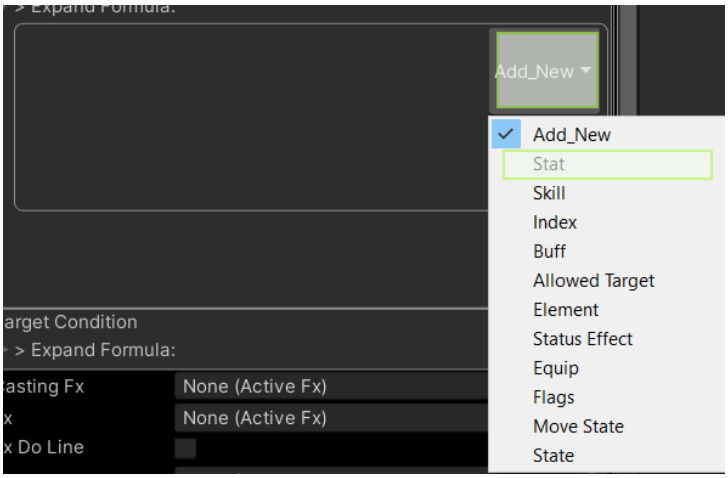
Duplicate a Skill and make it your own.

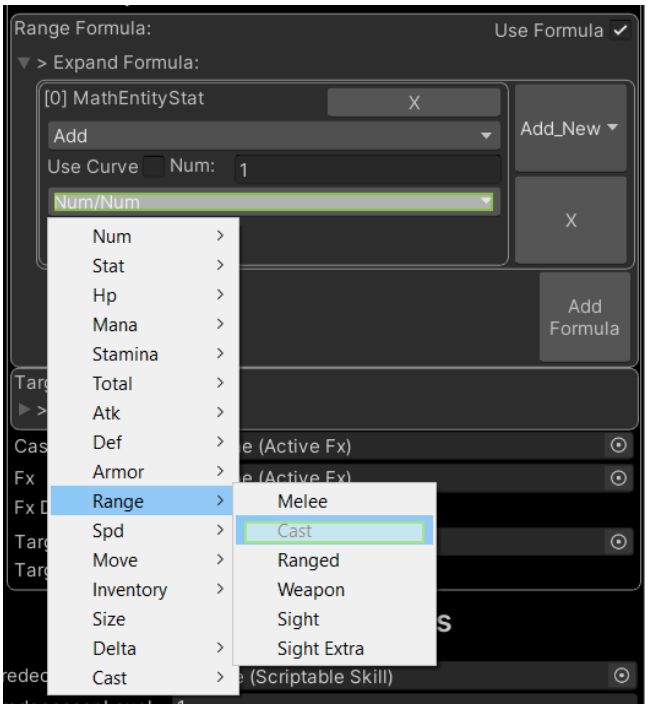
E.G. IceArmor -> ROT

Intro

In our example, we will be using IceArmor, a skill within the **ACS Library**. Lets understand the key values and go over what makes an IceArmor first.

This is the values inside the **TargetUnit** section. Its range is set to CastRange, which is typical for a magic skill. *Seen from the picture on the bottom right*



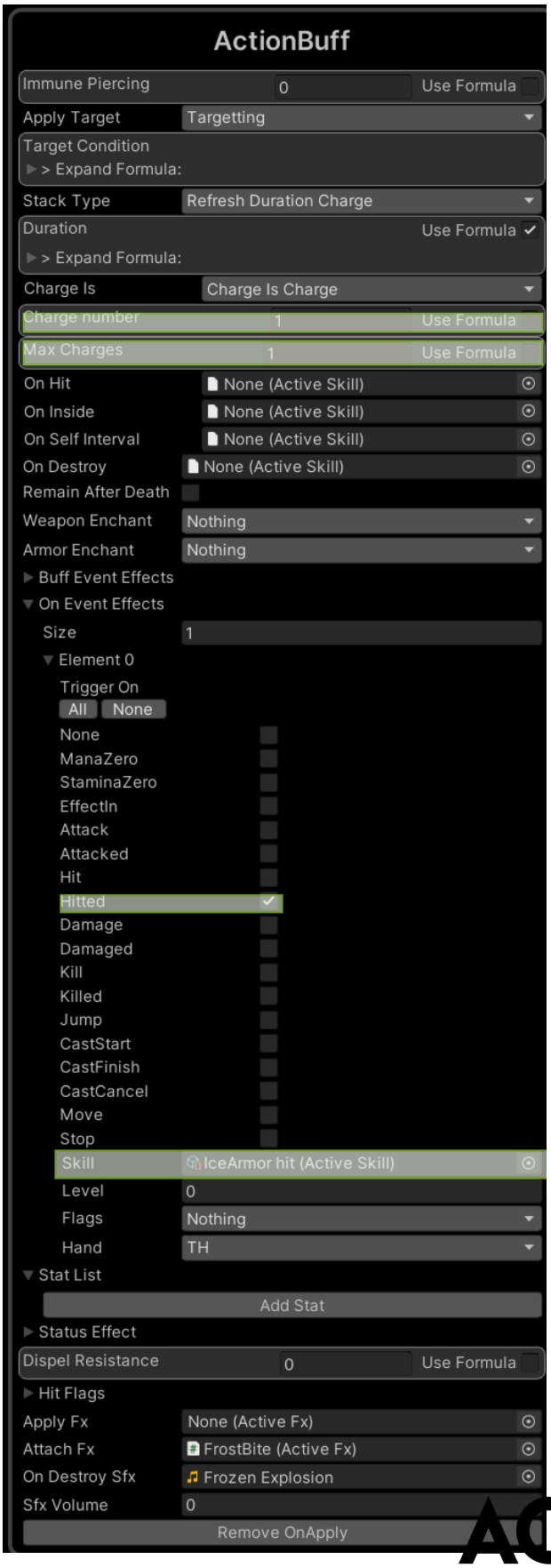


ACS
ACTIVE COMPOSABLE SKILL

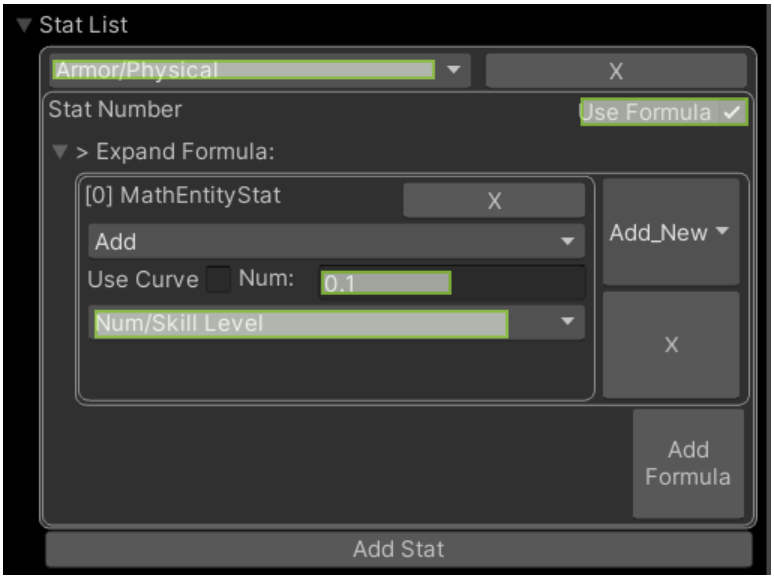
If you see inside OnFinishCast, the ActionBuff is set as so,

Charge number set to 1,
Max Charges set to 1,
Trigger On set to Hitted
and Skill set to “IceArmor Hit”

IceArmor Hit is an active skill that will tick a buff modifier. The buff modifier will decrease Attack Speed by 30% for anyone who gets in contact with Ice Armor.

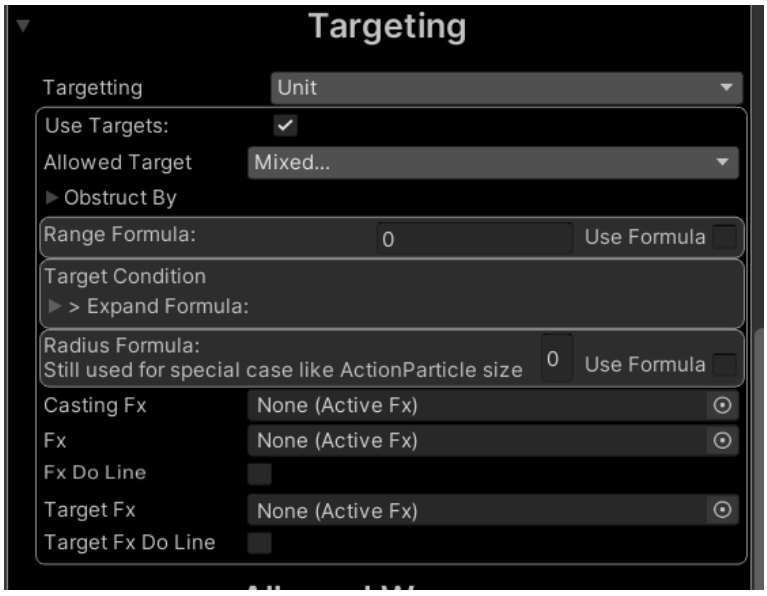


And inside the StatList of IceArmor, is the positive buff for the unit itself



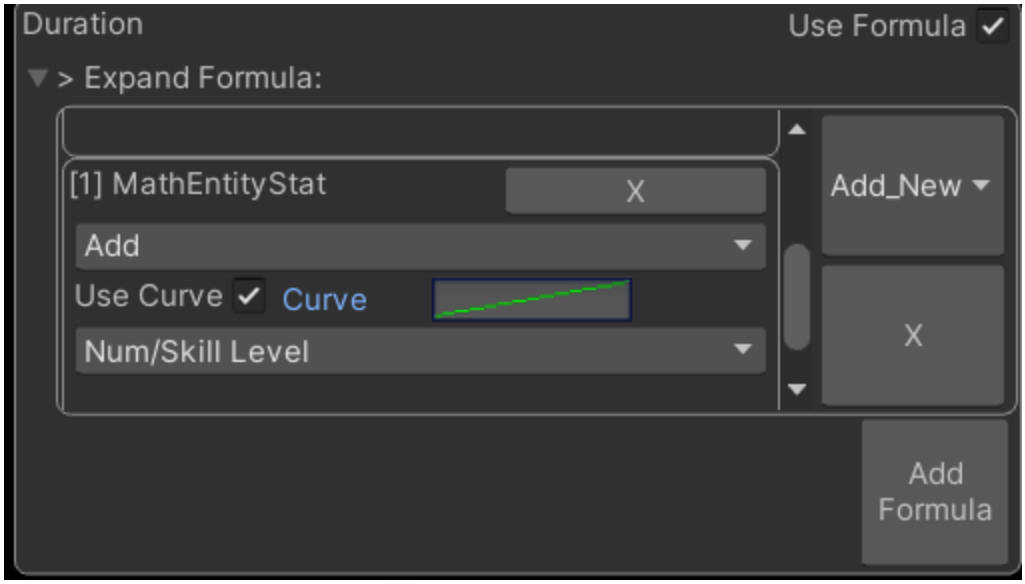
The key point is in "OnEventEffects", which is set to trigger when Hitted. The skill to apply in this event is called "IceArmor hit"

Typically, a chained effect like "IceArmor hit" is named the same as the main skill, contains a space in the name, and then an identifying word for it (in many case it's simply "hit")
A few values inside this "chained" skill is not really used, such as the combo mode/max combo, cooldown, reload, etc, as they're technically not manually casted but handled by the main skill
If it contains a buff, then the icon is still used to identify it. Anything in OnFinishCast is also what will be applied
Targetting will also be used, in this case it's a TargetUnit.



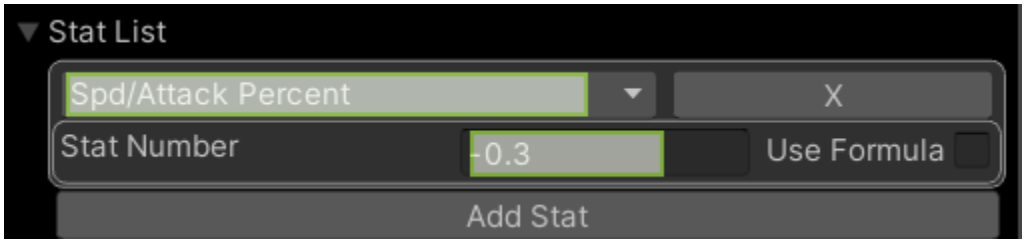
The key with OnEventEffect triggered in Attacked/Hitted/Damaged is, if the chain skill is TargetUnit, then it'll be applied to the attacker. If it's Self, for the hitted victim, which can have an area radius. And if you want an area radius around the attacker, then use Point

Notice range and radius is set to 0 as it's irrelevant for a chained skill



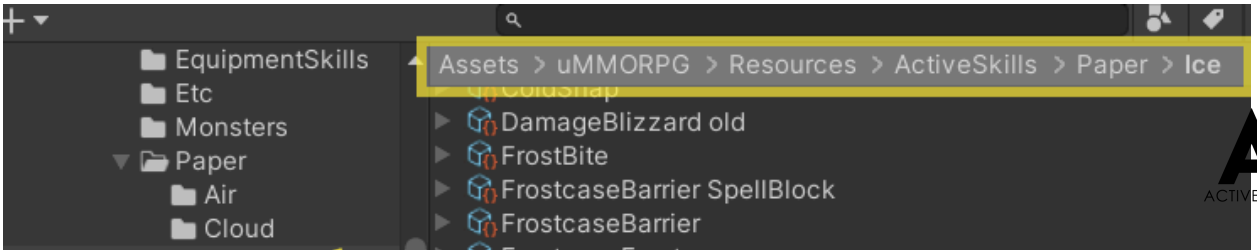
Duration uses the Skill Level, and set to a linear curve ranging from lv 1 = 0.7s, to lv 5 = 1.5s

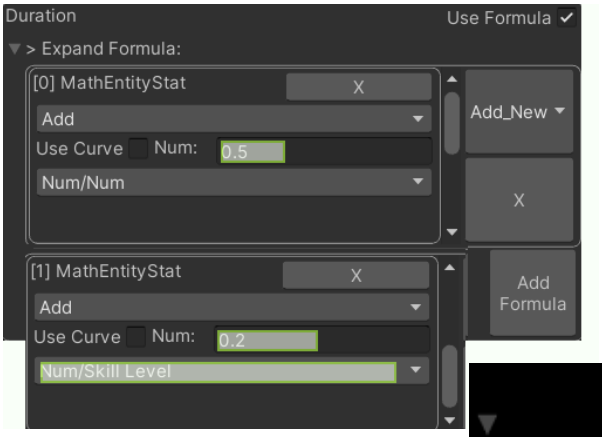
With that being done, let's get our focus into IceArmor hit. You'll find it within the **ACS Library** under Assets > uMMORPG > Resources > ActiveSkill > Paper > Ice. Open the inspector, and inside the ActionBuff, you will see the StatList of **IceArmor hit**, you'll find what happens when an opponent gets in contact with IceArmor, in this case reducing their attack speed percent by 30%



When you've understood the key values of IceArmor, Then you are ready to simply duplicate and adjust its settings to another skill. In this example we will be changing IceArmor into Rot, a skill seen in Dota 1. Where the caster gets to toggle an area damage skill that will slow the movement and deal damage to anyone in contact within the skills radius, in exchange for health.

First things first - Duplicate the IceArmor skill, and move the duplicate into the "Dark" folder and rename it to Rot

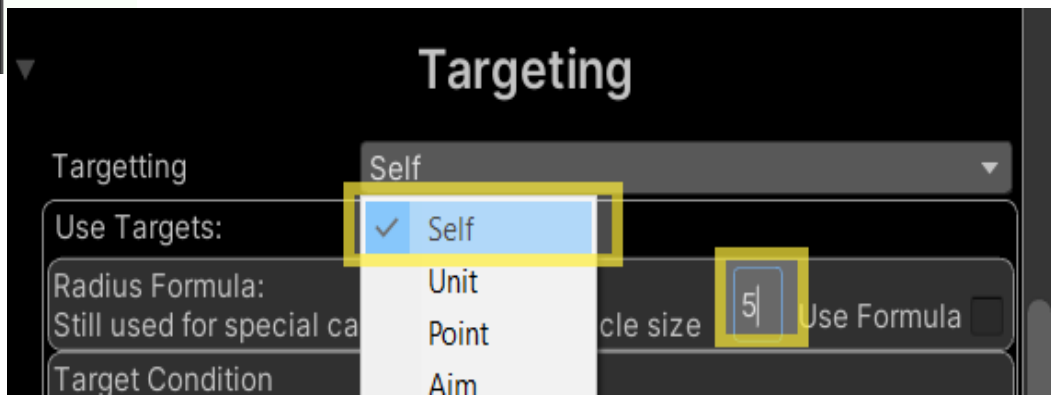




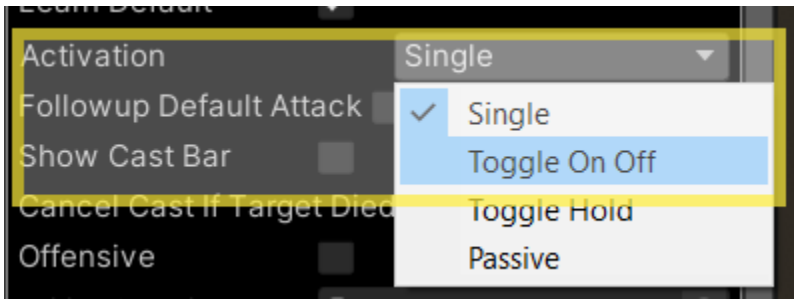
You may even assign an appropriate image if you'd like, this can be done within the Inspector.

Inside the ACS Editor -
Inspector Settings of **our duplicate**

We will go inside **our duplicate** (From here onwards we'll be referring to it as '**Rot**') So within the Rot's inspector settings, we will be changing the targeting to Target Self. And changing the Radius to 5.



Change the activation from Single, to Toggle On Off.



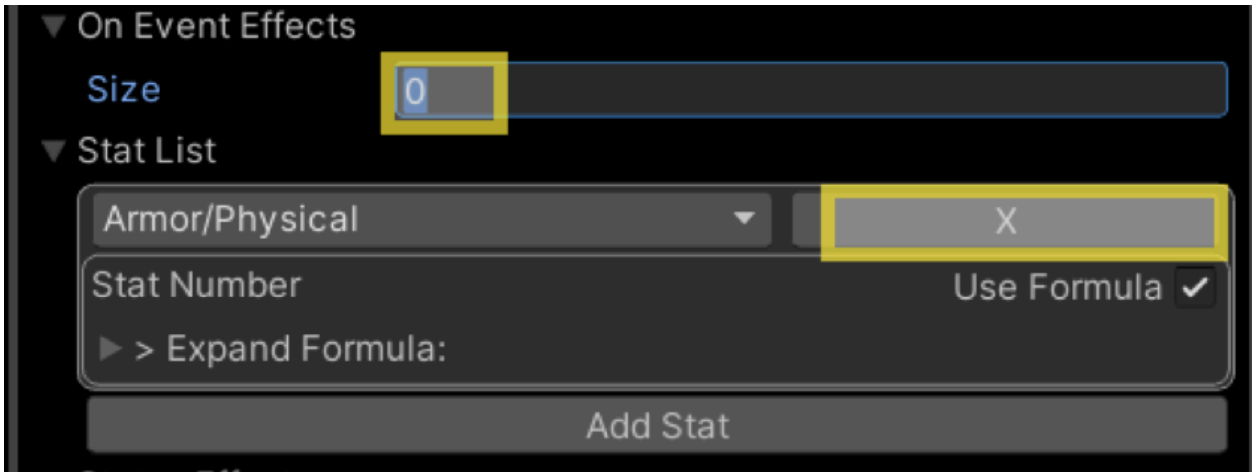
Not super necessary but one of the use is so AIs would use it offensively

Within the Targetting within the ACS' Inspector we'll need to make sure that we have go the "Use Formula" Checklist -> Checked
We go ahead and '**Add formula**'

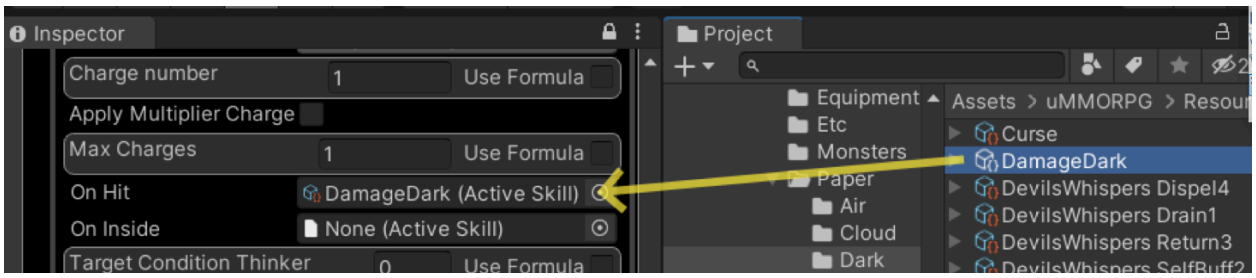
Keeping these values low, because you'd want the de/activation to be spammable
Developer Notes: Just remember to never have castTime.baseValue at 0!

Because of what we understood from dissecting the IceArmor skill, what we gather is that the duration had some components to it. For Rot, we will just uncheck UseFormula and set a flat amount of 0,

Developer Notes: because SkillActivation.ToggleOnOff commonly lasts forever until untoggled. Having a duration for toggles will have a different effect



Set onEventEffects size to 0, and remove that armor bonus with the X. As we wont be needing them here

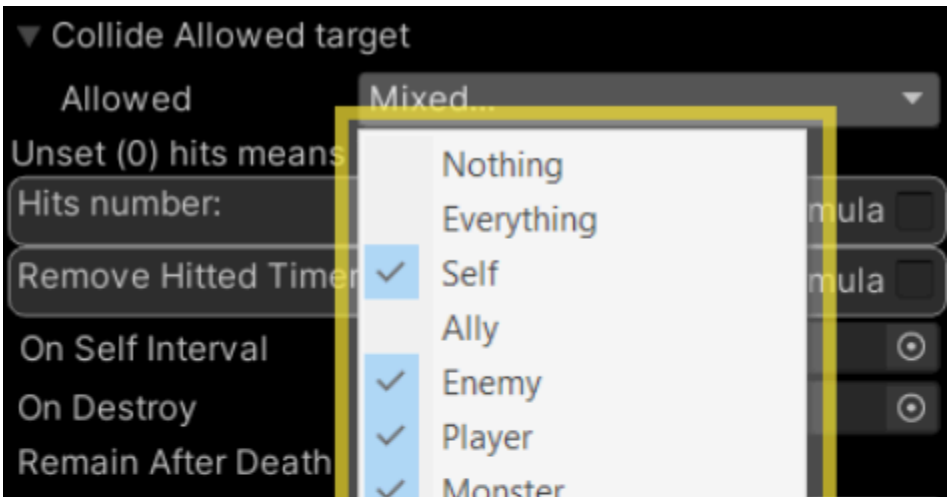


We recommend using DamageDark where it can be found in the subsection named as “Dark” within the Paper folder `Assets > uMMORPG > Resources > ActiveSkill > Paper > Dark`
We’ll need to drag this onto the OnHit field, and start readjusting its behaviour.

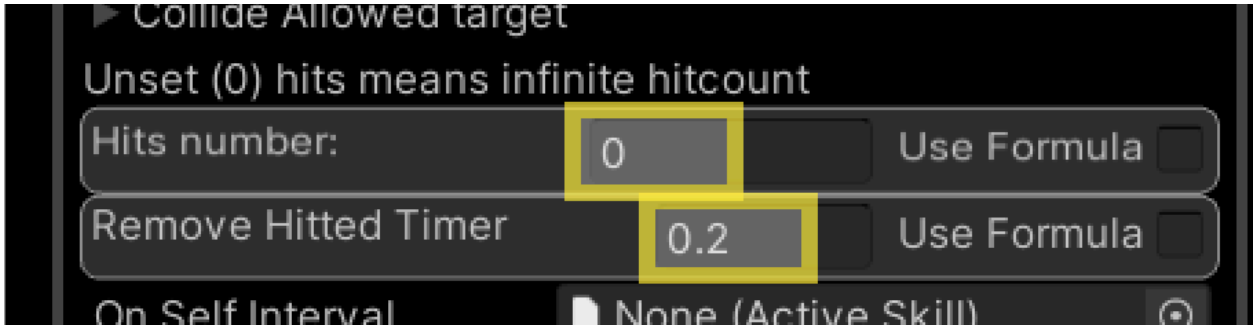
So we’ll first change the affecting radius, and set the Collide Radius to 3.

Next on the list, we’ll be changing the hit check. Making sure that Enemy, Player, Monster, Pet, Self and Alive is checked.

Notice that “Self” is included, this is because Rot will not only inflict damage on opponents but it will also be a skill that inflicts damage to the caster; Self.



ACS
ACTIVE COMPOSABLE SKILL



The **Hits number** and **Remove Hitted Timer** field will be shown only when there's something in the OnHit/OnInside field.

DEVELOPER NOTES: If you remember 3 steps back, we've placed DamageDark into that field

Within these field you'll need to set fields as shown above or written below:

HitsNumber, 0; Which means the buff won't end from hit count
RemoveHittedTimer, 0.2. Which means how long until the target is valid to be applied again

It's currently a toggle skill that deals damage to an area every 0.2 seconds, you will be able to check if this works by Playing the Unity Scene, and learning the Rot Skill that you've created.

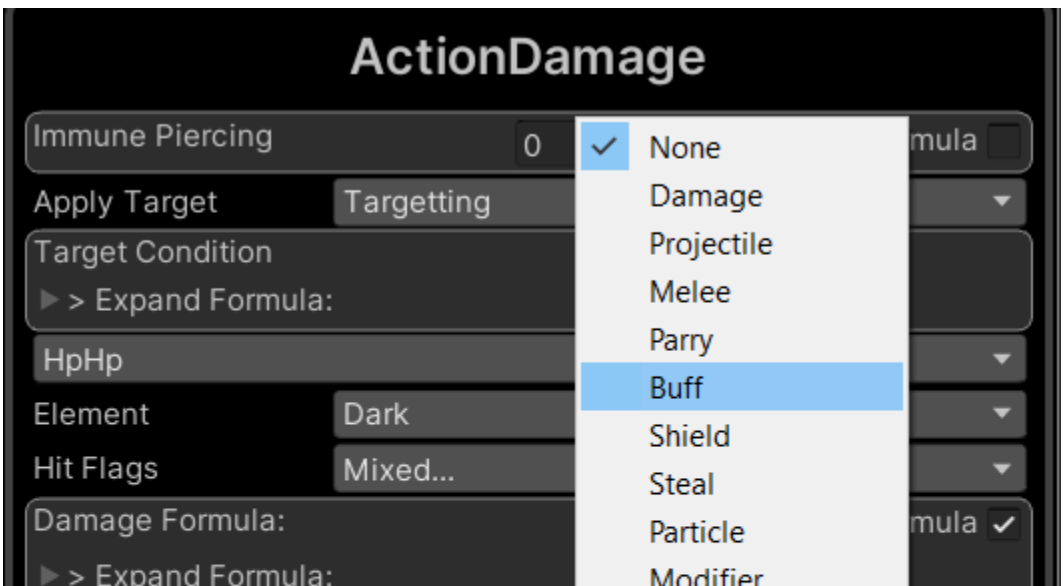
Give your self a pat on the back, You've made a "simplified" Rot!

Next up, Completing the Rot Skill -

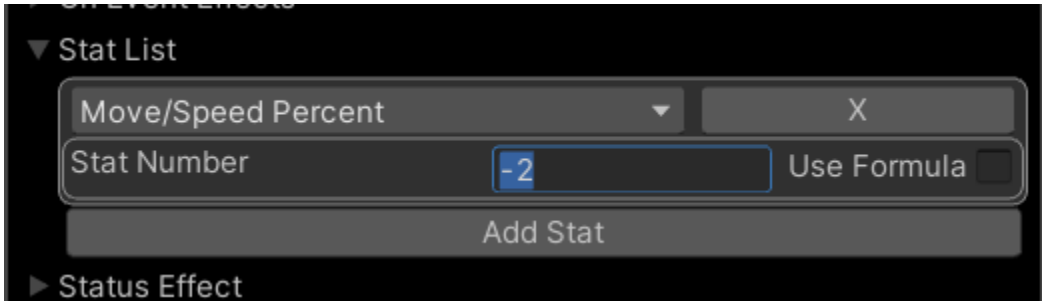
Continue on to complete the original Rot, which includes the self damage, and the extra slow debuff to nearby enemies

You may now duplicate DamageDark and rename it into "Rot hit", You will then need to come back to the OnHit field within the Rot's Inspector settings and replace the OnHit with the new Active Skill, Rot hit.

Inspecting the Rot hit, By default, our Rot hit [Action Skill] would already have an ActionDamage but on top of this you'll also need to add a ActionBuff. You can do so by clicking a dropdown that presents the list as seen from the image on our left.

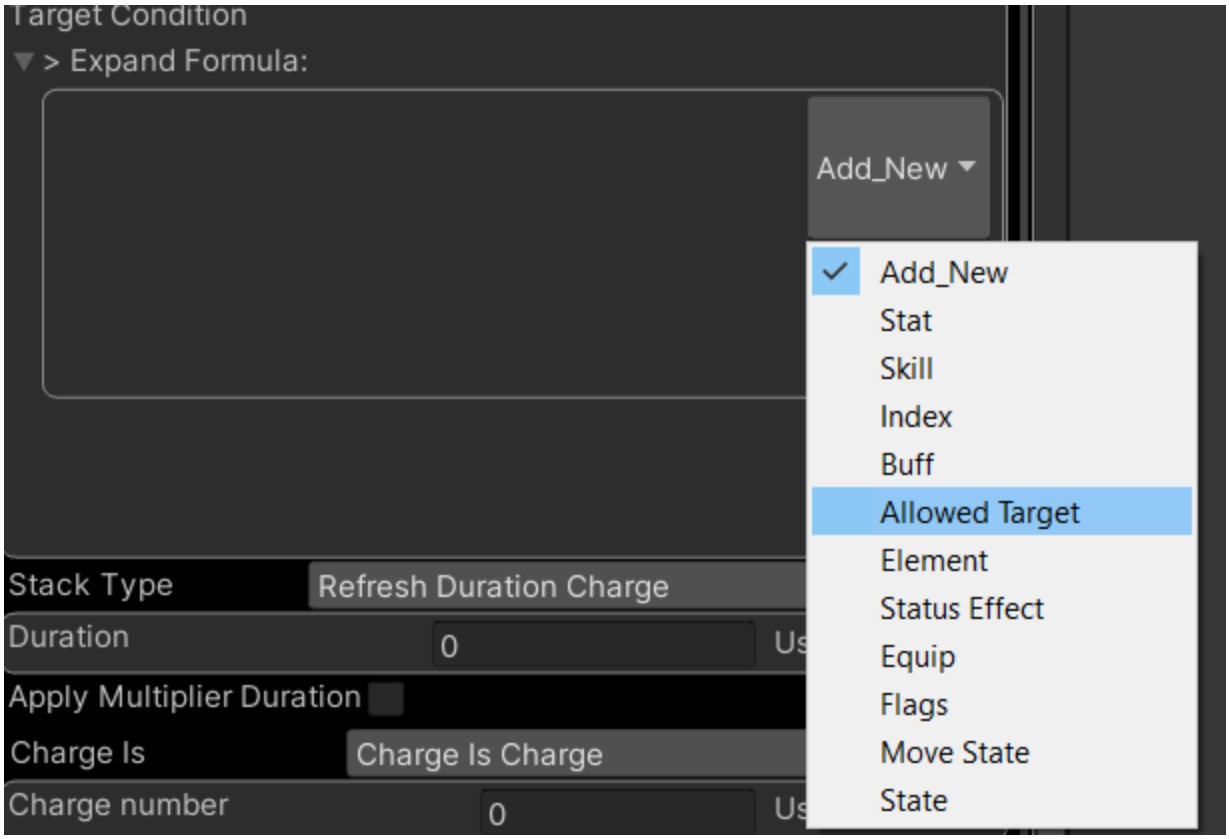


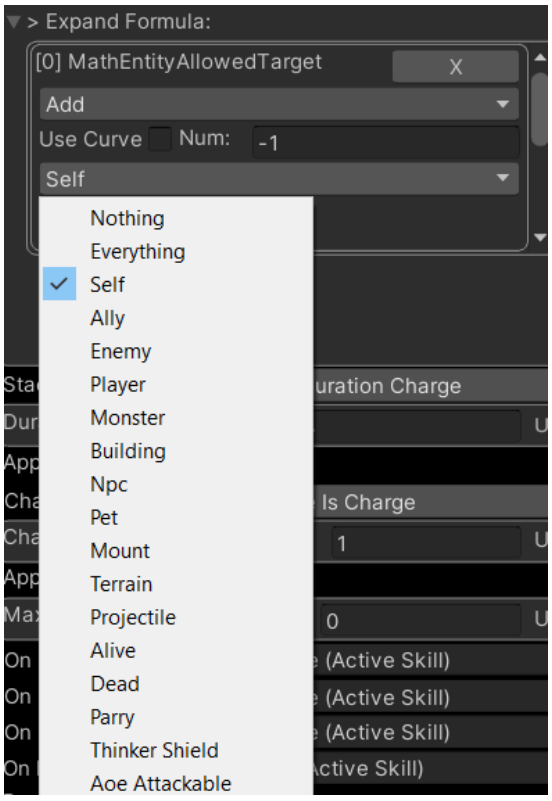
From a fresh new ActionBuff, the only thing to add is the movement speed debuff. Just for testing purposed we'll put -2, which is -200% movement speed reduction.



But we don't want this to be applied to the caster, right? So, for a "per-action" condition, there is the TargetCondition formula that we can use to exclude self from getting this part of the skill

TargetCondition is a special kind of formula. The final result has to be 1 or more to be valid, else it's invalid and won't be applied. We'll want the AllowedTarget formula as seen below.

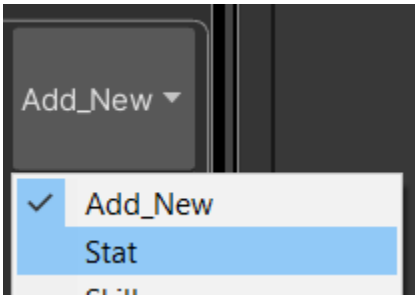




Select “Self”, and set the Num to -1.

Developer Notes: What this means is that if it’s self, do a -1 to the equation.
But after this you’ll need a +1 by default, so if it’s Self it’ll equate to 0, else it’ll be 1

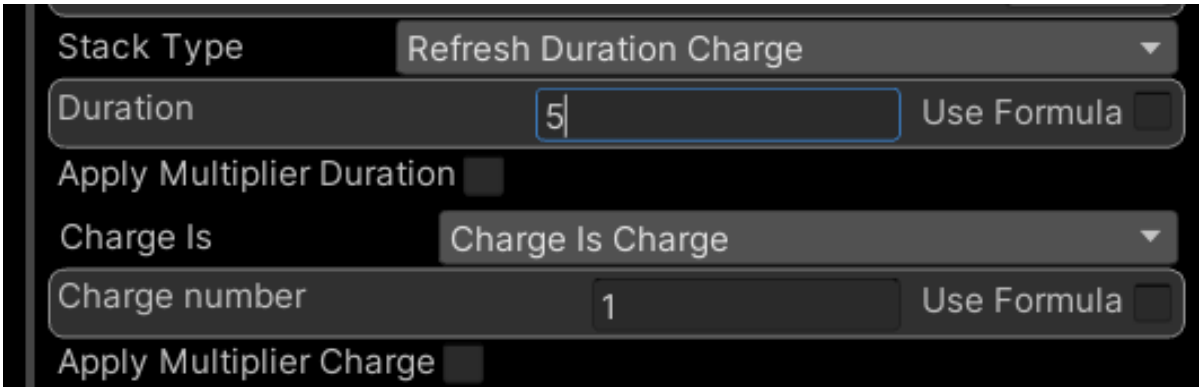
To do that, we’ll need to add another Stat formula



And since it already defaults to Num/num, with Num set to 1, this is done!

And finally, Set the Charge number to 1. And set the duration to 5

Developer Notes: I've set the duration as 5s just to test, so this number might need to be changed again if it doesn't mimic the exact Dota 1 Rot to its fine details.



To test out the skill that you've created, play the unity scene that is provided and learn the skill that you've created, in this case Rot. Type in, "/learn Rot" in the chat field and try toggling the skill on and off.

Developer Notes: Make sure to leave a space after "learn" and follow the naming convention that you've created.

And there there you have it, 100% just like Dota Pudge!

[Image of Rot using Duplicate of IceArmor and DamageDark]

How To

Particle skills
-Under construction-
These are defined in 2 places. The Acs data itself, and the simulation part in the unity's particle system prefab

Masteries:
Weapon mastery
-Under construction-
Passive skill with AcModifier, bonus damage uses MathEntityEquipment

Skill mastery/talents
-Under construction-
Passive skill that can be empty, and affected skills have MathEntitySkill in certain parts of the number, checking the existence of this passive (with levels)

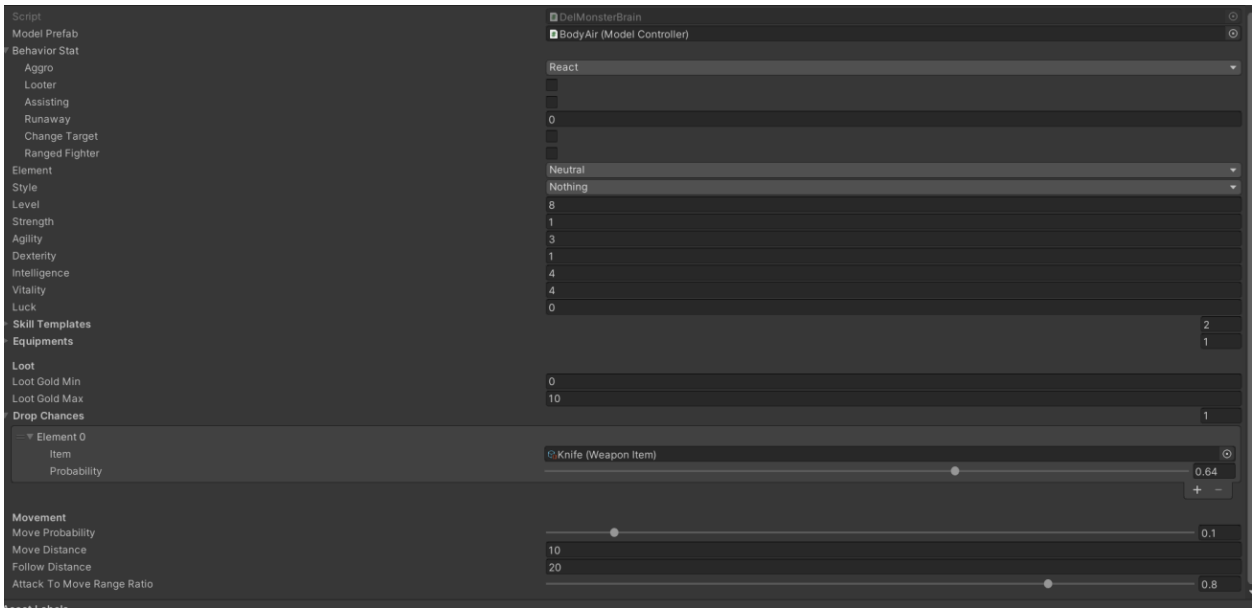
Specific SE cure potions
ItemSkill with ActionDispel, with filter of the SE tags that it can cure

Others:
Dota Antimage's ManaBreak into Naix's old Feast
RO's IcePick bonus damage per target's defense

ActiveFx (Afx)
The container script that holds multiple FxBase: Particle, Line, Object, Projector

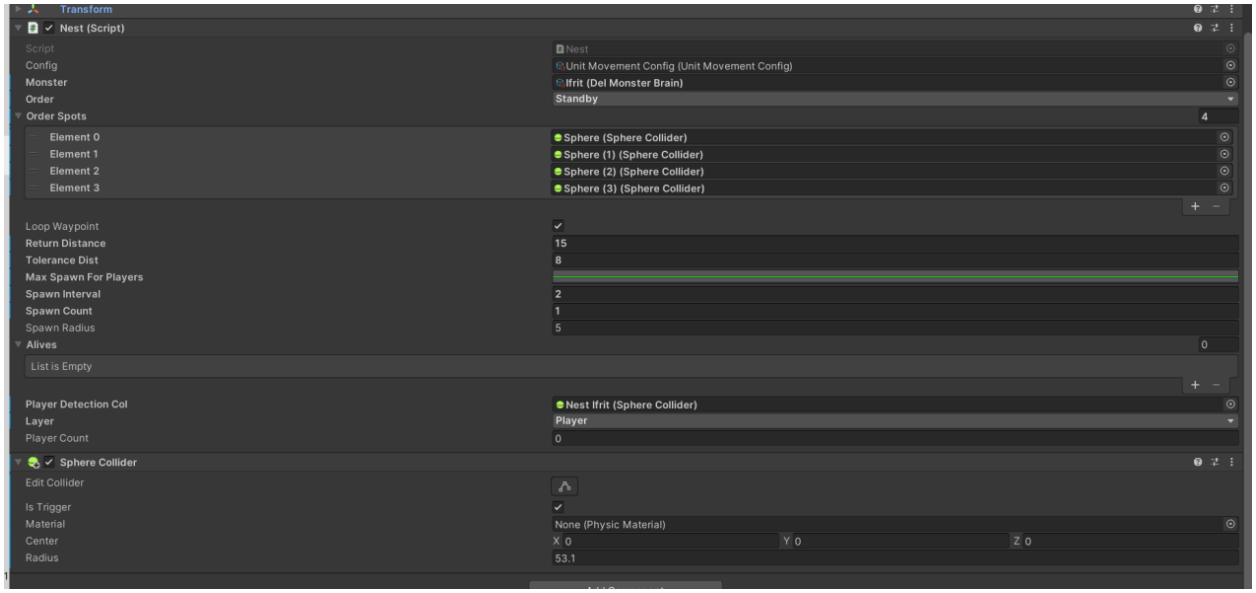
FxParticle
Supports "forever" on off particles, dynamic "spawn on certain body parts/weapons", parametrized updates to accurately indicate the buff/aoe it represents, correctly maintains particle position per size of the source

Monsters Behavior AI



AI, Utility (giving orders), looter, aggressive, assisting
A way to fine tune spawned mon's behavior, giving order to the Utility AI. Can set the mons to stay in a spot facing a certain direction, patrol or follow waypoint (for Moba)

Nest System



Nest monster spawner despawner/minion controller
Able to persist monster's health and state (less than max hp, have buff/debuff). Correct filtering to still allow buffs related to regen and aura, but despawns correctly unless the buff/debuff matters when players return to the half-fought monster.

Also an optimization to only spawn monsters when there’s players around, despawn when no players.

Monster: The monster data that will be spawned
Order: None, Standby, Follow, Patrol. None means wander, or not if the monster's "moveProbability" is 0
Order spots: Standby - Col 1 position is the standby position. Col 2 is facing direction. Return position is Col 1 Follow - Col 1 is target of the follow (for now can't be set at runtime, only at design time). If col is another nest, then the spawned monster will follow what that nest spawns (almost implemented). Return pos is follow target Patrol - Cols are waypoints for patrol. Return position is always set to current patrol target
Loop waypoint: If order is patrol, then goes in circles. Untick for Moba path to enemy base thing ReturnDistance: Where the spawned will prioritize returning to start position
ToleranceDistance: Only for order Follow. The maximum distance the spawned will be closest to target. Must be lower than ReturnDistance. ReturnDistance - ToleranceDistance is the "range" which the spawned will maintain its distance to the followed MaxSpawnForPlayers: A curve. Value is number of monster, Time is number of players
SpawnInterval: time between each spawn. If a spawned dies when at full spawned number, then next spawn will be this interval after that spawn's death
SpawnCount: When spawn interval reached, how many monsters to spawn in one go. Maybe this can be a curve too..
SpawnRadius: The random area of the spawn location Alives: The current living spawned. (Not to be edited in design. Runtime use only). If a dead spawned get ressurected when the nest is at full capacity, then alive count can overflow
MaxSpawn Layer: Triggered by player entering. I guess could also be set to trigger when npc or pet (or another monster?) enters
PlayerCount: Runtime only
SphereCollider.Radius: The player detection area. Higher than SpawnRadius means the player will have more reaction time. Lower than SpawnRadius will feel like entering a trap building. Both low radius means totally ambush scenario

Road Map: Tech

Redefine ActionMelee - ActionMelee can prooobably be a SkillTarget. As a target acquisition method

It’s an Action now bcoz it needs a Thinker to maintain the targeting and directional handling

Server-wide Quests - Cooperation based server progression by player contribution. Players can contribute to the success (or prevention) of the quest. Rewards goes along the line of, making dungeons and npc available, server wide story unlocked, etc
This can have some sort of visual presentation for stakeholders to see, roadmaps are planning when time and what milestone will be achieved.

Road Map: Strategic Product Road Map

-

Customizations:

- Design A Skill (Showcase ACDef and SE's)
- Allocating Skills to Units (Monsters, Terrains, NPC, Players)
- Fidget with Numbers (Primary / Secondary Attributes)

Other Features:

Definitely for this we need to shine light on all the features and have a spotlight for each

DelHybridPlayerMovement - WASD and Click to move in one
Supports "walkToCast" out of range targets
Shooter mode switch to enable camera rotation without right click drag

Cinemachine integration, region based camera override, for directed CCTV effect on certain areas

SkillItem, EquipmentItem, ammo effect accumulated for bows/guns

AI, Utility (giving orders), looter, aggressive, assisting

Nest monster spawner despawner/minion controller
A way to fine tune spawned mon's behavior, giving order to the Utility AI. Can set the mons to stay in a spot facing a certain direction, patrol or follow waypoint (for Moba)

Also an optimization to only spawn monsters when there's players around, despawn when no players.
Able to persist monster's health and state (less than max hp, have buff/debuff). Correct filtering to still allow buffs related to regen and aura, but despawns correctly unless the buff/debuff matters when players return to the half-fought monster
Example scenario when a player casts a poison skill to the monster, but died and returns to town. Within the duration, the monster may die from the poison. If the monster wont die from this poison, it will not be despawned until the poison is over and the monster regens back to full hp, on which it passes as having an initial state, gets despawned so it's excluded from interest management and saves server CPU overall

SpatialHashingInterestManagement3D
Just a 3d grid version of the default 2D spatial hashing from Mirror

Vending
Player can set shop in the field ala Ragnarok/Lineage. Supports buy, sell, and crafting service

Elemental table for damage calc

Mouse cursor depending on skill casting state, or what's under the cursor (UI or whatnot)

Slot for Vfx for skills, projectiles. Supports start/on/end effect, intensity/emission/size based on the skill/buff

Screen effects from de/buffs

Resources:
https://developer.valvesoftware.com/wiki/Dota_2_Workshop_Tools/Scripting