- Navhkrin [QUESTION] Will there be improvement's to GAS system C++ parts to reduce unnecessary code complexity and make it more straight forward to implement and extend GAS system?

We would love to do this but there are no immediate plans to.

- johnathanharkeriscool: [QUESTION] This is just a followup to my previous question about predictive effect removal. Suppose my players can be 'crouching' and I want them to be able to stand up/crouch without waiting for the server. Is a 'CrouchingEffect' the wrong way to handle this? Maybe there's a better GAS way to do it?

Not being able to predict GE removal makes this tricky. I can't think of a natural way in GAS to do this…

- praise_solek: [QUESTION] Why has a lot of the boilerplate stuff like GrantAbility and CancelAbility not been exposed to blueprints by default?

I don't see any reason why it couldn't be. In Epic's internal projects we usually grant abilities through some higher level system / set. But it seems reasonable to expose this for other use cases.

- Anderson Cruz To use the GAS plugin i have to write c++ code?

You can't really add attribute/attribute sets in BP only, so I would say it requires at least some C++ code.

- jsun1o: [QUESTION] Would we use the Attribute sytsem this also for a city management simulation game or economics simulation?

If you are referring to a simulation with 1000s or 100,000s+ of agents, I would not recommend giving each one an AbilitySystemComponent with attribute set subobjects. For a large scale simulation I would recommend taking a data oriented approach and packing your per-agent data as tightly as possible in arrays of structs that you manage yourself. This can be a hard problem: large scale simulations that also allow for high fidelity interactions on an individual level.

- maiconpintoabreu: [QUESTION] is it possible to grant abilities basen on the Slot that the Weapon is attached? For example; when you add the weapon on the Left or Right hand with different animations and damage values for the same weapon. Thanks

It is certainly possible to do something like this in GAS. However it would be up to the game itself to define the concept of left and right hand and how that plays with the animation system. But you can imagine granting abilities with different InputIDs that map to left/right hand: it would be up to a higher level system to do that setup work through.

- jsun1o: [QUESTION (if not already topic later]: Relation to UMG to create a ability interface?

I don't fully understand this question. I wouldn't personally put "ability system business logic" in UMG, but maybe that makes sense in some cases. In our games there is usually a game-specific layer between the UI system and ability system.

- Joseph Howard - Is anything possible with only (blueprints)?

The lack of BP based attribute sets is probably the biggest limiting factor, and some boiler plate/setup that has to happen in C++. It would be nice to improve this but there are no immediate plans to.

- polymorrah: [QUESTION] Why not attach attributes to the ability system component?

Don't fully understand this but if you are asking why UAttributeSets are not subobjects of the ASC itself (and instead are usually subobjects of the owning actor - e.g, siblings of the ASC though managed by the ASC) - this was historically done due to limitations of Default SubObjects in Unreal. I don't remember all of the details but I know for sure SubObject replication does not preserve deep hierarchies - it basically only supports subobjects underneath the actor.

- SniperGhost - [QUESTION] Is there any plans in C++ area where its really a pain especially for newbies while digging through the code with intellisense when using VS and VSCode

I'm sorry I don't have any tips here other than sticking with it! I will plug Entrian Source Search as a useful tool for navigating large code bases.

- Navhkrin - Is there any way to extract how much change an GE exerts on an attribute regardless of whether it was set by caller or attribute based on multiplayer game before the actual application happens on serv?

No, there is not a built in perfect way to do this unfortunately. There are so many customization points within a GE being applied to a target that it's impossible to really know ahead of time "what will happen if I apply this".

Memory is fuzzy but on Paragon we had something custom that could approximate the cases we cared about. One that came to mind was we changed the crosshair if we thought your next LMB attack would kill a minion… but it was basically specific to our LMB attack and to minions.

Sorry this isn't a better answer but this is one thing I wish the system was better at - how to expose the 'what will happen' / 'what will this do' to outside systems. This would be especially nice for AI.

- **Mighteemouse3: [QUESTION] one repo suggests for MP the AttributeSet and AbilityComponent should reside in playerState, where would the best place for them in MP setups**

I would personally recommend player state if your game has any form of respawn mechanic (where the actual pawn/character is destroyed and then a new one is created on respawn). This way your ability system data can persist across deaths/respawns. Otherwise you will need to recreate/store it somewhere while doing the respawn.

PlayerController may be a consideration but PC's do not replicate to non-owning clients, so this isn't a great choice for most MP setups if you want to be able to see ability system data on other clients.

- Pequeno0: [QUESTION] Does this ability system work well with server/client architecture with an authoritive server?

Yes though there are some caveats around prediction and rapid ability activation that can cause a player's latency to have a small effect on the simulation. I would say it is not perfect but good enough to ship video games :)

- Sudonymously: [Question] Should GAS be used to drive animations or should animations/animation state machines be setup our sit GAS.

Good question! I like to have gameplay (GAS included) drive animations as much as possible. Even going as far as setting animation play rates / durations and being in much control as possible. But this isn't what everyone prefers. I think it depends on the game and the team.

Making your gameplay simulation too dependent on precise animation evaluation can introduce issues in multiplayer and predictive movement. Be a bit cautious when using root motion animation to drive character movement. Generally long animations are good but lots of quick, separate, chained together animations will have issues. This is a hard problem and one we really want to fix in the future in Unreal.

- KyleL [QUESTION] Regarding the gameplay attributes, can you set them up in an actor component (like a health component) that you can add to a character or any other actor for reusability in a simple manner?

In general, yes that's the goal with something like GAS. But in practice there is a bit of setup and sub classing that has to go on before you can get to the "I've made a component that I drop on anything and now it has health and can be damaged".

- last_devil_: [Question] Is there a way to predict the ability level update?

I don't believe so.

- last_devil_: [QUESTION] Did paragon use custom Animation-Framework along with GAS for better results. In 4-series Shinbi Animation stream the developer talked about custom animation framework used in paragon and its publicly released.

Paragon had a pretty sweet locamation animation system for it's characters. But Someone from the animation team would have to explain the details and what aspects of that system have made it into the engine (or will in the future).

- yakistudios_: [QUESTION} GAS is very powerful but not very intuitive. Are there any plans to improve that?

We'd love to make it more intuitive but there are no immediate plans to.

- loraesh: how would you tell your cue to only fire for real damage and not 0.0 or something? didn't spot anything like that in the bp

I'm not speaking to the example that was on stream but one option is to emit the gameplay cue in code in your damage calculation only if it meets your conditions (damage was > 0, etc). I don't believe there is a generic way to condition a GE-contained gameplay cue from being emitted if the GE is applied (and for example does no damage).

- praise_solek: [QUESTION] How would a GameplayCue handle Animation Trails like a glow from a sword swing?

My naive programmer answer is there would be a particle system attached to the sword that makes it glow. But a tech artist could probably give you a much better answer.

- praise_solek: [QUESTION] If there were one question I could ask it would be this: What would be a good way to override TargetActors since the system built-into GAS spawns an actor just to make TargetData.

There used to be a "static" path on AGameplayAbilityTargetActor that could be taken to avoid instantiation but apparently that was removed years ago...

You are right though - all we need is the TargetData and instantiating the actor is heavy handed. If you want to replace it you will need to write a new AbilityTask to replace UAbilityTask_WaitTargetData. Both Paragon and Fortnite did this.

Targeting systems in general can be quite tricky, especially when needing to preview without executing the ability.

- last_devil_: [QUESTION] Which should be called first -> PlayMontageTask or RootMotionTask?

I might be misunderstanding the question but these are mutually exclusive if you are talking about playing a root motion animation? Or do you mean you want a non-root motion animation to play and play a RootMotionTask? If the latter, I don't think it would matter what order they are called in but I am not sure.

- **loraesh: [QUESTION] is there any "official" interaction between GAS and the builtin damage feature that's on AActor?**

No, they are completely separate.

- Rayzen__: [QUESTION] Is EPIC going to improve GAS ? Can we expect a major update or such ?

We'd love to make improvements but there are no immediate plans. I do hope the NetworkPrediction Plugin can be integrated into it once it is more stable to improve the networking aspects of the system but that may be a ways off.

- Prime GamingStryck96: [QUESTION] Is there an ideal way to handle attributes that are generally smoothly changing (like stamina use and regeneration for instance). It seems like periodic GameplayEffects are not designed to be run every frame.

Yeah, support for "continuous" attributes would have been really cool. Rather than periodic effects you can have like Stamina and StaminaRegenRate and do the regen yourself in a tick function. You can mod the attribute directly in code with UAbilitySystemComponent::ApplyModToAttribute

- Pequeno0: [QUESTION] Can you blend Montages and normal animation, so you could make an attack, and still move your legs (if you can move while attacking)?

Probably more of a question for animation programmers but yes this should certainly be possible to do.

- **Fractured Fantasy - QUESTION - what is the biggest consideration or issue you have experienced or run into by implementing this type of system?**

If you mean implementing GAS itself… The biggest challenge for me personally is generic networking and prediction are hard problems. Supporting many different workflows and use cases can be overwhelming. There is a constant struggle to achieve the right balance of

providing powerful tools that can be used in many different ways vs something intuitive that people can just jump into. It's hard to satisfy everyone :)

If you just mean writing a game with GAS - I would just make sure you have a good grasp on the system and expect to want to restart/reboot your use of GAS a few times. After using it a while you may think a different systemic approach works better for you than what you started with.

- **polymorrah: [QUESTION] Is there any sense in managing player states with the ability system?, e.g. stunned**

I don't fully understand this question. But yes, using granted tags to do states like stunned, haste, silenced, etc makes a lot of sense. Coming up with the rules around these states and how they all interact with each other can be tricky, but is ultimately game specific.

- **yamen zureikat - Hi, thanks a lot for the awesome steam! I have a question: can the GAS allow the player to customize the abilities themselves, like in Diablo or Path of Exile? Like for instance, they can add an explosion to the fireball when it hits, or add projectiles to it etc? Thanks a lot!**

It would be possible to build this kind of behavior out but it is not really a direct out of the box feature. It would really depend on the design of the customization system.

For the fireball example you would grant the player a passive ability that is listening for the fillball ability to do damage and then do it's own explosion on top. But it would be up to you to define that passive ability (not the player: just give him it as an option to unlock). There are also edge cases around things like, does the explosion create a separate instance of damage or just add to the initial fireball damage on the original target? This kinds of details can be hard to get exactly right.

- Pequeno0: [QUESTION] follow up on current answer: How does these constructed gameplay effects work on the network? Are they completely replicated, and not just by some id?

UAbilitySystemComponent::ActiveGameplayEffects is where the active GEs are replicated. The UGameplayEffect (data only bp asset) is replicated as a reference (id, basically) and the dynamic data on the FGameplayAbilitySpec and FActiveGameplayEffect are replicated "by value".

There is a "ReplicationMode" on the ASC that determines who gets what data replicated to them. For example you can put an ASC into minimal replication mode and it will not replicate the actual ActiveGameplayEffects data, only the primary side effects: granted tags and gameplay cues.

- VinceBGameDev - @Unreal Engine question : do you have a good explanation on how to store an attribute set? for an rpg or something like this?

I’m not sure what this question is asking - do you mean for doing save games? I’m unfamiliar with the save game system but saving data on the attribute set should be similar to saving any other actor owned data.