



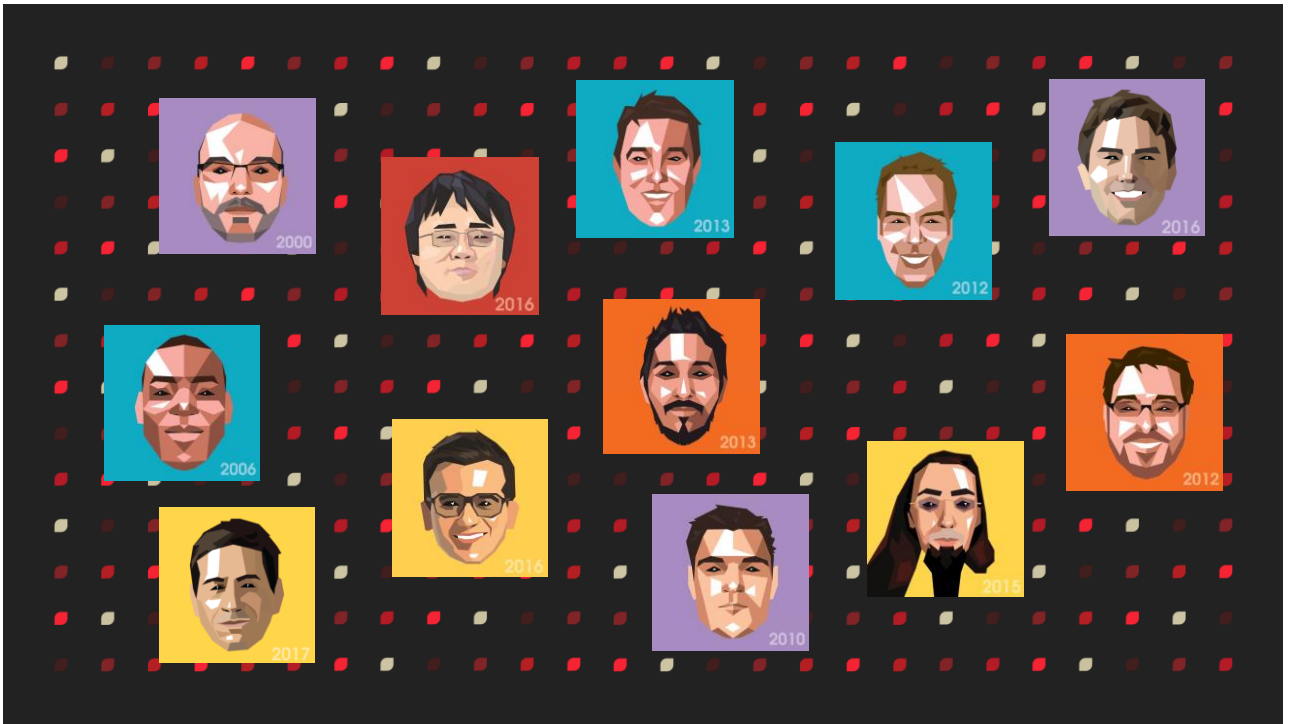
EVOLVING GOD OF WAR'S COMBAT

FOR A NEW PERSPECTIVE

Mihir Sheth
Lead Combat Designer
SIE Santa Monica Studio

Hi everyone, welcome to "Evolving God of War's Combat for a New Perspective". Thanks for coming by on your last day of GDC, after an exciting but very tiring week! As a quick show of

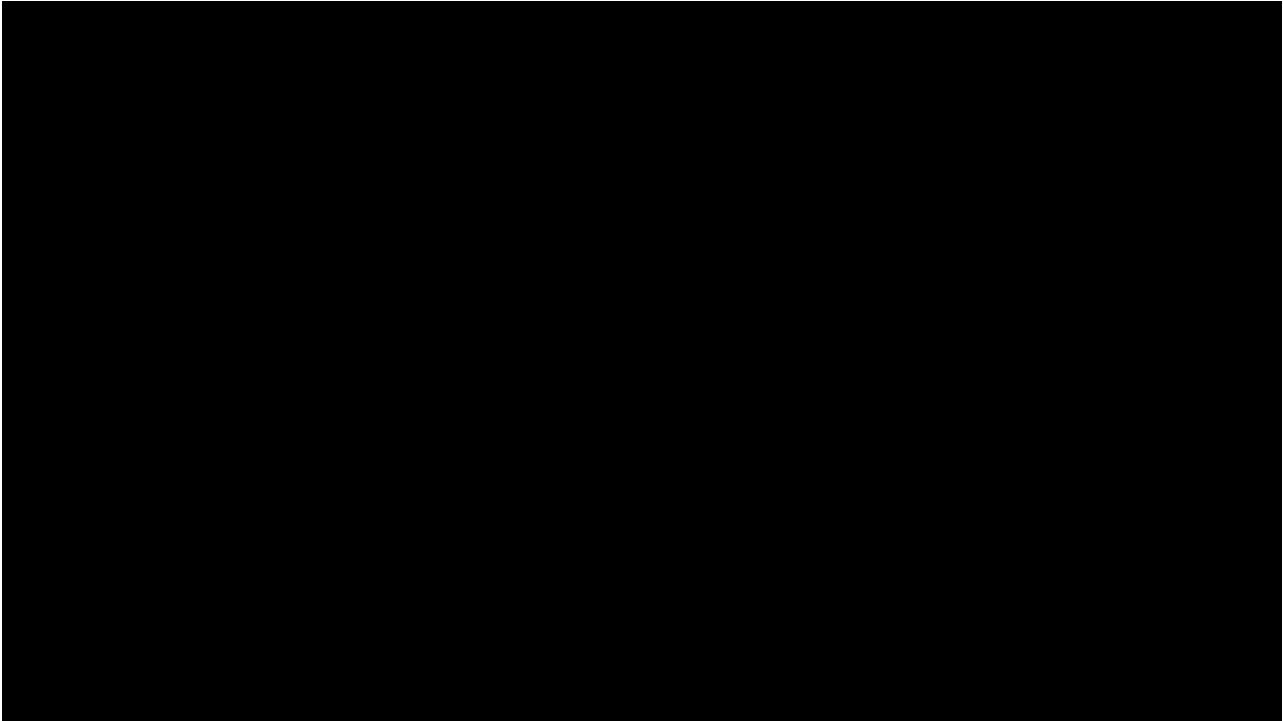
hands, who here played God
of War? Now, who here
played it on the Give Me God
of War Difficulty?I
apologize for your pain.



My name is Mihir Sheth and I'm the lead combat designer here at Santa Monica Studio. *click* Joining me during the Q&A will be our gameplay director, and former gameplay engineering lead Jeet Shroff.

Together, we worked alongside

click a host of talented
gameplay designers and
engineers in **evolving our
combat systems** on 2018's God
of War. Though I'm the one up
here presenting today, everything
I'm about to go through is the
culmination of a team's effort.
We've got a lot to cover, so let's
get started!



For those unfamiliar with past and present God of War, here's a small video to give context to the evolution in combat that took place.

God of War is a third person action-adventure game series where you play as the

Spartan Demigod Kratos, and use your legendary weaponry to overcome obstacles in a visceral delight. Clearly the two games shown look very different... but the gameplay design underneath both of them shares common DNA.

To talk about "Combat Evolution", we first need to look at the past.

WHAT WAS 'GOD OF WAR' COMBAT?

- Intensity and finesse
- Fun for all
- Offense power fantasy

...All of this from a fixed camera perspective

What was the traditional core of "God of War" combat? For over a decade, the series' combat has focused on a few central pillars:

- *click* **Fundamentally**

we wanted to adapt the intensity and finesse of Japanese action games to the West.

- *click* **In doing so**, we aimed to expand the audience by increasing the accessibility of gameplay, and bringing over the top bombastic presentation to the forefront.

- *click* **Kratos was a character** all about offense, and the power fantasy of our combat represented that by

making fighting your enemies feel like “playing with your food”.

And of course, *click* all of this was done from a pulled back, game controlled camera that framed all of the action in a way that was larger than life. Each game in the series built upon the foundations of the original by refining combat mechanics and escalating spectacle to epic

proportions.



With this new God of War however, the overall vision demanded a more grounded and intimate tone. This applied to all aspects of the game, including combat.

WHAT'S NEW?

- Grounded tone
- Leviathan Axe
- **Player controlled, third person camera**

This would fundamentally change the combat experience...

...AND IT HAD TO BE BETTER

There was so much new with this vision – the Norse mythology, a new melee and ranged weapon in the Leviathan Axe, and a new player controlled close camera that never cut.

And I'm of course skipping over a brand new companion. If you want to learn more about that and missed Hayato Yoshidome's talk yesterday, check out his talk "Raising Atreus for Battle" when it hits the GDC Vault.

So the idea behind this new camera was to create a more visceral experience that brought you closer to Kratos,

and to make the combat more deliberate and unflinching.

click This would fundamentally change the combat experience of the franchise –

click **And it had to be the better.**

GOALS

Reinvent combat for a new style of camera

Retain the core identity of 'God of War' combat

Well, that meant two very clear things:

click - We knew we couldn't refine and iterate on the combat formula that existed like previous titles had done. Instead,

we would have to re-imagine combat for this new style of game – and it would have to be largely governed by the camera.

click - And in this new style we also wanted to retain the core identity of what fans know and love about God of War's combat...



"UH... they should have pulled back the camera.
What were they thinking?!"

- The Internet, E3 2016

And... these goals drove us crazy! At the time these two things were completely at odds with each other! Fulfill an offensive power fantasy with fast and intense action.. but for a more

grounded game with a super close camera?!?

It was funny seeing people's response when we eventually showed the game for the first time at E3.. People thankfully responded really positively overall... but many kept saying "Uh... they should have pulled the camera back.. You can't see anything.." - the internet was experiencing what we agonized over for years!

BREAKING DOWN THE IDENTITY

Combo attack gameplay → Fun enemy hit reactions

Powerful attacks move you forward

Embrace One VS Many

Accessible and low barrier to entry

But let's roll back a bit to understand what I mean by that. To know why these two goals were a problem, we have to dive in a bit deeper into the tangible elements which define the combat identity of God of War. Now, obviously you're doing a lot of things

while playing the game, but I'm focusing on what you're doing MOST of the time - fighting basic grunt enemies such as the Draugr.

At the heart of God of War's combat design for moment-to-moment gameplay lies 4 tenants:

click - Responsive and combo-oriented gameplay leads to varied and exaggerated enemy hit reactions

click - Satisfying and powerful attacks move you forward physically through the space

click - Kratos can handle, and often excels, when he fights multiple enemies at once

click - Gameplay and controls are accessible, hassle free and are on the player's side

PAUSE



Dark Souls



Hellblade: Senua's Sacrifice



For Honor



Bloodborne

Now with that in mind..

click Most other action games with similar types of camera perspectives are designed to focus primarily on one versus one combat with a lock-on

system, not on fluidly fighting multiple enemies together. They tend to have combat loops which promote relatively defensive and cautious player behavior compared to traditional action games, as it's harder to see your surroundings.

The player's attacks typically don't translate them forward through the space too much in order to help with the camera framing, and it keeps

the combat aimed at gauging the distance and timing of attacks as opposed to manipulating fun enemy hit reactions... **and this is all for good reason!**

Players love these games, and these types of combat systems make sense for a closer camera since the combat is designed around it!

DESIGN CONFLICTING WITH VISION



Need players to be **confident** and **offensive**

Unfortunately this was not the case for us and our combat identity - we needed to stick to our special ingredients if we wanted to deliver a different, but true, God of War experience. We

needed players of all skill levels to be confident and offensive, and it didn't seem like this was possible with the proposed change. In fact, in the beginning, many of us on the combat team were really opposed to the new camera for this very reason!

But the vision was set, and after many initial months of squirming and fighting about it, we trusted in the vision and committed to the

thought that we were going
to make it work.

A ROUGH BEGINNING



And it wasn't easy... Our early attempts of making the camera mesh together with combat did not go well, as you might imagine. Frankly speaking, it was terrible to play, and it was almost impossible

for players to make sense of what was going on around them! Can you tell where all the enemies are in the video and what they are doing?

Players just panicked, evaded around constantly and mashed buttons without a semblance of strategy because the combat systems were fighting the camera in a way that made the game extremely clunky.

TAKING A STEP BACK

The difficulty needed to come from
the game itself

Not the controls and camera.

To solve this, we had to re-evaluate our combat late into the fall of 2016, and go forward with a strategy.

For players to play
confidently and
offensively we needed to

remove as much clunkiness as we could. After the initial adjustment of getting used to the game, the difficulty needed to come from the game itself, not the controls and camera. If we didn't get this right, players would be discouraged to experiment with our combat, and would maybe even rely on the pattern of play that they're used to in some of the games I showed before.

OUR APPROACH



So with these challenges and goals framed, we broke down the core combat systems and identified the aspects which we needed to solve: *click* **Tracking**, *click* **Targeting** and *click*

Engaging enemies... and I'm excited to go through them with more technical detail, and share some of what we learned in the process.

If you would like to hear more about the design process covered in this section in much more detail, Jason McDonald, our Design Director, did a great talk on this yesterday called "Taking an Axe to God of War Gameplay", and it should be

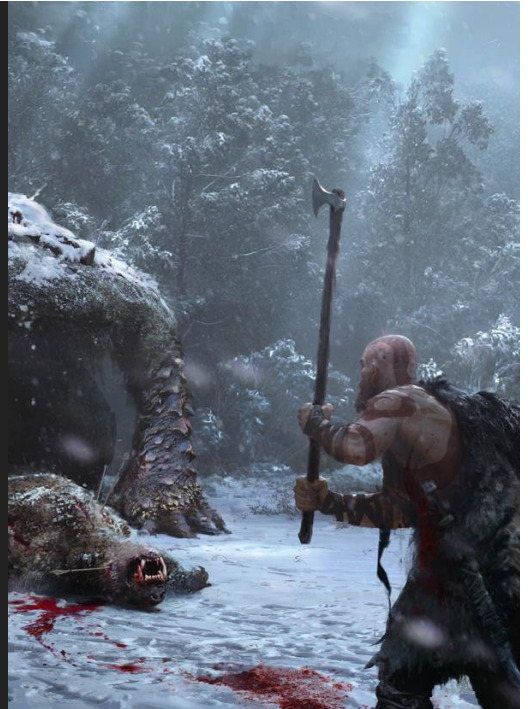
up on the vault soon.



TRACKING ENEMIES

Enabling players to be aware of threats in the fight

 Santa Monica Studio



Let's start with Tracking Enemies - by which I mean everything related to enabling players to be aware of threats in the fight

ENEMY AGGRESSIVENESS

Who are the **threats**?

- **Aggressive**: Actively attacking
- **Non-Aggressive**: Hanging back and waiting

How do we select who is aggressive?

In order to track enemies, the player has to have an idea of who the active threats are. You could be fighting 8 enemies at a time, but which group should you be paying the closest attention to?

Aggressiveness affects how enemies will behave and where they will position themselves. Our enemies can be either Aggressive or Non-Aggressive in a fight. Think of it like a fight scene in a kung fu movie. Aggressive enemies are those who are actively trying to attack the player in the front, while Non-Aggressive enemies hang around in the back and wait to become aggressive.

... So how do we select who is aggressive at any moment, and who isn't? We mostly updated our old Aggression scoring and token system from GOW: Ascension, but accounted for the new camera.

ENEMY AGGRESSIVENESS: SCORING

Aggression Score:

0-00-0-000

1. Can enemy become aggressive?

2. Aggression priority

3. Is player's current target?

4. Action Rank:

a) "On/Off Screen"

b) Angle from camera

c) Distance from Kratos

We evaluate all enemies in the fight periodically on a short timer, and give them a numerical aggression score built by the following criteria in order of importance:

***click* 1)** Can the enemy become aggressive? (This is to ensure that they not in a hit reaction, or a similar state)

***click* 2)** Their Aggression Priority - this is a number set by designers per enemy type, along with a range in which the priority is applied... so a high priority enemy (Ogre) who is super far away doesn't out prioritize a lower enemy (Draugr) who is right next to the player

***click* 3)** Is the enemy currently targeted by the player - I'll go over this soon in the Targeting Enemies section of the talk

***click* 4)** Their Action Rank - a calculation that looks at whether they are on or off screen, their angle from the camera, and their distance from Kratos

ENEMY AGGRESSIVENESS: TOKENS



Player tFightSystem participants								
Creature	A	T	S	P	Range	State	Prox	Attack
gobandit00 (508)	*	~	1	0	first		6	0
gobandit00 (504)	*	~	7	0	first		4	0
gobandit00 (510)	*	~	7	0	first		4	0
gobandit00 (508)			87	2	second		4	0

The enemies are sorted with this score, and are handed Aggression tokens from a fixed pool in order. Each enemy type can claim a different number of Aggression tokens to signify size or importance,

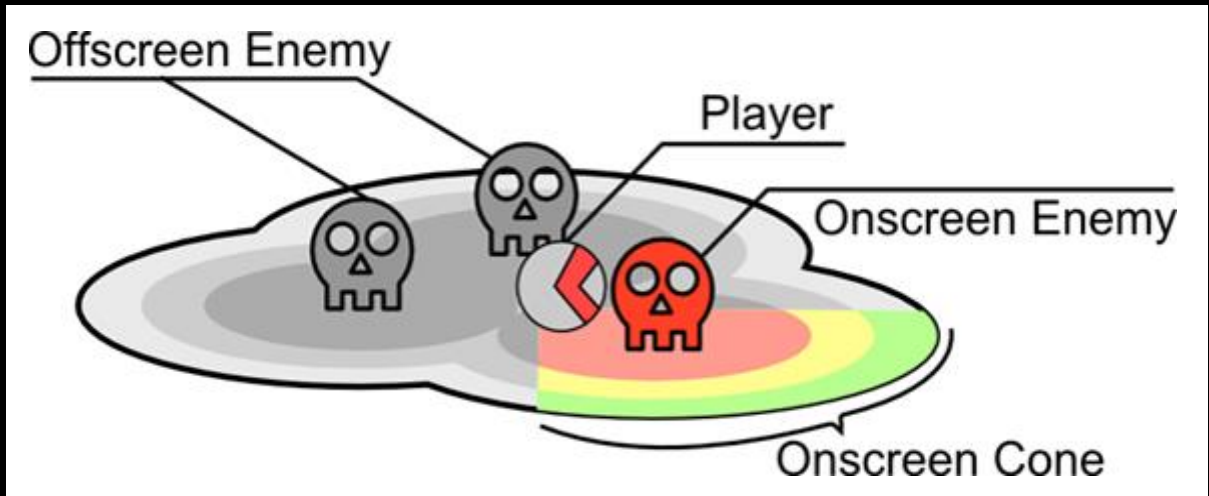
and after the max token count is reached all others in the list become non-aggressive. In this example, the player has a pool of 14 tokens.

In the table, *click* the outlined column on the right denotes how many tokens that enemy claims. After the third enemy, the pool of 14 runs out and the remaining enemy is set to be Non-Aggressive. *click* The

outlined column on the left shows who is aggressive, and as you can see, only the blue enemy was set to Non-aggressive. *click* You can see in the image that he is standing all the way in the back.

Careful tuning of these values helps to ensure that the player always has a manageable threat at any time. These values are also changed per difficulty to

either increase or decrease
the overall enemy
aggression.



Once we know which enemies are Aggressive and which are Non-Aggressive, we have to figure out where they want to be in the fight space. Positioning needed a complete overhaul and

was built from the ground up
with the new camera in mind.

ENEMY POSITIONING



Old GOW

- Surround Behavior
- Can see everyone



New GOW

- On or Off screen and Angle matters
- Depth perception
- Easier to see enemies in front

In older GOW games, enemies used their Aggressiveness to determine whether they would stand near or far from the player, effectively surrounding Kratos in two circles. There was no

notion of nicely spreading the enemies out, or caring about one area more than another. The pulled back camera allowed you to see everyone at all times so it was easy to perceive your threats, and non-aggressive enemies occasionally moved along the perimeter of the fight just to keep the scene more dynamic.

Now however enemies could be on-screen or off-screen,

and where they stand relative to your perspective is important. The camera perspective makes depth perception more difficult to gauge distances, and enemies in front obscured those standing behind them. As such, we needed to make sure enemies spread out comfortably.

NEW POSITIONING: FIRST ATTEMPT

Goal: "Find optimal positions"



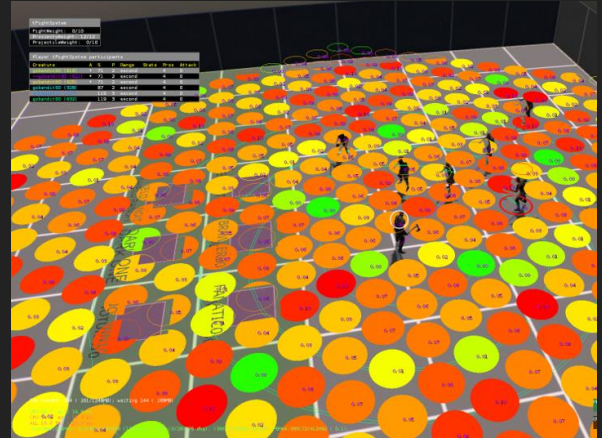
Early on in the project, we tried to find the optimal positions for each enemy using a weight-based positioning system. We used a fixed-size circular grid around Kratos, which would update periodically

as he moved around. Each grid position would evaluate itself by checking if it was on the navmesh and if it had a valid straight path to Kratos. Valid positions were then weighted based on many different heuristics.

NEW POSITIONING: FIRST ATTEMPT

Problems:

- Extremely complicated
- Constant invalidation
- No mental map → no confidence



There were a few valuable learnings early on from using this system, but over time it became clear that we needed another solution. The weight structure became monolithic, complicated

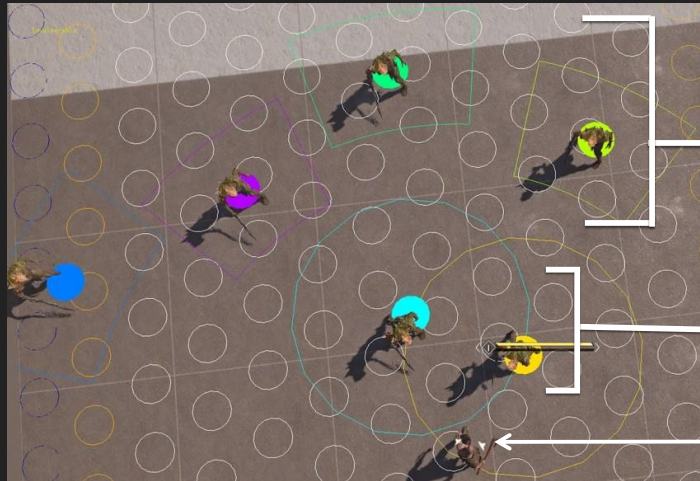
and messy - making it almost impossible to debug effectively. Weights kept constantly getting invalidated due to both player and enemy movement, and enemies frequently bunched together because of this.

click Most importantly, we saw from early internal testing that players immediately got scared as soon as an enemy went off-

screen, and shut down any kind of offensive or confident gameplay they were engaged in. It didn't look like God of War. We needed players to build a mental map of enemy locations in the fight so they could feel comfortable in combat.

NEW POSITIONING: SECOND ATTEMPT

Goal: "Prevent bad positions"



Non-Aggressive
Enemies

Aggressive
Enemies

Kratos

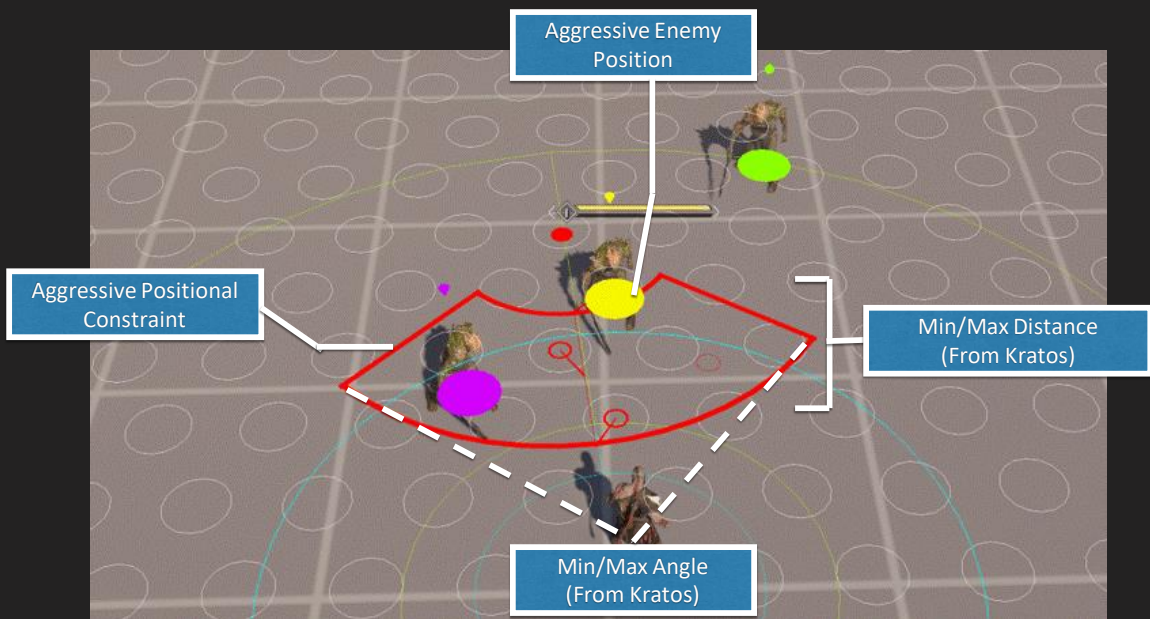
After almost a year of iteration, we scrapped the system entirely. This time, driven by the madness of managing the complex weights, we changed the goal from "picking optimal positions for each enemy"

to be something much simpler: "prevent enemies from getting into bad positions".

This led to a strict, zone constraint-based positioning system. The system required each enemy to have two elements: a Positional Constraint ("Where do I want to stand") and a Separation Constraint ("How much space do I take?"). As long as these constraints were met, no

position was better than any other.

Both constraints would take the form of zones, and would be fully data driven. These zones required a minimum amount of navmesh to be available, otherwise more lenient fallback zones would be used.



Instead of using a grid that moved around Kratos, we generated a grid on the navmesh in world-space.

click The Positional constraint we used for aggressive enemies was

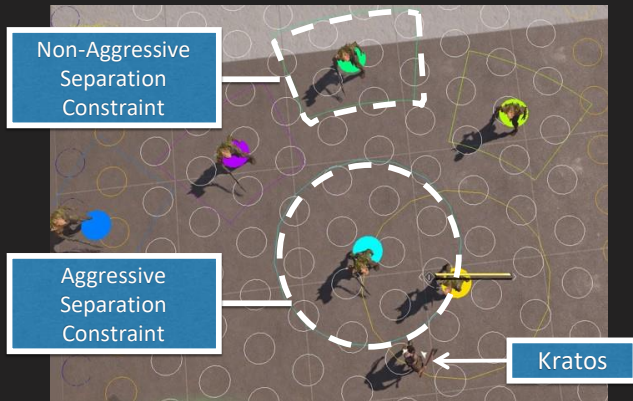
this kind of crescent shape in front of Kratos to keep them on screen, which you can see in red. This particular shape played well with the camera's perspective, and prevented enemies in the back of the zone from standing off screen as you moved forward.

Non-aggressive enemies would just stand around Kratos in a wide circle, similar to old God of War, since they

didn't care about being on or
off screen.

Aerial View

In-Game View



Separation Constraints – the ones that specify how much space the enemy takes – had different shapes. The Aggressive enemies simply used a radius to define how much space they took, *click* so

no other creature was allowed to stand in their circle.

Non-Aggressive enemies however had a wider, angled separation constraint that looked like a slightly curved rectangle. *click* This shape was helpful to spread out the enemies laterally in the back, but didn't take up as much space, since there could be many more Non-Aggressive enemies at any time than

there could be Aggressive ones.

This shape also meant that enemies in the back could be seen behind the enemies in the front. You can see how this led to enemies being clearly perceptible by these two images showing the same enemy layout from both an aerial, top down view and an one from the in-game camera.

NEW POSITIONING: SECOND ATTEMPT

Wins:

- Much simpler
- Less invalidation
- Visually pleasing

Problems:

- Awkward positioning
- Still trouble with **mental map**

This approach had much less nitty gritty control, but yielded far better results. The zone data was easier to author and debug, and the simplicity of solving the constraints led to both less frequent and less

noisy invalidation. The constraints also caused the enemies to spread out in a way that was much more visually pleasing.

click However.... It still wasn't perfect. By making all aggressive enemies want to be in the crescent shape in front Kratos, many of the off-screen enemies behind him who had recently become aggressive would walk directly into the back of him

while trying to get to their new position on screen. In fact, off screen enemies in general were still really hard to track. You didn't know where they were, and at any time one could turn aggressive and be a new threat from a blind-spot. Players still didn't have a mental map of the fight, and couldn't play confidently.

POSITIONING: FURTHER IMPROVEMENTS

Needed to make enemies be where you **expected** them to be...

- Try to preserve them as either on or off screen!
- Try to keep them in the same area!

We relied on this version of positioning for a long time on the project. In fact, it was only during the last few months during intense playtesting that we re-examined it and tried to address remaining issues

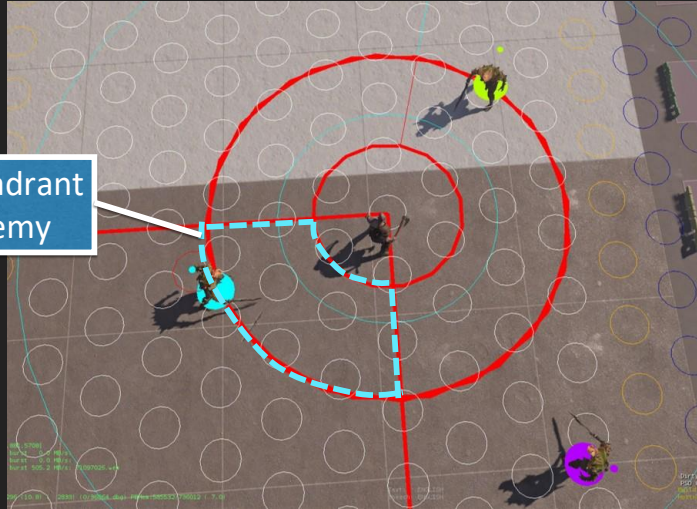
we knew existed. We had to figure out how enemies could position themselves in a way that you “would expect” as a player.

We employed new techniques to help **preserve** enemies as being either on or off screen - that is to say that an on-screen enemy should only go off of the screen if the player caused that to happen... and it should stay there! Similarly,

an off-screen enemy who turned aggressive should NOT be forced on screen again until you looked at it. This was all done in the logic that assigned the constraint zones to enemies.

POSITIONING: QUADRANTS

Zone shaped by quadrant
for off-screen enemy



Another technique we used was to keep off-screen enemies in their same world-space oriented quadrant relative to Kratos' position. This was just another zone constraint they had to

satisfy.

This way, enemies that you weren't actively keeping track of would stay relatively in the same area as you'd last seen (ie: an enemy currently in the back left of the players view would stay in that general area.. So if the player turned right and put that enemy off-screen, they would know that the enemy was now directly behind them).



Here's a demo of the system with enemies who are not attacking for debug purposes, but aggressiveness as a system is still on. Those are two different concepts.. I'll commentate on what's

happening, but the main
takeaway is to notice if the
enemies move in a way that
you would expect. Take a look
click ON VIDEO

POSITIONING: FURTHER IMPROVEMENTS

Wins:

- Reduced awkward positions
- Quadrants were a huge success
- **Mental maps → confident play**



These key improvements, along with many other small changes led to significant outcomes. We were able to solve the on screen/off screen problem, and prevent enemies from standing in awkward

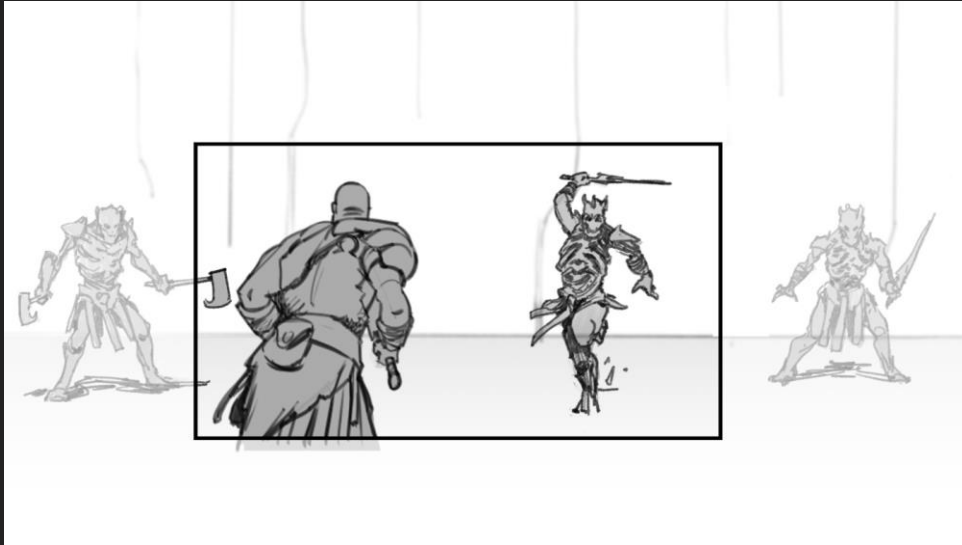
positions. Quadrants were a huge success in making enemy positions more predictable, and were especially helpful in more complex environments.

click Overall, these changes helped our goal of allowing players to make mental maps of the battlefield and begin to play confidently - **and this was evident in playtests!**

Players, whether they were aware of it or not, would start

to feel confident enough to take risks and continue to be engaged offensively even when handling multiple enemies.

ENEMY OFF-SCREEN INDICATORS



To help players further we wanted to make them even more aware of off-screen enemies and what they were doing, without it being so distracting that it shifted attention away from the threats

immediately in front of them. We knew we didn't want to do something as overt as a radar or mini-map, so we experimented with off-screen indicators.

Most players just need to know if off-screen enemies were attacking, but as they became more and more confident with the game it was helpful to get an idea of where even nearby idle enemies were to best

maximize their combat strategies. This information could encourage them to be more offensive instead of hanging back all the time.

ENEMY OFF-SCREEN INDICATORS



Initially used screen edges



Arrows around Kratos were much clearer!

Early on *click* we tried flashing the edge of the screen closest to the enemies off screen, but players thought they were damage indicators instead.. Not ideal. We tried many other

implementations which involved the edges of the screen, but finally settled on changing them to be arrows *click* which circled around Kratos and pointed more clearly towards the enemies themselves.



The red indicators are for incoming attacks, and the more subtle white arrows just show nearby idling enemies. It was important to design the white indicators in a way that weren't distracting for

players who weren't trying to use them. Ranged attacks actually had a slightly different purple color for players who paid attention - but again, the goal was to not overload information. If you saw a red-ish flashing arrow, you knew something was coming.

It's important to point out that the work done on the positioning system described earlier in terms of reducing

the noise and invalidation enemies, made the off-screen indicators far more stable, predictable and easier for players to process.

CAMERA ASSISTS

“What if players never use the right analog-stick?”

Led to **Strafe Assist**

Our indicators helped players track enemies off-screen, but what about enemies that are right in front of us? Controlling the camera itself with the right analog-stick in the heat of battle is an added

complexity this time that we needed to help make accessible for a wide audience looking to play an action game.

click We tried to find a solution for tracking enemies without relying on the player to constantly manage the camera by implementing our own version of a Camera Strafe Assist.

STRAFE ASSIST



The camera will automatically try to adjust and keep relevant targets and threats in view when the player moves left or right. To do this, we weight enemies to find a center point for the camera to

focus on. In particular, evading laterally really showed this behavior in action.

We also eventually added a behavior to re-orient the camera automatically in the direction of Kratos' attacks to ensure that we reduced frustration for players struggling with controls.

As many of you likely know, anticipating the player's

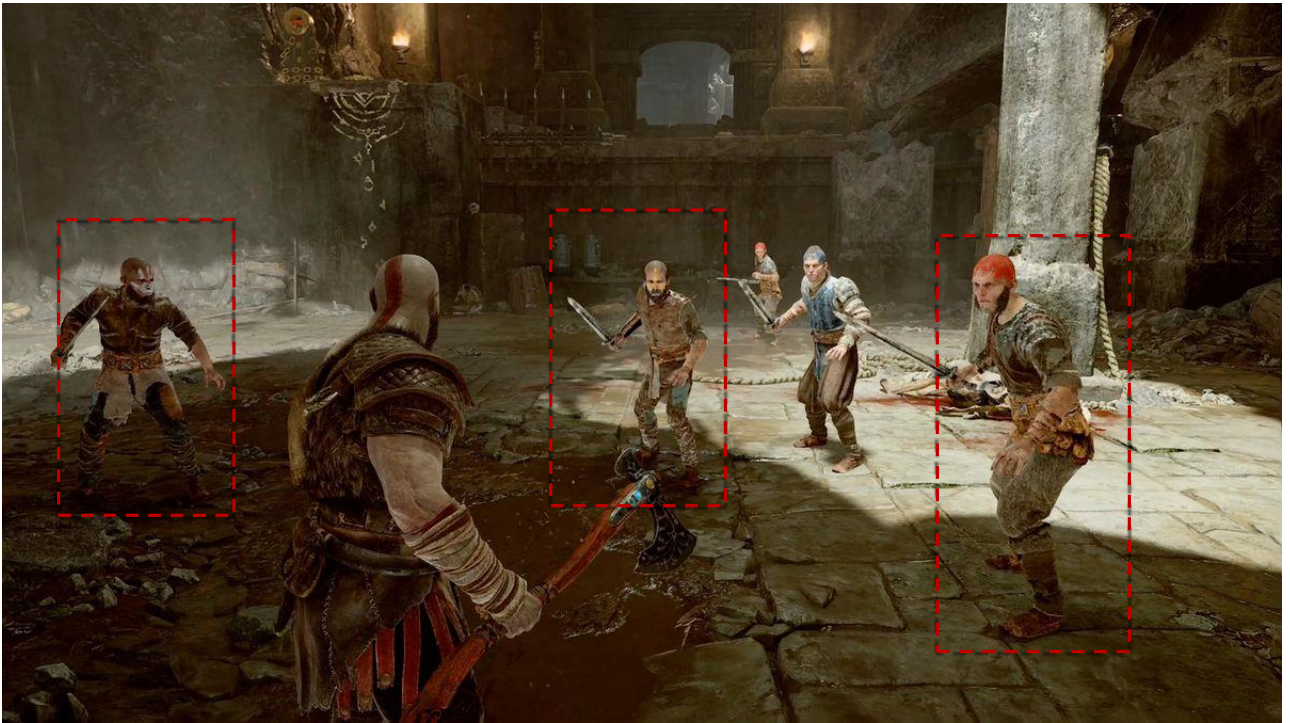
intentions with how they wanted the camera to behave is a difficult task and we didn't always get it 100% right. Our playtests proved, however, that these camera assists dramatically improved gameplay for most players, and those who found them frustrating could just turn them off in the settings menu.



TARGETING ENEMIES

Empowering players to attack who they want

Now that we've gone through systems that helped players Track their threats, let's look at others which helped players choose who they wanted to attack - by diving into Targeting



In a common fight scene like this, which enemy should Kratos target when the player presses the attack button? Should it be the bandit on the right side because he looks like he's the closest? What

about the one in the middle because he's in the center of the screen? OR should Kratos just attack directly in front of wherever he's facing without aiming at anyone in particular?

Target selection is an extremely important system for us, since we need to allow players to fight multiple enemies at the same time fluidly due the one vs many fantasy. **It's a core part of**

our combat identity. We needed to make this possible without requiring players to use a lock-on system - which was a directive from the top.

A new challenge was also that the Leviathan Axe enabled both ranged and melee combat on the fly, and switching between them should be seamless. A lot of time was spent on this problem in particular, in

order to get it feeling just right.

MELEE TARGETING

- **Old GOW**: targeting was left-stick driven
- **New GOW early on**: “Let’s make it like a shooter!”

Both of these didn’t work...

Let’s start with melee targeting. In old GOW games with the pulled back camera, we relied heavily on the left analog stick to decide who Kratos should target with his attacks. The problem with

doing something like that for the new game is that we didn't want Kratos attacking into the camera if you pulled back on the stick. One early approach we took was to "make it like a shooter" and have it be entirely dependent on the reticle point in the center of the camera.

click This didn't work - it wasn't intuitive for our style of game, and players kept complaining "Why can't I hit

that enemy - he's right there!". We weren't able to get their intent across, and the result was gameplay that felt clunky.



The solution ended up being a mix between both the left and right analog stick, as well as other factors. Let's take a look at it in action. You'll see a green circle on whoever Kratos' target is at any

time. *click ON VIDEO*

after video ends

As you saw in the video, we still tried to retain the left stick as the main way to show intent as much as we could, so that it felt familiar to players.

MELEE TARGETING: WEIGHTS

Solution was mix between:

- Camera facing (right-stick)
 - On or Off screen
 - Horizontal angle to camera from Kratos' position
 - Bonus for center of screen
- Player intent (left-stick)
- Distance from Kratos
- Already current target

*Items in green could force the target candidate to fail selection

I'm not going to go into them in detail, but for those curious, here's a list of the factors that played a role in target selection for reference if you want to take a picture.

As long as there is a viable enemy candidate, the targeting system will always choose it over not having any target at all - You can't choose to attack into empty space instead of a valid target. When there are no possible targets, Kratos will always attack directly forward along the direction of the camera. You can't steer his attacks left, right or backwards.

It's also important that we fulfill the fantasy of being Kratos - we don't want him to miss his attacks clumsily when enemies are right in front of him. As with past God of War games, If Kratos has a target, all of his attacks are automatically rotated to face them.

THE PROBLEM OF DEPTH PERCEPTION



One interesting issue that we saw within the studio during the development was that people controlled the camera differently... Cory Barlog, the game director, always had the camera flat, aimed right at

the horizon line to get the best cinematic presentation. Those of us who played the game correctly, of course, angled the camera down towards the ground so we had a reference point for judging distances.

Having gameplay that was on your side meant that regardless of how you played, we knew it felt terrible to attack and miss an enemy who looked like it was

right in front of you, even if it really was a good distance away.

'SUCK TO TARGET'

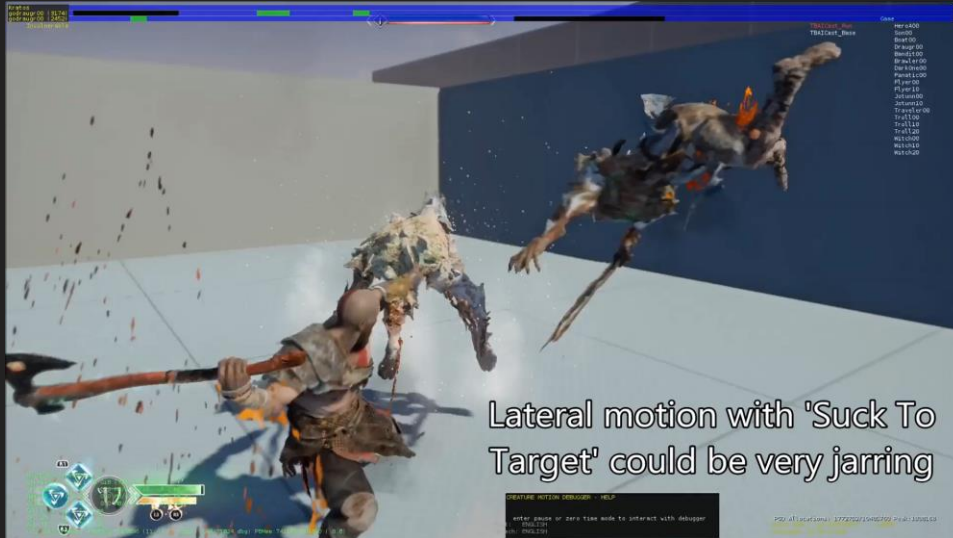
Used old GOW: Ascension 'Suck to Target' feature

- Each attack specifies: **range**, **speed** and **stop distance**
- Kratos will "suck" towards his target, making it easier to hit enemies

We re-used an old God of War: Ascension feature called Suck-to-Target on most of Kratos' attacks to help with this issue. Each attack specified a range, speed, and stop distance, which the engine would

use to "suck" Kratos towards his target. This would make it significantly easier to hit enemies.

'STT': PROBLEMS



As with most older systems, STT had problems. From this new camera perspective, the behavior often felt jarring and disorienting. This was especially true if an enemy was on the side of your

screen and Kratos "sucked" to it laterally - since the same distance would appear to be much longer because, once again, depth perception.

'STT': PROBLEMS



Worst of all was the case where you would be sucked in - only to still miss your attack because the enemy happened to move right out of range during that time!

'STT': IMPROVEMENT

Old 'STT' Problems:

- Disorienting
- False positives

New 'STT' Wins:

- Ensure contact or fail gracefully
- Scale STT based on angle to target

So the old STT had problems.. We had to update and change the behavior to help ensure that you reached your target at a specific time in the animation – kind of like a simplified form of

motion warping.

To help with the jarring lateral suck problem, we scaled the range of STT depending on the target's angle to Kratos: the wider the angle, the lower the range.

This way, Kratos would miss his attacks more frequently against enemies away to the side of screen compared to enemies in front of him, even if they were at the same

distance.



We were pretty happy with the results! Some of the examples of Suck To Target in the video are obvious over large distances, but it's happening much more frequently than you might realize.

You'll notice it more when the

camera is aimed downwards
towards the ground.

...

Fun fact: we toned down this
'suck' distance dramatically for
our hardest game difficulty,
"Give Me God of War", where
we wanted players to pay more
attention to their positioning in
combat.

RANGED TARGETING



For our ranged combat targeting, we looked to shooters for guidance on building both Aim Friction and Zoom Snapping. Aim Friction refers to system slowing the reticle over targets as you pass by

them, and zoom snapping is the system which snaps the reticle directly to a nearby target when you pull aim.

Both of these systems were tuned very liberally, as we wanted to encourage players to mix ranged and melee attacks easily, without hassle.



In doing so, we made it on the players side. Kratos could both swing and throw while attacking an enemy fluidly.

LOCK-ON

...Some players just want it.

- Added extremely late in the project, close to ship
- Made many encounters clunky...
- Players approached the game differently – **but that was OK**

I know, I know... I talked earlier about how other games use lock-on, and that it isn't very "God of War". And I meant it - it was a directive from the top to make the game completely playable

without a lock-on system,
since managing lock-on can
be cumbersome and
frustrating for many players..
And honestly, for most of
development, right up until
close to ship, we didn't have
it at all.

click In the end however,
we realized through playtest
feedback that some players
just always wanted lock-on,
partly due to the fact that
they were already familiar

with it from other games and expected the same. Players approached the game quite differently with lock-on - and that's OK. But it also meant that we needed to develop a lock-on system late into the project that could work, even if the game wasn't designed around it. Many players might have experienced frustrations with using the system for this very reason.

That being said, we

attempted to adapt it to suit the needs of our game, and thought about how it could still adhere to our identity tenants.



We used our already existing melee target as the initial locked target when the lock-on system became engaged, so the transition would be smooth. The right stick could be flicked in a

direction to change the target selection in screen space, as opposed to cycling through an ordered list, to make it more intuitive when fighting multiple enemies.

We decided to allow players to adjust the camera's elevation even when they were locked-on to a target, unlike most implementations of the feature, so they could see elevated points of interest, or even just take in

the environment, without
losing their target.



One particularly interesting feature we added to lock-on happens when you pull aim. We allow you to aim around freely, but retain your lock-on target, so you'll return to it when the aim button

is released.

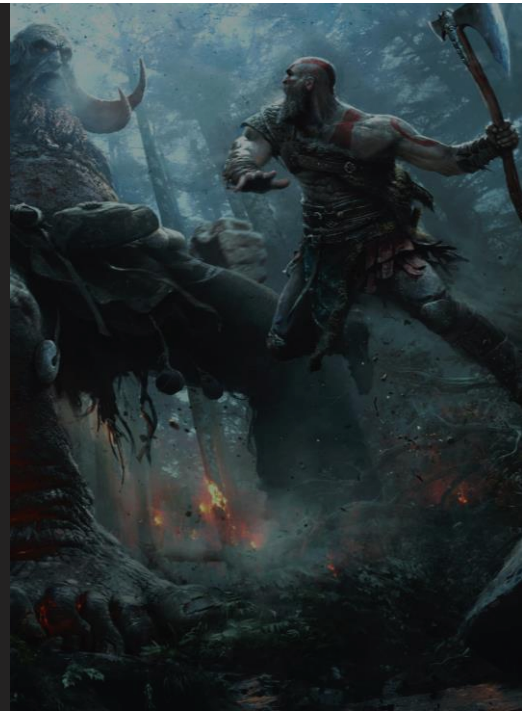
Keeping with the goal of supporting fights with multiple enemies, this feature allows players to handle two enemies at once even when using lock-on, as they could focus primarily on their locked-on target and occasionally throw their axe at another.



ENGAGING ENEMIES

Allowing players to continue combating their targets effectively

 Santa Monica Studio



Now that we've covered systems that helped players Track and Target their enemies, let's focus on those which helped players continue to combat them once they had been engaged.

THE PROBLEM WITH ANIM TRANSLATION

GOW enemies translate backwards in hit reactions...

...As a result, Kratos is always moving forward to keep up.

So, to better understand this, let's take a minute to talk about animation translation, and how it affects engaging enemies in combat.

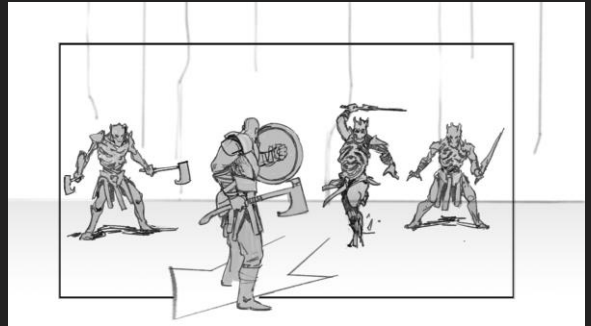
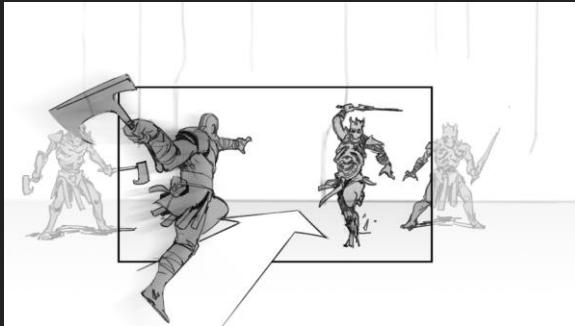
I mentioned during the

introduction that having exaggerated & fun enemy hit reactions, as well as strong attacks for Kratos which move him forward through the space, were both integral to our combat system's identity. These two concepts play into each other due to animation translation.

GOW enemies translate backwards substantially during hit reactions. As a result, Kratos is always

moving forward to keep up. Eventually, Kratos and the enemy keep moving in one direction and cover a large distance over the course of a combo.

THE PROBLEM WITH ANIM TRANSLATION



One problem is that animation translation like this was challenging with the new camera. It's easy to knock enemies off screen to the left or the right and zooming forward through the world is

disorienting as you see the environment pass by rapidly.

There's also the fundamental issue that in this perspective, moving FORWARD is naturally disadvantageous since you are reducing the amount you can see of your surroundings. *click* On the other hand moving BACKWARDS is naturally advantageous, as it allows you to see more. It's one of the reasons why players are

naturally drawn to kiting, or defensive gameplay patterns with this type of close camera.

We had to be careful with how we balanced the forward movement of Kratos's attacks, which helped deliver their power and satisfaction, with the disadvantageous state that they put the player in.

ANIM TRANSLATION

- Reduced translation on **enemy basic hit reactions**
- Reduced translation of **Kratos attacks**
 - 'STT' added the **umph** we needed
- **Procedural translation adjustment** via script was awesome for iteration

Because of this, we had to revisit how we authored translation in many of our animations.

click We reduced the translation of basic enemy hit reactions substantially

from how they were made in the past, and we even reduced the translation of Kratos's attacks.

click This worked out well, since STT could still allow him to reach enemies with his attacks and it also provided enough movement to give the satisfying 'umph' we needed, but he wouldn't go zooming across an empty room if he had no targets.

click Playing around with all of this took a lot of trial and error, and having the functionality in our design scripts to procedurally scale the translation of animations was incredibly helpful for testing outcomes with fast iteration speed.



This video shows the problem of highly translating reactions causing enemies to go off screen. The reactions themselves are fun and satisfying, but it's extremely frustrating to

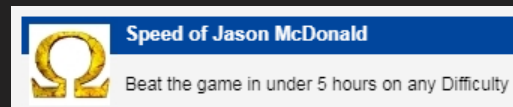
not be able to follow up on your target - with the added insult of having your target switched involuntarily back to another onscreen enemy. Even worse, the enemy that you just beat up is sure to return as an off-screen threat in a few seconds.

We were punishing you for attacking with your combo and causing fun hit reactions - the exact opposite of what our combat should be about.

STRIKE ASSIST

“What if we just kept the enemies you hit **on screen**?”

- Legendary game designer **Jason McDonald**, Fall 2016



Time to talk about one of my favorite features - Strike Assist – and how it helped solve this problem.

In a stroke of pure brilliance, our then Lead Gameplay Designer asked

the profound question..

“What if we just kept enemies that were hit... on the screen?”. *click* Seriously, check out Jason’s “Taking an Axe to God of War Gameplay” design talk when it’s available online. *click* There’s even an OG God of War trophy named after him!

Ahem, anyway. After a bit of back and forth Jeet and his programming team came up with something really cool.



And it looks something like this..

With Strike Assist, bare with me for a second, attacks cause their victim to snap to an initial orientation at the

beginning of their hit reaction, such that their adjusted trajectory, will now go towards a blend between the camera's facing vector and their original trajectory. And if that was too confusing, all that means is that enemies are now pulled onto the screen if they were going to go off of it - the larger the delta, the larger the correction.

Each attack can specify its

own blend, so some attacks can say "Make the enemy go to the center of the camera", or "Pull it just enough into the camera to stay on screen". Naturally, the farther the translation of the reaction, the more you'd see the effects of this. There's a lot of complexity to this feature and how we used it, but this is the basic overview.

STRIKE ASSIST

Wins:

- Stopped enemies from being hit off screen
- Successive hits on the same target pulled it to the center
- Kratos moves towards his target, so he would go to the center too
- Made transition to aim camera easier

With Strike Assist enemies stopped being hit off screen. Successive attacks in a combo on the same target would pull it more and more into the center of the camera.. And since, if you remember, Kratos rotates to

face his target when he attacks, this eventually led to both Kratos and his target translating back down in the camera's direction, which was great for framing.

As a bonus, this made switching between melee attacks and ranged targeting easier, as zoom-snapping worked better if the enemy was already near the reticle - which of course is near the center of the screen.



This feature led to a bunch of other new discoveries as well! See, players have control of the camera... so they could use this to "aim" where they wanted their enemies to be hit. To be clear: **This was huge!**

Building up combat systems with this new camera had led to a new type of moment-to-moment gameplay, and expression of player intent and skill.

For example, strategies such as intentionally herding enemies together before pulling off a massive Runic Slam Attack, or moving the camera to direct an enemy to fly off the side of a cliff with the last hit of your combo

were now possible - it was so intuitive through just 'looking at what you wanted to do' that most players had no idea that this was even happening at all.

Strike assist may have possibly been most important aspect in making the new combat system come together, and it was huge victory for the team!

HIT REACTION COLLISIONS



Apart from Strike Assist, we used a few other tricks to help players engage their enemies by allowing them to keep up the offense while enemies were already playing hit reactions.

Many of the hit reactions themselves would fire invisible collisions, allowing enemies to bump into each other and even hit enemies to their sides, despite there sometimes being no visual contact. This rewarded players by helping them stay safe, since enemies who were 'bumped' were sure to not attack them until after they had recovered.

DISRUPTING ENEMY AGGRESSION



Another trick was disrupting enemy aggression. The close camera naturally makes Kratos and his target take up a large part of screen real estate and obstruct the player's vision. We

realized that we would have to incentivize offensive gameplay even more if we wanted players to feel confident despite being able to see less.

Remember the Aggression system – the topic I covered earlier about how we choose which enemies are active threats? We made some simple tweaks to it to handle this: If an attacking aggressive enemy gets

interrupted and plays a hit
reaction, it retains its
aggressive tokens
temporarily instead of the
relinquishing them
immediately for others to
use.



For a short period of time, that enemy would block other non-aggressive enemies from taking these tokens, but couldn't do an attack itself since it was already in a hit reaction. This effectively gave the

player a “**power play**” every time they were being offensive.

If they extended their offense for too long without worrying about other threats they could be punished, but for a window of time they'd have the advantage. In most scenarios on our normal difficulty, say against a pack of draugr, only 2 enemies would be allowed to be aggressive at any one time.

This meant that the player just had to worry about one other enemy apart from the one Kratos was already engaged with - which isn't all that bad to keep track of.

Enemy air reactions specifically would prolong this effect for more time giving you yet another bonus to keep your combos going despite the threats around you.. Speaking of..

AIR JUGGLING ENEMIES



At first we didn't have enemies getting launched or juggled in the air due to the new grounded tone. However, juggling enemies is part of the "playing with your food" identity we needed to retain, so we

had to solve juggling for our new combat system and close camera.

The largest issue is that enemies could now be constantly hit upwards, until they are above the camera and you can't see them anymore. Not to mention, the Leviathan Axe had limited melee attack range, and since we removed the jump button from this entry in the series there would be

no way for players to stay engaged on a target like that.

To make matters worse juggled enemies took up a lot of real estate on the screen and occluded nearby threats, so players rarely felt safe while attacking them.



To tackle the first problem of enemies exiting the camera frame, we implemented a “float height” system which could be used to tell an enemy how high it was allowed to go relative to

Kratos. If the root joint of the enemy ever went higher than the specified float height, some correction velocity would be applied to keep it in bounds. This could be tuned per enemy hit reaction move by designers in script.

We tried to make this look as natural as possible, while still keeping enemies low enough where they were in range of melee attacks with the Axe. It's worth noting that to get

this feeling right we had to
animate the characters
falling instead of relying on
gravity or physics, since the
trick was that the bounce of
juggling always had to feel
good.



Enemy air reactions also warranted the same technique we used on ground reactions by having invisible impact collisions that hit other nearby enemies... except we made them much

larger than their grounded counterparts.

Enemies who could fire ranged projectiles from afar however were still always a threat, so players still had to be smart about when they should, and when they shouldn't, juggle an airborne target.



LOOKING BACK AT THE JOURNEY

So, after the long journey of development, what did this process yield for us in the end? Was it possible to make a combat system within a close and intimate camera feel like an honest evolution of God of War?



Not all players might have shared the same experience, and there is still much room to grow in all of the aspects I've covered in this talk, but by and large, it was rewarding to see most players

engage in our combat in an **offensive** and **confident** manner despite the many issues posed by the new close camera.

By keeping our combat identity at the forefront of all of our systems for Tracking, Targeting and Engaging enemies, we were able to work towards gameplay that felt satisfyingly familiar yet fresh from a new perspective. The result was a more

deliberate and **unflinching**

God of War combat
experience that felt authentic
to the franchise's legacy.

This line of thinking and
problem solving, in terms of
breaking down your
gameplay's identity and
keeping in mind at all times
during development, can
apply to other IPs in any
genre which are similarly
going through a challenging
reboot or re-imagining.



Reaching our results took a very, very long time of more than 3 years of constant iteration and testing.

This was a crucially joint problem between our

gameplay design and programming departments and involved many talented people. We needed designers deeply involved with the technical implementation, and programmers invested in the design context and player experience – this was true for both on-the-ground work as well as department leadership.

As a result, these two

departments work very closely with each other, both in terms of feature development and proximity at our studio.



Finally, the close camera seemed counterintuitive to almost all aspects of GOW style melee combat.. When traditionally “good design principles” conflicted with vision, the team took a difficult risk and

committed to the latter.

The process was challenging, and there was a lot of doubt throughout development...

But it was what the game needed to shine as a whole, what the combat needed to evolve in a bold new direction, and at the end of the day, it's what made the journey absolutely worth it.

Our Journey, Your Story

We're hiring for what's next!

We're expanding our family across disciplines and would love to meet you. Please visit sms.playstation.com/careers for all openings or drop us a line at sms@sony.com



@santamonicastudio



@SonySantaMonica



@santamonicastudio

And that's the talk! If you're excited by anything I've gone through today and would like to join us on our next journey, please get in touch! We're hiring across many departments, including combat design

and gameplay programming,
and we'd love to hear from
you.

Thank You!

Lead Combat Designer, **Mihir Sheth**:
mihir.sheth@sony.com | [@youtheremehere](#)

Gameplay Director, **Jeet Shroff**:
jeet.shroff@sony.com | [@theshroffage](#)



Thanks so much, it was a pleasure, and Jeet & I can now open it up to questions.



**Santa
Monica
Studio™**