

ANIMATING AN AGENT OF MAYHEM

Mike Jungbluth
Lead Animator, DS Volition
@lightbombmike



**AGENTS
MAYHEM**

GDC¹⁷

Hi, I'm Mike Jungbluth, Lead Animator at Deep Silver Volition, and I'm here to show how we animate an Agent of Mayhem.



And what is Agents of Mayhem? The way I personally like to think about it is GI Joe meets Saints Row. It's an open world game with vibrant, diverse characters, an 80's cartoon aesthetic and the over the top, irreverent tone of past Volition games. But why explain it, when I can show a trailer with some of our E3 announcement footage.



Working on this game has been a dream project for me. It requires the animation department to focus on character development that integrates both narrative and game design. We have the ability to implement and drive game feel and fluidity of the characters. And it is wrapped up in a stylized aesthetic and tone. What more could an animator dream for?!



But as with every dream, there are challenges that threaten to turn it into a nightmare. And for us, the scary part of our dream was a large cast of unique, playable characters in an open world. This was because each character needed:

AGENTS MAYHEM

THE CHALLENGE

- Out of Combat Movement
- Combat Movement
- Combat Abilities
- Emotes
- Interactions
- Vehicles

Out of Combat Movement. Combat Movement. Combat Abilities. Interactions. Vehicles. Essentially, all of the things . Which means the million dollar question was, “How many characters do we want to play as?!” Answering that question meant defining and understanding a number of variables.



A good place to start answering that question and understanding expectations was by looking at what other games were out there at the time. But we had no direct competitor to give us all of the answers.



Games with multiple hero characters are largely arena based, focused on one primary mechanic, often combat. Fluidity and feel of multiple mechanics was fairly limited.



And open world games are largely focused on one playable character with a deep suite of moves that fluidly support all mechanics. Our directive was that we wanted to focus on the unique, individual personalities of characters in a MOBA but with higher fidelity of movement to be competitive with other open world games. So again, all the things!

AGENTS MAYHEM

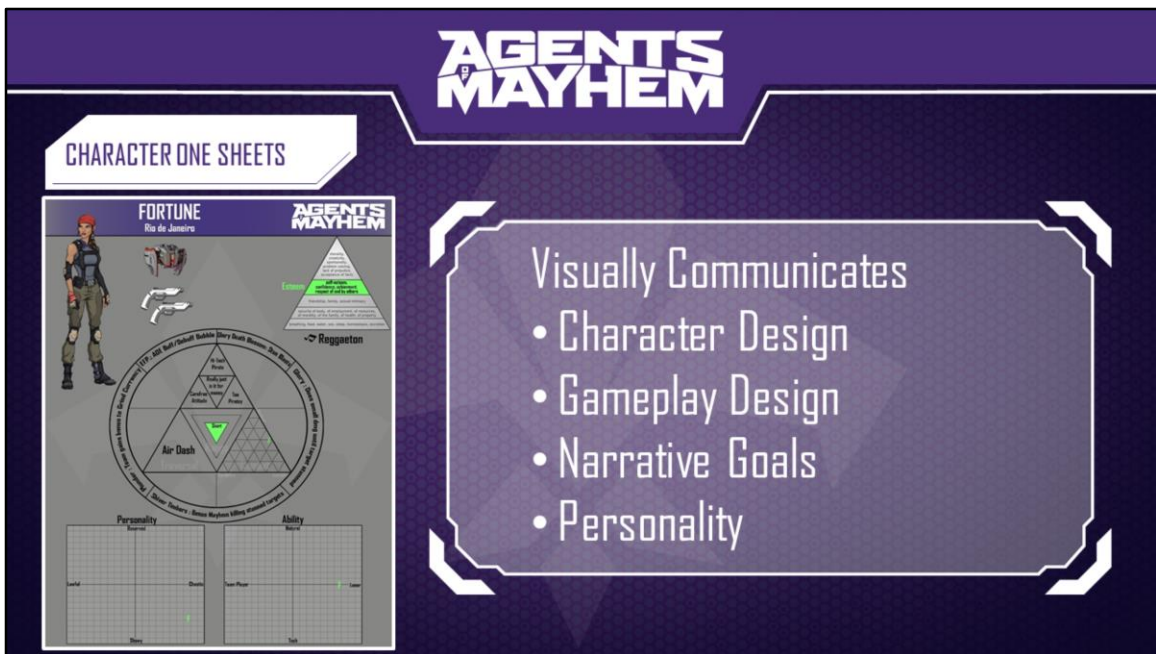
TEAM & TOOLS

- 8 Animators
- 3 Animation Programmers
- 3 Traditional Technical Animators
- 2 Editor/Animation Graph Technical Animators
- Motionbuilder
- New Proprietary Editor & Node Graphs

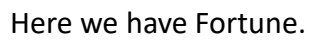
The one thing we did know however was the team and tools. This is the largest animation team Volition has had on a project, with twice the number of animators and dedicated technical animators focused entirely editor side for the majority of the project. This was also the first project to use Motionbuilder, as 3dsMax/Biped was used on previous games. So we certainly had the support of the studio to rise to the challenge.

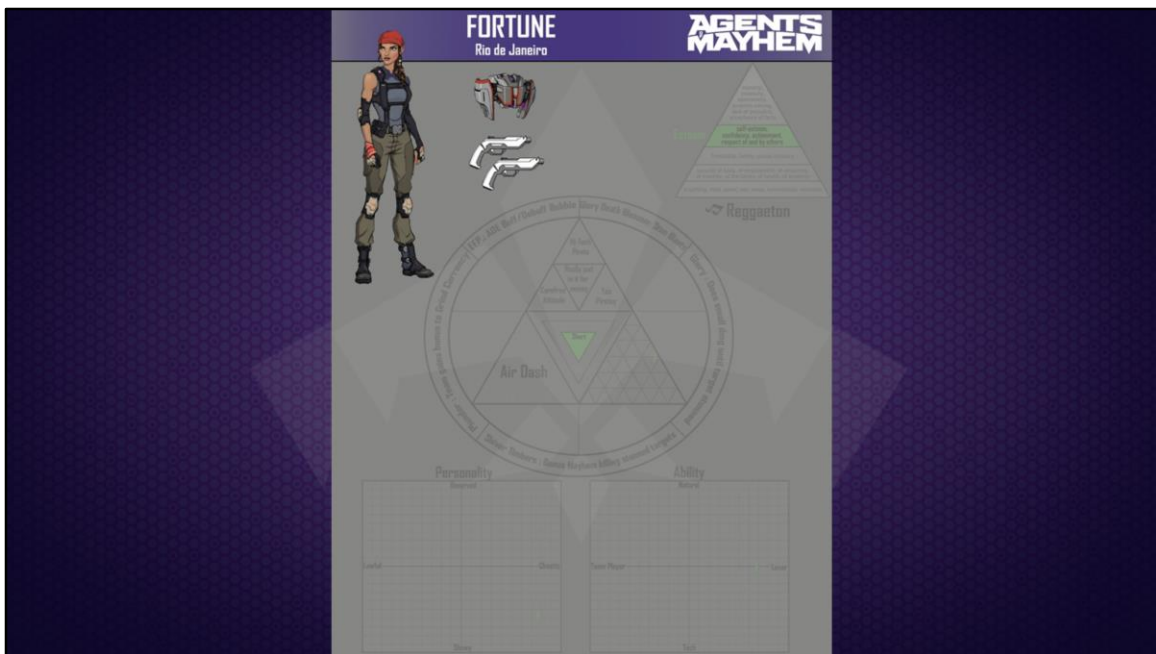


But before we can start animating the characters, we need to know who they are. We need to understand both the design and narrative of each character so that we can communicate a fully realized and cohesive performance. And the way we make sure we are consistent and clear, is with character one sheets.



With these sheets, we can quickly define all of the core aspects of a character that we need to communicate with our animation. We see the art, the gameplay design, the high level narrative goals, their personality, and motivation. Instead of having a wall of text to always refer to, this provides a consistent and clear reference point for everyone on the team. So let's break down each portion of the one sheet, to see what we consider necessary and important when animating an Agent of Mayhem.





First up, and pretty straight forward, is the character's name, where they are from, their concept art, and their signature weapons and equipment. Fortune is from Rio de Janeiro, has her drone Glory and two pistols.



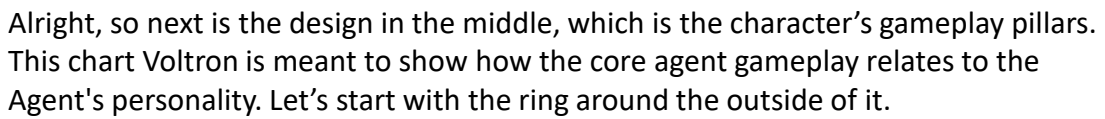
Next is Maslow's Hierarchy of Needs. This is used to quickly define the driving objective of the character and define the core motivation as to why they are fighting LEGION. The color of the level is then used in the rest of the sheet to make sure that core objective is carried throughout everything. Fortune is all about esteem, driven by her confidence and achievement. She doesn't worry about or feel the need to fight for things like family, security or basic physiological needs.



Below Maslow's Hierarchy Triangle is the Agent's favorite music. Fortune's music of choice is Reggaeton. Figuring out their favorite music became one of my favorite elements.



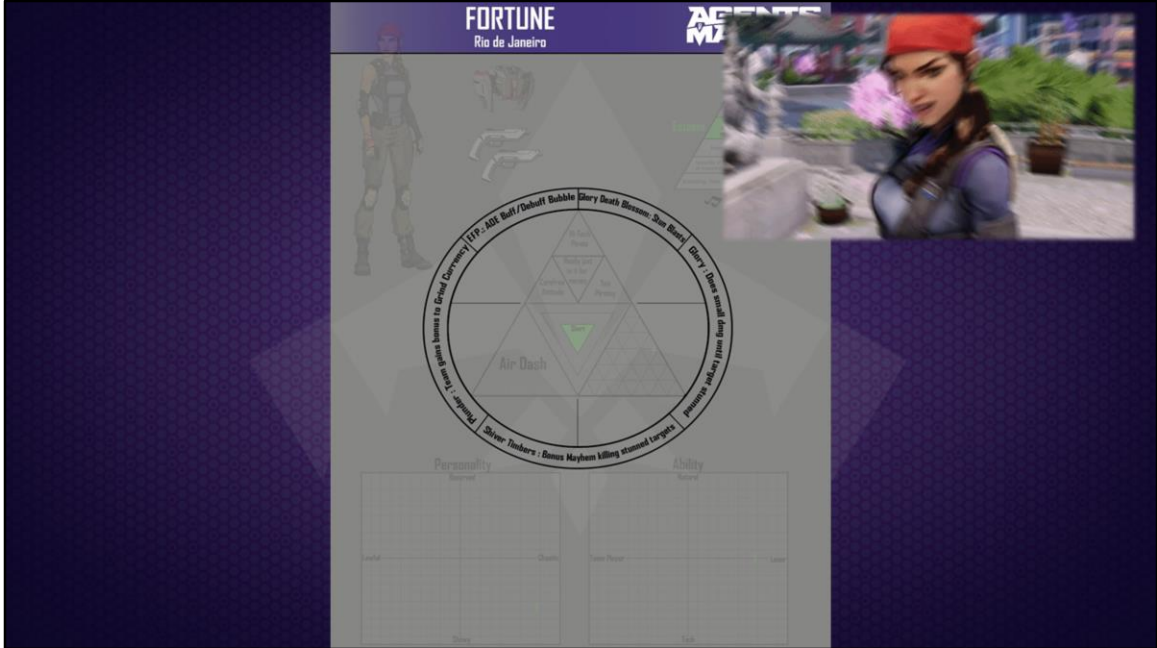
See, a big part of a performance is being able to maintain the emotional state of the character. For actors, maintaining that emotional state throughout a shot or action can be measured in minutes. For animators, it is hours. Or days. This is where knowing their favorite music becomes important. Listening to their favorite genre or artists allows you to stay in their head space for longer amounts of time. It can speak to and influence timing and personality choices. And unlock additional depth to their character as you search out new artists and becomes more immersed in the culture of the music. Even if it is a genre of music you dislike, as I initially did with Reggaeton, sticking with it can become an important part of the animation ritual when working on a character. For Fortune, if I'm not listening to Reggaeton, it just doesn't feel right.





This is the Ability Ring, broken into 5 sections. In the top left is the agent's *Special Ability*.

And in the top right section of the ring is the agent's *Mayhem Ability*.



This is meant to be used once every few battles. Fortune has her drone create a death blossom stun blast.

The bottom three sections of the ring are the Agent's passive abilities. While these often don't require animations, it does speak to their playstyle and personality, which is important for animators to keep in mind as those help to define the character's in game performance and playstyle.



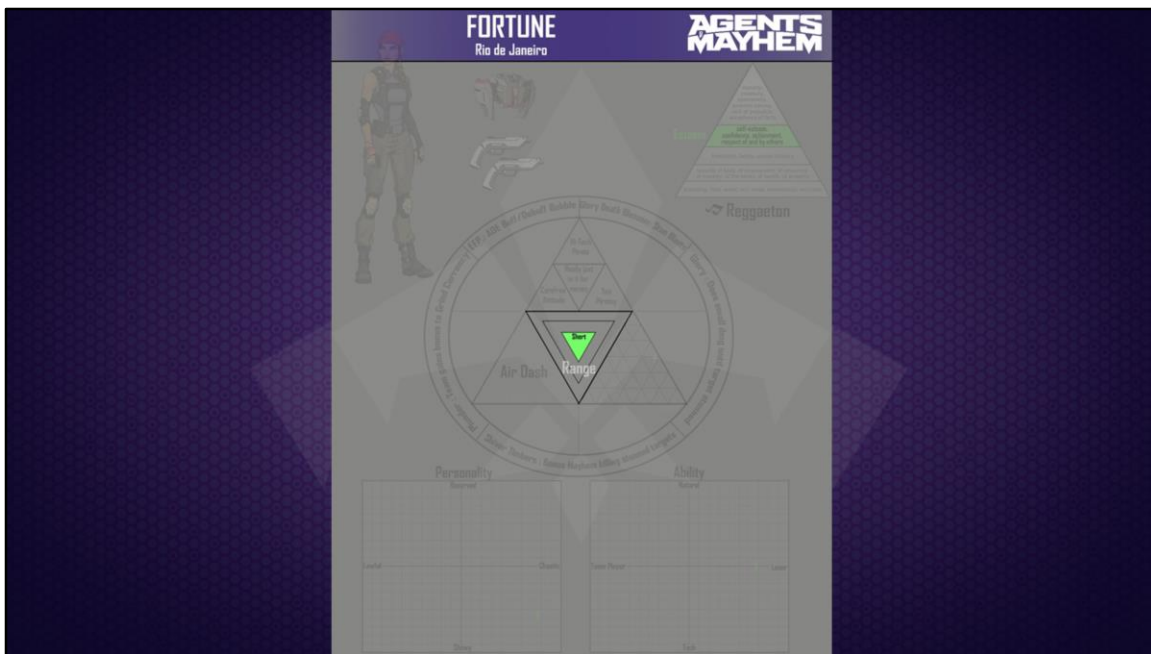
Now looking at the triangle in the middle of the page, at the top is the *Traitforce*, which itself is broken into 4 sections. Let's enhance it CSI style.



This is broken into 4 sections. The triangle at the top is the character's must have defining characteristic. This is their high level pitch. Fortune is a Hi-Tech Pirate. The triangle at the bottom left is a strength of the character, in personality or gameplay, that we want to make sure is often highlighted. Fortune has a carefree attitude. The triangle at the bottom right is potential pitfalls we can see the character falling into. These are areas we want to make sure we are always weary of, as it could make the character unenjoyable. We didn't want Fortune to be "TOO" pirate. No arrrrrrs and AHOY MATES! And the triangle in the middle is the character's secret. This is something we don't overtly state to the player or in the game, or that the character will tell any other, but something we use to help round out the character and give them depth. With Fortune, she is really just doing all of this for the money and tech.



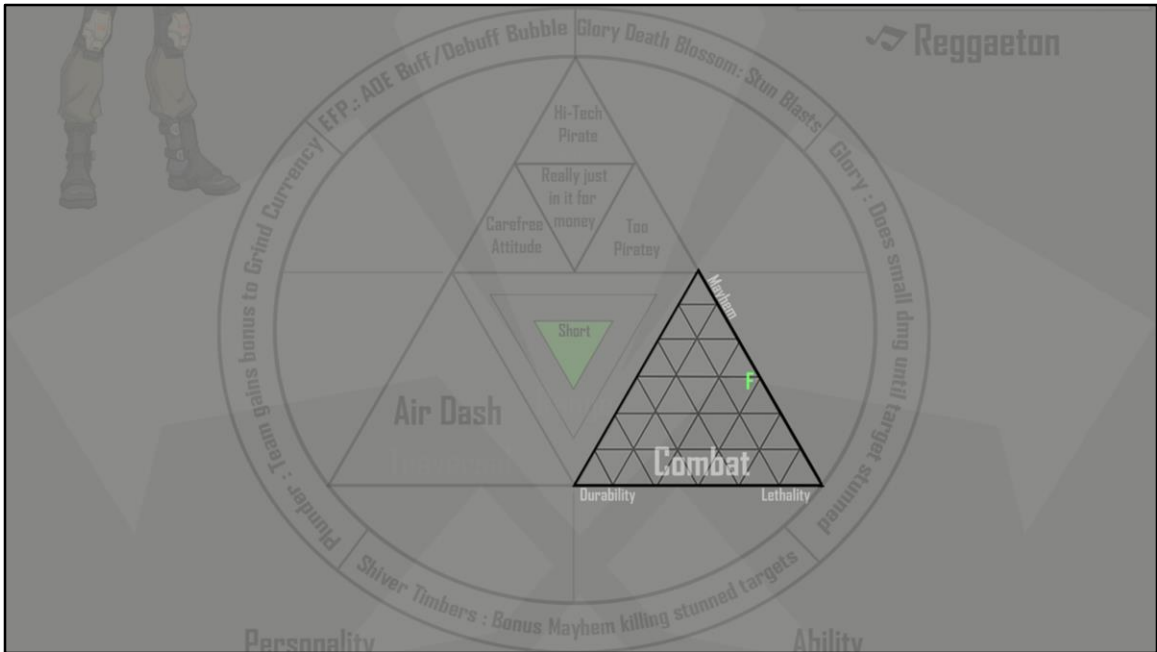
This triangle is the Agent's unique traversal type. Fortune can air dash.



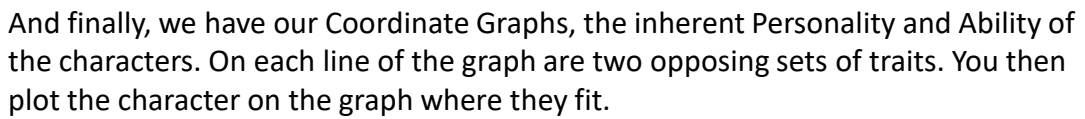
The Range Triangles in the middle signifies at what range the agent is best suited with their weapons and abilities. Fortune is a short range character.

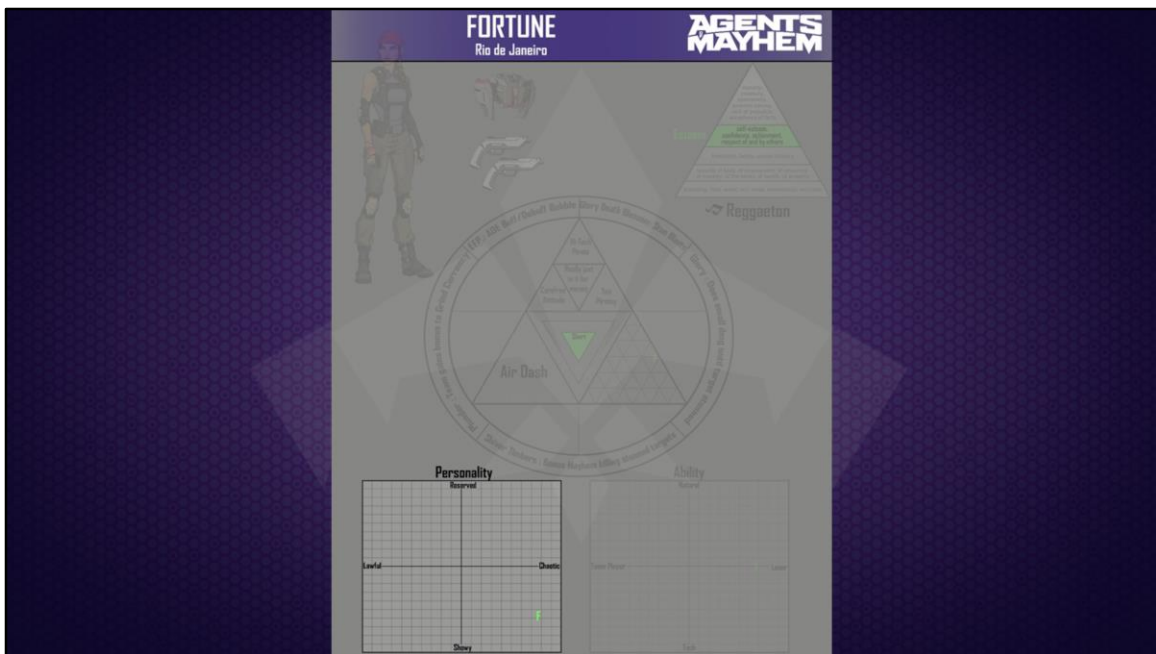


25

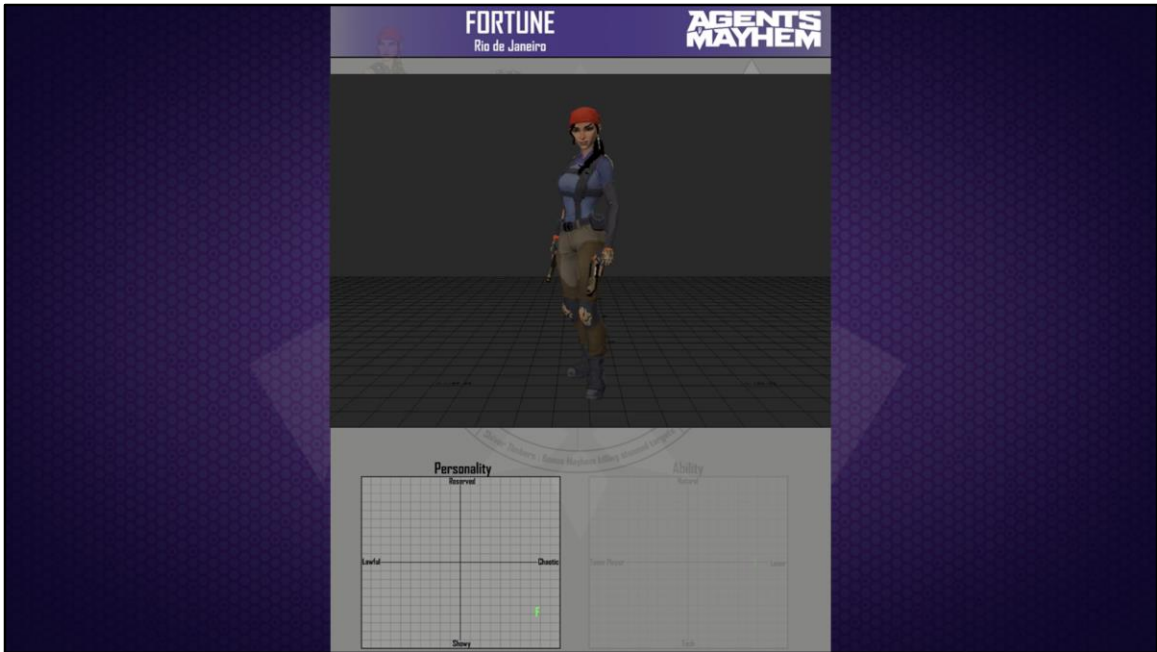


... within regards to Mayhem, Lethality, and Durability. Fortune is between Mayhem, which is how much they should use and rely on their abilities, and Lethality, how much damage they can output.



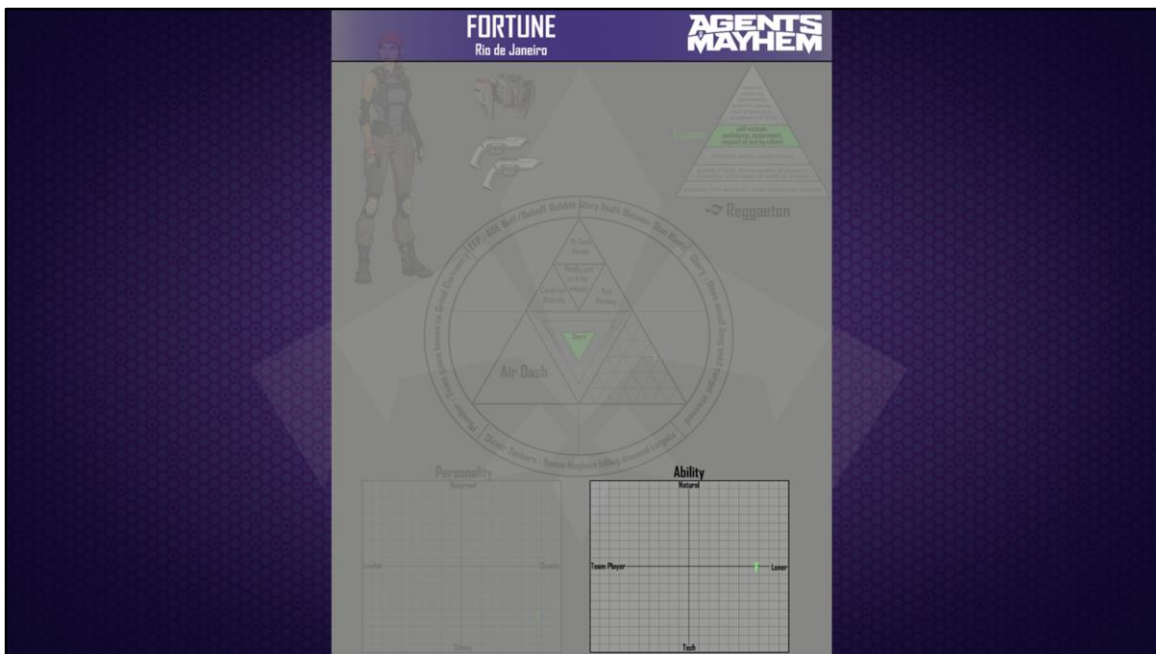


On the left is the **Personality Graph**, which from top to bottom has *Reserved* vs *Showy* : This describes how flamboyant an Agent is.

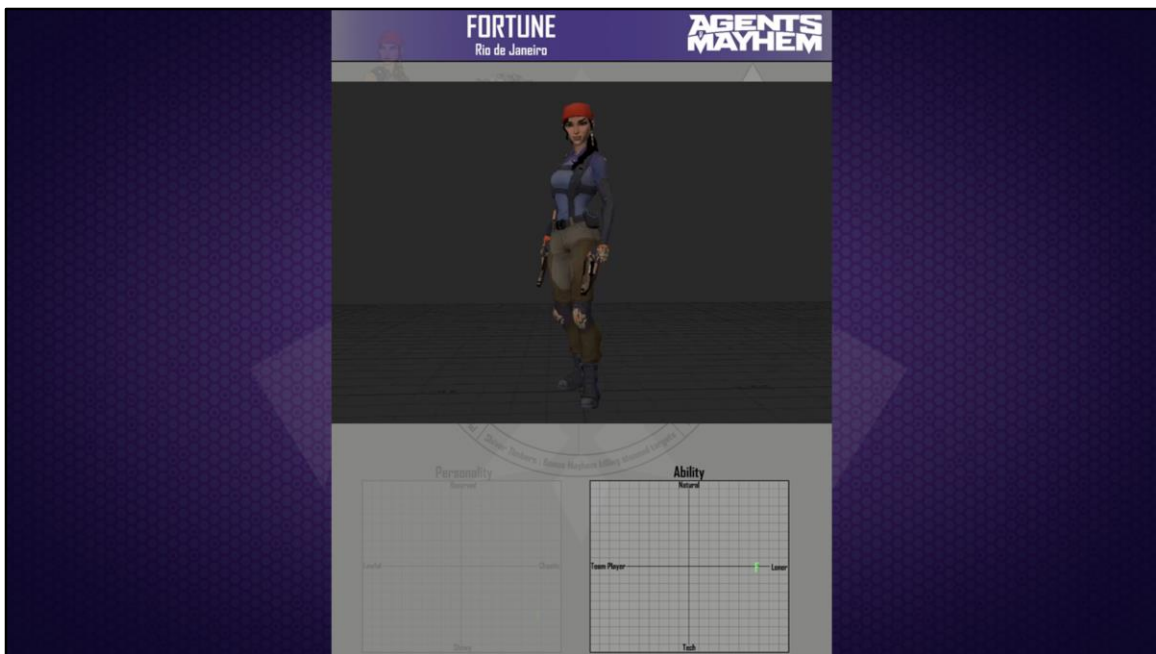


Are they more willing to be the strong silent type or are they super boisterous in everything they do? Fortune is pretty damn showy.

From left to right Lawful vs Chaotic :This describes how much an Agent follows a sense of order. Are they more honorable or are they more reckless? Fortune is rather chaotic.

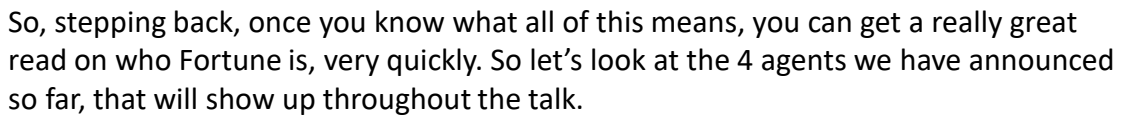


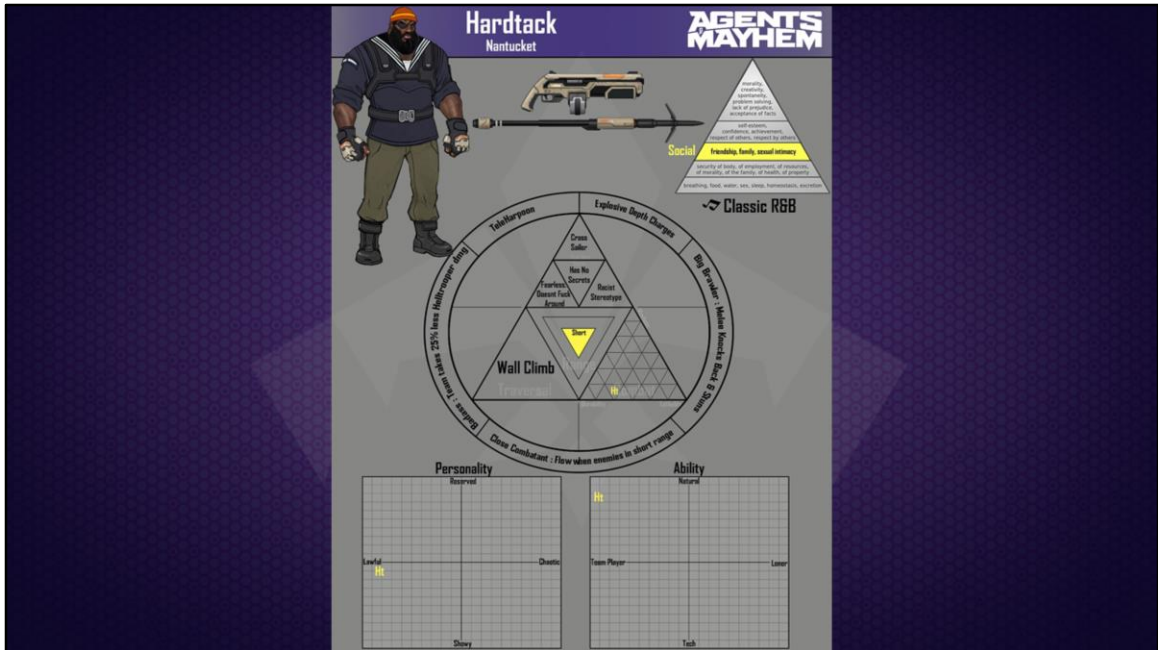
On the right is the **Ability Graph**, where top to bottom we have *Natural vs Tech* : This describes the Agent's inherent ability set. Are they naturally capable or do they rely primarily on their tech?



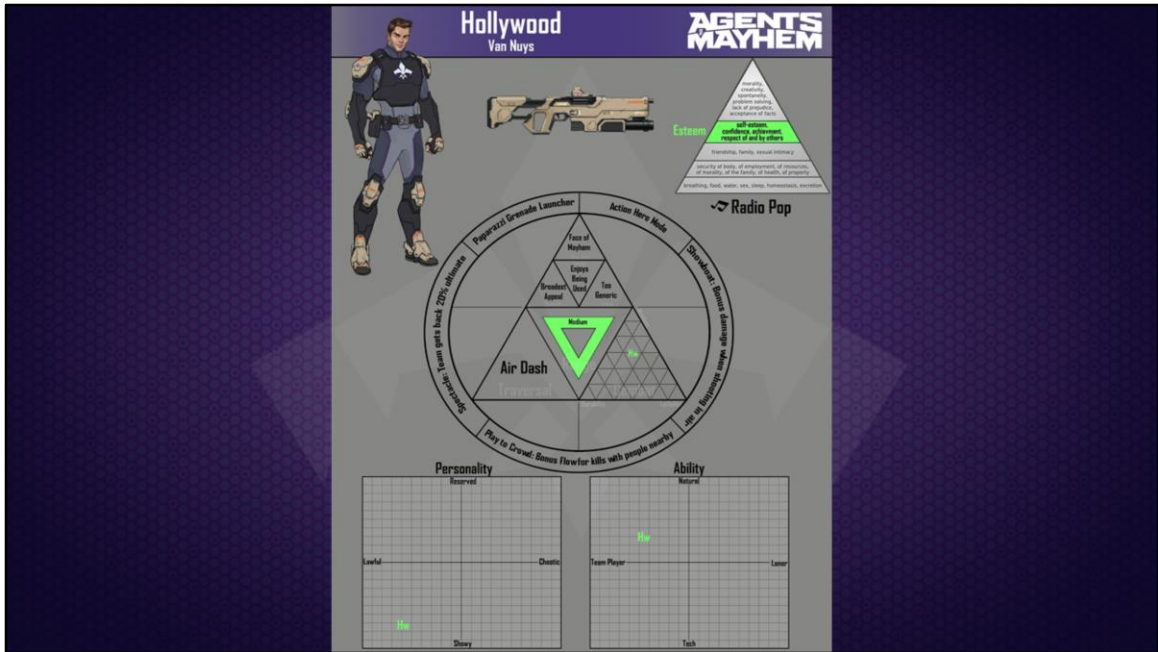
Fortune is right in the middle, meaning that she is as naturally capable and gifted, as she is as using tech.

Team Player vs Loner :This describes how well an Agent gets along with others. Are they the type that rallies around teammates or do they prefer to hang out on their own? Fortune is much more of a loner, reflecting both her placement on Maslow's Hierarchy and her secret.

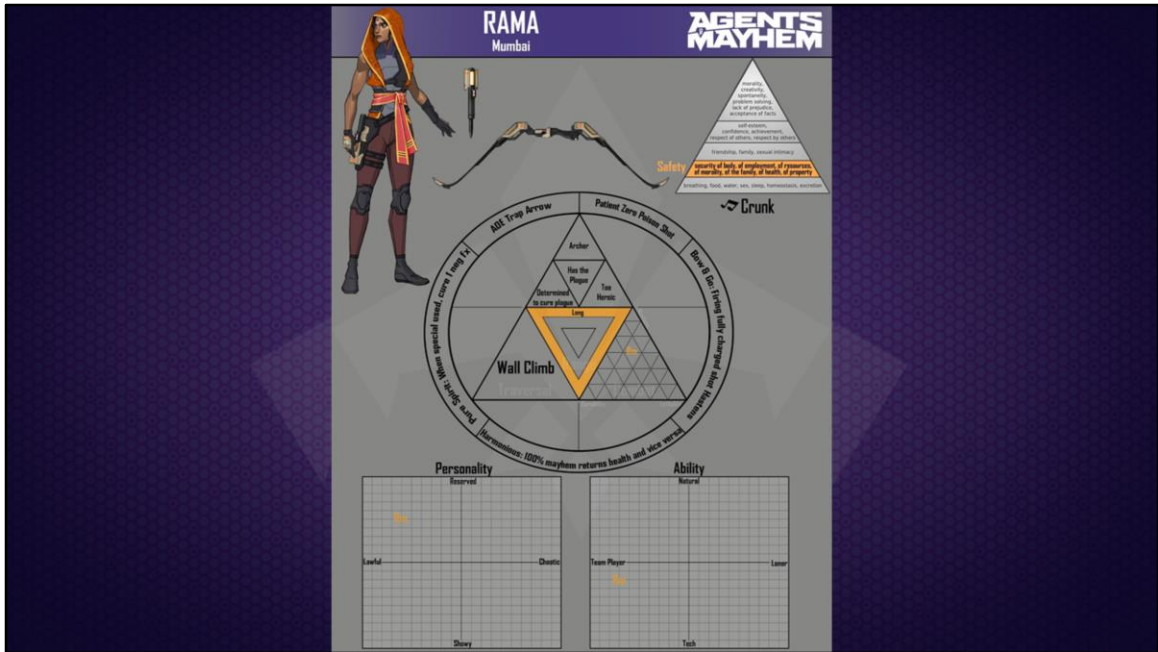




First up is Hardtack. We can see he is a social, crass sailor from America that doesn't fuck around. He loves Classic R&B and doesn't keep any secrets from anyone. He can wall climb, has a short range shotgun and is armed with a teleharpoon and explosive depth charges for his abilities. He is pretty lawful, only a little showy, and is a born natural along with being a team player.



Hollywood is from Van Nuys, California and is the Face of Mayhem. He has an automatic rifle that fires off grenades and lives for the glory and attention of others. He loves trashy radio pop music, is on the lawful side of showy, and is more of a born natural than tech person in his abilities and skills, which includes Air Dashing.



And finally, Rama uses her bow to fight and secure her homeland of India from the scourge of LEGION. She uses different types of arrows and enjoys crunk music. She is one of the most lawful and reserved members of MAYHEM, and by that nature makes her more of a team player, using her tech to help everyone she can.



With our characters more defined, the other core part of the tone we needed to communicate was the game's art style, which is defined on the project as Theatrical, Simplified and Vibrant. So our animation style needed to match that. And what that boiled down to for animation was

**AGENTS
MAYHEM**

ORDINARY VS EXTRAORDINARY

The Ordinary vs the Extraordinary

AGENTS MAYHEM

DEFINING STYLE

- Ordinary
 - Walking. Running. Standing. Turning.
 - Standard Mocap Polish

AGENTS MAYHEM

DEFINING STYLE

- Ordinary
 - Walking. Running. Standing. Turning.
 - Standard Mocap Polish
- Extraordinary
 - Jumping. Dashing. Abilities.
 - Exaggerated Style

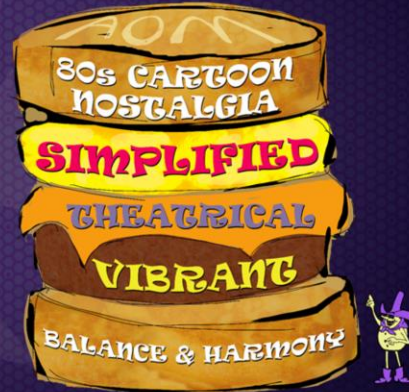
NON ARBITRARY STYLE CHOICES

What is key about this is that this approach is that it allows style to be used in a non-arbitrary way, attaching more pushed animations to clearly defined gameplay systems. Which is necessary, because stylized animations take more time than standard mocap polish, so we need to be smart in how and when we choose to embellish. That intentionality is what helps to create a unified experience and quality bar.

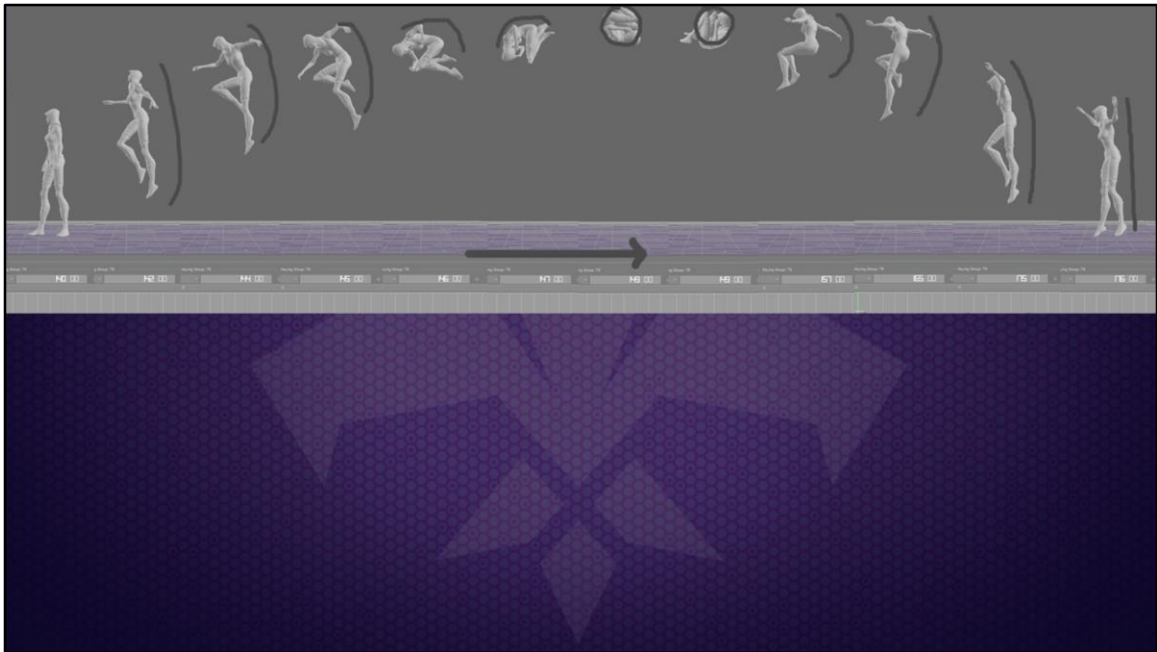
AGENTS MAYHEM

DEFINING STYLE

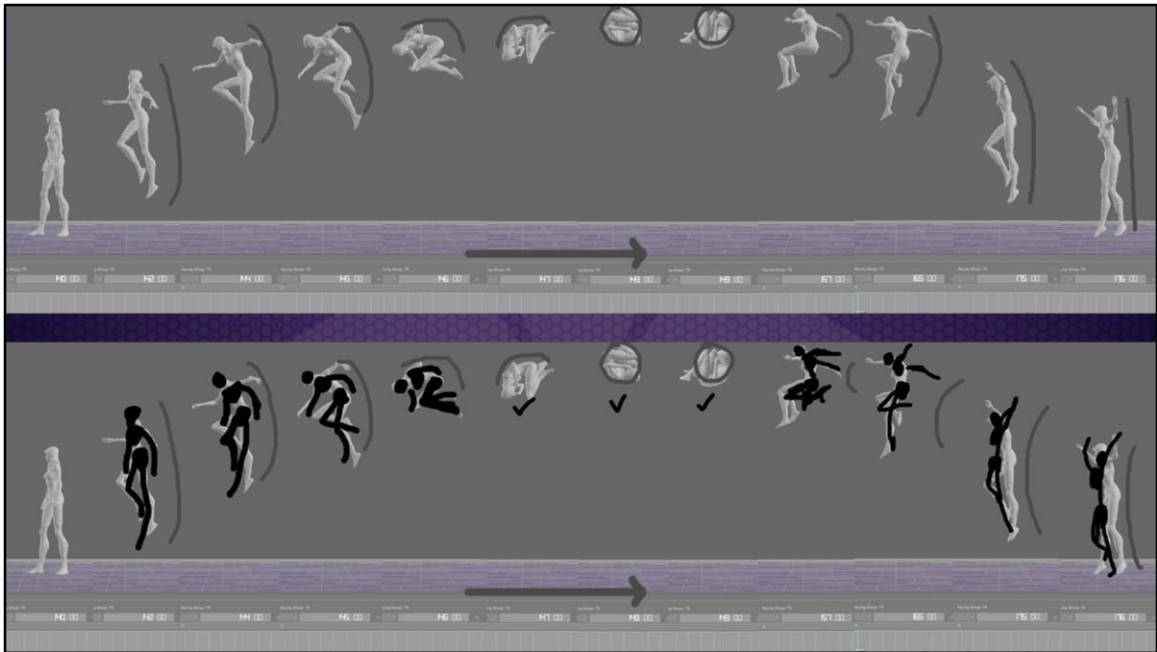
- Extraordinary Style Pillars
 - Simplified



With that in mind, our first style pillar is simplified. Which we interpret as a limited 2D style, matching our overall aesthetic. The timing of extraordinary movements will be snappy, faster through the motion with longer held poses conveying the key moments of the movement or ability. This means super simple lines of action during quick movements. Think of them as smear poses and less as storytelling poses.



When I say smear frames as a reference, we aren't using literal smear frames or shader tricks. We are thinking about the poses like smear frames, where we want to reduce the noise and complexity of the pose, during those snappy transitions. This front flip would be animated in a more traditional style. On the way up, the arms and legs are offset as they rise, leading into the spin, and then unrolling naturally as she descends.

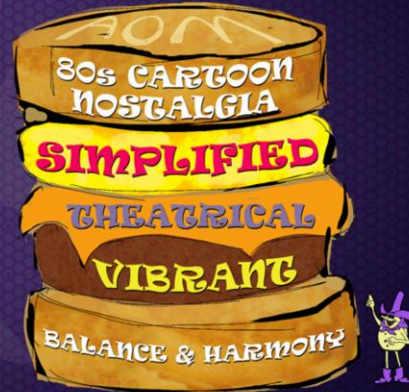


But when thinking of the poses in a simpler, smear manner, on the way up, the body has the limbs brought in, creating a cleaner, singular line of action, removing complexity and noise. Same getting into the ball at the apex. Then, coming out the ball, the character should snap out of it, in a more iconic, “in air” pose. When they being to stretch into the fall, again the character becomes a clean, singular line of action. The essence and intent of smears, without having smears.

AGENTS MAYHEM

DEFINING STYLE

- Extraordinary Style Pillars
 - Simplified
 - Theatrical



In regards to theatrical, by attaching our extraordinary actions to gameplay systems, the exaggeration of abilities and personality moments will help to draw focus towards the theatricality of the character.

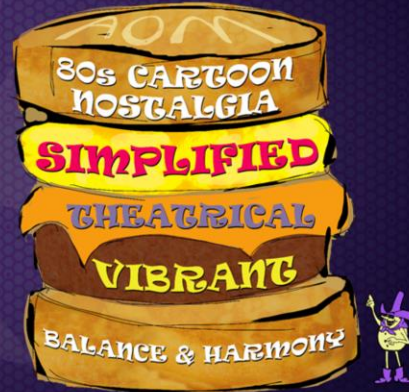


With Fortune's special ability, how her guns create such a large ball of energy isn't as important as making sure the action fits her personality. Wildly launching it forward while jumping with a double bounce land makes it FEEL like it makes sense.

AGENTS MAYHEM

DEFINING STYLE

- Extraordinary Style Pillars
 - Simplified
 - Theatrical
 - Vibrant



Which leads to our Vibrant pillar. This means that the personality of a pose is just as important as the functionality of the pose. Instead of animating a traditional jump, which hits the proper mechanical poses along the way, think of the characters more like Spider-Man.

AGENTS MAYHEM

DEFINING STYLE

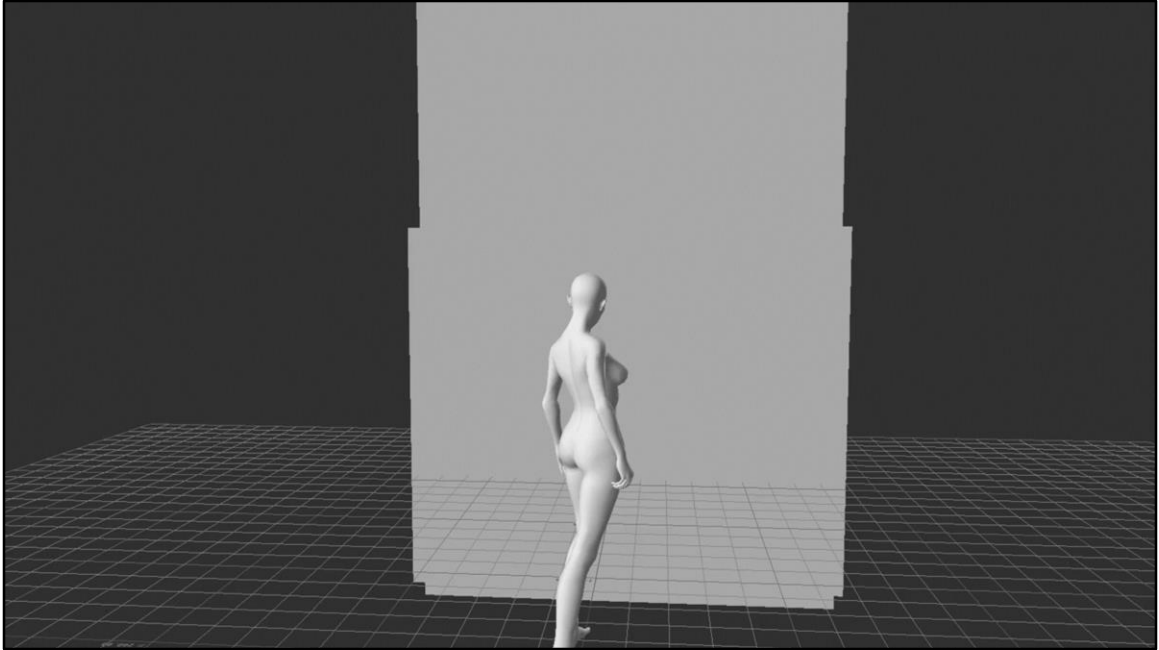
- Extraordinary Style Pillars
 - Simplified
 - Theatrical
 - Vibrant



His poses when jumping through the air say less about the mechanics of the jump, and more about that character being in the air. By clearly defining ordinary vs extraordinary we will strike both the bold and balanced demonstration at the heart of the pillar.



With that in mind, Jumping in Agents of Mayhem is the definition of our Extraordinary style and what we used to establish it early on.



This is the jump test I did early on to show how jumping in our game could look and feel a bit different than a traditional jump. Fortune was the first character I was going to work on, and this model was one of our early test female meshes. We knew we wanted multiple jumps and air dashing, and this was an attempt to show how this could look. When everyone on the team responded well to it, breaking it down became important, to make sure everyone on the animation team could match the style, and design would know what the timing and elements were that could be tweaked.



So let's break down the first jump, which is in fact Fortune's base jump in the game. Notice the simple line of action on the first stretched pose. It is also worthwhile to note she has no anticipation on her jump.



As she begins to blend into the top of the arc, she begins to hit her personality pose. Her crossed legs aren't a mechanically true pose for a jump, but they speak to her more fun, freewheeling nature. Remember, we want to always be asking, how does THIS character look when in the air?



As she squashes at the top of her jump, the core of the air pose is held, and it is the follow through of her limbs that keeps the momentum alive.



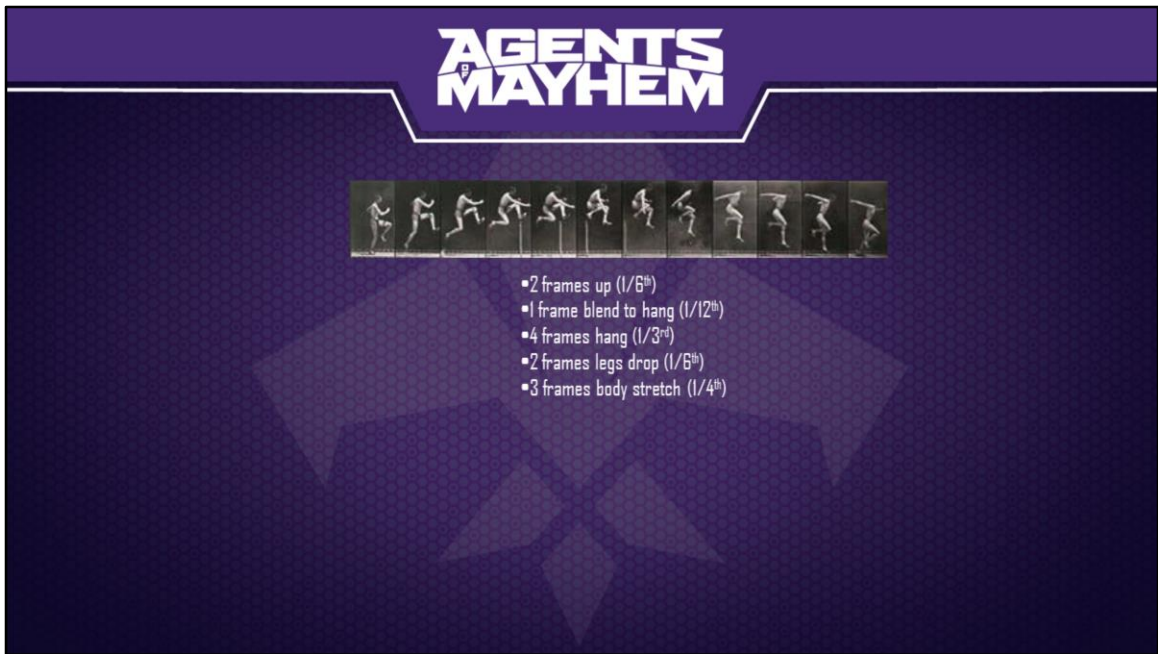
As Fortune begins to release into their descent, the legs drop first. This gives a sensation of gravity pulling them down, but still allows for the feel of the hang time to be believable, because of the character's core strength to stay in the air. Notice also that the arms maintain a similar world space, even as they begin to rotate up, to further sell the hang time.



Going into the proper fall pose, again we have reduced the body to as clean a line of action as possible. The arms are kept bent to keep some semblance of control/strength within the character, but also to alleviate the blend to land. If the arms were entirely outstretched above the head, they could noticeably interpolate/flash as they play their land/settle animation. With the poses defined...

SPACE AND TIME

Examining the spacing and timing is the second part of breaking down the motion.



For reference, here is the timing breakdown of the 12 frames in a Muybridge jump. This shows how many frames make up each part of the action across rising, hanging in the air, and then dropping.

AGENTS MAYHEM



- 2 frames up ($1/6^{\text{th}}$)
- 1 frame blend to hang ($1/12^{\text{th}}$)
- 4 frames hang ($1/3^{\text{rd}}$)
- 2 frames legs drop ($1/6^{\text{th}}$)
- 3 frames body stretch ($1/4^{\text{th}}$)

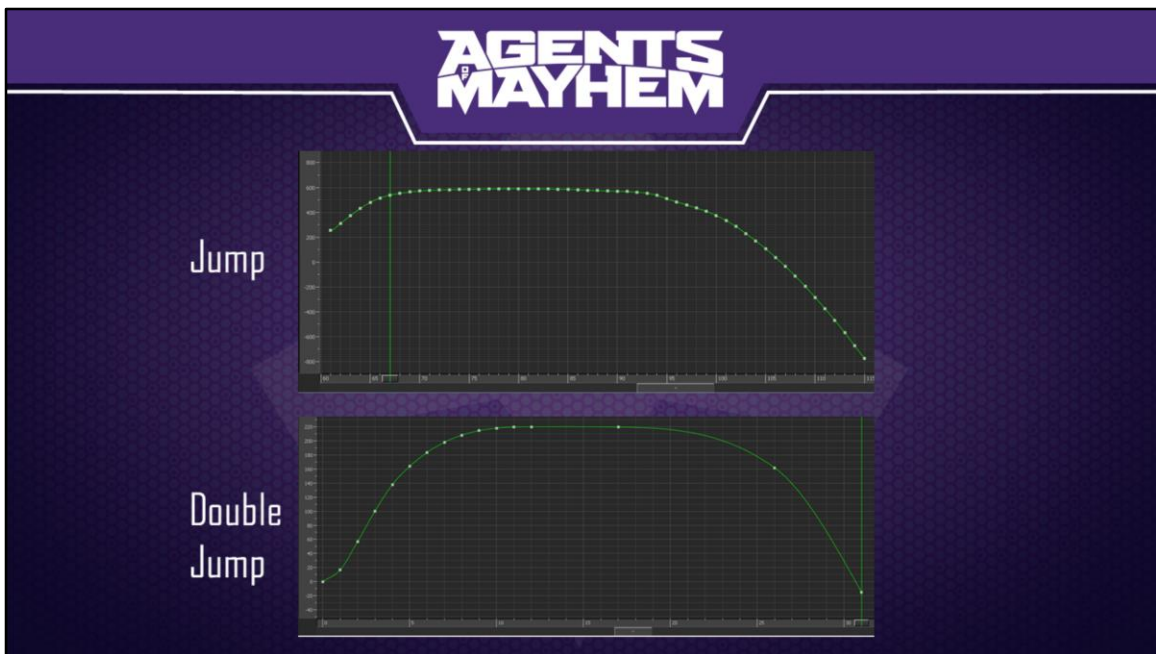


- 4 frames up ($1/6^{\text{th}}$)
- 3 frames blend to hang ($1/8^{\text{th}}$)
- 13 frames hang ($1/2$)
- 3 frames legs drop ($1/8^{\text{th}}$)
- 4 frames body stretch ($1/6^{\text{th}}$)

And here is the 27 frames of Fortune's jump for comparison. You can easily compare her spacing and timing in relation to a more standard jump.



Meaning that instead of a standard parabolic curve, we wanted more of a plateau. Faster up, longer hangtime. This timing helps to sell the style of having an in air pose instead of a standard jump. Which is key, because the benefit of bringing jumps into this style is that it allows us to use the same jump animation for standing in place vs running, because mechanical function isn't what is important. Feel and personality is, which this approach accomplishes.



And after implementing and tweaking, this is what we have in game. The top one is curve of Fortune's Base Jump. And the bottom is her double and triple jump. The rise and hang remained pretty consistent with the first test. The drop off was eased, as having such a harsh drop made gauging jump distances more punishing than we wanted.



Which then looks like this in game. Here is Fortune's triple jump chain. And because triple jumps are considered extraordinary, it means that with each jump we can push the pose and personality further with each part of the chain, like you might do with an attach combo in a fighting game.



And how far we should push that chain goes back to how showy the character is on their one sheet. Fortune has a much more showy personality so it is totally within her character to do flips, spins and all manners of flourish in her movement and poses.



Alright. *Next*, we knew we would be using mocap as a base for much of the character's animation suite as part of their ordinary actions. But we also wanted to use it as a base for some of the extraordinary actions, like melee attacks and special abilities. Let's break down how we took Fortune's melee attack from a mocap action into a stylized action.



Let's first start with the raw mocap. Yep, that looks like mocap of solid spin kick by our actress Maggie Macdonald.



And here is how we would clean it up if it was an ordinary move.

- It was sped up about twice the speed of the raw mocap, but not as a universal scale.
- Attention was paid to proper spacing and timing to sell the believability of the motion while still hitting design needs.
- It was also made to start and stop without moving off of origin.



And here it is with a style pass. Adjusting the timing has shaved off another 7 frames from the motion. But much of the style comes in the posing and timing before the point of contact, so let's break that down.



We are starting the character in the stretch pose, like the jump.



The height of the in air motion has been lifted to it give it some more hangtime and anticipation. The leg that will be kicking starts dragging immediately.



The body has largely hit the held, in air pose, as the leg continues to drag, gleefully breaking the joints of the leg.



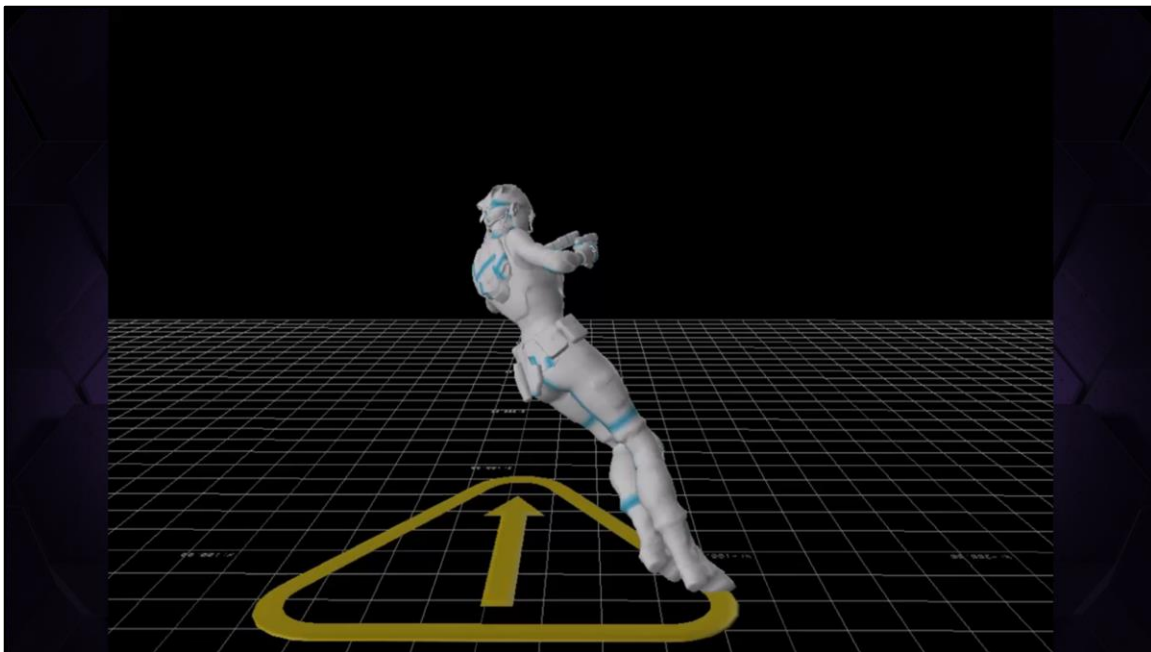
The body remains in the held pose as the leg starts to whip around.



Watching it again, you can feel the extra attention spent on the anticipation as it feels more snappy and exciting. At this point in the style conversation, defining how far we could push it beyond timing and posing started to happen.



And no conversation about animation style can go long before squash and stretch is brought up. So this was something we needed to define the use of. Using motionbuilder, and not custom rigs, we were limited to how much we could do. And with mocap a large part of our movement suite, it couldn't be too over the top. So, like our overall style, it needed to be purposeful and non arbitrary in its use.



Here is Fortune's Dash Left. There is some squash and stretch built into her spine but it is meant more to carry the energy of the motion through the spine than to make it SUPER snappy or over the top. Feel it more than see it. It isn't meant to be used on everything brought into style, only when needed to keep the pose or energy alive. Posing and timing are still at the core, this just serves to support them.



Now that we knew more about the characters and the style, we could start to define the quality of the animation overall, to help find the answer of how many characters we can support.

AGENTS MAYHEM

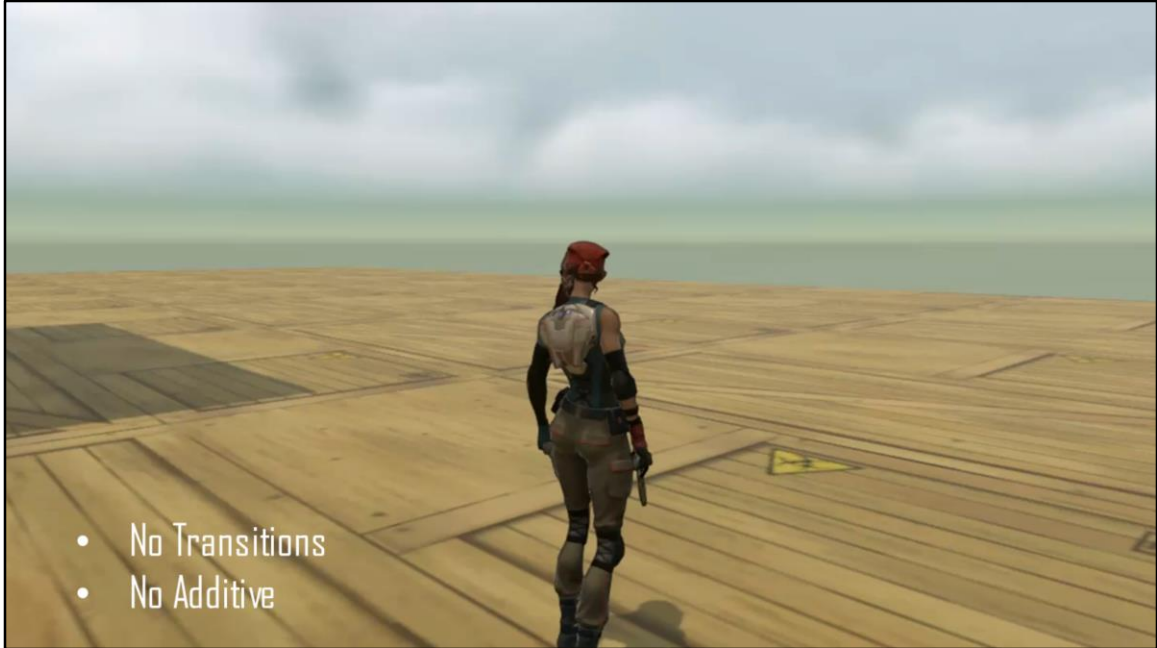
DEFINING QUALITY

- Out of Combat > In Combat
- On Ground Movement

To do this we started with two rules.

First, we agreed that out of combat needed higher quality visuals vs in combat. This was because out of combat the player focuses more on the player character. And in combat, the player is more focused on what is happening in front of their reticle, with the player character becoming more of a supporting element.

Second, we wanted to start with on ground movement, since that is something common in every game and has a lot of expectations in regards to fluidity and responsiveness



So we started with the bare minimum of just a jog and blends, to see how terrible that would look. And of course, it does. But it allowed us to identify the exact areas we wanted to target.



Which lead us to 4 Directional Jog Starts. 2 180 Jog Turns. Walk, Jog and Sprint Stops. Additive Leans. Additive Head Looks. Additive Torso Looks. Establishing this standard allowed us to define gameplay metrics & markup for universal responsiveness, to ensure turn rates at different parts of the animation could be leveraged by all characters. So, for jog starts, every character needs to have turned by frame 10-12 and be at jog speed by frame 11-13. No matter the character's size, they need to match this metric.



Which means selling weight and speed is entirely aesthetic, like with Hardtack here. After animating Fortune, one of our smallest and most agile characters, Hardtack was my second character, which was perfect, because he is one of our largest, and allowed us to prove out that the metrics we defined with Fortune could still work for characters of any size. And what has become one of my favorite things is that during internal and external playtests, we always get feedback that Fortune is called out as faster and more agile than Hardtack, even though under the hood, they are 100% at the same metrics.



Another example of meeting metrics but selling the weight through animation is jumping. Unlike Fortune's base jump, which has no squash at the beginning, for larger characters, we do have up to 3 frames of squash, which is the duration of the blend into the jump, to help them feel a little bit heavier. They still hit the same height metric, and have the same overall time in air, but those 3 frames at the front help to elicit the same feedback from playtests of Fortune feeling more agile and capable in movement, even though both can traverse the world with equal skill.

The logo for 'AGENTS OF MAYHEM' is displayed in a stylized, white, blocky font. The word 'AGENTS' is positioned above 'OF MAYHEM'. The 'O' in 'OF' is a small circle. The background of the logo is a dark purple banner with a white outline. The entire banner is set against a dark blue background with a subtle pattern of light blue geometric shapes.

**AGENTS
OF
MAYHEM**

SOLVE IT WITH SYSTEMS

Which sort of leads me to one of my favorite mindsets of our team. “Solve it with universal systems” With multiple player characters, the number of unique assets we could support per suite was limited. So the more common AAA approach of throwing lots of animations into a unique suite for each system wasn’t going to be possible. And the first big test of that for us, was...

STAND TURNS

Out of Combat Stand Turns. One of the deepest, darkest holes you can try and fill with animations forever. Slow turns, Fast turns. Upperbody vs Lowerbody. Keeping the feet locked down vs slight translation. We knew there was no way we could support a complex turning system for our characters. So, we didn't. We in fact, didn't make any animations for it at all.



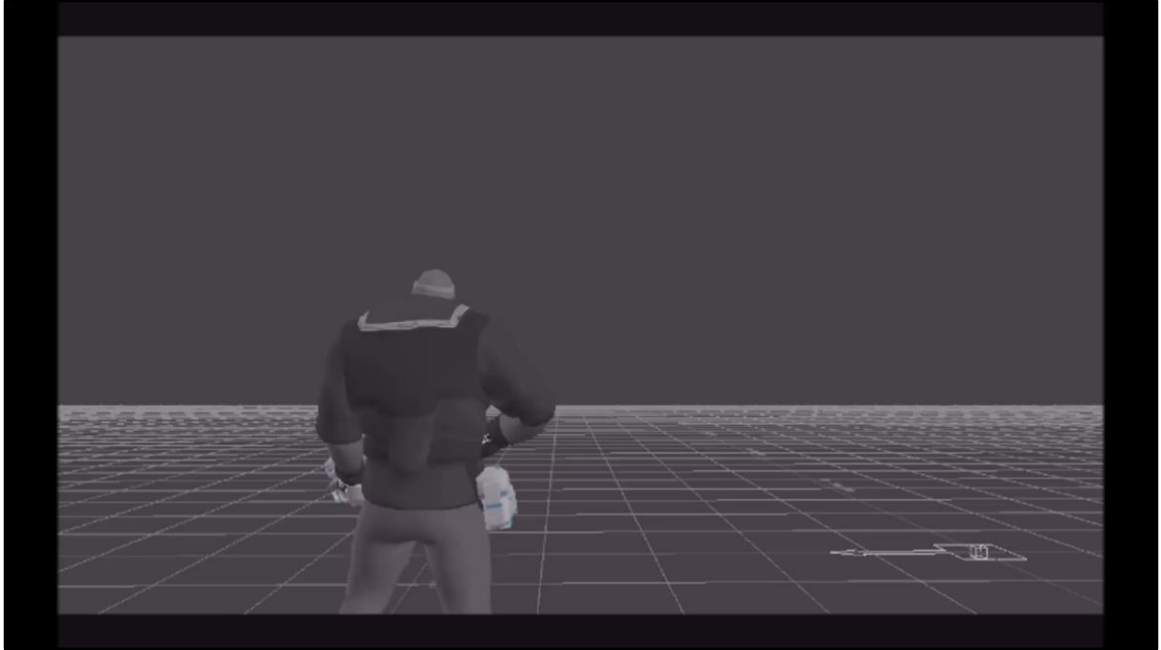
We used walk stops. And we let the system handle the rest. When you deflect the stick, you are getting a couple frames of the additive head and torso look, as well as the jog start. So the character has at least SOME facing in their body. And if we author the walk stop to have some additional foot shuffles back into place, it sells the momentum of the turn. Sure, it isn't as pretty as a turn system, but it is free and responsive. And as a general rule, if there are a couple of ugly frames on input, as long as the follow through looks good, people are okay with it.

VEHICLE ENTRIES

Vehicle entries can be another huge investment in animations, once you take into account multiple vehicle types needing entries from different angles and different versions if they already have drivers or passengers inside. What can be a giant suite and investment for even one character, across multiple, would be scope crushing. So when realistic functionality becomes too much, it is time to lean into the style and tone of the game.



And for this task, we had two animators, Josh Barends and Nick Neibling, light the way.



Teleporting and swapping characters is a core part of the gameplay and tech Mayhem uses. So using that as a cheat for vehicle entries made sense. Josh prototyped and implemented it. And then with addition of giving Hollywood finger guns as he cheeses to the camera, Nick added that extra bit of personality that made the feature fully embrace the style and tone of the game. At which point these have now become some of the more signature animations for characters, as we continue to try and one up how they play to the camera.

NEAR VS WIDE TURN SPEEDS

The other place we couldn't afford to add additional assets was turn fidelity while running. Designers and world builders wanted more narrow areas to move and the controls needed to feel tight and responsive in those situations, but still look good when you are running down an open street. Instead of giving two entirely different suites for each situation, or additional transitions based on different stick deflections or speeds...

AGENTS MAYHEM

NEAR VS WIDE TURN SPEEDS



one of our senior animation programmers and professional chicken wrangler, Isaac Vanier, took an approach similar to out of combat vs in combat.

AGENTS MAYHEM

NEAR VS WIDE

- Near Mode Goals
 - Focus on Character Position
 - More Responsive Controls
- Wide Mode Goals
 - Focus on the Character
 - More Fluid Movement

When you want responsive controls near ledges or doors, you are more likely looking at where you want to position the character, and not the character itself. But when you are in a wide open space, you want the character to look and feel more fluid. So Isaac's solution was to toggle a near vs wide mode, that adjusted the turn speeds in realtime, based on specific factors.

AGENTS MAYHEM

NEAR VS WIDE

- Near Mode Triggers
 - Near A Wall
 - Large Stick Deflections
 - Jog Starts



If there was a wall nearby, the character goes into near mode, because you may want to turn a corner or turn into a door. If you have a large thumbstick deflection, it turns on. When you start moving, it turns it on. And it stays on for about 1-2 seconds, in each instance, at which point it then switches back to Wide mode. Adding this meant moving down narrow corridors, making tight turns, and small adjustments to interact with an object were no longer an issue. And since the turn speeds governing this were controlled by switching between different blend trees in the animation graph, it meant we could also adjust our additive head and torso looks between each mode.

AGENTS MAYHEM

NEAR VS WIDE

- Wide Mode Head/Torso Looks
 - Head Turns First
 - Torso Turns Second
 - Body Turns Third



In wide mode, we have the body turn at a slower rate to look more fluid, so we lead with the head and then torso. This gives visual responsiveness to player input and an order of operation to the turn/twist, so it doesn't look like turning a character on a stick.



Essentially, it is what we established in the original quality video. Notice the order of operations on the head, torso and body as she turns while running.

AGENTS MAYHEM

NEAR VS WIDE

- Near Mode Head/Torso Looks
 - Head Turns First
 - Body Turns Second
 - Torso Third



In near mode, the body turns at a faster pace to allow for positional responsiveness. We also wanted the head to turn fast, to make the character look aware of their quick directional heading. But then we have the torso counter rotate, to simulate some drag and again break up the order of operations between the parts of the body. At a certain point, if running in a circle, we let the torso catch back up, so the head and torso are leading the body.



This obviously looks a lot twitchier, and not as fluid or appealing. But again, this is only triggered when the player cares about that twitch movement, and fluidity and weight of the motion can get in the way. But notice as she runs in the circle that the torso drags behind the head and body. Being able to break that order of operations is key in making this look in any way convincing or compelling.



Another big animation asset sink is combat aim movement. Even though we decided not to focus as much on combat movement as much as out of combat, one area we wanted to address was legs crashing through each other as they cross from front and back moving angles. But the traditional 8 animations per speed for this would even add up across multiple characters.

**AGENTS
MAYHEM**

COMBAT MOVEMENT

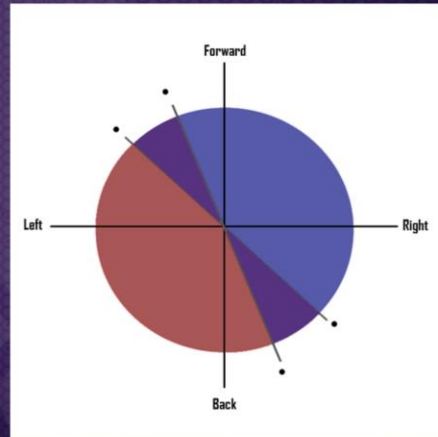


Enter the mad genius of senior animator Clay Matthews.

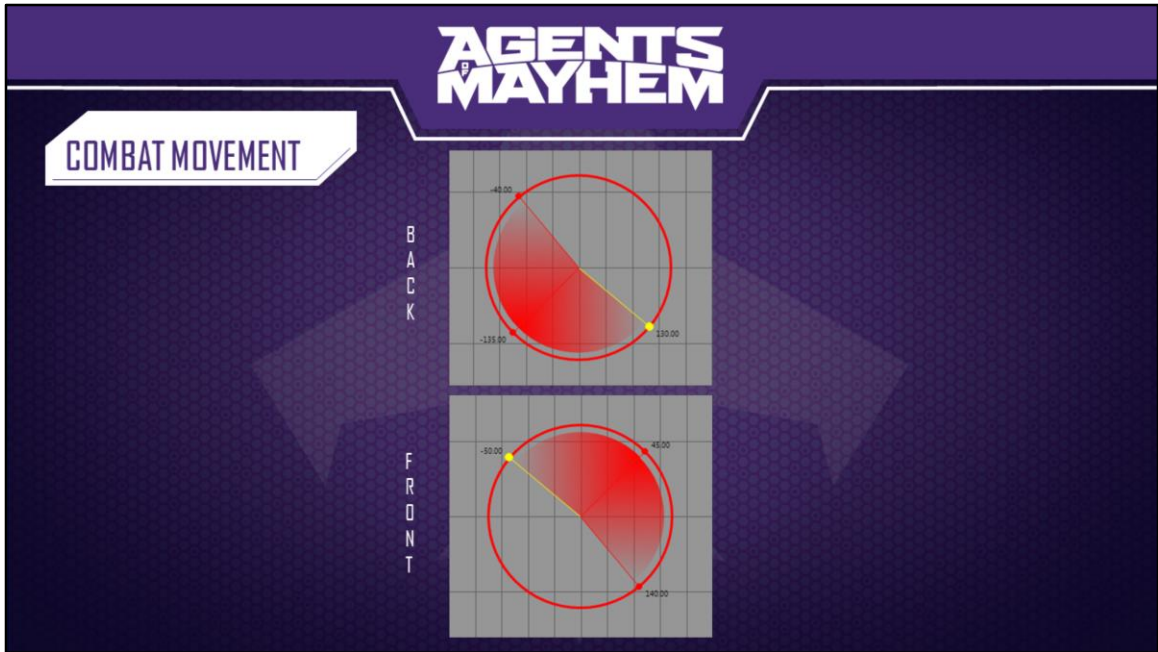
AGENTS MAYHEM

COMBAT MOVEMENT

- Standard Method
 - 4 cardinals
 - 4 diagonals
- Mathews Method
 - 6 diagonals



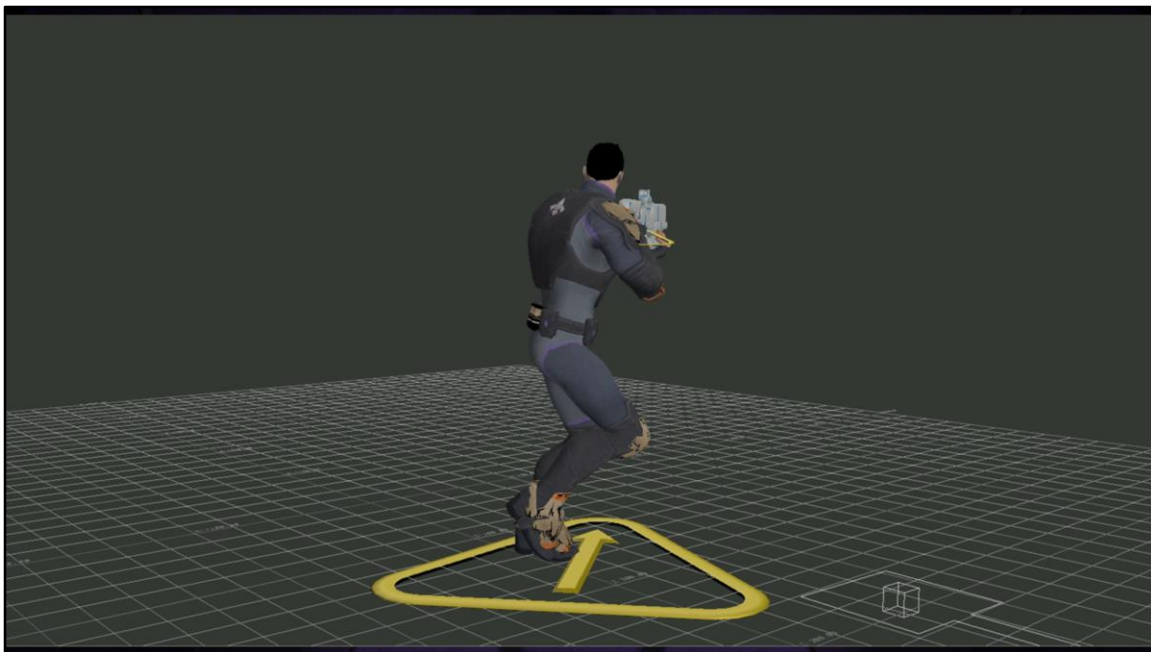
The previous approach used at the studio was to capture the 4 cardinal directions, and then some diagonals, as shown here, to cover the transition of the character changing from running forward to peddling backward. You put them all in a single blend tree, and blend them all smoothly as the thumbstick rolls around. Then in game, IK aim orientation with the upperbody is used to keep the character aiming at the reticle. And while this works, it still isn't an ideal method for quality or scope, and not as competitive as other approaches that use even more assets. With the Mathews Method, we use only 6 animations, which are all diagonals. And requires IK aiming.



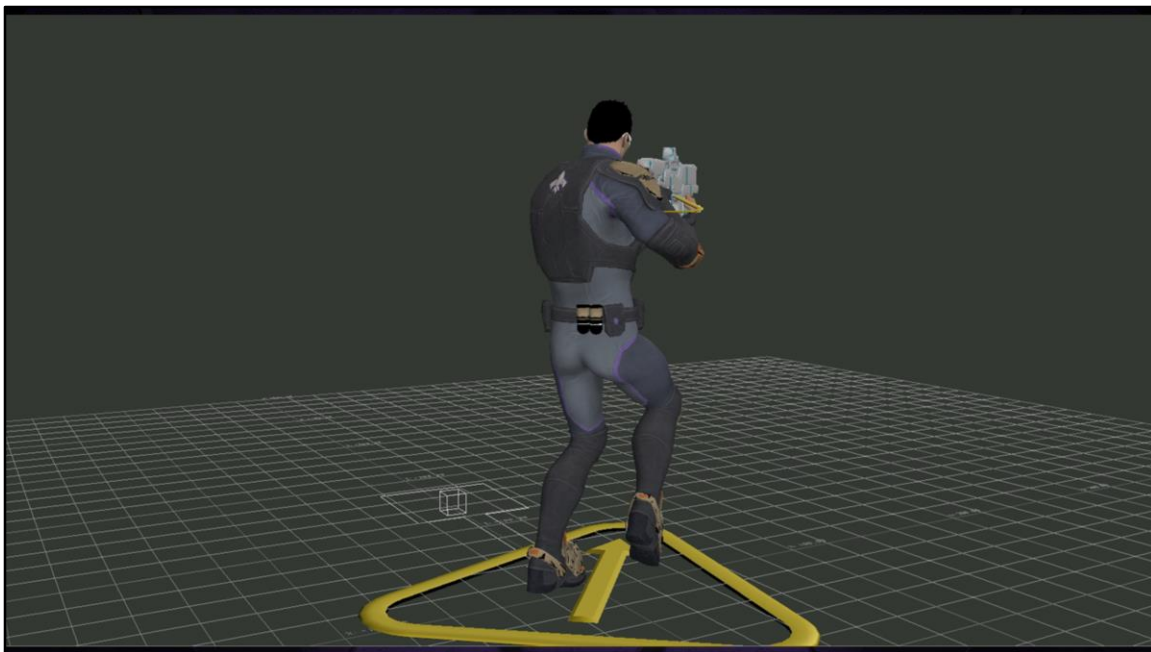
Instead of capturing the 4 cardinal directions, we capture back left and front right as single assets, and then the crossover angles of front left and back right. The 3 front moving animations are put in one blend tree, and the 3 back in another. And then a state machine sits above these, switching between the two as driven by stick deflection. There is a 10 degree overlap on those angles that switch between forward and back, to keep from having an ugly bounce if someone is hanging out in one of those regions. So let's see what these animations look like, before showing how it looks in game.



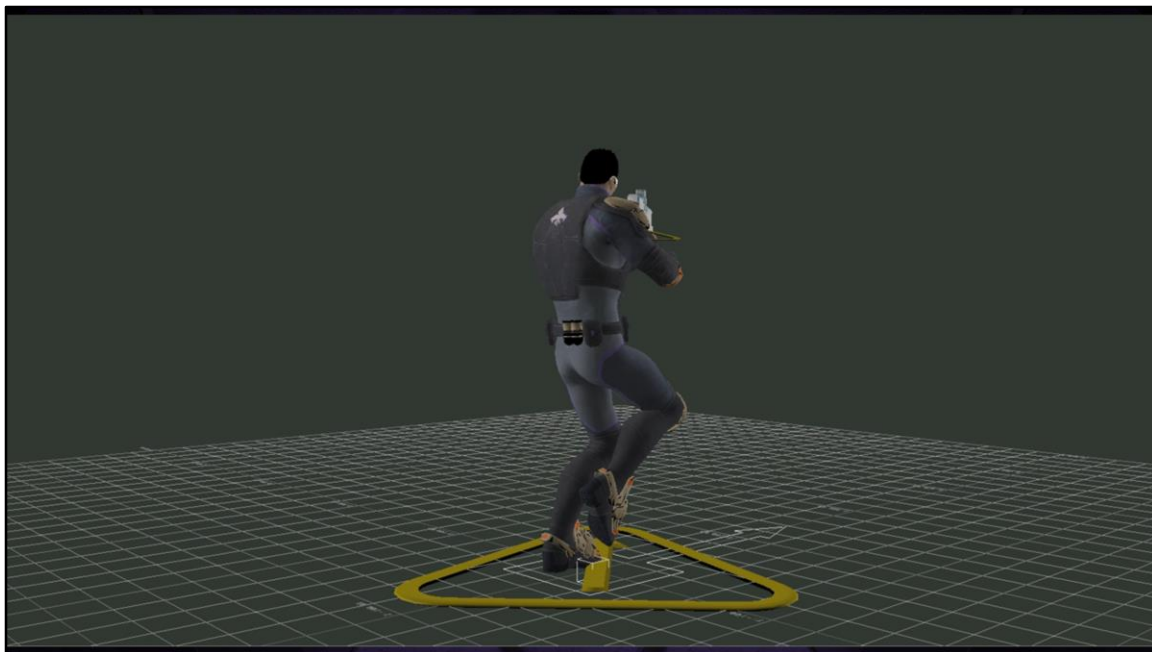
This is the “forward” animation, which is a 45 degree forward right animation. The gait is a pretty straight forward jog and the arrow at his feet shows where the camera relative direction is.



And here is “forward right”, which is 135 degrees to the right. More than being a forward animation, this is a sidestep animation, where the left leg crosses in front of the right leg.



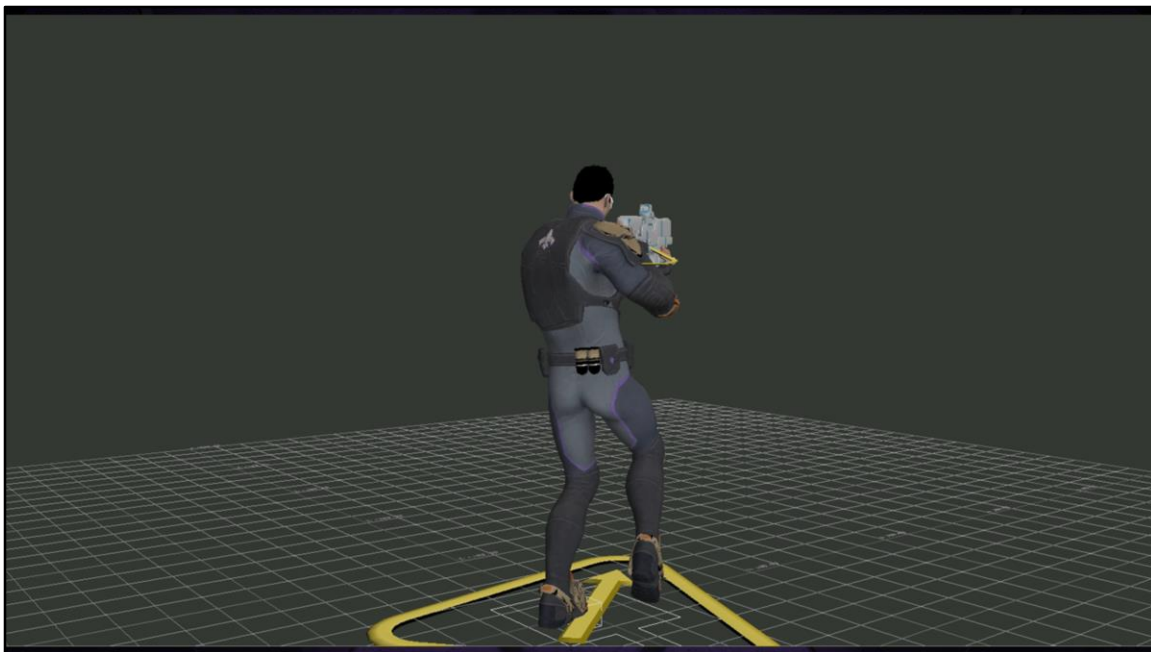
Now, here is the “back right”, which is again at 135 degrees to the right. Again, this is a side step motion, but this time the right leg is crossing in front of the left leg.



Here is the “back” run, which is 135 degrees to the left and a fairly straight forward back peddling motion.



Here is the “back left” which is another side step, moving at 45 degrees left. Here, the left leg crosses the right leg.



And finally, the “forward left” which is again a 45 degrees left side step, but now the right foot crosses in front of the left.



And here they all are in action. 6 animations for jog. And another 6 for walks. And that gives us all of our combat movement for a character. It takes a little getting used to, when directing the mocap and authoring the animations, but once you wrap your head around it, you get everything you need. And it is entirely animation and pose driven, no IK tech needed.



Alright, since we are already talking graphs, let's fully indulge our techy side. Beyond the evolving animation graph improvements and additions, the big gamble for us was procedural as inspired by David Rosen's Animation Bootcamp talk a few years ago. What this meant specifically for us, was ik modifiers...



as created by another ace animation programmer and Buddy Christ look alike, Rob Rypka. And it all started with...



Jump Lands. Jumping is a big part of our game and jump lands can account for nearly 40 animations. Jump Land Stand. Jump land walk, jog sprint. Aiming Jump Lands in all directions across all speeds. For a largely functional animation, it can be giant time sink. Some games might just play the jump land stand, even while moving, to get the compression of the land, and just blend out sooner. But that wasn't going to meet our visual quality needs. So we decided to try procedural jump lands.



Here is the jump lands when standing still. Everything the body and head are doing are procedural. The arms are animated and exported as override animations. But under the hood, this is just the default idle animation playing arms only animation.



And here is jump lands when running. It uses the same arms override animations, but with an earlier blend out. Other than that, this is just the run animation. And this here saved us over 30 animations per character. Not to mention it maintains full player control and velocity of the character, which design loves.

The logo for 'AGENTS OF MAYHEM' is displayed in a stylized, blocky font. The word 'AGENTS' is positioned above 'OF MAYHEM', with 'OF' being smaller and centered between the two words. The logo is white and set against a dark purple background that features a faint, repeating pattern of the game's star-like logo.

**AGENTS
OF
MAYHEM**

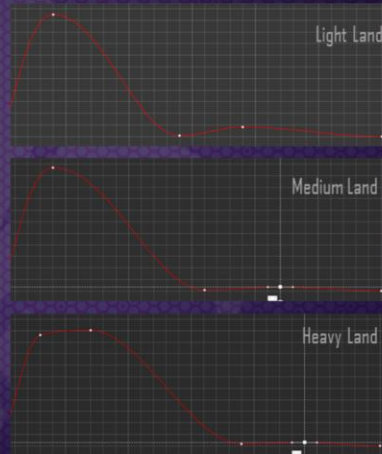
HOW DOES IT WORK?

So how does this work? I can't speak to the code but I can show how we can adjust the knobs and dials exposed to animators to manipulate what it does.

AGENTS MAYHEM

CONTROL DRIVEN IK MODIFIERS

- Uses Defined Curves

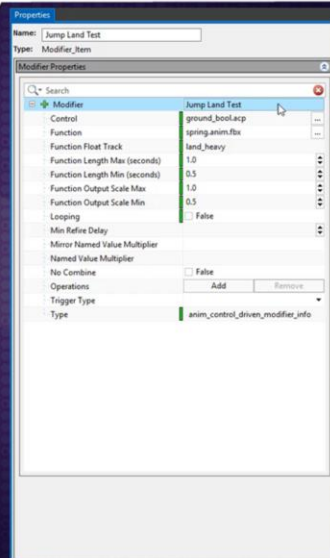


It starts with defining a curve, that is created in Mobu. For the purposes of jump lands, we have ones defining how heavy we want the land to feel. The curves drive what the bones will do. So in the light land, there will be a large value followed by a smaller value second. A sort of double bounce. For the heavy land, there is a large value that holds for a bit, and then settles back without much of a noticeable second value. This gives a land and hold before rising.

AGENTS MAYHEM

CONTROL DRIVEN IK MODIFIERS

- Uses Defined Curves
- Driven by Control Parameter

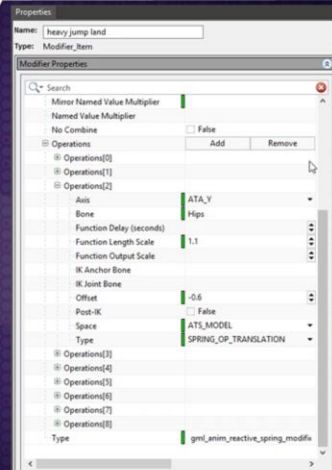


We then set what control parameter we want to use to fire off the modifier. In most cases this is a bool, like ground bool == true in the case of the jump land here. The function is the mobu file containing the previous curves. And the function float track is the land heavy curve. We can also set global values for length and output scale of the curve.

AGENTS MAYHEM

CONTROL DRIVEN IK MODIFIERS

- Uses Defined Curves
- Driven by Control Parameter
- Set Per Bone, Per Axis
 - Translation, Rotation, and Scale
- Set Delay
- Set Bone or Model Space



Next, we add an operation. First is the axis we want to modify and then the bone. We can also set the delay at which we want these to happen to allow for offsets. And finally, if we want the value to be defined by bone or model space. These are created for each bone, each axis and each function. As you can imagine, the number of operations can get pretty insane, so defining the order of operations you want to influence is key. So let's create one of these from scratch, to show what is going on.



Here is what our jump lands look like with no modifier. Just a stand with overlapping arms.



Now lets add translation down in Y on the hips.



Now, some rotation forward in x on the spine, with a delay set to happen later than the translation. Having these live update is a MUST, as there is a lot of iteration that needs to happen and being able to quickly compare and edit is key.



And about 15 operations later, we arrive at our jump lands. Obviously, creating these still take some time and iteration to get right, so the other big win with these comes in that once one is created, we can add it to any character, and their jump land is done. Once the 3 weights were created, all we needed to do was animate the arms for each character, and we were good to go.



The only time this isn't used is during long fall lands. There is no fall damage for agents, but they are locked in place for a bit when they land, which means it is a perfect time to add personality to the character. Like Hollywood, who by his name sake, strikes a 3 point landing.



BUT, Rob wasn't done. He made it that when you toggle sprint while jogging, we play a modifier that compresses the character a bit, giving the feel of pushing off and exerting some force into speeding up. Since this layers on top of the sprint that is blended in, it doesn't change the velocity, making it feel fluid while still having weight.



It was at this point I fell madly in love with modifiers, and they quickly became the Frank's Hot Sauce of animation tech.



We added it to the Out of Combat to Combat transition. The blend to aim is only a couple of frames so the modifier gives us a nice bit of follow through and overlap on the character without delaying combat responsiveness. Contrast this with the bespoke animations we make when transitioning out of combat. Again, this matches with our decision to focus on out of combat animations. Showing how they transition out of combat has become a fun little way of adding a bit of personality to the character, and help again draw the focus back to the agent.



Then, we actually used them as the combat move starts and stops. We have 4 cardinal starts and a single stop. Since combat movement quickly ramps the character's speed to the player's desired speed, this gives us a bit of a push off like the jog to sprint transition and a settle like the out of combat to combat transition. And is shared across all characters.



Weapon fires also use these to great effect, because as the library grows, we can add these to new characters or quickly dupe and edit to get unique weapon fires per character. They are also easy to adjust as shot fire speeds are iterated on by design, by playing with the overall output scale or delay of different operations.



This also gives us additive hit reactions, which you can see here, compared to the larger hit reactions that actually interrupt the character's actions. By the nature of the modifiers, it means they can work while standing still, moving, crouching, mantling, it doesn't matter the state they are in. We can also use these on the player without interrupting player control.



Then, we got the ability to scale bones with modifiers, so I added some squash and stretch to the agents when they swap out.



At which point we can get silly and procedurally add squash and stretch to anything in the game.



And finally, the last big hurdle we needed to jump was facial animation. The tech solutions and expectations out there are incredible and while we wanted higher quality facial animation than past games, investing in facial capture or lots of custom hand keyed animation wasn't going to work. So leaning into style here made a lot of sense.

AGENTS MAYHEM

FACIAL STYLE

- Strong, Expressive Expressions



This meant focusing on a strong, emotive personality expression, like this iconic Glen Keane expression sheet created for Tangled. We would try to hit poses like those, then animate within them.

AGENTS MAYHEM

FACIAL STYLE

- Strong, Expressive Expressions
- Snappy Transitions



Specifically:

- Expression changes are what take the most amount of time to sell, on a physicality and emotional level. We don't have the bandwidth or need for that. Best to make a scrunch face to broadly sell the contrast of the emotions.
- This keeps it consistent with the rest of our extraordinary style assets: Create strong, personality driven poses to carry you through the action, with only small bits to keep it alive.

AGENTS MAYHEM

FACIAL STYLE

- Strong, Expressive Expressions
- Snappy Transitions
- Procedural Mouth



People are willing to accept mouths that aren't perfect in games, but it is the dead eyes/expressions that people comment on. We nail the eyes/expressions, we can let procedural handle the mouth.



This leads us to the 7 universal emotions.



Anger. Disgust. Fear. Happiness. Sadness. Surprise. Contempt. Every human has them. And while each expression has specific traits needed for them to read, everyone has their own unique versions.



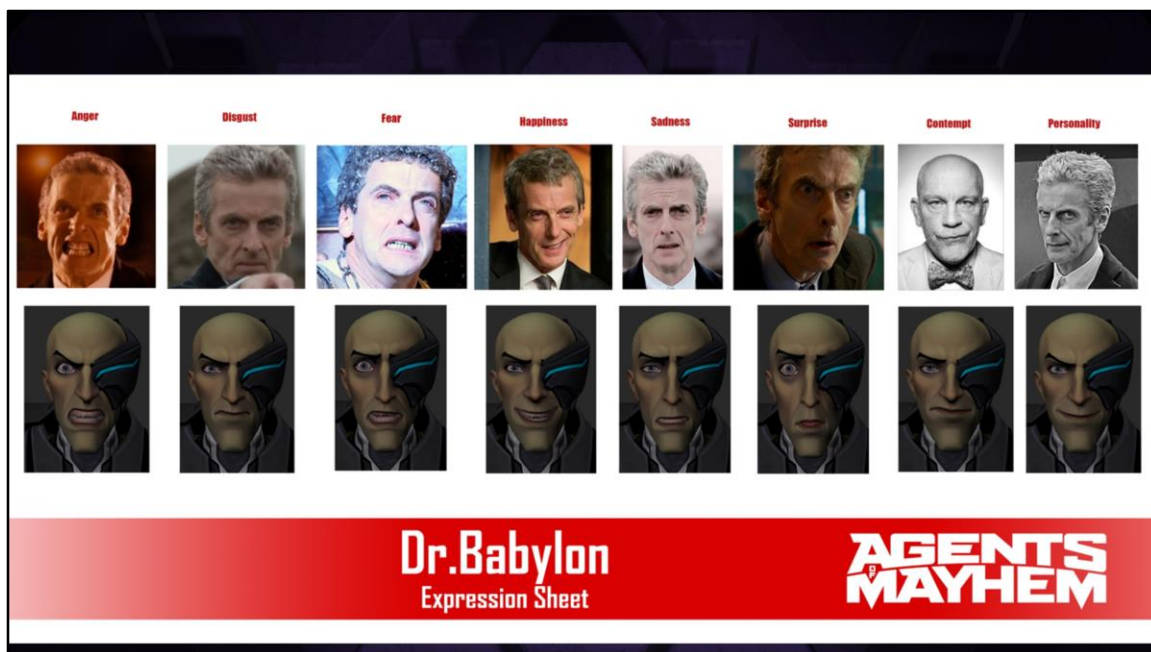
So we did this for each character as well. And much like defining their music, figuring out which actors or characters parallel our own character's personalities meant we had another quick touchstone for finding and communicating performances.



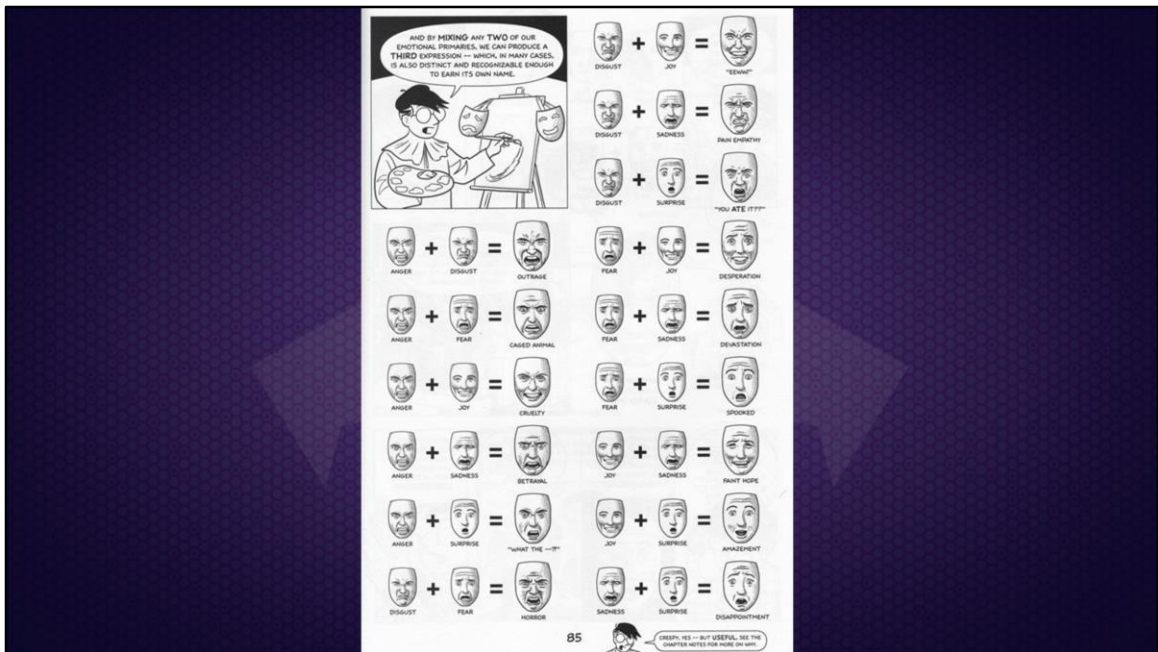
These expressions, along with the annosoft visemes, were all outsourced to animators with a past working in film. The reference expression pictures are what we gave them as direction. And what that gave us were solid building blocks that were able to be crafted by a dedicated team for us to then animate with later. Essentially, animator driven blend shapes, as demoed here by a tool written by our Anim Tech Lead Evan Cox. We can also bring in the Annosoft generated lip sync animation, to get first pass lip sync for anything that requires it.



Outside of a few specific animations (idle, fidgets, emotes, mayhem ability, front vehicle entry) where the player can easily move the camera around and expect the face to match the actions, most animations are authored with no expression. This allows a default expression playing at all times to show through. Most of the time when running, jumping, etc, this default expression is what is seen. Which lead us to creating an 8th expression for each character.



As seen here as the last expression, this is a unique personality pose for each character, that they would play in game as their default state. In this case, it is our head baddie, Dr.Babylon. We also decided at this point, to break up the expressions in our tool to have sliders for the entire face, eyes only or mouth only.



Using Scott McCloud's approach to facial expressions as inspiration, being able to mix and match the eyes and mouth of expressions meant we could quickly create variant expressions as needed. Combined with all of the viseme shapes, we could now quickly make any expression needed, entirely unique to the character.



So with Fortune, getting her reach her full Rihanna potential took nearly no time at all. With those static expressions now crafted and implemented in the tool, having them leveraged software side seemed like only a half measure. We wanted to bring them to life in realtime.



So we had our facial outsourcing gurus give us a single, living expression to keep the face alive for each character. Again, having them focus on it meant it would meet a consistent, higher quality than if we tried to rush and do it internally, when balanced against gameplay needs.



Then, we can quickly copy multiple takes of this animation, drop each unique expression on a layer over it and now we have all 8 expressions alive and at a fairly consistent quality. And at that point,



we have a system to call those expressions in game. Is the agent happy to pick up a shiny new object? Play their happy expression! Are they angry that LEGION is around? Play their angry expression. Did they just witness something weird? Play their disgusted expression! Are they at low health? Play their fear expression. Facial animation in our game is the definition of solving something with a system, and then leveraging that system in as many places as possible.



Alright, so after defining the characters, style, quality and tech we were finally able to extrapolate the cost of creation for one agent.

AGENTS MAYHEM

THE BREAKDOWN

- < 100 animations per agent

We ended up just shy of 100 animations per player character.

Out of Combat State					Combat Actions					Remaining Int Days to Fully Functional	Remaining Int Days to Fully Polished	
Group	Animation Name	Asset	Implement	Int/Ext	Group	Animation Name	Asset	Implement	Int/Ext			
Default	Stand	1	0.25	External	Group	Animation Name	Asset	Implement	Int/Ext	19.65	0	
Default	Weapons Out	0.5	0.1	External	Combat	Special Ability	2	0.1	Internal			
Default	Walk	0.5	0.25	External	Combat	Ultimate Ability	3	0.1	Internal			
Default	Walk Stop	0.5	0.25	External	Combat	Melee	2	0.1	Internal			
Default	Jog	0.5	0.25	External	Aim	Aim Reload	1	0.5	External			
Default	Jog Stop	0.5	0.25	External	Aim	Aim Reload Override	0	0.1	External	Shared Assets		
Default	Jog Start	0.5	0.25	External						3.93	0	
Default	Jog Start Left	0.5	0.25	External	Group	Animation Name	Asset	Implement	Int/Ext			
Default	Jog Start Right	0.5	0.25	External	Default	Head Turn Left	0.1	0.1	Internal			
Default	Jog Start Back	0.5	0.25	External	Default	Head Turn Right	0.1	0.1	Internal			
Default	Jog Turn 180 Left	1	0.25	External	Default	Fail	0.5	0.1	Internal			
Default	Jog Turn 180 Right	1	0.25	External	Default	Fail Left	0.25	0.1	Internal	22.2	0	
Default	Sprint	0.5	0.25	External	Default	Fail Right	0.25	0.1	Internal			
Default	Sprint Stop	0.5	0.25	External	Default	Run Off	0.5	0.1	Internal			
Jumping					Default	Long Fall Land	1.5	0.1	Internal			
Group	Animation Name	Asset	Implement	Int/Ext	Aim	Jog Forward Left	0.75	0.25	Internal	Shared Asset Time		
Default	Jump	1	0.25	Internal	Aim	Jog Forward	0.75	0.25	Internal	19.2		
Default	Double Jump	1.5	0.25	Internal	Aim	Jog Forward Right	0.75	0.25	Internal	22.2	0	
Default	Procedural Jump Lands	1	0.1	Internal	Aim	Jog Back Right	0.75	0.25	Internal			
Default	Short Jump/Edge Assist	1	0.1	Internal	Aim	Jog Back	0.75	0.25	Internal			
Default	Unique Traversal	1.5	0.1	Internal	Aim	Jog Back Left	0.75	0.25	Internal			
Aim Jumping					Aim	Walk Forward Left	0.75	0.25	Internal			
Group	Animation Name	Asset	Implement	Int/Ext	Aim	Walk Forward	0.75	0.25	Internal	4.44	0	
Aim	Aim Jump	0.3	0.1	Internal	Aim	Walk Forward Right	0.75	0.25	Internal			
Aim	Aim Double Jump	0.3	0.1	Internal	Aim	Walk Back Right	0.75	0.25	Internal			
Aim	Aim Triple Jump	0.3	0.1	Internal	Aim	Walk Back	0.75	0.25	Internal			
Aim	Aim Short Jump	0.3	0.1	Internal	Aim	Walk Back Left	0.75	0.25	Internal			
Emotes					Aim	Aim Run Off	0.25	0.1	Internal	Remaining Weeks		
Group	Animation Name	Asset	Implement	Int/Ext	Aim	Aim Fall	0.25	0.1	Internal	Remaining Weeks		
Default	Compliment 01	1	0.25	External	Aim	Aim Dash Forward	0.5	0.1	Internal	5.2	55	
Default	Tsunt 01	1	0.25	External	Aim	Aim Dash Back	0.5	0.1	Internal			
Default	Fidget 01	1	0.25	External	Aim	Aim Dash Left	0.5	0.1	Internal			
Default	Agent Swap Exit	1	0.25	External	Aim	Aim Dash Right	0.5	0.1	Internal			
Vehicles					Aim	Aim Slide Forward	0.5	0.1	Internal			
Group	Animation Name	Asset	Implement	Int/Ext	Aim	Aim Slide Back	0.5	0.1	Internal	Shared Implement		
	Front Entry	2	0.25	Internal	Aim	Aim Slide Left	0.5	0.1	Internal	5.2		
	Passenger Entry	1	0.1	Internal	Aim	Aim Slide Right	0.5	0.1	Internal	Int/Ext		
	Driver Entry	1	0.1	Internal	Aim	Aim Stagger Front	0.5	0.1	Internal	Internal		28
	Back Entry	1	0.1	Internal	Aim	Aim Stagger Right	0.5	0.1	Internal			
Combat State					Aim	Aim Stagger Back	0.5	0.1	Internal			
Group	Animation Name	Asset	Implement	Int/Ext	Aim	Knockback	0.5	0.1	Internal			
Aim	Aim Stand	0.5	0.1	External								
Aim	Aim to Default	0.5	0.1	External								
Aim	Aim Turn Left	0.5	0.1	External								
Aim	Aim Turn Left Fast	0.5	0.1	External								
Aim	Aim Turn Right	0.5	0.1	External								
Aim	Aim Turn Right Fast	0.5	0.1	External								

This is all of them, copied from our character excel doc, because I know you all animate in excel sometimes and secretly love it.

Out of Combat State					Combat Actions					Remaining Int Days to Fully Functional	Remaining Int Days to Fully Polished
Group	Animation Name	Asset	Implement	Int/Ext	Group	Animation Name	Asset	Implement	Int/Ext		
Default	Stand	1	0.25	External	Group	Animation Name	Asset	Implement	Int/Ext	19.65	0
Default	Weapons Out	0.5	0.1	External	Combat	Special Ability	2	0.1	Internal		
Default	Walk	0.5	0.25	External	Combat	Ultimate Ability	3	0.1	Internal		
Default	Walk Stop	0.5	0.25	External	Combat	Melee	2	0.1	Internal		
Default	Jog	0.5	0.25	External	Aim	Aim Reload	1	0.5	External		
Default	Jog Stop	0.5	0.25	External	Aim	Aim Reload Override	0	0.1	External	3.93	0
Default	Jog Start	0.5	0.25	External	Shared Assets						
Default	Jog Start Left	0.5	0.25	External	Group	Animation Name	Asset	Implement	Int/Ext		
Default	Jog Start Right	0.5	0.25	External	Default	Head Turn Left	0.1	0.1	Internal		
Default	Jog Start Back	0.5	0.25	External	Default	Head Turn Right	0.1	0.1	Internal		
Default	Jog Turn 180 Left	1	0.25	External	Default	Fall	0.5	0.1	Internal		
Default	Jog Turn 180 Right	1	0.25	External	Default	Fall Left	0.25	0.1	Internal		
Default	Sprint	0.5	0.25	External	Default	Fall Right	0.25	0.1	Internal		
Default	Sprint Stop	0.5	0.25	External	Default	Run Off	0.5	0.1	Internal		
Jumping					Default	Long Fall Land	1.5	0.1	Internal		
Group	Animation Name	Asset	Implement	Int/Ext	Aim	Jog Forward Left	0.75	0.25	Internal	19.2	Shared Asset Time
Default	Jump	1	0.25	Internal	Aim	Jog Forward	0.75	0.25	Internal		
Default	Double Jump	1.5	0.25	Internal	Aim	Jog Forward Right	0.75	0.25	Internal		
Default	Procedural Jump Lands	1	0.1	Internal	Aim	Jog Back Right	0.75	0.25	Internal		
Default	Short Jump/Edge Assist	1	0.1	Internal	Aim	Jog Back	0.75	0.25	Internal		
Default	Unique Traversal	1.5	0.1	Internal	Aim	Jog Back Left	0.75	0.25	Internal	22.2	0
Aim Jumping					Aim	Walk Forward Left	0.75	0.25	Internal		
Group	Animation Name	Asset	Implement	Int/Ext	Aim	Walk Forward	0.75	0.25	Internal		
Aim	Aim Jump	0.3	0.1	Internal	Aim	Walk Forward Right	0.75	0.25	Internal		
Aim	Aim Double Jump	0.3	0.1	Internal	Aim	Walk Back Right	0.75	0.25	Internal		
Aim	Aim Triple Jump	0.3	0.1	Internal	Aim	Walk Back	0.75	0.25	Internal		
Aim	Aim Short Jump	0.3	0.1	Internal	Aim	Walk Back Left	0.75	0.25	Internal	4.44	0
Emotes					Aim	Aim Run Off	0.25	0.1	Internal		
Group	Animation Name	Asset	Implement	Int/Ext	Aim	Aim Fall	0.25	0.1	Internal		
Default	Compliment 01	1	0.25	External	Aim	Aim Dash Forward	0.5	0.1	Internal		
Default	Tsunt 01	1	0.25	External	Aim	Aim Dash Back	0.5	0.1	Internal		
Default	Fidget 01	1	0.25	External	Aim	Aim Dash Left	0.5	0.1	Internal		
Default	Agent Swap Exit	1	0.25	External	Aim	Aim Dash Right	0.5	0.1	Internal	5.2	Shared Implement
Vehicles					Aim	Aim Slide Forward	0.5	0.1	Internal		
Group	Animation Name	Asset	Implement	Int/Ext	Aim	Aim Slide Back	0.5	0.1	Internal		
Front Entry		2	0.25	Internal	Aim	Aim Slide Left	0.5	0.1	Internal		
Passenger Entry		1	0.1	Internal	Aim	Aim Slide Right	0.5	0.1	Internal		
Driver Entry		1	0.1	Internal	Aim	Stagger Front	0.5	0.1	Internal		
Back Entry		1	0.1	Internal	Aim	Stagger Right	0.5	0.1	Internal	55	28
Combat State					Aim	Stagger Back	0.5	0.1	Internal		
Group	Animation Name	Asset	Implement	Int/Ext	Aim	Stagger Front	0.5	0.1	Internal		
Aim	Aim Stand	0.5	0.1	External	Aim	Stagger Right	0.5	0.1	Internal		
Aim	Aim to Default	0.5	0.1	External	Aim	Stagger Back	0.5	0.1	Internal		
Aim	Aim Turn Left	0.5	0.1	External	Aim	Stagger Front	0.5	0.1	Internal		
Aim	Aim Turn Left Fast	0.5	0.1	External	Aim	Knockback	0.5	0.1	Internal		
Aim	Aim Turn Right	0.5	0.1	External	Aim						
Aim	Aim Turn Right Fast	0.5	0.1	External	Aim						

Of note, something we track and define here is which assets can be shared from another character, which is managed by having all of our characters animation sets live within a hierarchy. So as soon as one character is complete, every character can inherit their animations, allowing design to start testing them right away, and we can update animations as needed.

Out of Combat State					Combat Actions					Remaining Int Days to Fully Functional	Remaining Int Days to Fully Polished	
Group	Animation Name	Asset	Implement	Int/Ext	Group	Animation Name	Asset	Implement	Int/Ext			
Default	Stand	1	0.25	External	Group	Animation Name	Asset	Implement	Int/Ext	19.65	0	
Default	Weapons Out	0.5	0.1	External	Combat	Special Ability	2	0.1	Internal			
Default	Walk	0.5	0.25	External	Combat	Ultimate Ability	3	0.1	Internal			
Default	Walk Stop	0.5	0.25	External	Combat	Melee	2	0.1	Internal			
Default	Jog	0.5	0.25	External	Aim	Aim Reload	1	0.5	External			
Default	Jog Stop	0.5	0.25	External	Aim	Aim Reload Override	0	0.1	External			
Default	Jog Start	0.5	0.25	External	Shared Assets							
Default	Jog Start Left	0.5	0.25	External	Group	Animation Name	Asset	Implement	Int/Ext	3.93	0	
Default	Jog Start Right	0.5	0.25	External	Default	Head Turn Left	0.1	0.1	Internal			
Default	Jog Start Back	0.5	0.25	External	Default	Head Turn Right	0.1	0.1	Internal			
Default	Jog Turn 180 Left	1	0.25	External	Default	Fall	0.5	0.1	Internal			
Default	Jog Turn 180 Right	1	0.25	External	Default	Fall Left	0.25	0.1	Internal			
Default	Sprint	0.5	0.25	External	Default	Fall Right	0.25	0.1	Internal	22.2	0	
Default	Sprint Stop	0.5	0.25	External	Default	Run Off	0.5	0.1	Internal			
Jumping					Default	Long Fall Land	1.5	0.1	Internal			
Group	Animation Name	Asset	Implement	Int/Ext	Aim	Jog Forward Left	0.75	0.25	Internal			Shared Asset Time
Default	Jump	1	0.25	Internal	Aim	Jog Forward	0.75	0.25	Internal	19.2		
Default	Double Jump	1.5	0.25	Internal	Aim	Jog Forward Right	0.75	0.25	Internal	22.2		0
Default	Procedural Jump Lands	1	0.1	Internal	Aim	Jog Back Right	0.75	0.25	Internal			
Default	Short Jump/Edge Assist	1	0.1	Internal	Aim	Jog Back	0.75	0.25	Internal			
Default	Unique Traversal	1.5	0.1	Internal	Aim	Jog Back Left	0.75	0.25	Internal			
Group	Animation Name	Asset	Implement	Int/Ext	Aim	Walk Forward Left	0.75	0.25	Internal		4.44	
Aim	Aim Jump	0.3	0.1	Internal	Aim	Walk Forward	0.75	0.25	Internal			
Aim	Aim Double Jump	0.3	0.1	Internal	Aim	Walk Forward Right	0.75	0.25	Internal			
Aim	Aim Triple Jump	0.3	0.1	Internal	Aim	Walk Back Right	0.75	0.25	Internal			
Aim	Aim Short Jump	0.3	0.1	Internal	Aim	Walk Back	0.75	0.25	Internal			
Emotes					Aim	Walk Back Left	0.75	0.25	Internal	Shared Implement		
Group	Animation Name	Asset	Implement	Int/Ext	Aim	Aim Run Off	0.25	0.1	Internal	5.2		
Default	Compliment 01	1	0.25	External	Aim	Aim Fall	0.25	0.1	Internal	55	28	
Default	Tsunt 01	1	0.25	External	Aim	Aim Dash Forward	0.5	0.1	Internal			
Default	Fidget 01	1	0.25	External	Aim	Aim Dash Back	0.5	0.1	Internal			
Default	Agent Swap Exit	1	0.25	External	Aim	Aim Dash Left	0.5	0.1	Internal			
Vehicles					Aim	Aim Dash Right	0.5	0.1	Internal			
Group	Animation Name	Asset	Implement	Int/Ext	Aim	Aim Slide Forward	0.5	0.1	Internal	Int/Ext		
	Front Entry	2	0.25	Internal	Aim	Aim Slide Back	0.5	0.1	Internal	Internal	55	
	Passenger Entry	1	0.1	Internal	Aim	Aim Slide Left	0.5	0.1	Internal			
	Driver Entry	1	0.1	Internal	Aim	Aim Slide Right	0.5	0.1	Internal	External	28	
	Back Entry	1	0.1	Internal	Aim	Stagger Front	0.5	0.1	Internal			
Group	Animation Name	Asset	Implement	Int/Ext	Aim	Stagger Right	0.5	0.1	Internal			
Aim	Aim Stand	0.5	0.1	External	Aim	Stagger Back	0.5	0.1	Internal			
Aim	Aim to Default	0.5	0.1	External	Aim	Stagger Front	0.5	0.1	Internal			
Aim	Aim Turn Left	0.5	0.1	External	Aim	Knockback	0.5	0.1	Internal			
Aim	Aim Turn Left Fast	0.5	0.1	External								
Aim	Aim Turn Right	0.5	0.1	External								
Aim	Aim Turn Right Fast	0.5	0.1	External								

We also track here which are animated internally vs externally. A lot of the ordinary classified movements are outsourced, including jog starts and turns. Since the mocap gives a solid base and direction, and our metrics are so well defined, communicating our needs to external studios is relatively straight forward. We still have to do a final polish pass on implementation, but it helps to make time to focus on the more extraordinary animations internally.

AGENTS MAYHEM

THE BREAKDOWN

- ~100 animations per agent
- 10 weeks for Unique Characters
- 5 weeks for Shared Characters
- 3 weeks additional polish

After leveraging style, sharing, outsourcing and tech, 100% unique character are estimated to take 10 weeks internally to create/implement all animations. Shared weapon holds can save us 5 weeks. 3 weeks of polish and iteration were then added to each character, to react to any needed design changes.



And this how many characters we have in total. Beyond the 12 playable characters, we have 10 enemy troopers, 7 unique npc, 7 unique bosses and male and female pedestrians. And I can not stress enough how important and inspiring the diversity of these characters has been for the team.



Bringing the characters and animation process to life on this game has required the perfect mix of art, design and tech which speaks to the heart of game animation as I see it. Taking the time to fully define the characters, style, quality and tech up front has allowed us to fully realize what I saw in my head when I started on the game. And moving forward, it has given us an amazing foundation to build upon. I hope you all have been inspired by our techniques and I can't wait for everyone to not only see more of the characters we have created but actually get to play it when we release the game later this year. I couldn't be prouder of what our animation team has accomplished.

AGENTS MAYHEM

West Hall 2002

ART DIRECTING VFX FOR STYLIZED GAMES

Bryanna Lindsey, Senior VFX Artist

2:00-3:00

'AGENTS OF MAYHEM': PHYSICALLY-BASED MATERIALS IN A STYLIZED OPEN WORLD

James Taylor, Principal Artist

3:30-4:30

And if you would like to hear more about the art style on Agents of Mayhem, there are two more talks today in West Hall Room 2002. At 2:00, Senior VFX Artist Bryanna Lindsey will speak about how VFX approaches the game. And at 3:30, Principal Artist James Taylor will speak about how we physically based materials in a stylized game.



Thanks for listening! Who has questions?