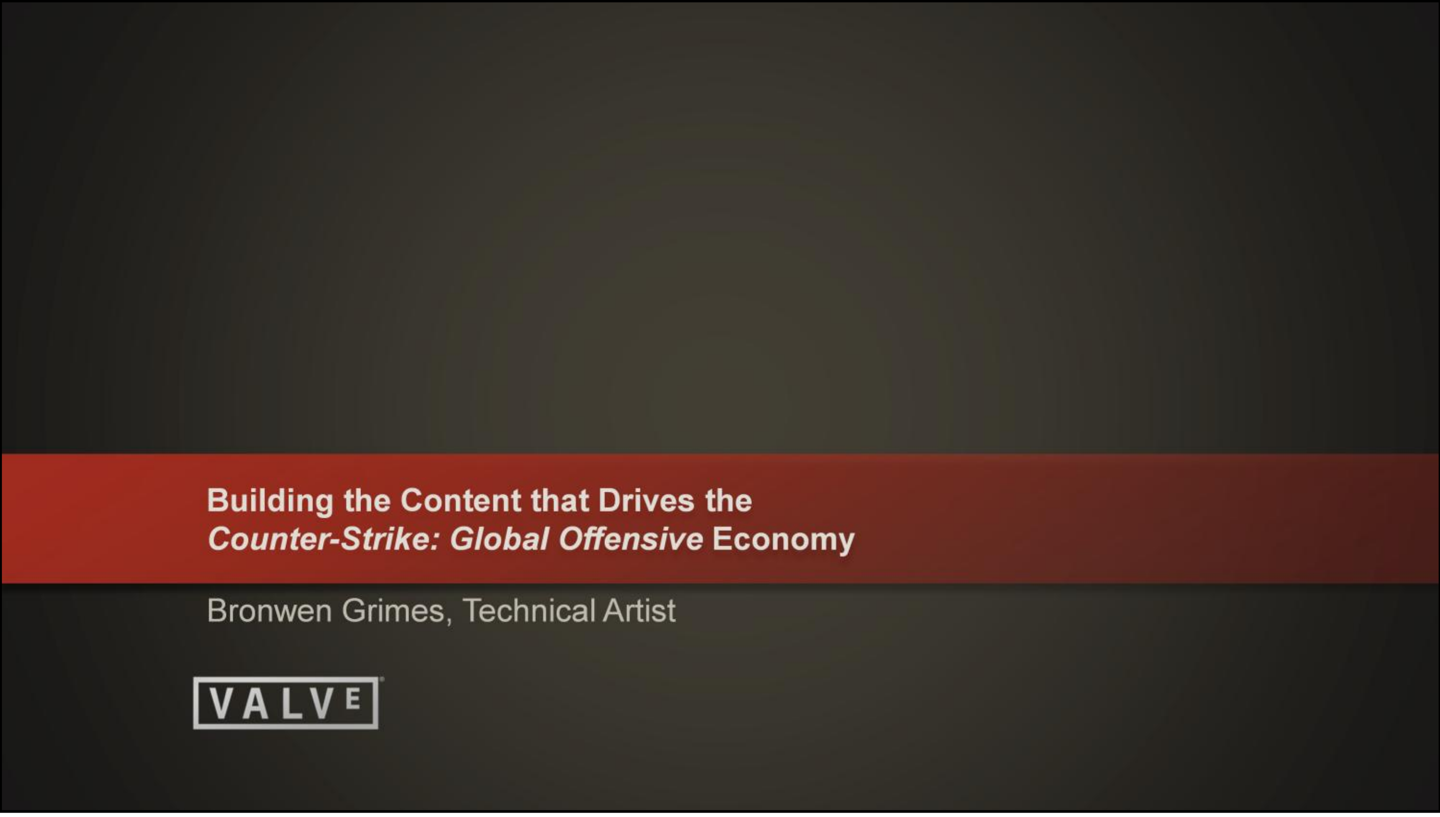


Building the Content that Drives the *Counter-Strike: Global Offensive* Economy

Bronwen Grimes, Technical Artist

VALVE

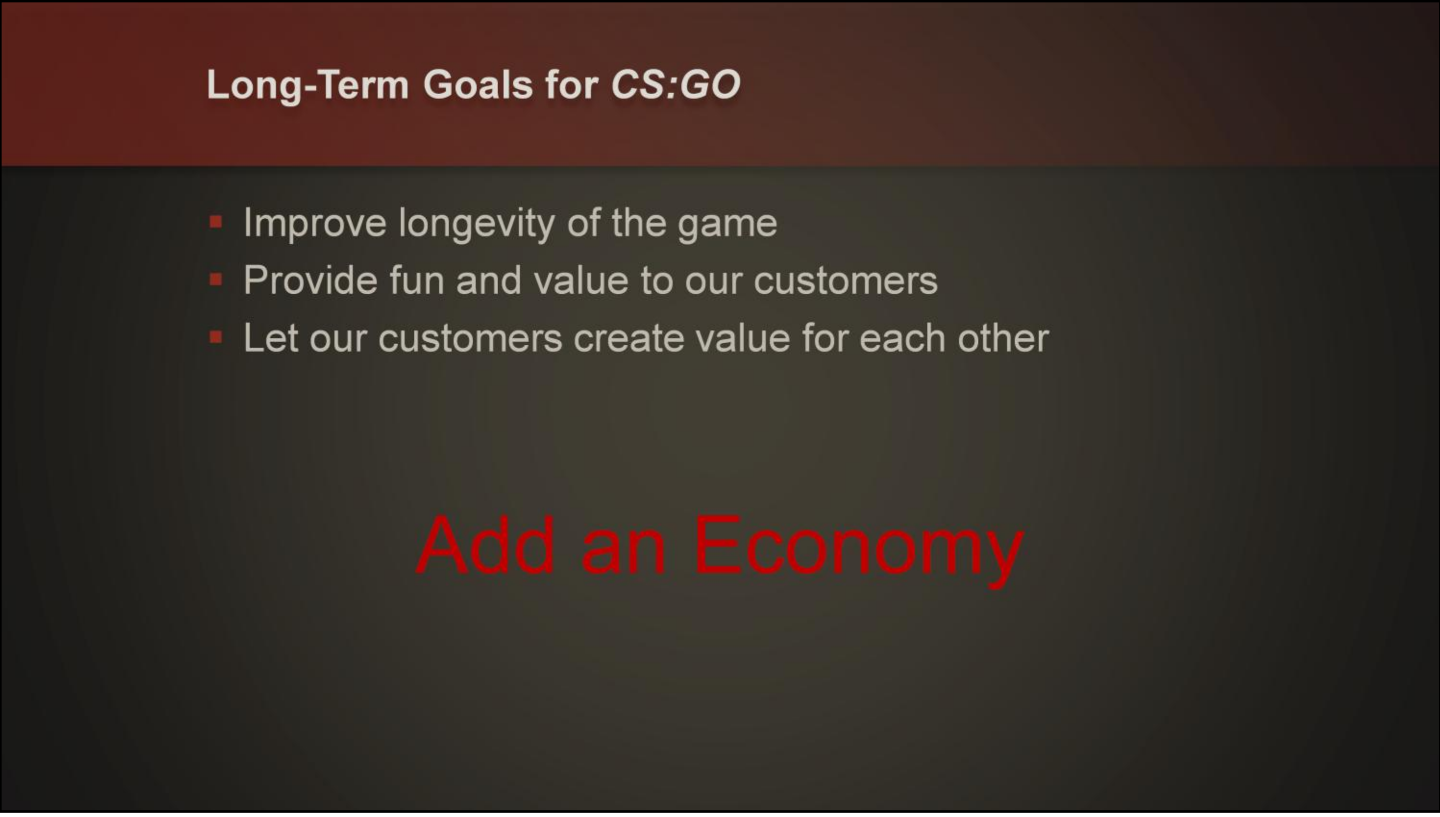
This talk is about how we generated a lot of content with a very small team in order to kick-start our economy. We'll also be discussing the things our customers value about the economy content, and how that information is helping us continue to make good decisions going forward.

This talk won't spend a lot of time on the advantages of adding an economy to your game. We are going to assume the reason you're reading this is because you already know something about that. If, on the other hand, that is something you'd like to learn more about, we recently gave a series of talks on economies at the Steam Developer Days. They're freely available on YouTube. Just head over to steamdevdays.com for some links.

Counter-Strike is an online multiplayer game.

It’s a team-based first person shooter with a variety of game modes, and it’s had several different versions over the last 14 years.

Counter-Strike: *Global Offensive* is the lastest, greatest version. Like we do with our other products, we use Steam to update *CS:GO* regularly as we strive to create the best experience for our customers.



Long-Term Goals for *CS:GO*

- Improve longevity of the game
- Provide fun and value to our customers
- Let our customers create value for each other

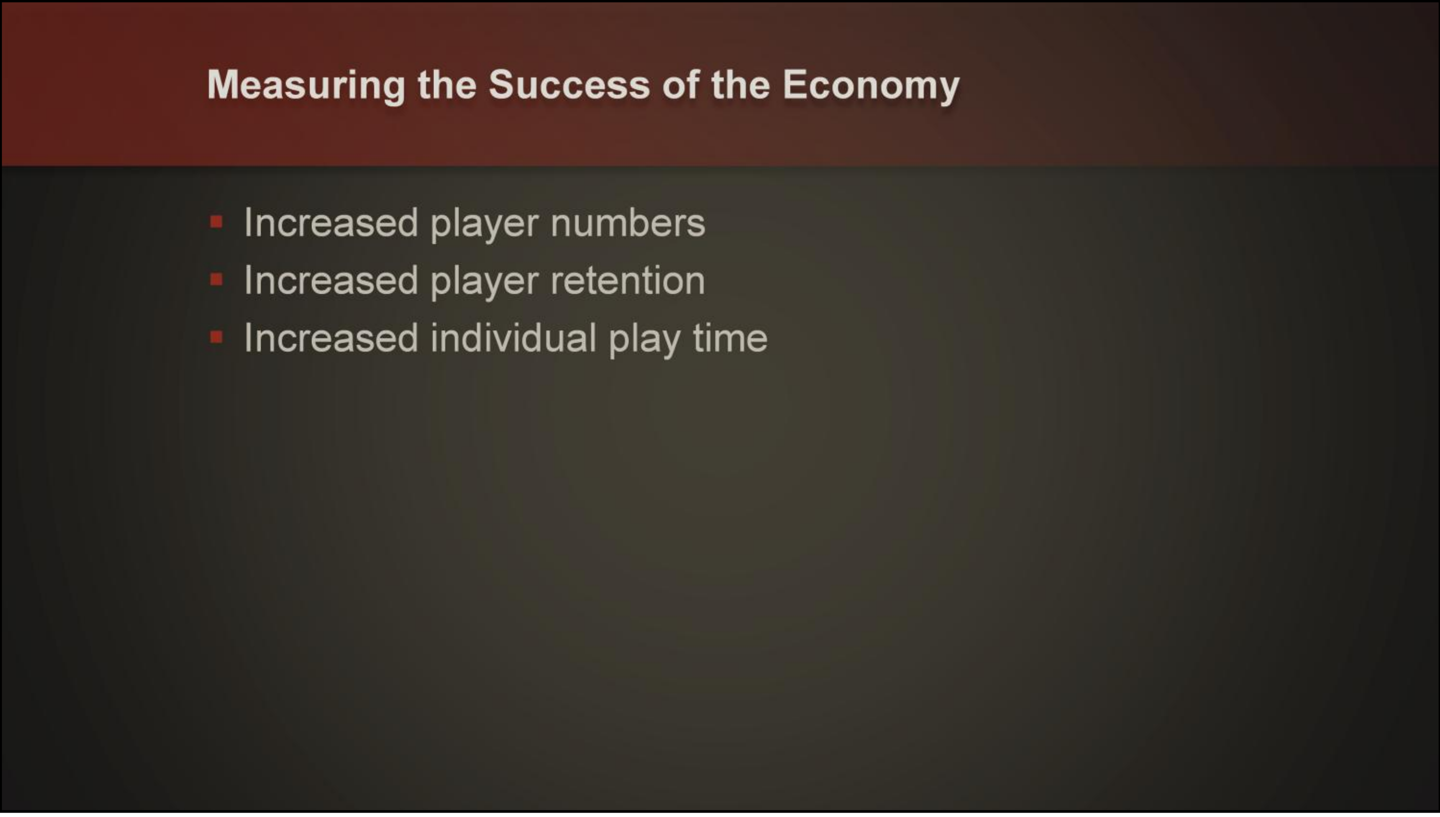
Add an Economy

In a multiplayer game, a big part of a good experience is provided by other players. Our customers generate a huge amount of value for each other just by participating in the game, so we want to encourage them to keep doing that. And we want to do it for years to come.

We encourage our customers to play by, of course, providing them fun and value.

More importantly, we encourage them by making it easy to provide fun and value to each other.

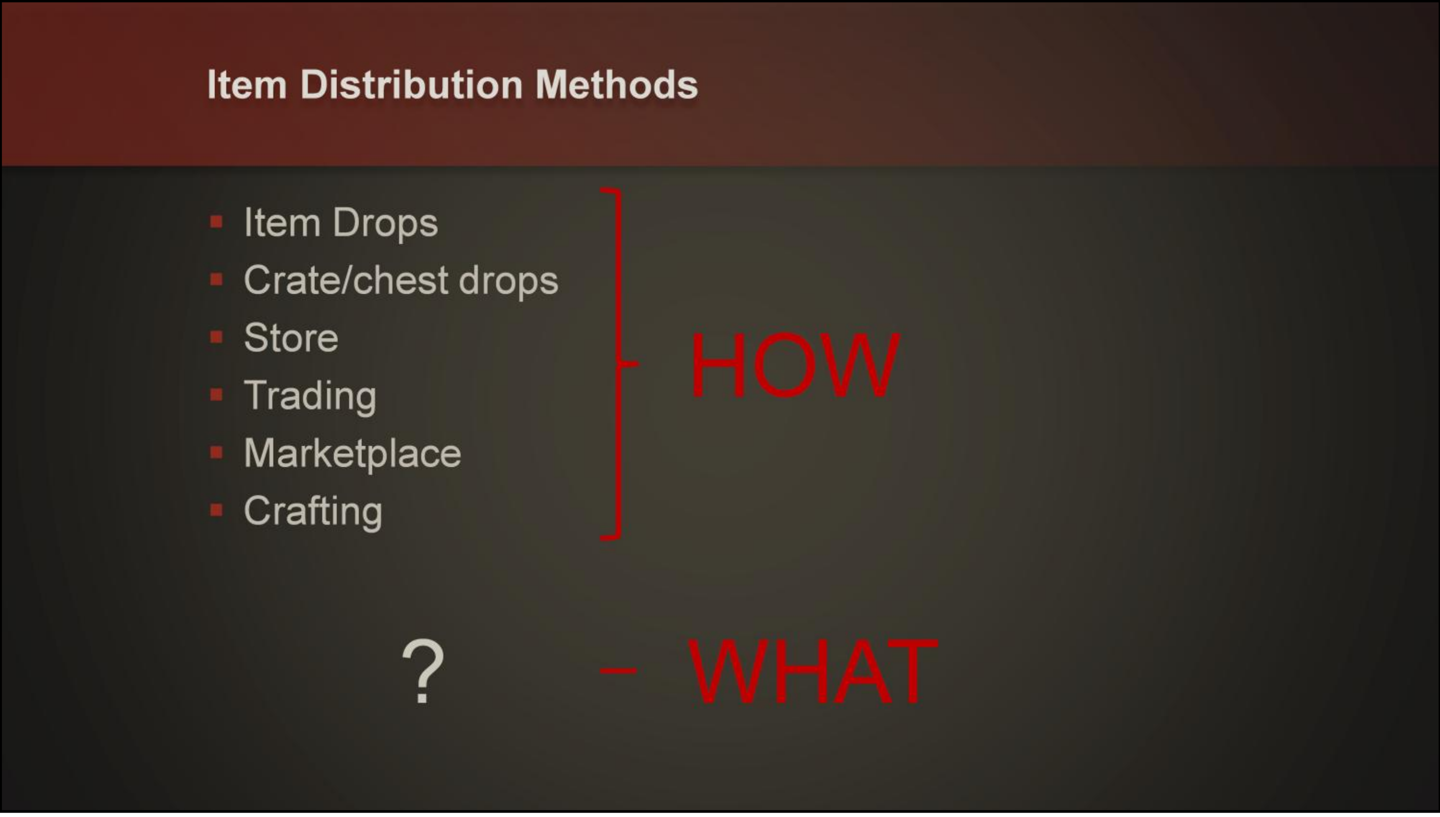
In Valve products like *Dota 2* and *Team Fortress 2*, we've found that a great way to help meet all these goals at once is to add an economy to the game.



Measuring the Success of the Economy

- Increased player numbers
- Increased player retention
- Increased individual play time

We can measure the success of our economy by checking on just a few broad metrics, and they're maybe not the ones you'd think. After all, adding the economy isn't an end in and of itself. We're adding the economy in service of our goals. So we'll look at player numbers, player retention, and play time. If our customers are having fun, they'll come back for more.



Item Distribution Methods

- Item Drops
- Crate/chest drops
- Store
- Trading
- Marketplace
- Crafting

HOW

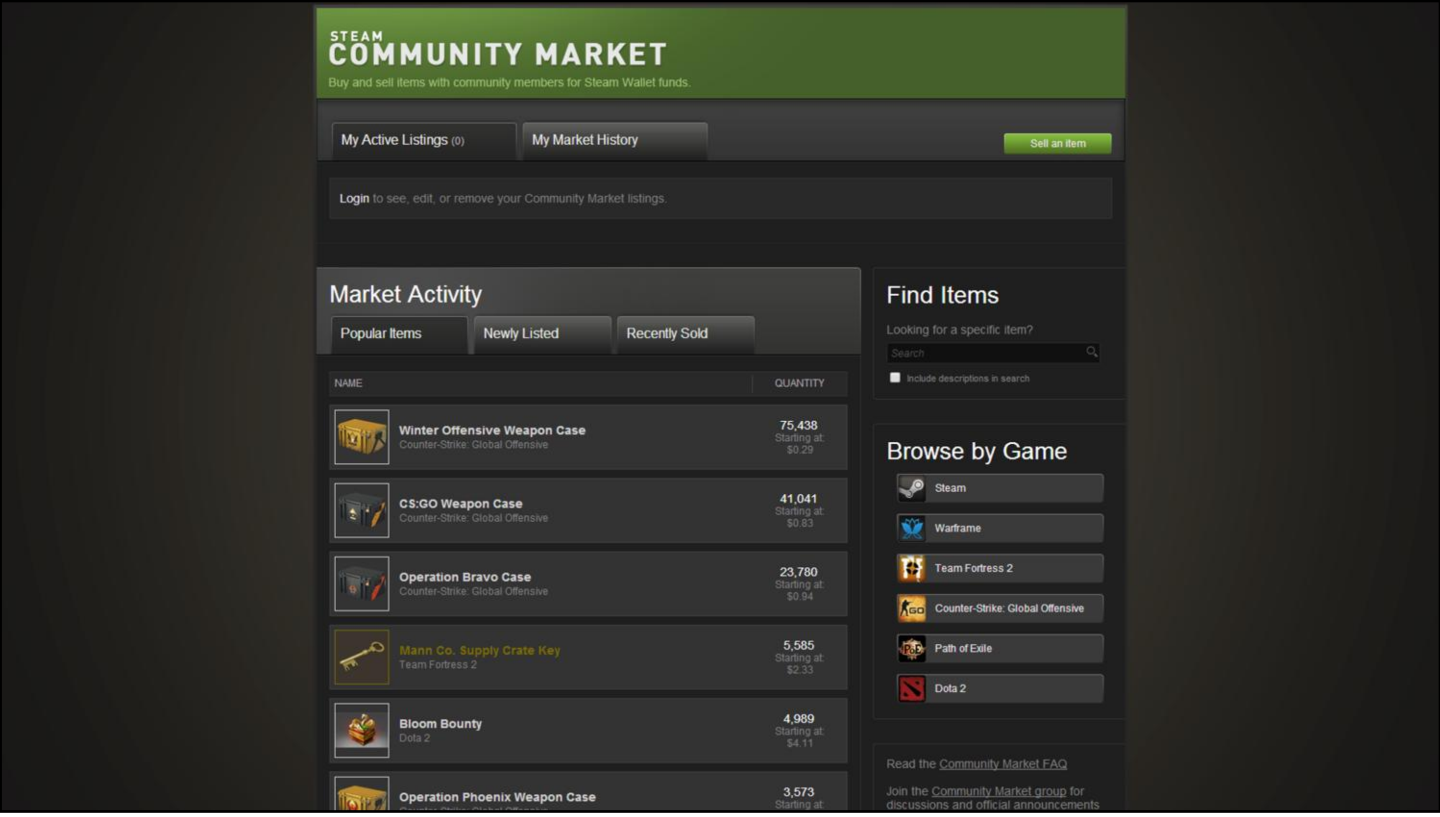
? – WHAT

Since *Team Fortress 2* and *Dota 2* both have thriving economies, we were able to crib a lot of ideas from them. They both use multiple channels for distributing items to players.

That gives customers lots of options for getting items. Having lots of ways for customers to interact with the economy means a broader participation, which is good because economies are essentially multiplayer games: the more people playing, the more fun it is.

In both *Team Fortress 2* and *Dota 2*, not every item is available through every channel. This is perfect when the goal is to provide lots of different ways to interact with the economy and see *how* customers want to consume content. But it's difficult to directly compare the value of items in different categories.

At this point in *CS:GO*'s economy design, figuring out the value of individual items was really important, because we still weren't sure *what* our customers wanted.



STEAM COMMUNITY MARKET

Buy and sell items with community members for Steam Wallet funds.

Login to see, edit, or remove your Community Market listings.

Market Activity

NAME	QUANTITY
Winter Offensive Weapon Case Counter-Strike: Global Offensive	75,438 Starting at: $0.29
CS:GO Weapon Case Counter-Strike: Global Offensive	41,041 Starting at: $0.83
Operation Bravo Case Counter-Strike: Global Offensive	23,780 Starting at: $0.94
Mann Co. Supply Crate Key Team Fortress 2	5,585 Starting at: $2.33
Bloom Bounty Dota 2	4,989 Starting at: $4.11
Operation Phoenix Weapon Case	3,573 Starting at:

Find Items

Looking for a specific item?

Browse by Game

Read the Community Market FAQ

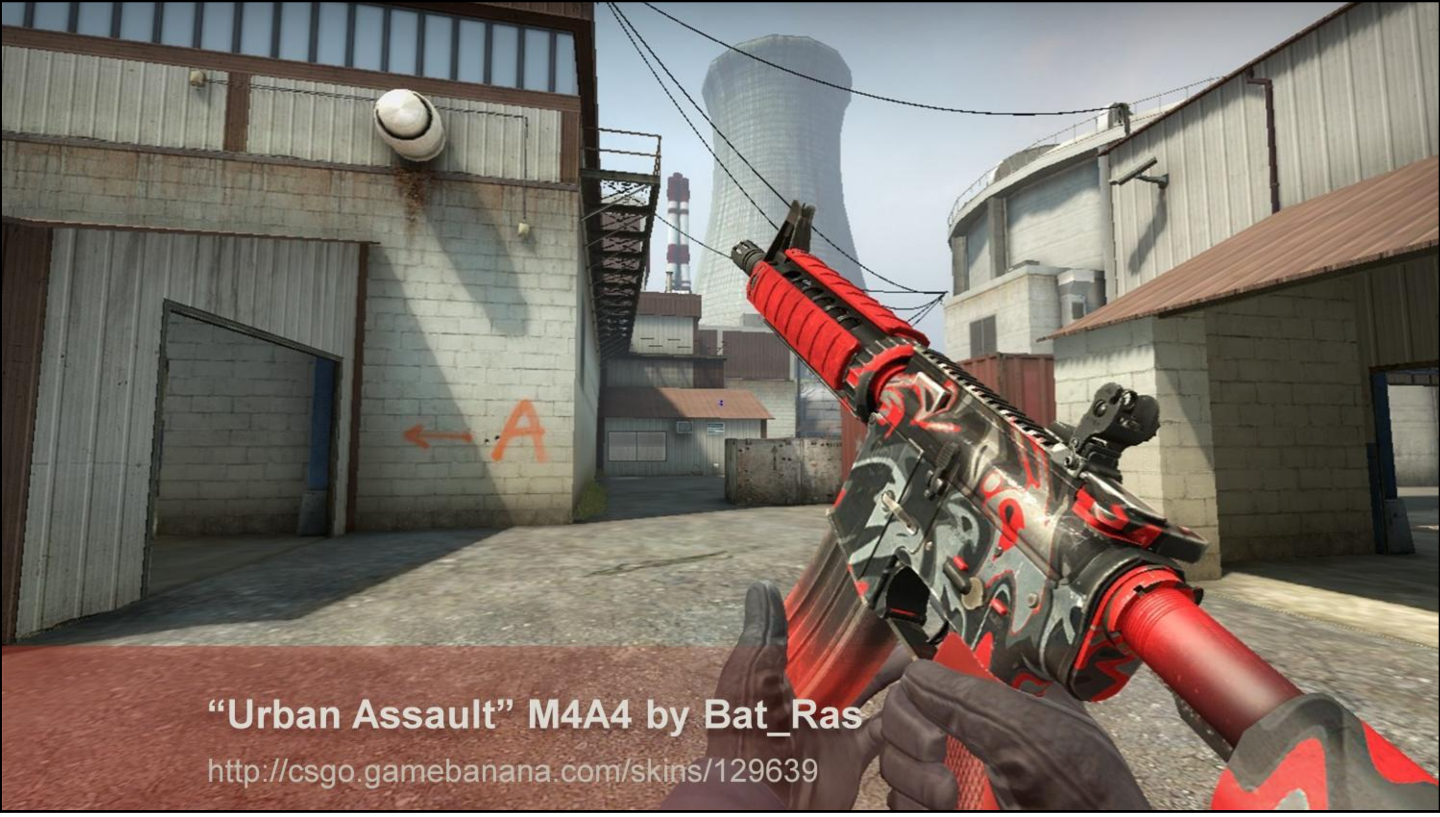
Join the Community Market group for discussions and official announcements

Let's focus on the marketplace. This is the Steam Community Market, just one of the channels we have available for distributing items. Customers can list their items on the marketplace to be sold for Steam wallet funds. The transactions are in real currency amounts and are set by the lister. Identical items can be listed for different amounts, so there's a constant downward price pressure until supply and demand at a given price point equalize.

As a company, we had a unique opportunity. This was the first time we'd launched an economy since the market went live. We decided that *CS:GO* would become our first game in which every item can be placed on the marketplace – and individual items can't be bought any other way. The only thing we planned to sell directly were the keys used to open our crate drops.

In this proposed market-centered economy, the supply of items comes from in-game drops only, and is driven by our customers. The more people we have playing, the larger the supply of item drops. Demand is also driven by our customers. They'll only purchase items they feel are being sold for a fair price, and with enough listings, the downward price pressure of the free market ensures a fair price gets found.

This sounds like a great way to ask our customers what they value.

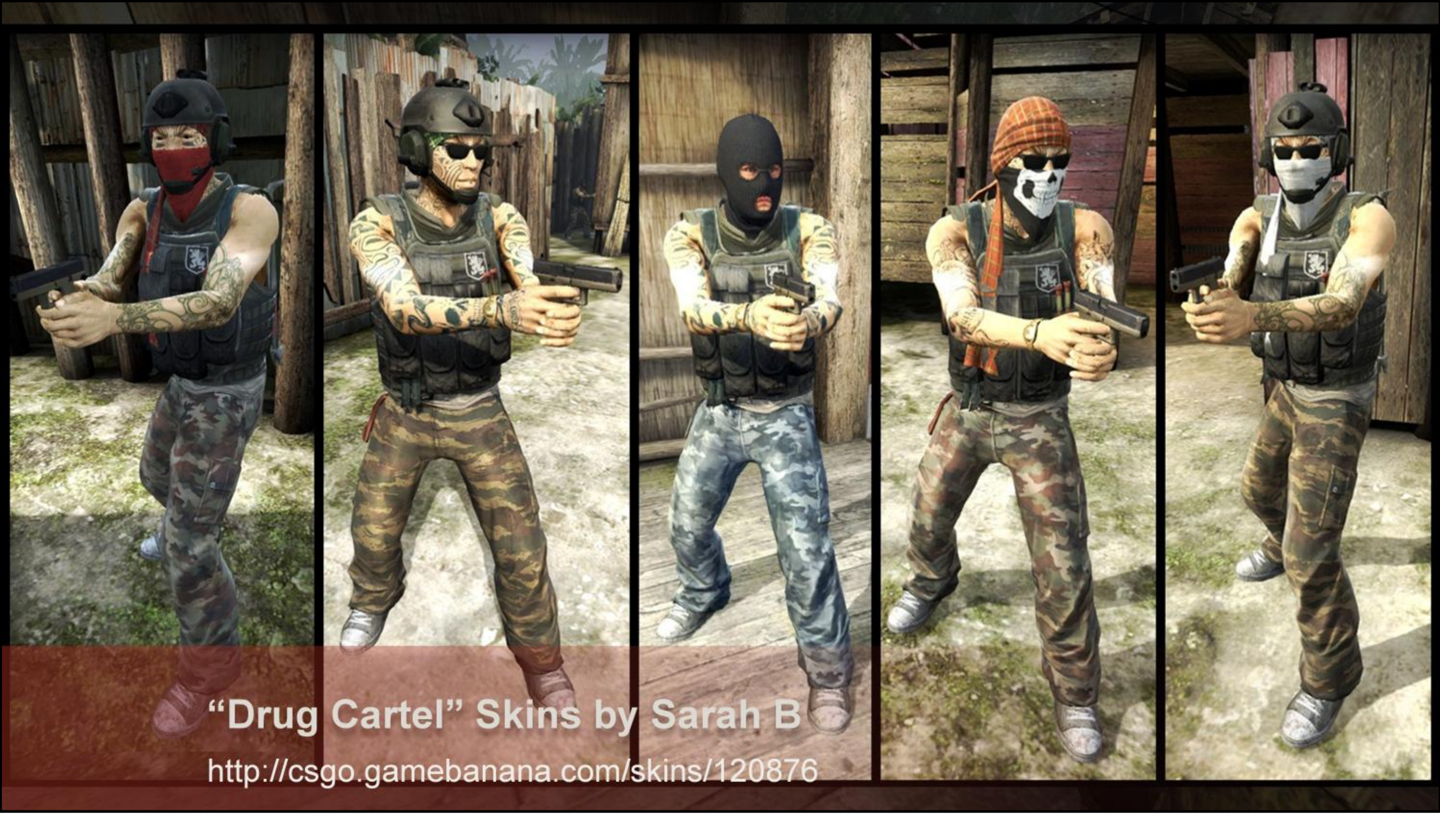


“Urban Assault” M4A4 by Bat_Ras

http://csgo.gamebanana.com/skins/129639

So what kinds of items did we decide to create?

We could create weapon skins,



“Drug Cartel” Skins by Sarah B

http://csgo.gamebanana.com/skins/120876

Character customization…

AKS-74U | Millenia 2012
7903 ▲'s | 2048 diffuse/specular/exponent/normal

AKS-74U by Millenia

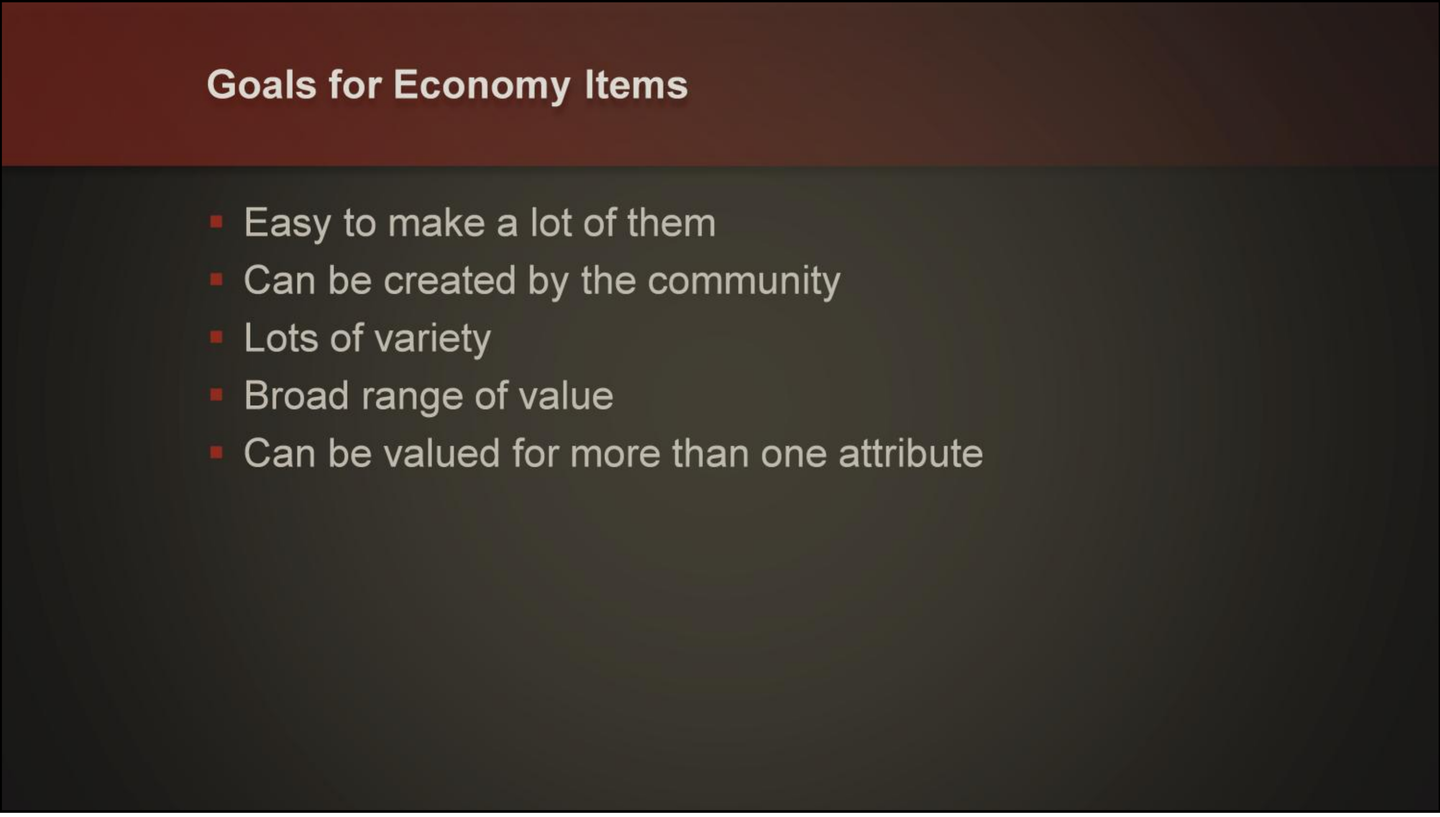
http://csgo.gamebanana.com/skins/120092

New weapons…

Stock M4A4 to SOPMOD M4 by Asteroidrules

http://csgo.gamebanana.com/skins/125515

Or even cosmetic mesh changes to existing weapons. All of those things have been modded by CS players in the past – in fact, all the examples we showed just now are community mods from gamebanana. So there’s clearly a demand for lots of types of customization.



Goals for Economy Items

- Easy to make a lot of them
- Can be created by the community
- Lots of variety
- Broad range of value
- Can be valued for more than one attribute

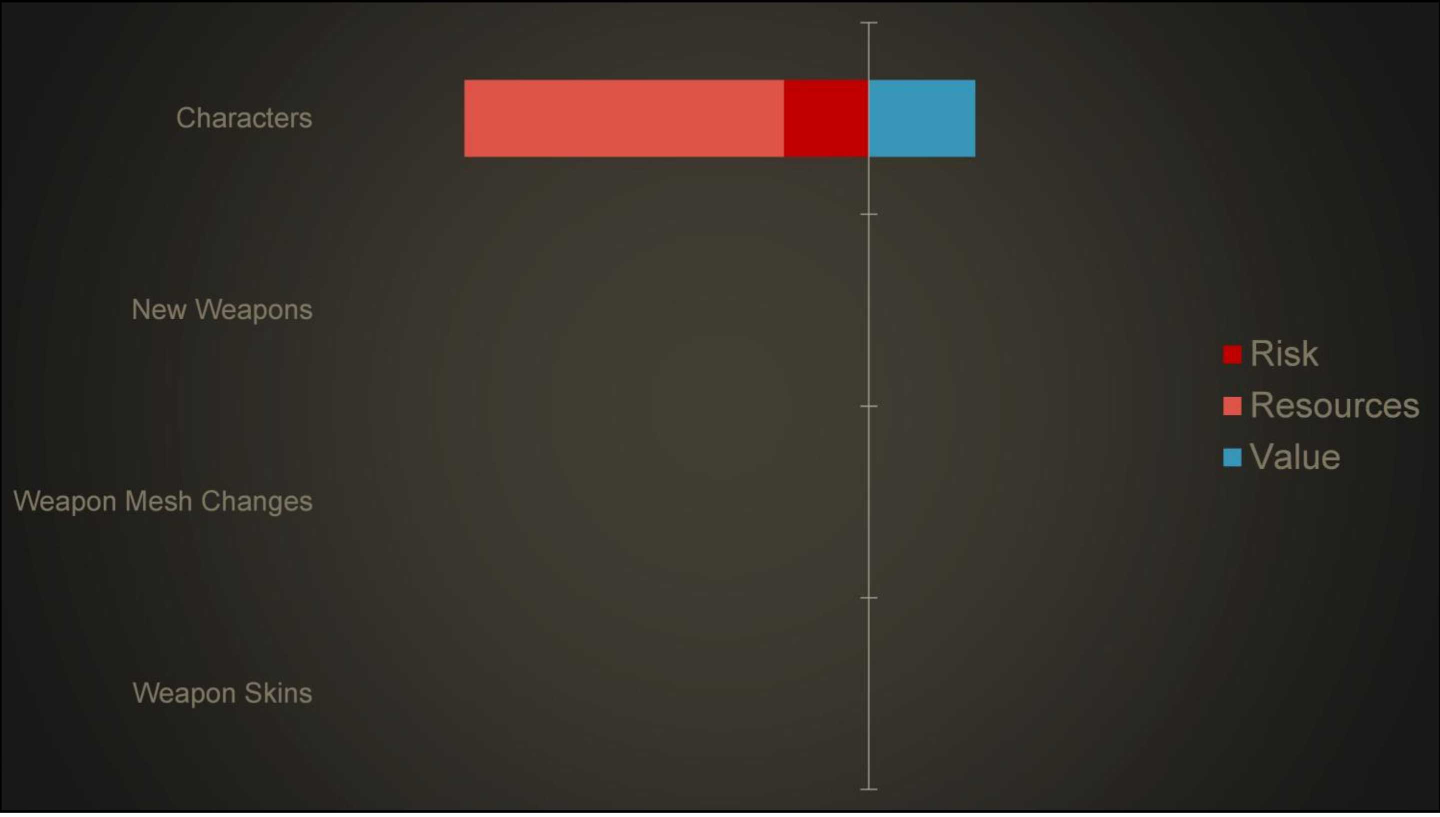
We'd already decided that, whatever we shipped, we needed to create a lot of items. So it's important that we make it as easy as possible to do so. And in keeping with our goal of helping players generate value for each other, we'd like to let community members items and earn real money from their in-game sales.

We also needed a lot of variety in order to figure out what our customers value.

We already know that economies are more fun when there are lots of participants. They're also more fun when there are lots of items to compare and lots of ways to compare them.

So we didn't just want to create items with different values, we also wanted to create items that have value in multiple axes, because different customers might care about different aspects of the same item.

We'll want to keep these goals in mind as we choose which of our four options is the best place to start.

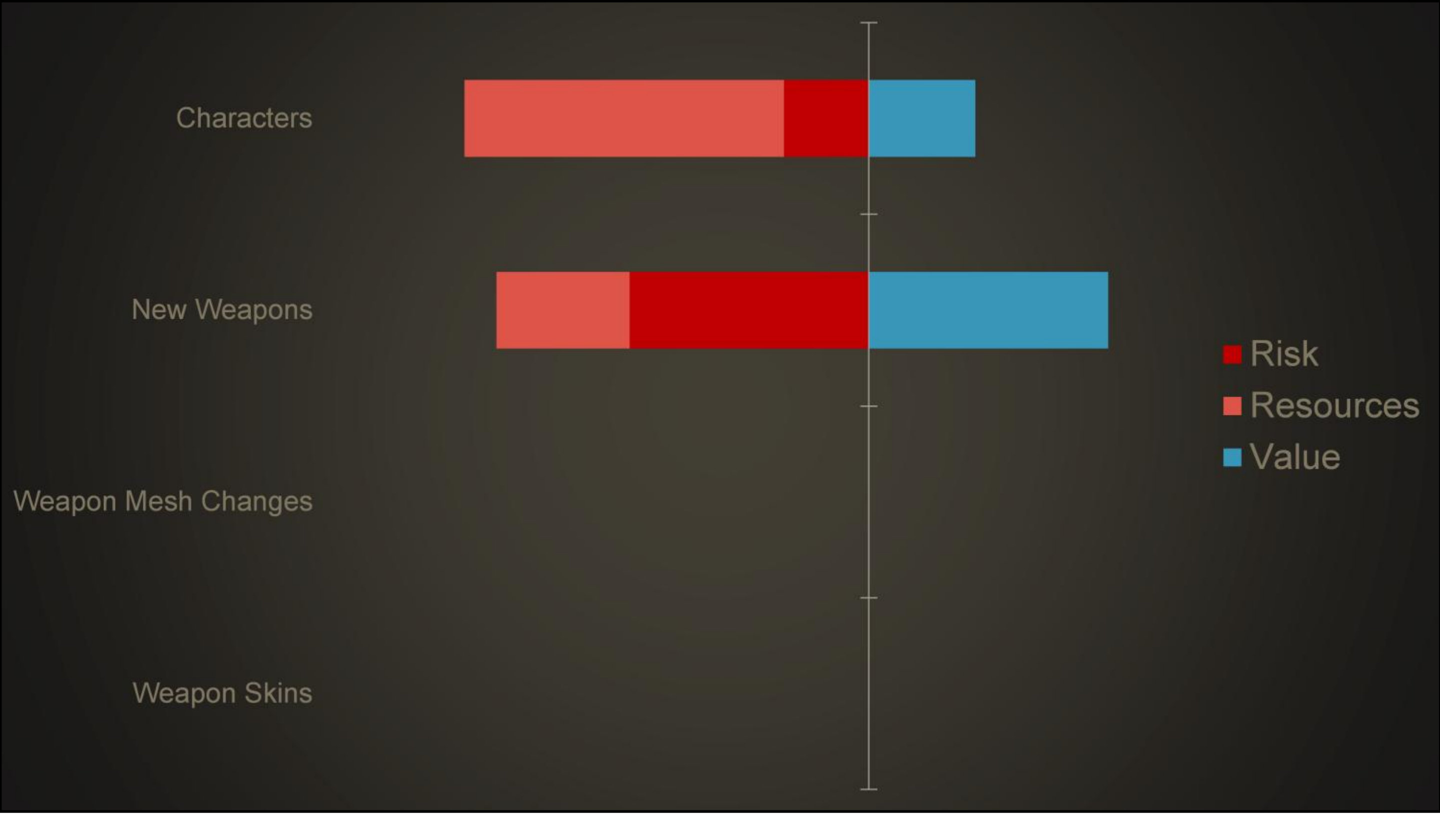


The way we choose is to assess the benefits associated with each type of product, then the risk involved and resources needed. At the time, *CS:GO* was a pretty small team. Our art team consisted of two technical artists, and that's it. We were pretty resource-limited.

So take characters, for example. Customizable characters in *Dota 2* and *TF2* are popular, but they work best when you have lots of chances to see your own character. *Dota 2* is top-down and *TF2* lets you use taunts to see your character from the 3rd person view. Since we're a first-person game with no taunts, the chances that you get to enjoy the customization are fewer than in our other games. So the benefit is perhaps not as high as it first seems.

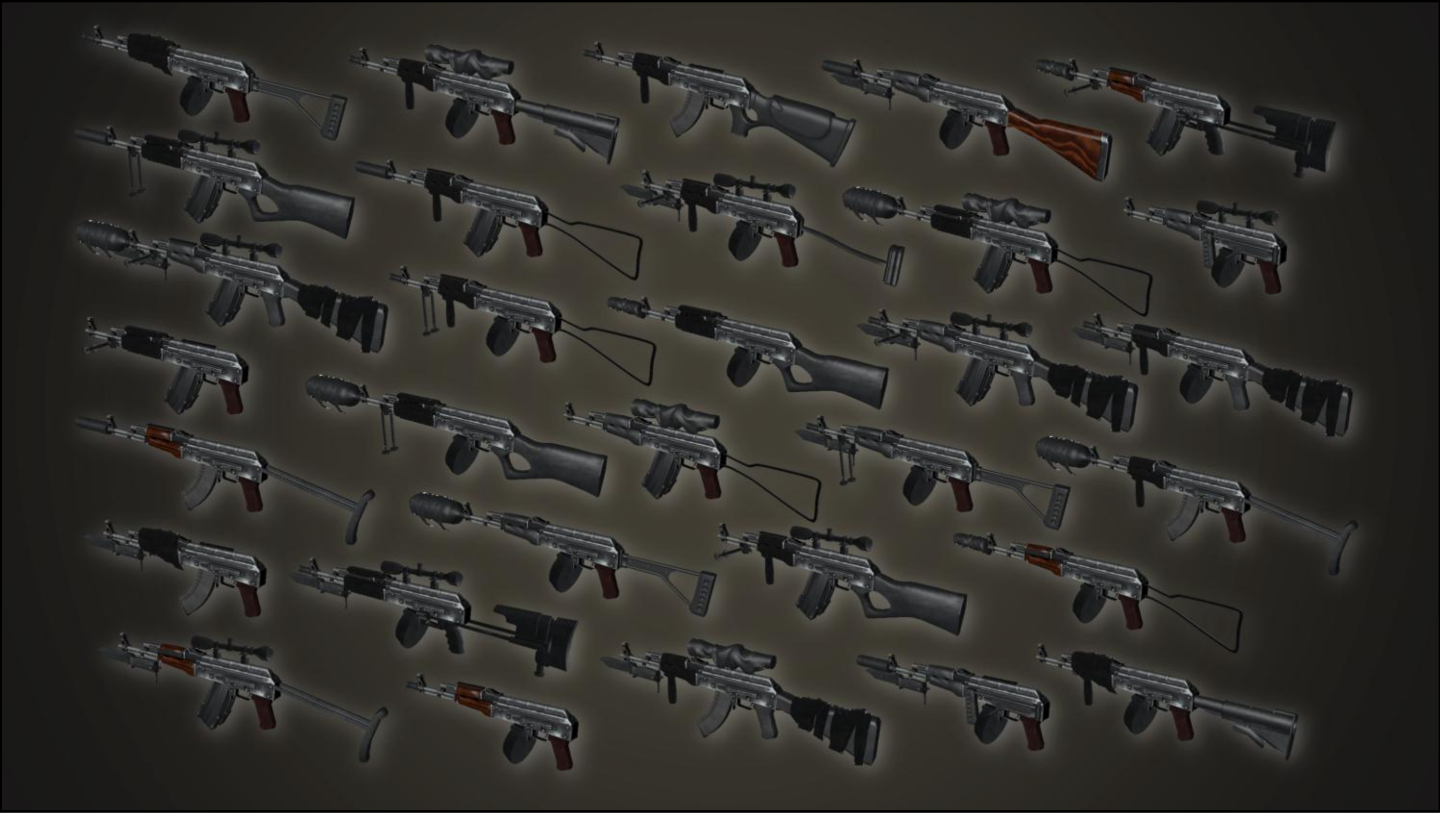
There are also a lot of challenges here and a risk that we would make team ID difficult. So someone with a customized character might be happy to have something unique and cool, but everyone else who plays with them is less happy because they can't tell friend from foe.

There's lots of work to make sure that doesn't happen. On top of that, just making the content is a resource-heavy option, so it'd take a lot of time with our small staff. We don't want to put in a lot of work to something we have doubts about. We'd much rather ship something as early as possible and get data that either validates our decision making or indicates a different direction to go. On the whole, character customization didn't seem like the best place to start.



What about new weapons? Adding some new gameplay options could be really fun. But there's a cost to adding them. Our players practice intensely with their favorite weapons, right down to learning to compensate for recoil patterns. Variety just for the sake of it is not going to convince players to try something new. We'd have to ship only weapons with unique strategic value. We could run out of meaningful variation to ship, probably sooner rather than later. So the value of shipping new weapons actually decreases the more of them we ship.

And each time we ship new weapons, there's a risk of destabilizing competitive play and making our customers so unhappy that they stop playing. Weapon balance is a bunch of work. So it looks like here, too, the risks and costs outweigh the benefits.



Customizing individual weapon components has a fair amount of benefit. For one, you'd see the customization the whole time you were holding the weapon, and so would other players. You'd have a visually unique item. And if you drop your weapon, other players can pick it up and play with it and have a positive experience. You'd make other players happy by having a customized item, creating value for them as well.

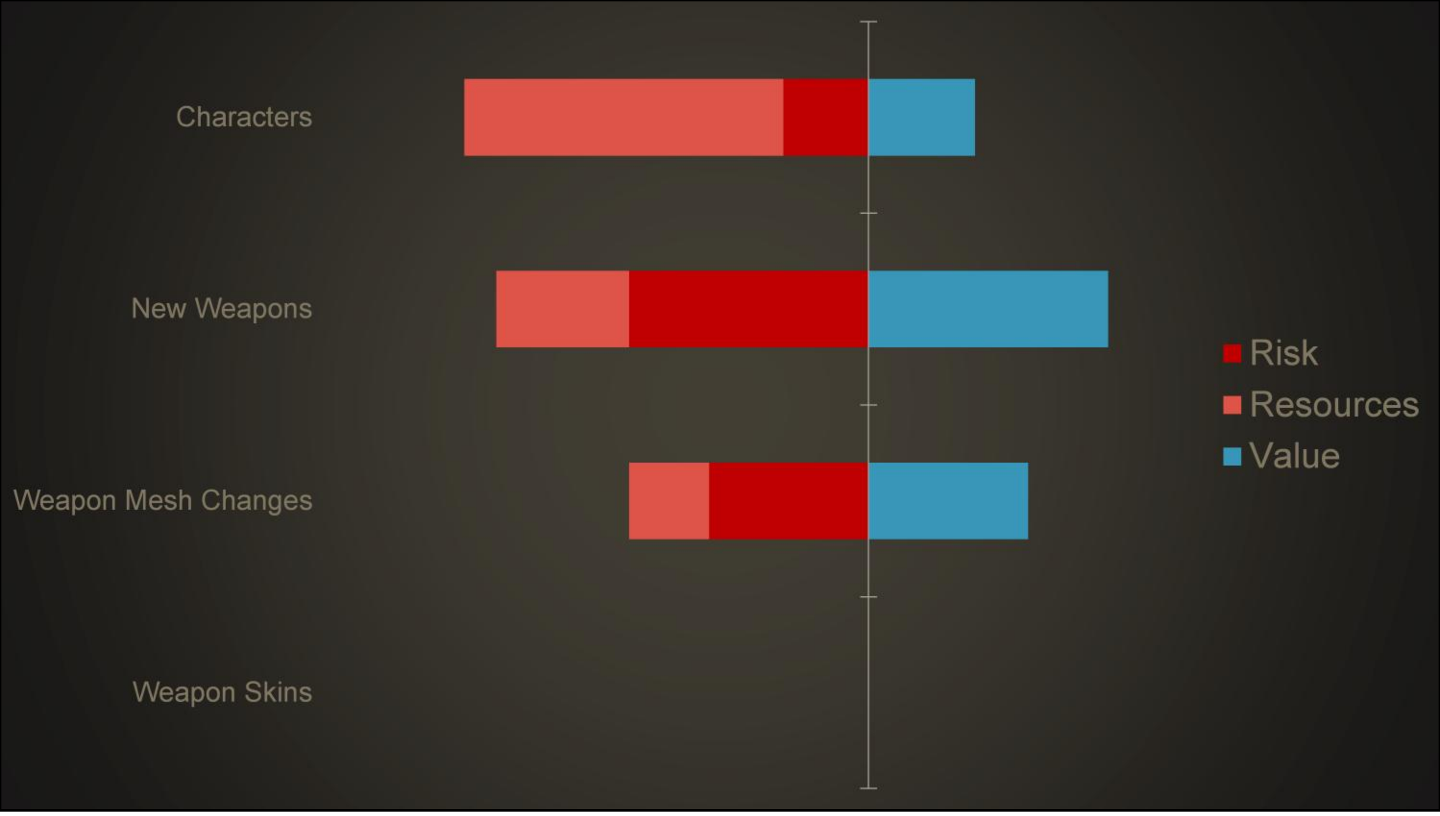
We also deemed it a lot less risky than gameplay changes. Changing out stocks, grips, magazines, adding wrappings – all these could be cosmetic changes only.

We actually got pretty far down this path, taking the AK47 and customizing the heck out of it so that we could try it out.

Through playtesting we found a couple of things:

First, weapon recognition reduced significantly with the different silhouettes. It was really hard to tell just looking at these different versions that all the weapons had the exact same functionality.

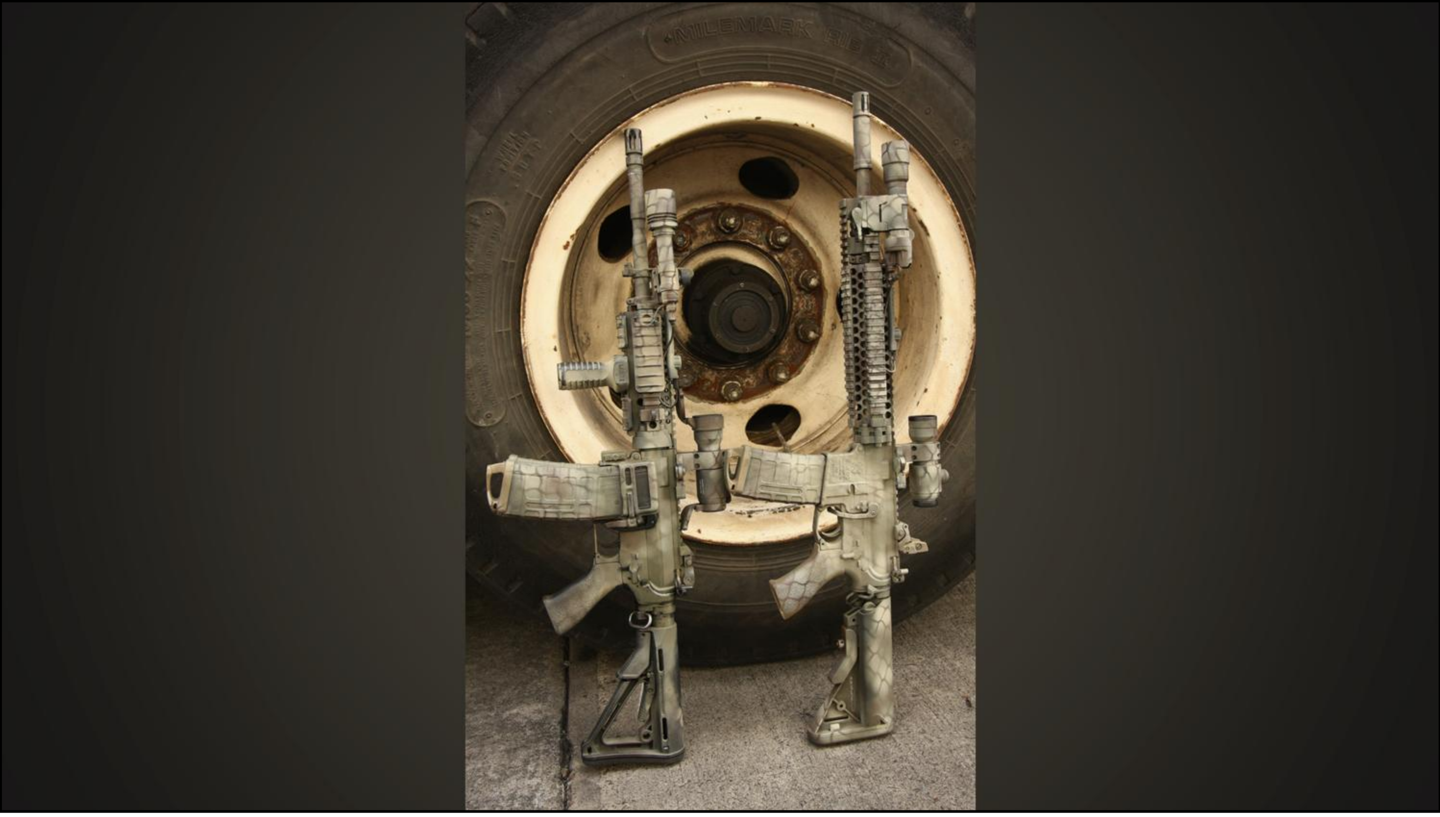
That brings us to the second point, which was that it was more disappointing than we expected when a different component had no effect on the weapon's functionality. For example, there are a bunch of different types of buttstocks and grips here. We'd expect that changing them would affect recoil and accuracy. When they didn't, the experience seemed artificial.



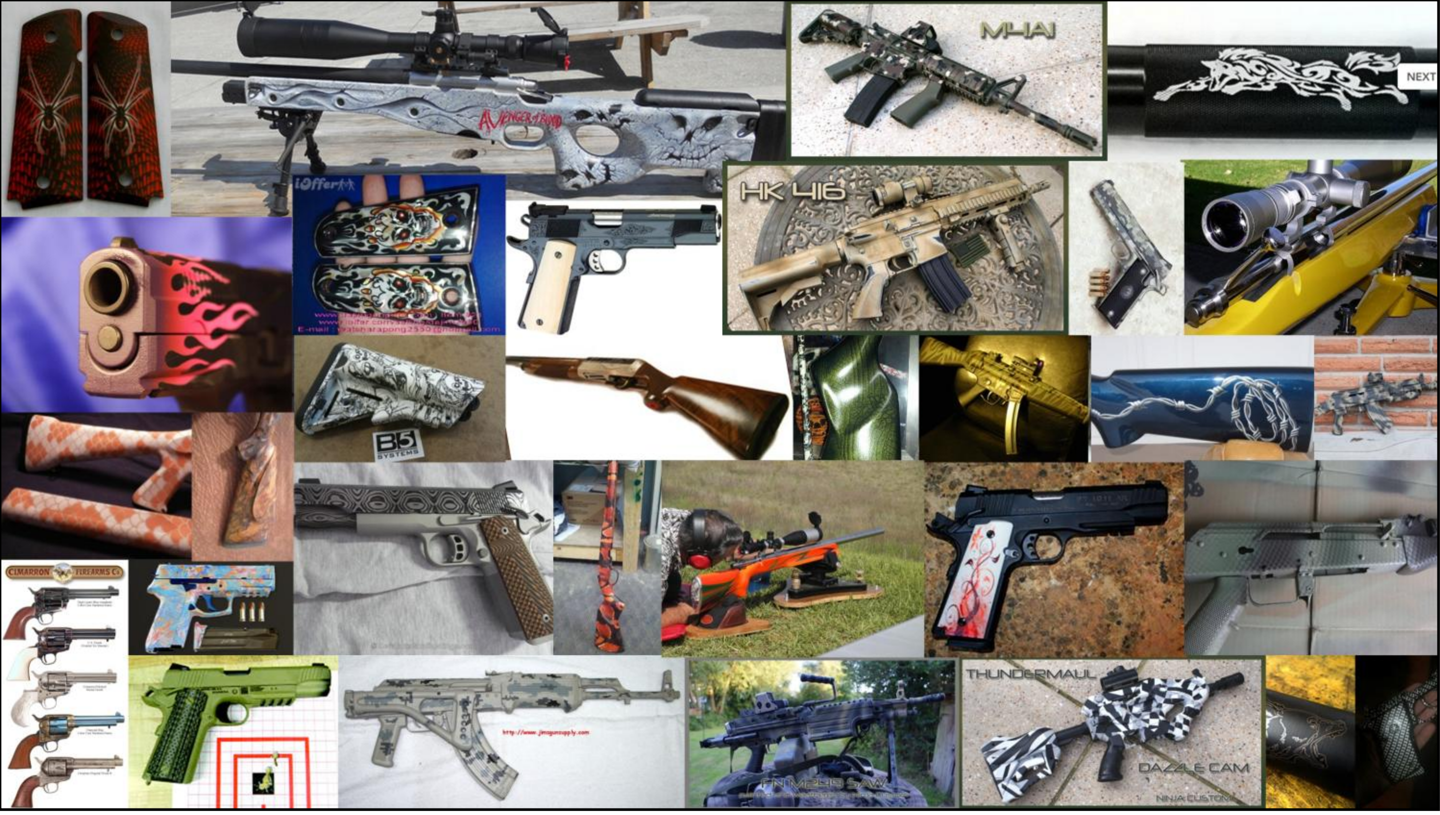
So we reassessed the risk with weapon mesh changes upwards by quite a bit.

It’s also fairly resource-intensive, not only to generate the components and figure out how they can work together, but to design the experience of getting parts and mixing and matching them.

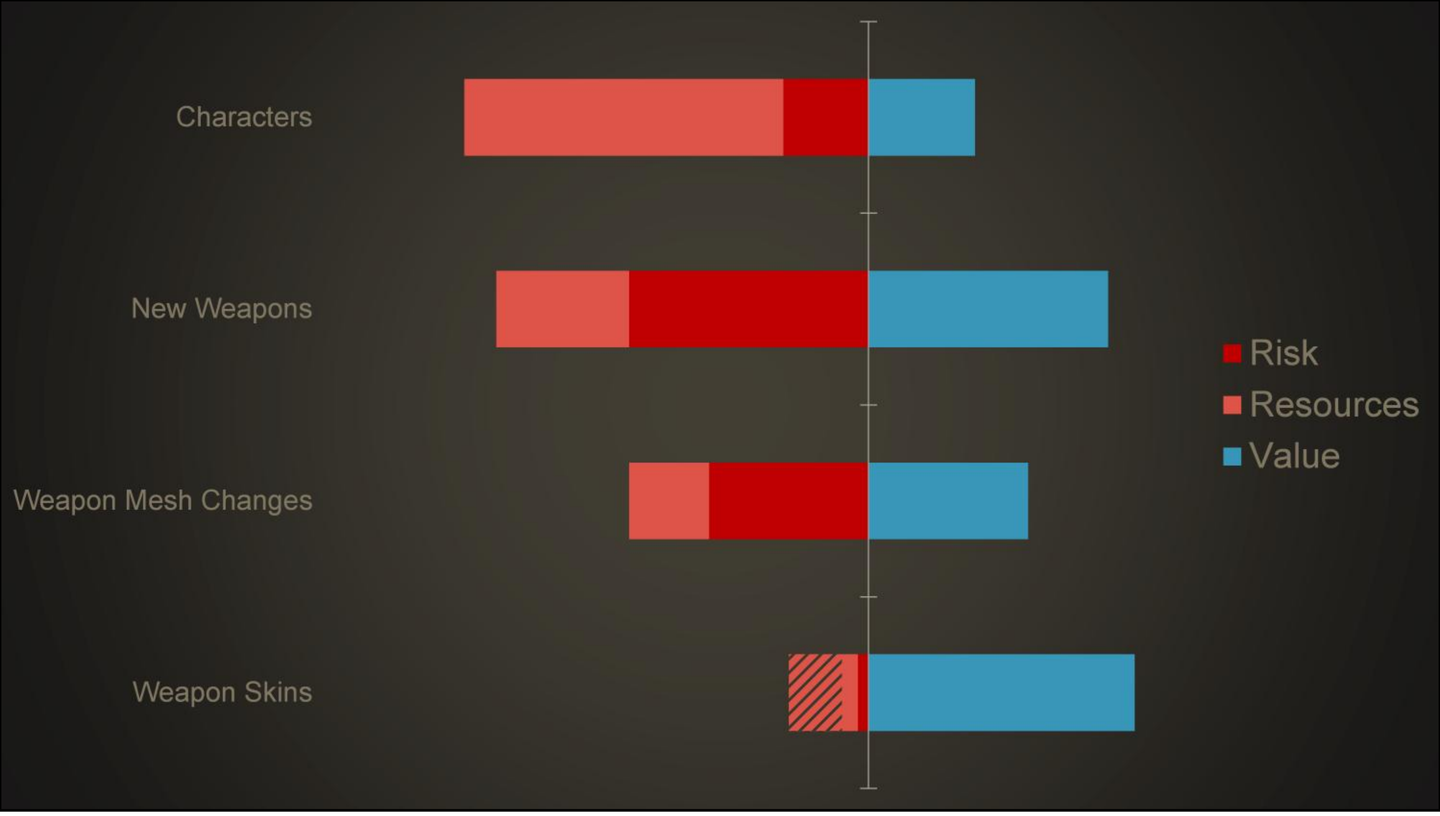
Our last option was weapon skins. At this point we were getting pretty worried.



We definitely thought hand-painted military camouflage was interesting and cool. But how many camo patterns could we make and still have it look interesting?



As we started looking into it, though, it became pretty clear that we could create a huge amount of variation, with everything from graffiti inspired paint jobs to intricate engravings. There's a lot of room for whimsy, much more so than with weapon components, which need to be based on real-world functional items.

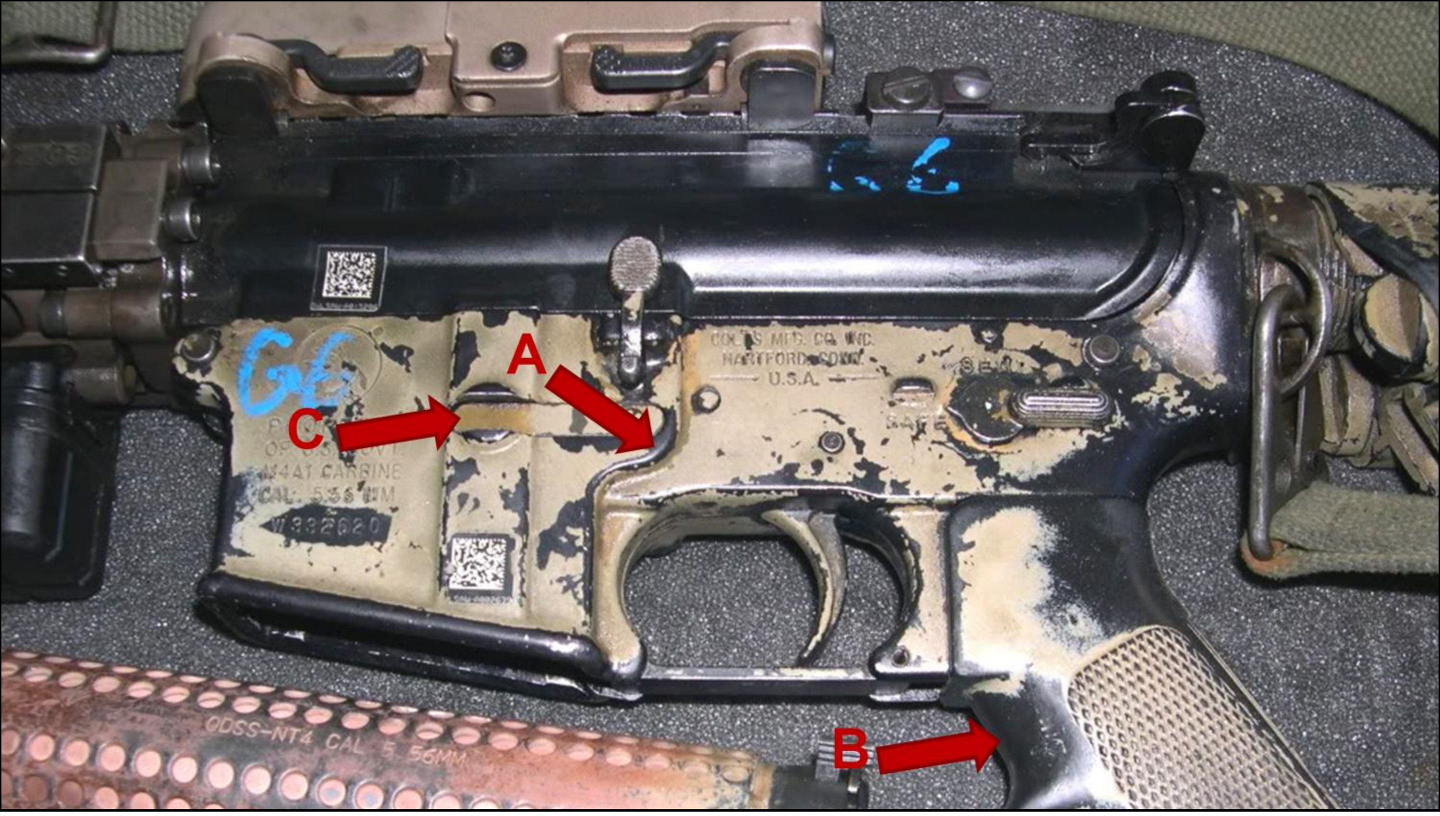


Not only can we create a lot of variety, new textures are also really easy for the community to make. So the benefit here is much higher than we originally estimated. We thought maybe color changes would make weapon identification more difficult for weapons dropped in the world. But mostly, the risk for these was deemed low relative to any of our other options. They're also fairly inexpensive to produce.

That gives us a pretty clear winner.

In the real world, weapon finishes come in a very wide array of styles and colors. So we already had a good indication of how far we could go in creating variety. Now we just needed a way to make a lot of things with our small team.

Since both our artists were technical artists, we did a very typical tech-art thing. Instead of beginning to design how the weapons might look, we began by trying to reduce the amount of work it took to make one.



We started with a basic idea:

Painting detail into textures takes a lot of time. Doing it by hand on a large number of items seemed really inefficient, so we wanted to automate it.

Since *CS:GO*'s art style is fairly realistic, we started by taking a look at real-world reference. Paint wears off metal in a fairly predictable way, so there are certain effects that we wanted to reproduce.

-It chips and scratches off external corners (A)

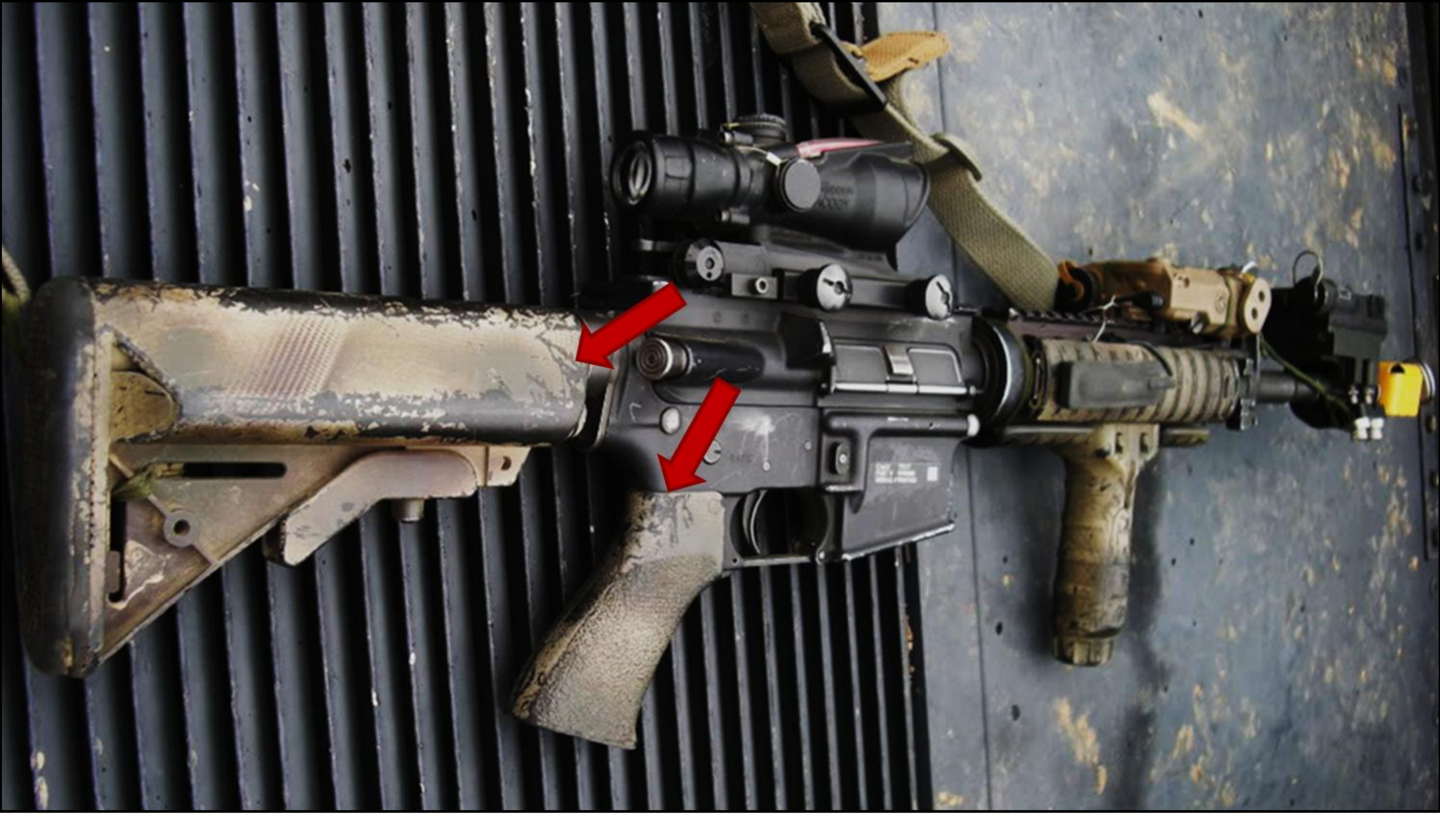
-It wears quickest where it gets the most contact (B)

-Grime and discoloration gather in cavities and corners (C)

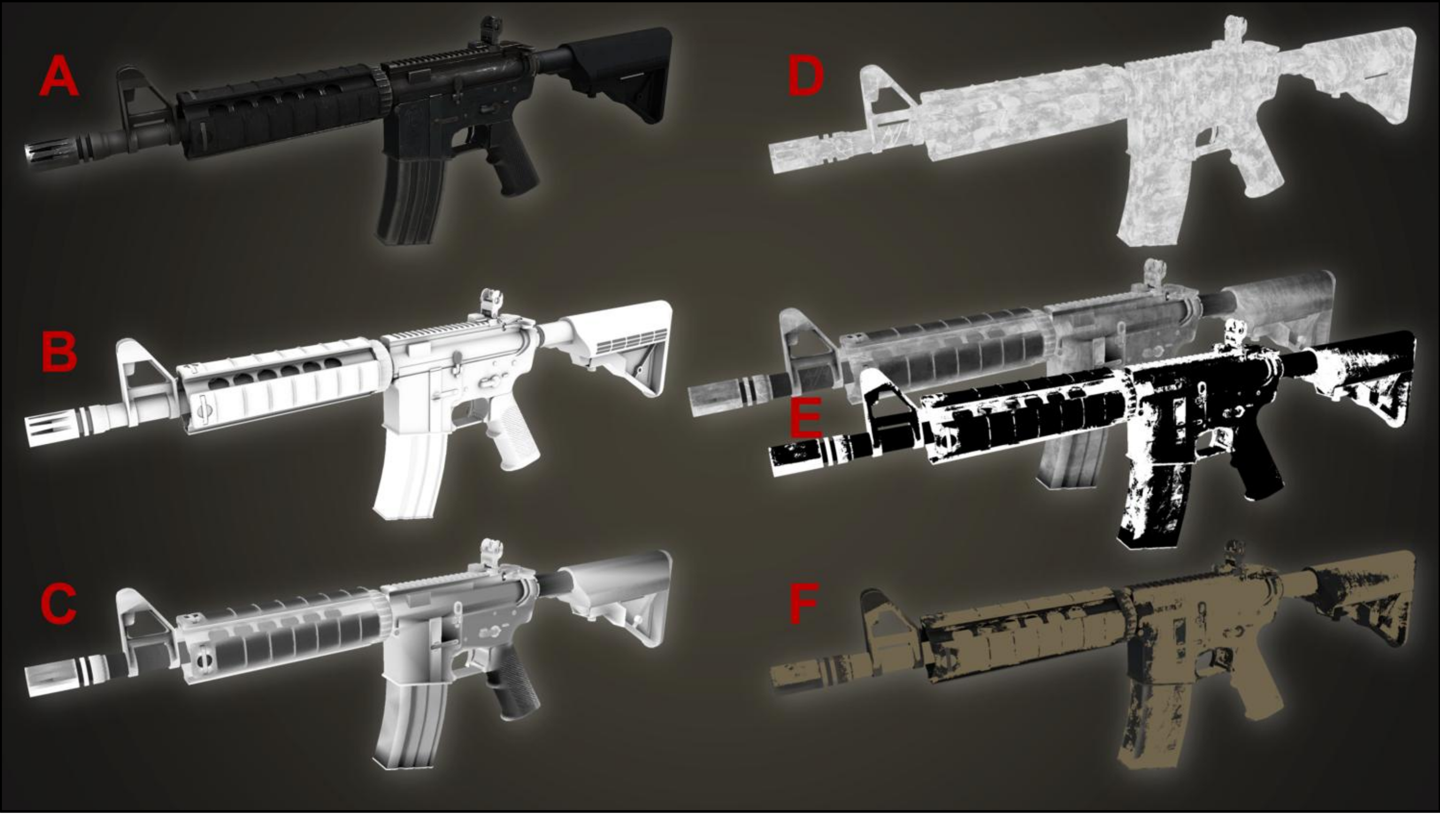
And when people paint weapons, they don't paint the whole weapon. They're particularly careful not to get paint in the mechanism. So each weapon type has its own characteristic areas where the original suface -- the substrate -- is always visible. In this case, only the receiver, grip, and buttstock have been painted.

There's one thing about this example that isn't ideal, though. The paint is a pretty hard, durable paint, so the chips are quite large. This scale of detail looks great on a rifle but wouldn't fit well on a smaller weapon.

And that's an important distinction because we want to ship as general a solution as possible.



In this example, the paint seems less durable and the wear pattern is filled with smaller sized scratches. This scale of detail could apply easily to smaller weapons, so we'll try and target this look.



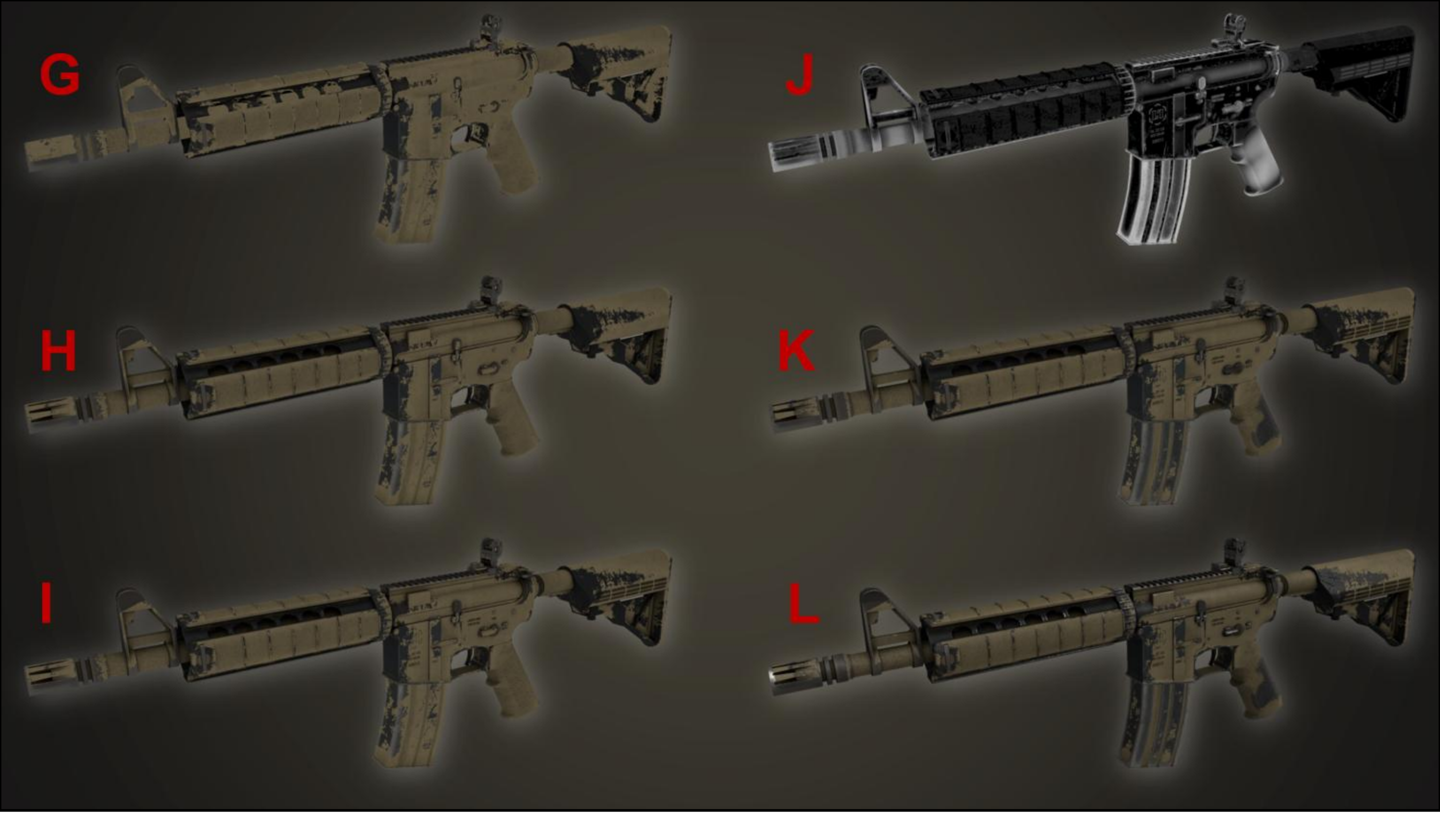
We’ll add finishes over the top of the substrate (A). All the features we named are easy enough to replicate as long as we have some decent information about the topology of the mesh.

The most important things we need to know are occlusion (B) and cavity (C). That lets us see which parts of the weapon have exposed edges that would chip and which parts would gather dirt. That part is pretty simple – we used off-the-shelf tools (3ds max or xNormal) to generate maps for our existing weapons.

For our scratches, we just tile a texture over the entire weapon (D).

We start by just combining the cavity and the scratches (E.1). The brightest areas are the places the paint gets scratched off. This is a really soft blend, so we just threshold using a smoothstep (E.2). For the artists in the audience, smoothstep is a shader operation that is really similar to Photoshop’s Levels operation.

This gives us our blend between the painted areas and the original substrate of the weapon (F).



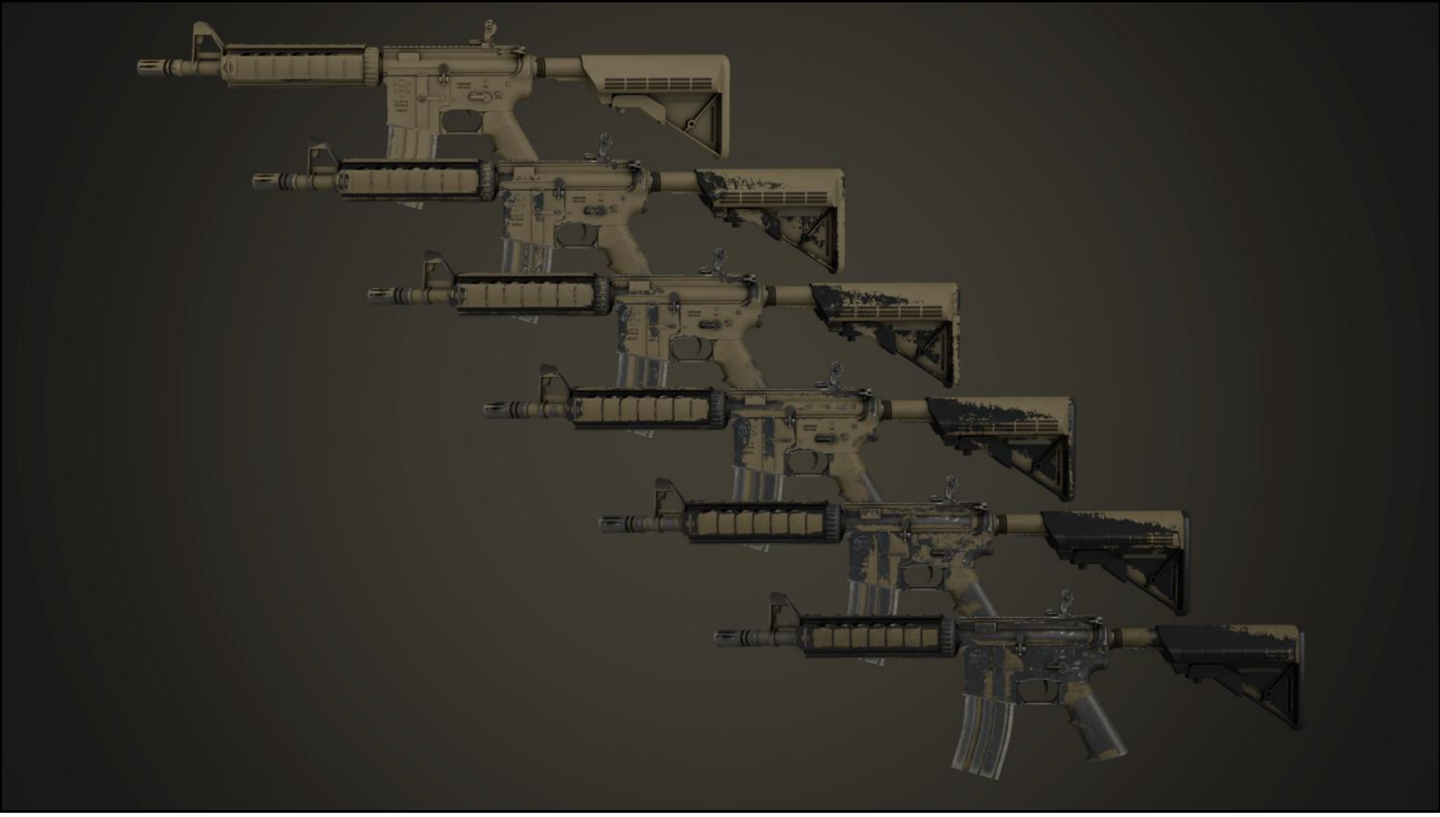
Now’s the time to pull in something else we noticed in our reference: the grime in the cavities. The cavity map is all we need to place the grime properly (G).

Next we’ll apply the ambient occlusion to get the paint to sit in the same world with the substrate (H).

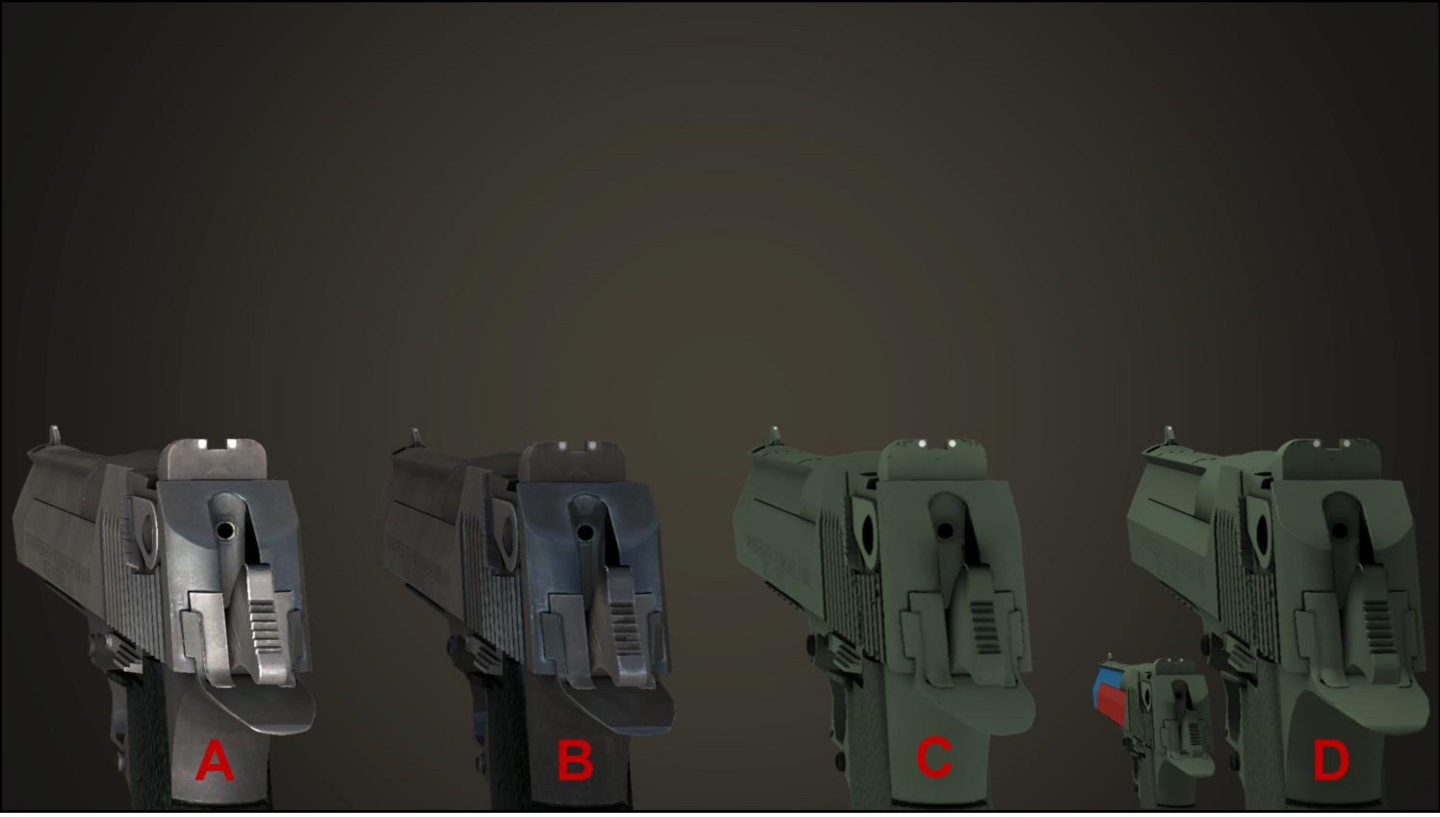
That’s already looking pretty good. There are a couple of problem areas, though. First of all, there are some details like text markings and etchings in the original gun that ought to show even if they’re painted over. We considered a couple of options here, like adding an additional layer that alpha-blended to the paint, but the cheapest and easiest is to simply include the details into the ambient occlusion pass. Since the AO is just multiplied over the top, any text will show up dark over the paint (I).

Another thing we noticed in our reference was that the most frequently touched areas wore more quickly. We can’t get that information from the topology, since it’s a result of human interaction. We probably want some additional artistic control over the paintable areas on the gun too, since not everything should be painted on every weapon. So we added an additional mask that allows you to paint wear influence. Since we’re lazy technical artists, we grabbed the opacities from the detail layers of the original texture PSDs, then painted some blobby details over the grip and magazine with large fuzzy brushes (J). Truthfully we spent very little time adjusting this mask. And here’s the result of the influence – you can see particularly the wear on the grip (K).

Let’s add lighting. At this point we’d hand-authored almost no new content, with the exception of the scratch mask and grunge texture which will be re-used across all weapons. And we’ve got a pretty credible result (L). The fun part is that since the wear is a shader input, we can animate it and see the wear crawl over the surface.



This is great, because it shows we can already get a variety of looks with just a single shader parameter.



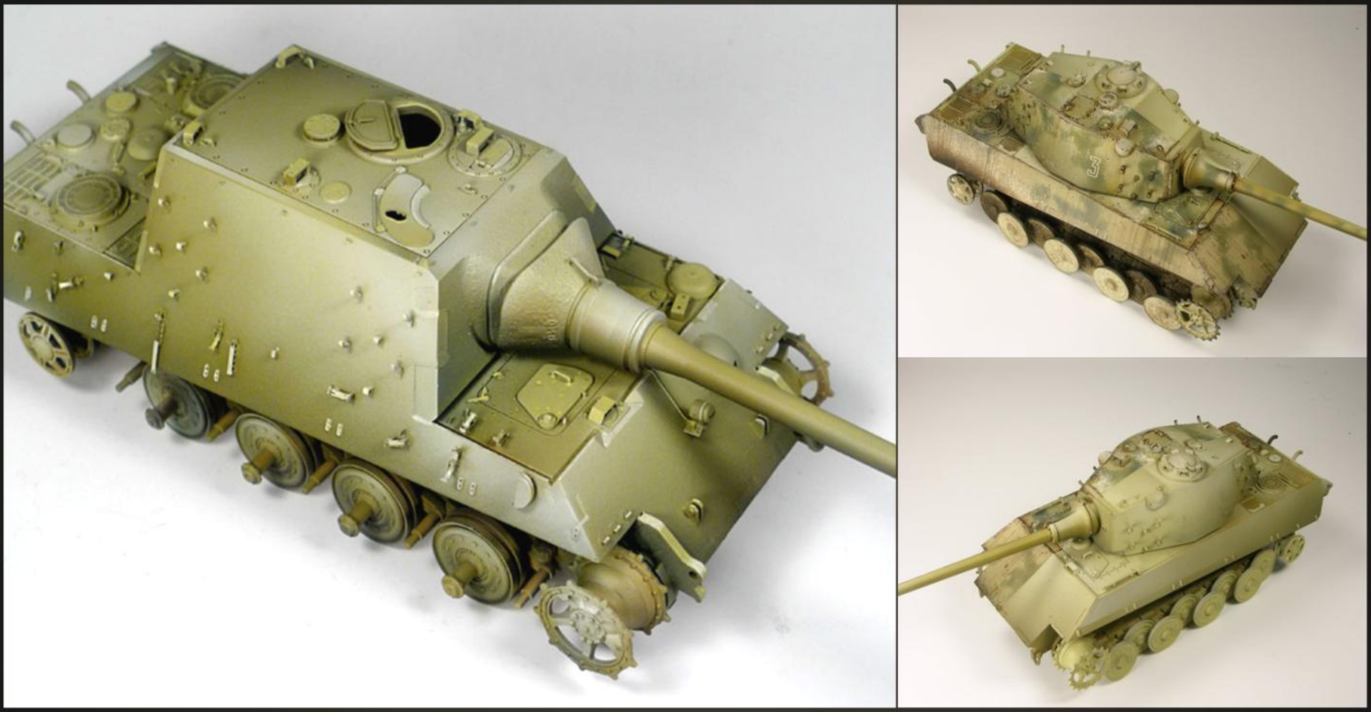
It was around this time that we started testing the finishes in-game under real game lighting. What we found was a bit concerning.

Here’s a typical view of a weapon. We’ve hidden the world, the view model arms, all that stuff so that we can just focus on the weapon.

You can see all the plane changes in the model (A), even when directional lighting is disabled (B), because the texture is authored specifically for this weapon to call out plane changes and edges.

But without directional lighting, when we apply a paint, all those plane changes disappear and the model looks oddly flat (C).

Even with directional lighting, planes with the same normal but different spatial offsets can’t be differentiated (D). For example, these elements in red and blue (D insert) are separate, but with the uniform paint color applied, they look as though they’ve been merged into one.

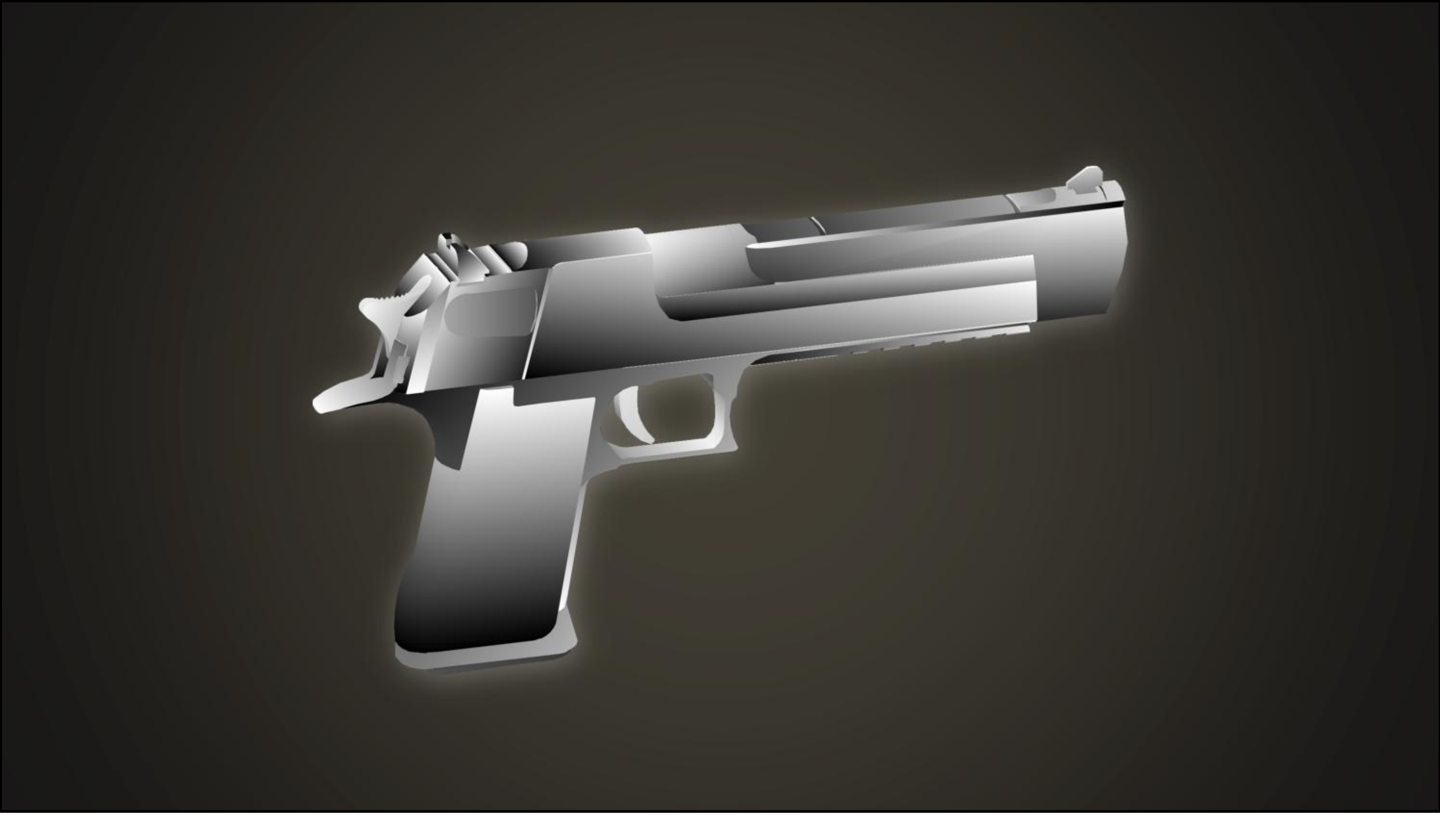


http://migjimenez.blogspot.com/

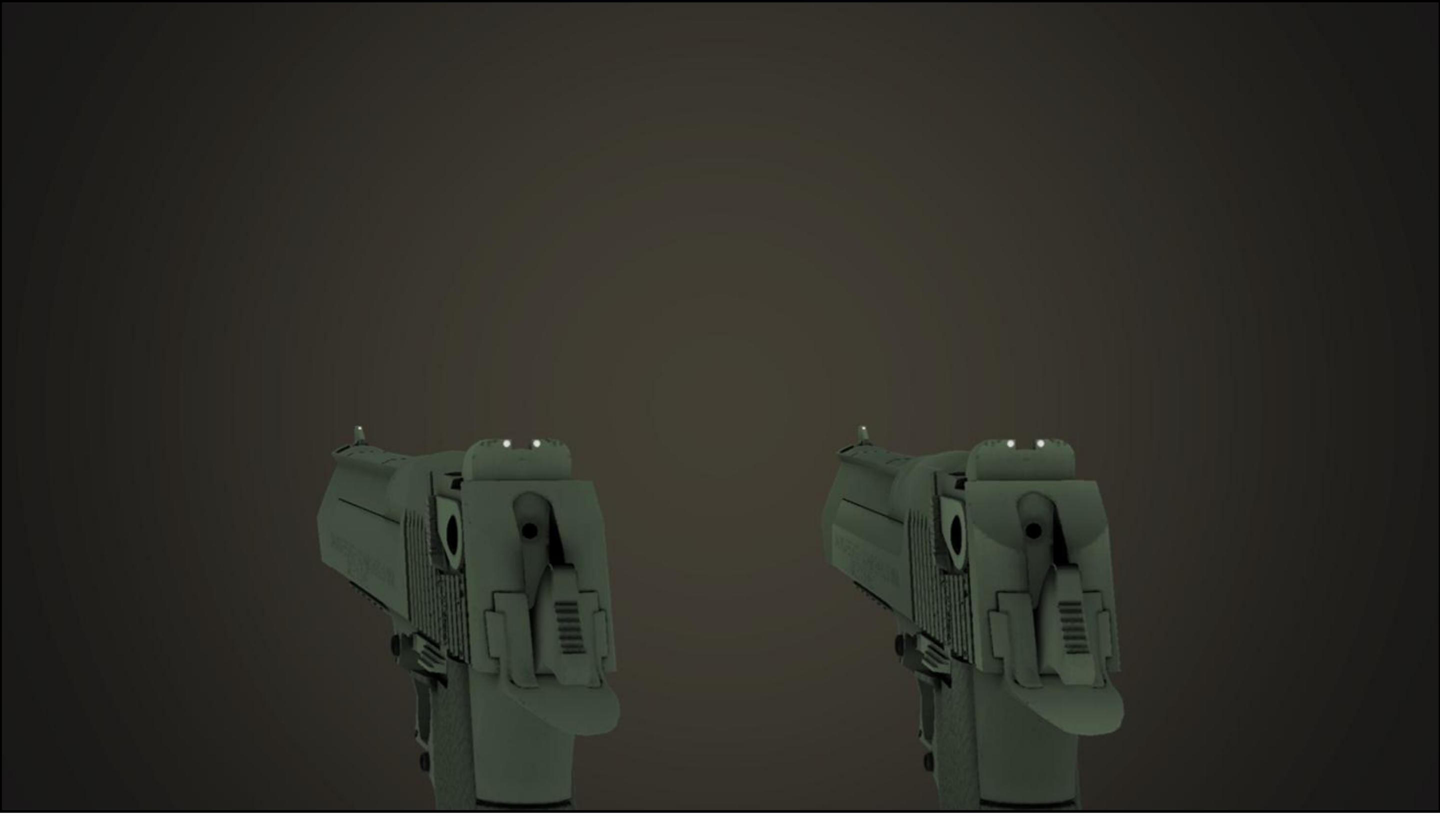
One of our concept artists is a model making nerd, and as we were figuring out the paint system he recommended we check out Mig Jimenez’s blog on model painting.

Mig Jimenez uses a technique called modulation that gets painted on as part of the base coat before all the details are added. The effect calls out geometric details and enhances plane breaks. It’s not quite lighting, though some decisions about the direction of a particular gradient might be influenced by lighting. Instead, you can see that the gradients are each oriented to create contrast at edges that helps the viewer read the form.

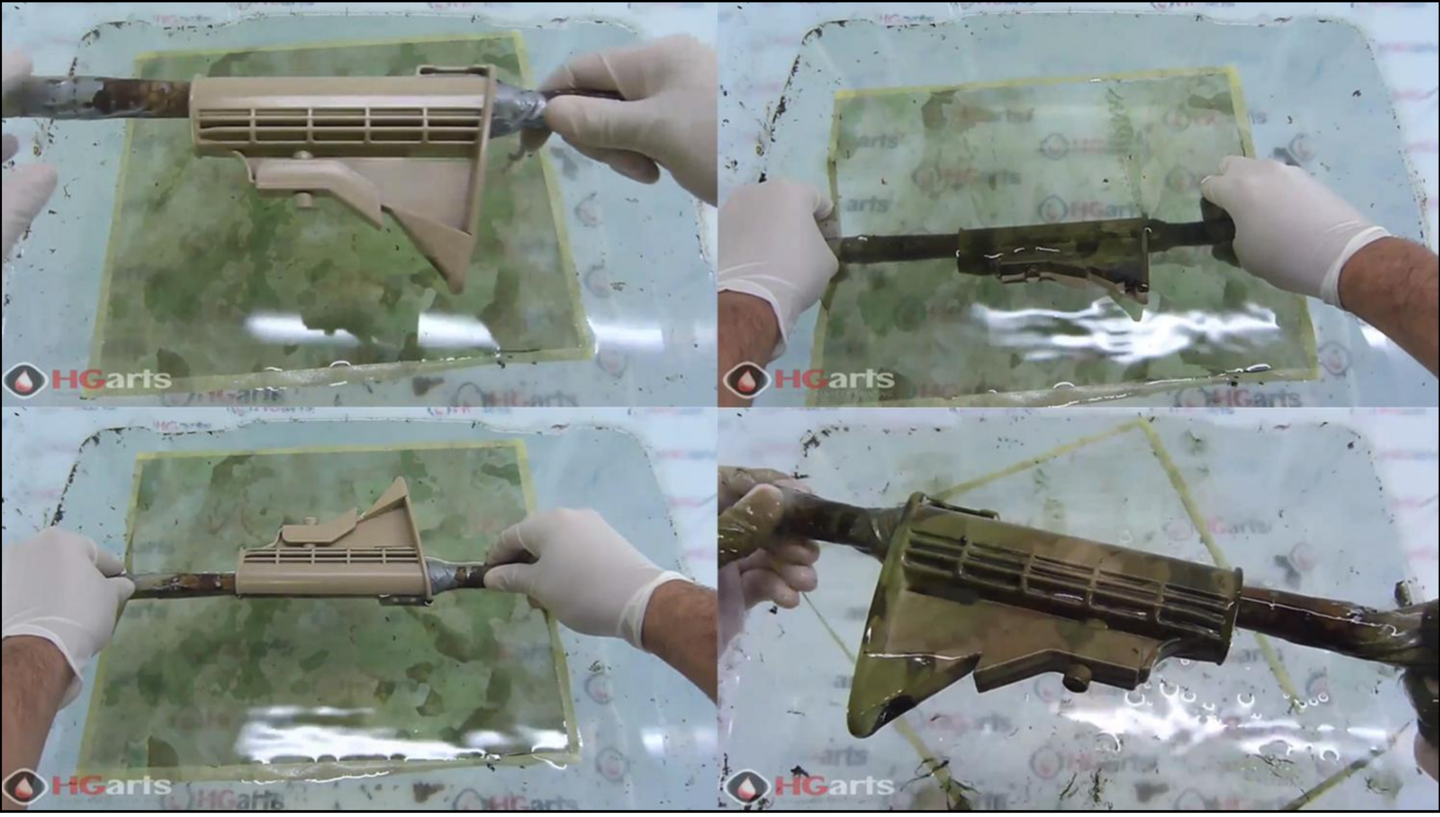
The details are painted afterwards and they reduce the obviousness of the effect. But the improvement in the viewer’s understanding of the form remains.



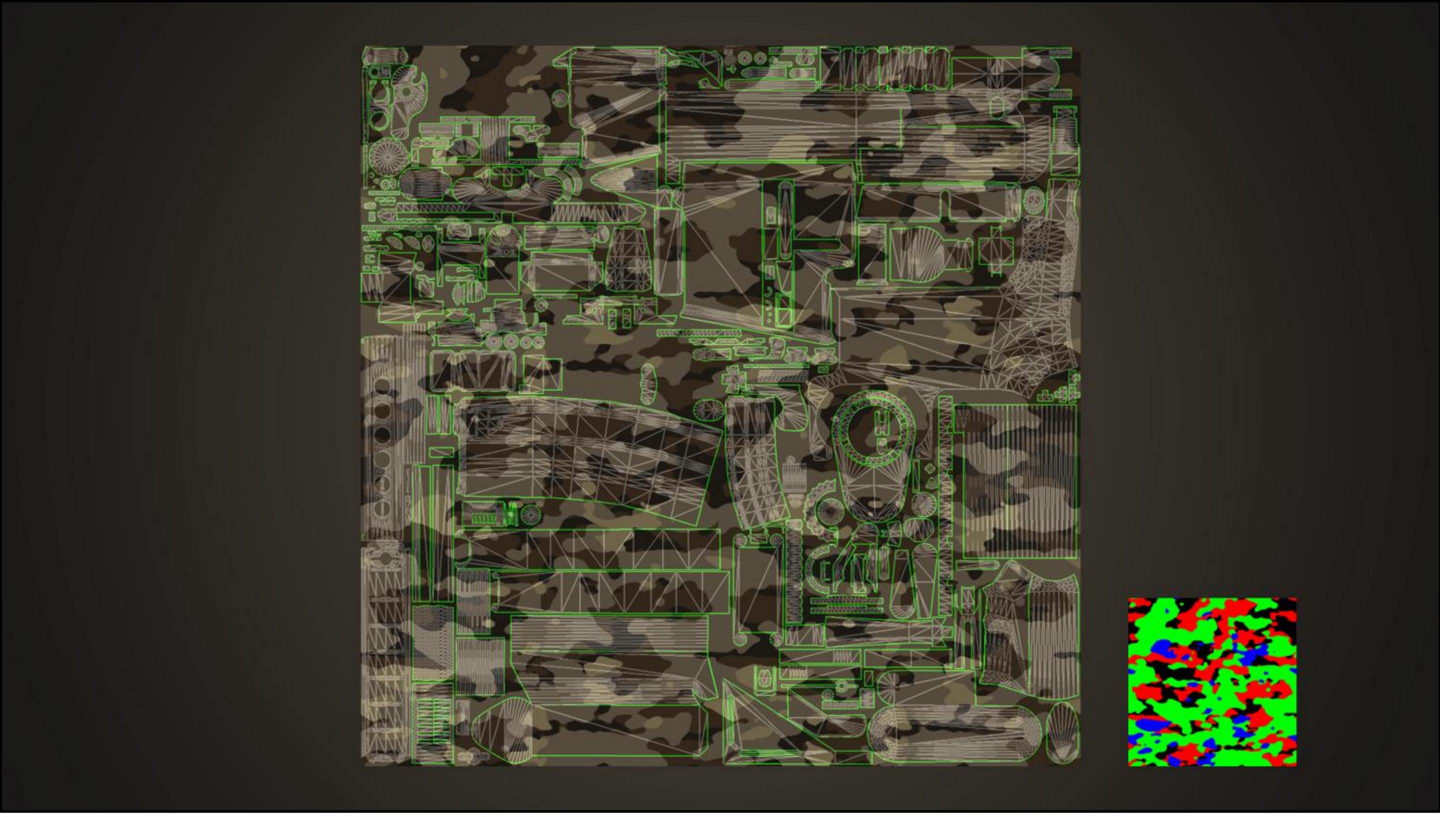
We tried it out. Truthfully, this was not very fun. The basic process was just to grab an element, planar map a gradient onto it, then rotate and mess with the UVs until the gradient was falling the way we wanted. Although the average time spent on a weapon is two hours, some of the more complex models took nearly two days. But the result…



Well, it works! Adding in the modulation shows us our plane changes again. We’ve exaggerated the effect here for visibility – we wouldn’t apply it at this strength.



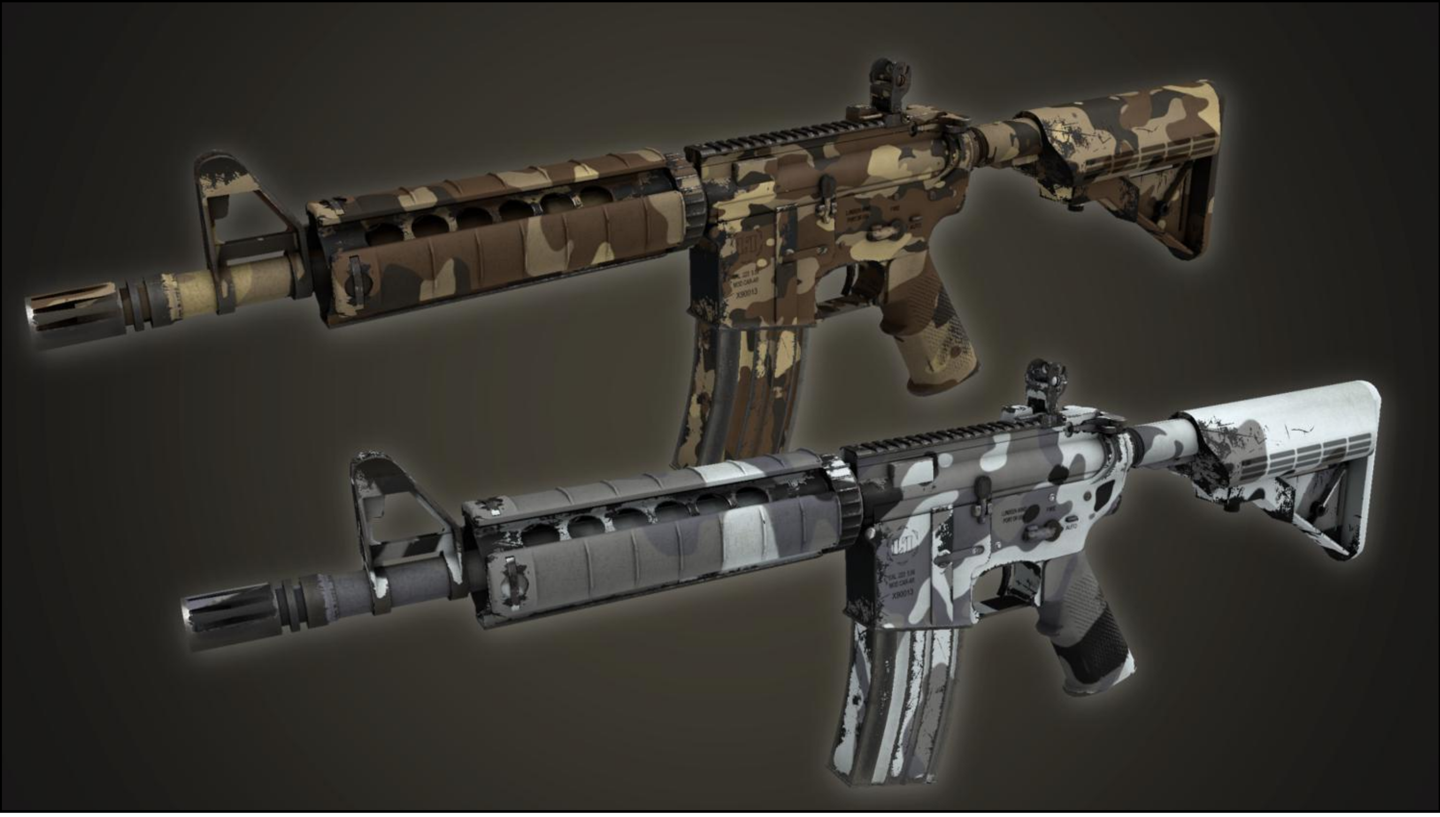
That’s a pretty promising start, but we don’t want to paint each weapon with a single color. We researched a bunch of different finishing techniques and decided our first test would be inspired by hydrodipping. This technique uses a film with a printed pattern on it. You float it in a tank of water then spray it with an activator that dissolves the film but leaves the ink, then just roll the primed component through the ink to pick it up.



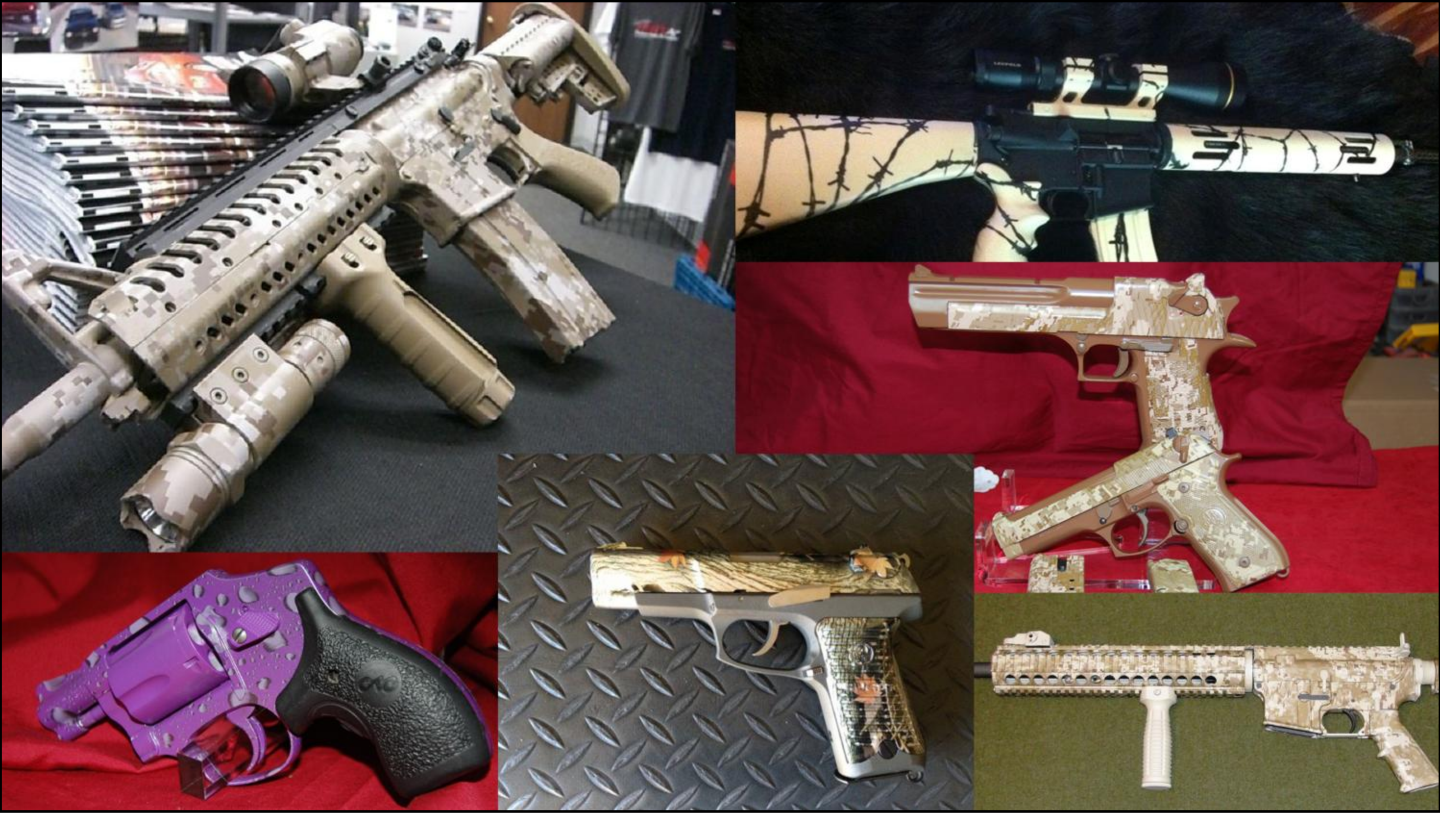
We imitate this in a super cheap way. Since hydrographics have seams from being dipped from multiple angles, and the since the pieces of a weapon are disassembled before dipping, we just use the original unwrap of the weapon and apply it in UV space. Seams in the result? Yeah, we just dialed up the realism.

You’d think the first thing we’d try would be to replace the flat color from our prototype with a texture, but it’s so… single-use. And looking down the line, that didn’t seem like a good first investment, because we’d still have to create a lot of different images.

In order to be more efficient, we decided to try procedural coloration of patterns. We authored a few test patterns, making them so that each channel could be interpreted as a different mask for a solid color. We’re showing you a pattern here really small because it’s blinding. Seriously, authoring these is really hard on the eyes.

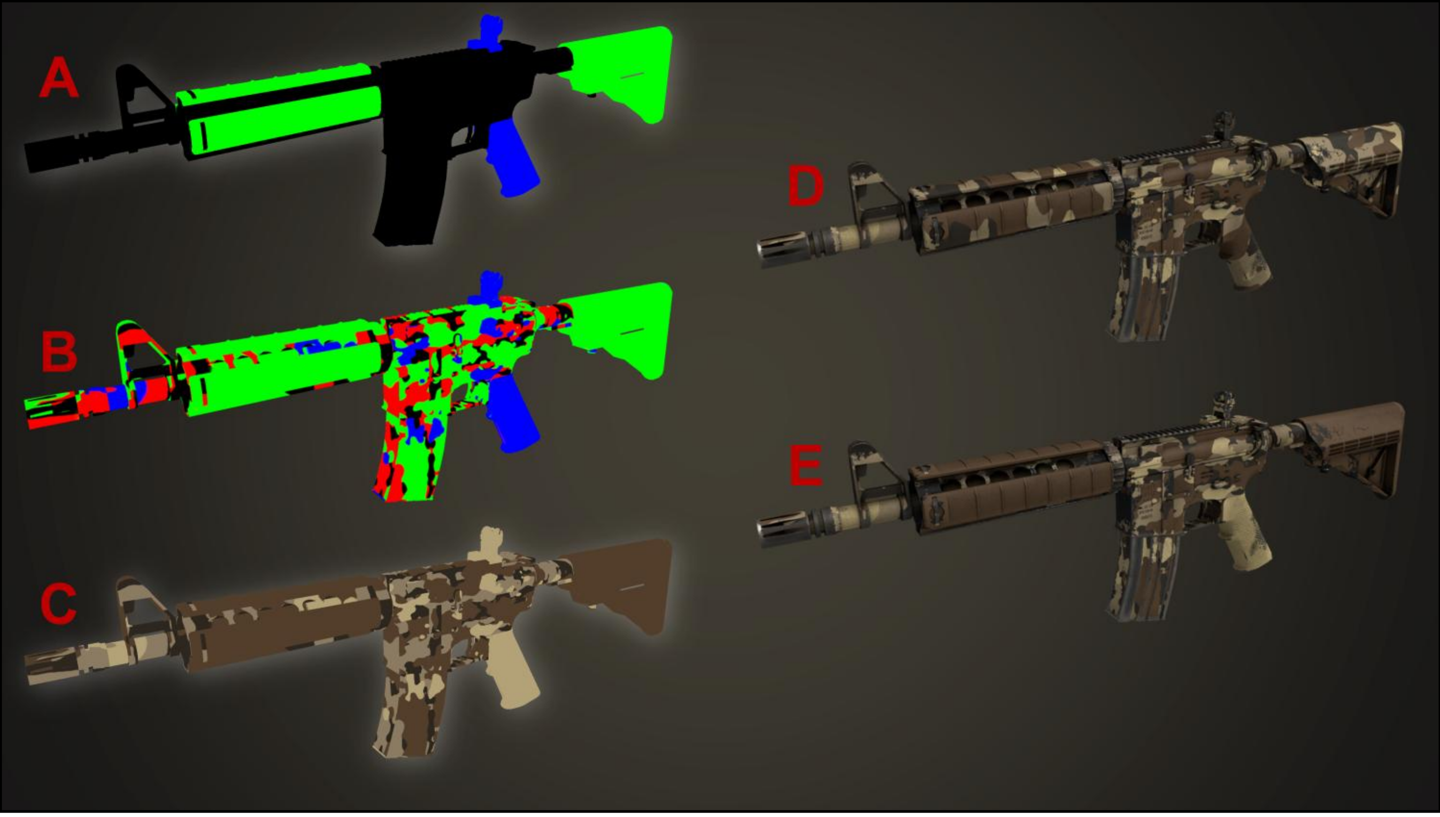


Once we have a pattern, though, we can colorize it different ways. And since it’s an extra texture lookup in the shader anyway, we can transform the UVs just for that lookup to change scale, rotation, and translation. As a result we can get some interesting variation just by tweaking shader parameters.



We looked examples at of this technique and found that, in a lot of cases, the person applying the hydrographic didn’t dip all the components. Instead parts are often painted in solid colors that are complimentary to the hydrographic.

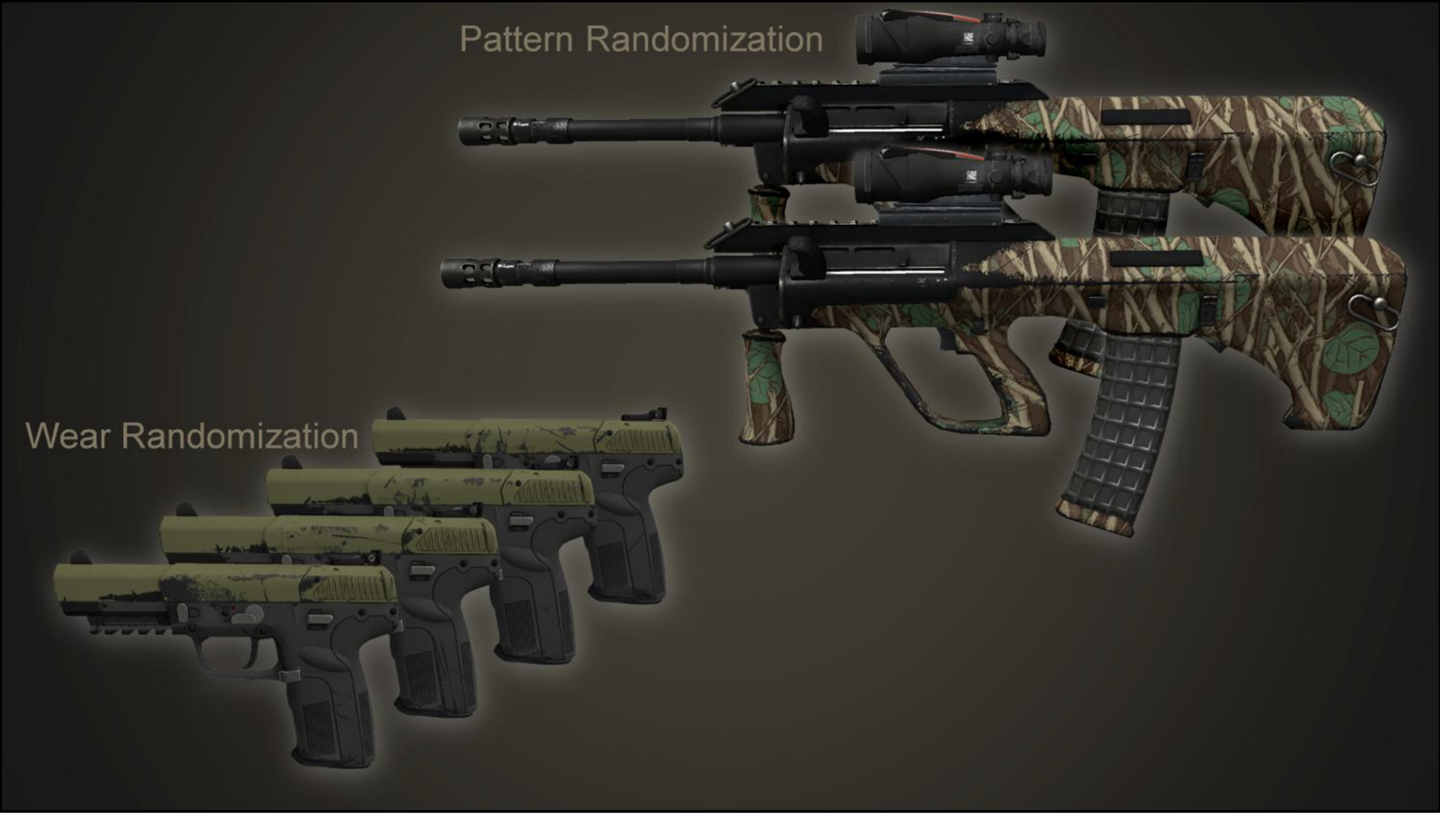
We found this look more appealing than head-to-toe camo.



In order to replicate this look, we created per-weapon paint-by-number masks (A). They combine with the patterns to create weapon-specific color masks (B).

The colorized paint ends up looking like this (C) – everywhere that’s blue gets the same color, everywhere that’s green, likewise.

This transforms our head-to-toe camo gun (D) into something with a bit more personality (E).

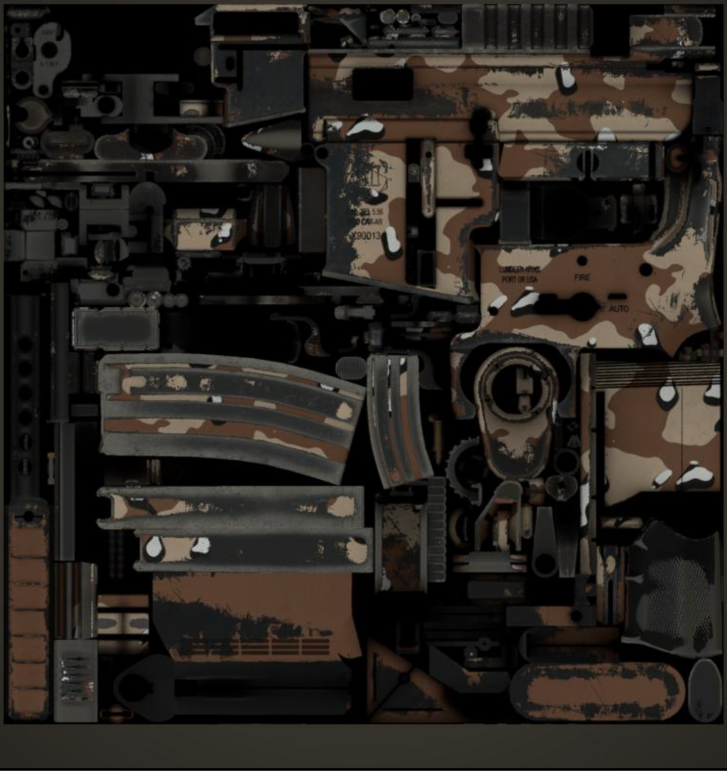


The results of the composite can be varied by randomly choosing different values for rotation and offsets.

So can the wear, so that every item generated is per-pixel unique.

This is all looking very promising. However, it’s a pretty expensive shader, and is extremely memory-heavy with the large number of textures it sources.

Obviously, it’d be better not to run this every frame. In fact, we’d like to run it for only one frame, then save the results.



Up ‘til now we’ve been showing everything in the context of the weapon mesh, but the secret is that the composite so far only needs texture information to work. The whole thing can happen just in image space.

That’s perfect because what we’d like to do is just run the shader for a single frame in a render target, save the result, and dump all the input textures out of memory.

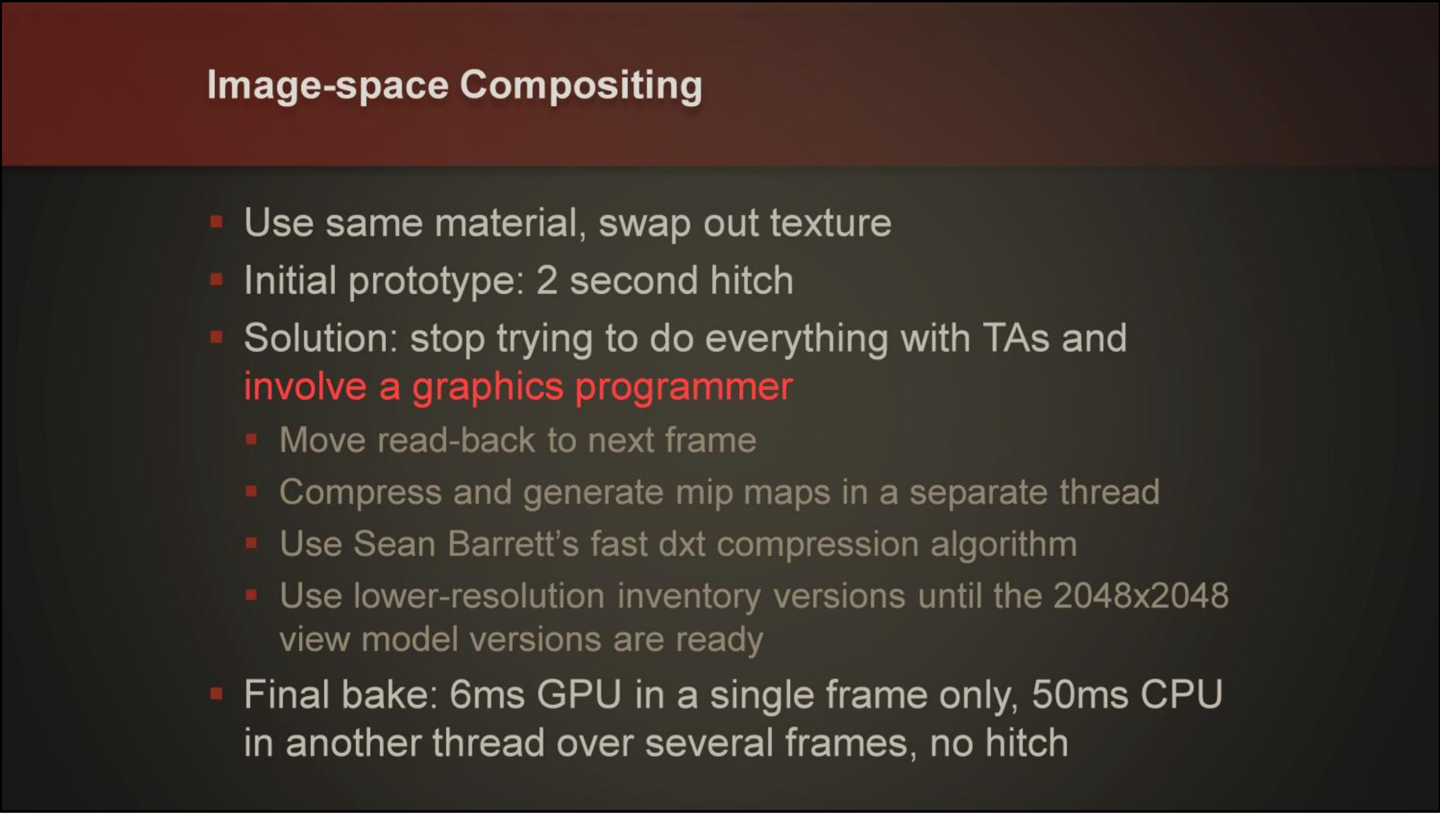


Image-space Compositing

- Use same material, swap out texture
- Initial prototype: 2 second hitch
- Solution: stop trying to do everything with TAs and involve a graphics programmer
  - Move read-back to next frame
  - Compress and generate mip maps in a separate thread
  - Use Sean Barrett's fast dxt compression algorithm
  - Use lower-resolution inventory versions until the 2048x2048 view model versions are ready
- Final bake: 6ms GPU in a single frame only, 50ms CPU in another thread over several frames, no hitch

This gives us an interesting advantage: we could use the original weapon material and just swap out the base texture. We wouldn't have to hand-author any new materials to go with our new finishes.

Our first foray into generating the new texture through a render target wasn't very promising – a 2 second hitch each time a weapon was generated, with no compression or mip maps on the new texture which obviously caused its own slowdowns. There were lots of ways in which we could improve, the first one being: stop trying to do everything with just technical artists. We asked a graphics programmer for help optimizing the step where we save the texture to memory.

-The read back from the render target was happening on the same frame as the composite, causing a massive stall. Just moving the read to the next frame improved things dramatically.

-Compressing the texture and generating the mip maps got moved into another thread. We just keep running the game and don't display the texture before it's ready.

-We use Sean Barrett's dxt compression algorithm, rather than the much slower algorithm that we use for generating high-quality compression results offline. The 2048x2048 compressed result is generally ready within 50 ms, and since it runs in another thread, there's no gameplay impact.

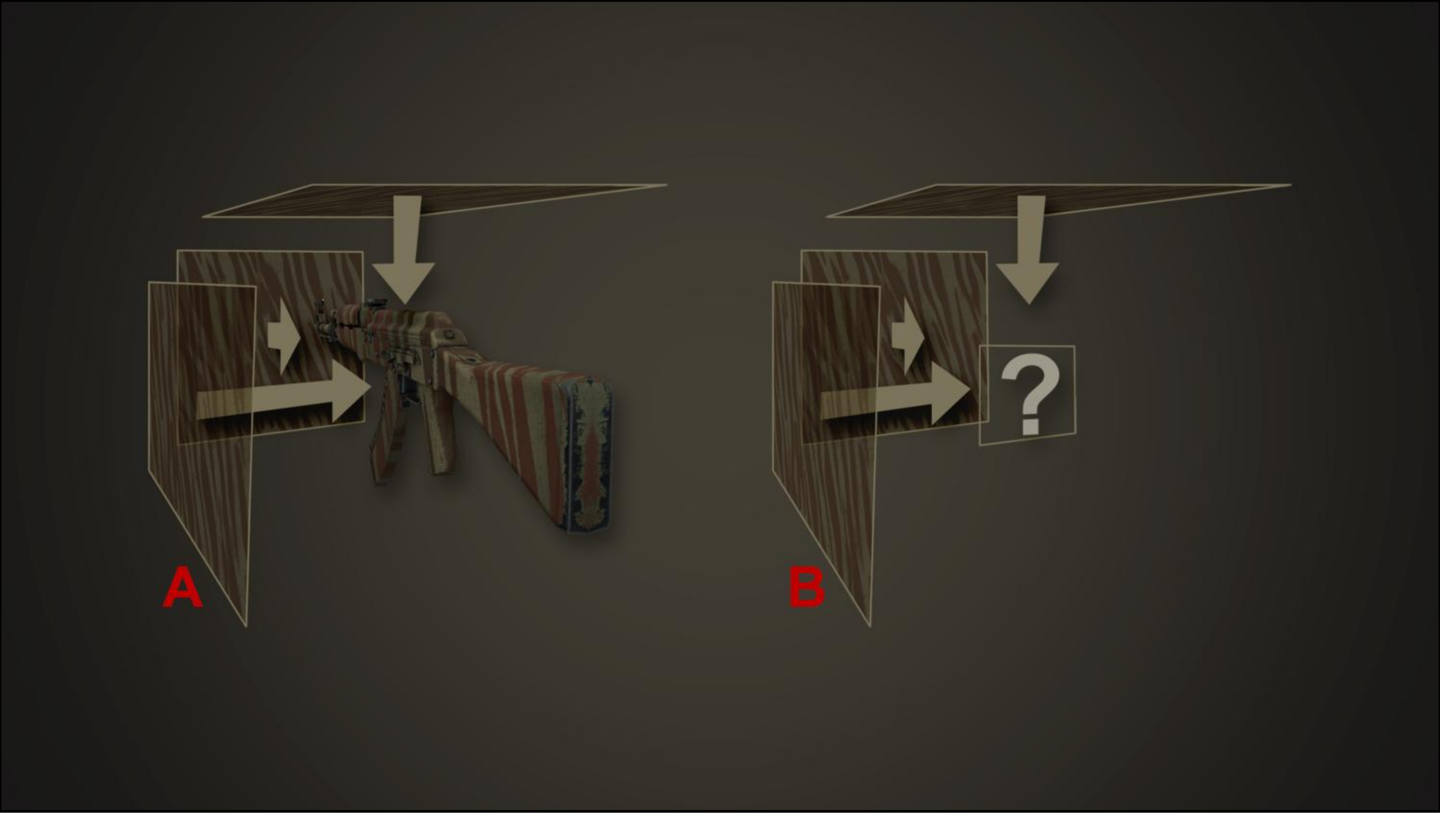
-We generate much smaller versions of the composited textures in the main menu for all items in a player's loadout, and just keep them around in-game. Other players see those versions, and until your 2048x2048 texture is ready for the weapon you're currently holding, you see it too. Generally you only see it for a couple of frames right at the beginning of the game, and it's not noticeable.

-If you pick up someone else's weapon in-game, you see the lower resolution version until the 2048x2048 version is ready. Again, most of the work happens in a separate thread so there's no gameplay impact. In that situation, you typically see the lower resolution version for a little bit longer, depending on your machine's specs, but there's no significant pop and it's an acceptable trade-off to prevent hitching during action.

So with a fully functioning prototype in-game, we felt confident enough to create some new finish styles.

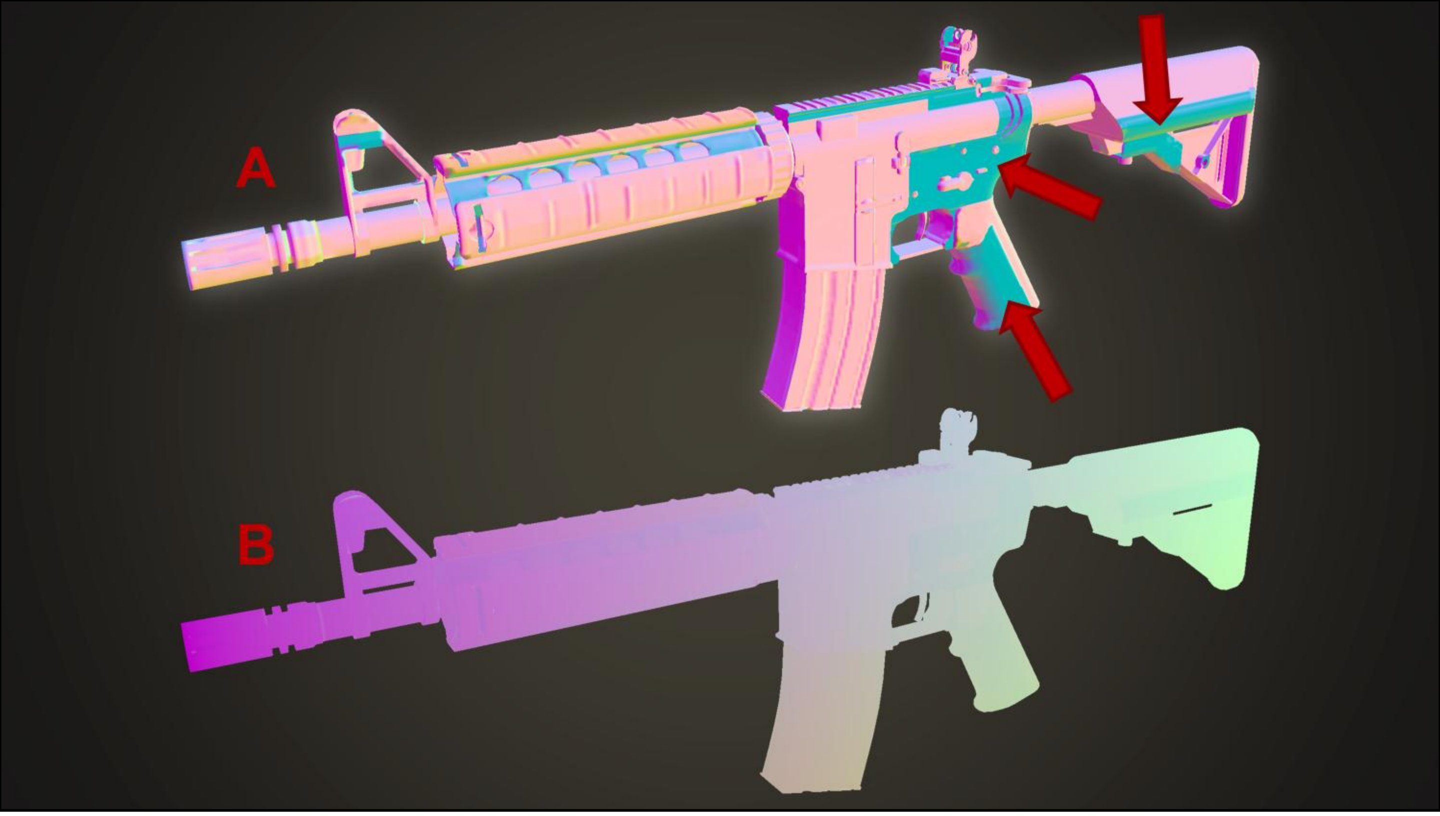
The first reference we'd looked at consisted of spray-painted weapons. We'd ignored them so far because solving the problems for hydrographic finishes was easier.

Here's the thing: spray paints should be seamless.



We’d normally just do a triplanar mapping in the shader, projecting the pattern texture from cardinal directions and blending across the surface normals. But this requires having mesh data to pass from the vertex shader into the pixel shader in order to do these operations (A).

But we’re rendering to a render target and doing the composite only in image space. Far from having all the weapon’s mesh data, what we actually have in place is a flat plane with no normals (B). So if we want seamless mapping, we somehow need all the surface information in texture format.



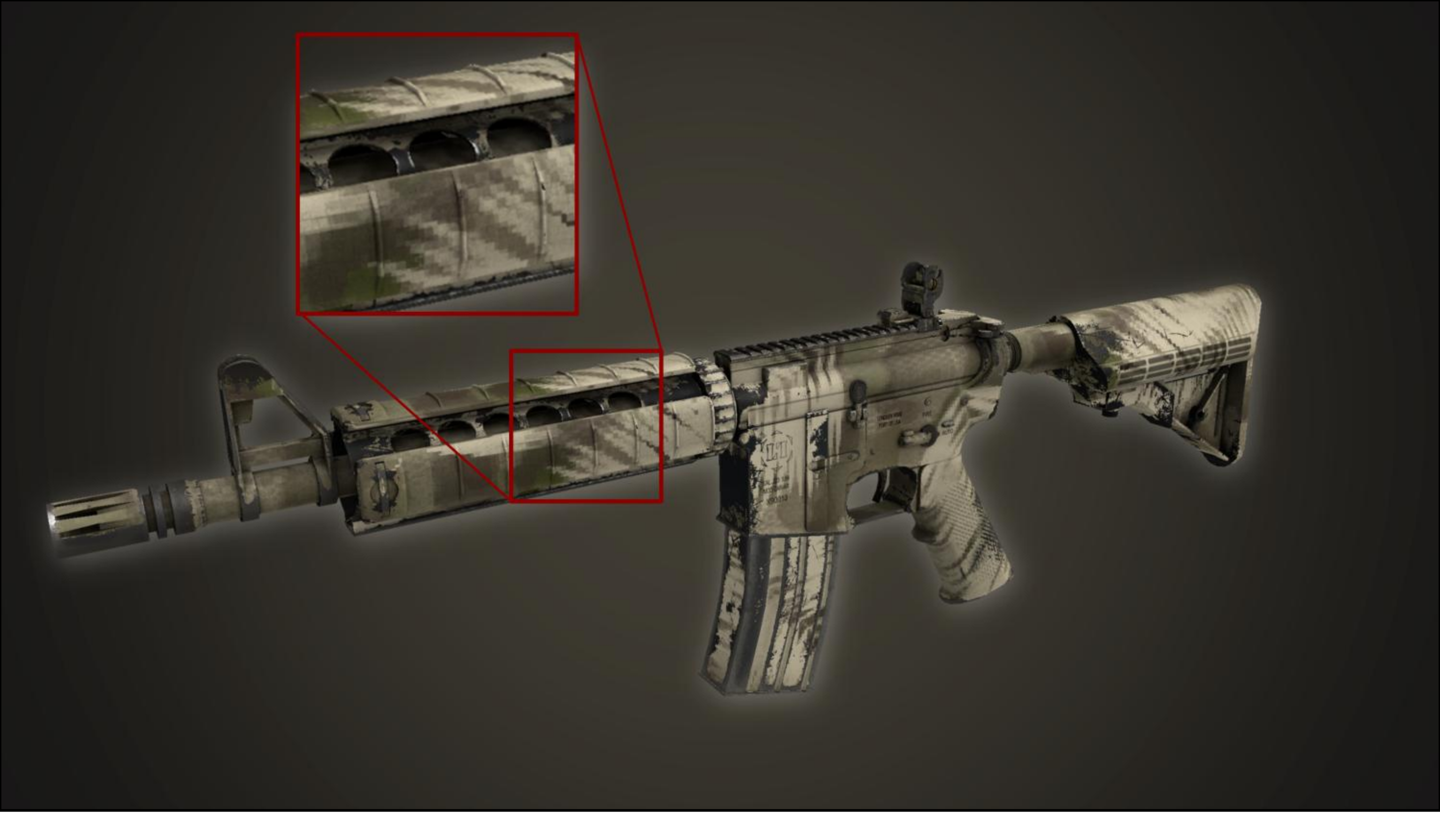
Normals are easy. We just create an object-space normal map for the weapon (A).

Bear in mind that we never authored the UVs of any of the weapons to optimize for any of these procedural solutions. This was all happening after we'd shipped the game. Our weapon models were already made.

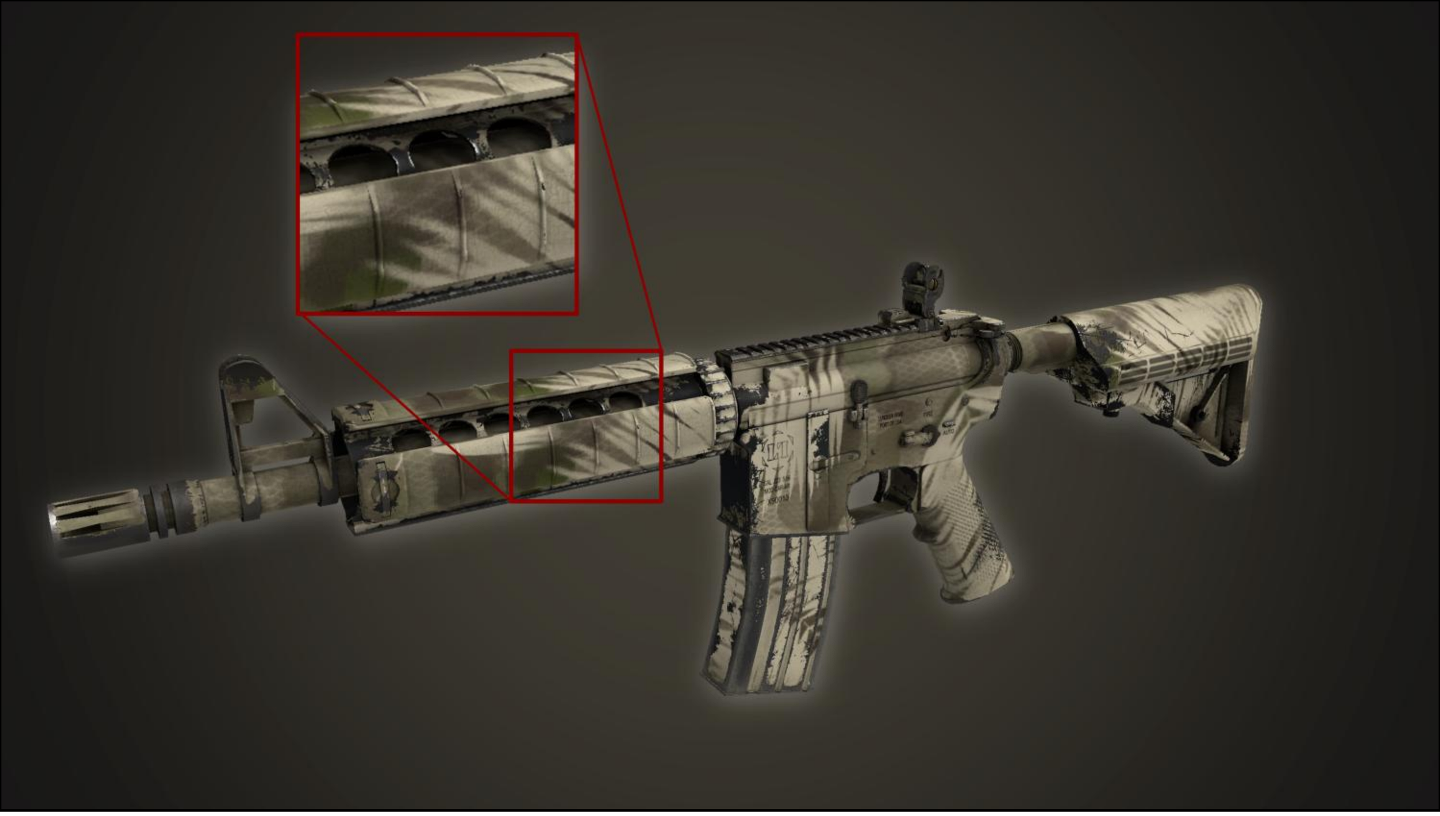
So since some of the UV islands overlap over re-used portions of the texture (indicated by arrows), this isn't a perfect representation of the object's normals.

It's hopefully good enough for working out blending between the different texture projections. It's not like we're actually shading with this information.

Position information is a little more difficult. We can actually bake out position XYZ as RGB (B), but precision is an issue.

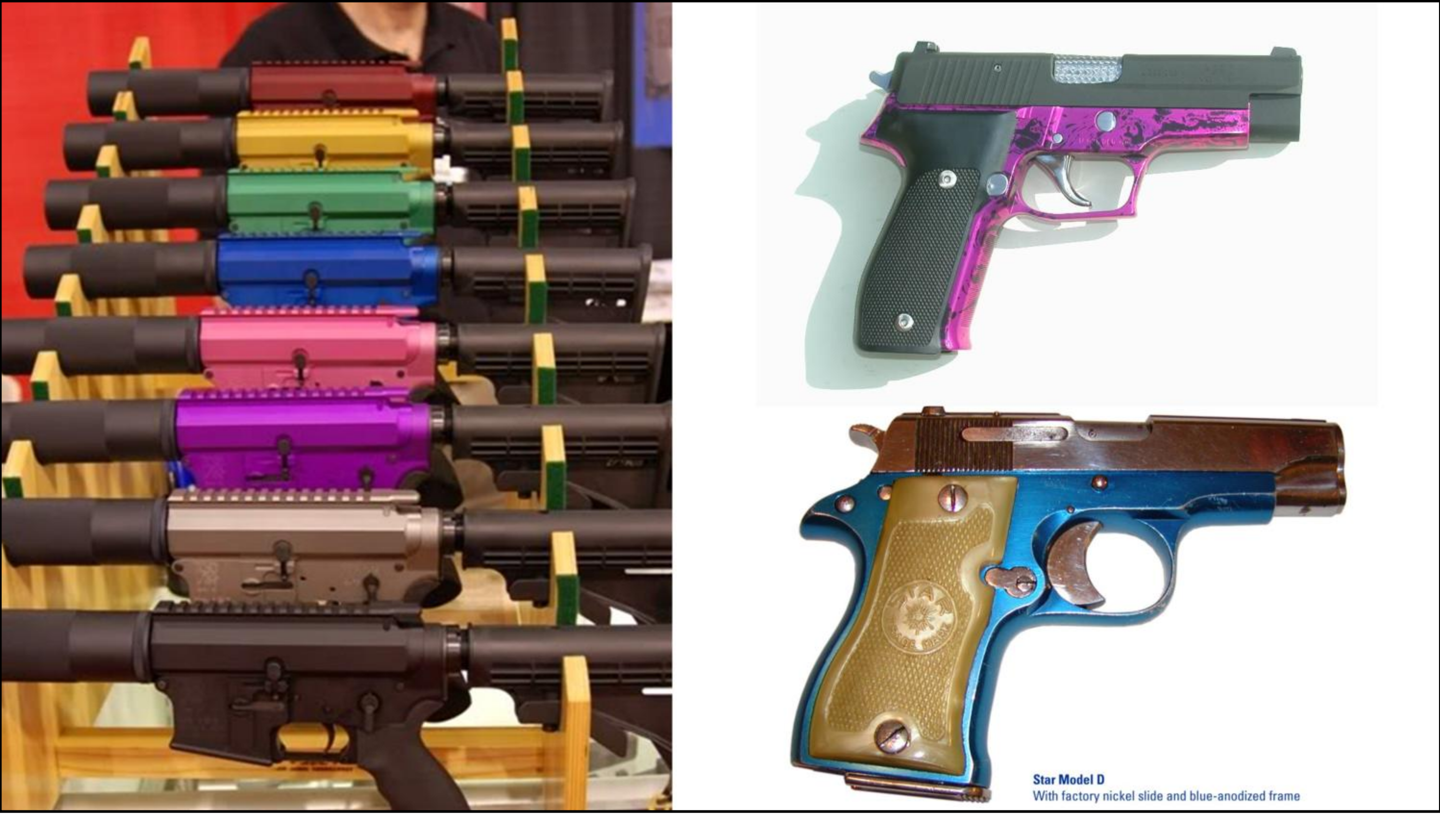


We saved out a 32-bit uncompressed texture, the most precise texture we can load in-game, but when we plug that information into the triplanar mapping what we get shows us some pretty nasty blockiness.



The solution is to multisample. We use a gaussian kernel to smooth the precision artifacts, and the results are surprisingly good.

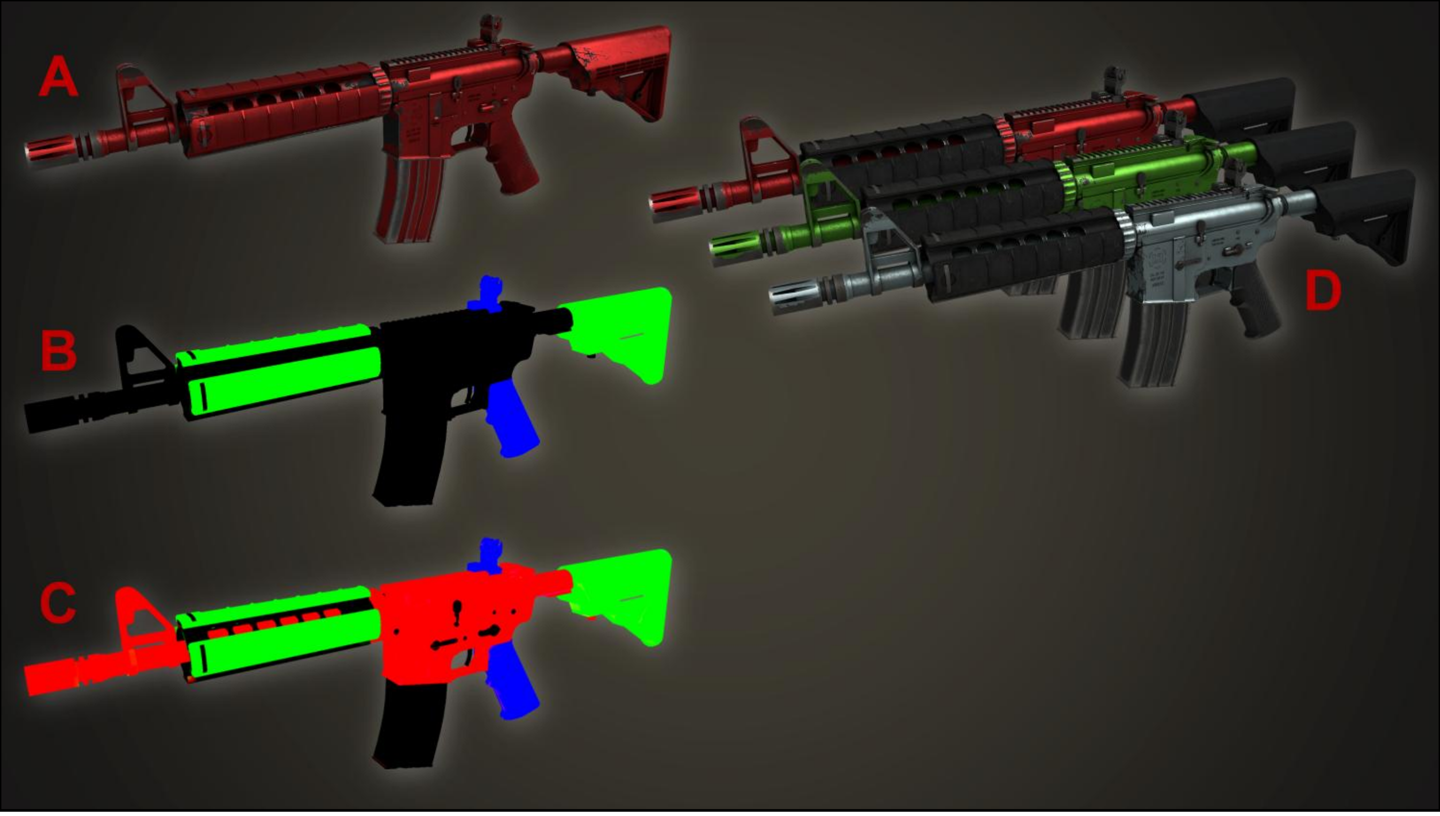
In terms of visual range, hydrographics and spray paints aren't all that dissimilar. We need more variety than just two styles in order to meet our goals.



Star Model D
With factory nickel slide and blue-anodized frame

We kept looking and found that a common finishing technique for firearm components is anodization.

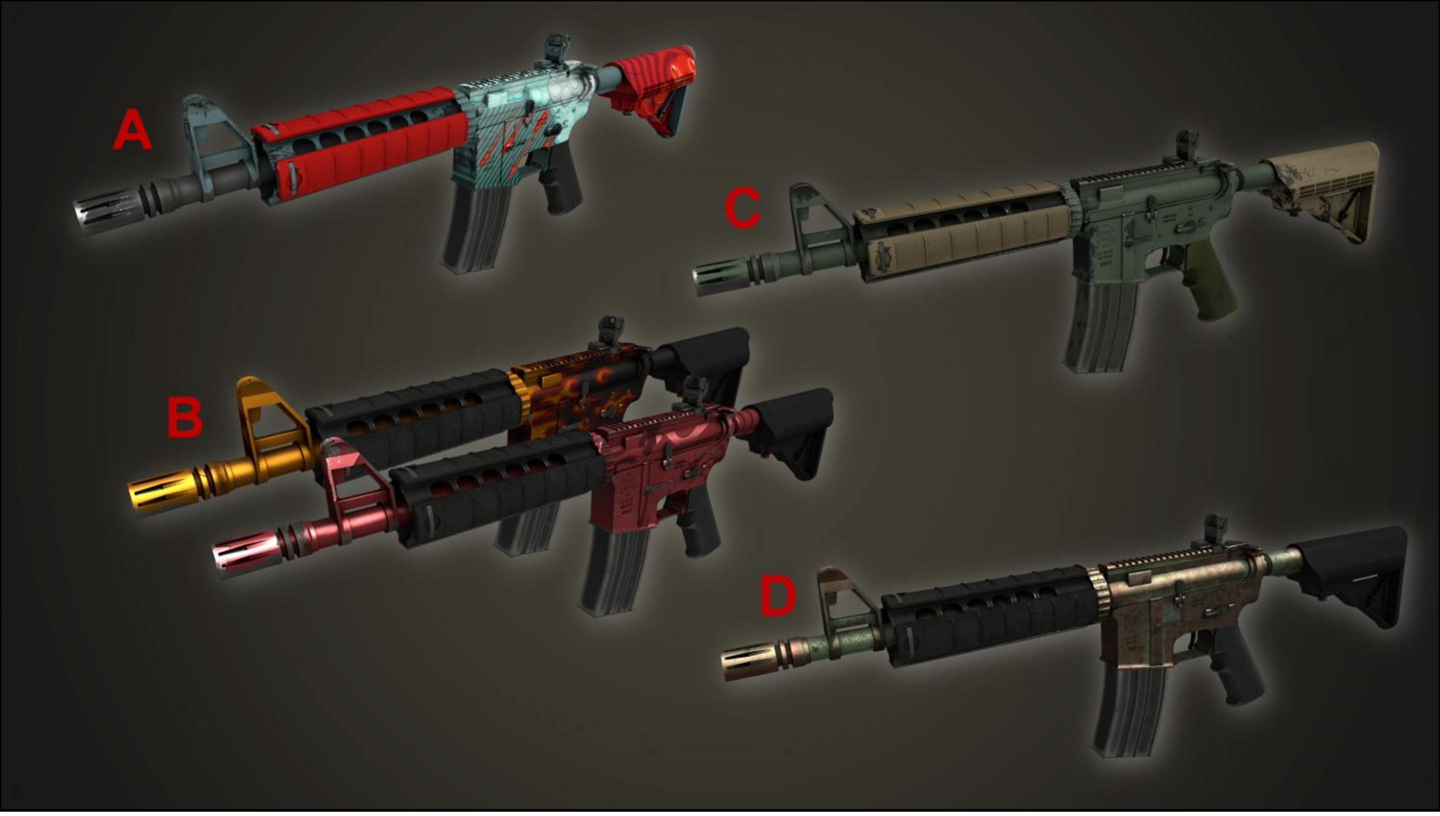
Our standard shader for the weapons does allow us to tint the specular highlights by the base texture's color. This lets us make metallic looking highlights and anodized effects like in these real-world examples.



In this example, the paint is colored red and baked into the base texture (A). The specular highlights are tinted by the base texture wherever the weapon is painted, so you get juicy red highlights over the paint but the substrate shades normally. It's pretty but it looks really weird having the entire weapon covered in the effect.

Remember the mask we used to set accent colors in the hydrographic example (B)? We've got a channel left – so we added another per-weapon mask to limit the anodization effect (C).

After the mask is made for a particular weapon, the only overhead to creating a new anodized finish is picking a color (D).



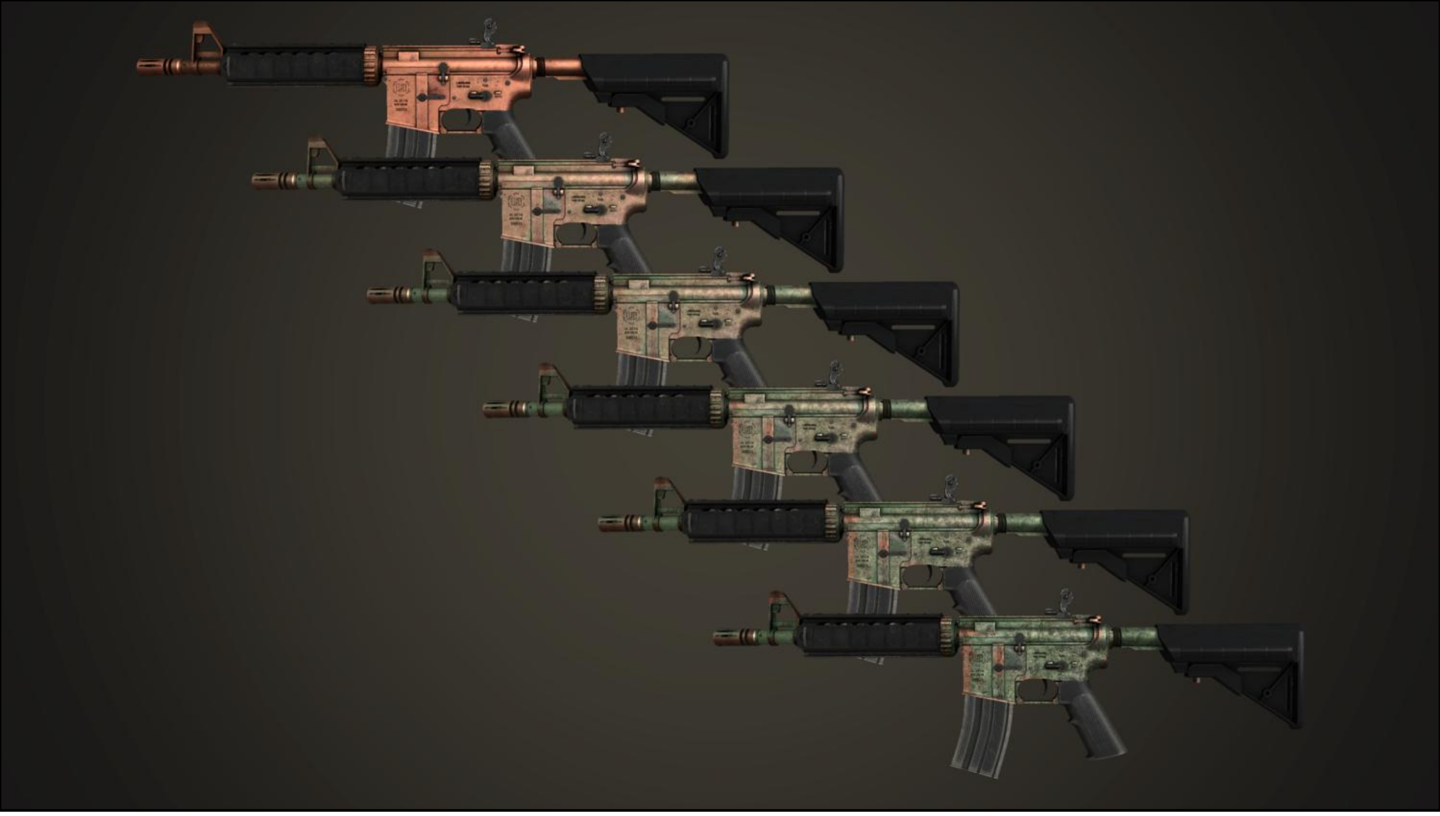
So did we stop there? Well, how could we? All those textures we made contain a lot of really helpful information and we haven’t taken full advantage of it yet.

We added some additional styles, finally getting around to using a non-colorized paint texture (A). We also converted the hydrographic and spray paint styles into anodized looks (B).

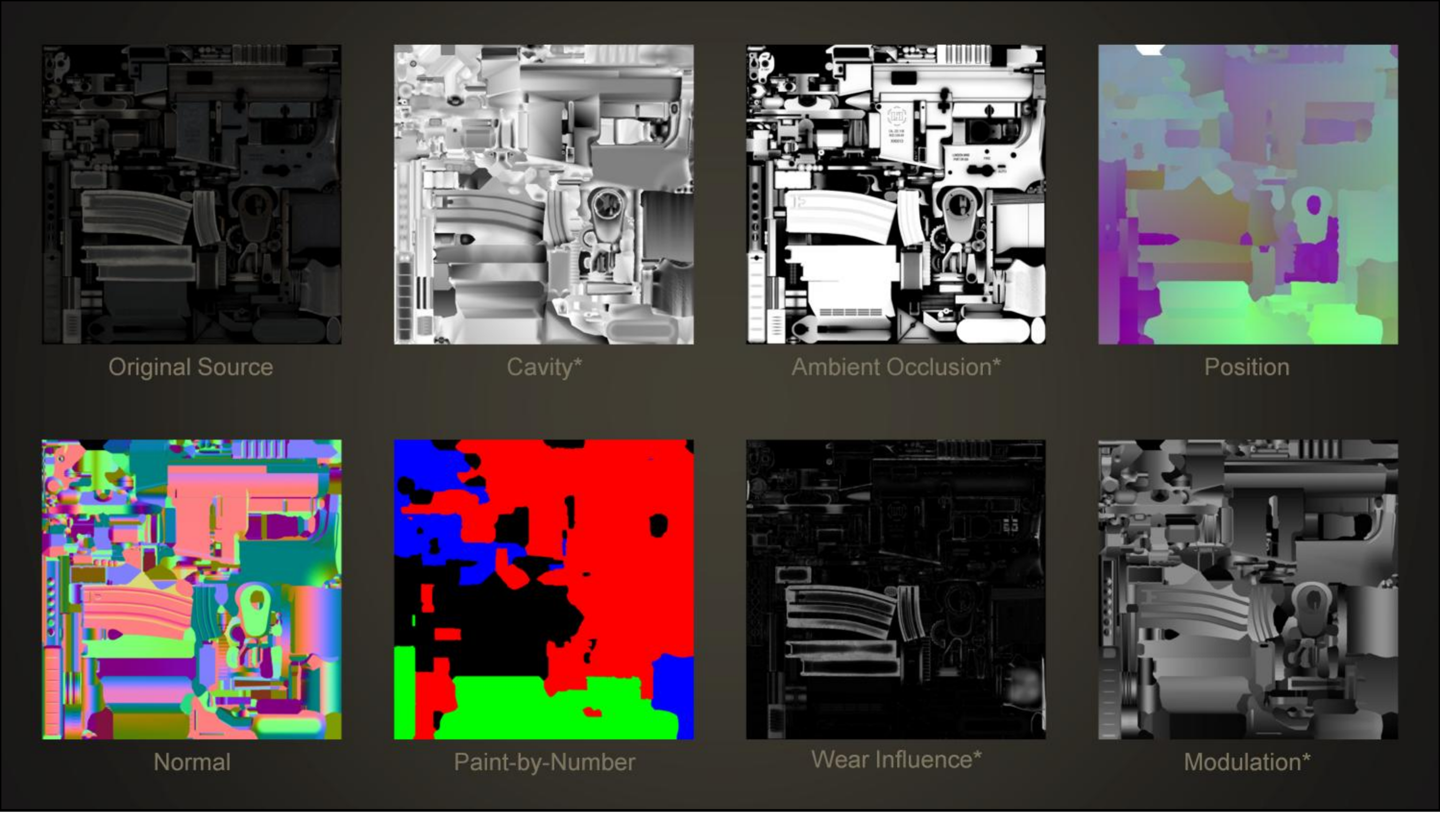
We used our paint-by-number mask to creat a solid paint style with up to four colors (C).

Lastly, we created a patina style that, instead of wearing down to the substrate, ages to a thicker patina (D).

We didn’t create any new per-weapon texture inputs for any of these styles, instead using shader math to repurpose our existing inputs.



This is pretty much just procedural art, the cheapest kind of art we can make. And yet, it communicates the copper material really well. It also looks interesting and different at all wear stages, which supports our goal of creating variety.



Original Source

Cavity*

Ambient Occlusion*

Position

Normal

Paint-by-Number

Wear Influence*

Modulation*

To recap so far: we have a procedural compositing system that we can use to generate an essentially infinite number of textures. In order for the system to work, we need the following inputs per-weapon:

-The original source textures, which were finished before we even started on our composite idea

-Cavity, ambient occlusion, position, and normal. All of these are just generated from the mesh data using out-of-the-box tools

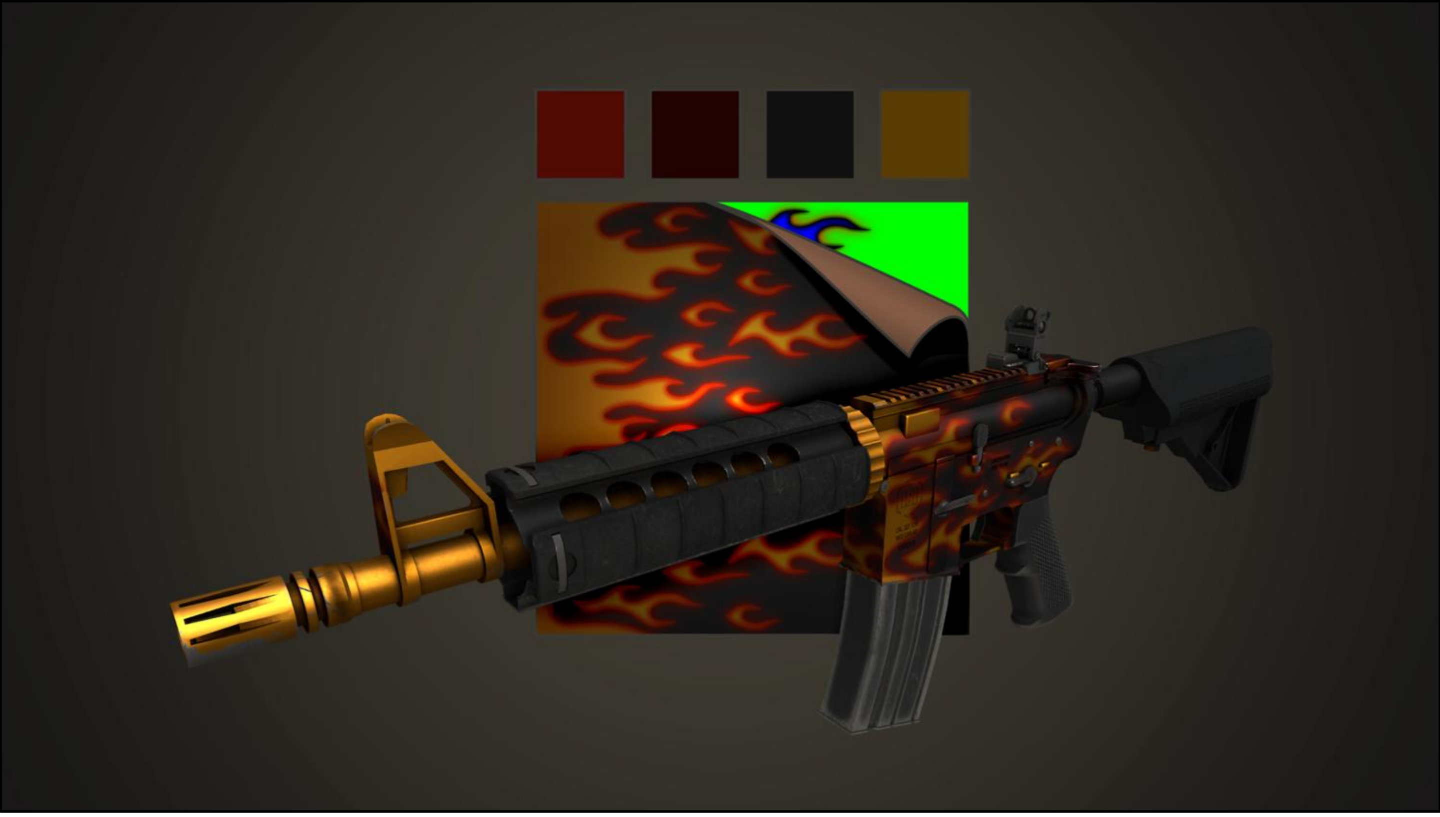
-A per-weapon paint-by-number texture calling out elements to be painted different colors or anodized, which we quickly generated by selecting elements of the mesh and colorizing them

-A per-weapon wear mask, which we mostly just cribbed from the original source PSDs

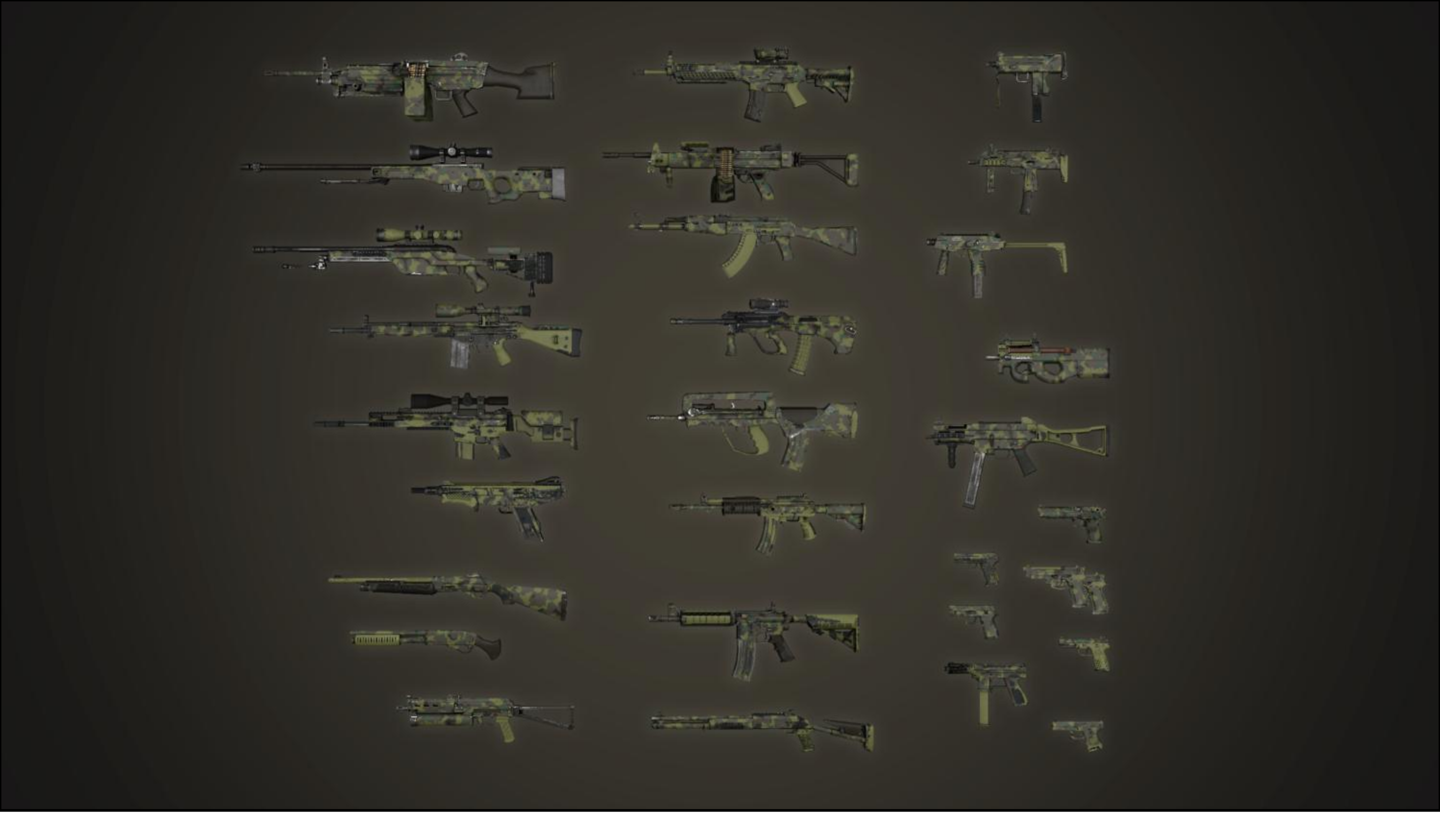
-The Mig Jimenez-style modulation, which we had to create by hand, the only texture on this list that represents a significant time investment.

In all, converting a single weapon to be paintable takes about half a day – not too shabby. Even with just a couple of artists working on it, we blew through all 30 shipped weapons in a week and a half.

*That looks like a lot of inputs, but we reduce memory consumption and shader operations by combining the black-and-white images into a single texture's individual channels.



For the finish, what we author depends a bit on the finish style, but typically we create a pattern, then choose the colors and set the parameters for scale and specular characteristics. The randomization system does the rest to create variations on the original, adding wear and UV offsets.

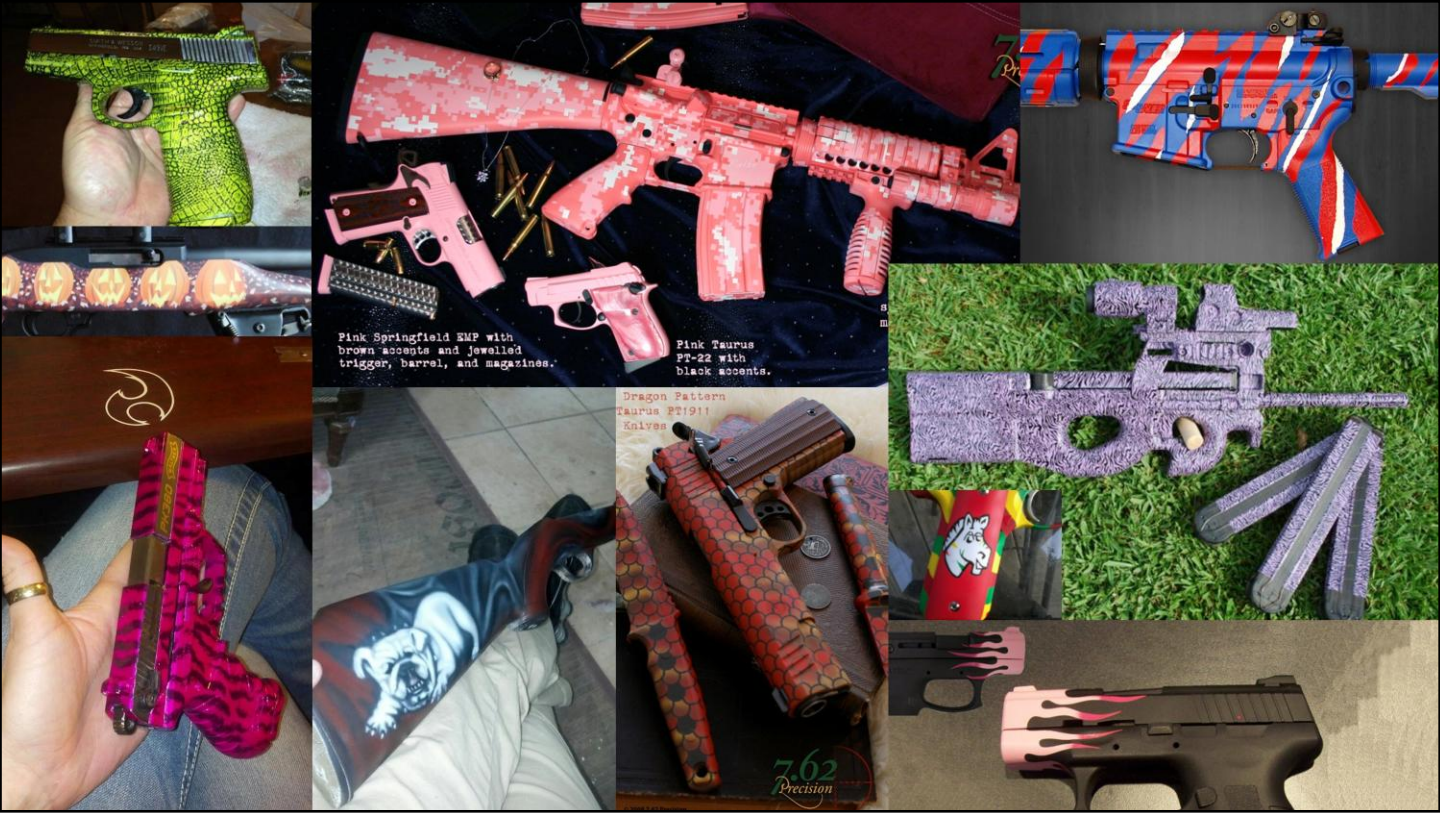


Once all the weapons had been prepped, we can very easily author a single finish that could apply to any weapon. All the detail is procedurally applied, so adding a finish to a weapon is essentially a drag-and-drop operation.

A single weapon finish can create a huge amount of variety by randomizing its parameters and applying it to different weapons.

At this point it was time to stop focusing on the tech, and start focusing on the art.

In the course of gathering reference we discovered that…

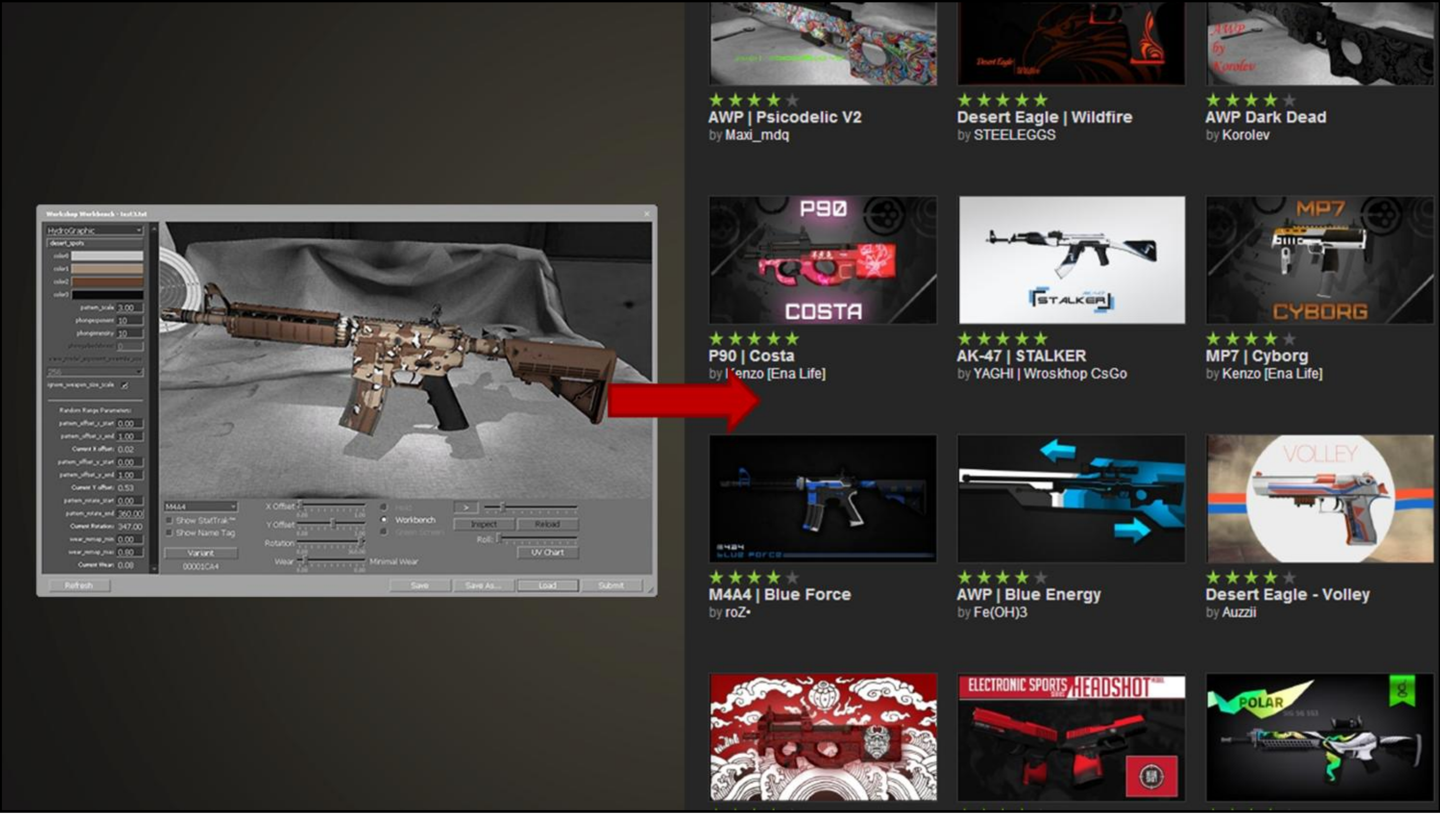

Pink Springfield EMP with brown accents and jewelled trigger, barrel, and magazines.
Pink Taurus PT-22 with black accents.
Dragon Pattern Taurus PT1911 Knives
7.62 Precision

…in the real world, people make things we found unattractive.

It begs the question, what do our customers want? The game’s art style is grounded in the real world and we’ve already gone down the road of replicating real-world finishing techniques. But there’s a really broad range of service providers in the real world catering to a broad range of tastes.

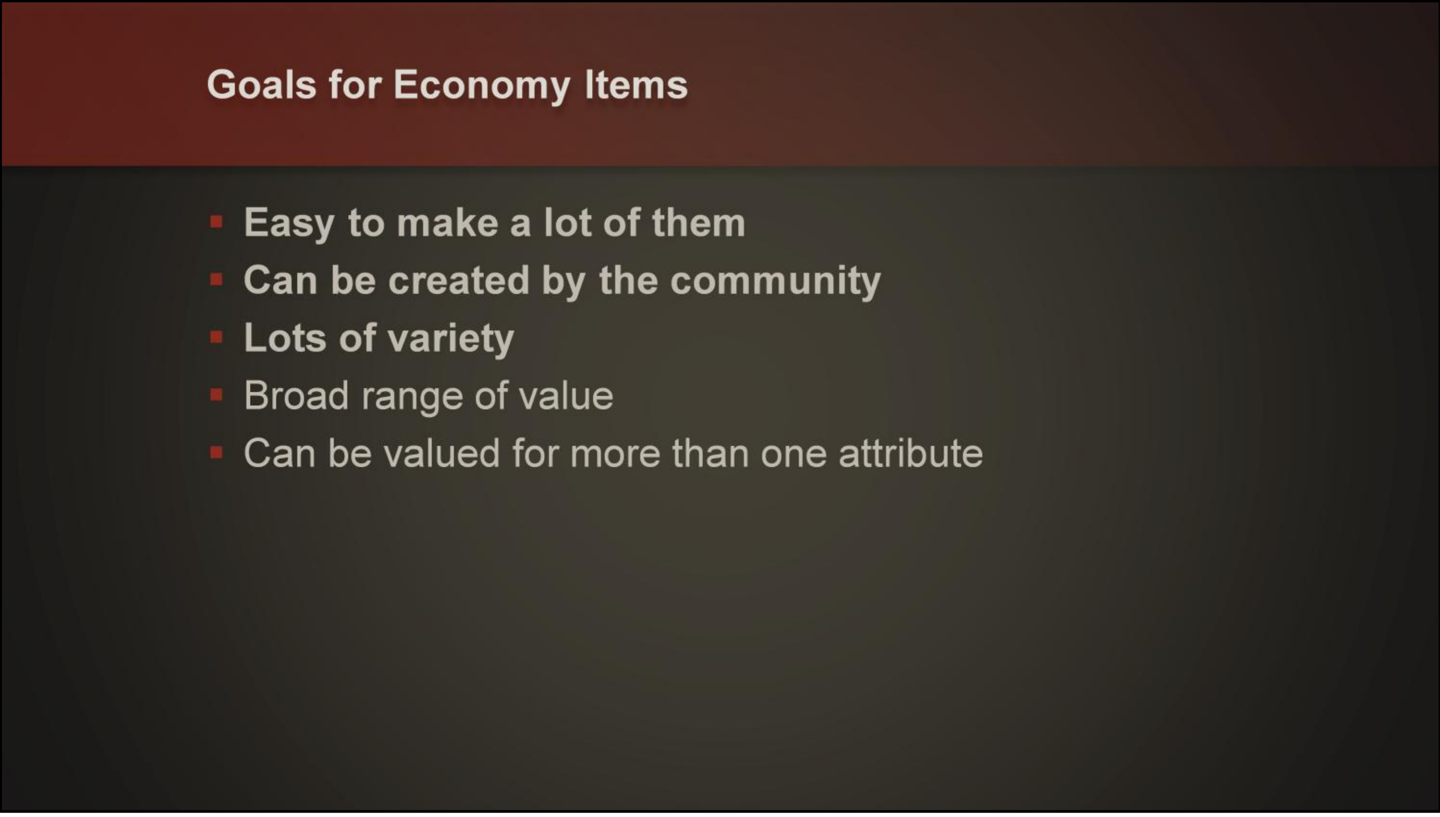
In our game world, we’re the only suppliers and our taste might not match demand. So we needed to focus less on what we personally liked and more on creating a broad range of variety in order to measure people’s reactions.

So we shipped things with a wide range of aesthetics, deliberately shipping items that were contentious within the team.



We also shipped an in-game tool for creating finishes that we call the workbench. This allows the community to create weapon finishes and upload them to the community workshop, which they've done with enthusiasm, creating over 30,000 entries. We've already begun to ship some of the highest rated items from the workshop, letting the community express their aesthetic preferences directly.

We called our economy update the Arms Deal Update, introducing the weapon finishes for players to collect and equip to customize their play experience. The initial offering contained over 100 weapon finishes, all created in-house, which we've since expanded almost fourfold with the help of our community workshop contributors. And each of those finishes has random variations created by pattern placement and wear, meaning that there are tens of millions of unique items in existence.

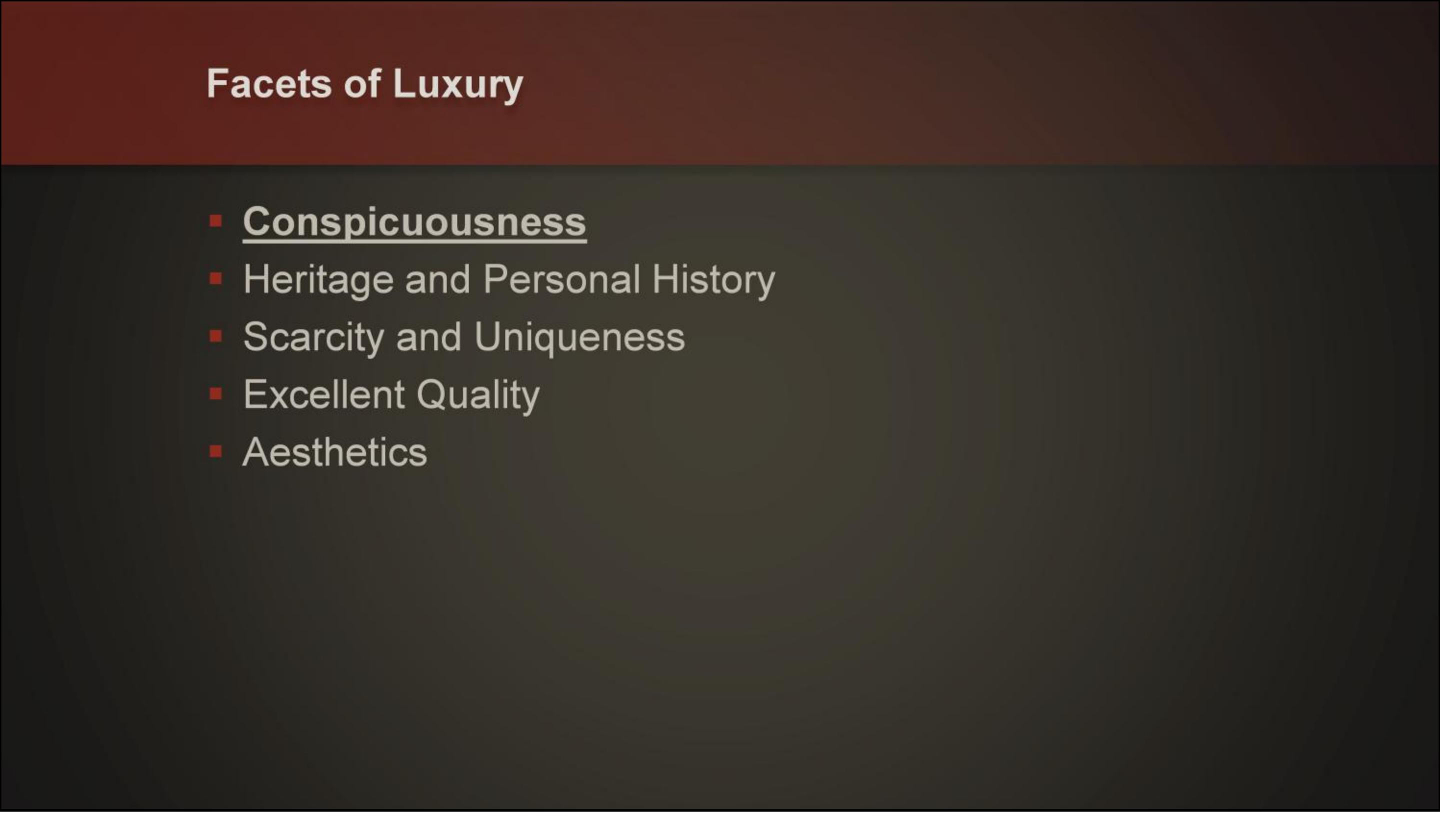


Goals for Economy Items

- **Easy to make a lot of them**
- **Can be created by the community**
- **Lots of variety**
- Broad range of value
- Can be valued for more than one attribute

Checking back in with our goals, so far we'd managed to meet the first three. It's easy to make finishes, the community contributes to their creation, and we have a large variety of looks.

Our last two goals are to create a broad range of value, and to create items that can be valued in different ways. To know whether we're meeting those goals we'll have to look at the data.



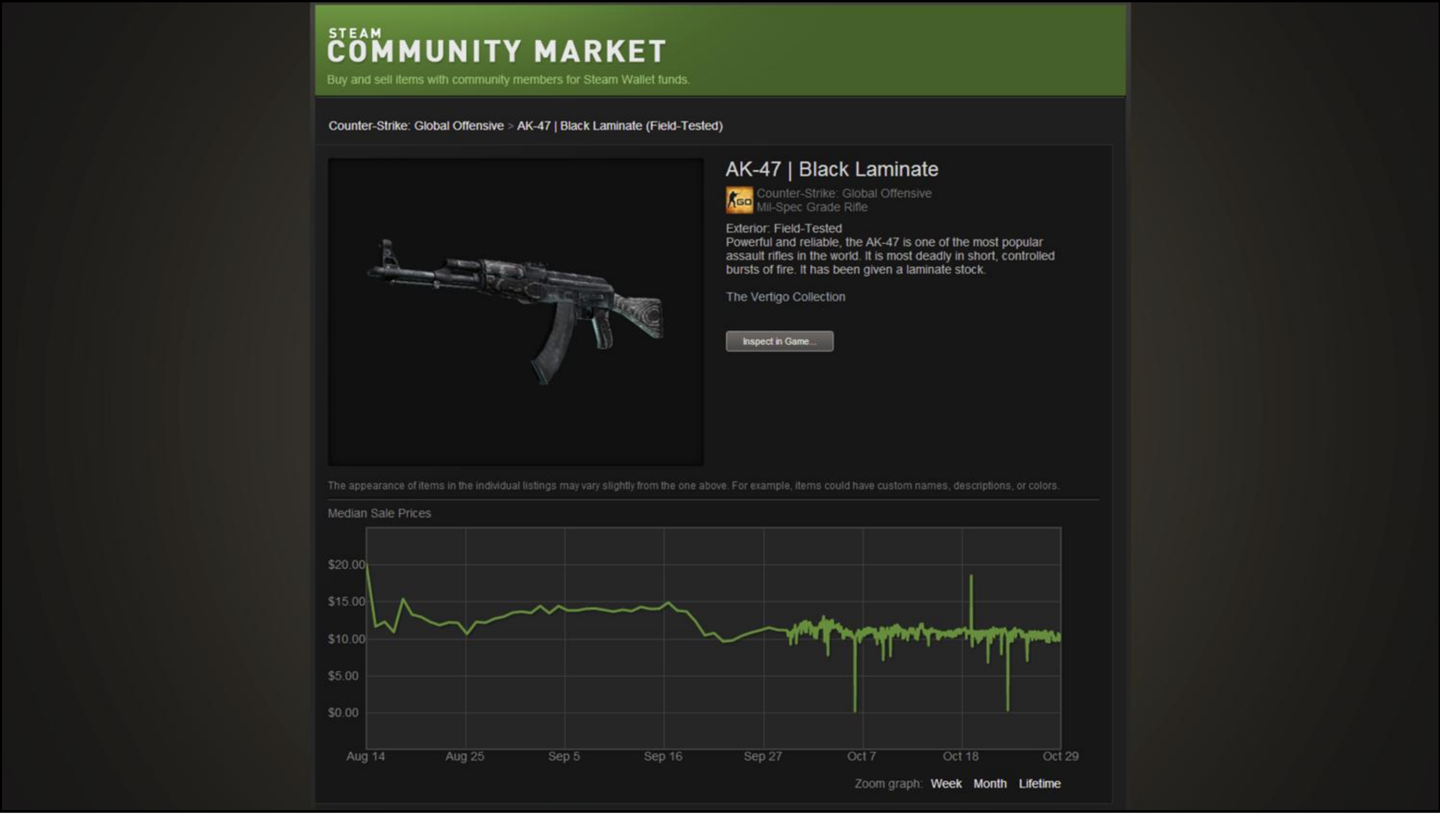
Facets of Luxury

- **Conspicuousness**
- Heritage and Personal History
- Scarcity and Uniqueness
- Excellent Quality
- Aesthetics

Weapon finishes are by definition luxury goods: they are things our customers like to have, but they are not necessary in order to play the game. In fact, they have no actual gameplay impact at all.

There is plenty of academic discussion on the dimensions of luxury goods. Here's an aggregated list of commonly identified facets that we can use as a jumping off point to discuss what contributes to the value of an item in the economy.

Let's start with conspicuousness – the public display of your status as a person with access to luxury goods.



STEAM COMMUNITY MARKET

Buy and sell items with community members for Steam Wallet funds.

Counter-Strike: Global Offensive > AK-47 | Black Laminate (Field-Tested)

AK-47 | Black Laminate

Counter-Strike: Global Offensive
Mil-Spec Grade Rifle

Exterior: Field-Tested
Powerful and reliable, the AK-47 is one of the most popular assault rifles in the world. It is most deadly in short, controlled bursts of fire. It has been given a laminate stock.

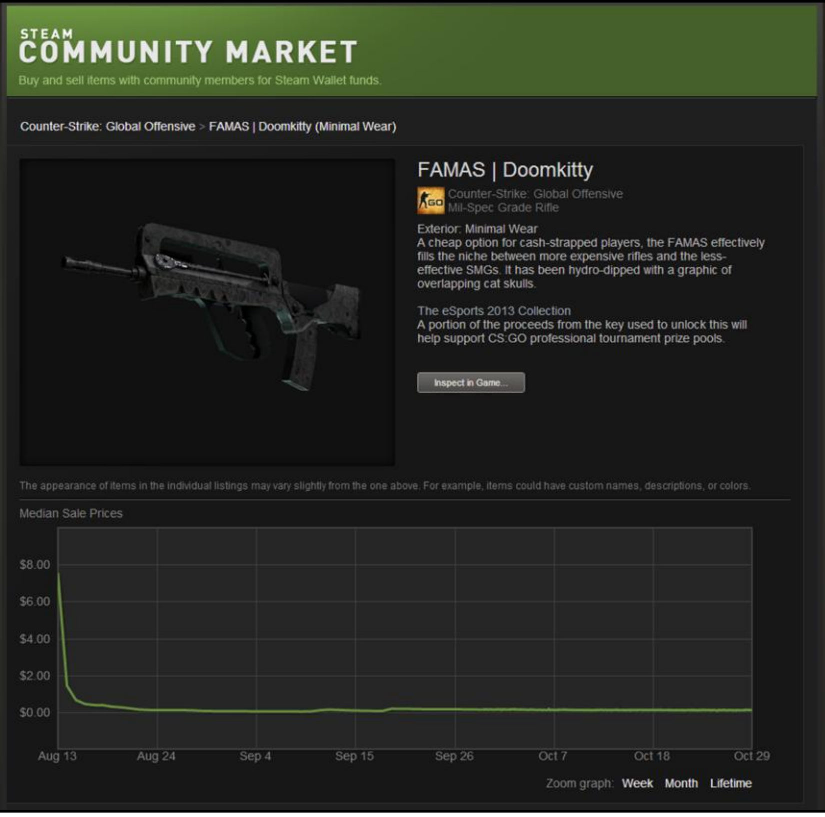
The Vertigo Collection

The appearance of items in the individual listings may vary slightly from the one above. For example, items could have custom names, descriptions, or colors.

Median Sale Prices

This is a graph of the sale price of a Black Laminate AK-47 over its first three months. This isn’t privileged information – anyone can see this graph just by visiting the Steam Community Market.

You can see the price is pretty stable around $10. So the community values this item pretty highly.



STEAM COMMUNITY MARKET

Buy and sell items with community members for Steam Wallet funds.

Counter-Strike: Global Offensive > FAMAS | Doomkitty (Minimal Wear)

FAMAS | Doomkitty

Counter-Strike: Global Offensive
Mil-Spec Grade Rifle

Exterior: Minimal Wear
A cheap option for cash-strapped players, the FAMAS effectively fills the niche between more expensive rifles and the less-effective SMGs. It has been hydro-dipped with a graphic of overlapping cat skulls.

The eSports 2013 Collection
A portion of the proceeds from the key used to unlock this will help support CS:GO professional tournament prize pools.

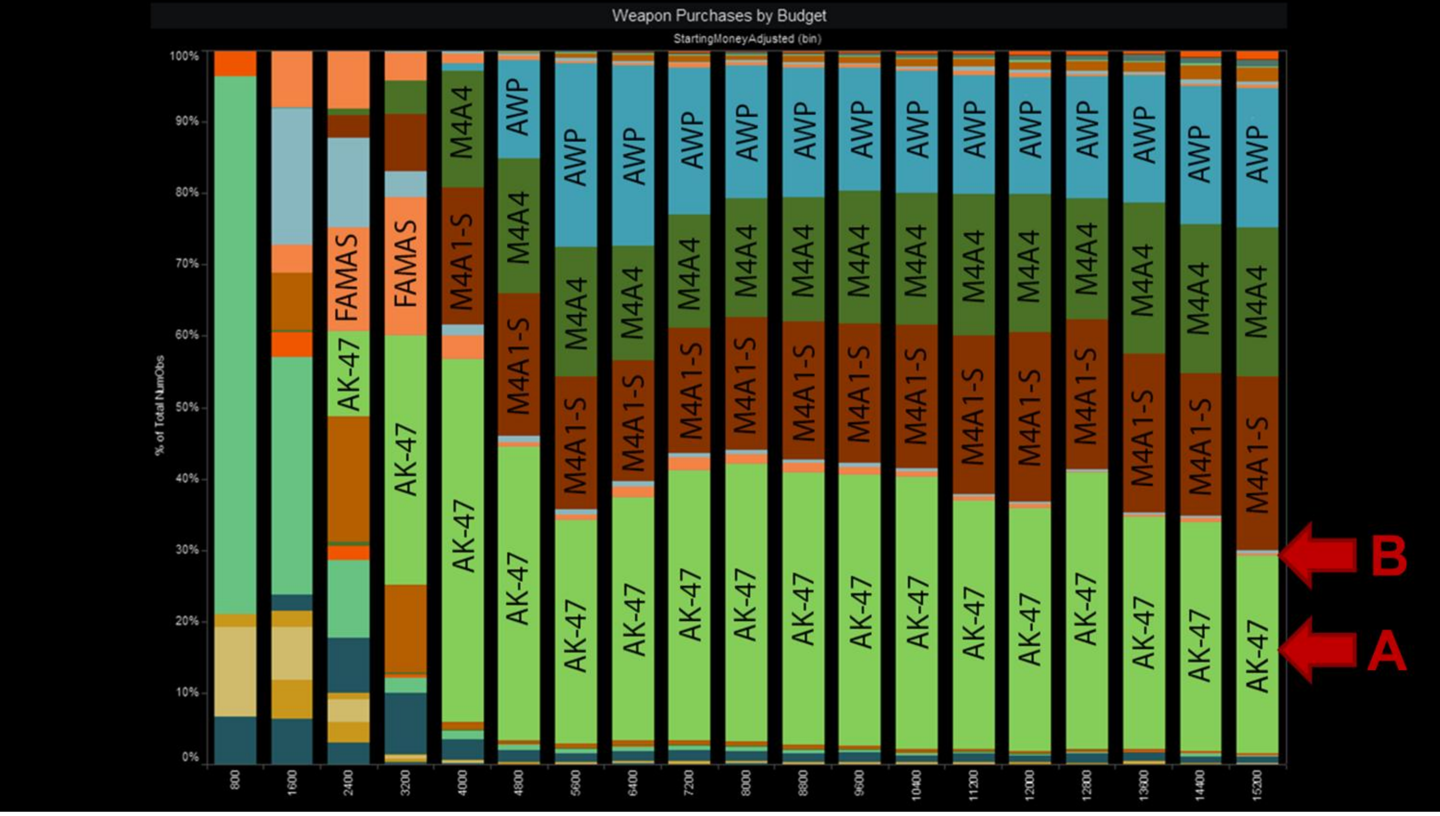
The appearance of items in the individual listings may vary slightly from the one above. For example, items could have custom names, descriptions, or colors.

Median Sale Prices

Here’s another item, the Doomkitty FAMAS. Prices for this item started fairly high, around $7.50, and dropped really rapidly. Now the items maintains a fairly steady price of 7 cents.

So this guy isn’t a huge hit.

What’s so different between these two items?

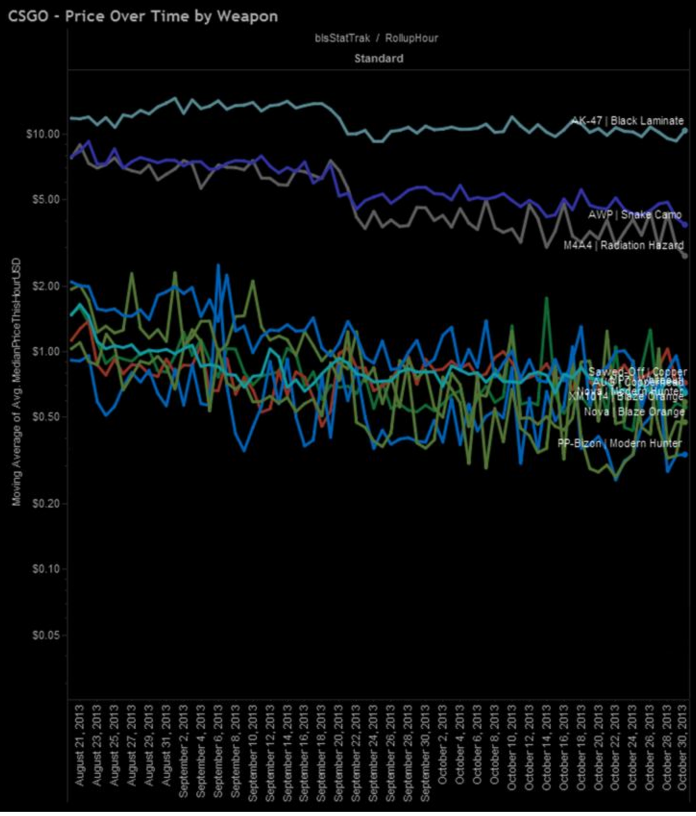


While the finishes themselves are superfluous, the weapons they apply to are not. This graph shows weapon utility. It’s a sampling of what primary weapons got used most often depending on the start conditions of a round. This data is from last October, when the market was about three months old.

The AK-47 has the highest utility (A). The M4s and the AWP are next with roughly the same utility each.

The rest of the primary weapons have a pretty small share, especially in later rounds. The FAMAS, our example from the last slide, is a tiny orange band sandwiched between the higher utility weapons (B).

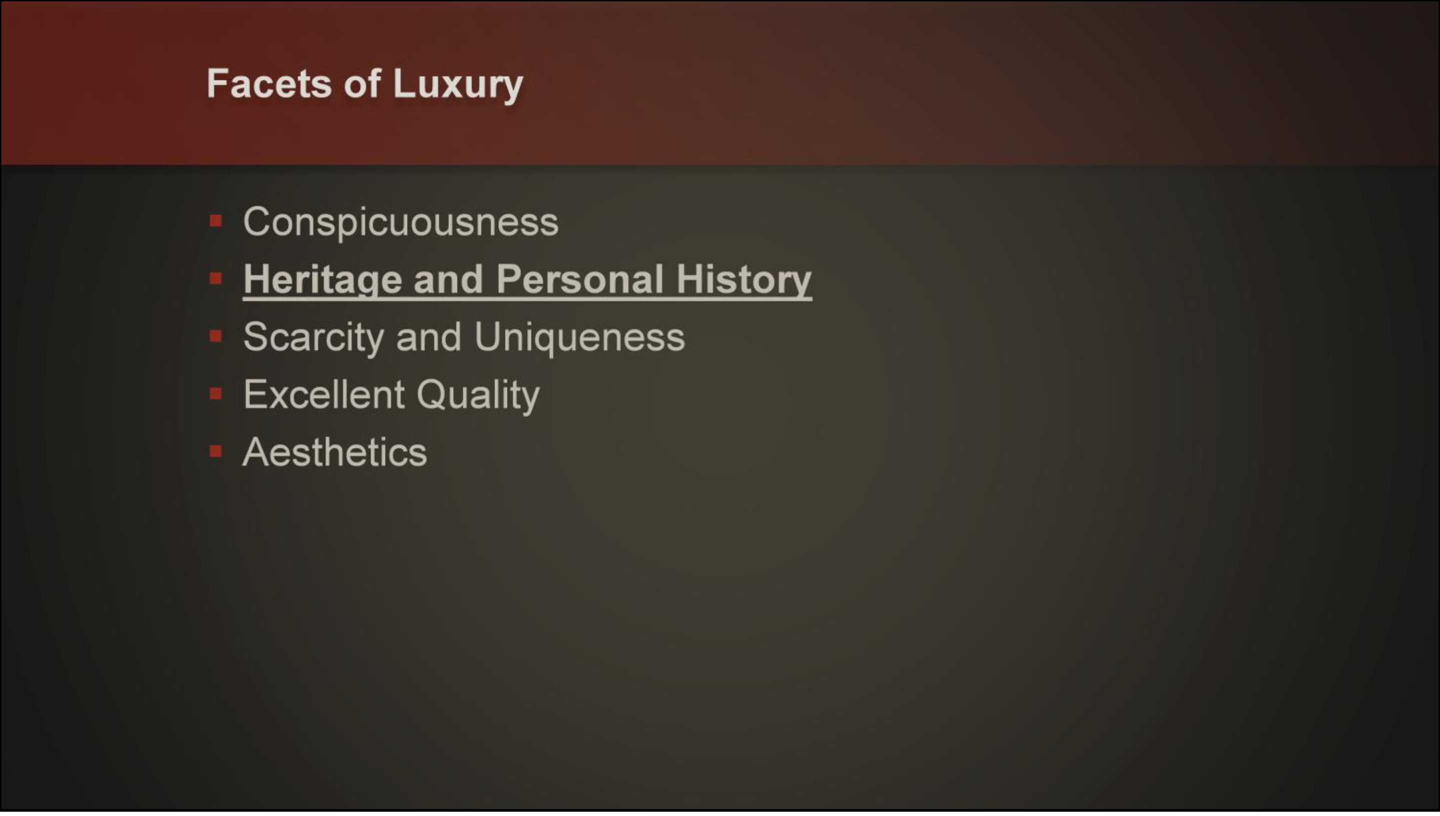
So if you have a fancy AK-47, you get to show it off a lot. If you have a fancy FAMAS, you don’t.



If we look at the prices of weapon finishes with the same scarcity, we see exactly that breakdown. AK on top, M4A4 and AWP, then a cluster of the rest of the weapons. (The additional spikiness at the lower end is due to the logarithmic scale of the graph, it's not that the prices of these items are more volatile.)

This show us that the amount of time you spend holding a particular weapon increases the value of a finish for that weapon. That's just what we'd expect – more time means more chances to publicly show off your luxury goods, as well as more time to enjoy them yourself. So conspicuousness is definitely a big factor.

That means when we create a weapon finish, just applying it to different weapons can create a big variety of value.

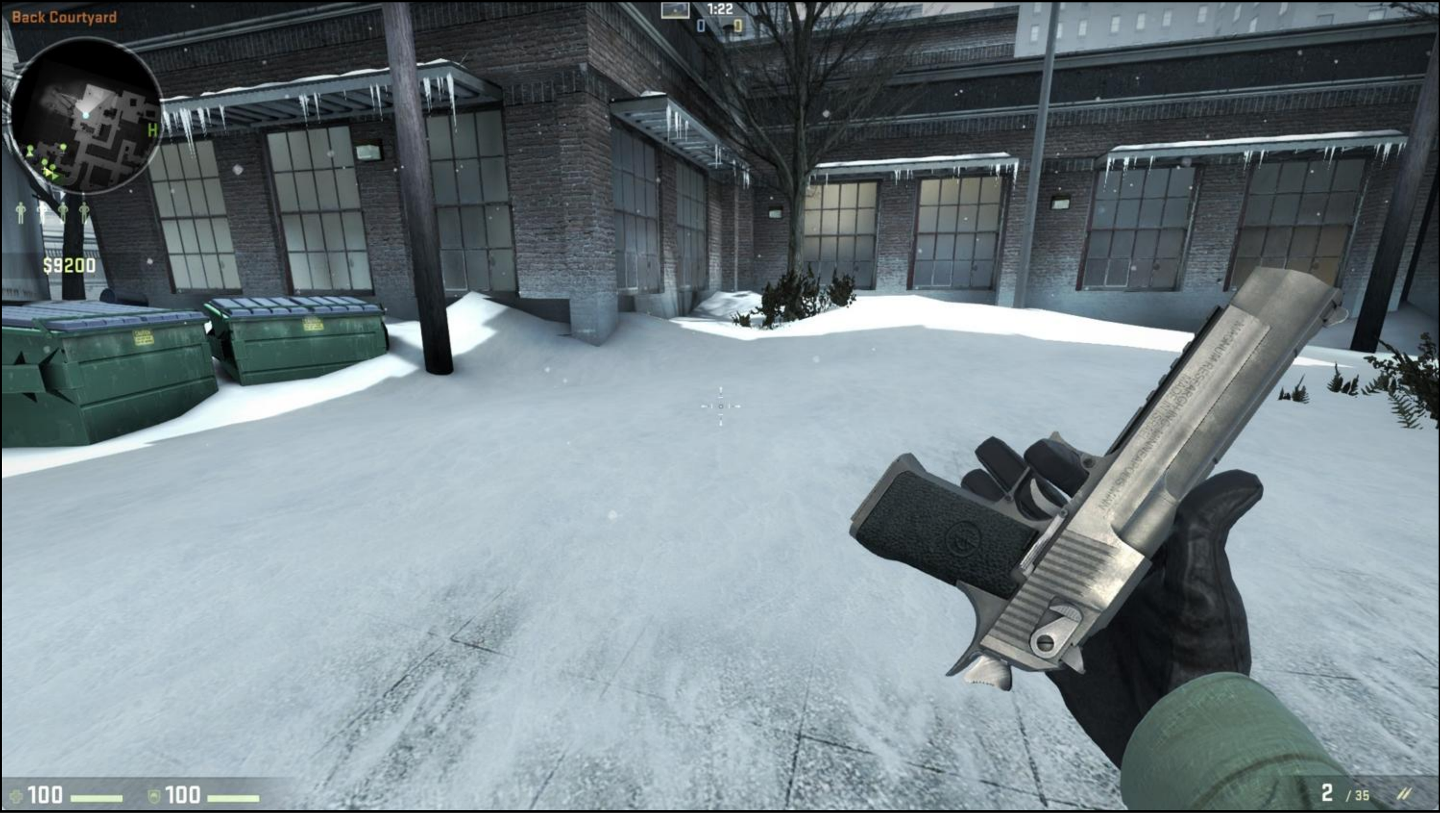


Facets of Luxury

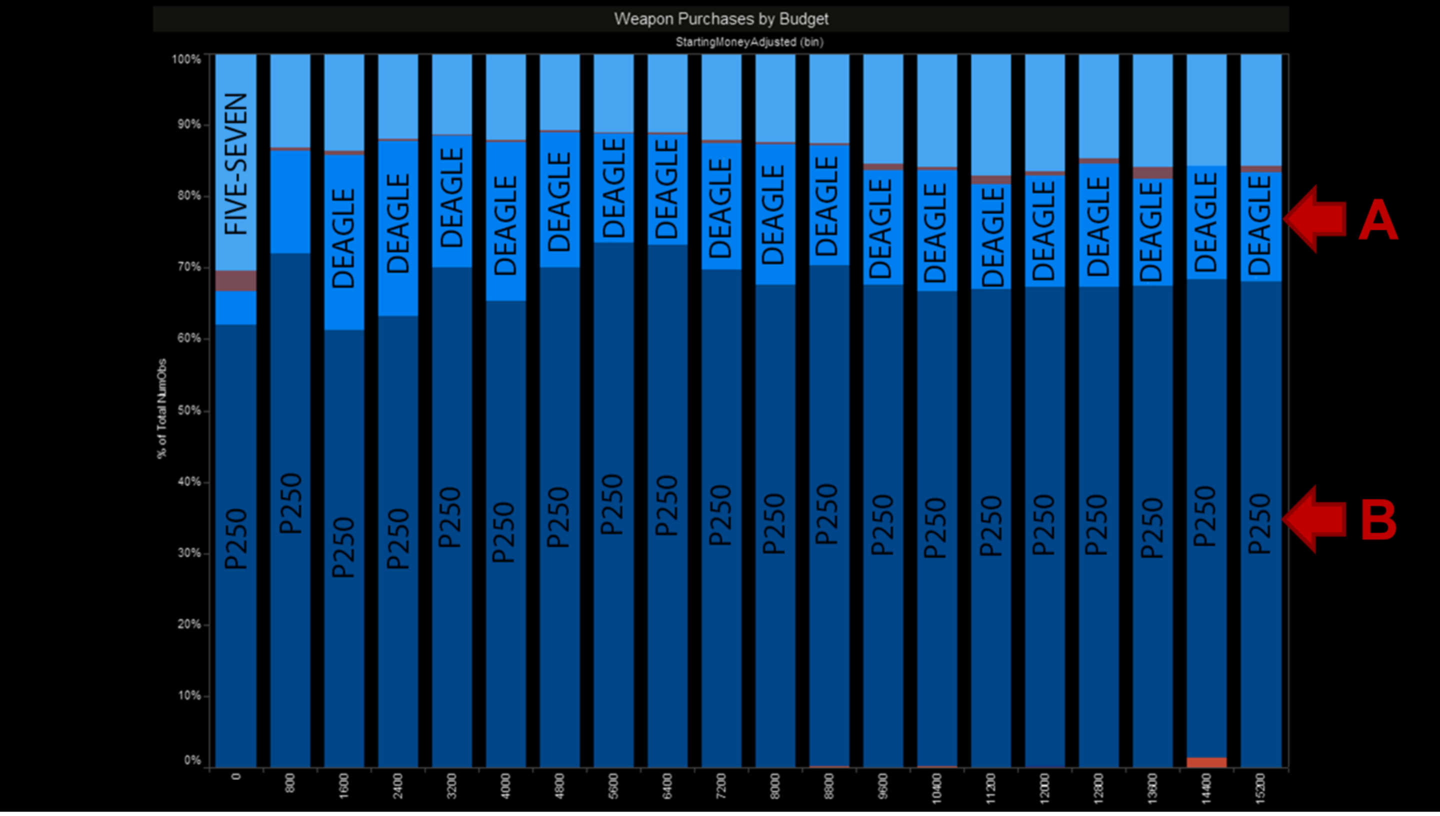
- Conspicuousness
- **Heritage and Personal History**
- Scarcity and Uniqueness
- Excellent Quality
- Aesthetics

The next item on our list, Heritage and Personal History, encompasses things like historic value, nostalgia and personal meaning. It's easy to see how those kinds of factors influence real-world items: venerable brands, items that belonged to famous personages, family heirlooms.

In a video game? Maybe it's less obvious, but *Counter-Strike* as a franchise has been around for 14 years.

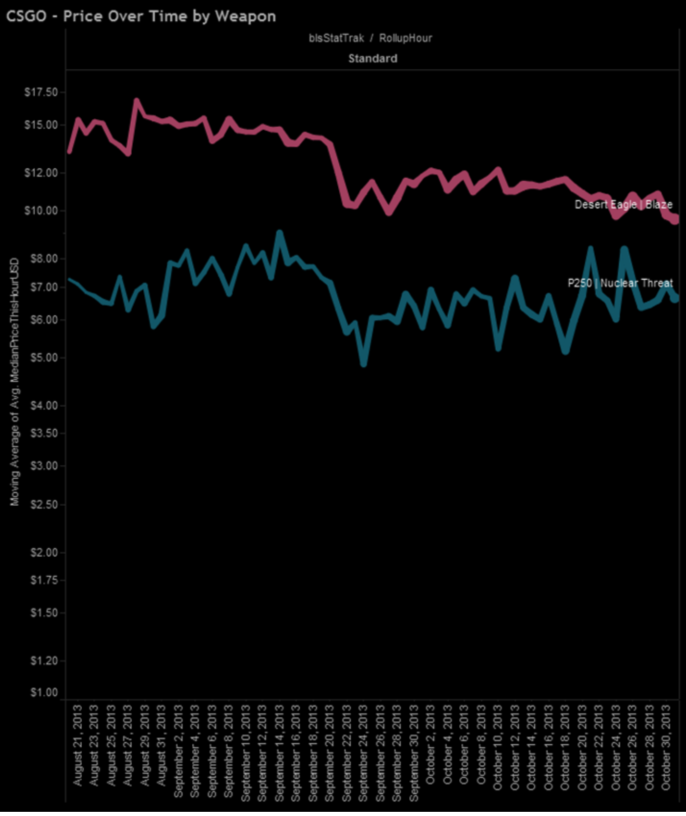


This is the Desert Eagle, know by its nickname, the Deagle. It’s been a very popular choice in previous iterations of the game where it was a weapon with a high utility.



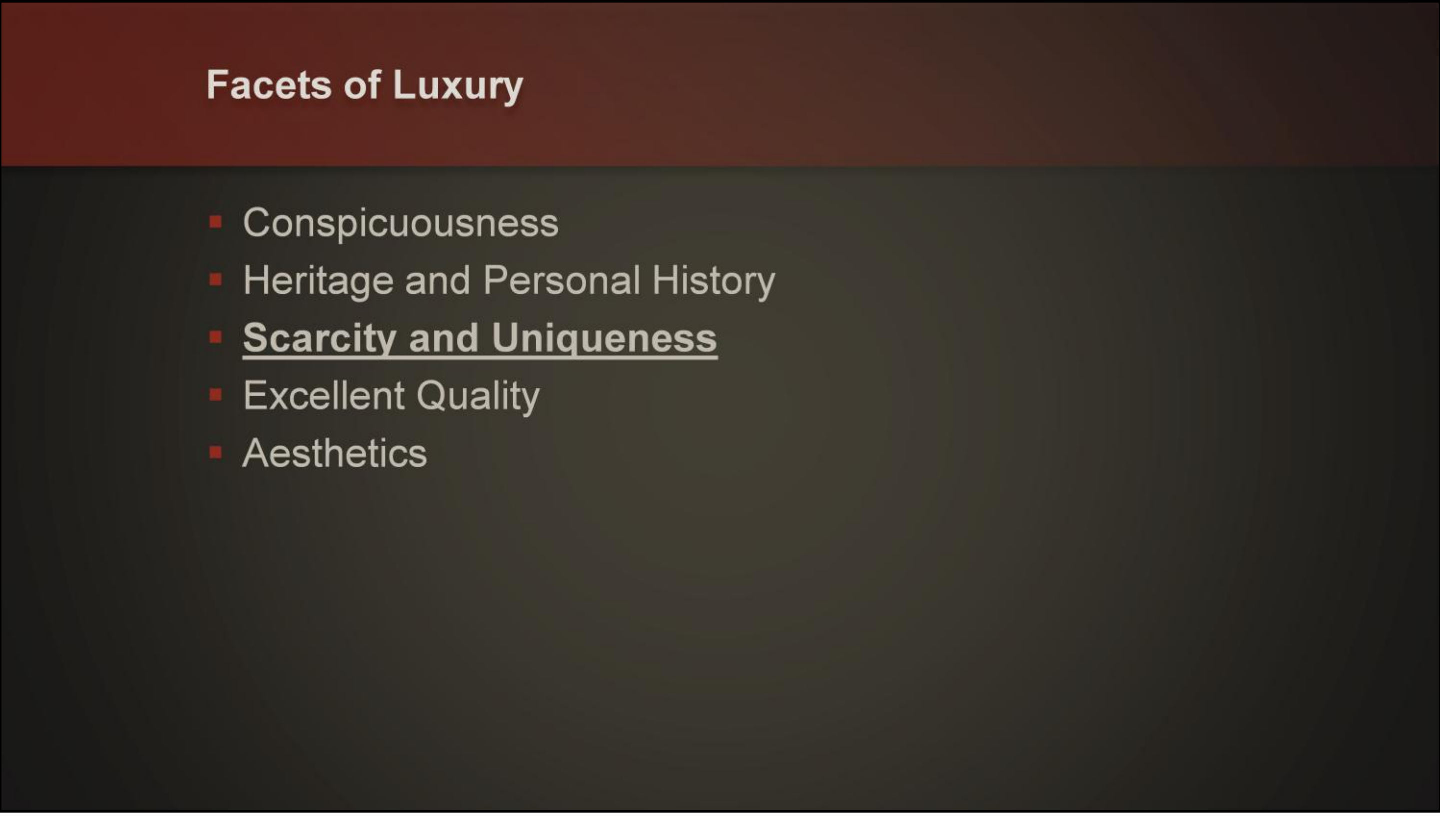
We’ve made some changes since, but back in October when we shipped our first round of content, its utility was not especially high compared to other pistols (A). The P250, for example, is a weapon with much more utility (B).

In previous versions of the game, this graph would have looked very different. The Deagle would have had a very high utility.



Since that’s the case, it’s understandable that the Deagle retains its high value. Here’s an example showing finishes on two weapons that have the same scarcity. The item on top is the deagle, despite its current status as a lower utility weapon. The P250, even though it now has more utility than the Deagle, has relatively less value. The price difference is pretty striking. *Counter-Strike*’s heritage and players’ personal history with this weapon is influencing its value, in this case by close to a factor of two.

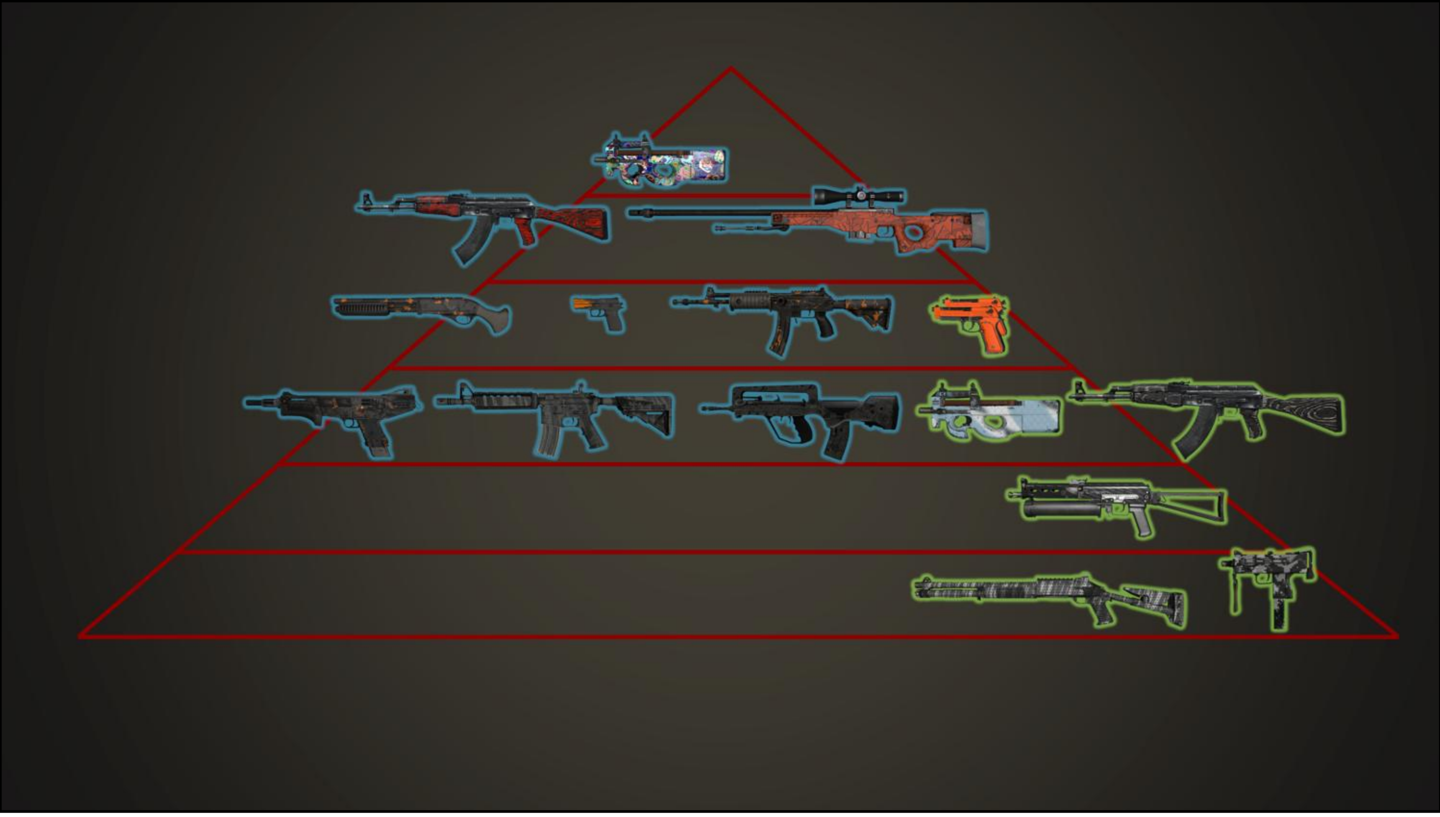
So heritage and personal history are also a very strong influence.



Facets of Luxury

- Conspicuousness
- Heritage and Personal History
- **Scarcity and Uniqueness**
- Excellent Quality
- Aesthetics

Scarcity is easy to explain. The less of something there is, the more value each unit has.



We sorted our items into collections.

Each collection contains items with different quality tiers.

The quality tiers also roughly correlate with scarcity. So items towards the top of the collection are generally more rare than items from the bottom.

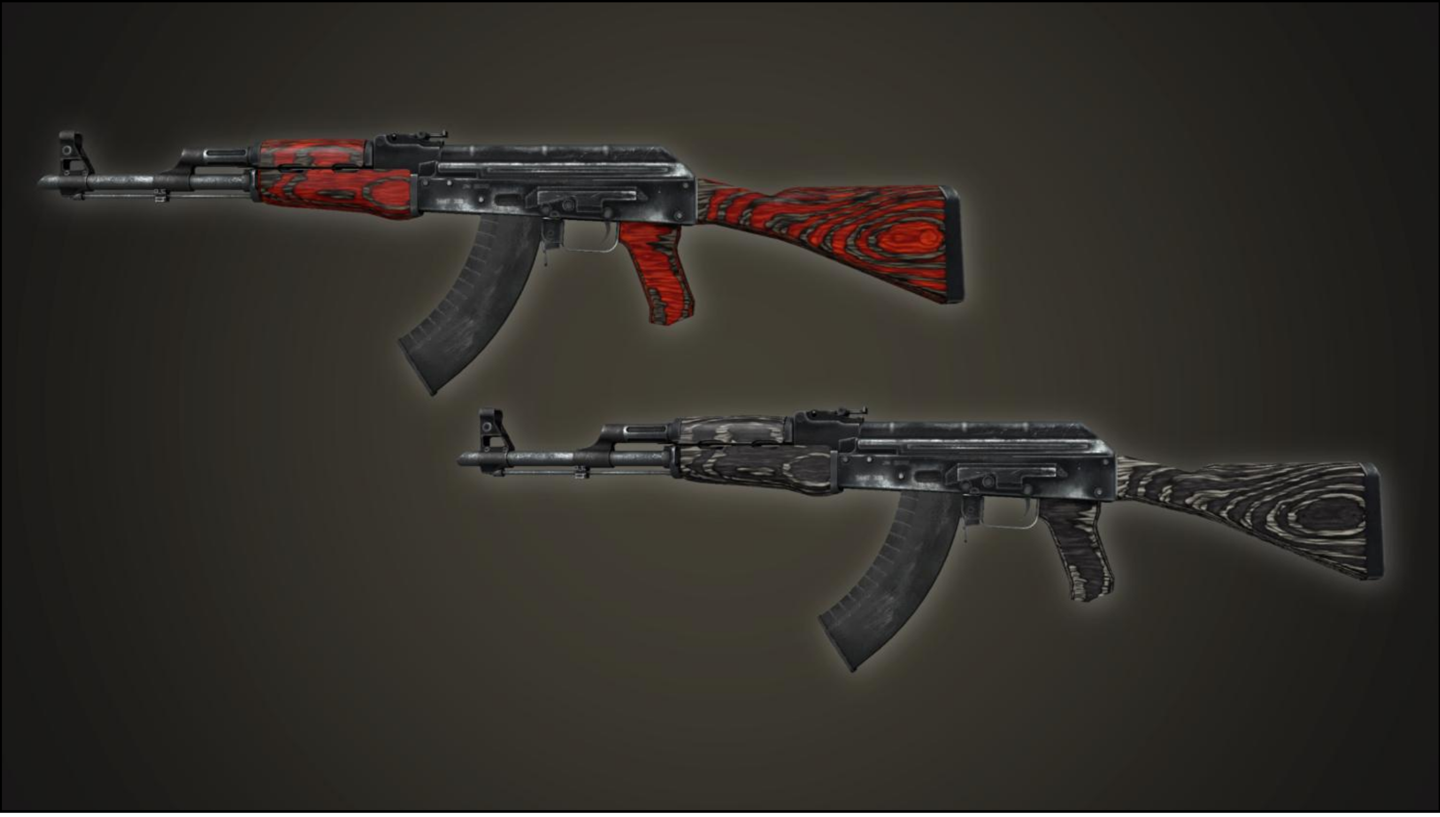
We have two types of drops: individual items and cases. Cases can be opened with a key, and give you a random item from a predefined collection.

The collection outlined in blue is from a case, and the collection outlined in green is made up of individual item drops.

We control the rate at which both cases and single items are dropped. What we don't control, though, is the rate at which cases are *opened.* We'd used our best guess at case opening rates to balance the number of drops so that quality and scarcity would stay correlated.

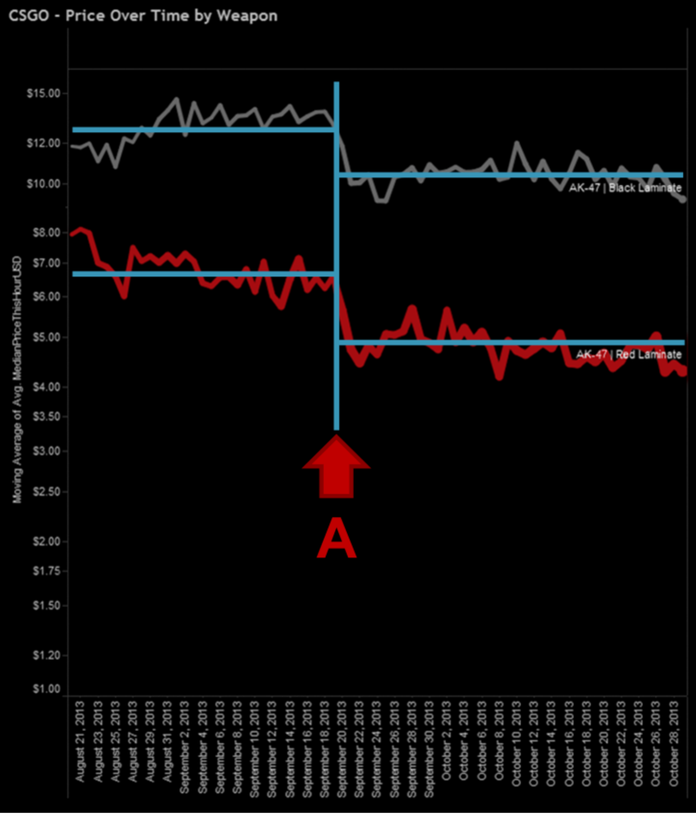
In our first couple of months, our customers opened way more cases then we'd predicted, so case items were actually more common than drops. As a result, the quality tier of an item didn't accurately represent its scarcity relative to items with a different provenance.

There are two items in particular from these collections to call to your attention.



The first is the Red Laminate AK-47. The second is the Black Laminate AK-47. Their quality tiers are different – the Red is two quality tiers above the Black. Aesthetically speaking they're very similar. If anything, the Red Laminate is more striking, making that two ways in which we would have expected the Red to have a higher value.

But due to the volume of cases opened, there are actually more Red Laminates in the world than Black ones.



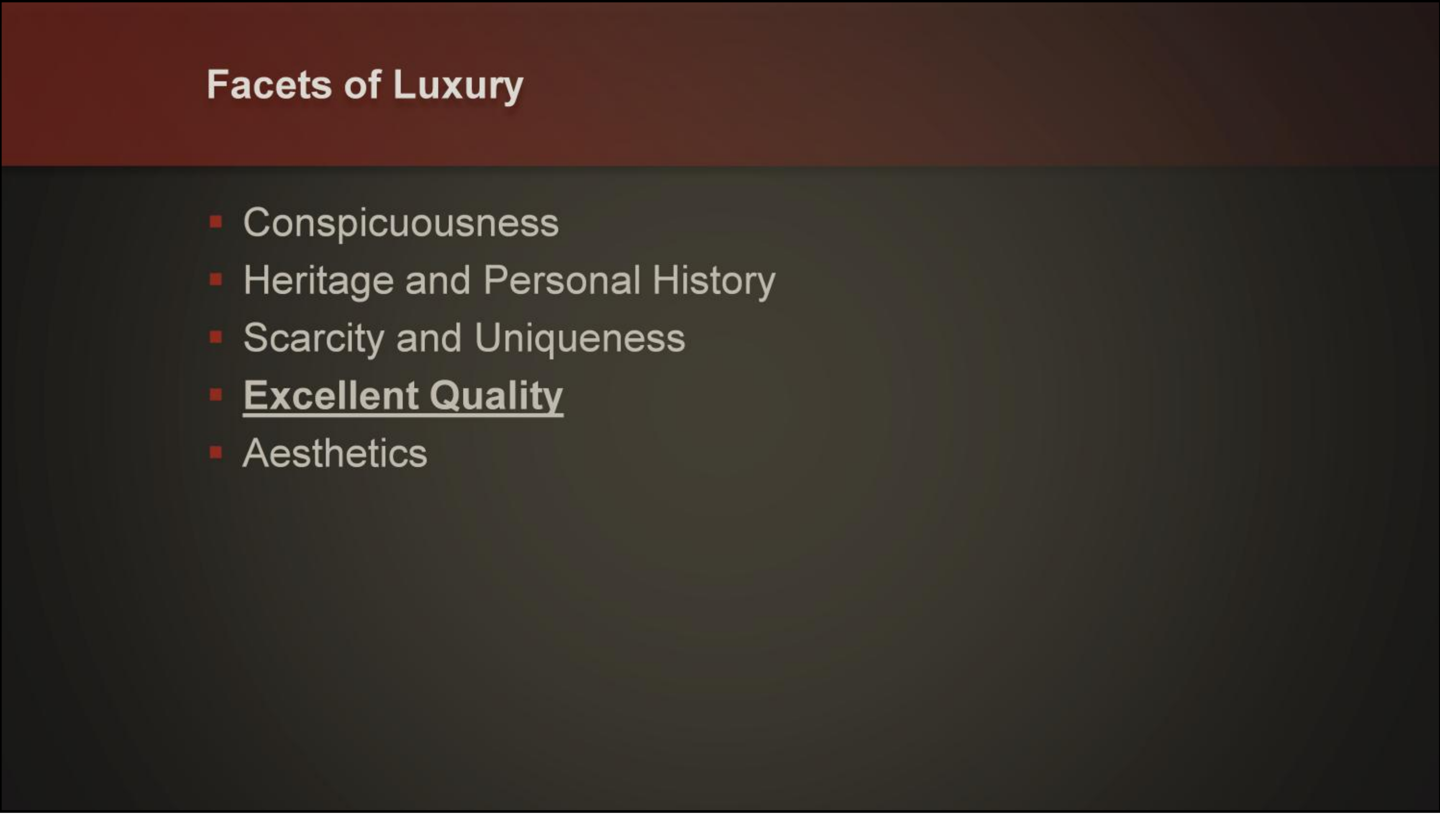
The Red Laminate AK-47, which we expected to have a high value, is consistently around four dollars cheaper than the Black.

Exacerbating the effect is that the Red Laminate is listed at a higher proportion relative to the total number of them in the world. If you own a Red Laminate, you're twice as likely to list it on the marketplace as the Black.

Scarcity's influence on price is pretty extreme. It's one of the clearest predictors of value that we've got.

There's another interesting feature on this graph, and it has to do with uniqueness, another facet of scarcity. See the dip at (A), where the prices for both weapons got depressed?

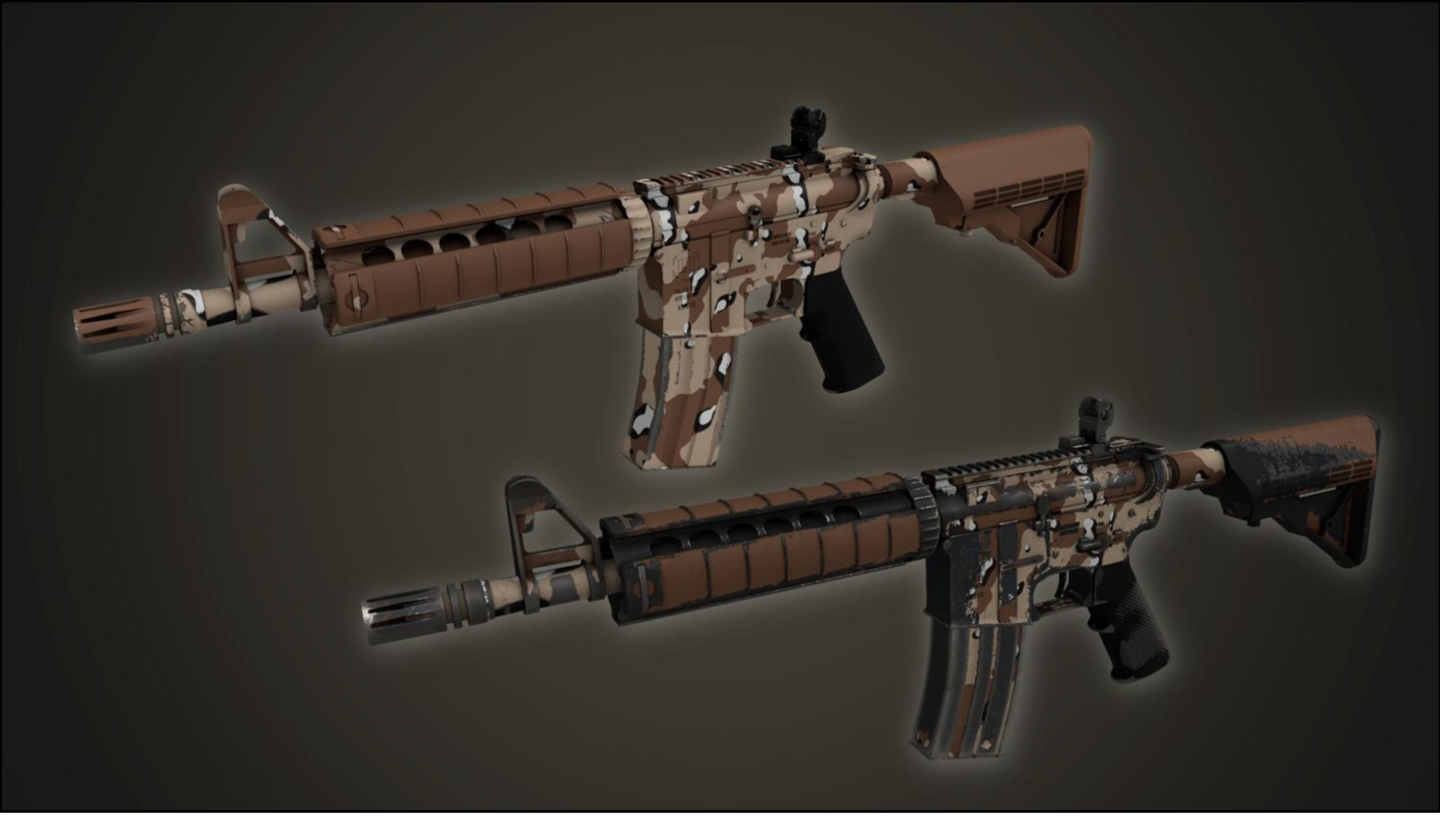
At this point, we shipped a bunch of new finishes, including some for this weapon, the AK-47. And so these finishes lost value relative to the new, more unique and more scarce finishes. Over time we've found that the effect of a new finish entering the pool is most striking when the pool is small. The more items we ship, the smaller the impact of this effect.



Facets of Luxury

- Conspicuousness
- Heritage and Personal History
- Scarcity and Uniqueness
- **Excellent Quality**
- Aesthetics

In the real world, quality assessment would be polysensual. Kinetic feel and smell would give indications of the quality of materials and assembly. In-game there are fewer cues, and they’re only visual and descriptive. But we discovered, to our surprise, that they’re just as influential.

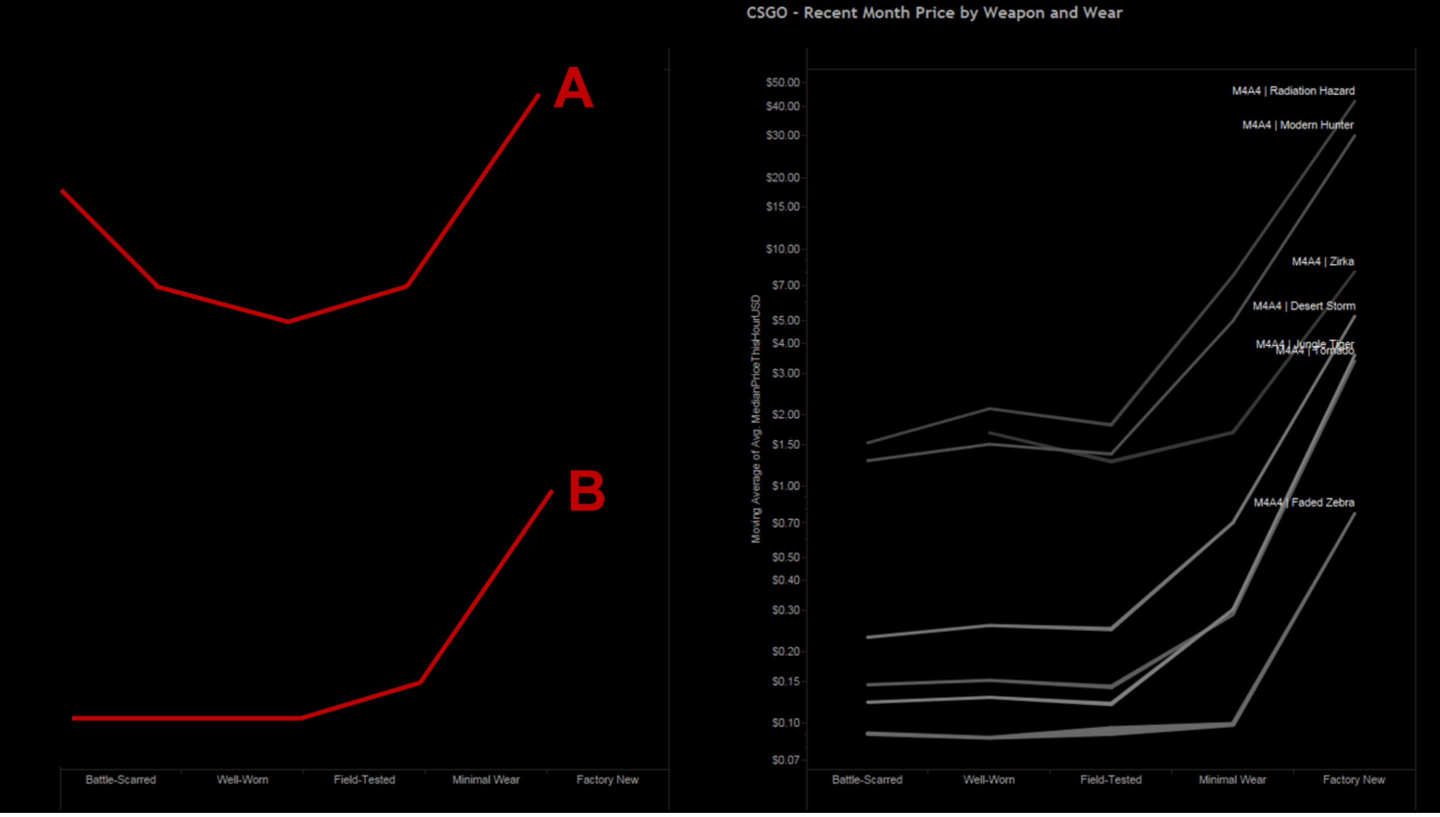


We expected wear would influence desirability. We expected worn weapons to correlate with veteran status, a sort of fictional bonus on top of the aesthetic interest wear provides.

Because of that, we considered making weapons wear over time, so the more you use a weapon the more worn it becomes. Pretty much everyone we showed the tech to had the same idea – it was a really popular concept. We wanted players to show off worn weapons as badges of honor. We ended up not doing it for two reasons. First was that, aesthetically speaking, some weapons looked better with less wear rather than more, and so it would be impossible to always get a weapon to wear towards its ideal state. Second was that for the wear to be meaningful, it had to represent a solid amount of time investment, so the wear might happen too slowly for players to perceive. We didn’t want to affect player’s inventories in a way that was difficult to communicate.

So we didn’t ship that feature, and instead each weapon has a random amount of wear applied to it at grant time. The randomization has a bell curve so that it’s least common to get the most and least worn finishes, and more common to get an item with a medium amount of wear. You could say that the most and least worn finishes have the greatest scarcity.

Given the previous examples, then, we might expect that the most and least worn finishes to have the highest value.

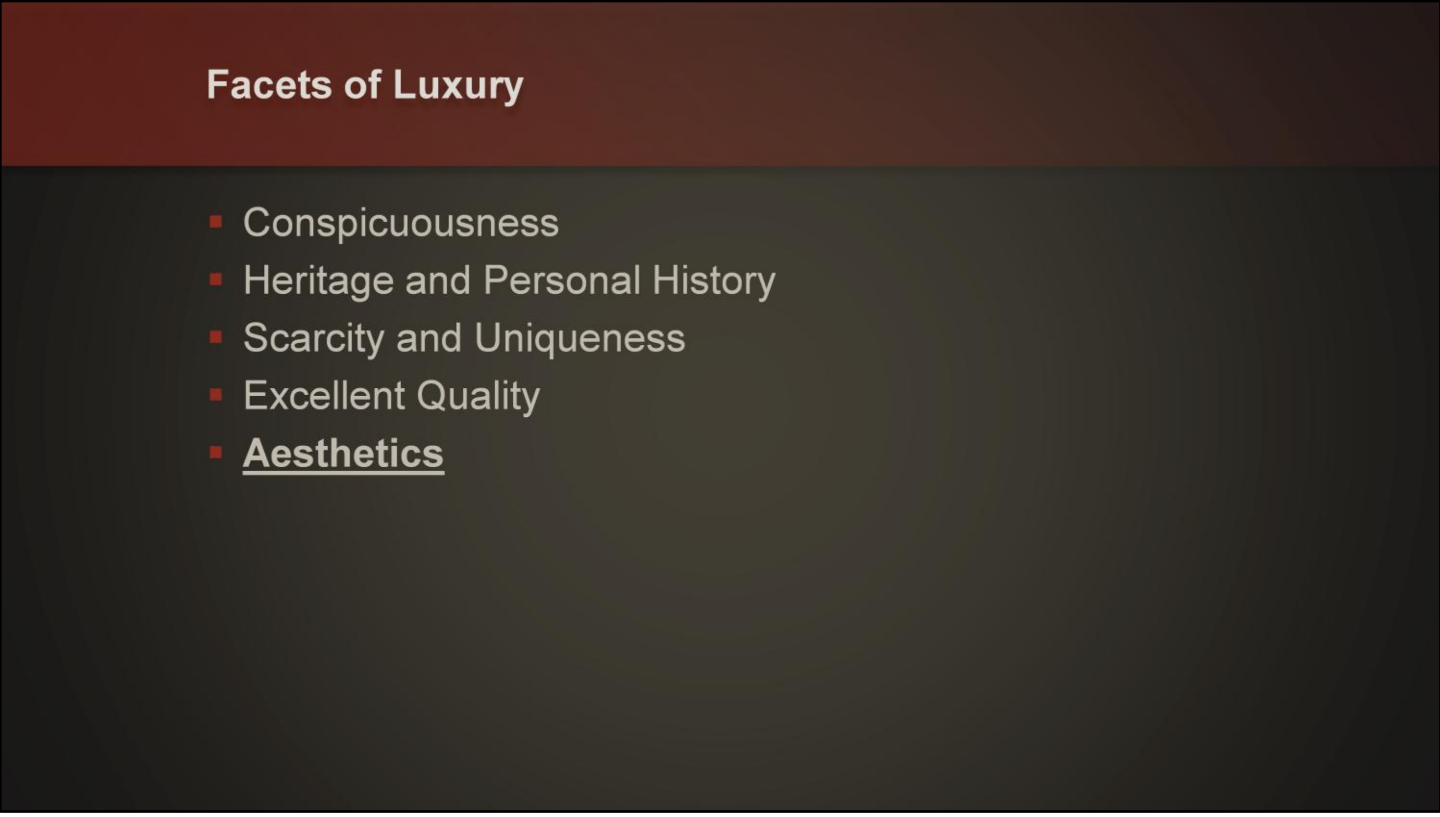


If that were true, our graphs would have this kind of shape (A), with Battle Scarred on the left and Factory New on the right. Instead, we see (B).

Almost without exception, the Battle-Scarred exteriors, the most worn exteriors, are seen as significantly lower quality than the Factory New finishes, and have a correspondingly lower value.

This shows, though, that perception of increased quality impacts valuation.

So creating the scratched aesthetic gave us a good range of value inside of even the same skin, even if we’d imagined the value spread exactly backwards.



Facets of Luxury

- Conspicuousness
- Heritage and Personal History
- Scarcity and Uniqueness
- Excellent Quality
- **Aesthetics**

Now let’s finally talk about the aesthetics of individual finishes.

Aesthetics in the context of luxury is a consumer’s ability to discriminate value or quality by visual characteristics.

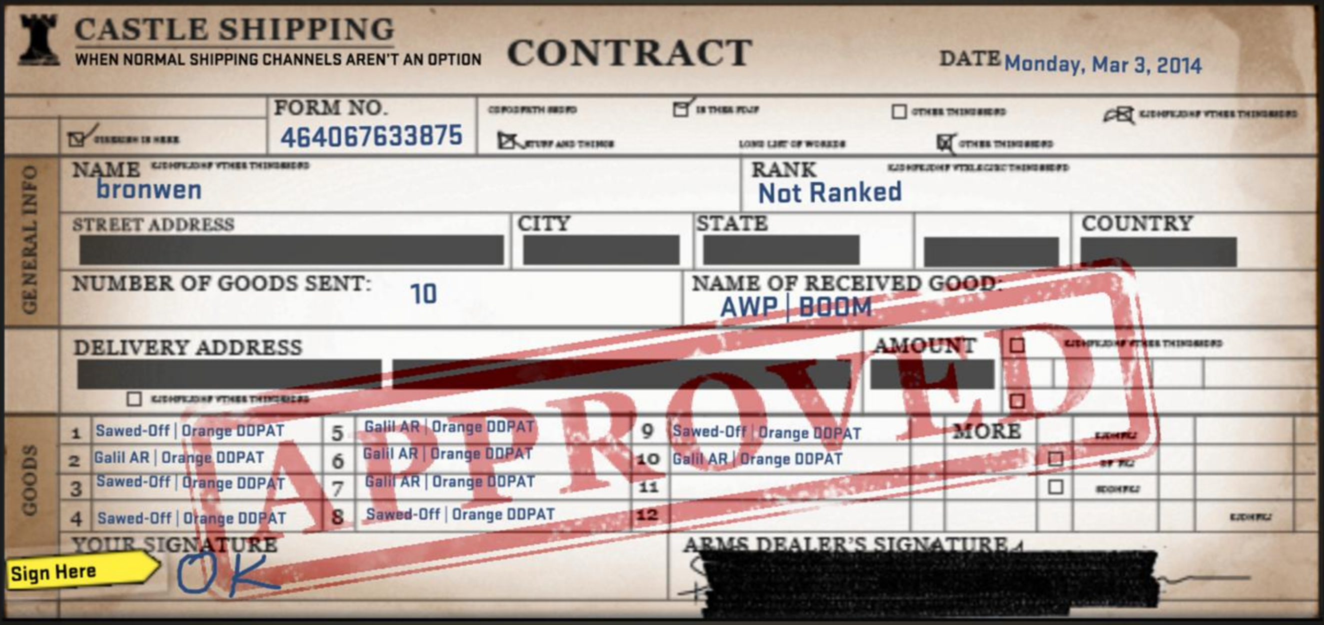
We created a broad range of aesthetics, from subdued military styles to flashy items. Because the game can have a stealth component to the gameplay, we thought maybe players would prefer items that gave them better camouflage.

So how do we test our assumptions?

Just looking at price isn’t going to show us aesthetic preferences. Price is too heavily influenced by the other factors we’ve already discussed.

It’s also worth noting that the price on the marketplace doesn’t capture the total value of items in the world. Less than 2% of items are listed on the marketplace. The rest of them are collected and equipped, and provide a value to their owners that isn’t expressed in dollar amounts.

So let’s look at some non-market data.



CASTLE SHIPPING
WHEN NORMAL SHIPPING CHANNELS AREN'T AN OPTION

CONTRACT

DATE Monday, Mar 3, 2014

FORM NO. 464067633875

☒ STUFF AND THINGS

LONG LIST OF WORDS

☐ OTHER THINGSBSRD

☒ OTHER THINGSBSRD

GENERAL INFO

NAME bronwen

RANK Not Ranked

STREET ADDRESS [redacted]

CITY [redacted]

STATE [redacted]

[redacted]

COUNTRY [redacted]

NUMBER OF GOODS SENT: 10

NAME OF RECEIVED GOOD: AWP | BOOM

DELIVERY ADDRESS [redacted]

AMOUNT [redacted]

APPROVED

GOODS

1	Sawed-Off | Orange DDPAT	5	Galil AR | Orange DDPAT	9	Sawed-Off | Orange DDPAT
2	Galil AR | Orange DDPAT	6	Galil AR | Orange DDPAT	10	Galil AR | Orange DDPAT
3	Sawed-Off | Orange DDPAT	7	Galil AR | Orange DDPAT	11	
4	Sawed-Off | Orange DDPAT	8	Sawed-Off | Orange DDPAT	12	

MORE

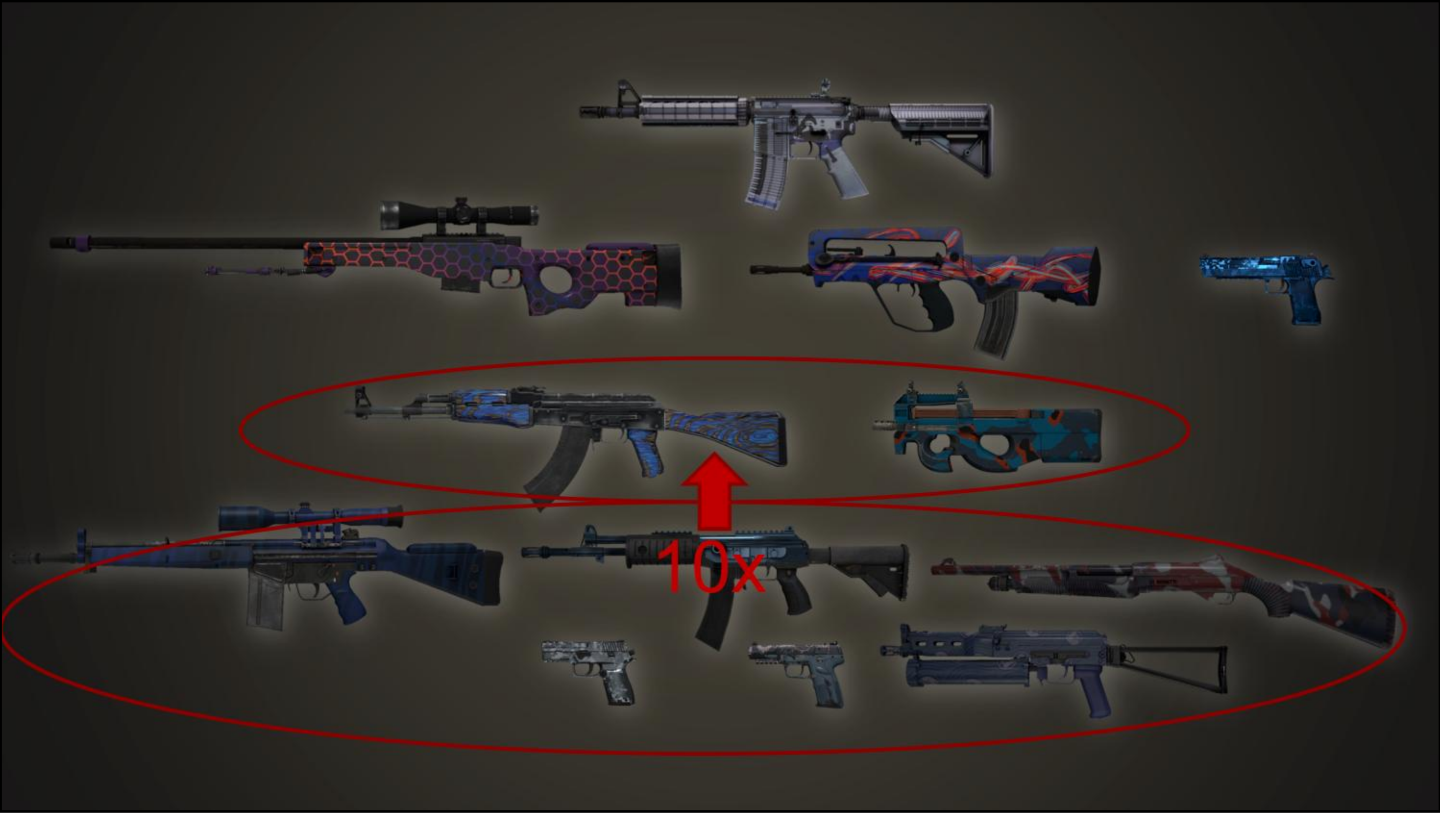
YOUR SIGNATURE

Sign Here

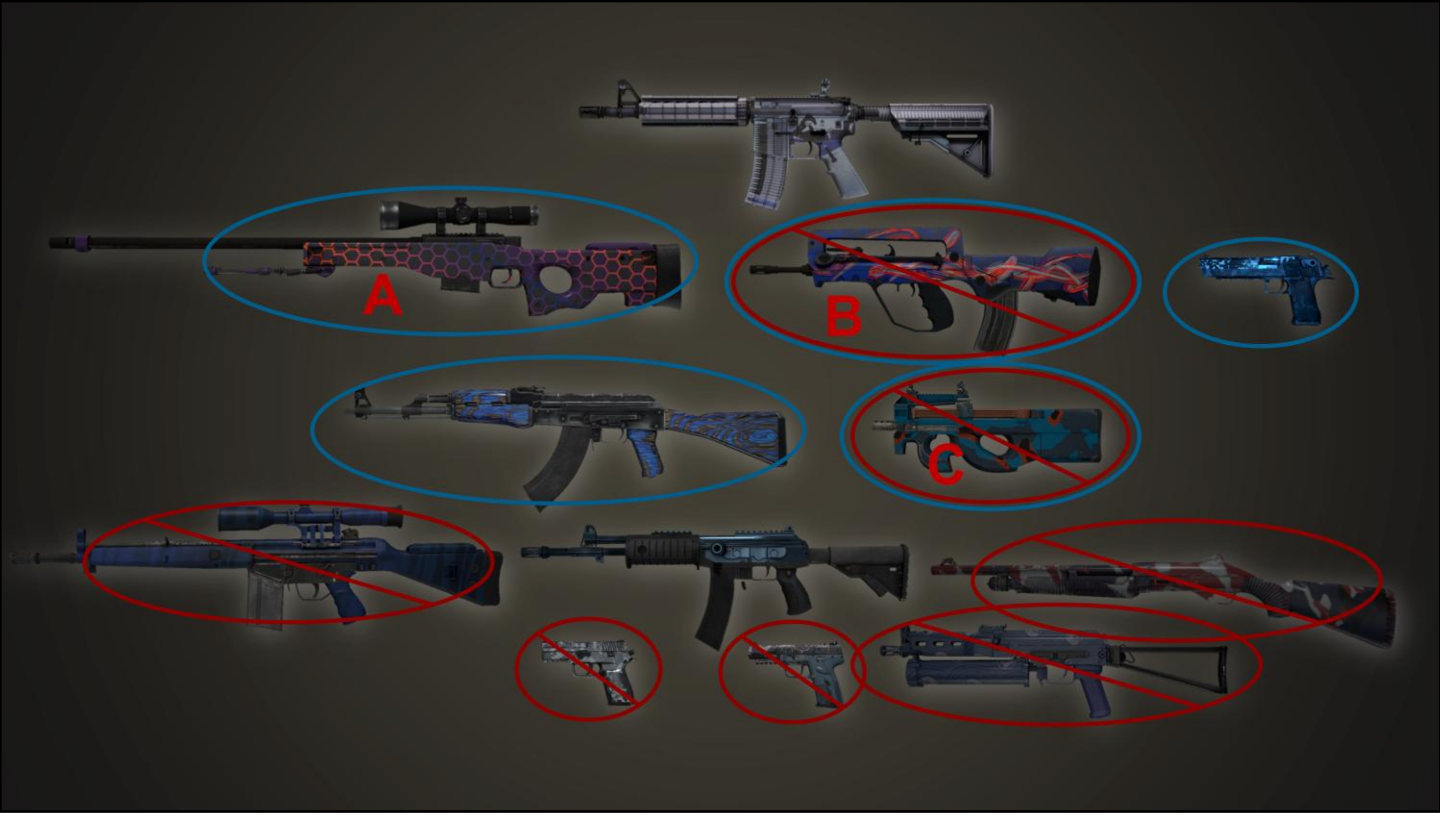
OK

ARMS DEALER'S SIGNATURE [redacted]

We added crafting to the game shortly after the economy went live. Crafting is taking a number of items, and using an in-game tool to convert them to another item. We use the fiction of a contract with the Arms Dealer. It's essentially a way of trading with the game instead of with another player. We use specific contracts that define which items can be crafted into which other items.



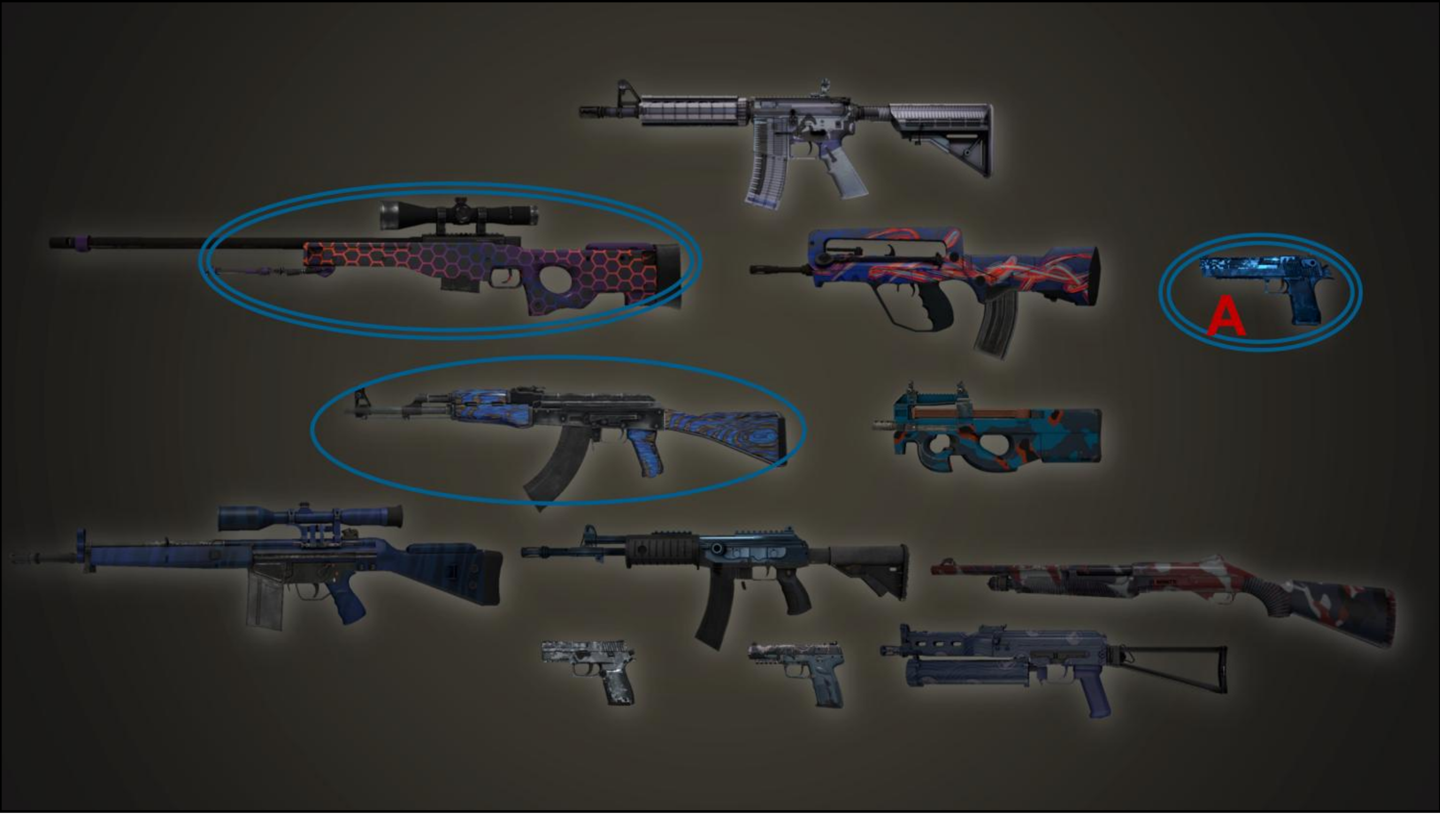
We’ve organized our items into collections, and allow players to trade up inside a collection. Ten items from one quality tier will give you a random item from the next tier up.



We can identify the most desired items by looking at a couple of factors.

First, we can look at crafting data. We can see which items are traded in more frequently than usual (in red).

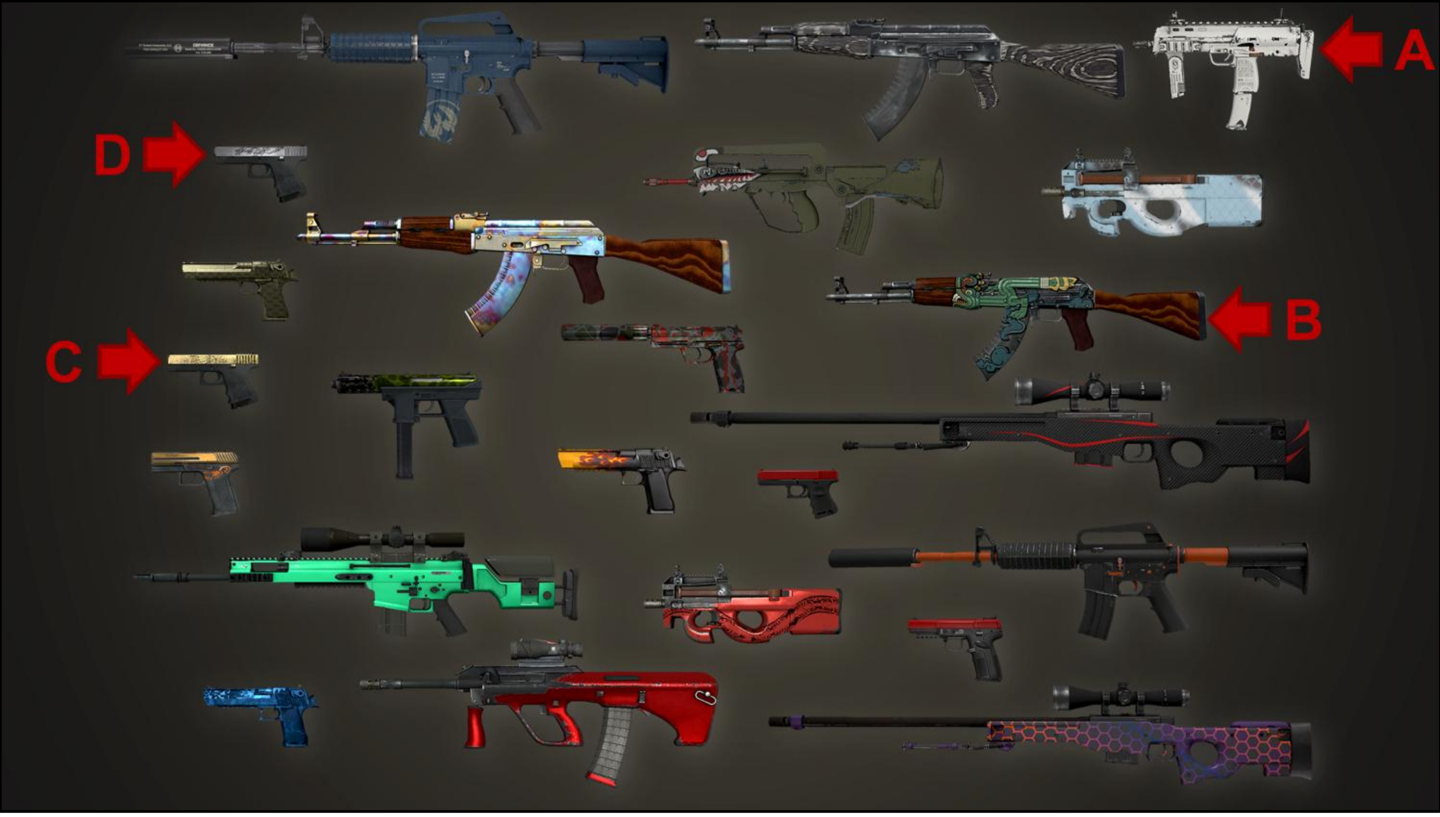
We also look at the origin of an item. If lots of them are created by crafting (in blue), we're either looking at the desired item (A), an item in the same tier as the one you want (B), or an item that is transitional (C): it's been crafted as a stepping stone towards another item in a higher tier.



When we compare the two factors, we can see items that are created by crafting but not subsequently traded in (all circled items).

We can further narrow aesthetic preferences by looking at equip rates. Items that are equipped at an unusually high rate that are also worked towards via crafting give us a good picture of the most loved items (doubly circled items).

For example, if you own a Desert Eagle with the Cobalt Disruption finish (A), even if you own lots of Desert Eagle finishes, you’re still 16% more likely to equip it than normal. If we look at only the craft targets that have such a high equip rate, we can really narrow the field.



If we list all items with these characteristics, we start to see a pattern.

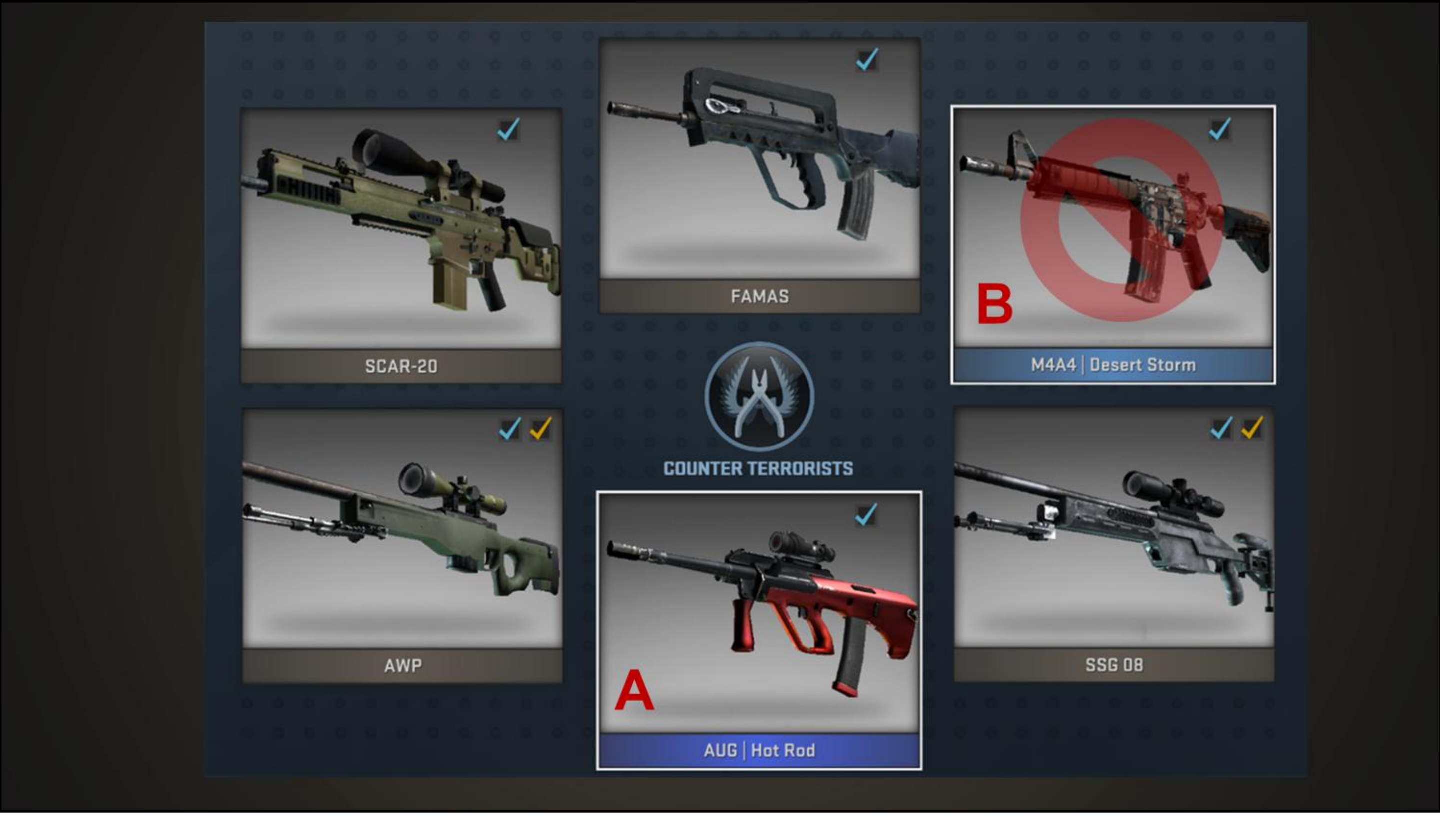
People like anodized finishes. More than half of these finishes are anodized or metallic. They also like saturated colors on black, very bright finishes, and weapons with a sleek, spy-fiction aesthetic.

But the easiest thing to see looking at this image is that most of these weapons have a single predominant color.

Please note that this is very different from looking at price on the marketplace. Although some of these items do have a high price, there are many that don't. In fact, the prices here range from 50 cents (A) all the way up to 30 dollars (B). The scarcity of these items varies quite widely, too. The total number of these items that exist in the world varies a lot, from 6,000 (C) to 156,000 (D).

So looking at the crafting and equip rates of these items gives us a new window into what customers value about our content.

That covers the high end of our aesthetic range. What about the low end?



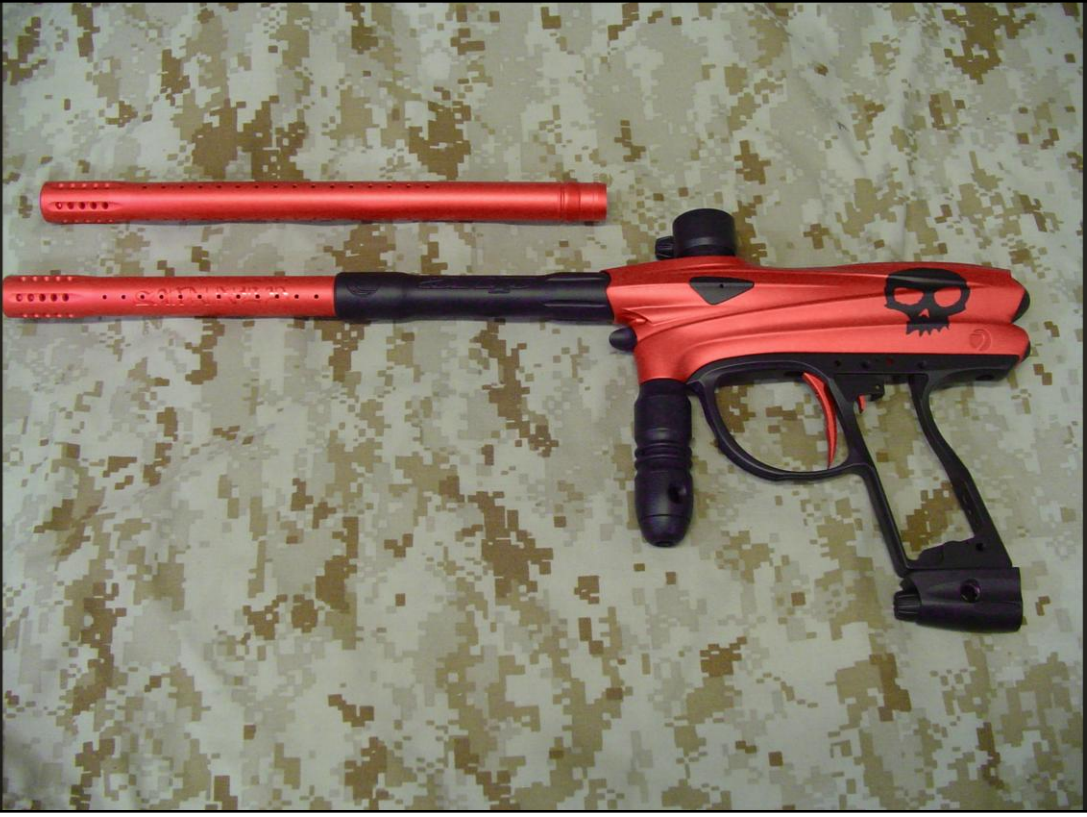
Everyone starts with the default loadout. As you play you get drops of weapons with finishes. If you don't have a big inventory and you only have one weapon with a finish for a particular loadout slot, 9 out of 10 times you'll equip it (A).

At least, that's true for most finishes. But some buck the trend. We can also see the items that you're a lot less likely to equip, even if your only other option is the default. In other words, you dislike it so much that you prefer the default (B).

Here are the worst of the worst. These are all camouflages, or use camouflage-inspired colors. In fact, they're all items that look like…

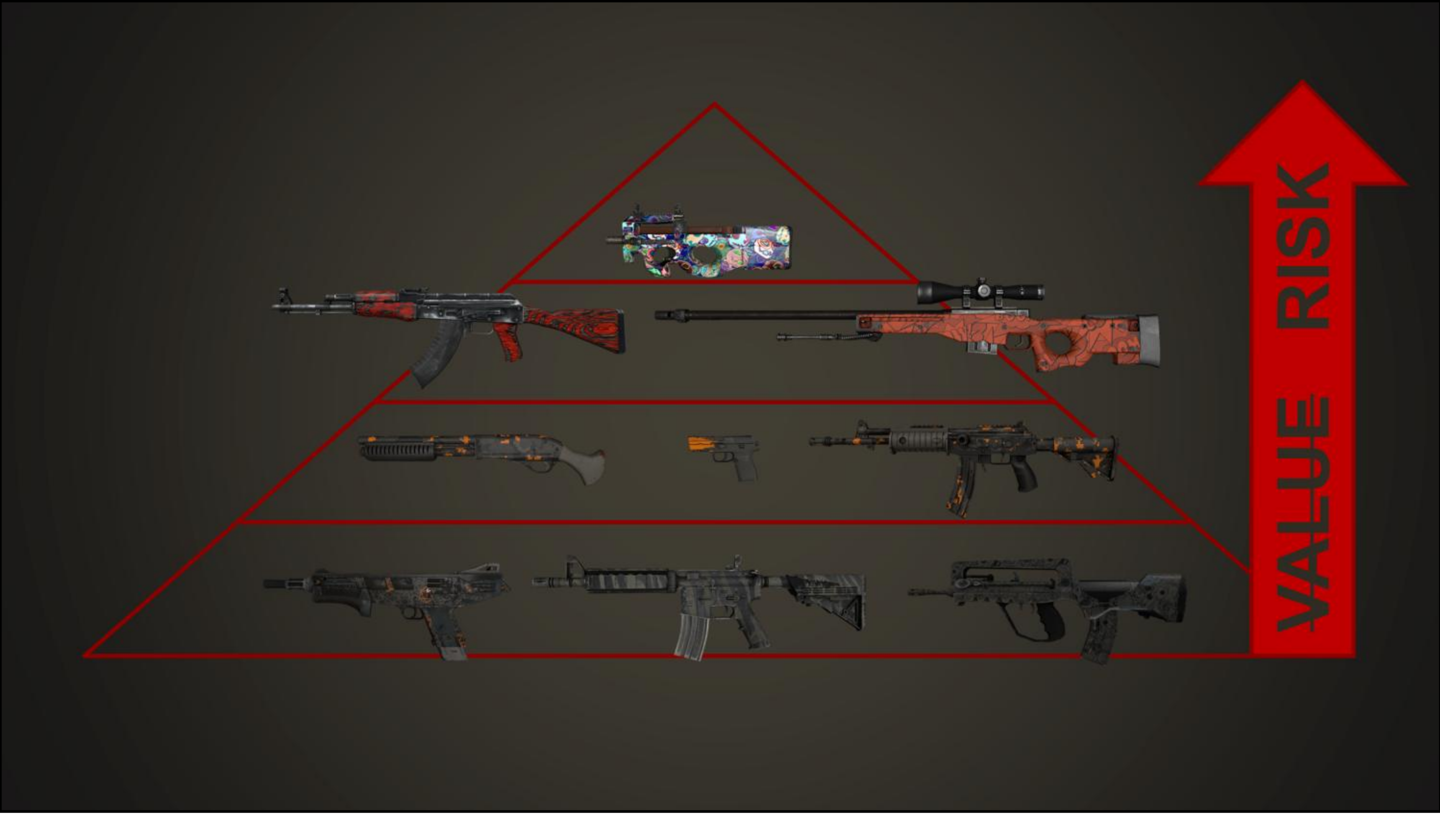
…our original inspiration.

So although we started off thinking military camouflage was really cool…



…it turns out what our community really values are finishes that look more like paintball guns.

We needed a reminder that although *Counter-Strike* is military inspired, it’s not a military simulation. It’s a sport. When our customers play, they don’t aspire to be soldiers, they aspire to be elite *Counter-Strike* players. So maybe it’s not that surprising that the closest real-world analogue we’ve got to our preferred aesthetic comes from a sport.

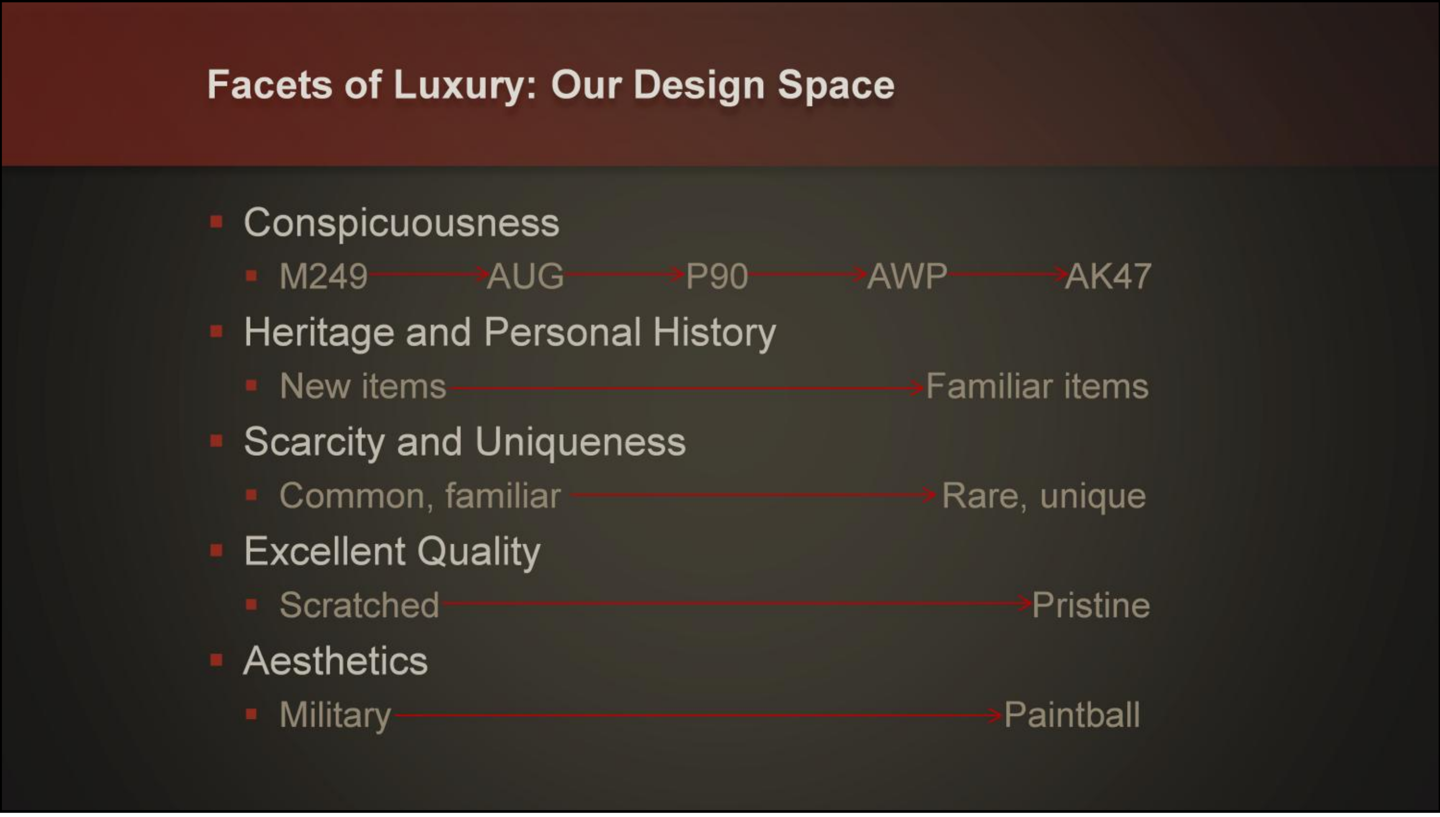


If you looked at our content, you'd assume that we understood this from the start and decided to put the bright, visually salient items into the top of our quality tiers because we knew they had high value.

But actually it was a risk-mitigation strategy. We were genuinely worried that people wouldn't like the bright items. So instead of thinking of these as being sorted by *value*, we originally sorted them by *risk*.

If you got an item from the top tier, you'd be happy because it was rare, even if you didn't like the look. And other customers would be happy because they didn't have to see bright items very often, since there were so few of them.

Now we know we were wrong. Would that have changed our design?



Facets of Luxury: Our Design Space

- Conspicuousness
  - M249 → AUG → P90 → AWP → AK47
- Heritage and Personal History
  - New items → Familiar items
- Scarcity and Uniqueness
  - Common, familiar → Rare, unique
- Excellent Quality
  - Scratched → Pristine
- Aesthetics
  - Military → Paintball

It may have. Let me explain how.

We need to properly map out our design space. We can think of the facets of luxury as the axes. We figured out the bounds of the axes by examining our data:

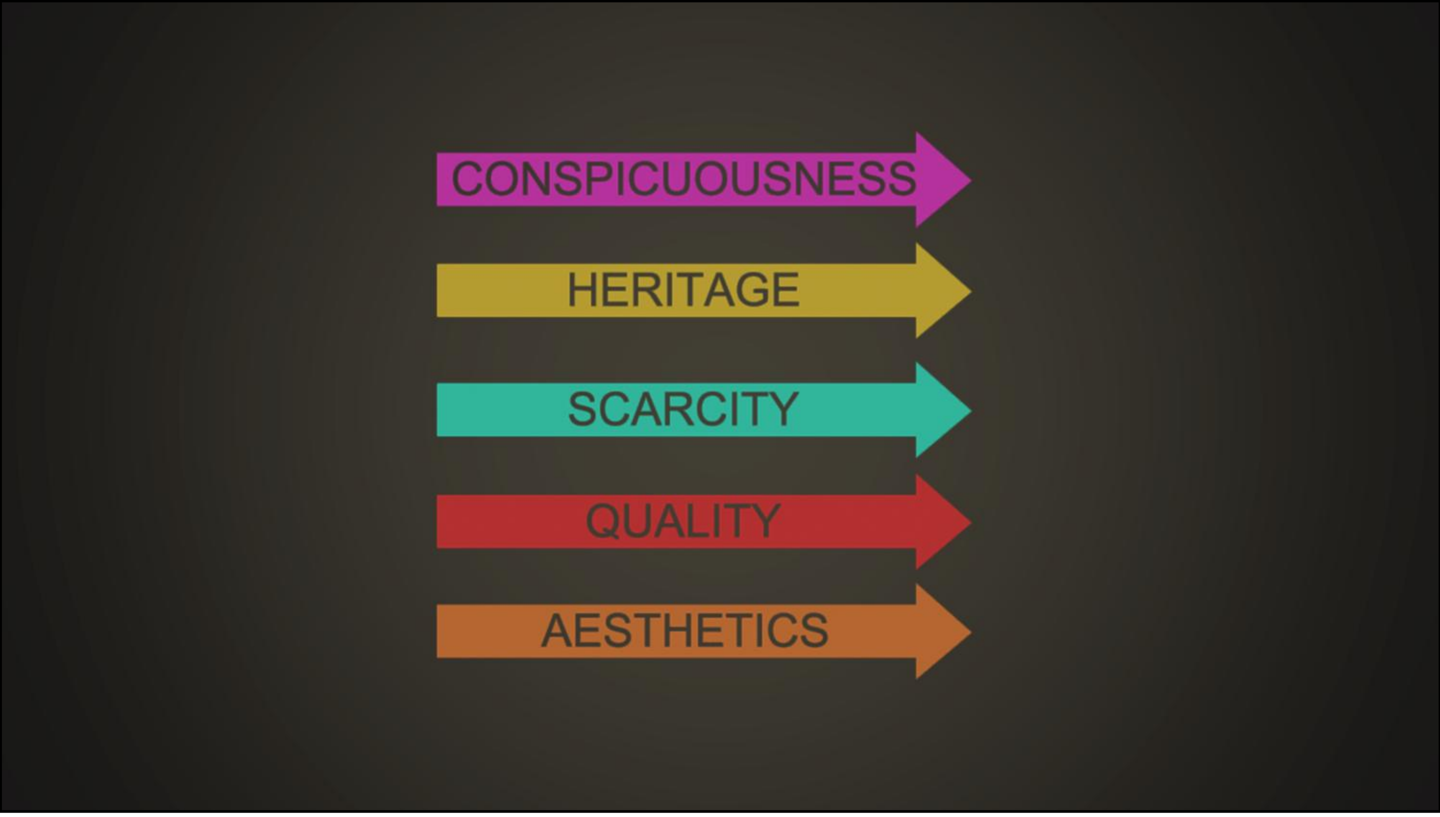
Conspicuousness varies with the utility of the weapon.

Heritage varies from the novel to the familiar.

Scarcity and uniqueness varies with supply as well as how frequently you encounter a finish for a particular weapon.

Quality varies from beat up to factory new.

And our aesthetic range, the most difficult to pin down, varies from our original military aesthetic to visually salient, sports-inspired style.



CONSPICUOUSNESS

HERITAGE

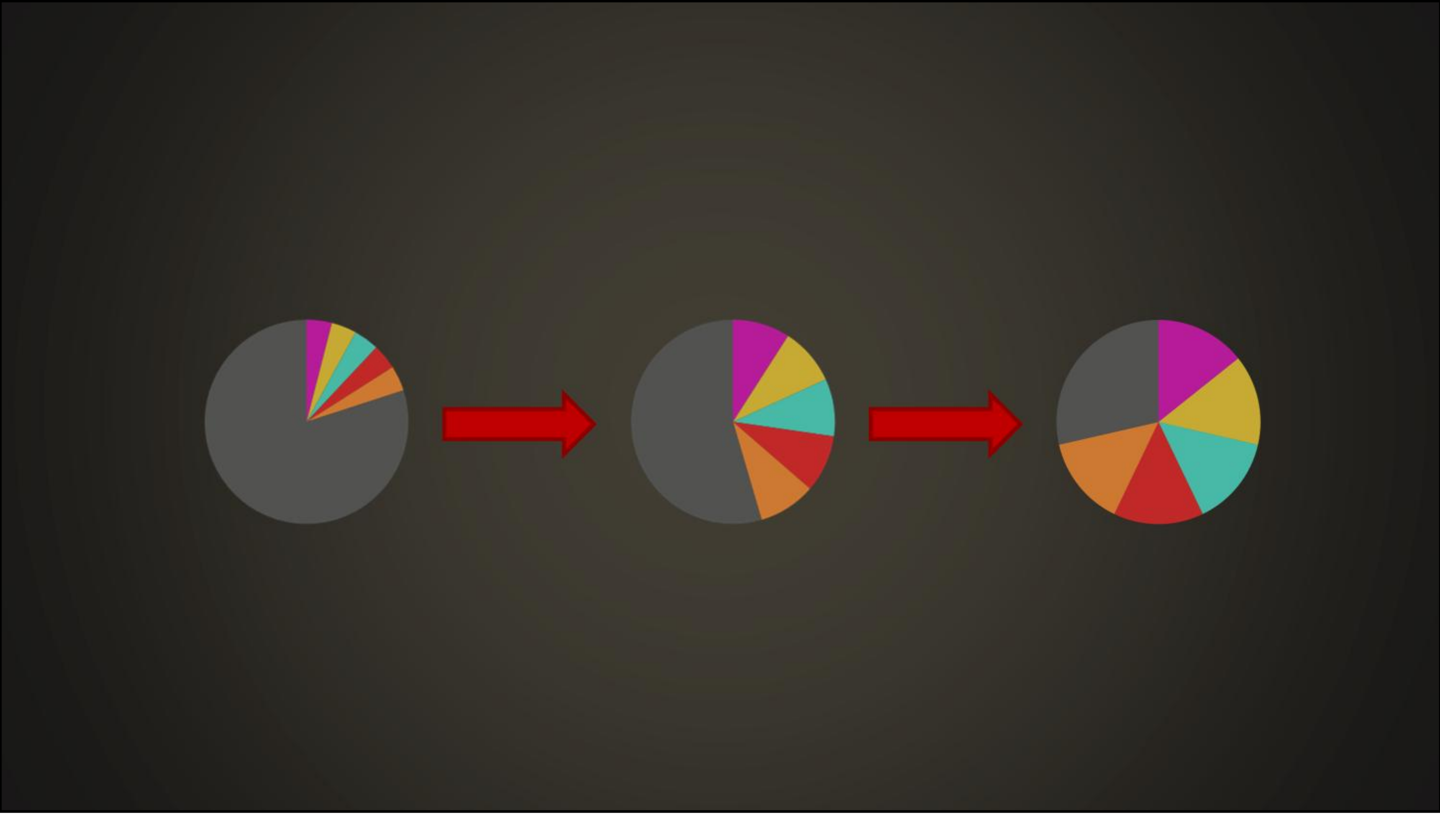
SCARCITY

QUALITY

AESTHETICS

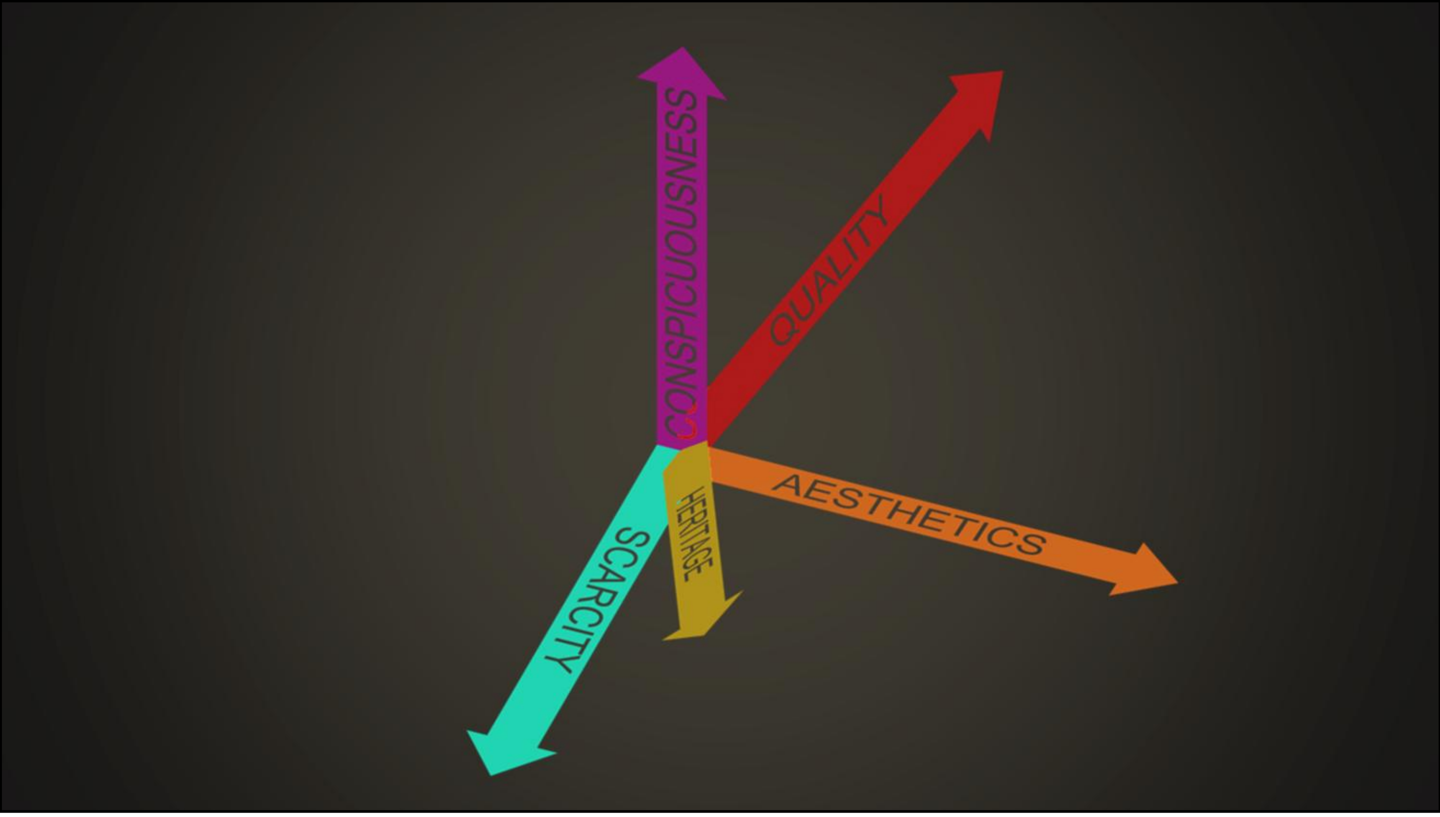
The last thing we want is for all these axes to be aligned, which would have been possible if we didn’t understand how those axes were defined.

In fact, we aligned *scarcity* and *aesthetics* by putting most of our visually salient items at the top of our collections. Fortunately our design space has more than just those two axes, or we could have had a serious problem.

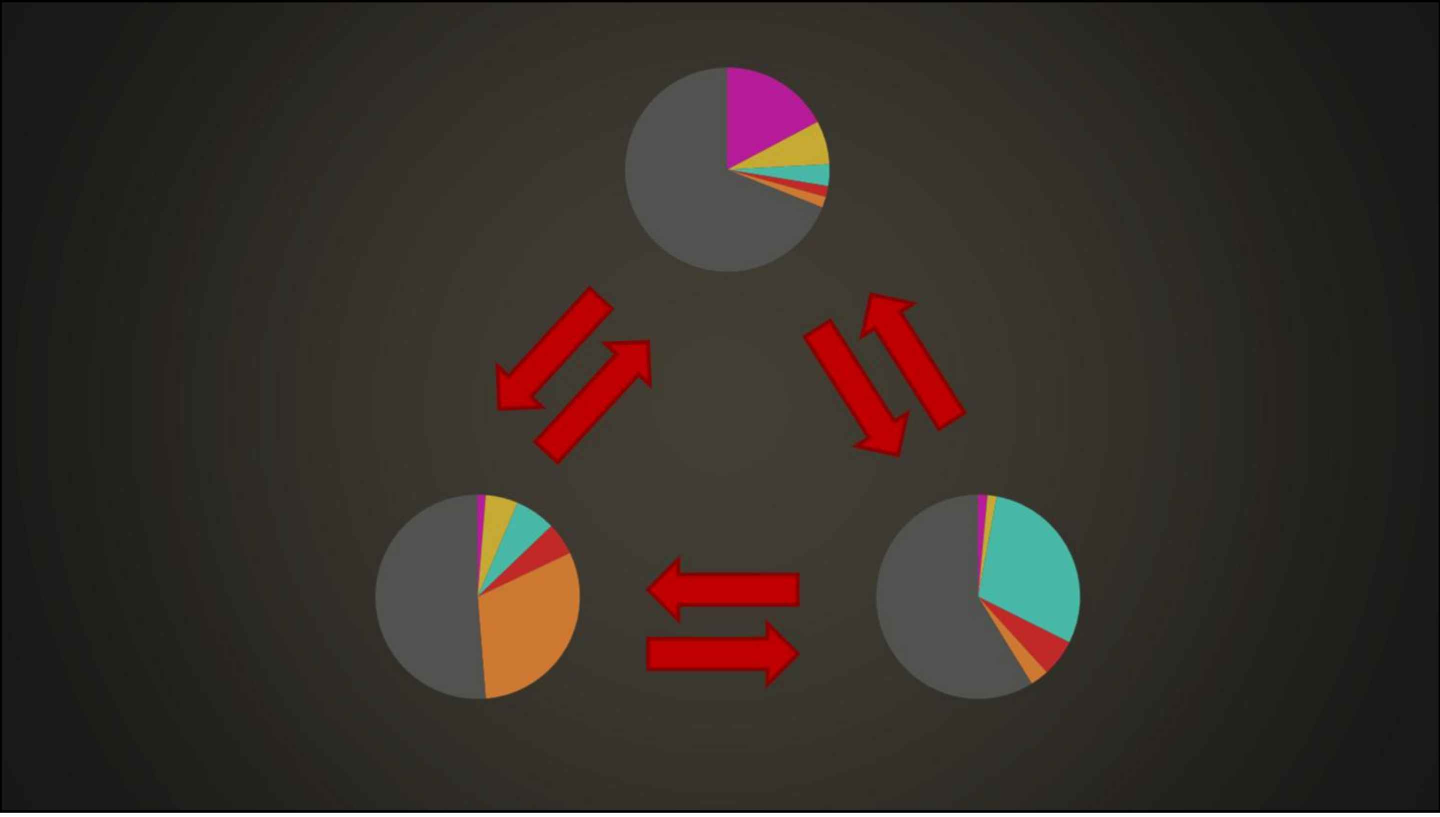


The worst scenario is that each item has all the facets of a luxury good, each in equal degrees. It doesn’t encourage interactions in which players both feel they’re getting the better end of the deal. Someone is always going to be ahead of you on the ladder, and someone behind, and there’s only one direction in which to trade that makes sense.

That’s a great way to stifle the economy.



Instead, we always want to have ways to create variety in every axis. Understanding not only what the axes are but how to place an item along them lets us determine if we're providing enough variety in value. Ideally, there would be no items in the economy with perfectly equal value on all axes.



That range of value is really important. When we create a range of items that have differing value on different axes, players with orthogonal goals can trade items that cater to each other's desires and both players can be equally satisfied.

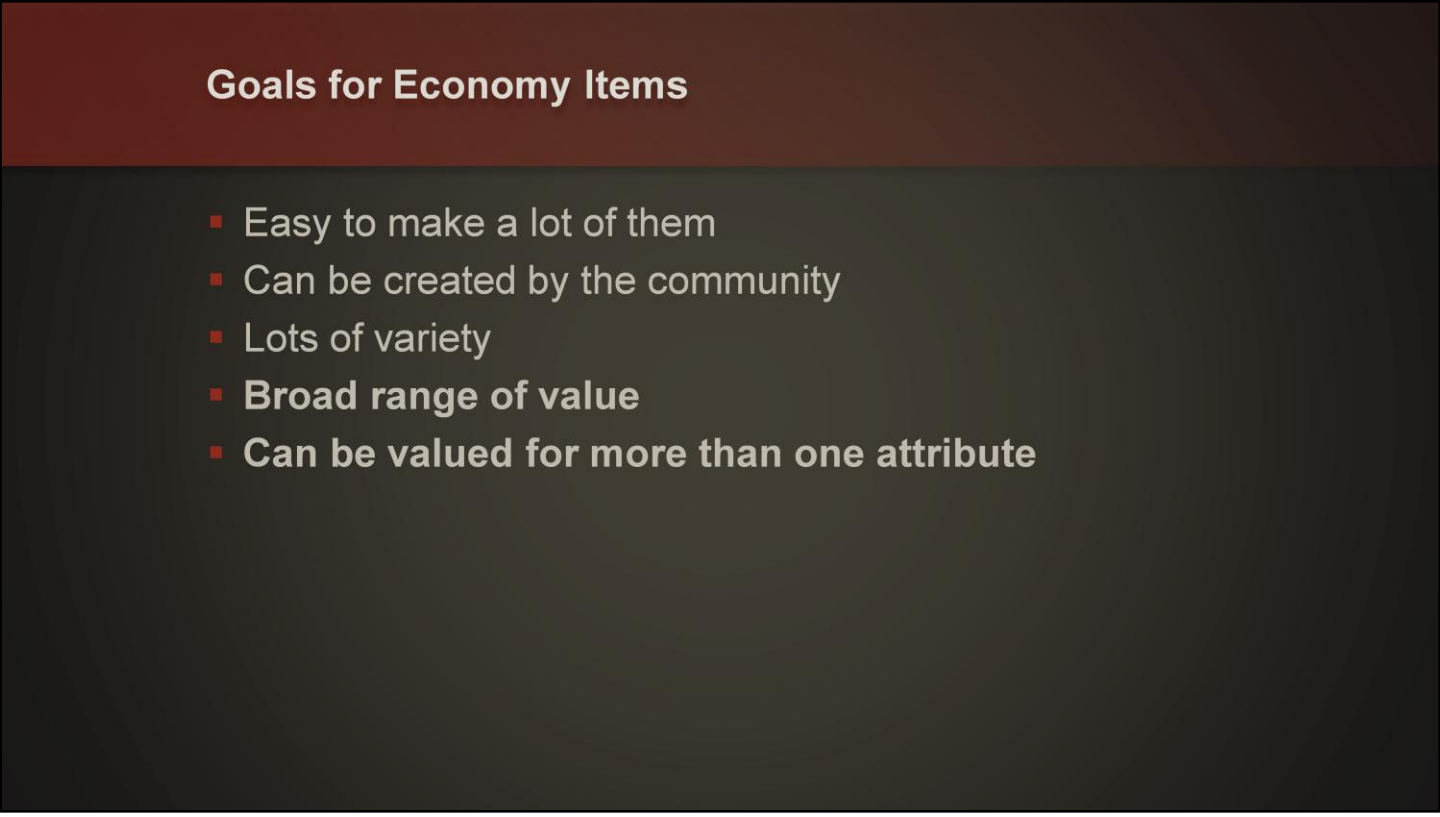
You could, for example, want to collect finishes for your most used weapons. Maybe you don't care how rare they are.

Or you could want to collect rare items, but maybe it doesn't matter if they're bright or shiny.

Or you could want to collect items that have bright colors. Maybe it doesn't matter if they're factory new.

Those trades can happen even if the total value is unequal. And players with different goals can value items in different ways.

There are also people who value the inverse of the expected. Some players do collect beat-up, military-themed finishes, so even our worst performers on the marketplace can make many customers happy.

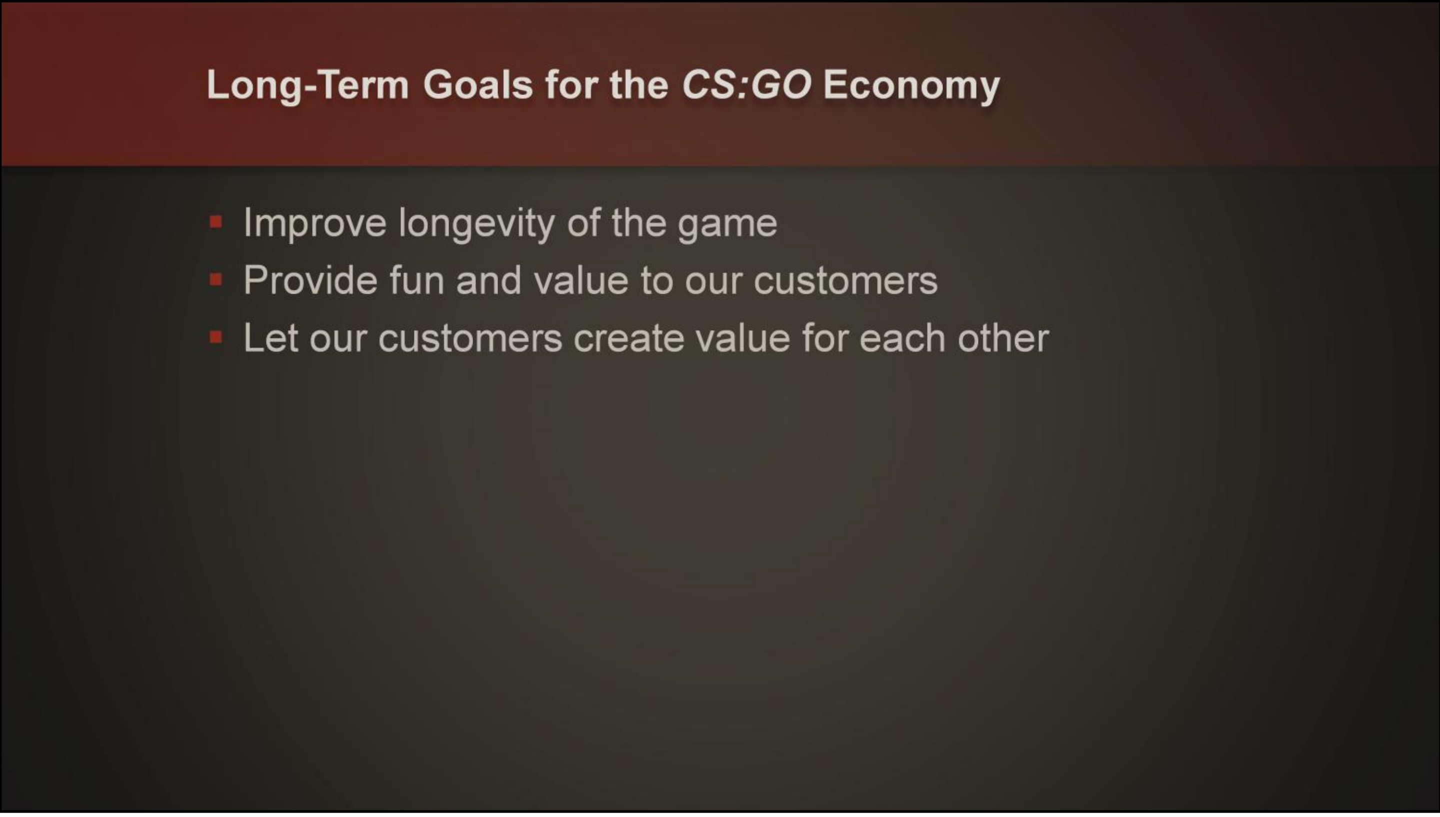


Goals for Economy Items

- Easy to make a lot of them
- Can be created by the community
- Lots of variety
- **Broad range of value**
- **Can be valued for more than one attribute**

The finer we slice our axes, the more nuanced trading and marketplace transactions can get. By creating our content in a semi-procedural way, we were able to slice our axes very fine indeed, creating a broad range of value, and creating value on multiple axes.

With players exchanging over 30 million items with each other on the marketplace, it looks like we’ve met our goals.

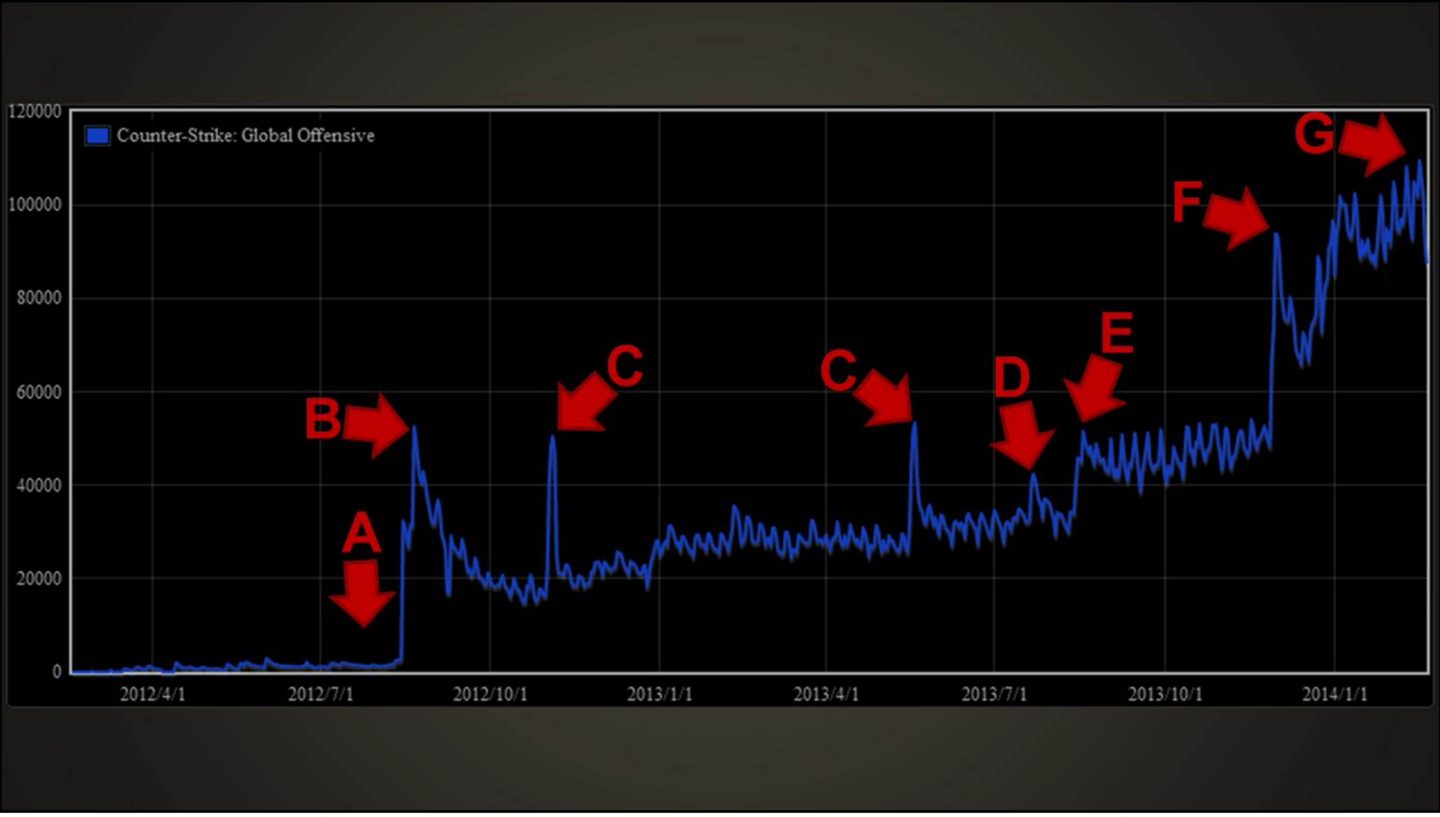


Long-Term Goals for the *CS:GO* Economy

- Improve longevity of the game
- Provide fun and value to our customers
- Let our customers create value for each other

Let’s wrap up by checking back with our product-level goals. We hoped the economy would improve the longevity of the game, provide fun and value, and allow our customers to create more value for each other.

Remember we only have to check on a few broad metrics to see if we’re getting it right: player numbers, player retention, and individual play time.



This is taken from steamgraph.net, which is a community website that monitors the Steam player numbers that we publish. It shows peak concurrent player counts.

Here's some notable landmarks:

The game starts in beta (A).

This is launch (B).

Here are some free weekends (C).

This is the steam summer sale (D).

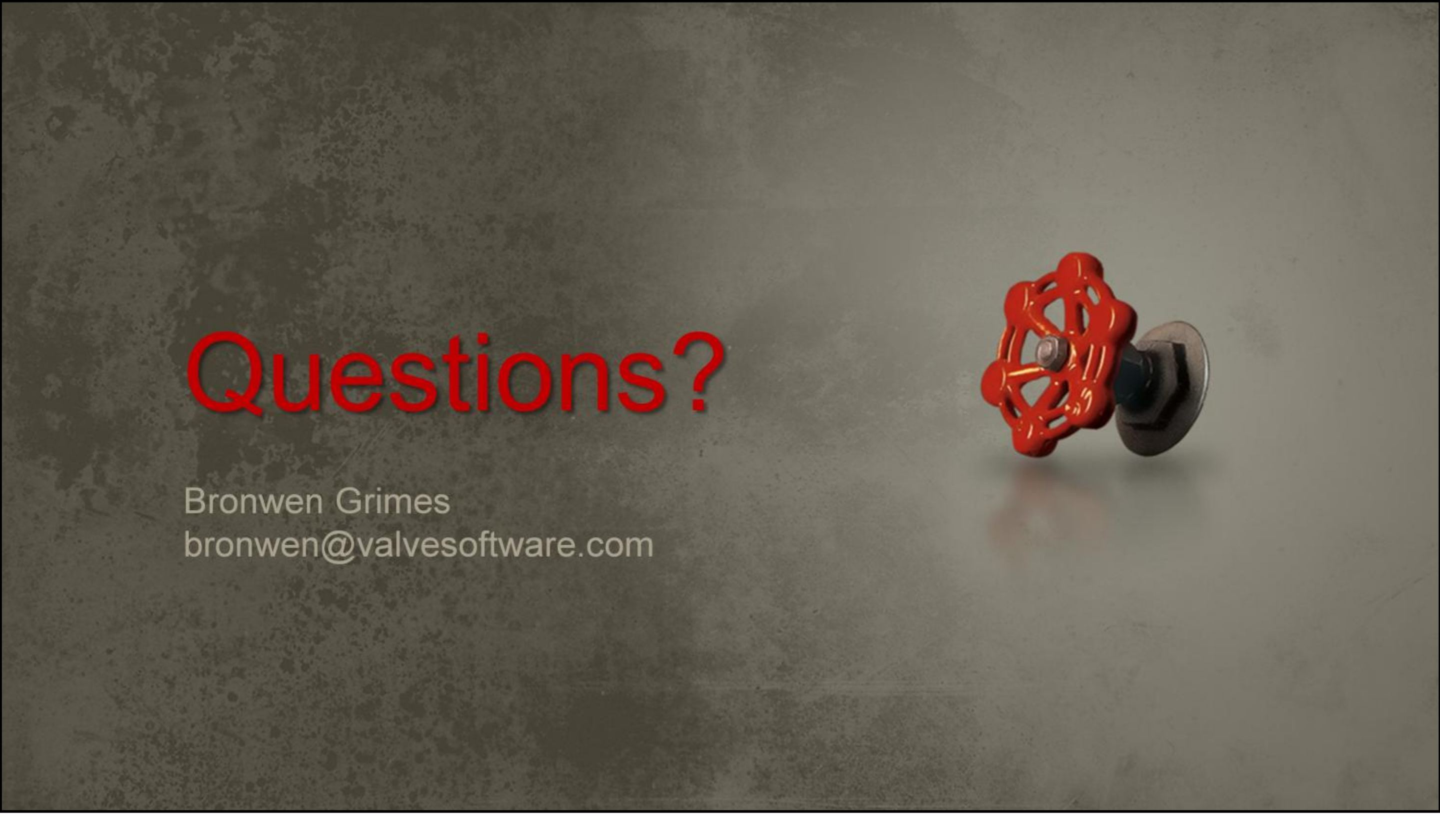
And here is the economy launch (E). This is a pretty dramatic increase in peak player numbers, bringing us to regular highs we'd only achieved previously by making the game temporarily free. So player numbers are up, which is what we were looking for.

The last big spike is Dreamhack (F), a big tournament event for *CS:GO* with a high viewership. The game was also on sale for this time period. This caused a huge player spike, nearly doubling our player count. There was a large influx of new players around this time. Lots of them stuck around. In fact, they stuck around at a much higher percentage than players from previous sales. So player retention is up as well.

Since that event we've been able to reach a new plateau of about 100,000 peak concurrent players and 2.3 million monthly unique players (as of mid-March, 2014).

In the aggregate it looks like we've met our goals for adding the economy. And individual players are sending us the same signal: they're playing more frequently and for longer, which is our best indication that we're providing them with lots of fun. As they continue to play together and exchange items with each other, they're creating loads value for each other.

Our one remaining goal is that we increase the longevity of the game. Only time will tell, but with such a large community of returning players, *CS:GO*'s future looks really bright.



Questions?

Bronwen Grimes
bronwen@valvesoftware.com

Links

www.steamdevdays.com

blog.counter-strike.net/workshop/

steamcommunity.com/market

migjimenez.blogspot.com