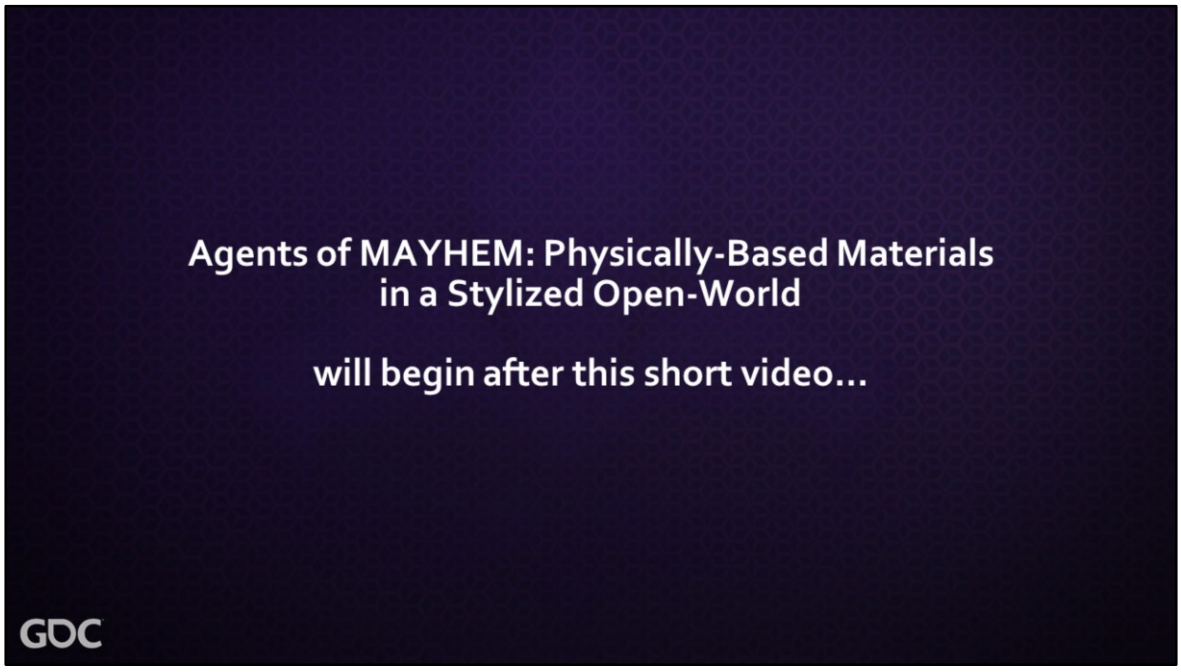

GDC
AGENTS OF MAYHEM
Physically-Based Materials in a Stylized Open-World
James A. Taylor
Principal Artist, DS Volition
GAME DEVELOPERS CONFERENCE® | FEB 27–MAR 3, 2017 | EXPO: MAR 1–3, 2017 #GDC17
UBM



The talk will begin after this short video – Don't worry about getting all the detail, just the rough idea is required for the rest of the talk.

Video Link: https://youtu.be/oNugtCnqDUk

I’m James Taylor, Principal artist and World Materials Lead on Agents of Mayhem

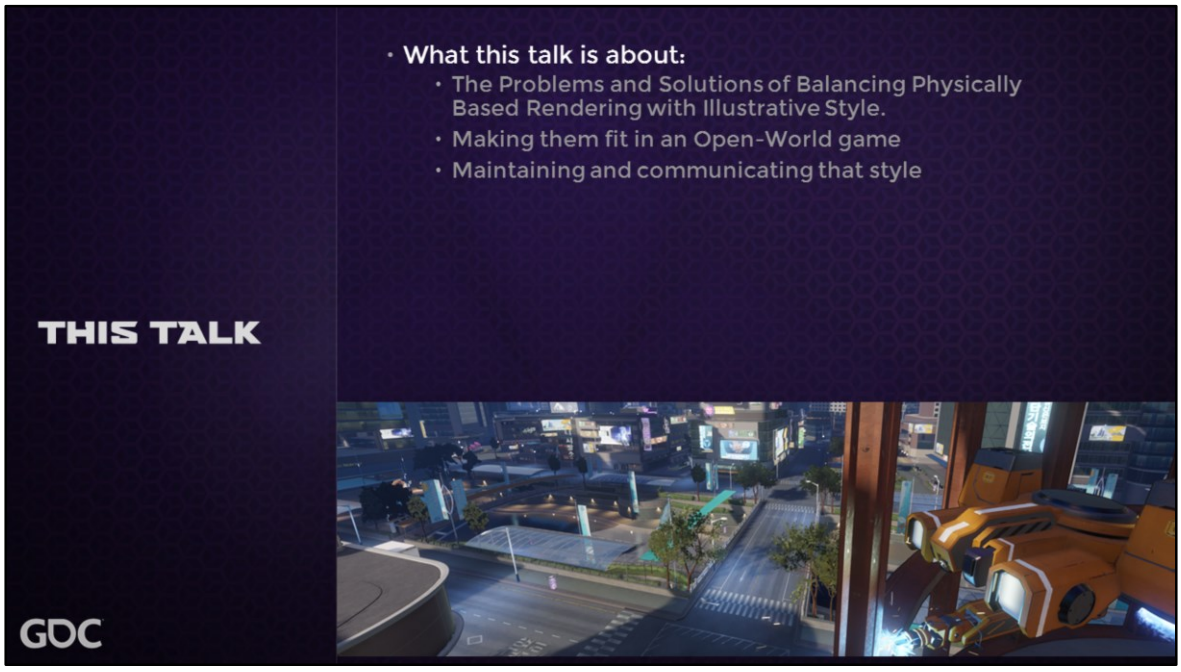
I’ve been at Volition for 13 years; with 16 years total industry experience… next month

Agents of Mayhem is a Third-Person action shooter set in a stylized open-world representation of a future Seoul.

MAYHEM is a privately owned and controlled network of agents, employed to fight against the evil forces of LEGION. MAYHEM aren't the good guys, they're just the better guys.

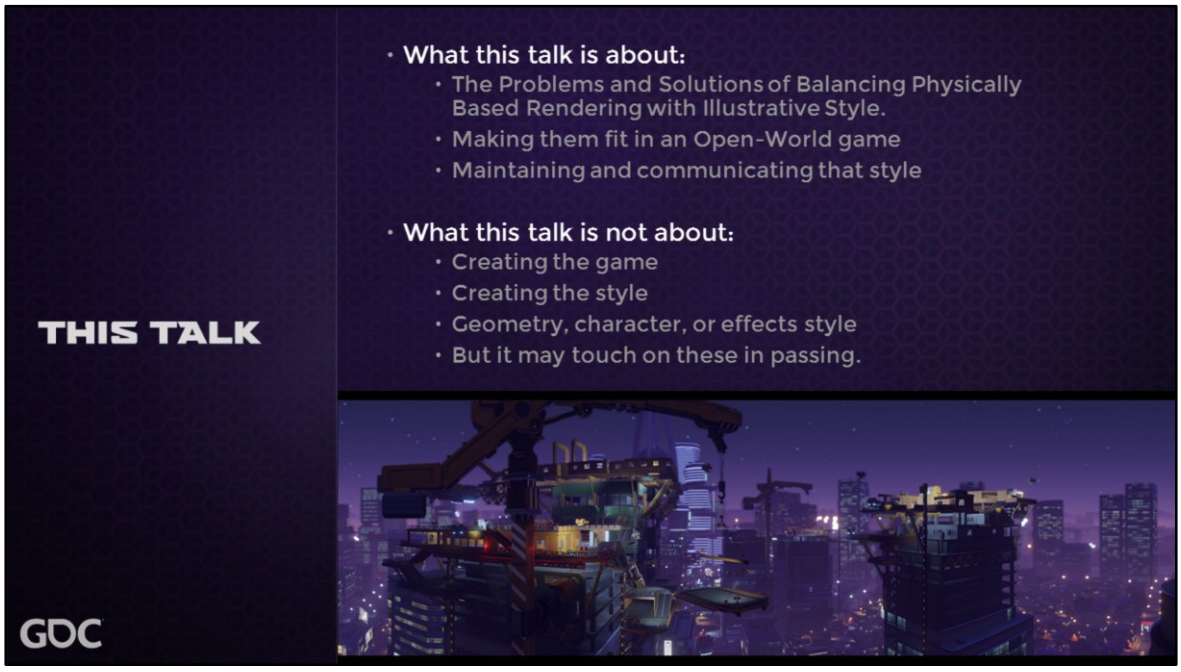
The player can control any agent, from a choice of three at any one time, with the three slots available for changing at any time. Switching Agents is as simple as switching weapons.

The game takes place against the backdrop of Seoul's Tomorrow Technology Expo, which has attracted the attention of LEGION's Ministry of Pride; determined to stem or delay humanity's technological progress at any cost.



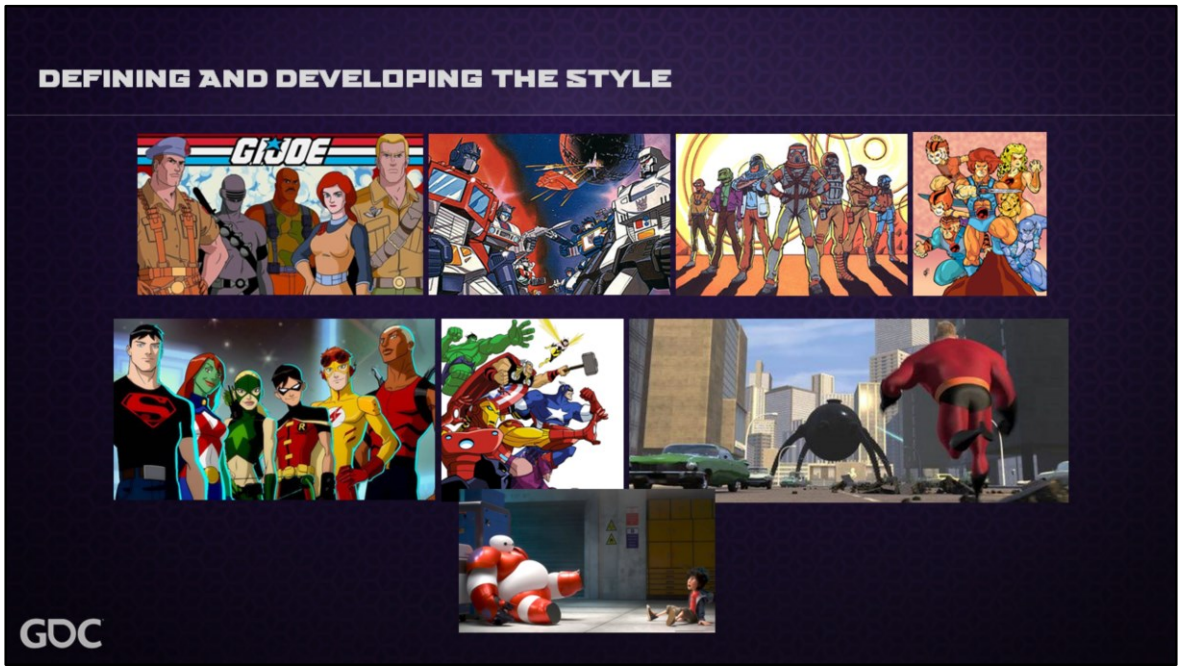
The talk is about Physically Based Rendering and illustrative material style, but more specifically, how we maintained that material style vision across the two seemingly opposing technical looks, as well as across various artists and vendors, within the limitations of an Open-World game.

The main talk will take on a Problem and Solution format, with each of the major problems faced presented with it's solution.



The talk is not about how we made the game, nor how we created the style. The talk is also not about geometry, or characters, and the technical challenges there, though many of the points, if not the exact solutions, apply to them too.

So while the talk may touch upon those things, it is not about them.



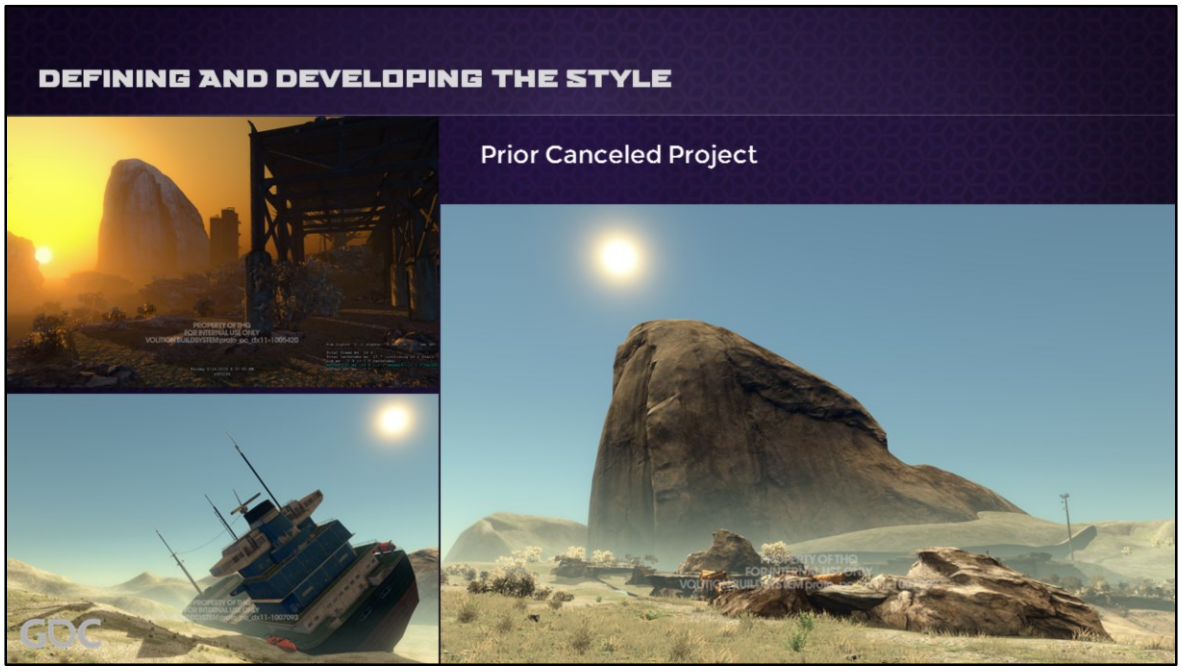
And now I'll speak briefly on one of the topics I said I wouldn't cover.

AOM's style is what we call "80's Cartoon Nostalgia"

Taking the theme of the game: Cool diverse characters Vs Megalomaniacal Villains, we extrapolated that to the themes of cartoons we grew up with, from the 80's.

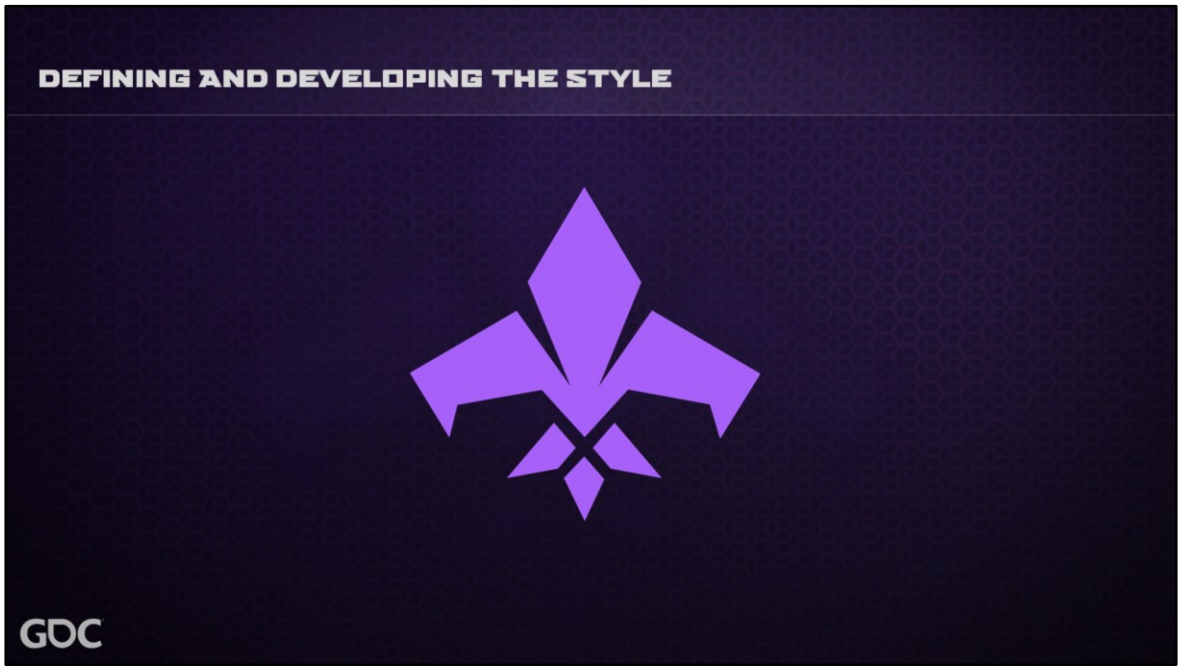
Transformers, GI Joe, Thundercats, MASK, Centurions

We wanted an updated look though, finding influence in more modern fair.
Young Justice, Earth's Mightiest Heroes, and pre-rendered movies such as The Incredibles, and Big Hero Six.



Developing this style took a long time, and it was initially felt we would go in a more realistic direction, like the previous, cancelled title AOM was replacing.

That project was a gritty post apocalyptic title, and it was from there we inherited the Physically Based Engine that drives AOM (this is the only environment really done for that title, with a very early version of the engine).



As mentioned, this talk is not about developing the style of AOM, but since there may be some curiosity, here is a short sequence of the various stages developing the engine and style went through abandoned over the course of the style's development.

Early engine Tests & Early Seoul Test

Concepts

Final style presentation test

That final look came together after developing the Core Pillars, created at the same time as the 80s Cartoon Nostalgia idea.

These guided our look for the rest of development. Briefly going through them:

Simplified:
Exactly what it says on the tin – the idea that things external to gameplay should not need explanation beyond what the player sees, except where intentionally vague (such as some LEGION tech), nor draw attention away with too much visual clutter. Simplification reduces confusion and the need for explanation, allowing the player to focus on the game itself.

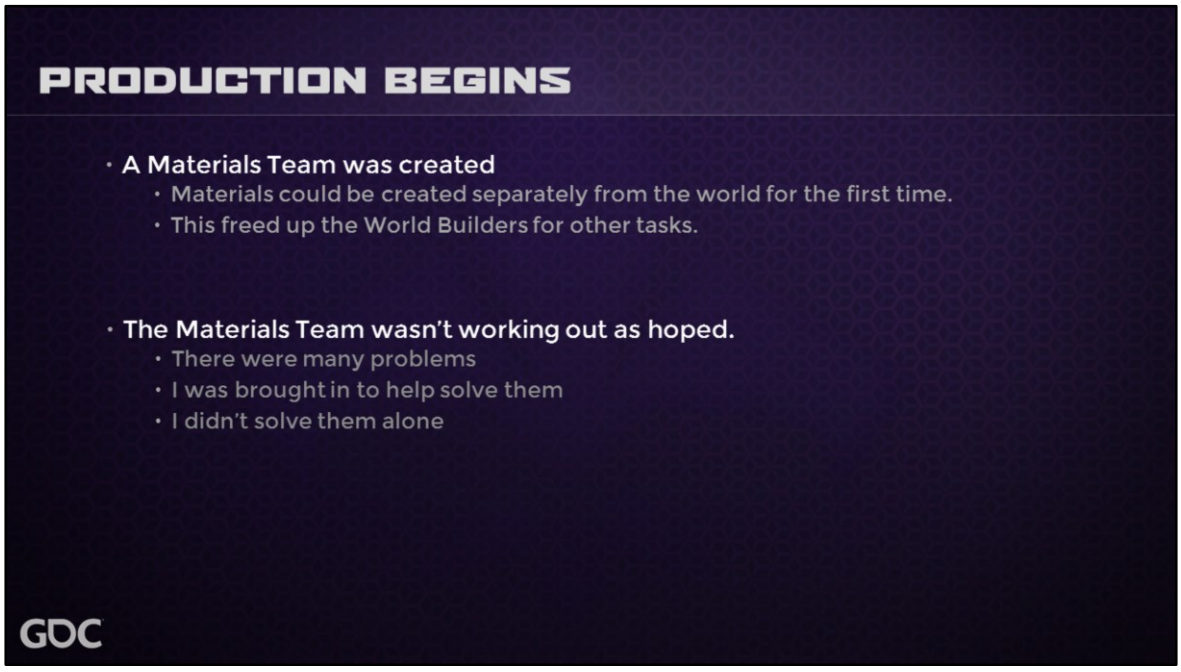
Theatrical:
Over the top, clear from a distance and dramatic. LEGION doesn't jog in on foot, they arrive in a ball of light or an explosion.

(Continued)

Vibrant:
An extension of the previous two, the game should not be dull and gritty, but colorful and pretty.

Balance and Harmony
The most important tenet, especially for the Environments. Everything should work together to enhance the gameplay, but not at the expense of the visuals.

If the players, enemies and action are the lead guitar and vocals, the Environments are the backing track; the Drums and bass guitar. Not the focus, but you'd notice if they weren't there, or were sub standard. We'll be referring to Harmony a lot in this talk.



After the final style was decided upon, A Materials Team was created, a first at Volition.

The materials could be created separately from the world builders, freeing them up to spend more time on layout, major world geometry, and design integration.

But after three months it was felt the Materials Team was not working out, and I was asked to join the materials team and take a look into what the problem was, with an eye on solutions.

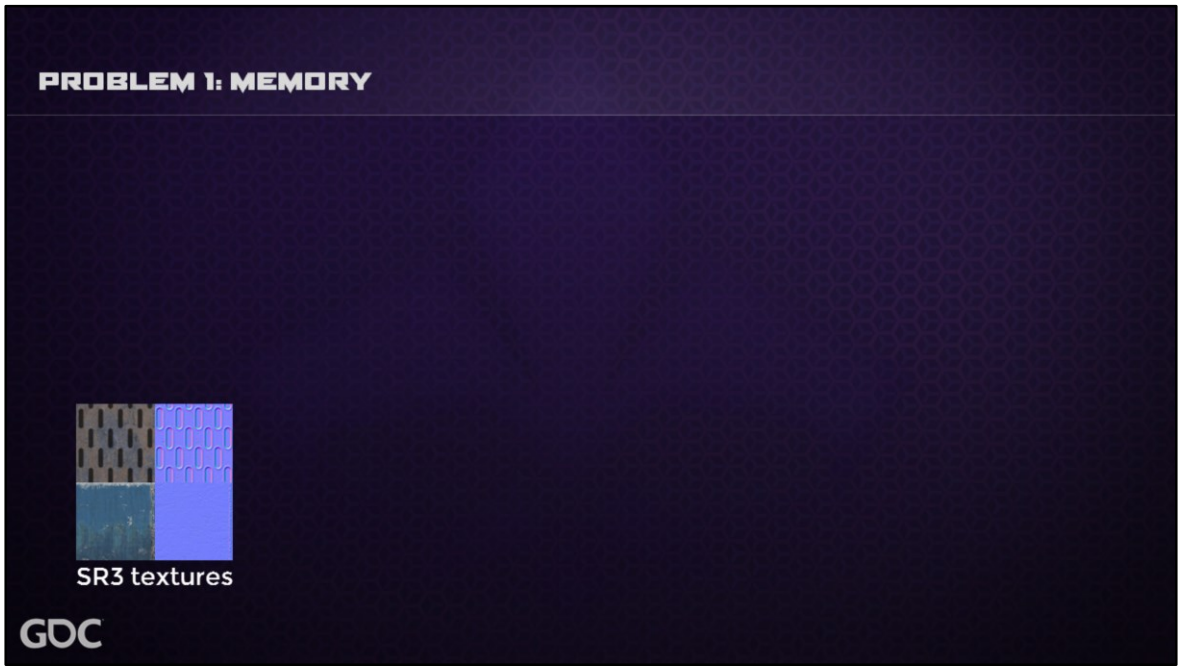
This had a lot to do with there being no precedent for making Materials in a vacuum at Volition, or Materials this stylized, or materials using PBR.

There were many problems

I didn't solve them all alone (but I am the one giving the talk)

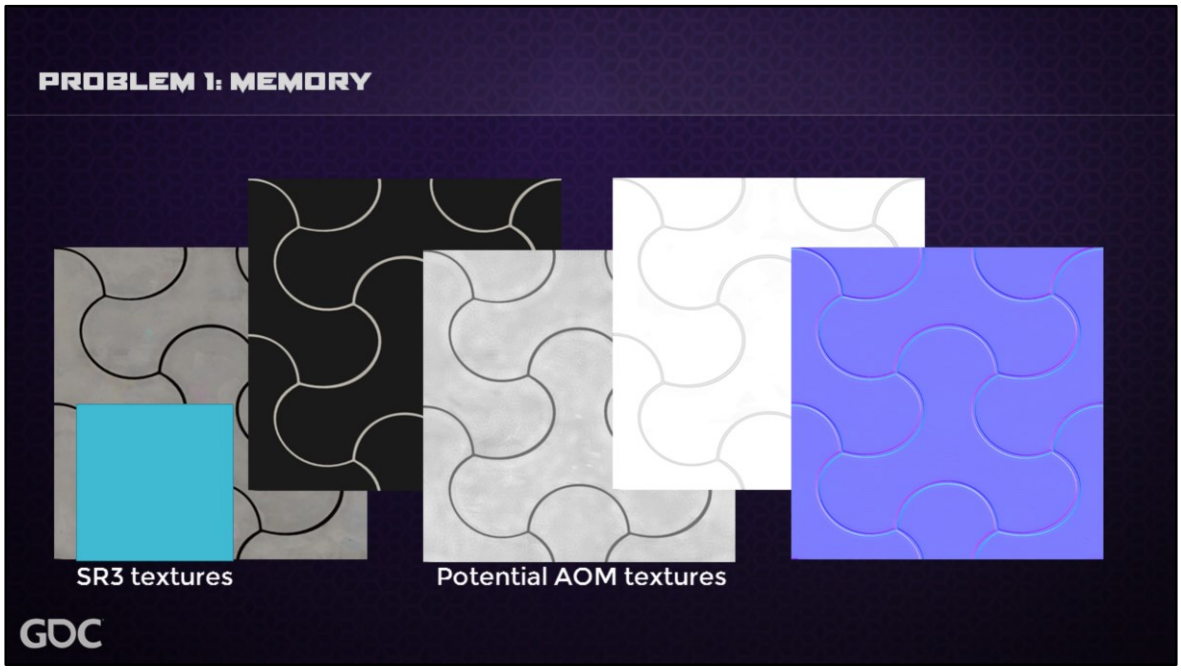
Initially we thought we'd have a lot of memory to throw at materials, and relative to what we'd had on previous console cycles we do. But it wasn't long before we realized the new engine brought with it new memory problems

Saints Row 3 (And by extension, 4) had few environmental specular maps, so most materials were a (usually) tintable Diffuse, and a Normal, both of them pretty small at 256x256.



Here are two example SR3 environment materials, forming the rough equivalent of a 512x512 texture. We'll stick with using 2 as our example, so it makes a nice square equivalent.

We wanted AOM to have much higher resolution textures, and take advantage of Physically Based Rendering.

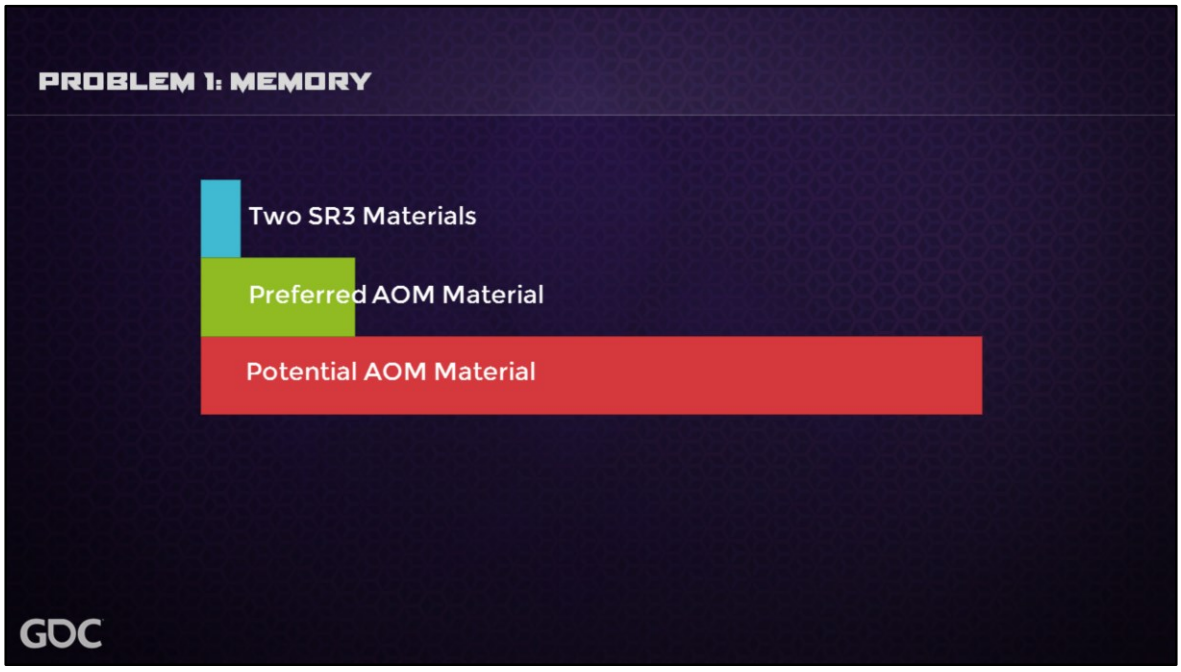


256 pixels per meter was chosen for texel resolution (before my time on materials), which resulted in most materials requiring 1024x1024 maps

With that in mind each non-alpha material generally needed at least 5 textures, all at the higher resolution than SR.

The chosen style also made tinting a problem, with subtle colour shifts and accents bringing the textures to life.

Even with the available memory on the new generation of consoles, this didn't add up.



Here's the comparison of an average SR3 material size in comparison to a theoretical AOM one.

Even if we take into account new compression methods and the new memory working in different ways to the old, this difference is limiting. To simplify things, we'll ignore the compression for now, and just focus on the source texture sizes.

We clearly need to get these closer together. For parity with the SR materials, we need to get the red bar down to the size of the green one.

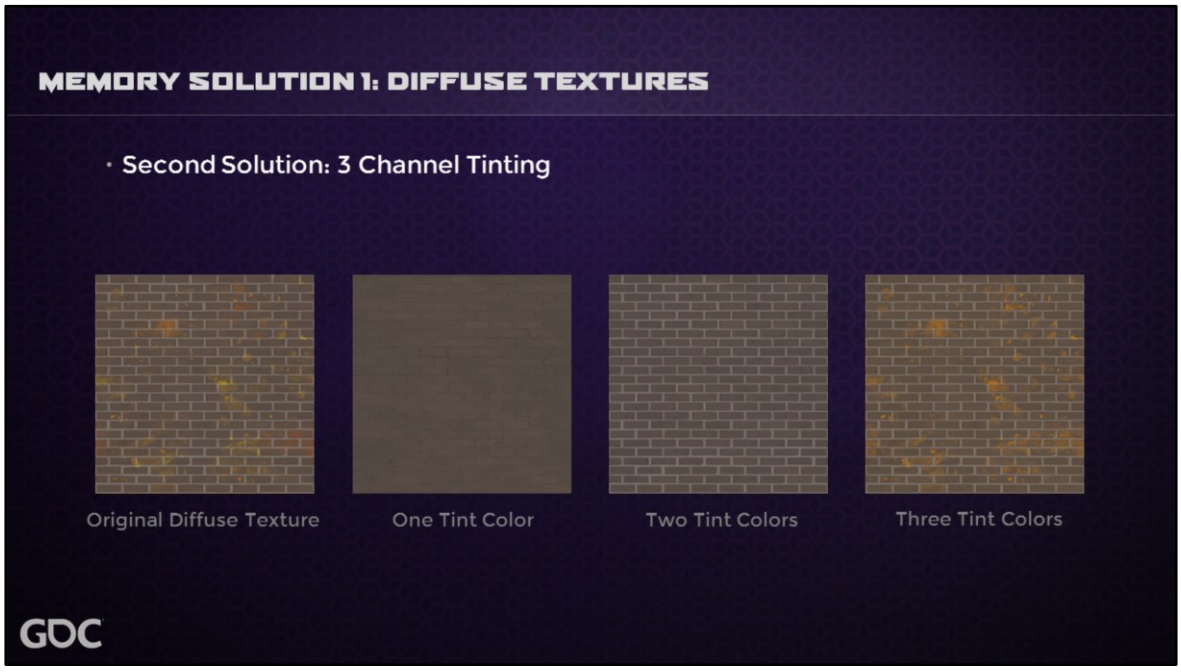
Our first solution, was to use smaller textures, custom painted for each colour (in this case by Jon Criner, who initially came up with the diffuse style we use).

The diffuse for these looked gorgeous, but while it may have worked from a memory standpoint (it probably wouldn't be, as ultimately we would want a lot of colours, and therefore require a lot of individual diffuse textures), it resulted in heavy and obvious tiling, which we wanted to avoid as being too visually noisy and ugly, removing the benefit of the pretty textures.

It would also be a time sink, which we couldn't afford, and be difficult to maintain.

Here's a still from that Style Test, showing the evident tiling, but also an area who's tiling is far less evident.

Part of our solution was to remove obvious 'hotspots' from the maps, such as the clearly darker bricks in a distinct pattern, but to overdo this was to make the materials bland, so we needed another solution, that maintained the subtle painterly look of Jon's initial textures, without compromising the size.

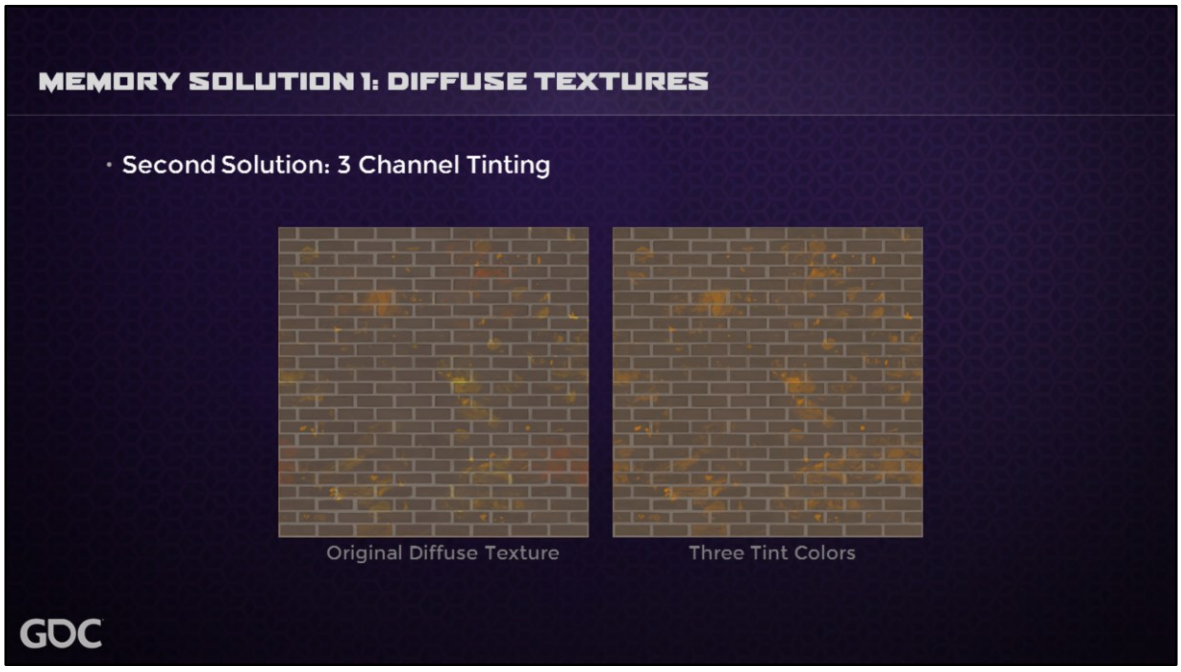


The next solution was to tint them anyway, and hope that tonal differences would be enough.

They were not, but we did come up with a tinting method allowing 3 tint colours masked by an Alpha (Using a mask of Black, 50% Grey and White representing each tint color), which was better than the two-tint mask we had had on previous titles.

The thought was that the tinting would be enough to offset the colour shifts when the three tints were not needed to differentiate diffuse colours (such as on brick). The ‘subtle’ colour shifts are greatly exaggerated in this example.

Sadly, this didn’t work very well.

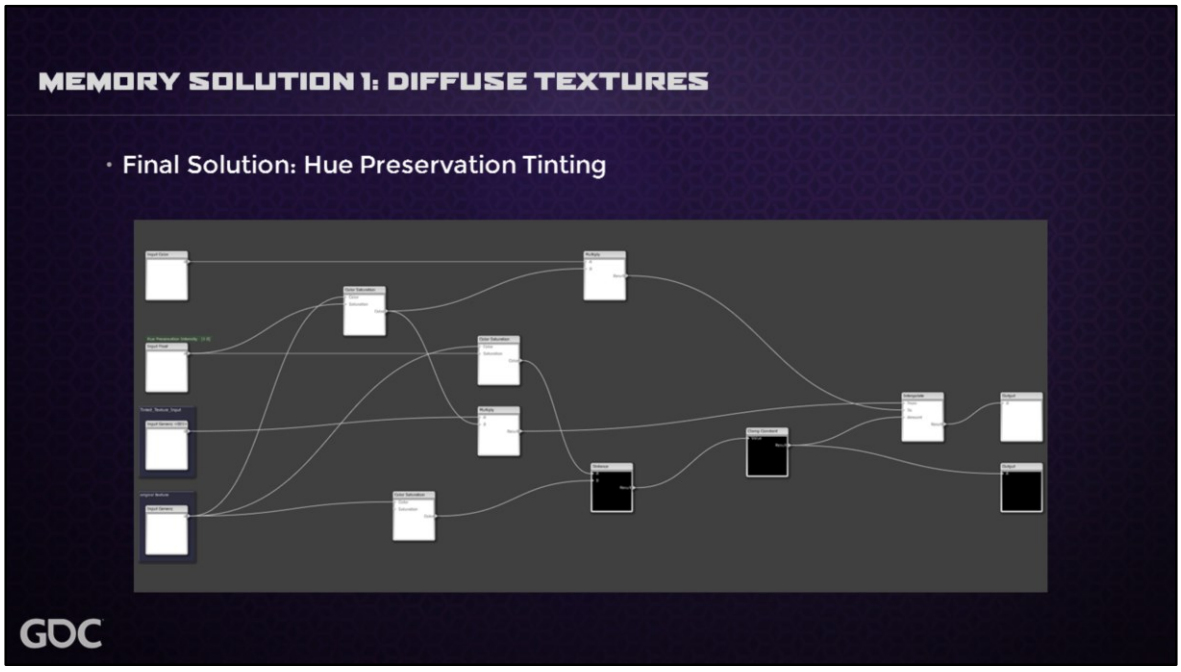


As you can see, in the original diffuse the colour shifts vary from red to yellow, while in the tinted version we've just had to split the difference and go with orange.

This also increased the required memory for every diffuse texture (for the mask); the opposite of what we were trying to achieve.

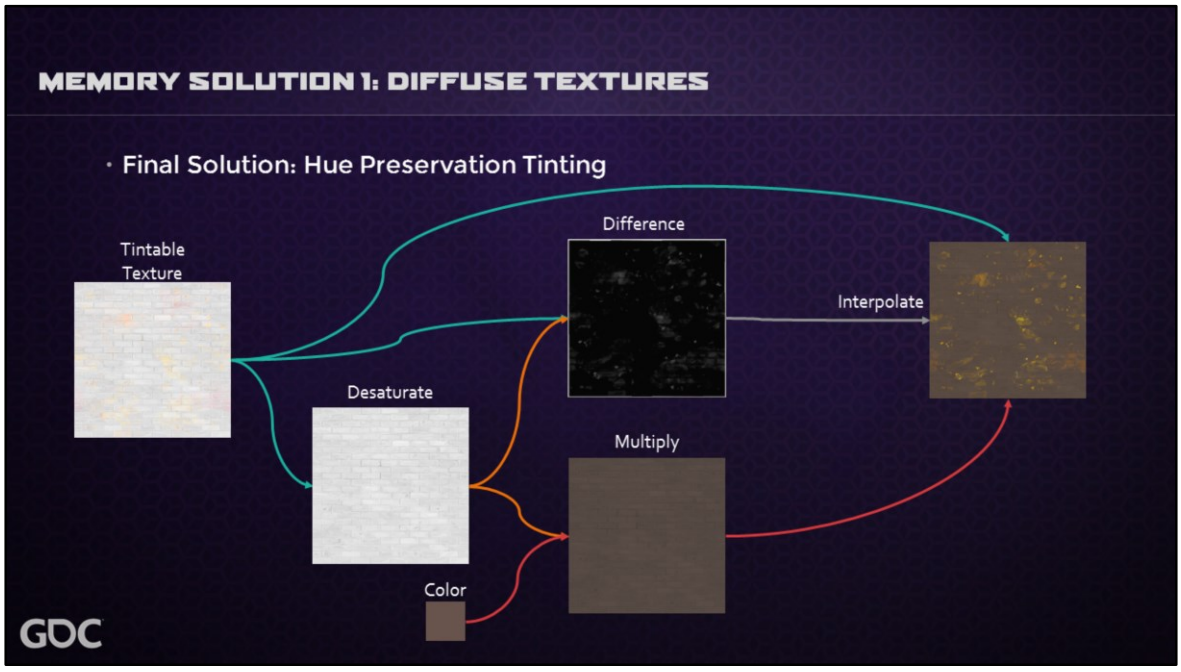
We did keep the three tinting colour option though, which increased the tint variation available for some materials when needed.

Finally our Matt Currey, our shader ninja, thought of a solution, allowing tinting, while preserving subtle colour shifts.



The method was simple in theory – Create the texture as a tintable greyscale, but containing the subtle colour shifts on top of this.

Here's the shader graph, which I'm sure you can all read just fine. No, don't worry if you can't, we'll simplify this a little and break it down.

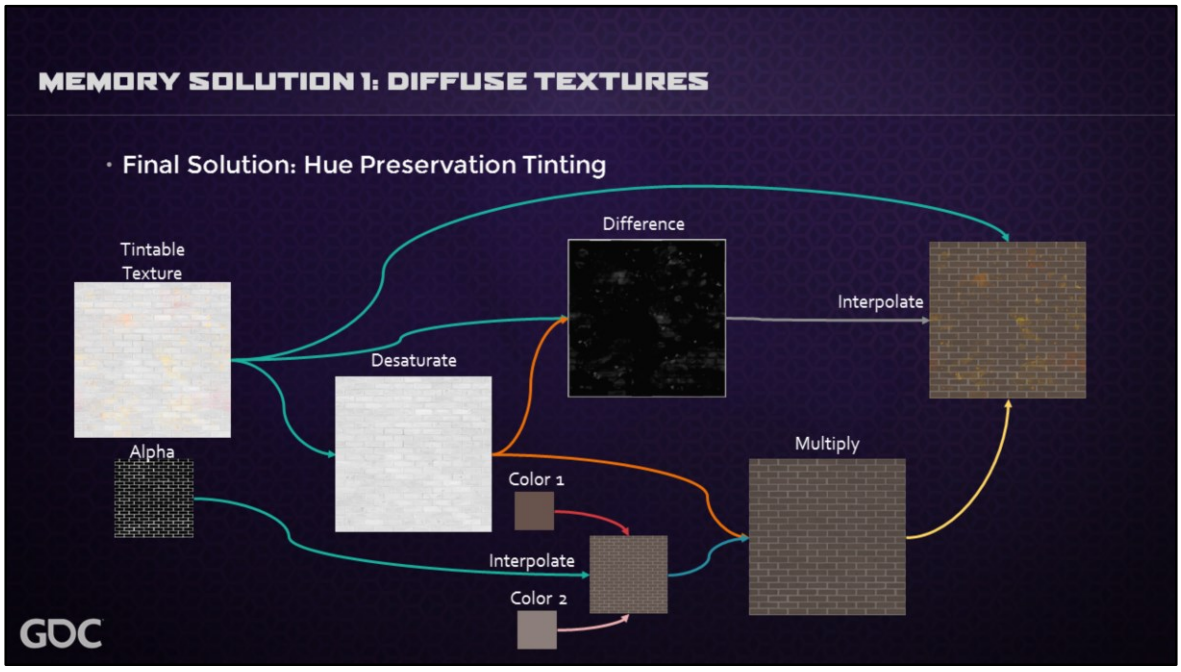


First the texture needs to be a greyscale, with some colour variance it. We'll deal with the problems of creating that soon enough.

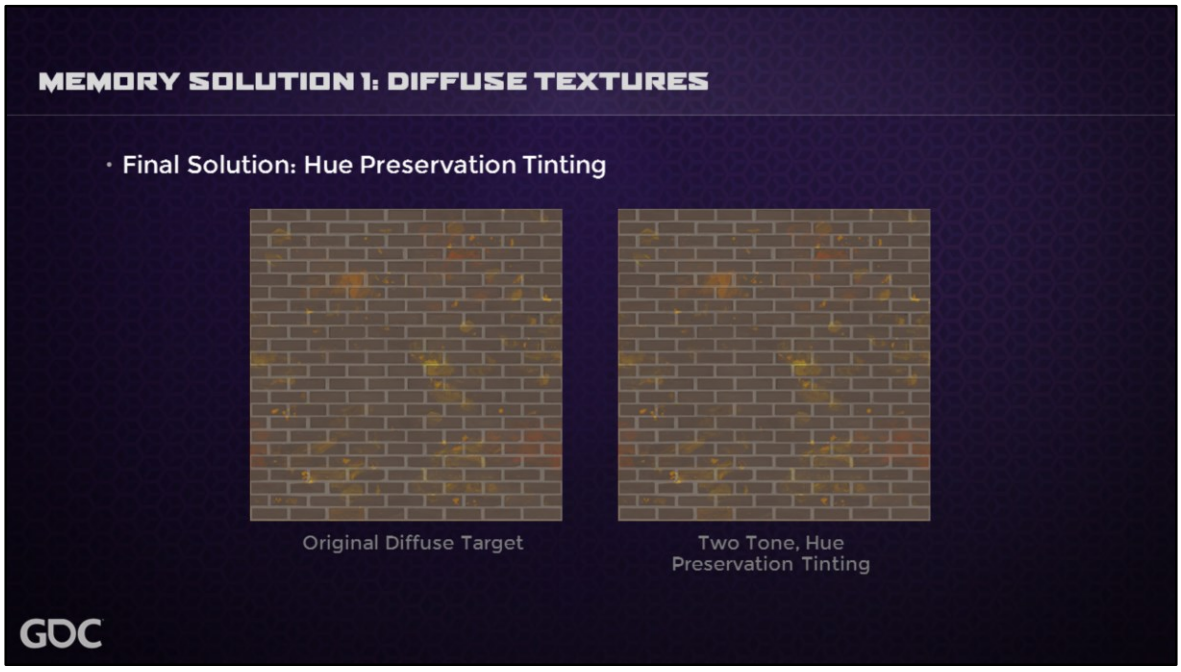
This is then desaturated, and multiplied by a colour to get our base tint.

The Desaturated version is then compared to the original texture, and differences found, creating a mask. (I've adjusted the levels here to make it more visible)
There's a little more to this to get the greyscale mask, as a true difference would return colours.

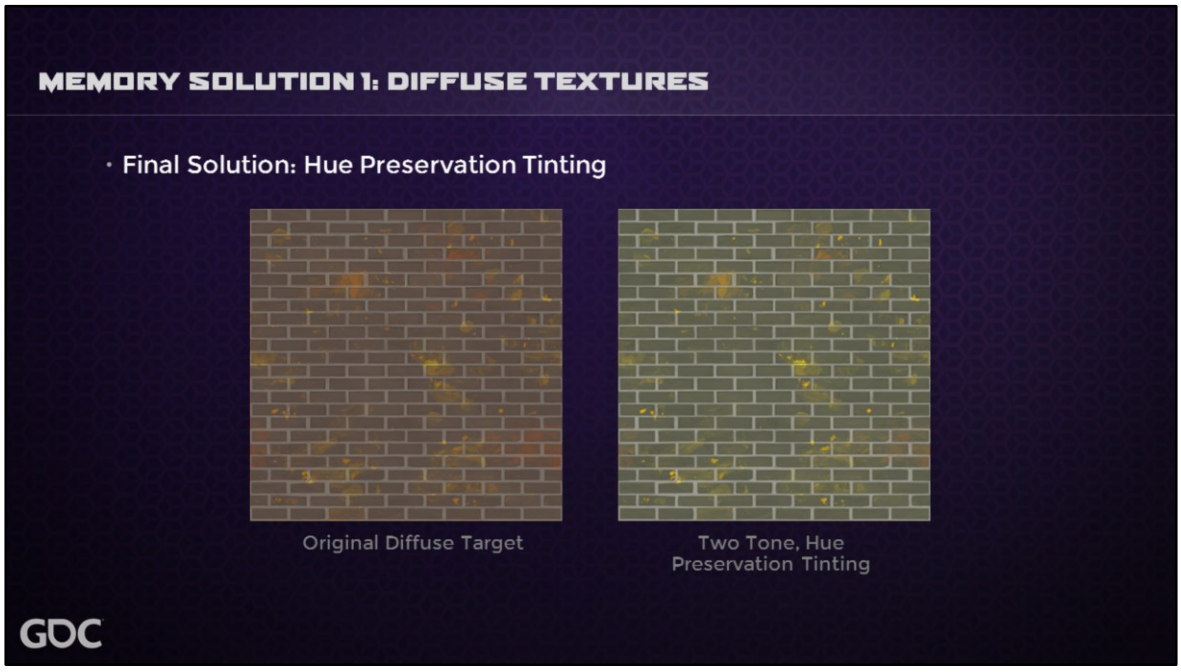
Finally, the Tinted version of the texture is blended with the Original texture, with the Difference mask allowing only the colour to come through (again, greatly exaggerated here).



This still left us with only one Tint, but we could use the preexisting Tint method, using the Alpha of the Diffuse, to allow an additional Tint or two (just one extra in this example).



The result is much closer, and often indistinguishable.



This shows a problem though. If a significantly different tint colour is used, the colour in the original texture may not harmonize as well.

To solve this problem, we also introduced a hue offset and power into the shader, which gave us control over the hue and the amount of colour retention from the original texture.

This is limited, in the amount the hue can be offset, but is enough for the majority or circumstances.

MEMORY SOLUTION 1: DIFFUSE TEXTURES

- Final Solution: Hue Preservation Tinting

Original Diffuse Target

Two Tone, Hue Preservation Tinting

GDC

Much better.

And it works for almost any Color – in this case shifting entirely to blue was too much, but getting close and desaturating does the job.

This also effectively gave us 4 tint colours to play with when needed (3 tints, plus an additional colour set that could be hue shifted). It was rarely used, but it was nice to know the option was there.

An example of the Hue Preservation tinting in use.

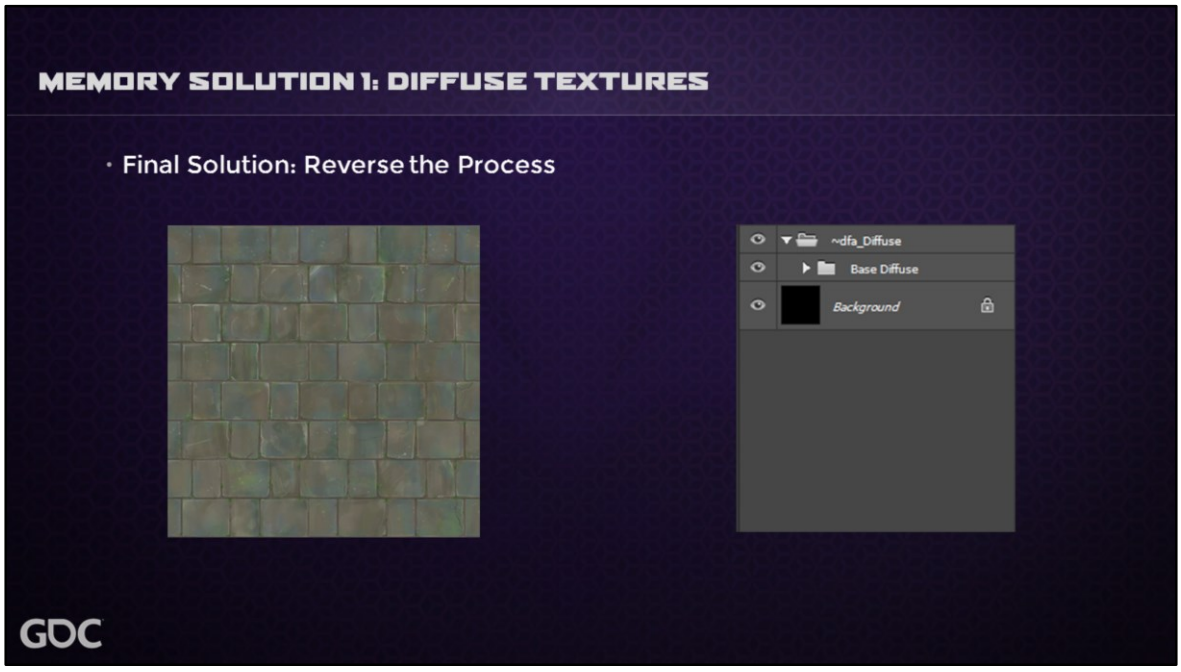
First is as it is intended in game, although we use different colour combinations elsewhere.

Second changes the stripe to match the ground

The third over-cranks the subtle color shifting (not usually recommended), and desaturates the warning stripe – all handled in the shader.

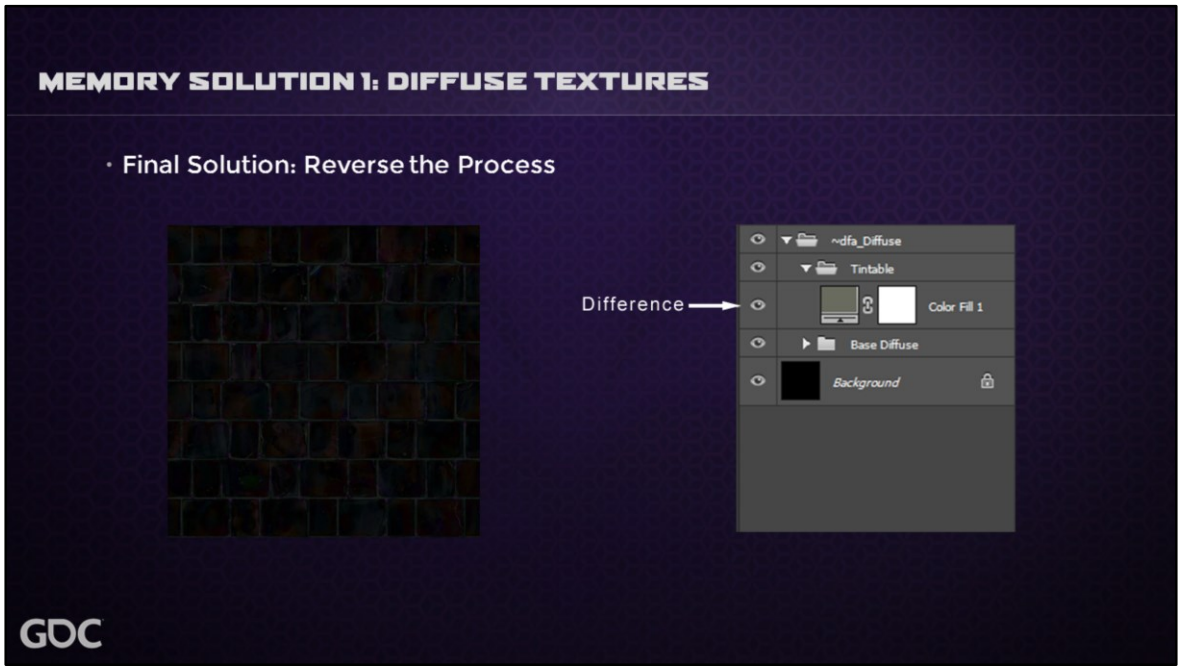
And you can go a little stronger with it for the secondary tint, and you can still see the subtle colour shifting.

This still left us with the issue of creating the textures in the first place. The two options were:
1 - Creating it as a greyscale, and adding colour on top – making it difficult to correctly harmonize the colour shifts
2 - Painting it in full colour, and then stripping out the base colour, and keeping the tonal shifts, while leaving the upper layers of colours untouched. This was a lengthy process, especially if we also wanted to maintain a version with the colour intact; and we did, in case we needed to make changes later.



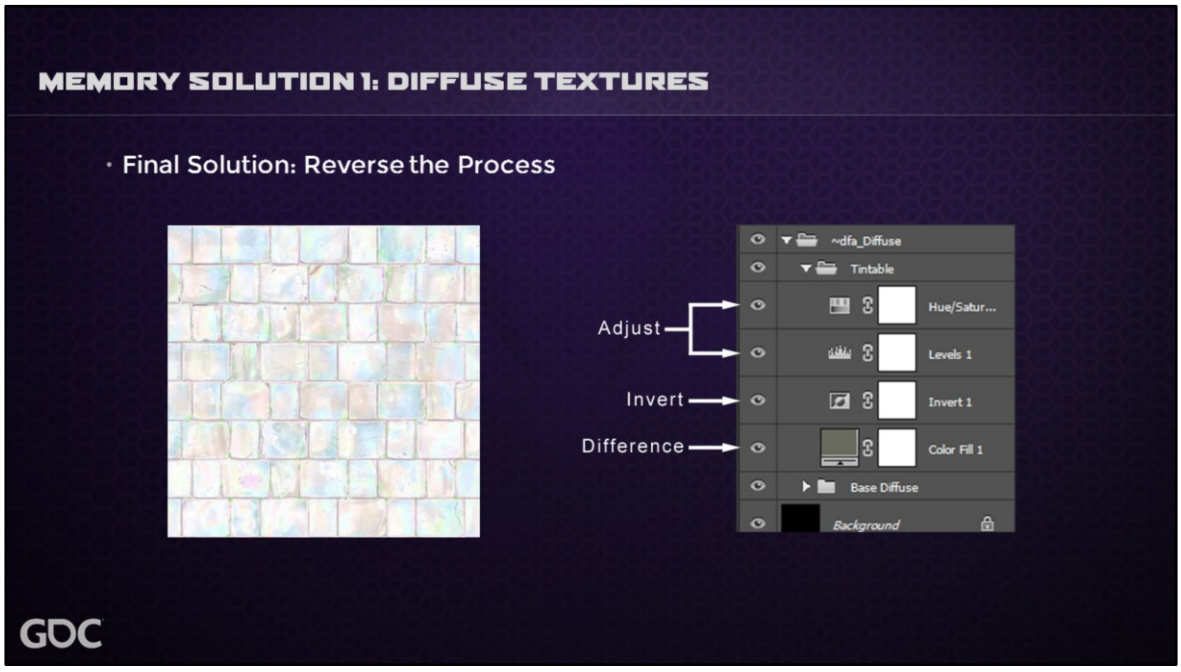
Finally, after an embarrassingly long time, the obvious solution presented itself. The process used to tint the textures in the shader could be used in reverse in Photoshop (which we use for painting the textures).

Here, all the layers comprising the painted texture on the left are stored in the Base Diffuse Layer Set. We don't need to touch them in any way.



We identify the 'base' colour of the texture (usually this is a solid colour layer at the 'bottom' of the texture anyway), and then add this as a flat Colour layer above the multi-layer painting set.

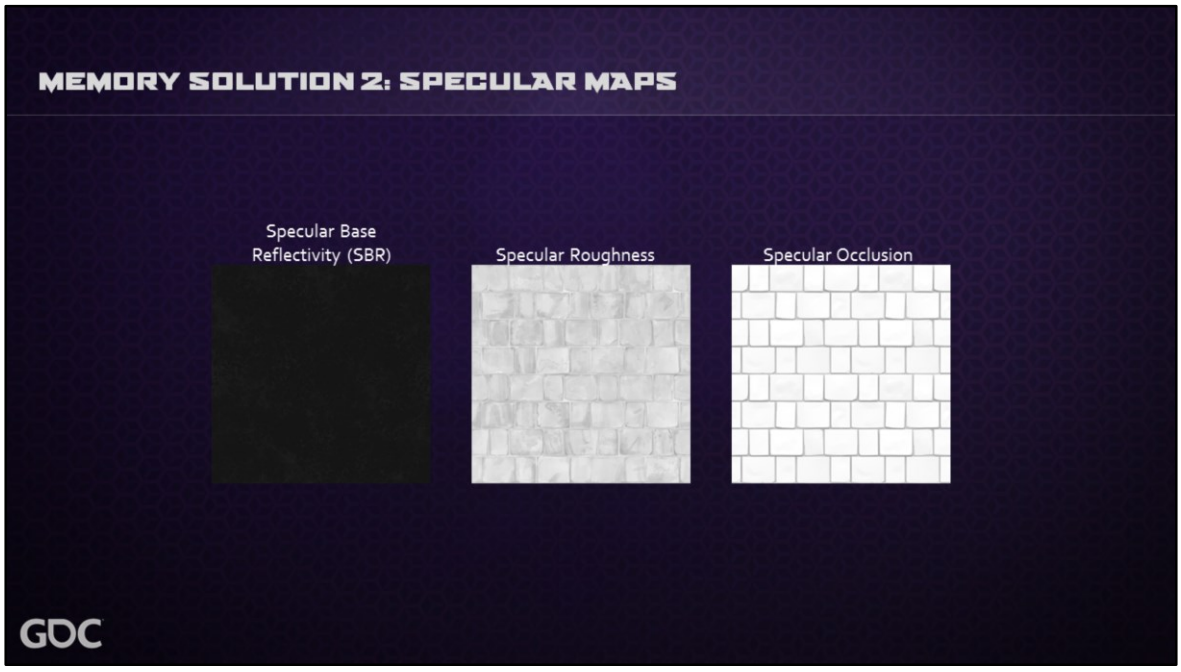
We then set the layer's blending mode to Difference. The result is an inverted image showing only areas other than the Base colour in colour. An Invert adjustment layer placed above this results in a positive greyscale image with subtle colour shifts.



Sometimes additional adjustment layers are needed for tonal balancing or compensating for compression on very subtle colours, but the result can be re-balanced in the Material.

The result is that the textures are much easier to initially author, far faster to make useable for tinting, and much easier to iterate, as these adjustment layers simply need to be turned off, and then restored after iteration is done.

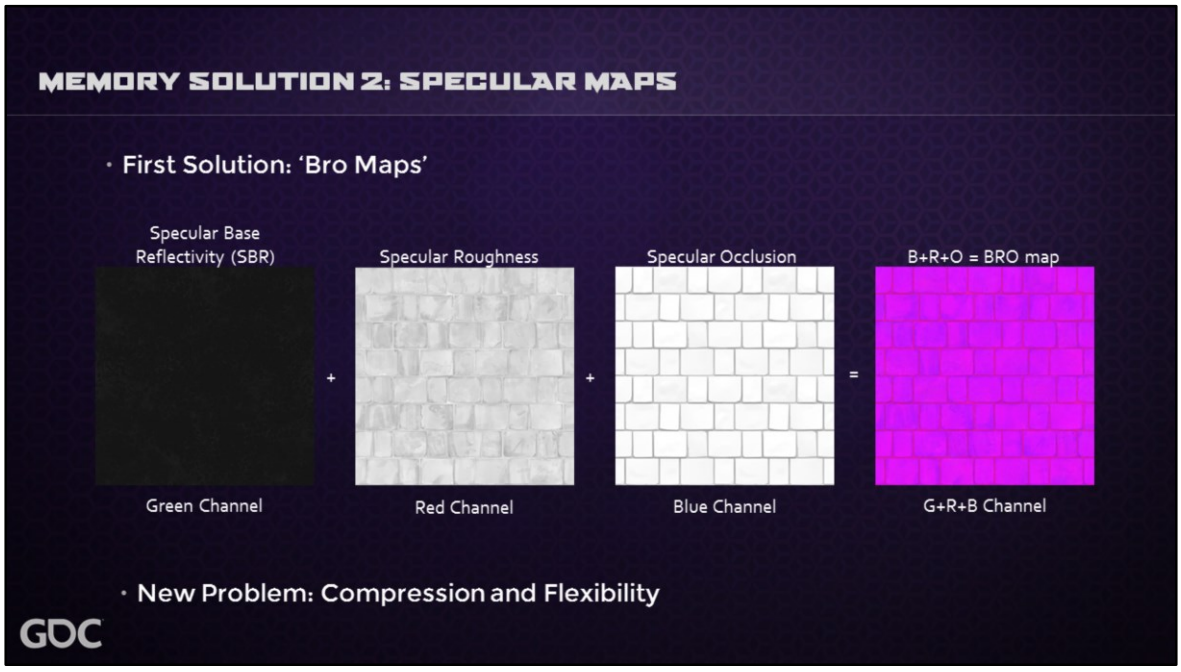
Some more complex materials require a little more massaging, especially those with three Tint colours, so this is a fairly basic example, but the majority of the world materials in Agents of Mayhem are authored in this manner.



The problem of the specular maps is that we potentially need three of them; The Roughness Map, the Specular Base Reflectivity map and the Specular Occlusion map; when on previous titles we often had none at all.

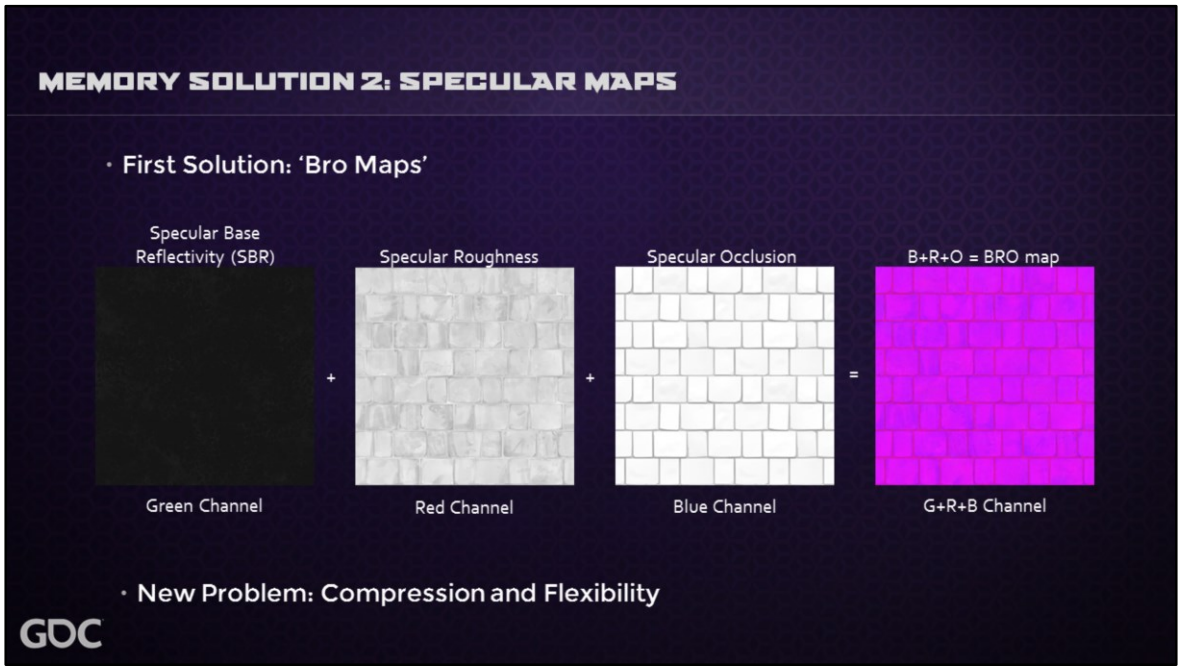
Not Metalness:
Unlike a metalness based engine, where metals are painted in the diffuse, and then isolated for special treatment with a mask, our engine uses a Specular Base Reflectivity map that allows for more subtlety in surface response. This map can use one or three channels depending on whether the material contains metal or not, as metals are coloured, and almost everything else is not.

It is possible to build the shaders to work effectively the same as a metalness workflow, but we didn't want to give up the potential level of control our engine allows.



Our first method of reducing the memory footprint for these three maps was to combine them into one. We called this a BRO map, for Base Reflectivity, Roughness and Occlusion. I've heard similar things called Greypacks.

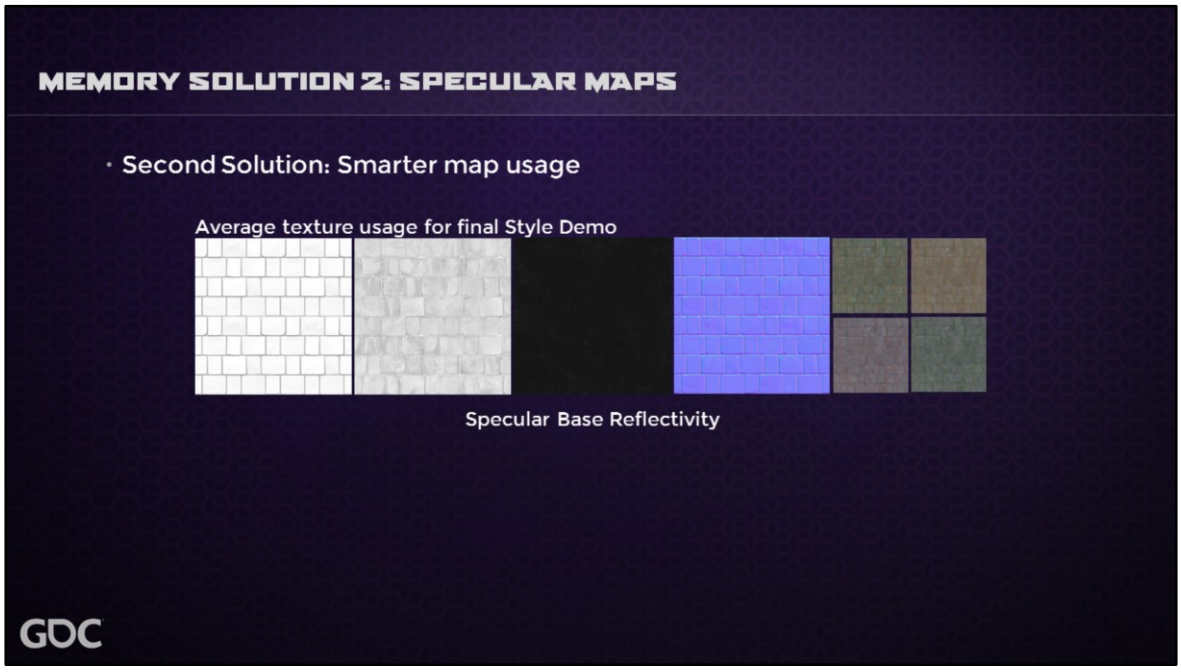
Each channel of the texture contained information for one of the Specular maps. We were going to use a threshold and a tint colour to tint metals different colours – anything too high a value not to be metal would be tinted – not dissimilarly to the way we masked colours from tinting in the Diffuse.



(Continued)

Unfortunately this was more difficult to author than we would have liked (though we had two available methods to simplify it), was still limiting if we wanted very dull metals, and compression often tore it apart since the maps needed to be Linear, and Linear SBR values are very, very low – just the range compression likes to impact the most. It also limited us to only ever using these three maps together, when in practice we knew that one roughness, or one SBR map, might be useable across multiple materials.

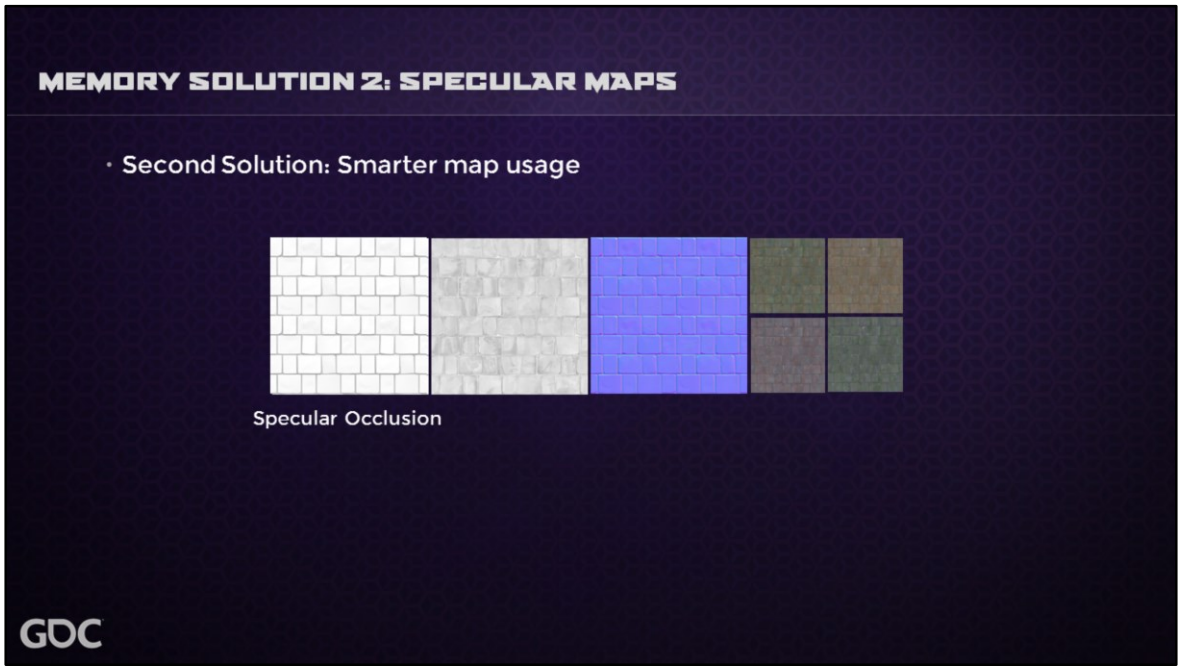
We still use 'FRO' maps for Characters, with the SBR map replaced by an another common character mask (the SBR map being it's own non-linear one), and a similar 'Triple Mask' for when we need multiple masks that will always work in concert.



Fortunately, a simpler solution presented itself, though it wasn't without it's own problems, such as people understanding it despite it's theoretical simplicity.

Much of the time, a material such as brick, or plastic or paint, will have a very consistent Specular Base Reflectivity. This meant that much of the time we didn't need this map at all, and could make do with only a value. We'll get into this in more detail in the next section.

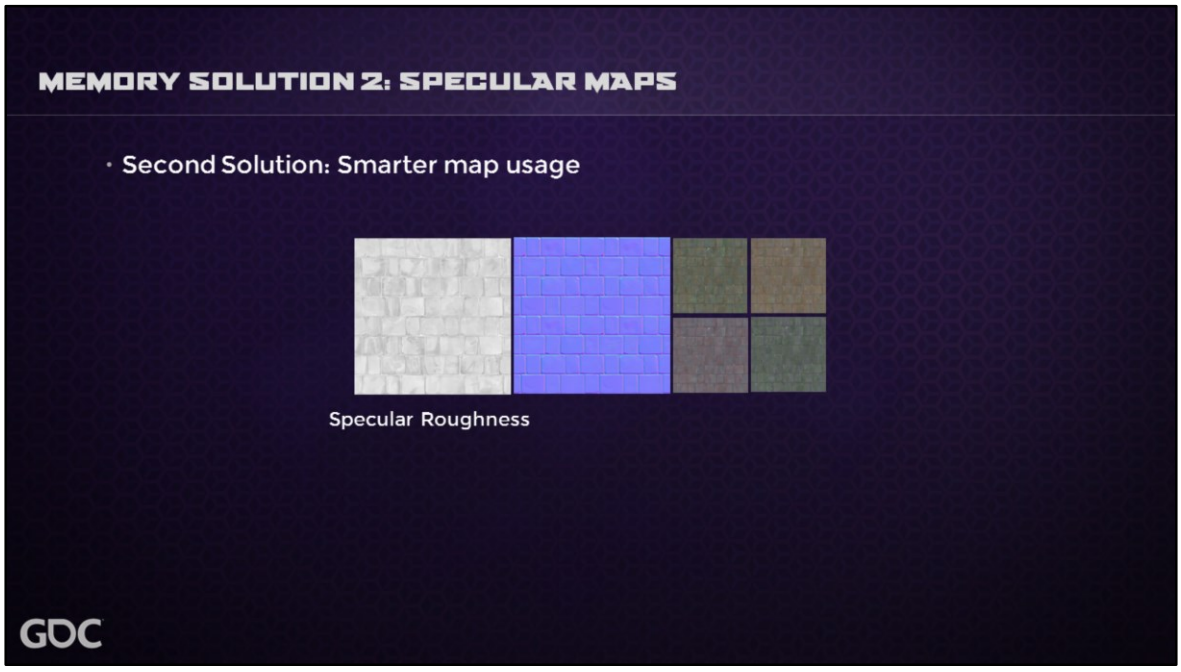
So we introduced the option of having a value instead of a texture. This gave maximum flexibility, and the memory saved over the many materials now requiring no SBR map can be used on those few that do require one with more variance or coloured values. In many of those cases, the texture could also be smaller, with the rougher material transitions from the smaller texture sizes being hid within the Normals at the point of transition.



We also found that the effect of Occlusion masks, at the fidelity we were dealing with in the Open World, was almost negligible in most cases.

So the easiest solution was to ditch them entirely. We didn't though, as there are enough edge cases when you do want to occlude something in the specular, that we still kept them as an option, but they are used much less often than we initially thought they would be.

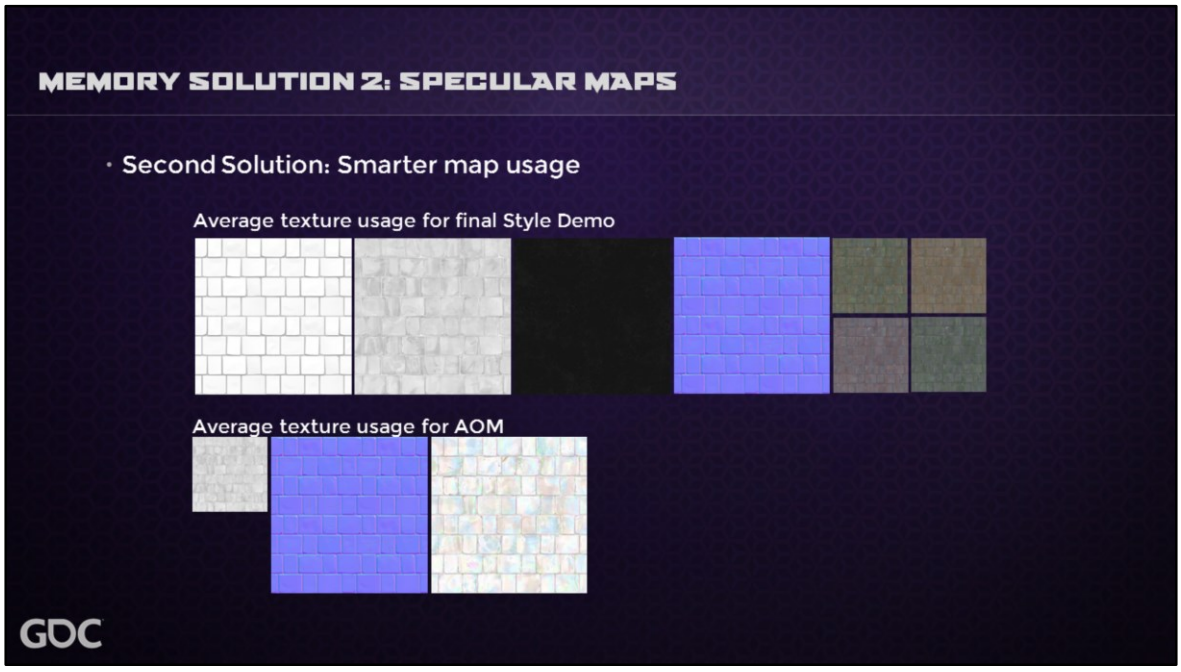
We also conceived of a system to derive an occlusion map from the normal map, but didn't feel the gain in quality across the scene was worth the additional shader cost of generating it.



Finally, we found that the fidelity of the roughness map did not need to be the same as that for the diffuse or Normals, so we simply made them half the size.

This is the opposite of the usually accepted wisdom (including our own), but we found that since the response usually takes up such a small slice of the screen, and because our diffuse textures are relatively high resolution, a lower fidelity in the roughness is not as noticeable as one in the diffuse, given our stylistic choices. In the cases where we needed a larger roughness, we could, given that it wouldn't be for *every* material.

And of course instead of multiple small coloured textures, we could not use one larger ones many times, as previously discussed.



This still gives us more texture per available MB than on previous titles, but it's much closer to parity, and tinting allows much more flexibility per set of maps. Furthermore, we can more easily mix and match Roughness, Diffuse and Normals from different materials to create new ones with no additional maps.

With all these memory savings, and others we don't have time to cover, we got the cost per material down by more than half, though it was still twice as much as we would have liked.

However, this was only the average size per material if the material is taken in isolation.

Since textures were shared among materials, and many materials were identical other than significant color changes, the average size when taken over many materials drops significantly to only slightly more than the Preferred amount. Once compression is again considered, the difference between a basic AOM material and the Preferred size is almost equal. Of course we had a lot of specialist 'expensive' materials too, but almost none would need to be larger than the orange bar.

Some of these solutions did increase the cost of the base shader, but the only time this ever became an issue was with Alpha textures, and even then the additional cost was fairly minimal.

MEMORY SOLUTION RESULTS

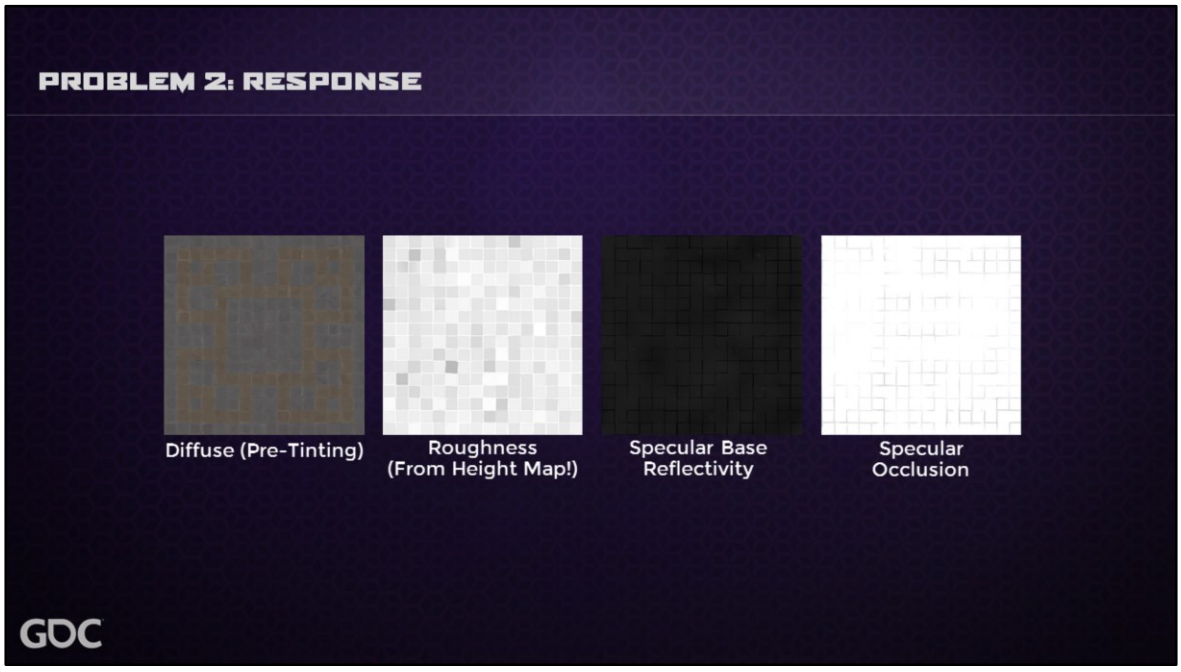
- Larger textures could be used where they counted most
  - Texture usage tailored to best fit a given material's needs
  - Resulted in 'cheap' and 'expensive' materials
- We could retain the Painterly style
  - And use fewer textures doing so
  - The savings on memory came at the expense of additional shader cost
- Memory still got tight toward the end
  - Much later that it otherwise would have been.
  - *Partly* down to poor layout planning.

GDC

While the Diffuse style, if not the solutions to having a lot of it, were clearly defined in the Style Tests, the Physical response was less well thought out.

Frankly, the response was initially authored as an afterthought, and this was evident once we moved beyond the single lighting scenario of the style demo.

Artists had received in-depth PBR training, but no insight as to how this would apply to the more stylized environments.



Some tried to ignore Specular Base Reflectivity alltogether, some were varying Specular Base Reflectivity too much as this was ‘more stylized’, and some were putting too much detail into these maps for a similar reason. Others were authoring them with a flat value, but using large textures to do so.

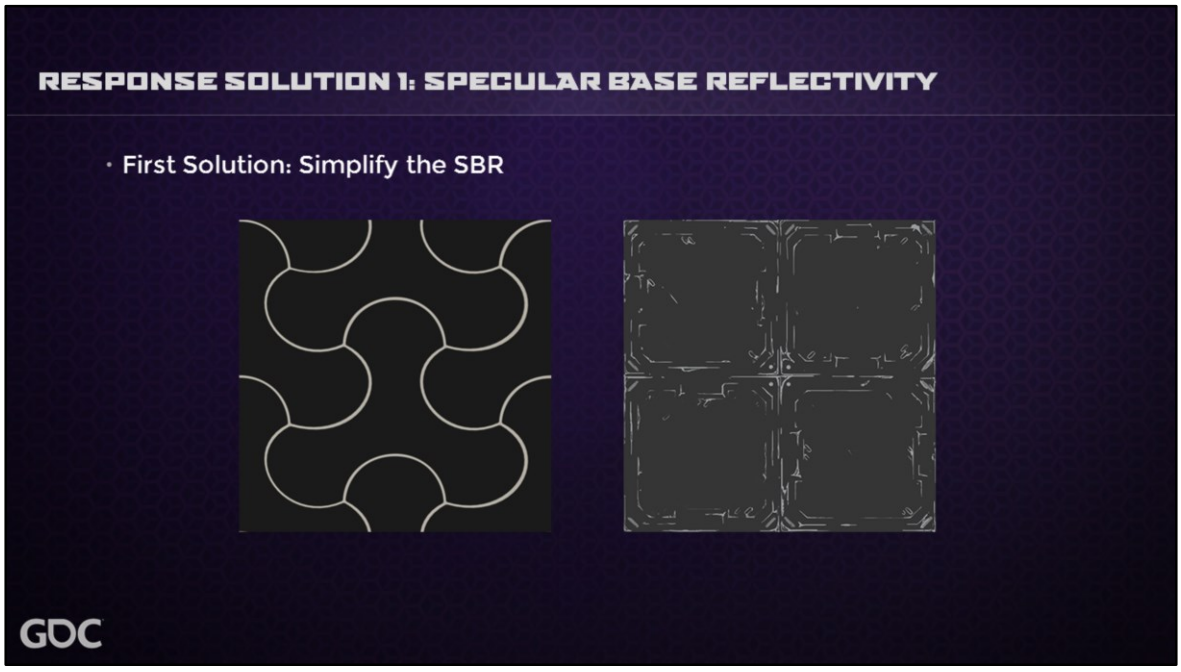
Usage and style of roughness varied greatly from artist to artist as well – in this example the roughness was just duplicated from a hightmap! None of these approaches were ideal, and they were wildly inconsistent.

We’ll come back to them much later, but even external vendors who were PBR knowledgeable were sometimes knowledgeable only on a Metalness workflow, and were getting a lot of things wrong as well

For the discrepancies in the specular base reflectivity maps, we established a set of more ridged rules that had to be abided to than for the other maps, where some interpretation by the artist was expected.

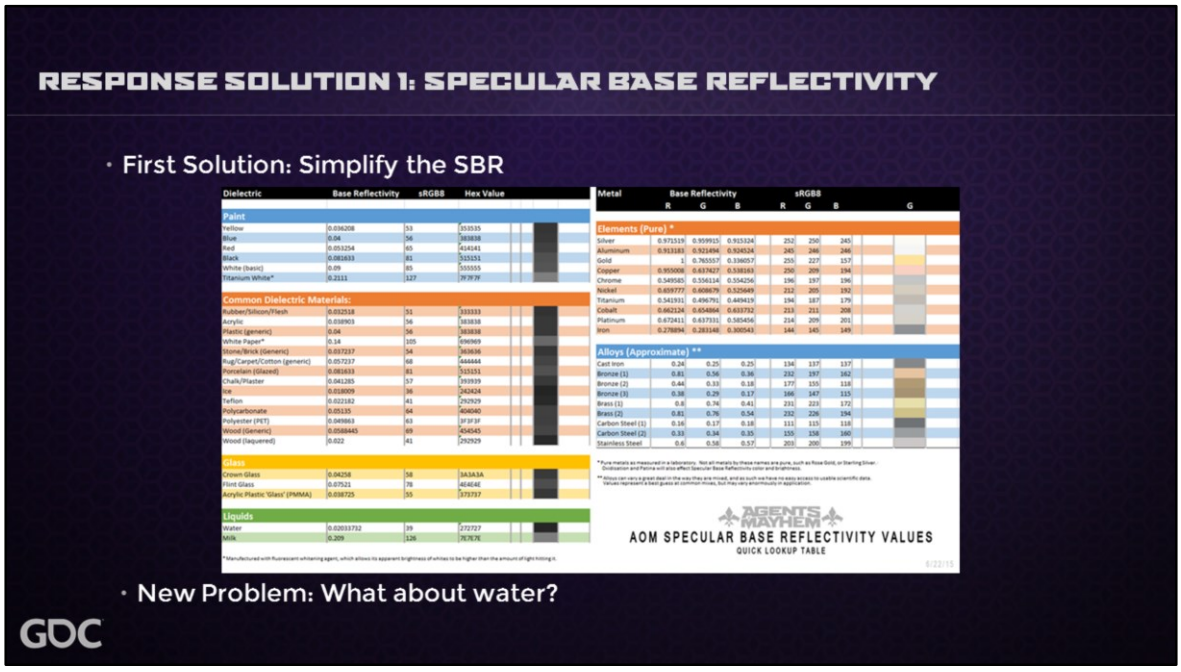
If a map had predominantly only one material type, it would be replaced by a flat value for the whole material. The reasons for this compromise were mentioned previously (under the memory solutions), but ultimately the values for grout and brick, or two kinds of plastic, were close enough that the differences could be ignored across a single material.

Materials that still needed SBR maps could be authored, but only drastic changes in materials should result in a change in SBR value, to visually tie the material in to the ones with no changes.



A Little grime on a surface wouldn't warrant a change, but thick oil or paint might, as well as metals, as in these two examples.

You can also see that the concrete on the left has a different Specular Base Reflectivity value to the paint on the right – This allowed us to still clearly differentiate the response between material types – but they are flat values for each, tying in with those that have just a value to represent SBR values.



To force correct (where possible) and consistent SBR values, a table was made with common materials listed.

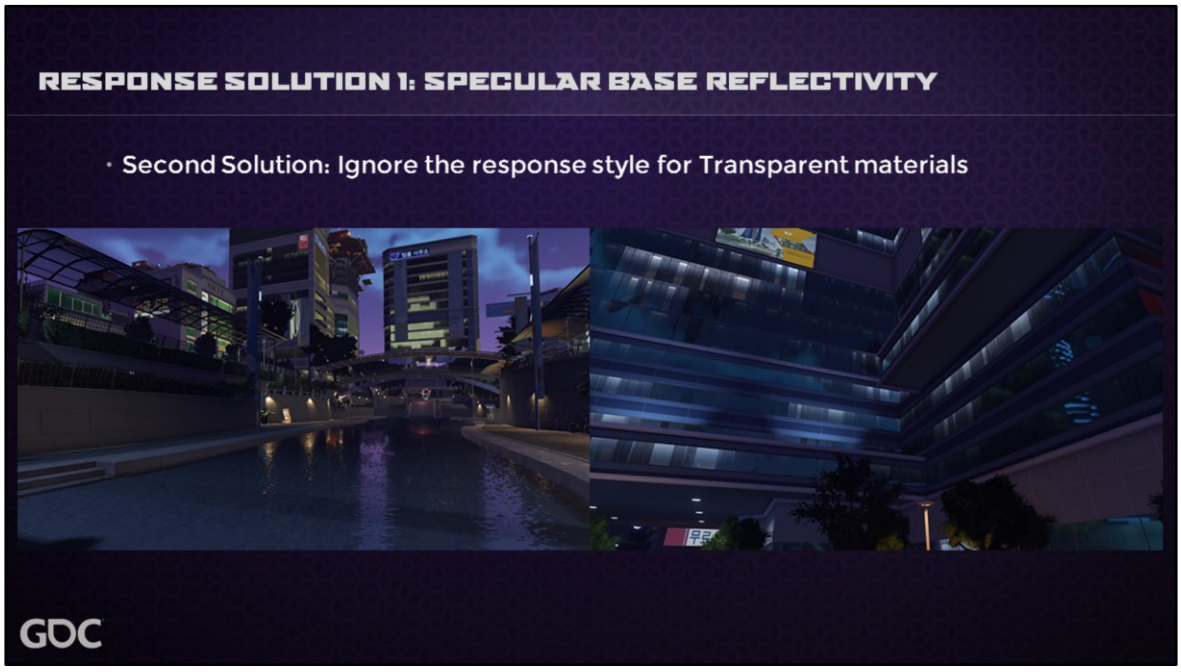
Right or wrong, these values were the AOM values for those materials, and had to be adhered to (we tried to get them right wherever possible).

A mechanical solution, rather than artistic.

The exception to this was for metals. We had Lab Accurate values for these, but variance was acceptable for build up of grime, weathering or the metal being mixed into an alloy with another.

Pure gold looks different to 12 carat gold for example.

The given value was still expected as the initial base value to work from.



With surfaces such as Water and Glass, which have minimal diffuse components, we tried using the diffuse style in the SBR instead. It did not work well.

We eventually compromised – SBR for these materials are usually a flat value, but if there is a diffuse component (such as for glass that is not actually transparent), we could use a small amount of the style in that map. Roughness in these cases was used if the material was supposed to be older and more worn, but water is a flat response, relying only on changes in the normals.

This worked out pretty well, and hasn't been questioned by anybody as being 'not in style' since we started doing it – usually an indicator that we did the right thing.

The solution to the problem of inconsistent roughness seemed the most obvious.

Roughness maps would follow the same style conventions as the diffuse, even if not being identical to it; though in many cases the diffuse could be used as a starting point.

Compare this initial roughness, which was taken from the material's height map…

…to the one based on the diffuse.

Obviously this did not apply to colour shifting part of the diffuse, as the roughness was greyscale

The amount of contrast in the roughness would be gauged on a case by case basis, in order to get the most archetypal response from each material.

One side effect of this was that the roughness sometimes still seemed very flat, as only a small wedge of it was being prominently seen at any given time. This example works well, as there is a lot of breakup in the form through the normal and parallax, but flatter surfaces suffered as a result.

In order to solve this new problem the decision was made to ignore the diffuse style *under certain circumstances*

The diffuse style should still be used as a base, but additional detail, and even noise, was deemed acceptable if it aided the material response – we could use *higher detail*, if it was *lower contrast*.

Since it was seen in detail only in small amounts, this did not unduly fill the scene with noise, and aided in selling the material types even when not in direct illumination thanks to the general reflectivity of Physically Based materials.

There were initial concerns that this could break the style. Tests on a small number of materials alleviated this concern.

We didn’t use this technique in every material, only those that needed that additional level to sell the surface; especially otherwise quite flat surfaces such as painted concrete.

RESPONSE SOLUTION RESULTS

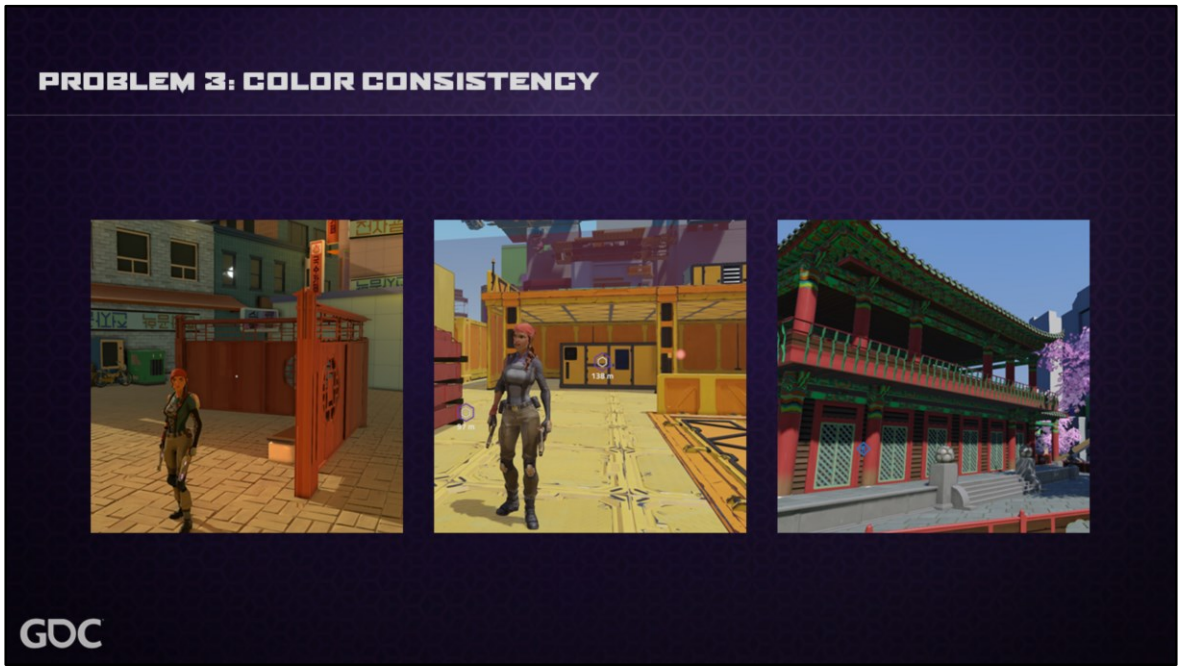
- Base Reflectivity was now more consistent across the board
  - Even if not the correct value for a given material, it was a consistent 'mechanical' solution.
  - Resulted in smaller memory footprint overall, as mentioned before.
- We could retain the Painterly style, even in response!
  - Still simplified, though noisier than the diffuse.
  - The overall result was more consistency between materials
  - Still allowed for a lot of artistic interpretation
- The one area we weren't sure about didn't matter
  - Water and Glass worked out fine with minimal style consistency
  - We think this is because they are things in the world people 'just accept'

GDC

Use of colour was all over the map.

Some textures took directly from the work concept artist Vasili Zorin had done for us. Vasili's palette and colour choices are distinct and unorthodox (though gorgeous!)

Others were eyeballed from photographs or cartoons



Each artist just used what they liked, based on the Vibrant Pillar and often the colours were just horribly saturated and bright as a result.

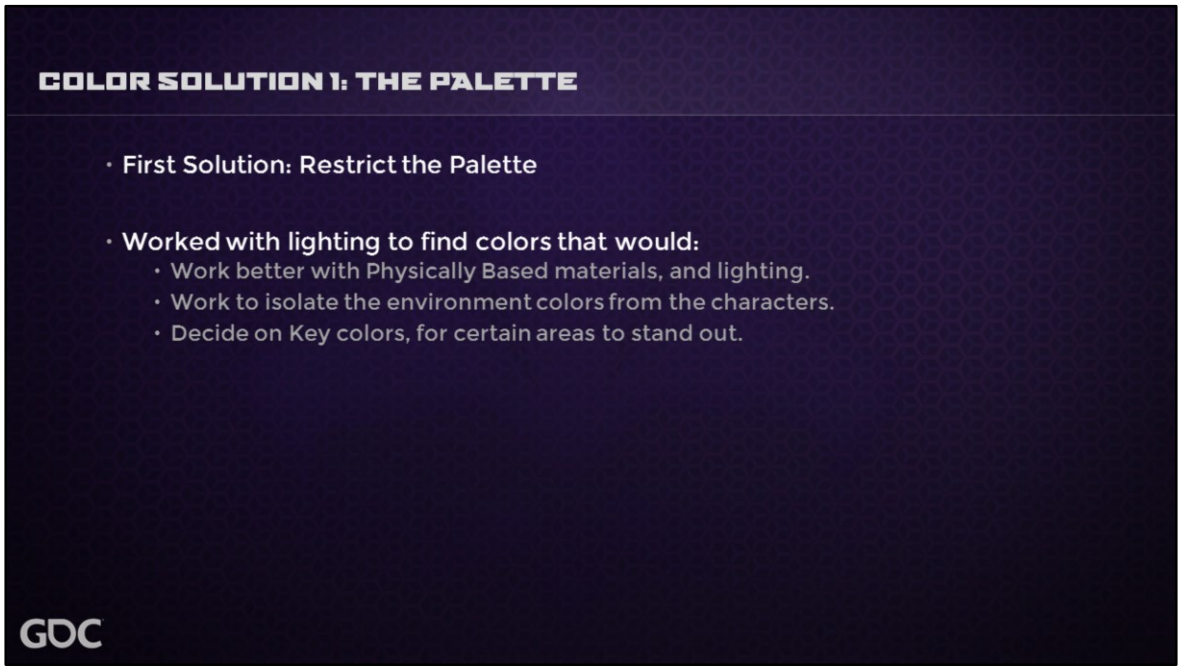
We had minimal art direction to keep things in line, due to not having an art director at the time (we had one eventually, but we'd solved this by then)

There was a general dislike of the results… OK, a severe dislike

We were also losing the Player Characters in the scene, as the lurid environment was overpowering the colours used on the characters.

The majority of the team played the game looking like this, to avoid the eye gouging colours! Nobody wanting to see your materials is a good signifier you're doing it wrong.

We'd lost the harmony! From looking at other games known for their strong consistent stylization, such as Team Fortress 2 and Dishonored, we found that one of the things most central to that consistency was not shape language, or textural style (though both were important), but colour. Fixing this was central to making the style of the game consistent.

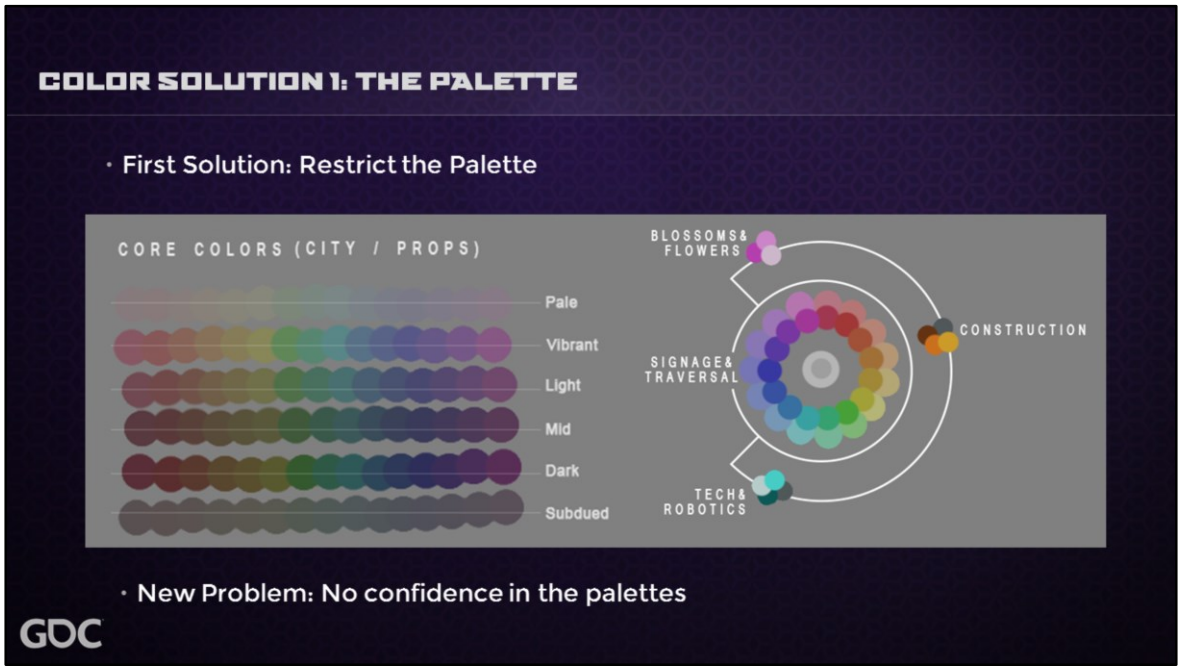


To solve this, we turned to the lighting team and worked with them to establish a palette that would do the following:

-Work better with Physically Based Rendering
-Work to isolate the environment colours from the characters, while still keeping the environment bright and vibrant.
-Decide on Key colours, that would allow certain areas to stand out as significant within the city.

After a short time, Lighters Susan Cenci and Troy Williams got back to us with their solution.

The result was not initially embraced.



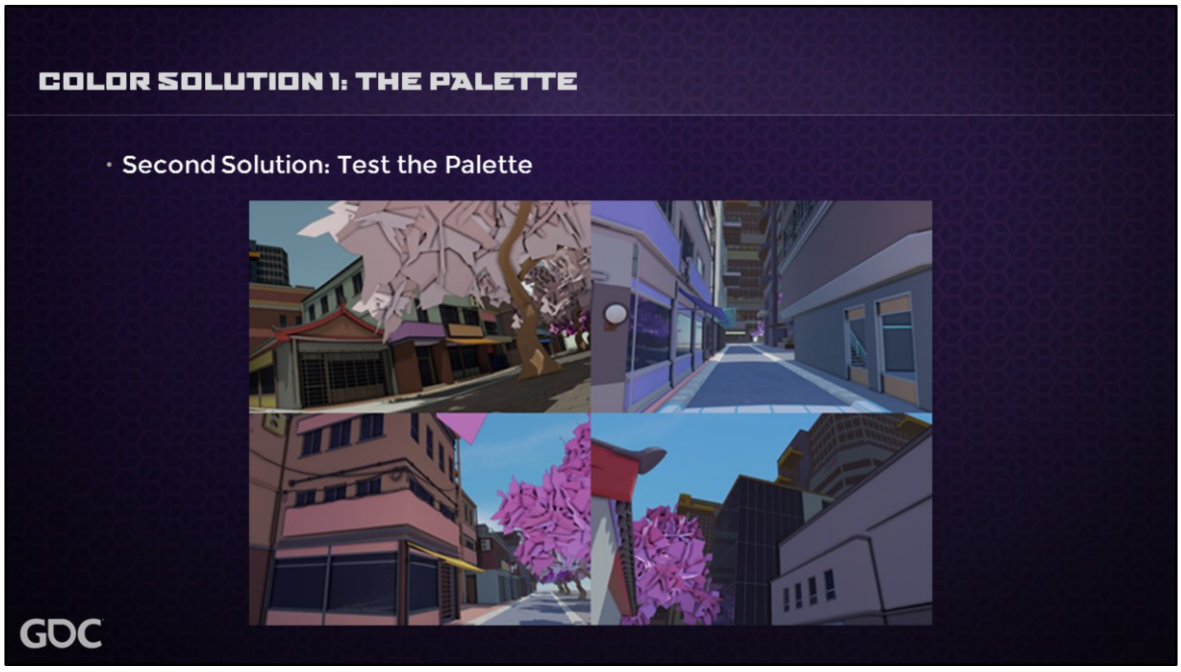
The ‘core’ Palette looked dowdy and sad – not at all the bright vibrancy we wanted around the city. This initial version had no blacks and no whites either, just colours. The ‘key’ palettes were better, but had a much more limited tonal range.

There was little confidence that this would work.

The easiest way to instill confidence in the palettes (which in all honesty I lacked myself), was to test them as quickly as possible.

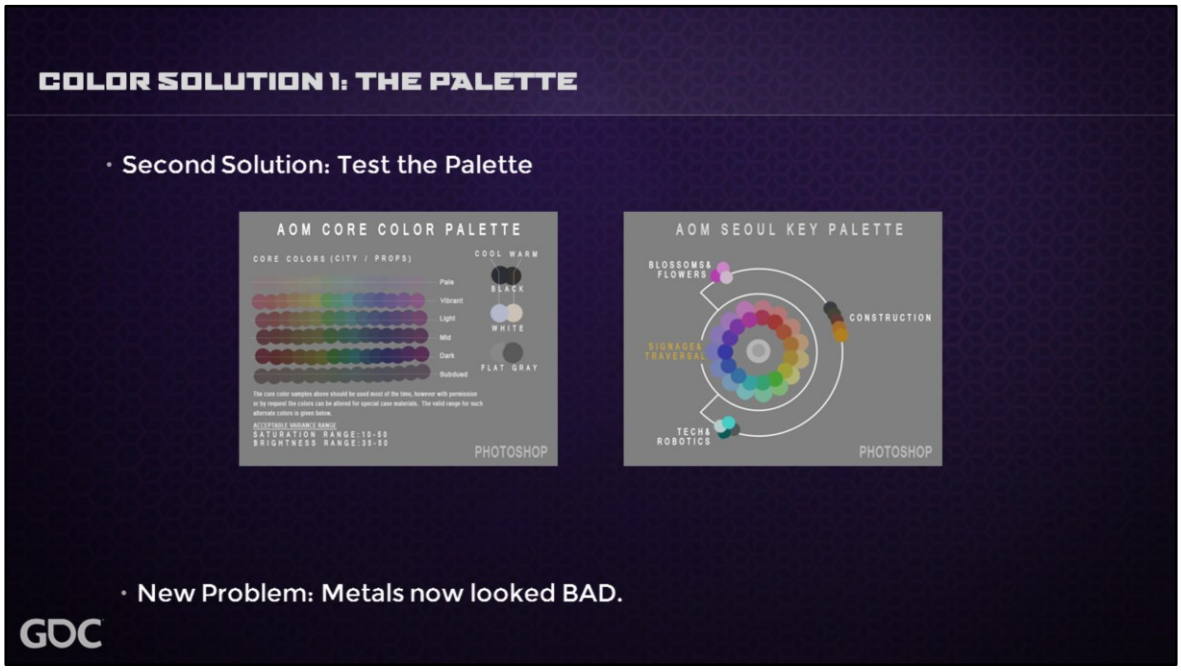
To test the palette, we isolated about 16 blocks of the city, and created a flat colour material for each of the colours in the new palette; both key and core.

These were then applied to the buildings and trees to see if these would work for our needs.



As you can see, this did indeed restore harmony to the city, and the result was attractive in it's own right, even with no other material aspects.

Some compromises were reached though:

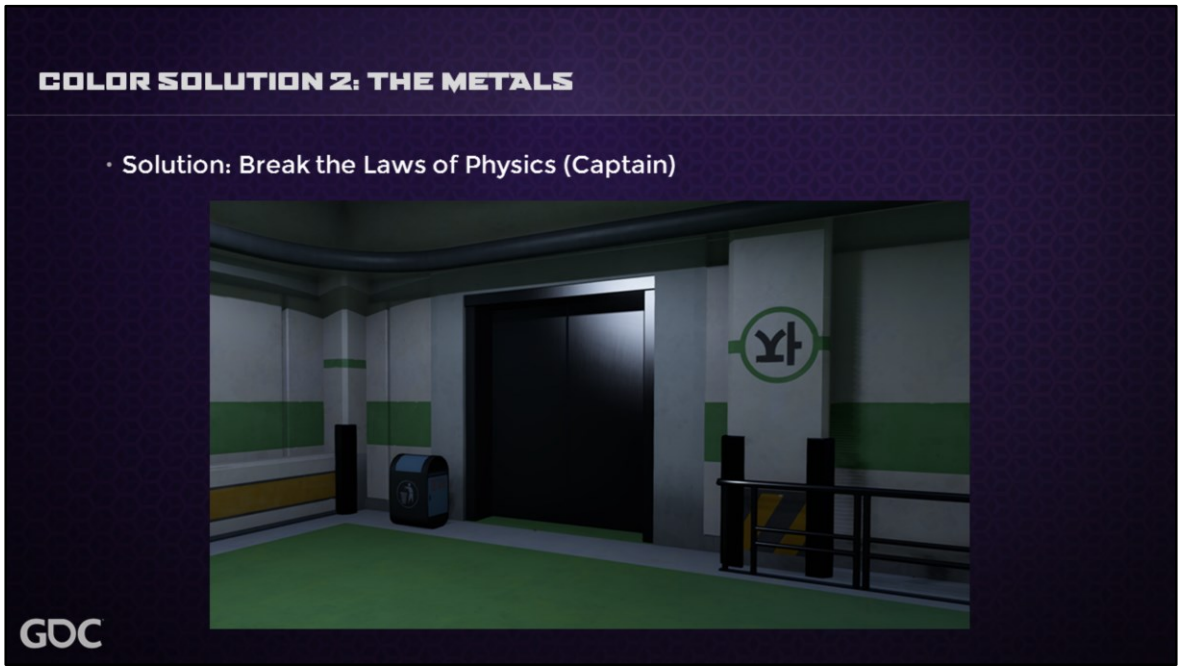


The addition of a warm and cool “Game Black”, which was 21% grey with a slight tint, and “Game White”, which was 80% grey with the same tint. This was *usually* the darkest and lightest Diffuse textures were allowed to go.
The allowance of a ’range’ that colours could be chosen from if no colours in the palette would otherwise be suitable, but the majority of colours would be chosen directly from the available swatches.

We also changed the Construction Palette to be more muted in tone, as the fairly unique colour range would be enough to highlight it.

The exact same core palette has stayed consistent throughout production, but the key colours, have had a few revisions and additions.

This fix did not work for metals though, in fact it made things worse.

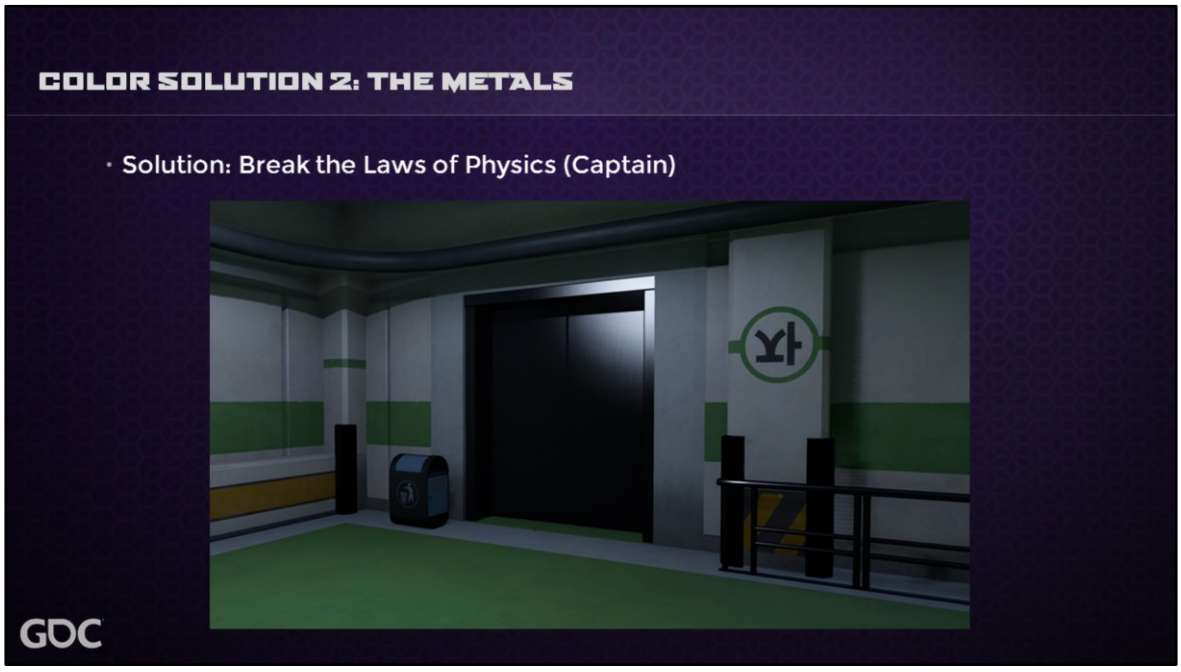


It was soon noticed, after the implementation of the new palette, that Metals now looked ‘off’ and out of place in the world, especially in deep shadow or at night.

We realized this was because we’d ‘cheated’ the palettes to reduce contrast, so “Game Black’ was not actually close to as dark as an actual black object might be in reality
(though they wouldn’t be truly black, they’d be a lot closer to it)

Since the Diffuse Albedo colour of Metal is COMPLETELY BLACK (or as close as makes no odds to a game engine), we had a problem when metals were reflecting black in the world – they were actually black, instead of Game Black.

In fact this is how a Metalness workflow works – the parts of the map masked as being metal are changed to black in the diffuse, and the SBR takes on the properties of the colour painted in the diffuse. Many artists are not aware that this is happening, but it is.



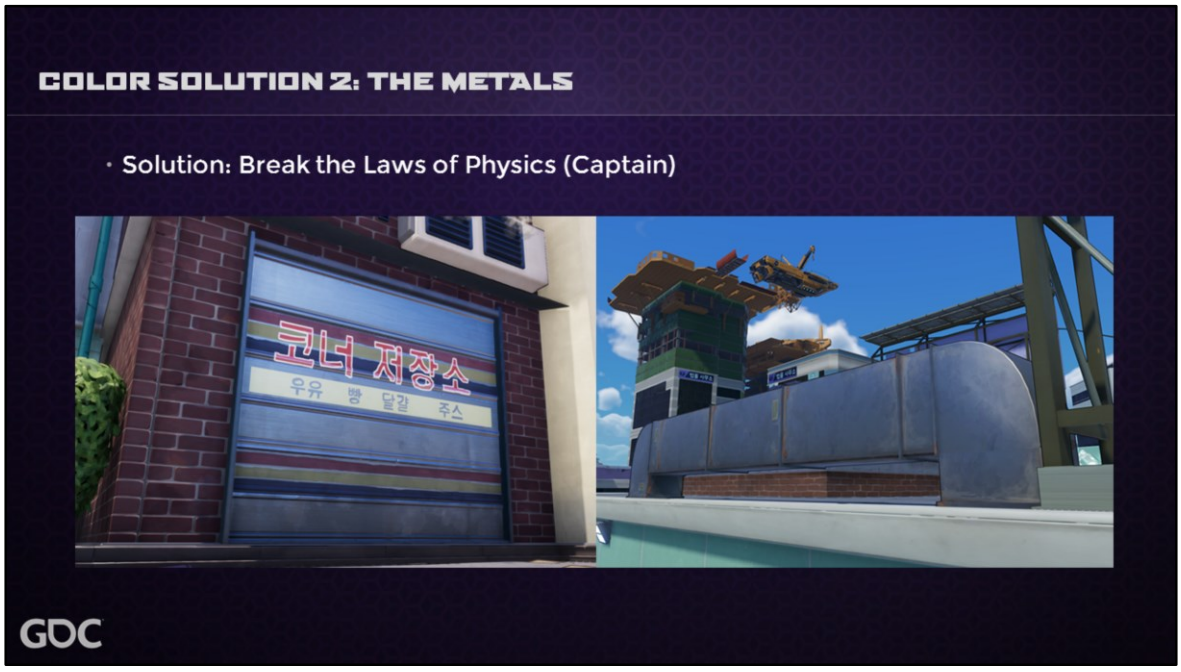
(Continued)

We realized that within the world of AOM we had to break the laws of physics for stylistic reasons, and switch true black for game black, or at least close to it.

Because we had tested this to verify the papers we'd read long before, we had instilled this Metal is Black as a fundamental thing in both our studio, and our outsourcers.

Now we had to tell them that we'd changed our mind. About a law of physics.

Awkward.

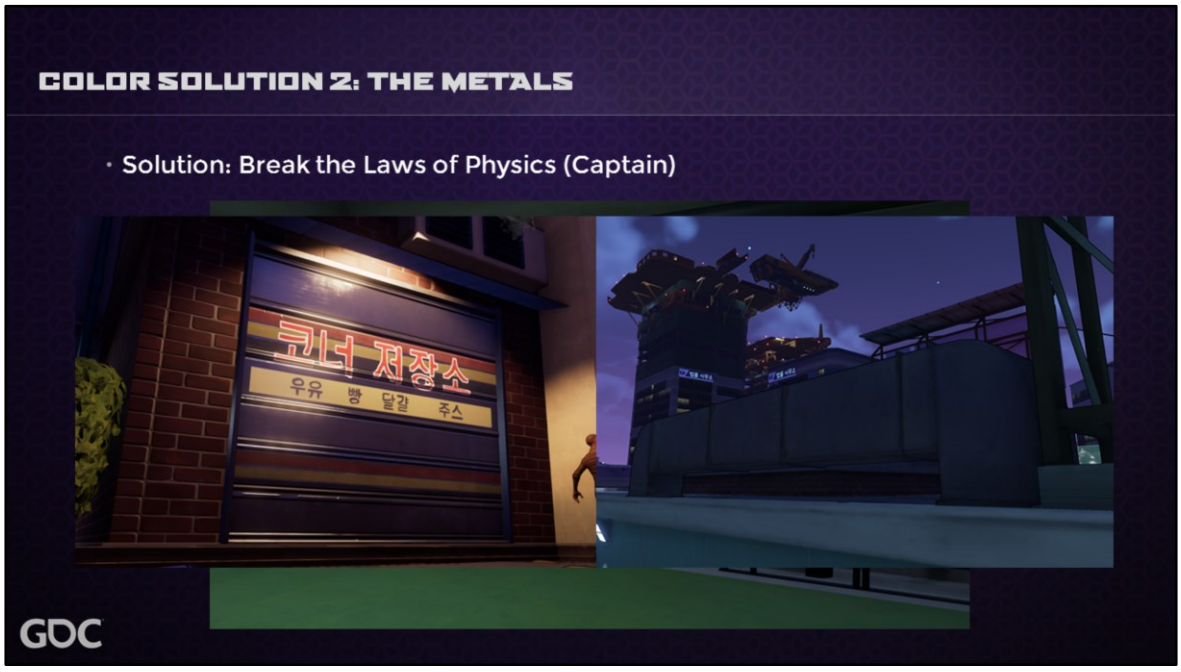


We did though – Diffuse colour for metals became the colour of the metal, but at the brightness of Game Black, minus a little (usually 2 or three percent). Despite the metals being and looking less real, they did look more consistent, and thus more real in context.

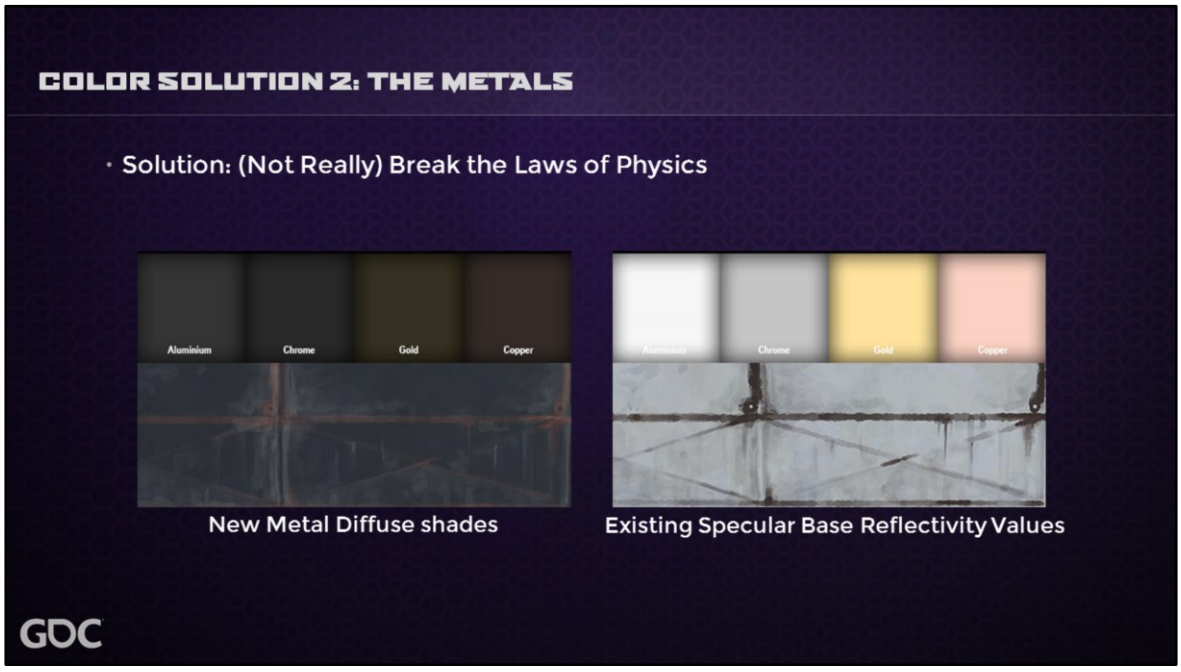
Technically wrong, stylistically correct. Even at night.
We did maintain the colour of the metals as the real colours though

OK, we weren't technically breaking any physical laws (except in the cases of some, like Gold, Silver and Aluminium – breaking the conservation of energy), but we were making our metals not really metal any more.

You can see here, that these two surfaces still read as metal in the daylight.



And at night. The ducting on the right no longer looks like a black hole into another dimension, despite having no immediate illumination.



These are the new shades we used as the base diffuse colours of metals, and a grimy ducting texture to show them in use, as in the previous example, and is part of the imagery shared with our external vendors to illustrate the change.

COLOR SOLUTION RESULTS

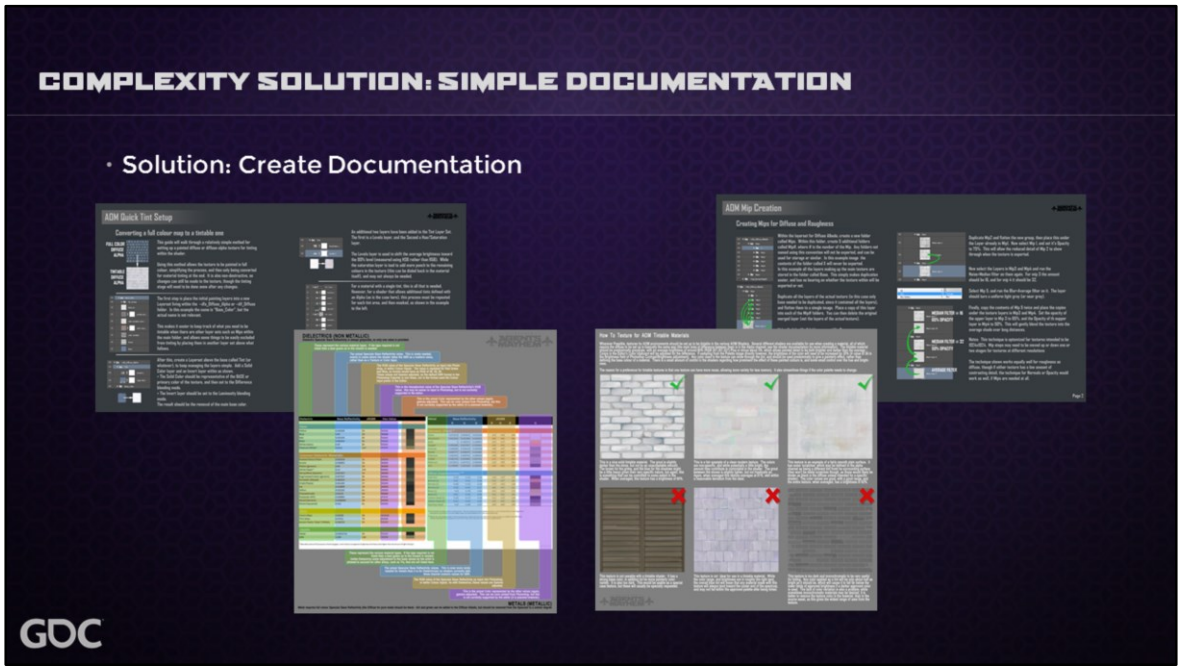
- People were much, much happier with the materials
  - Harmony was restored, and vibrancy maintained.
  - People stopped playing in greyscale!
    - (mostly)
  - It was easier to gauge how much of the city was presentable, and how much was still greenbox.
- The Characters and enemies were much more visible
  - This helped gameplay a lot.
  - Made the game more harmonious.

GDC

The final problem was that with all these new guides to consistency, in so many areas (and I've skipped a few for time reasons), any new people to the team, either internally, or at an external vendor, had a terrible time taking it all in.

Feedback took a long time to deliver, as so may seemingly small elements were wrong, but added up to something wildly inconsistent (and we were accounting for individuals styles not perfectly gelling, which was considered acceptable in light of all the other tools to maintain consistency). Often we were writing basically the same feedback, and doing the same paintovers repeatedly.

This got messy very quickly, and needed to be fixed. Fortunately we were transitioning between vendors at that point, which gave a little breathing room for finding and implementing the solution.



In fact the complexity was so high at this point, that even we had difficulty remembering all the things we had to adhere to.

So the really obvious solution was to document everything. We had documentation for a lot of these things already, but it wasn't in an easily accessible place or format. Even the Materials team had to dig around in order to look something up.

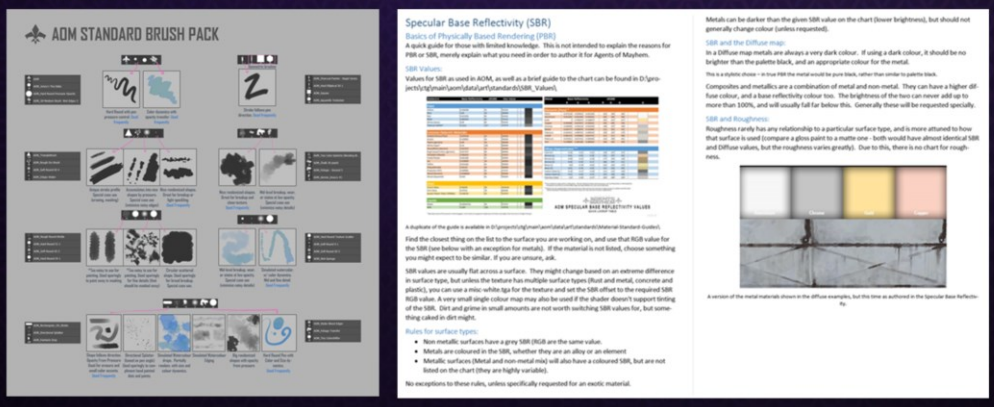
So we'd need to document things in a way that someone with moderate English comprehension could follow.

This even applied to PBR since, as we've previously noted, even vendors familiar with it might know it mainly through a Metalness system.

The resulting documentation had text, but not a huge amount, and lots of images showing common procedures, and was mainly stored as images, rather than documents, which could be found by looking at thumbnails.

COMPLEXITY SOLUTION: SIMPLE DOCUMENTATION

- Solution: Create Documentation



GDC

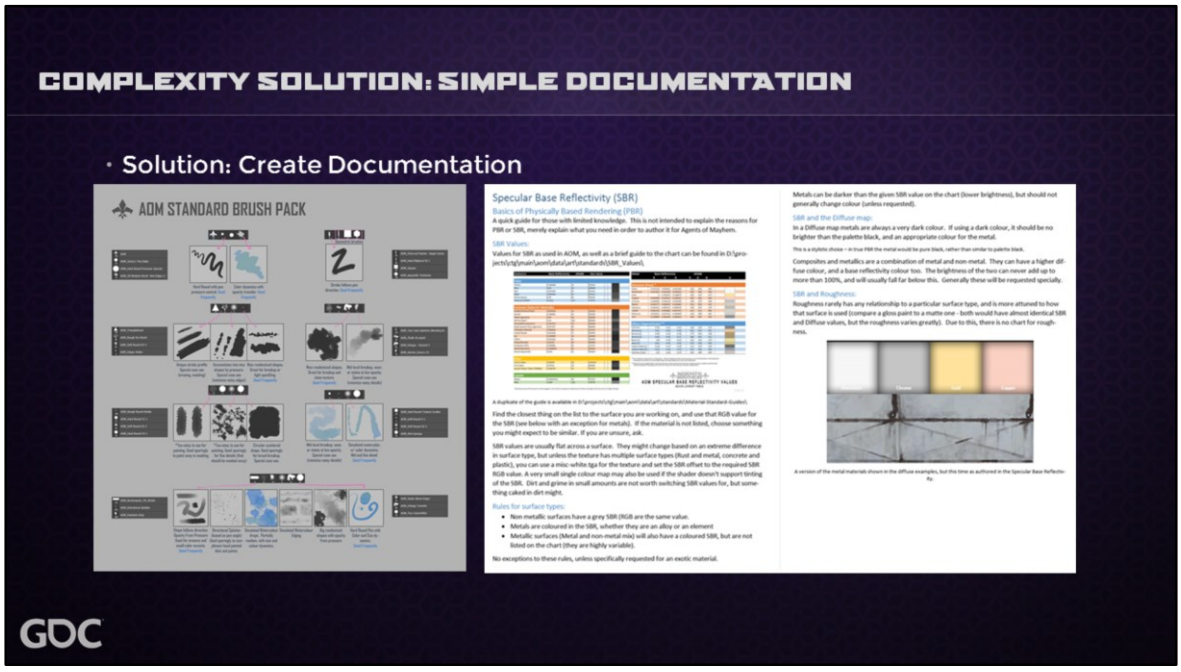
We also wanted to increase consistency in painting the textures themselves, so as well as Documentation, we supplied resources.

Photoshop Brushes were supplied, with instructions not to use other ones without asking for prior permission (a small number of extra ones were added over time).

We supplied actions, and palettes as well, and were quite anal about having the Vendor's use them.

Finally, there was an actual Word document breaking down how to use PBR. We couldn't teach the fundamentals, since we didn't know what level each Vendor was at, but we could break it down to a method that, if followed, would give the correct results.

We also needed all of this somewhere easily accessible to both the internal team, and the external vendors



(Continued)

All of these things were embedded in the Art folder of our game, which all artists share, rather than sent through FTP or put on something like One Note.

This ensured every artist had access to the exact same files immediately when they were written, or at least checked in

A combination of these things vastly simplified a lot of feedback – rather than having to write in detail what was stylistically wrong with something, it was often possible to link directly to the image that explained the correct way to do things.

COMPLEXITY SOLUTION: SIMPLE DOCUMENTATION

- Solution: Create Documentation

- Problem: You can't force people to read it.

GDC

(Continued)

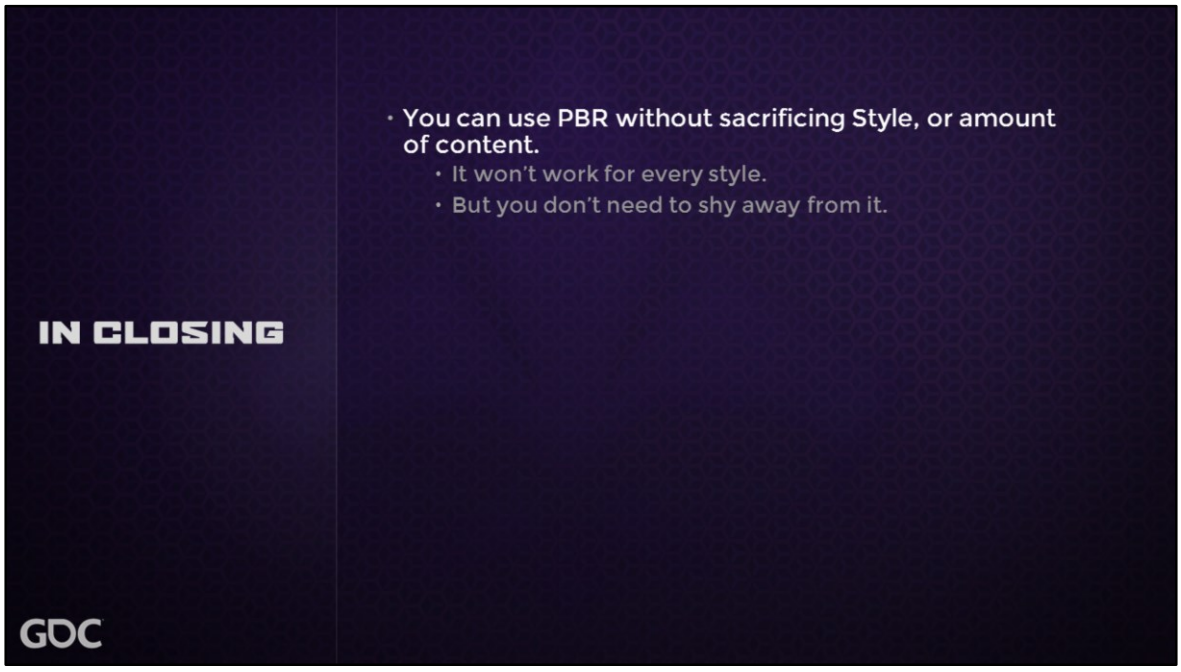
Once this had been done a handful of times, everyone got the idea, and could find their own documentation quite quickly. It is usually possible to spot when a new person has started at the outsourcing studio (or moved to our team there) as they get a lot of things wrong on the first asset.

We point them to the documentation and things are rarely wrong twice. Sometimes they are, because one drawback of this is that you can't force people to take a look at the documentation, even if it saves time in the long run.

COMPLEXITY SOLUTION RESULTS

- Materials became far more cohesive
  - Anyone working on the materials had a better time understanding why we do what we do.
  - Even when they didn't understand the *Why*, they could understand the *How* much more readily, which was much simpler to convey.
  - Supplying required brushes and techniques ensured a stronger similarity across diverse materials.
- Quick Lookup available to all
  - Even the experienced material team forgets things sometimes, this gave us an easy place to remind ourselves.
  - Mostly graphic guides made things easy to follow at a glance.

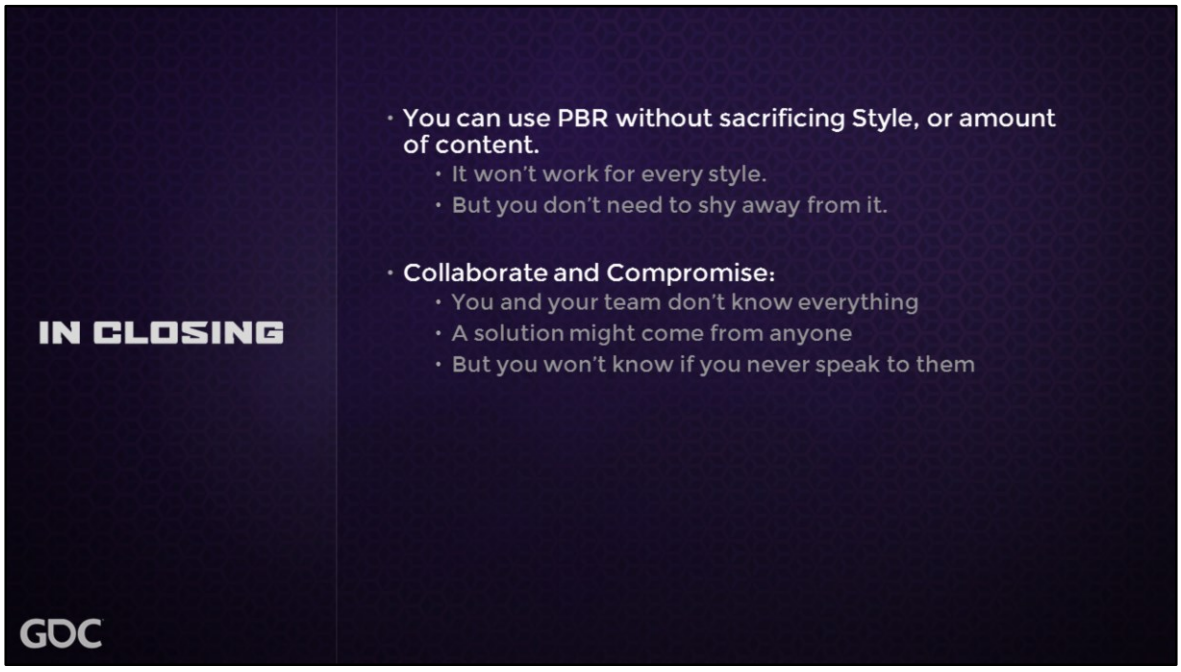
GDC



So, to close the talk, I'll go over some of the things learned. Not the things I just covered, because I just covered them, but things Those solutions have taught the team.

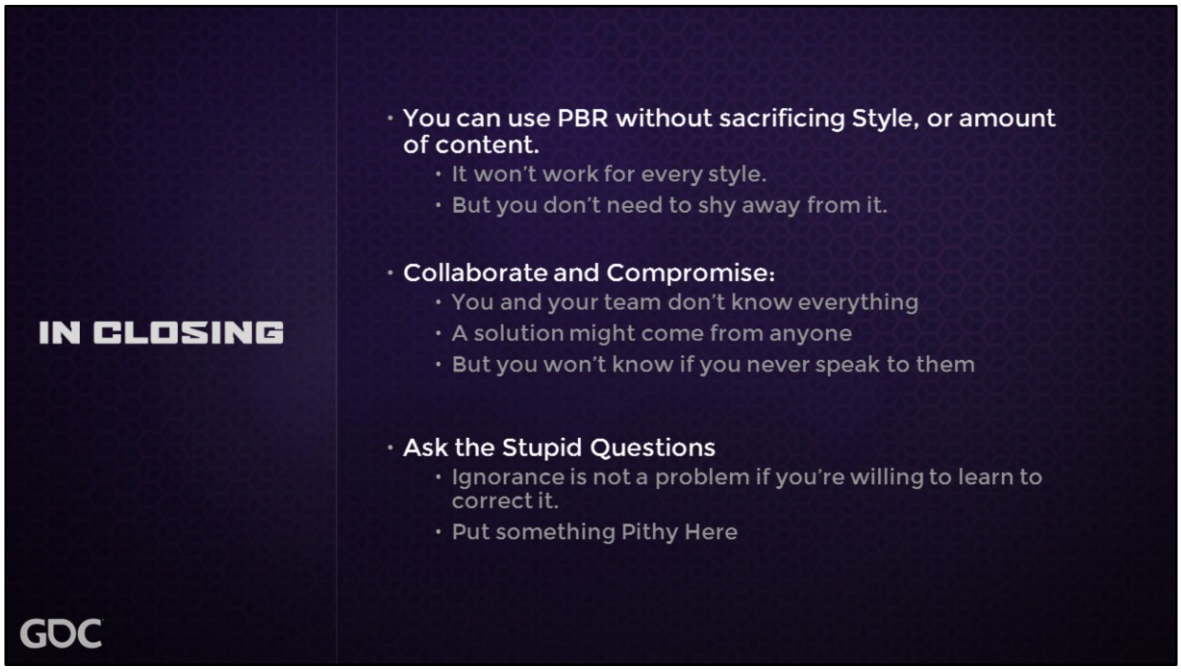
1 Don't fear PBR

PBR does not have to compromise a stylistic vision, but that doesn't always make it ideal for it.
Ardman animations, being Claymation, have a real physics built in, but are still clearly stylized.
If you want your game to look like 1960's Czechoslovakian cel animation, PBR likely isn't going to work for you.



2 Collaborate and compromise

Even though the Materials Team was it's own entity, we found solutions as often by going outside our team as we did internally. When working on the Palette, we collaborated with Lighting, and arrived at something that would work well for both teams. We didn't insist that our way was the right way, but we also didn't blindly accept what they gave us. We don't know their needs as well as we know our own, and Vice Versa, so collaborating on a solution worked out better for everyone.



3 Don't be afraid to ask stupid questions.

At least one of the solutions covered in this talk came from me asking what turned out to be a dumb question of the Rendering programmers. The answer to that was basically "Don't be Ridiculous, we can't do that", but in talking about alternatives we ended up solving a completely different problem – that of easily taking a fully painted texture and making it tintable in a way suitable for the engine. If I'd been too embarrassed to ask the initial question the Solution might not have presented itself, and I would have known less about the way the engine worked; knowledge of which would go on to help with other solutions.

Nobody thinks these questions are stupid, unless you continually ask the same ones, and clearly aren't learning from the answers.

I hope this talk has helped in some way to reduce fears of combining Physically Based Rendering with a stylized art direction, and has shown that even though it can make things more difficult to achieve initially, it can be leveraged to not only work well with the style, but to actually enhance it.

Questions?