Creating Content to Drive Destiny's Investment Game

One Solution to Rule Them All

Natalya Tatarchuk
Shiek Wang

BUNGiE®

Bungie’s Game History

The Early Years

Bungie’s Game History

The Last Decade

DESTINY™
BUNGIE

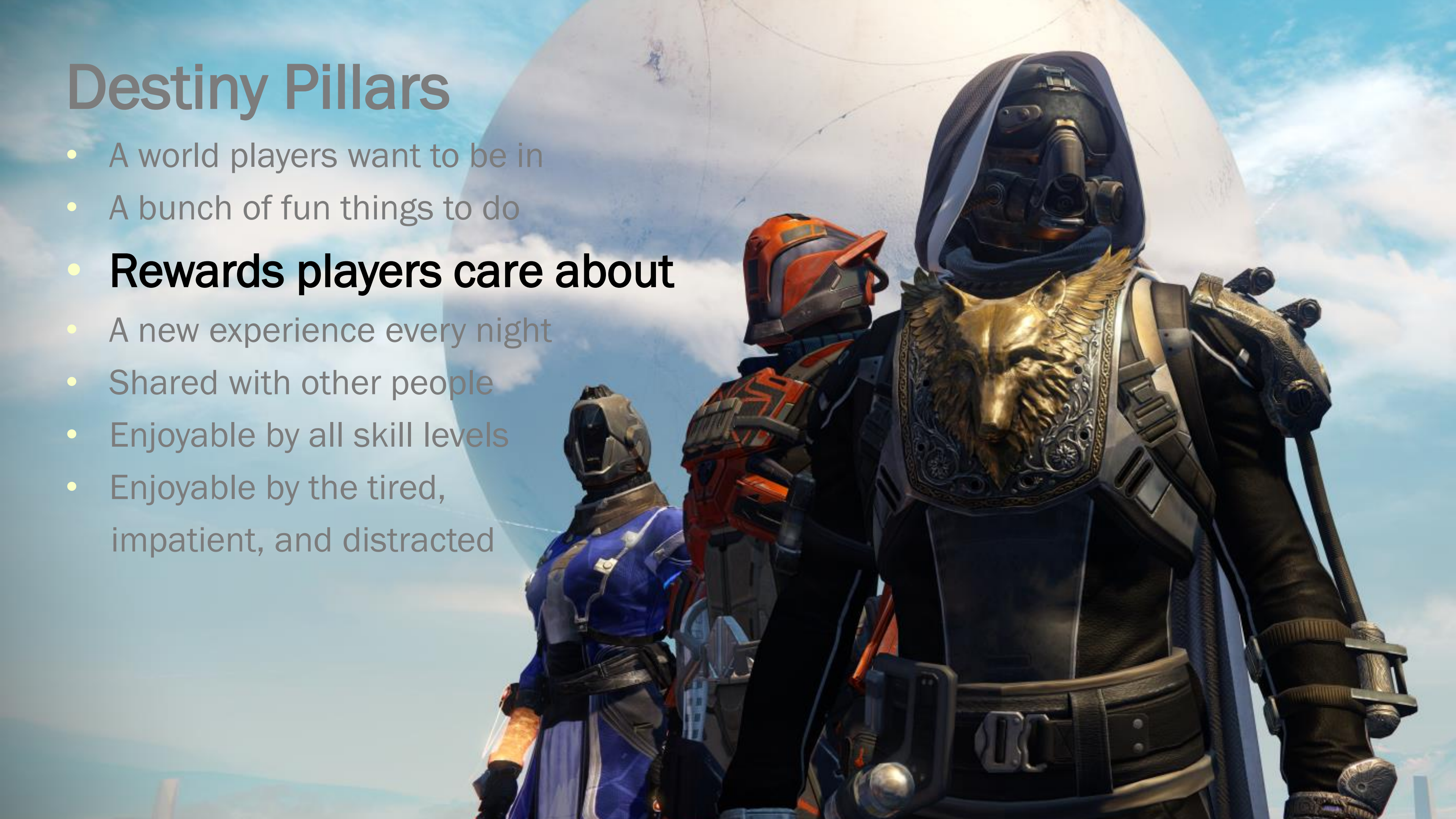
Destiny Launch Trailer

- http://www.bungie.net/7_Destiny-Launch-Gameplay-Trailer/en/News/News?aid=12050

Destiny Pillars

- A world players want to be in
- A bunch of fun things to do
- Rewards players care about
- A new experience every night
- Shared with other people
- Enjoyable by all skill levels
- Enjoyable by the tired, impatient, and distracted

Destiny Pillars



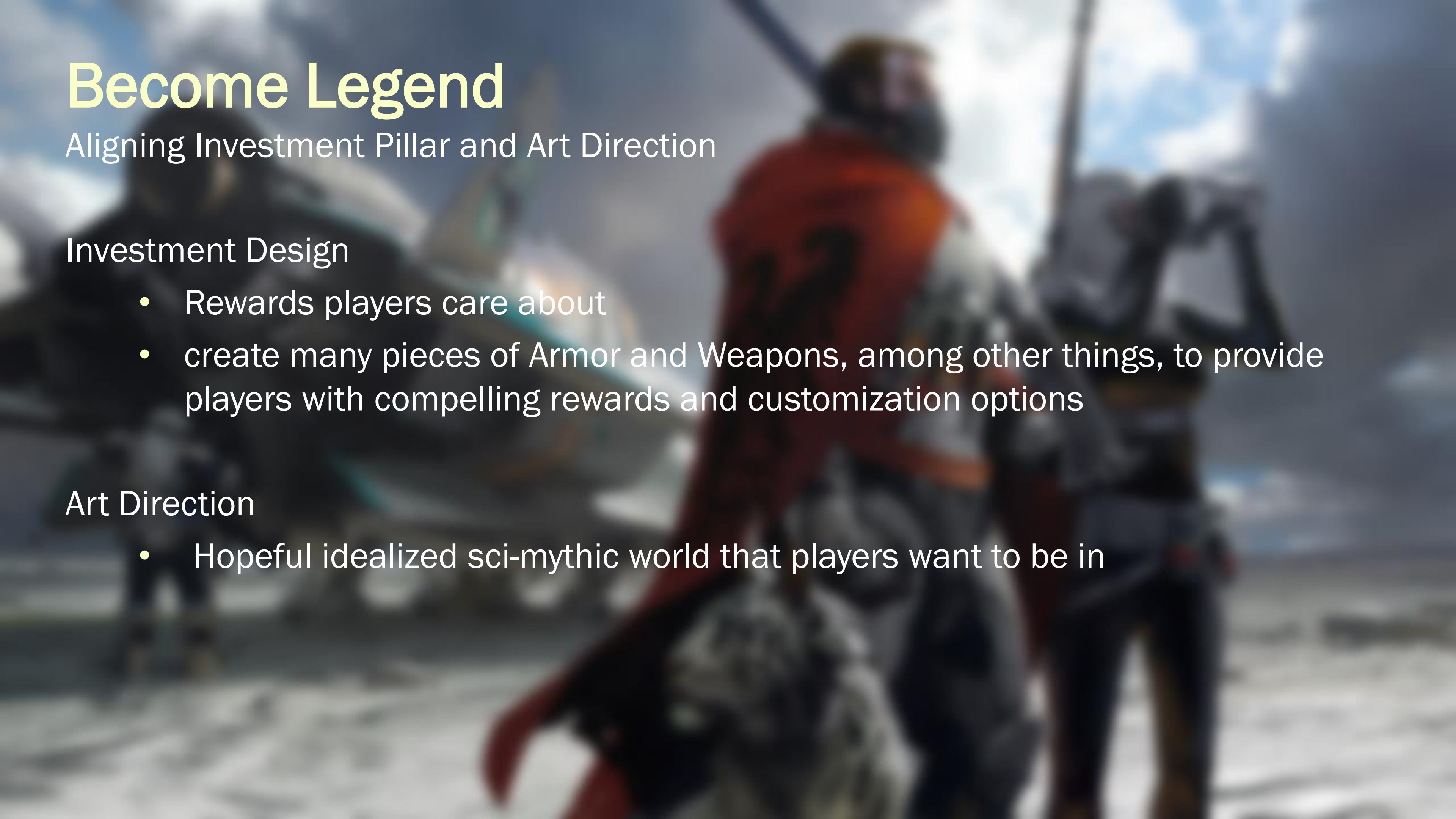
- A world players want to be in
- A bunch of fun things to do
- **Rewards players care about**
- A new experience every night
- Shared with other people
- Enjoyable by all skill levels
- Enjoyable by the tired, impatient, and distracted

Become Legend

Become Legend

Aligning Investment Pillar and Art Direction

Become Legend



Become Legend

Aligning Investment Pillar and Art Direction

Investment Design

- Rewards players care about
- create many pieces of Armor and Weapons, among other things, to provide players with compelling rewards and customization options

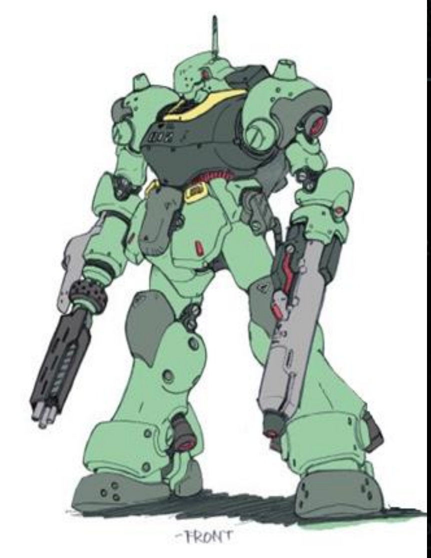
Art Direction

- Hopeful idealized sci-mythic world that players want to be in

Classes

Titan

Space Knight

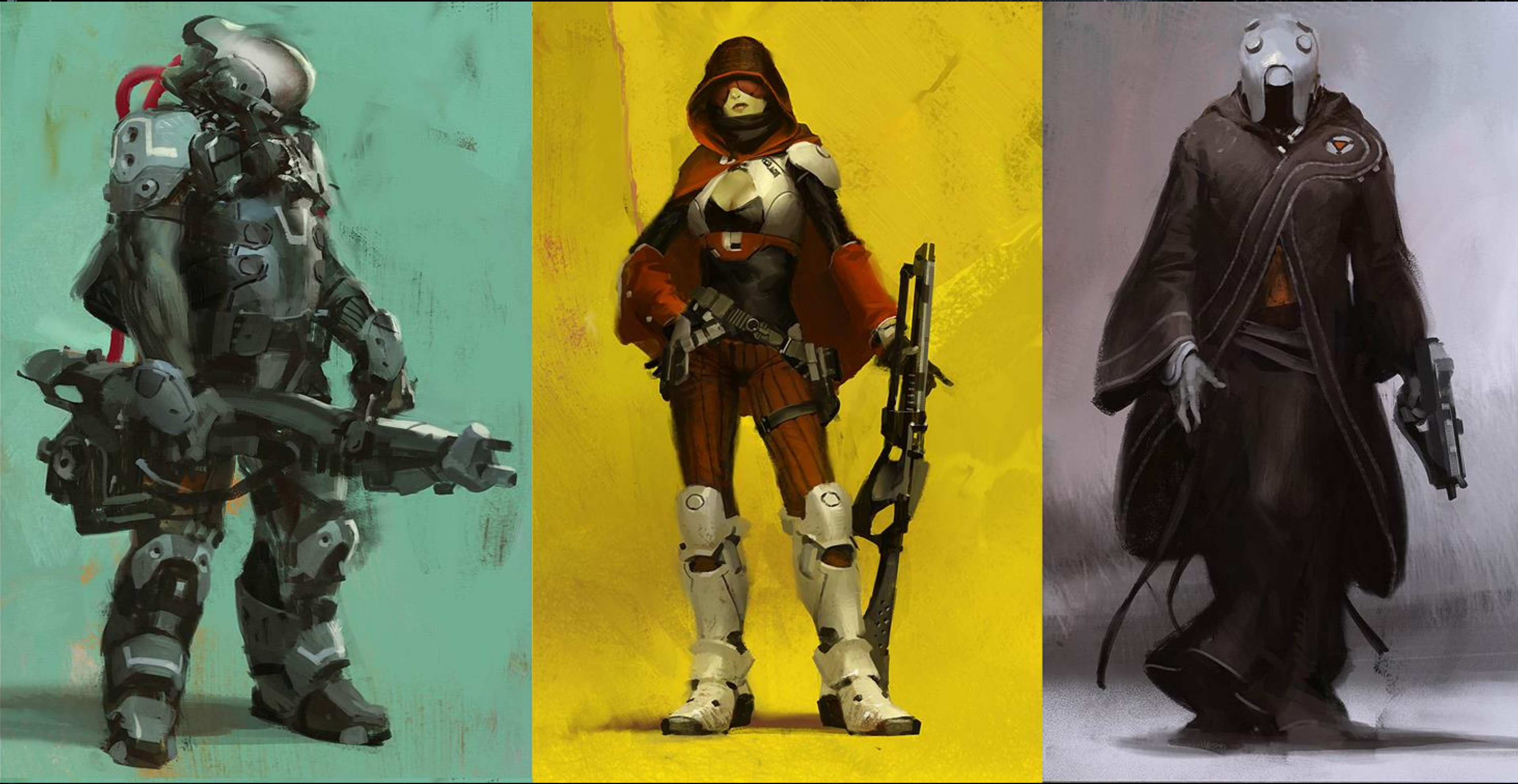


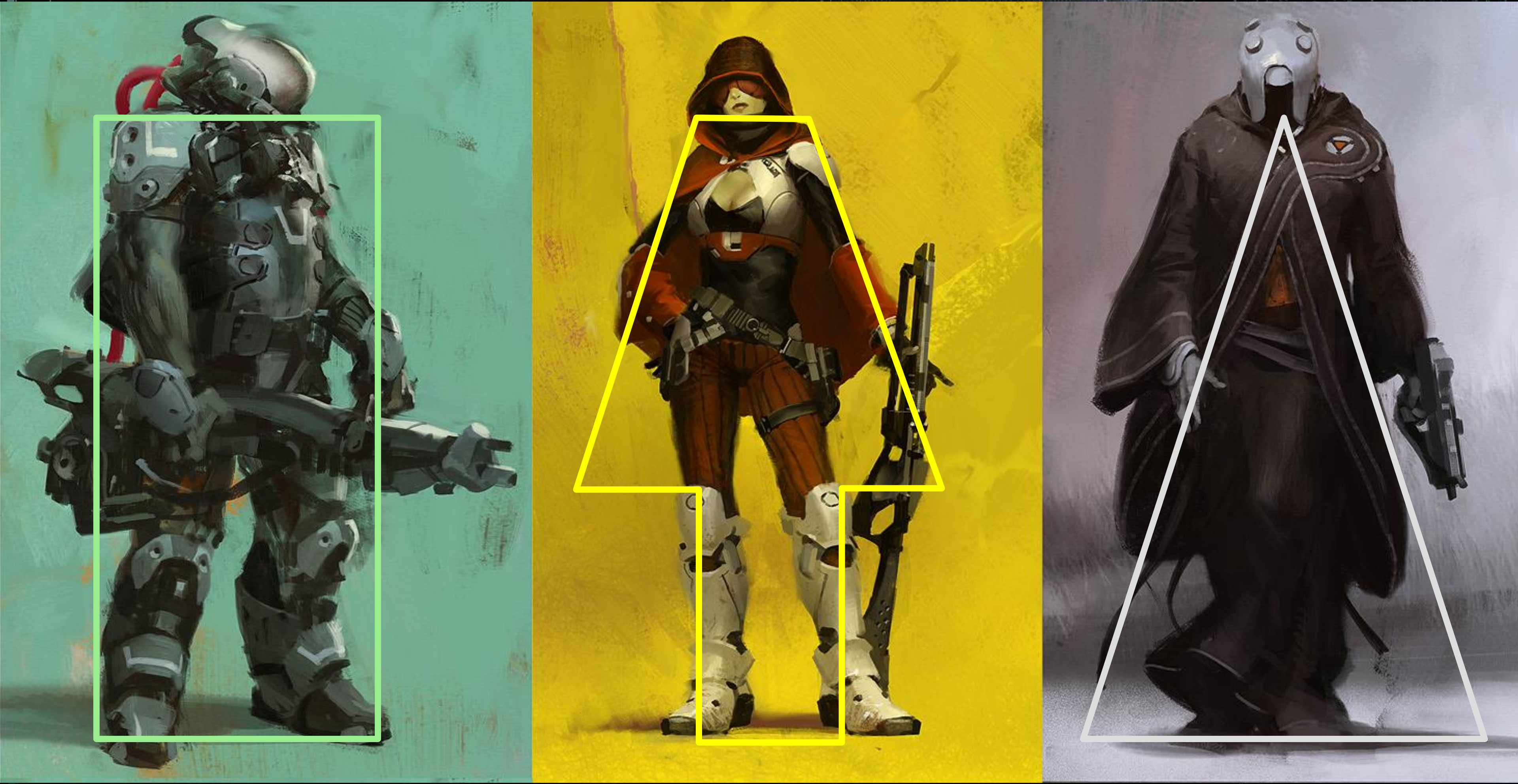
Hunter

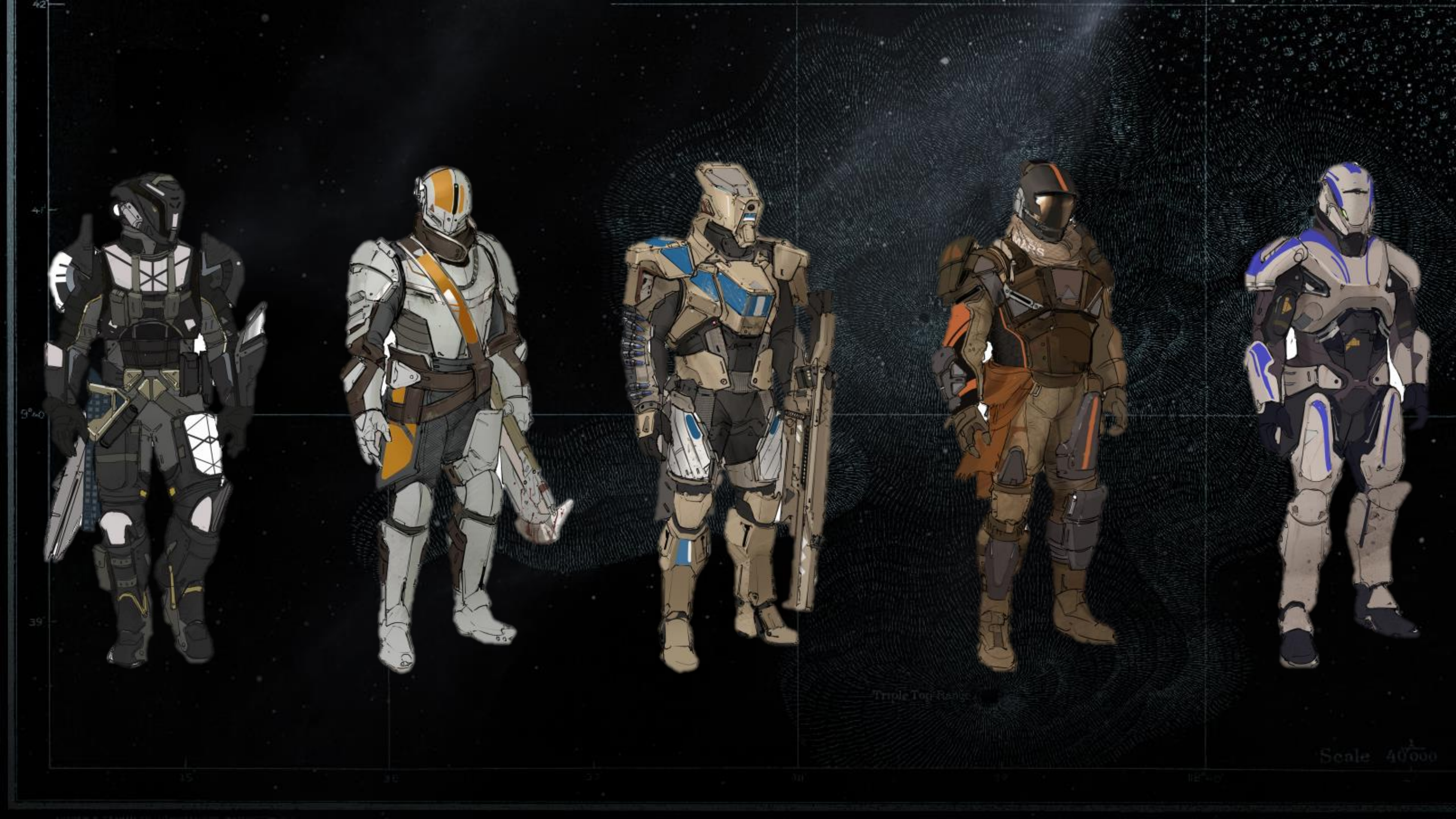
Bounty Hunter

Warlock

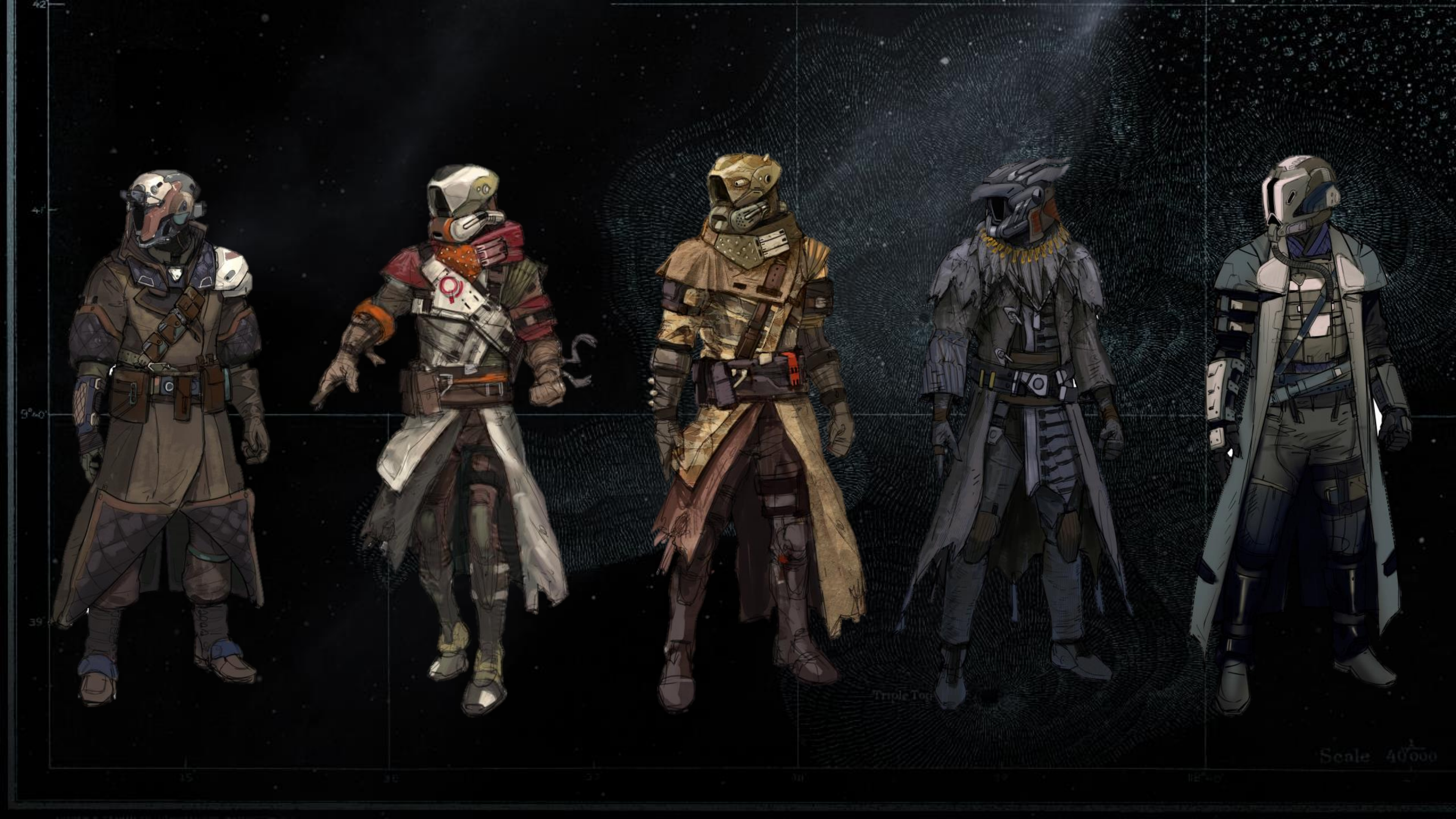
Space Wizard






MARS
Triple Top Range
Scale 1/40000

42
41
39
Triple Top Range
Scale 1/40000
15
36
38


42
41°
9°40'
39°
15
Triple Top
Scale $\frac{1}{40000}$

Weapons

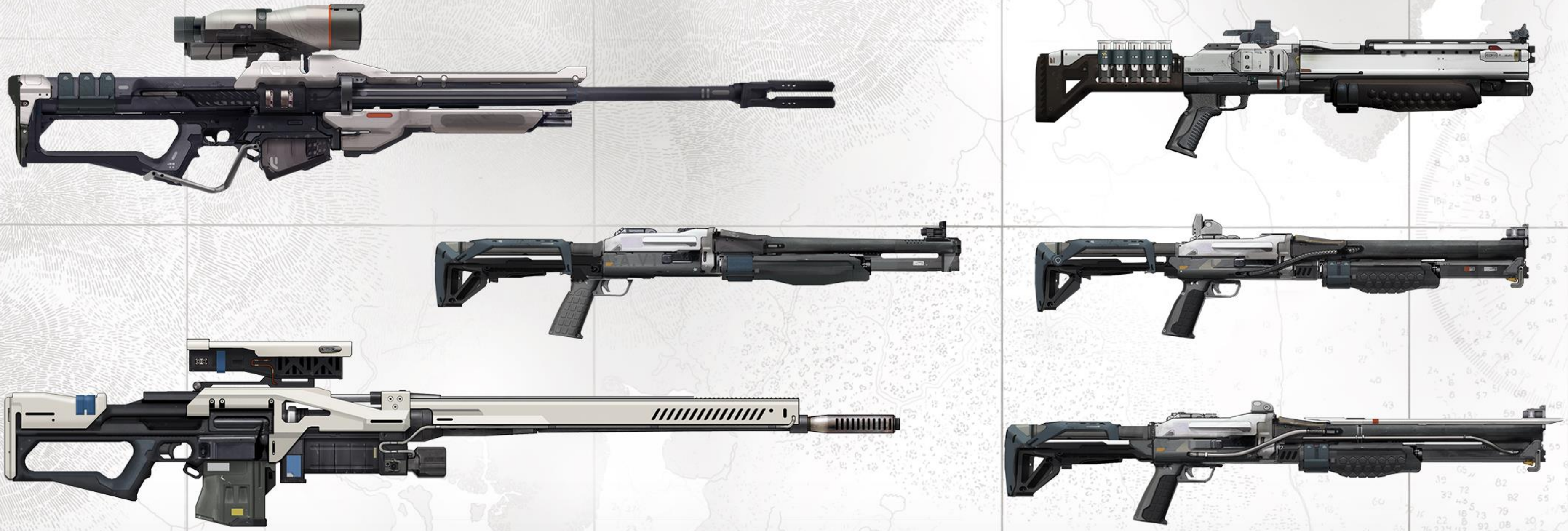
Titan

Hunter

Warlock

Village Range
IFWC
Triple Top Range
Scale 1/40000

FWC
Village Range
Triple Top Range
Scale 40000



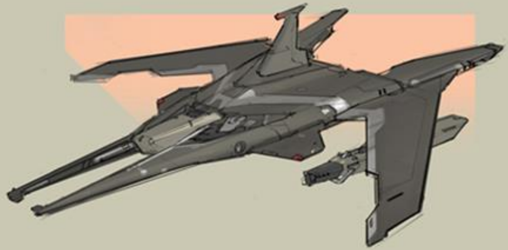
7R

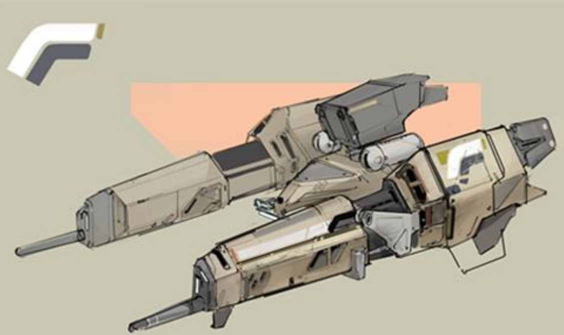
42
41
5°40'
39
77
Scale 1/40000

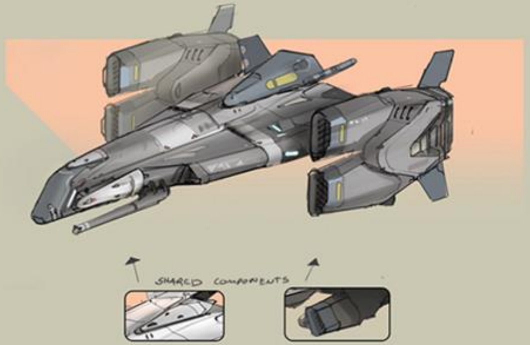
Player Ships

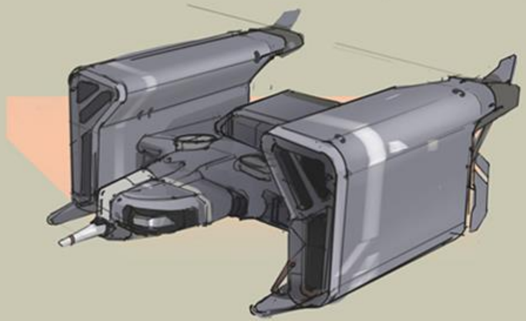
Gulf
11
Gulf

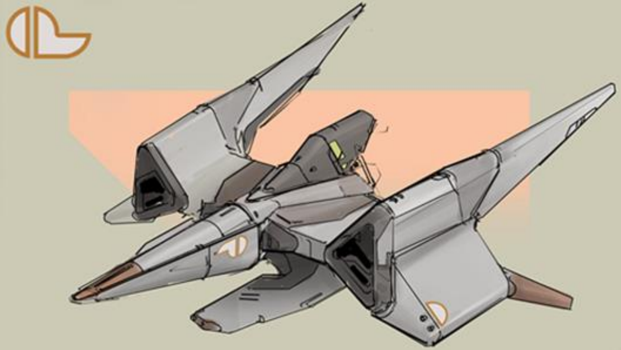
19
19





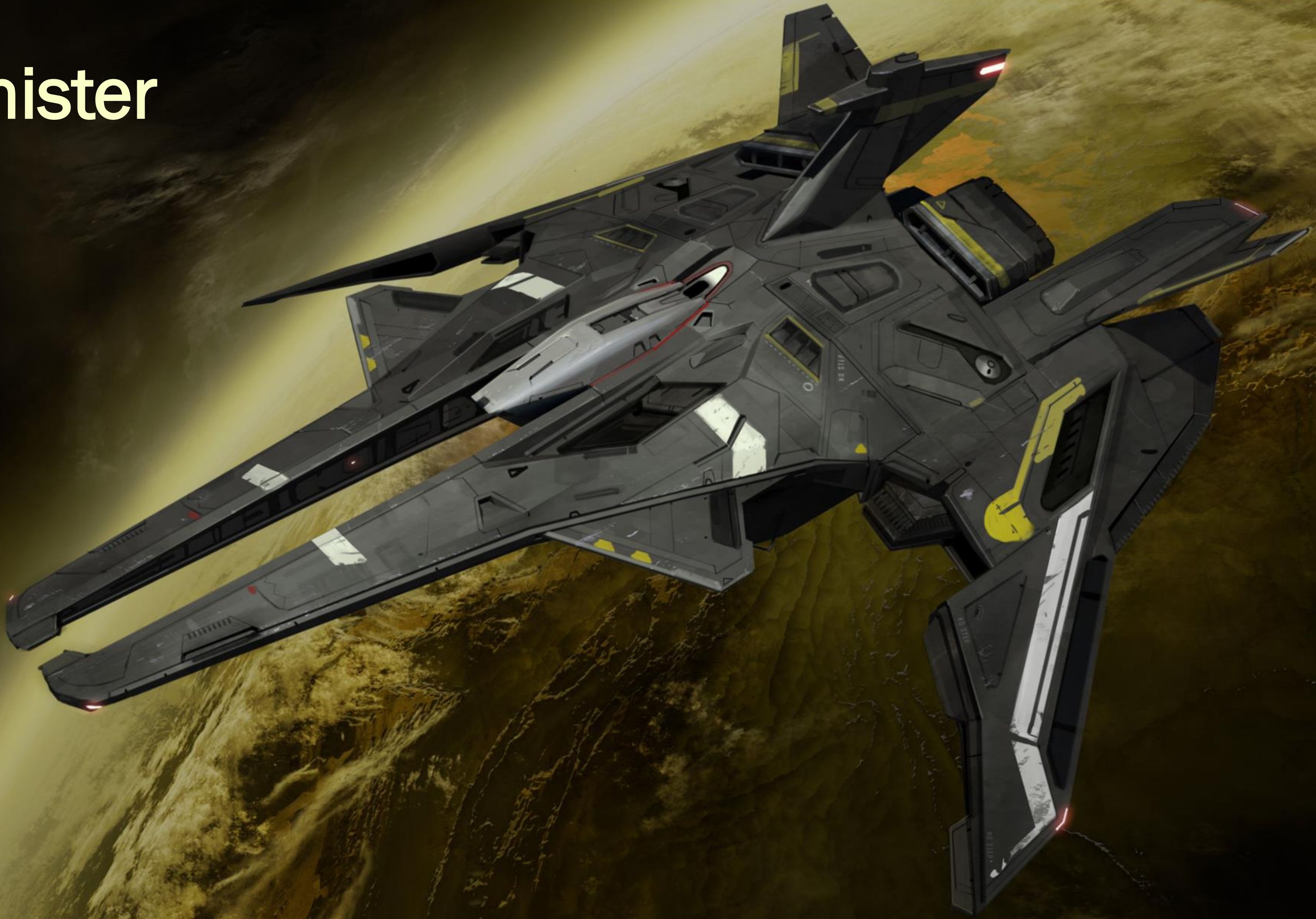

SHARED COMPONENTS



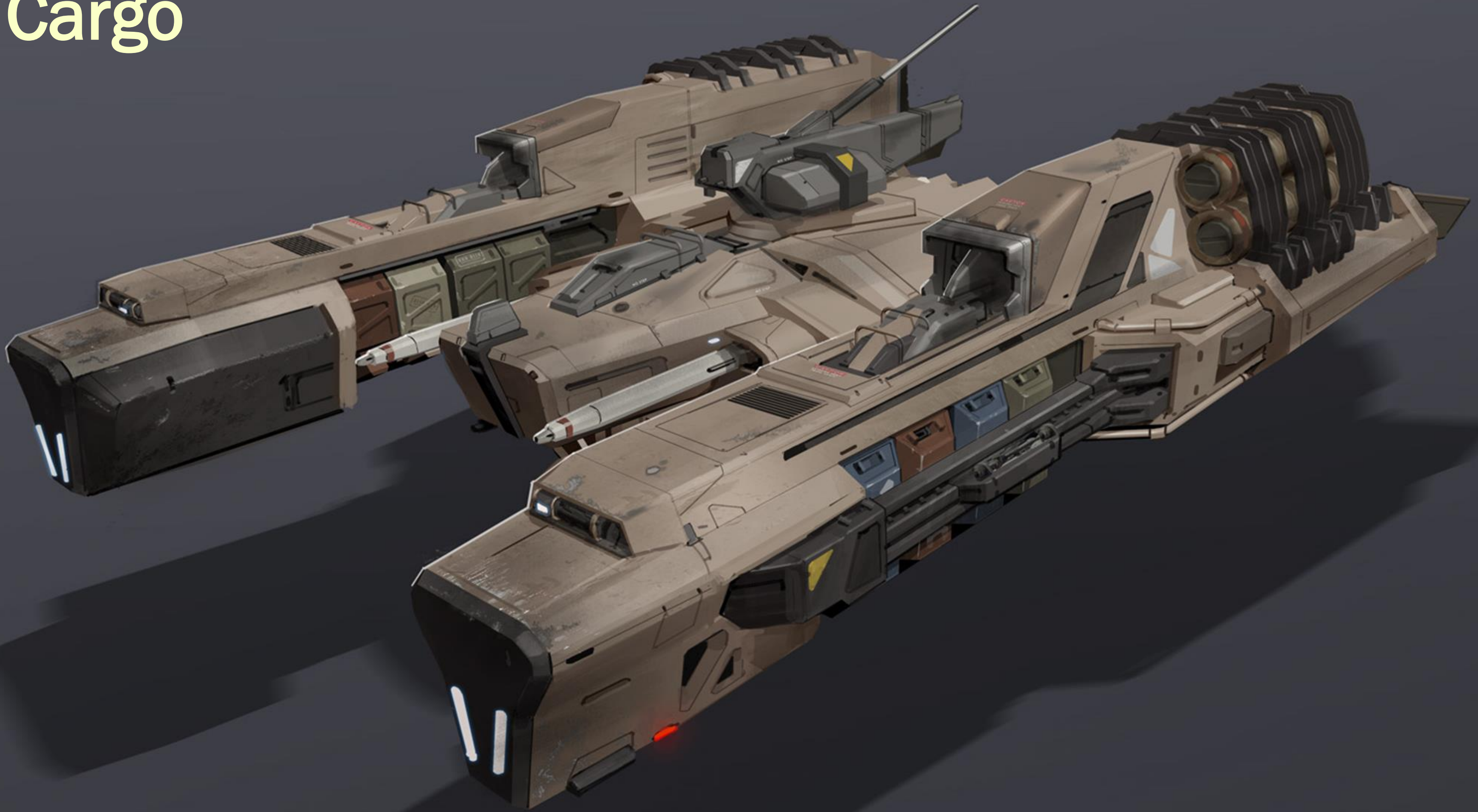


Racer

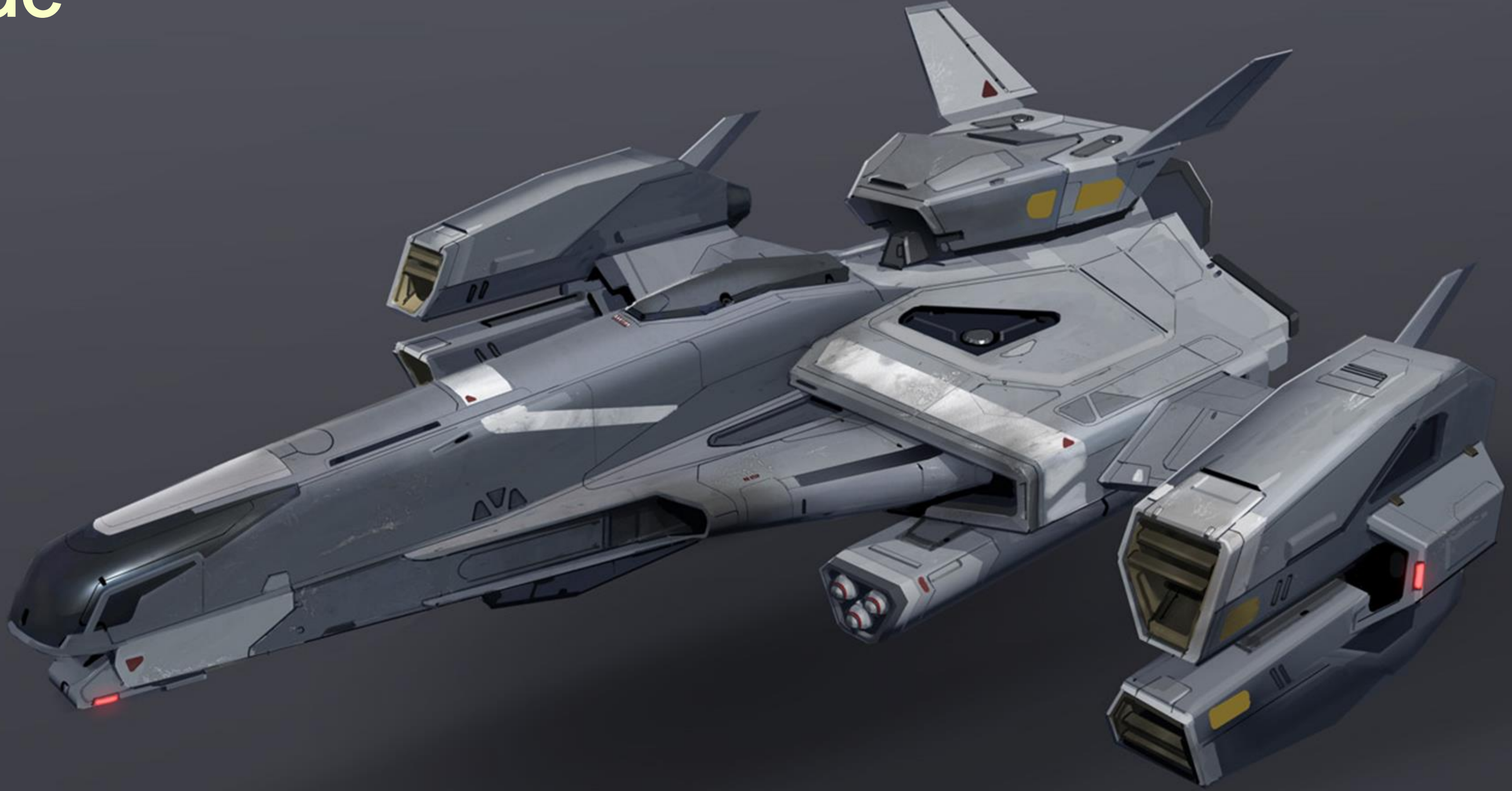
Sinister



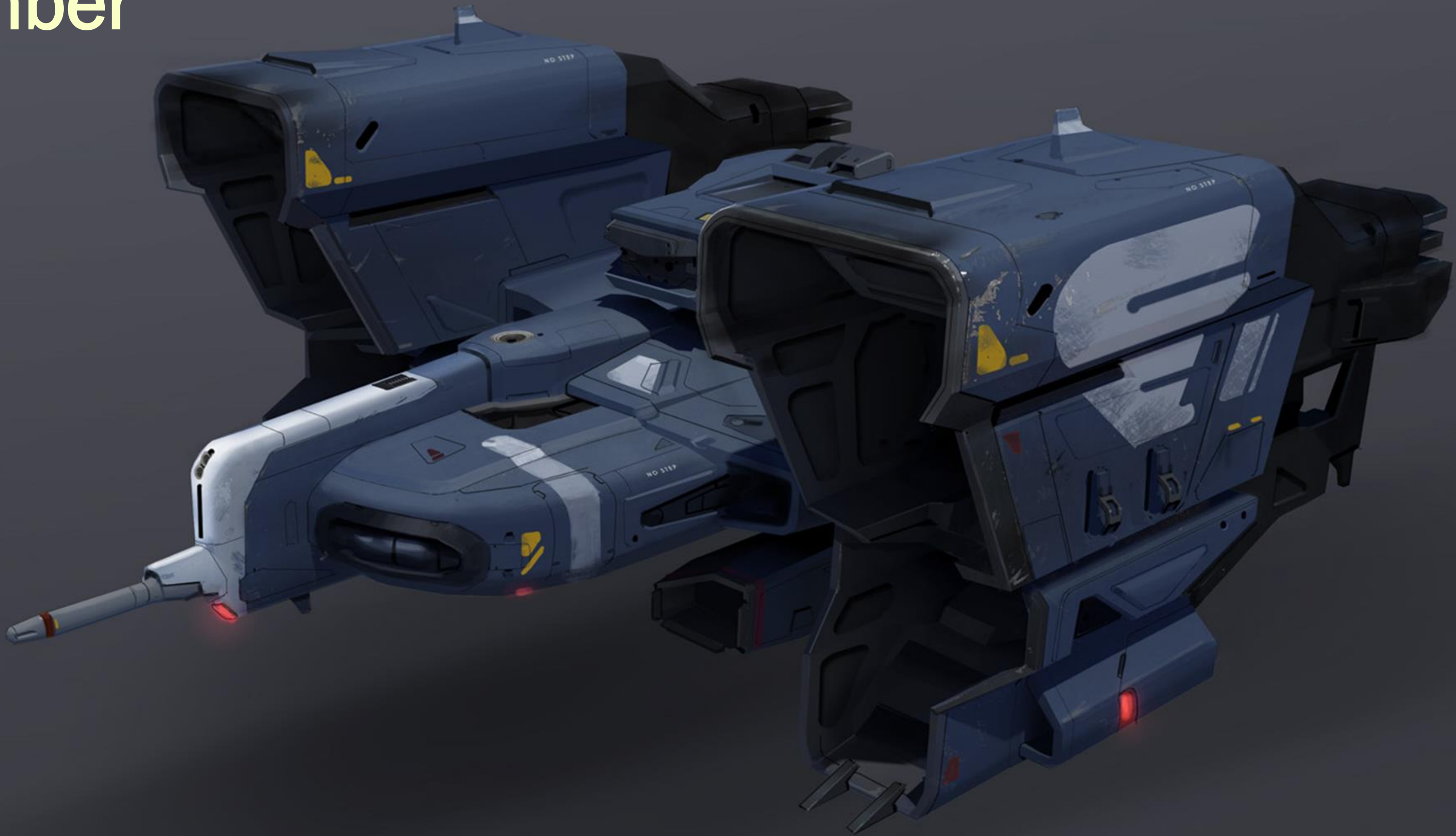
Cargo



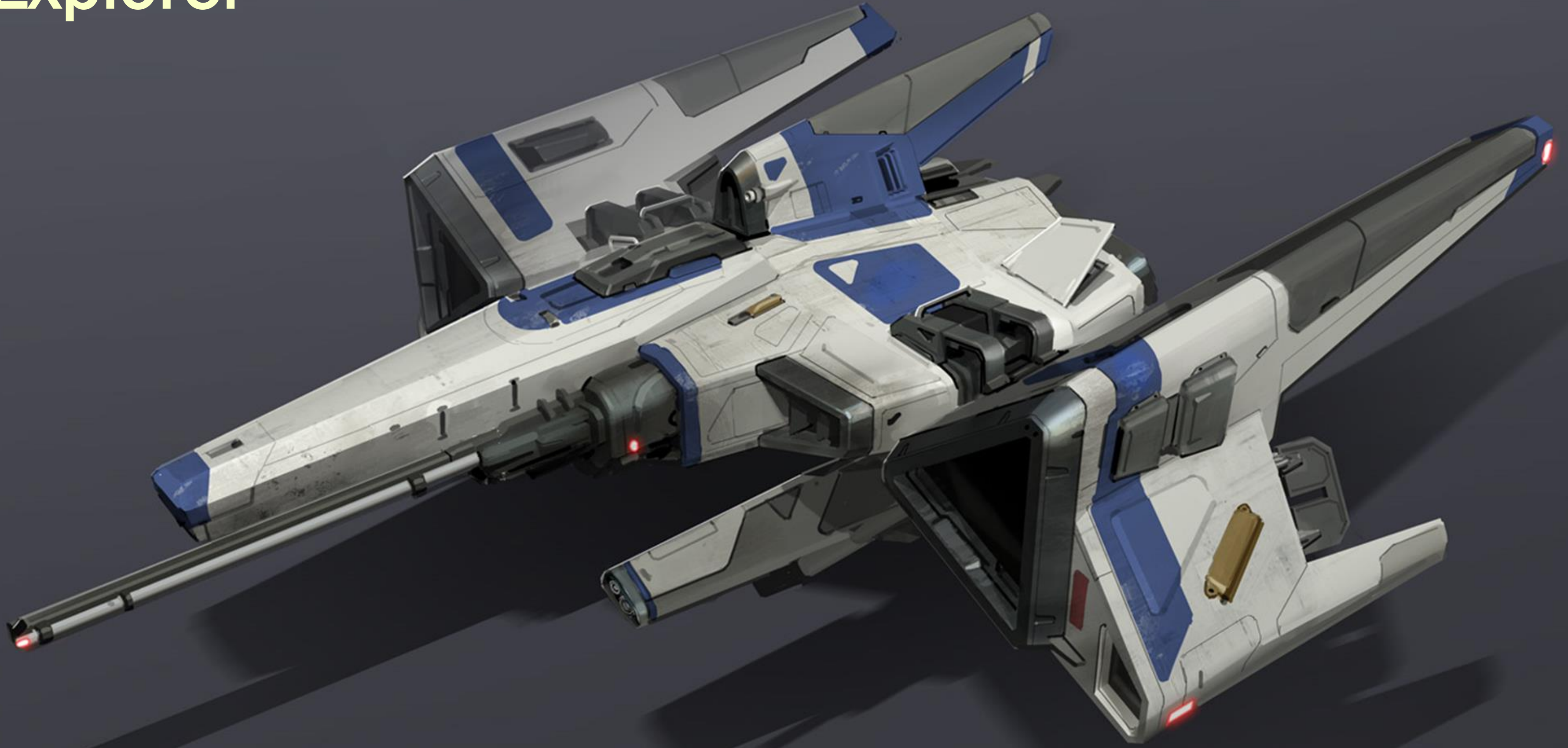
Rogue



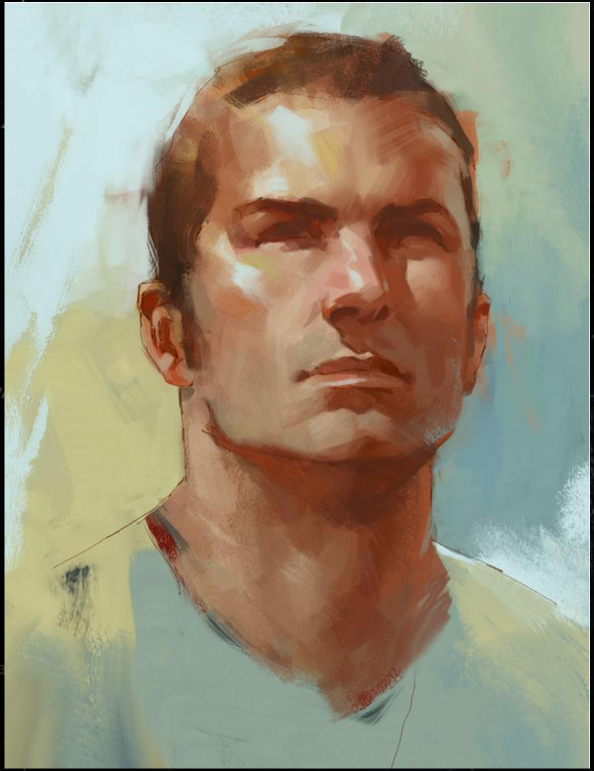
Bomber

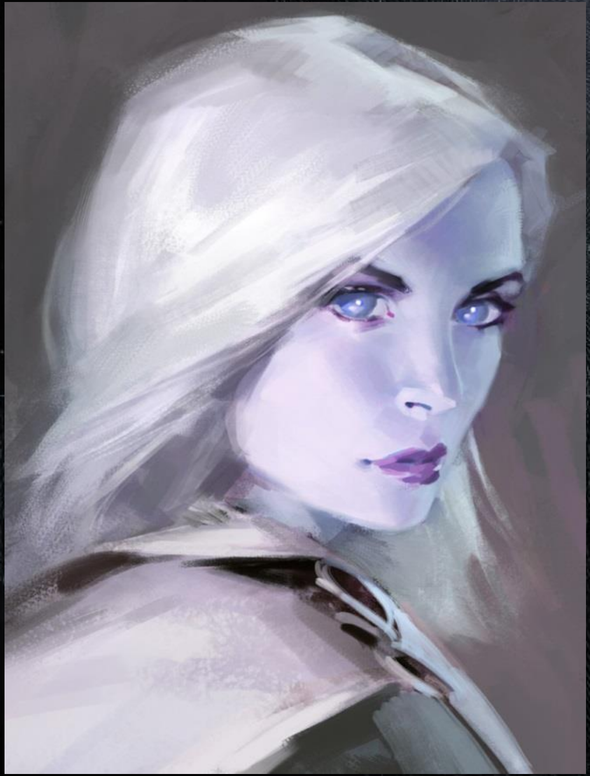


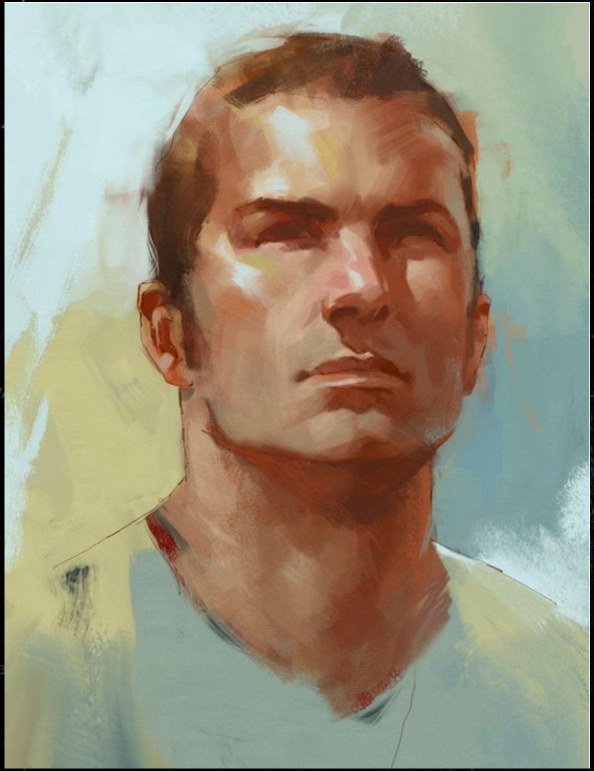
Explorer



Races





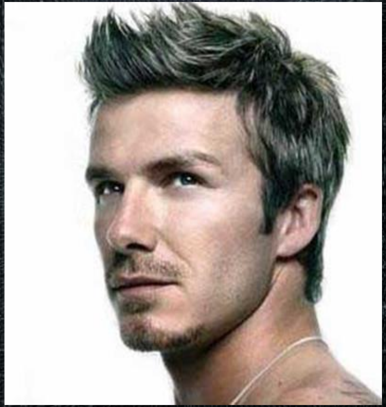


Human

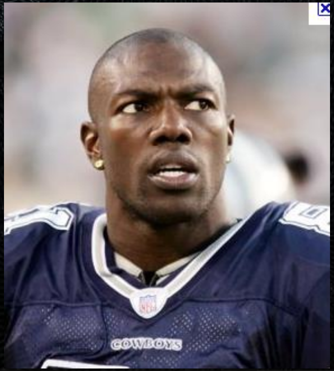
Relatable

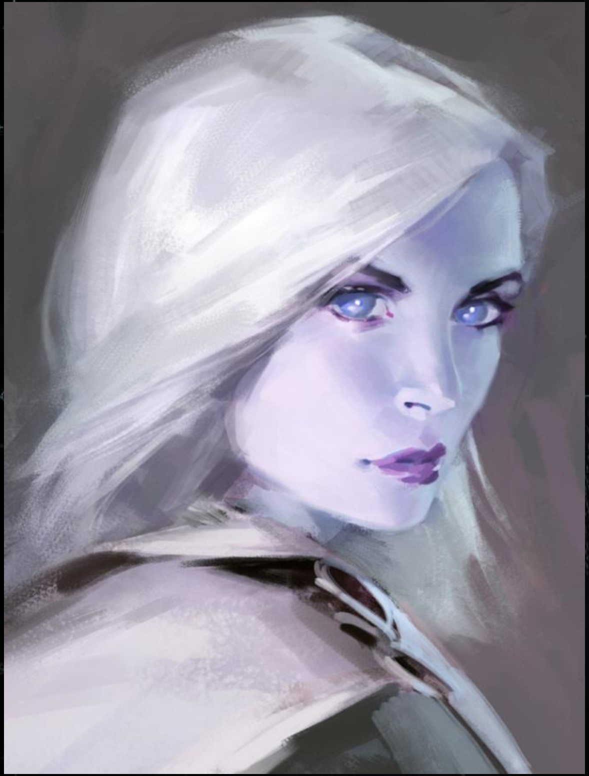
Tough

Uncomplicated



ABC/REISIG & TAYLOR





Awoken

Exotic

Beautiful

Mysterious

Exo

Powerful

Tireless

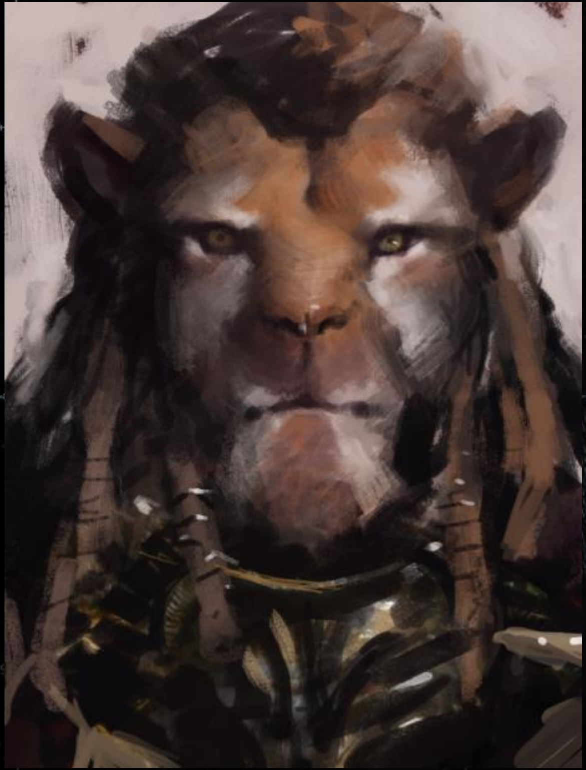
Sinister

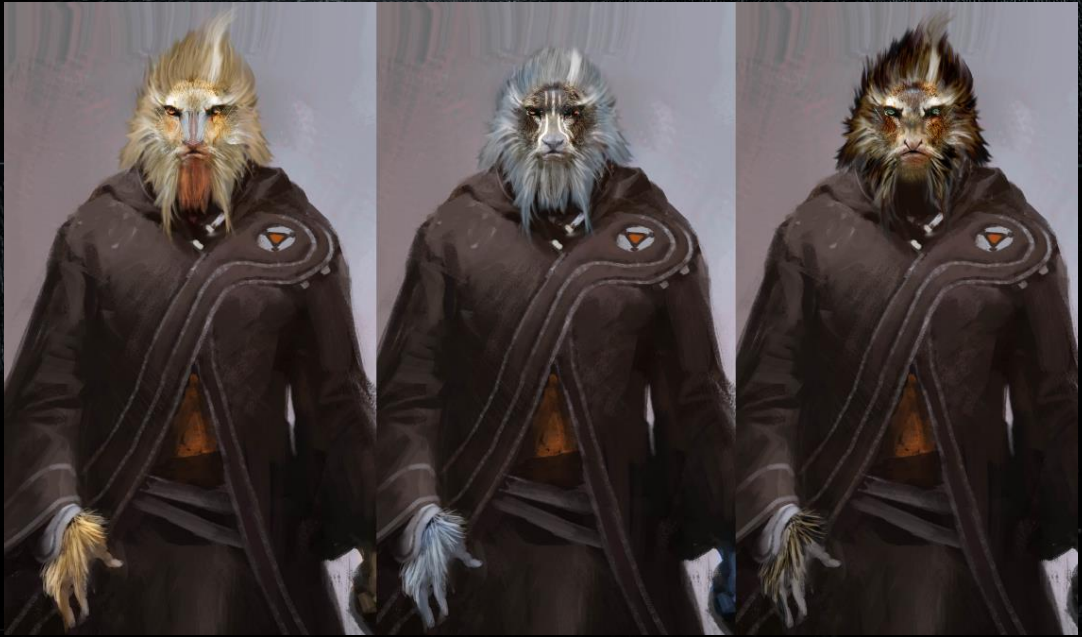
Tiger Man

Furry

Flexible

Somber





42
41'
9°40'
39'
15
20
16°40'
Scale 1/40000

Content Problem

- Need new content solutions, old methods do not scale
- New IP requires vast amount of compelling content in multiple areas of the player
- Production cycle within 2 years, and about 17artists
- Scalable between last gen and next gen

Tech Goals

- Shared experience: support many simultaneous unique avatars

Tech Goals

- Shared experience: support many simultaneous unique avatars
- Predictable memory footprint

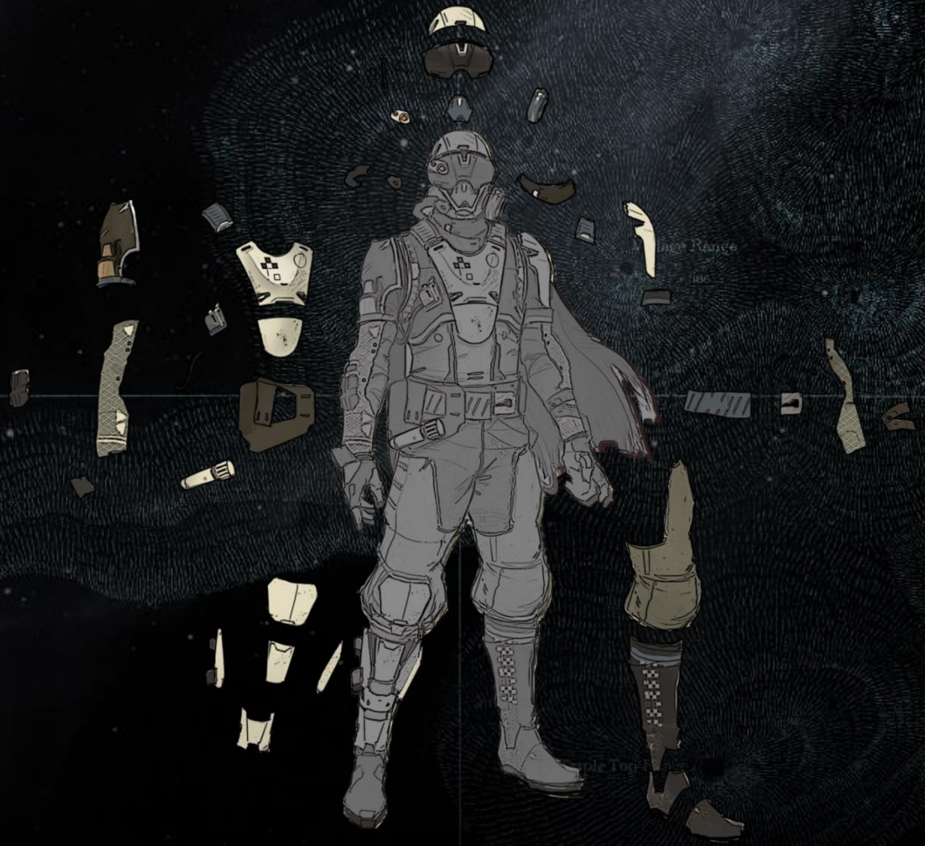
Tech Goals

- Shared experience: support many simultaneous unique avatars
- Predictable memory footprint
- Powerful and extensible pipeline

Tech Goals

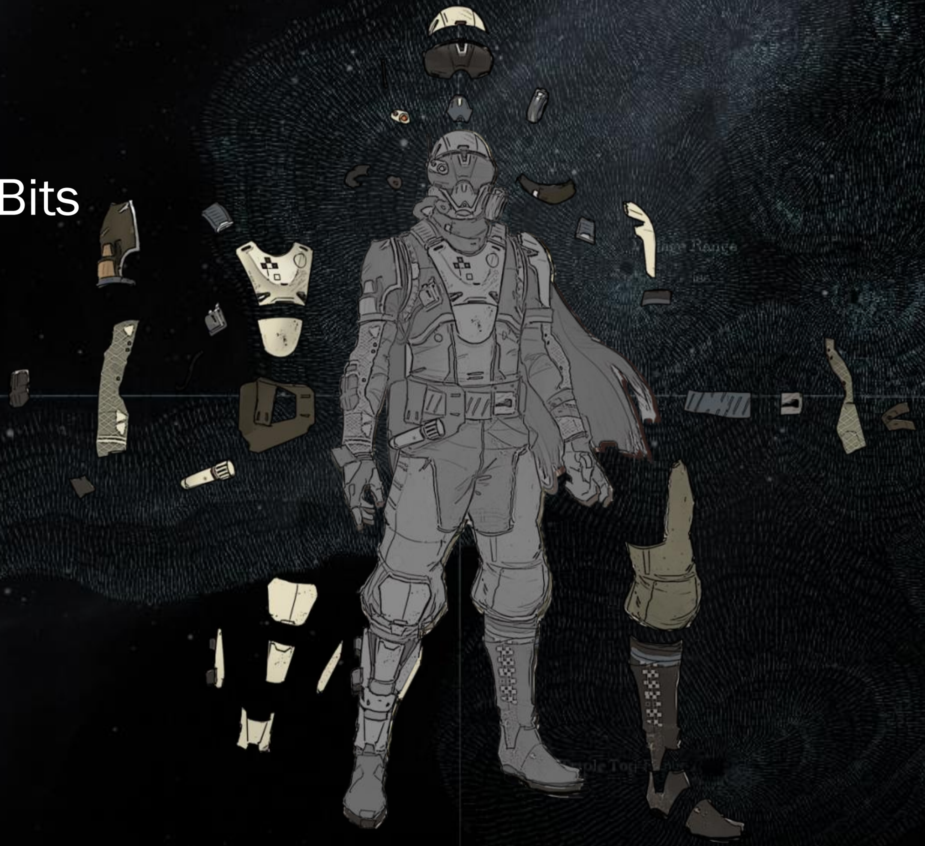
- Shared experience: support many simultaneous unique avatars
- Predictable memory footprint
- Powerful and extensible pipeline
- Reuse to minimize disc / download size

Our Solution: Gear System



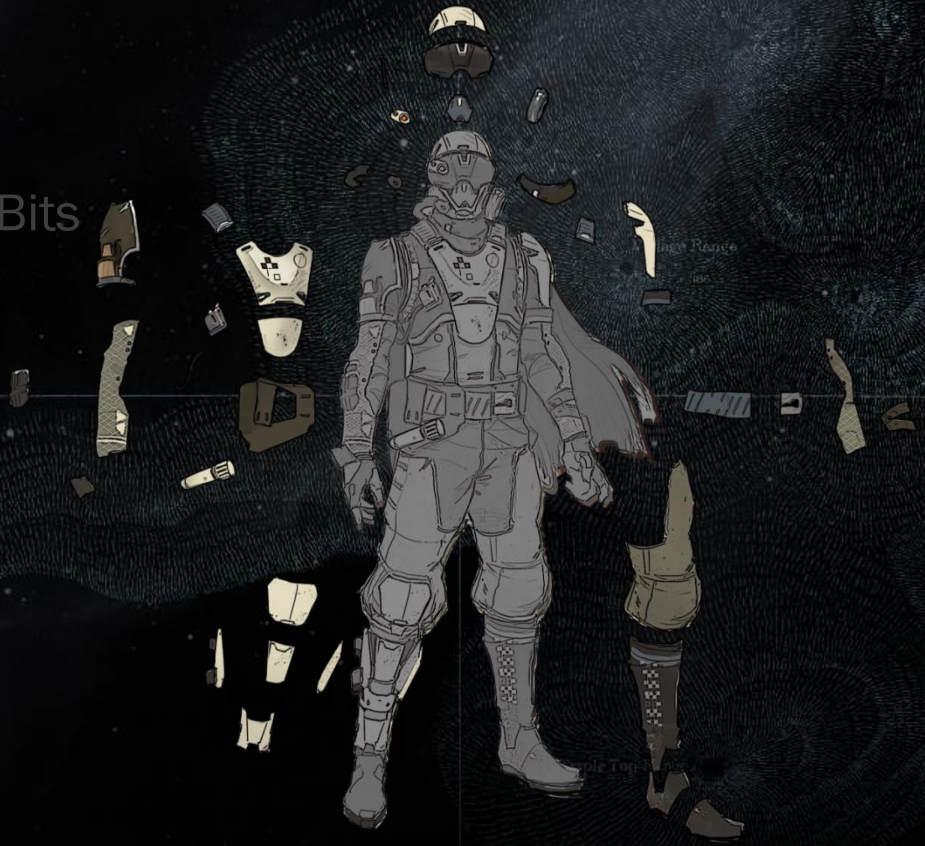
Our Solution: Gear System

1. Gear Slots
2. Arrangements and Bits
3. Plated Textures
4. Gear Dyes

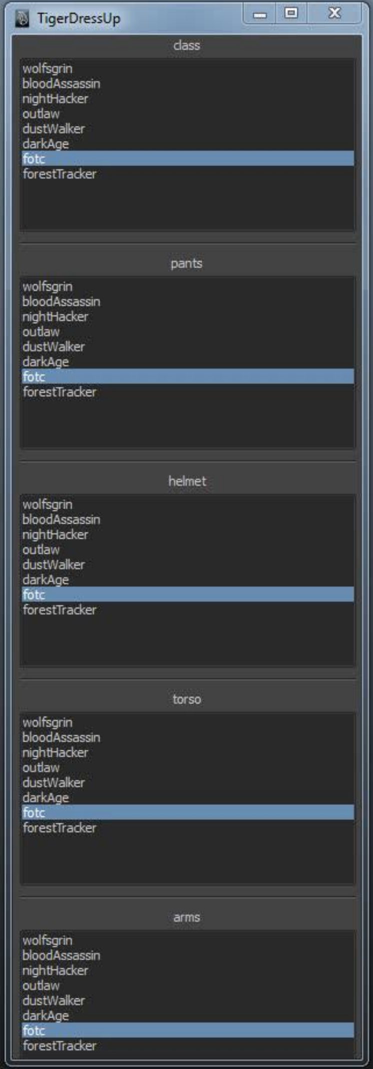


Our Solution: Gear System

1. Gear Slots
2. Arrangements and Bits
3. Plated Textures
4. Gear Dyes



Gear Slot System



TigerDressUp

class

wolfsgrin
bloodAssassin
nightHacker
outlaw
dustWalker
darkAge
[✓ fotc]
forestTracker

pants

wolfsgrin
bloodAssassin
nightHacker
outlaw
dustWalker
darkAge
[✓ fotc]
forestTracker

helmet

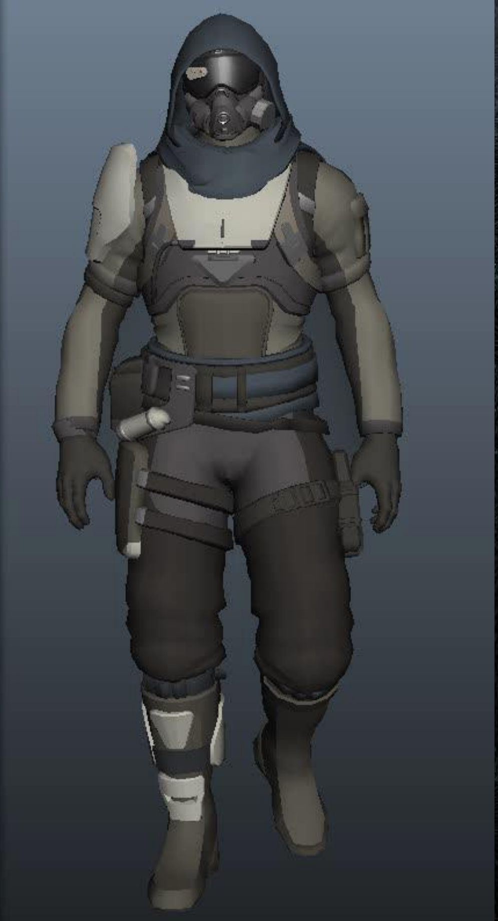
wolfsgrin
bloodAssassin
nightHacker
outlaw
dustWalker
darkAge
[✓ fotc]
forestTracker

torso

wolfsgrin
bloodAssassin
nightHacker
outlaw
dustWalker
darkAge
[✓ fotc]
forestTracker

arms

wolfsgrin
bloodAssassin
nightHacker
outlaw
dustWalker
darkAge
[✓ fotc]
forestTracker



Gear Slot

What it is:

- A basic gear component
- Content creation and management unit
- Runtime component unit

Gear Slot

What it is *not*:

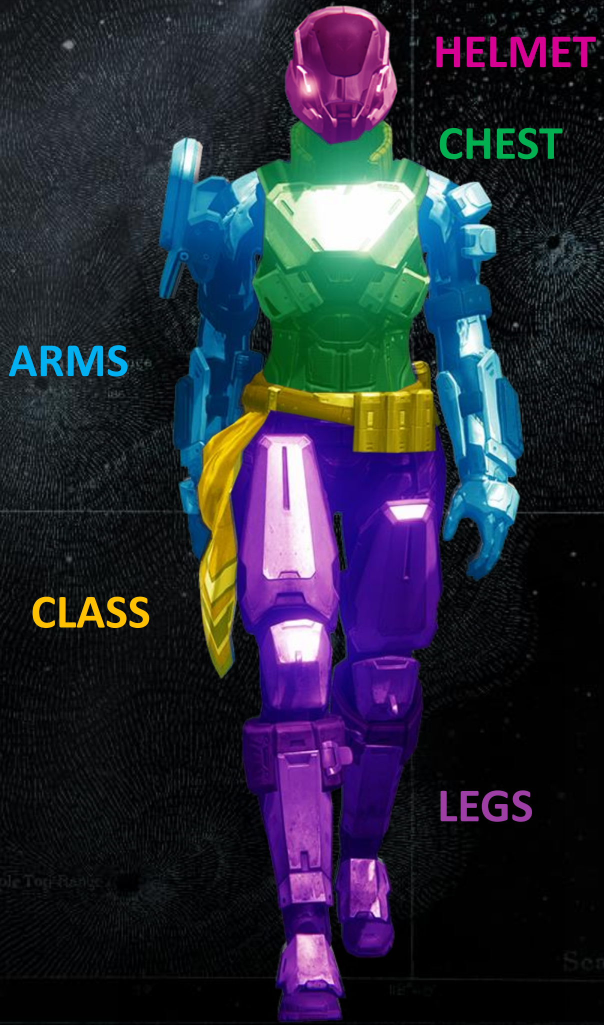
- Customization and investment unit

Gear Slot System

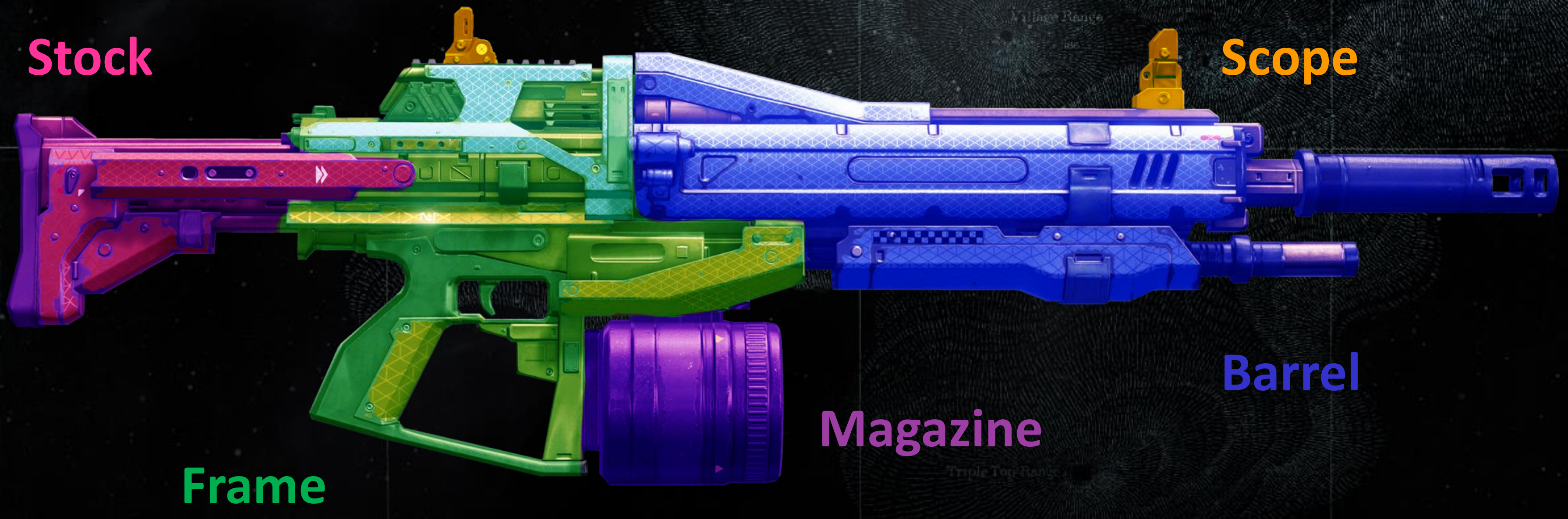
- Support multiple gear asset types
  - Player armor
  - Player weapon
  - Player ships
  - Player heads
- Extensible: each asset type defines a set of gear slots

Gear Slots : Player Armor

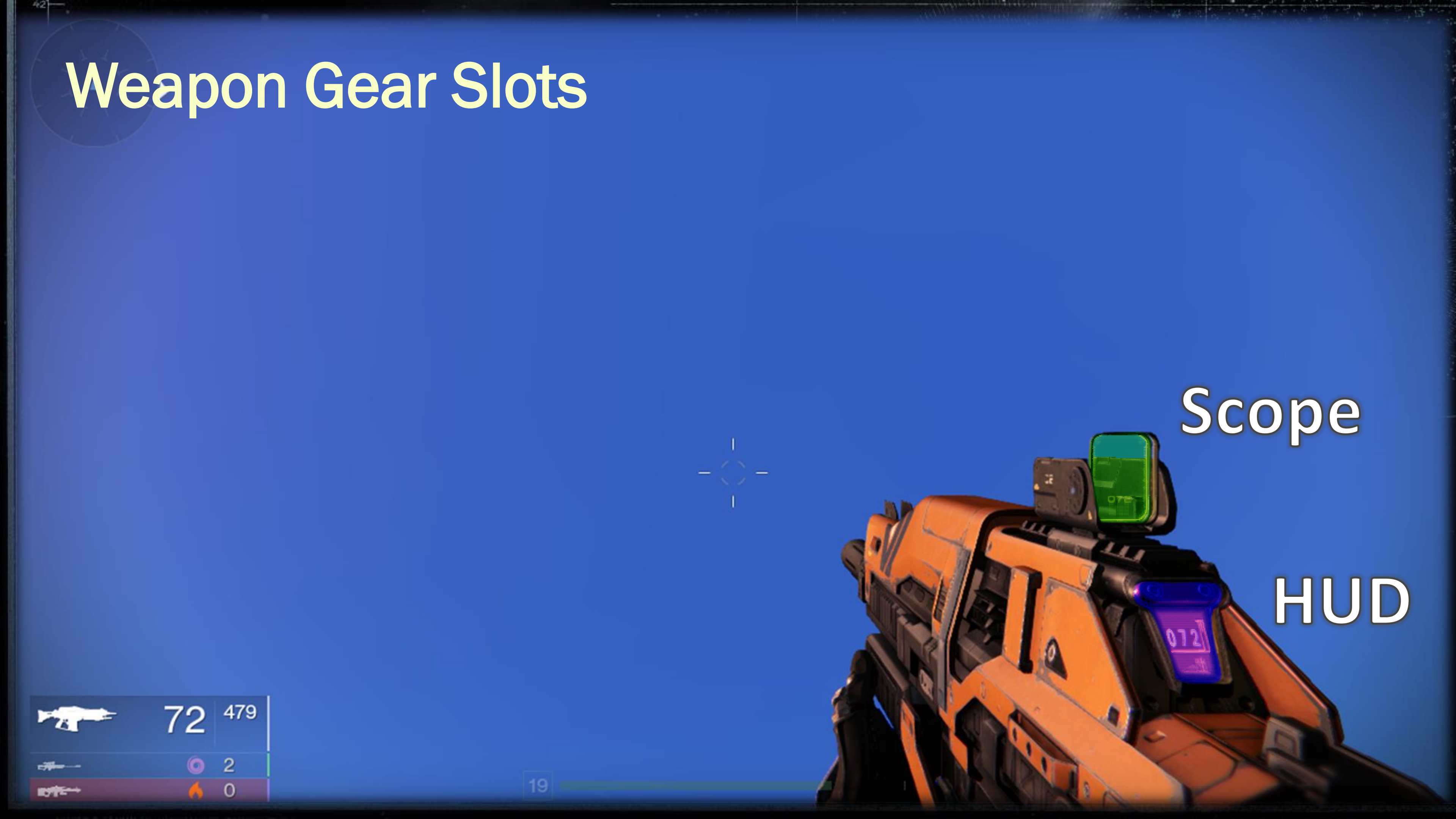
- Combat: 5 gear slots



Weapon Gear Slots

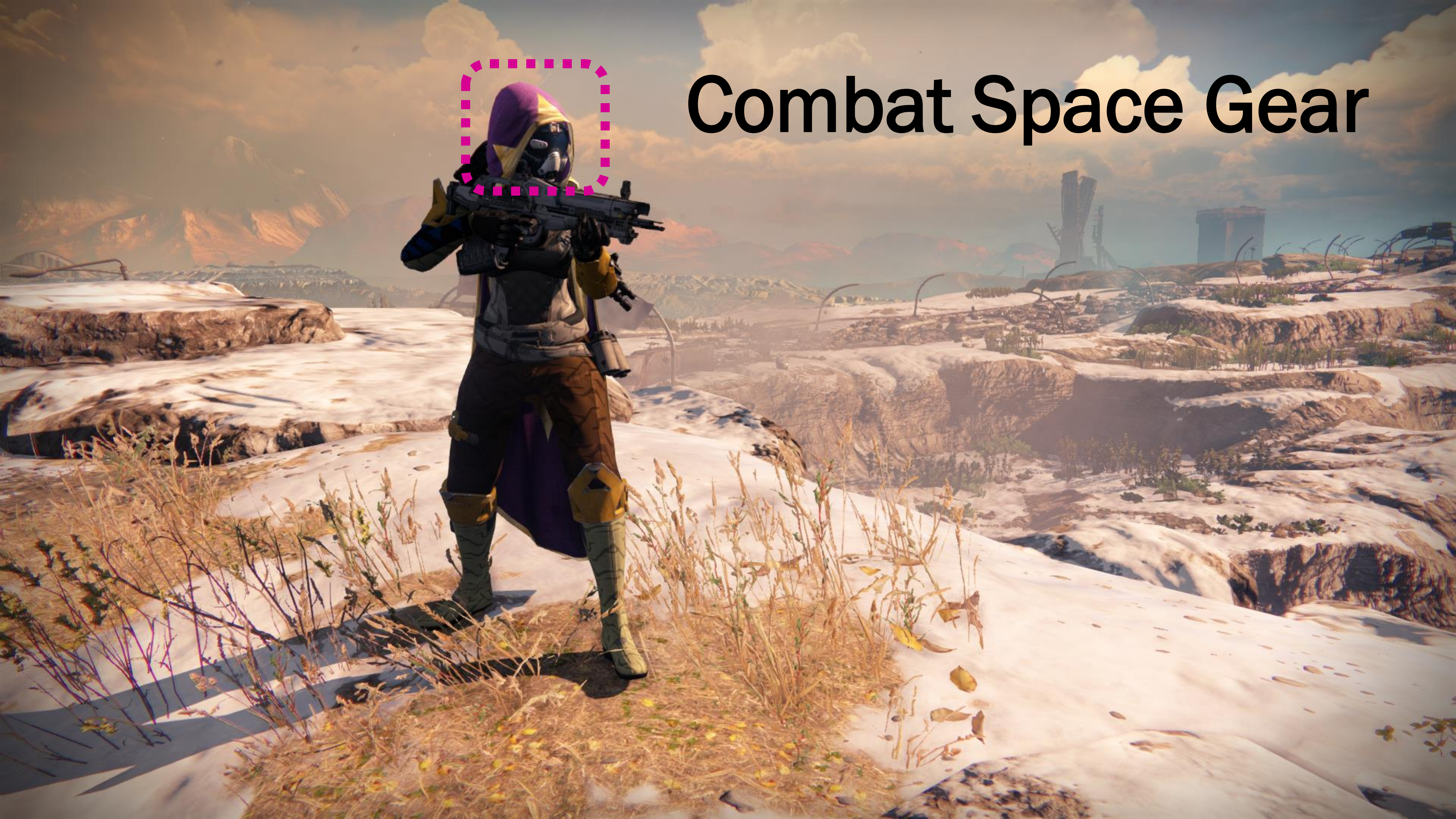


Weapon Gear Slots



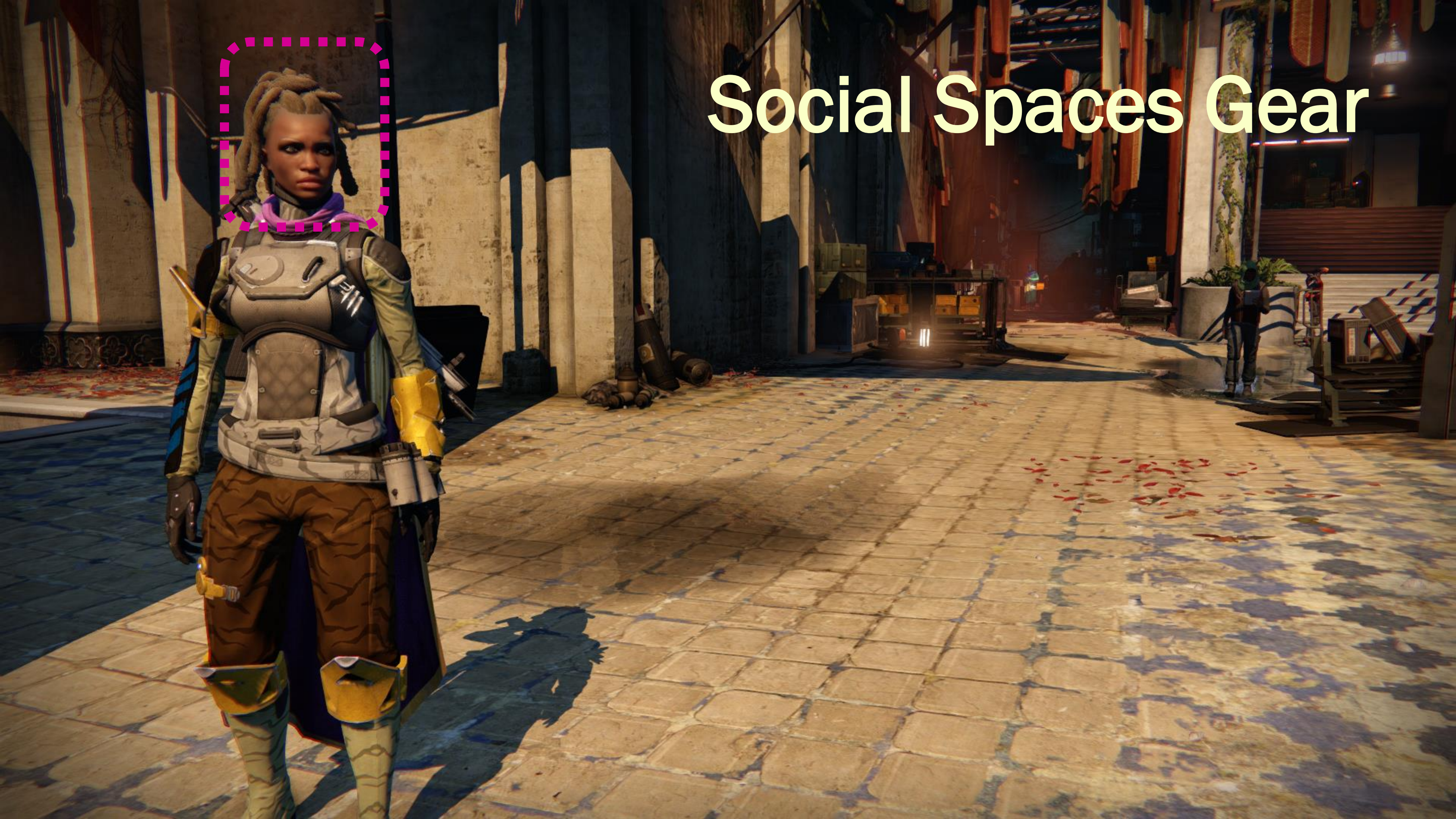
Situation-Dependent Gear Slots

- Enabled in different game scenarios

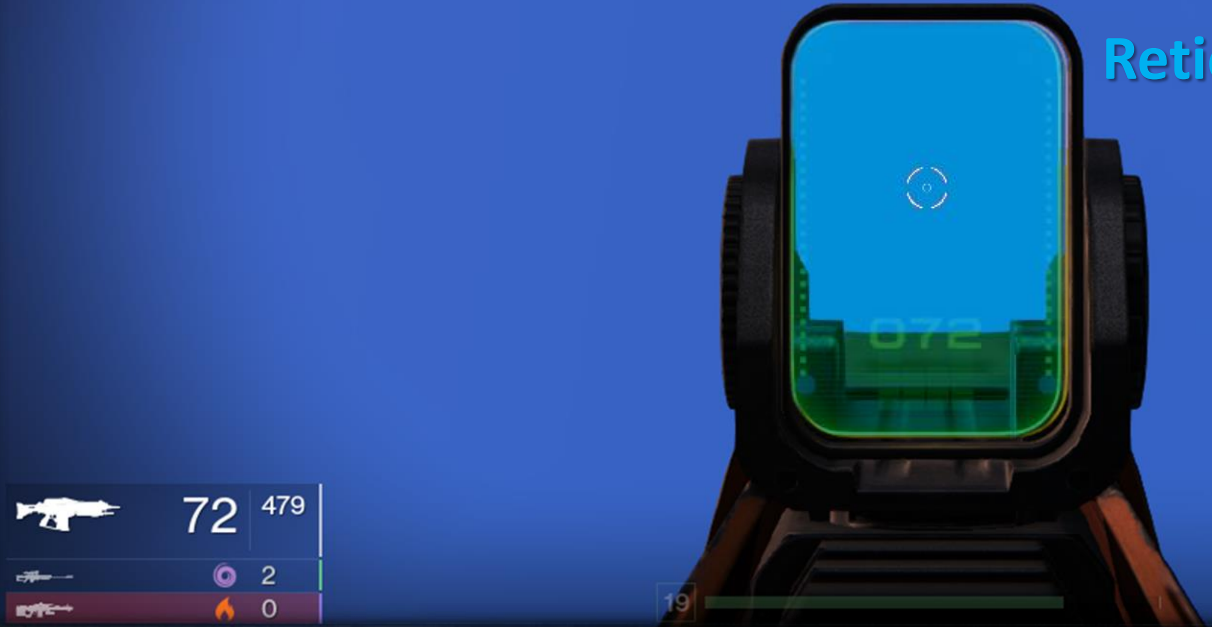


Combat Space Gear

Social Spaces Gear

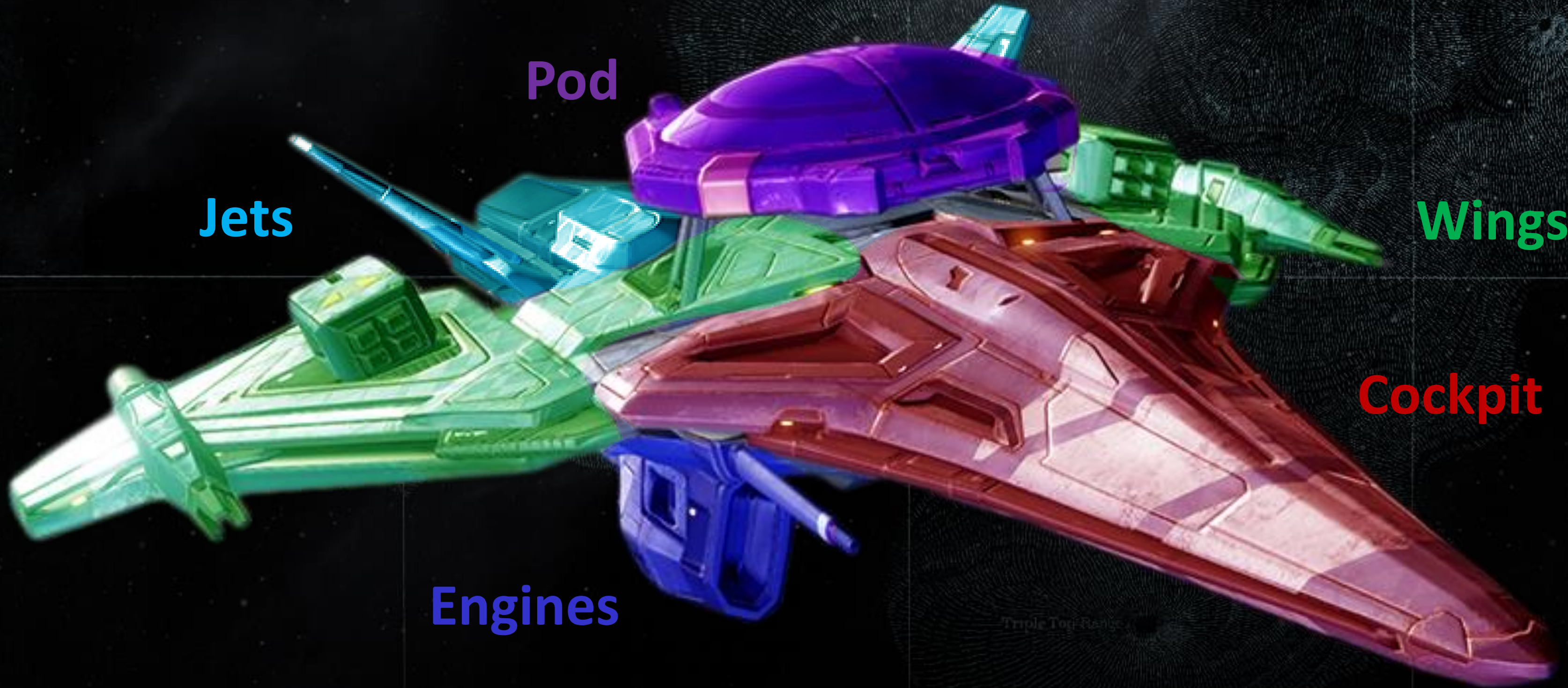


Weapon Gear Slots: ADS: Reticle

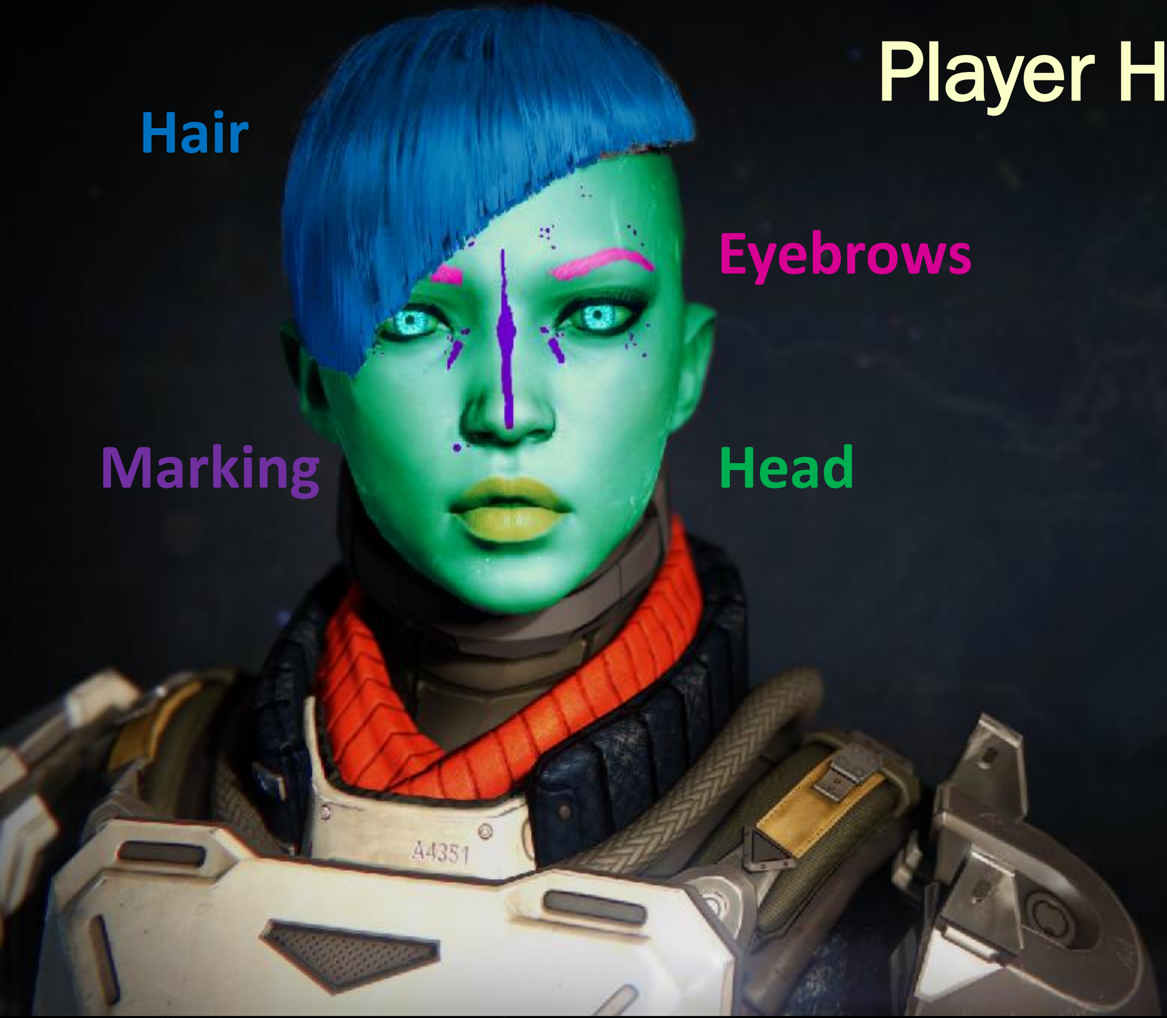


Reticle

Player Ship Gear Slots

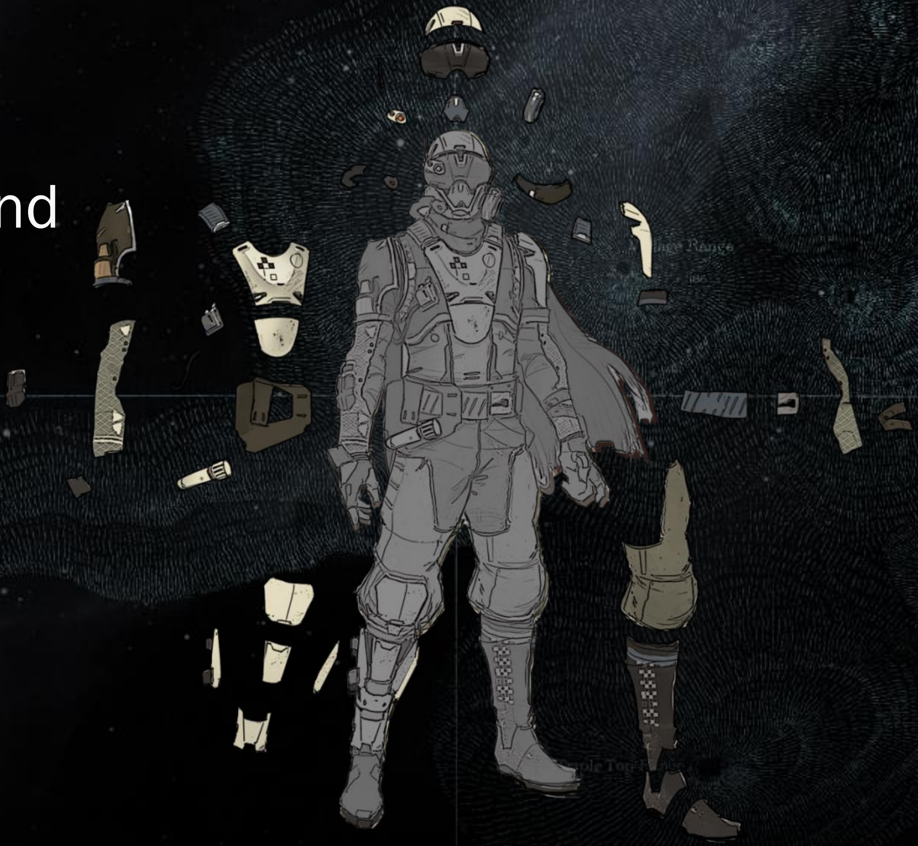


Player Head Gear Slots

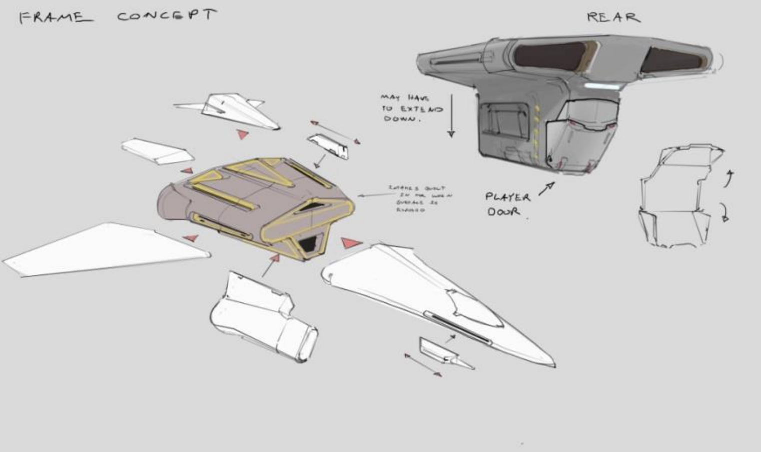


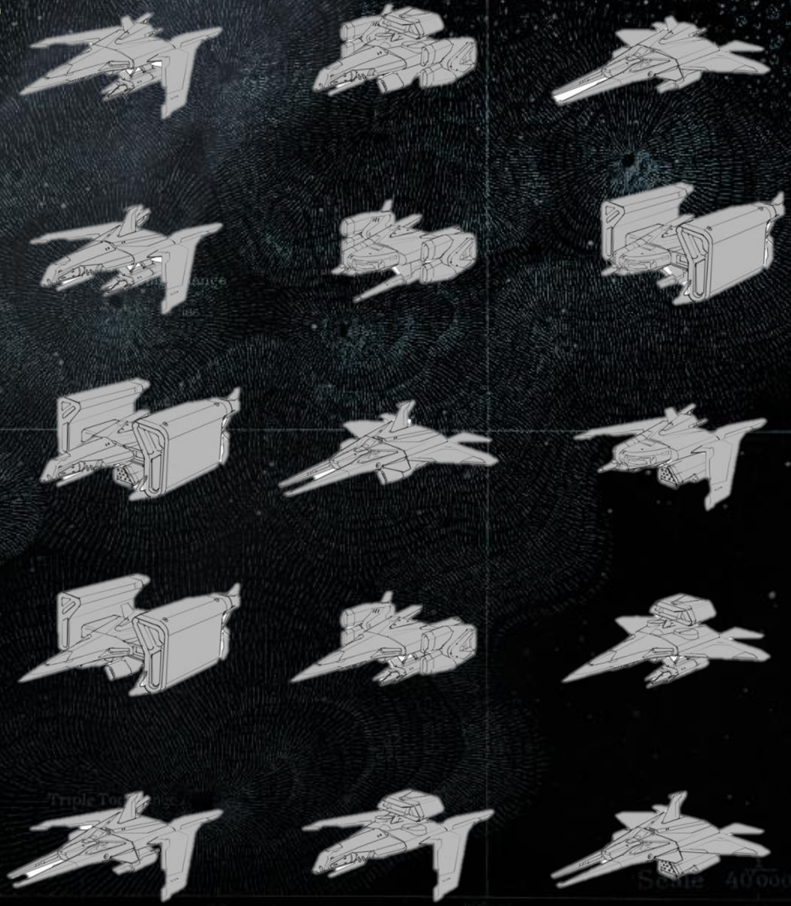
Our Solution: Gear System

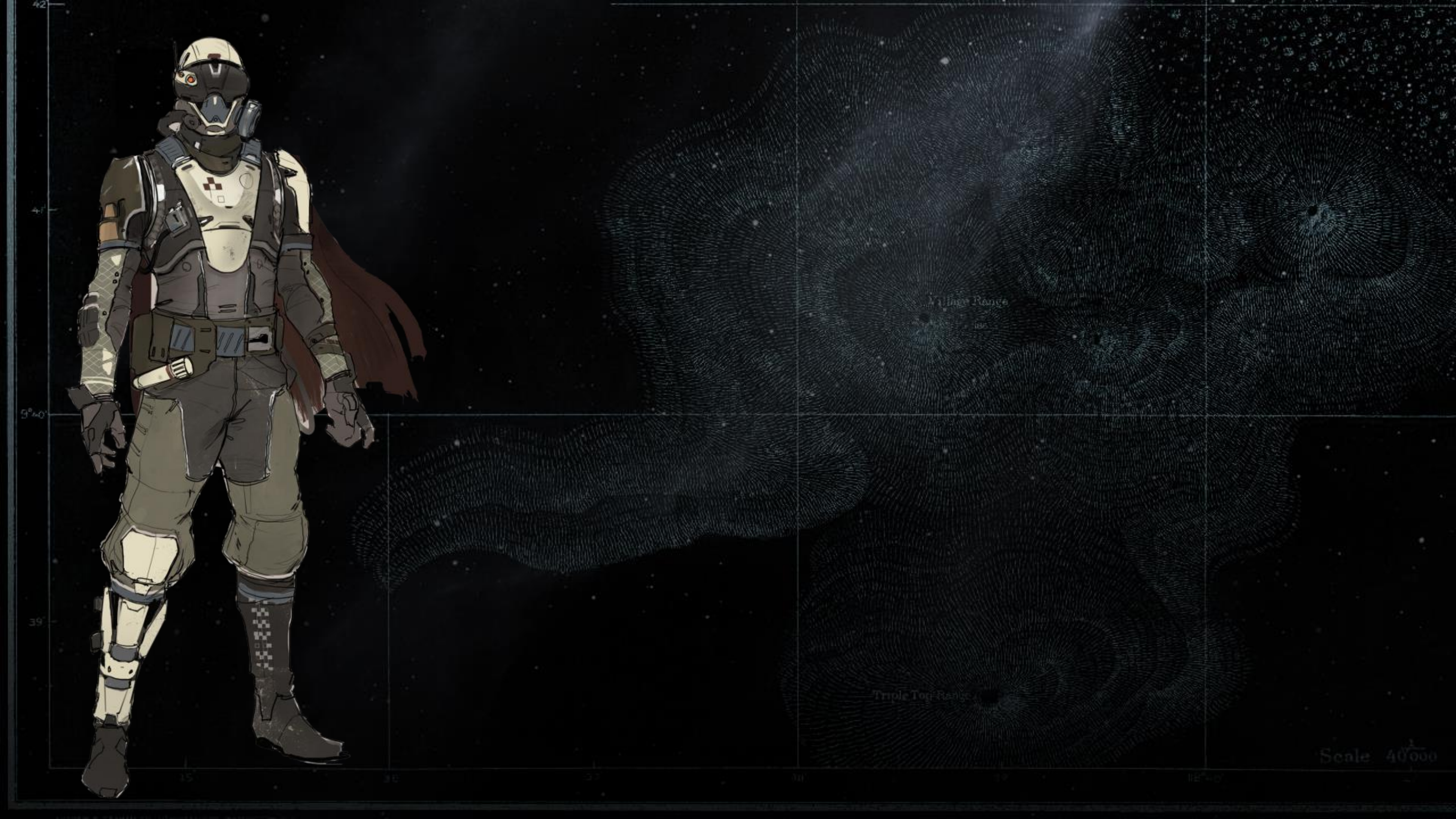
1. Gear Slots
2. Arrangements and Bits
3. Plated Textures
4. Gear Dyes

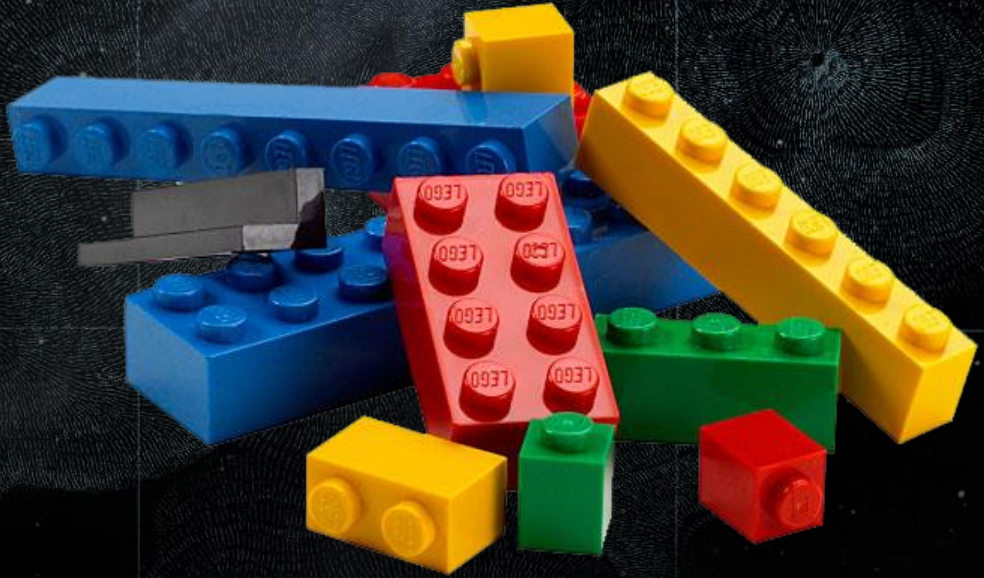


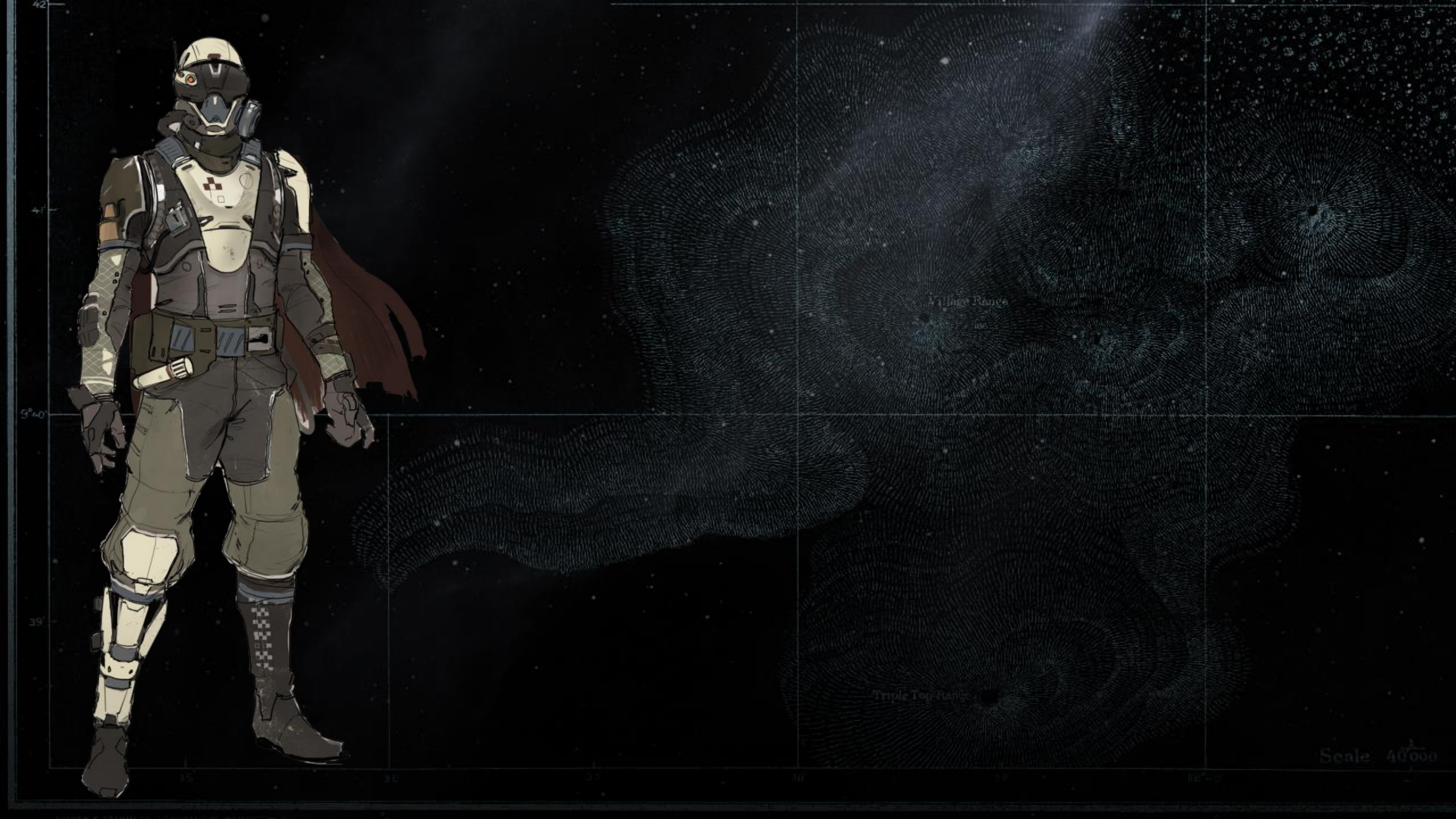
Gear Arrangements and Bits






42
41
5°40'
39
Village Range
Triple Top Range
Scale 1/40000

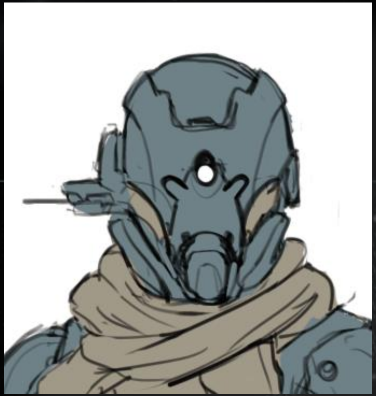



Village Range
Triple Top Range
Scale 1/40000

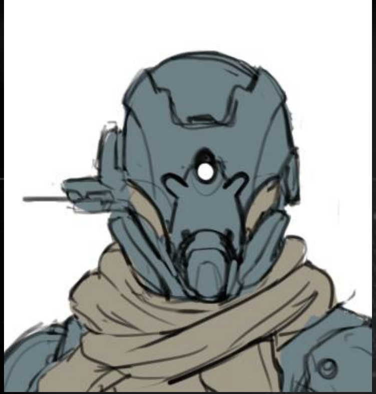
Village Range
Temple Top Range
Scale 1/40000

42
41
5°40'
39
Village Range
Table Top Range
Scale 1/40000

Gear Arrangement

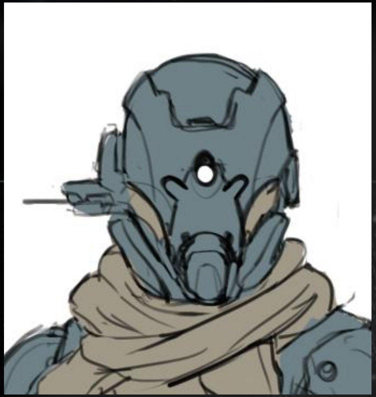


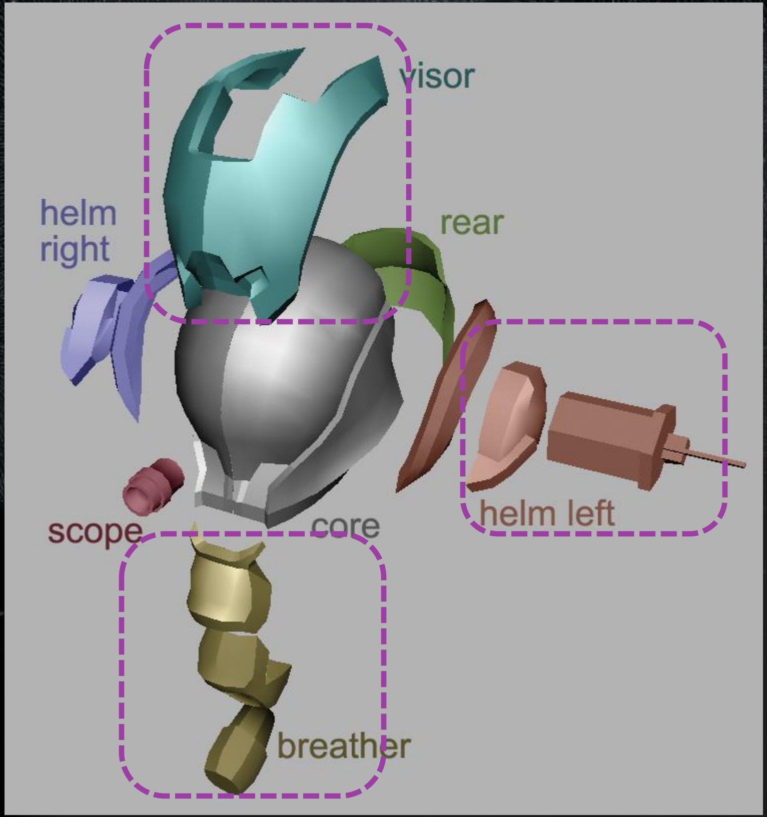
Gear Arrangement



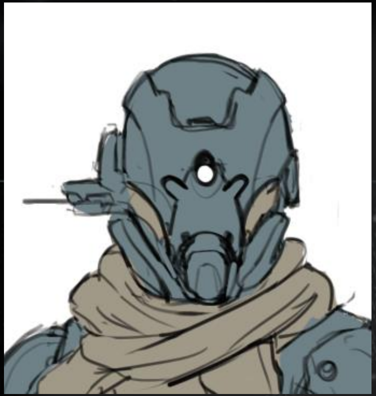
- One arrangement to one gear slot
- An asset container for that slot

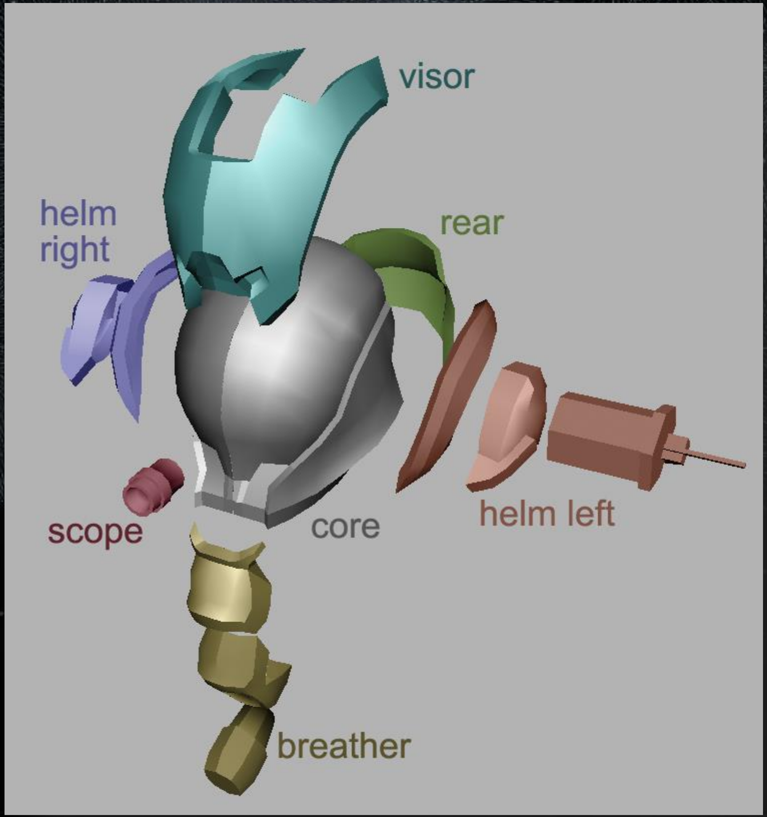
Gear Arrangement and Gear Bits



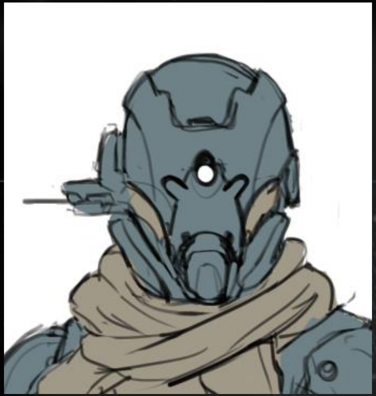


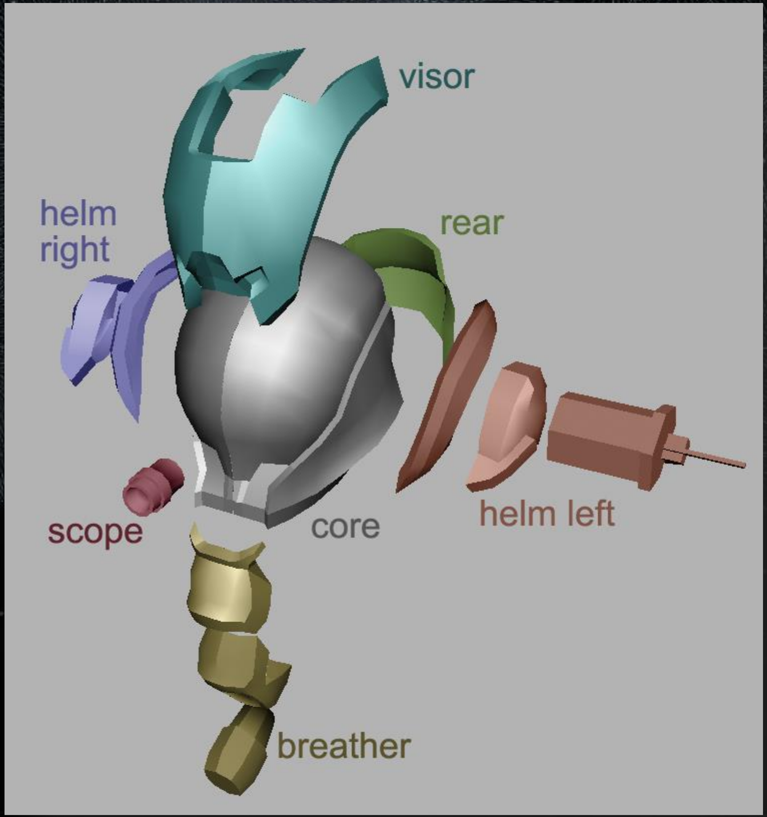
Gear Arrangement and Gear Bits

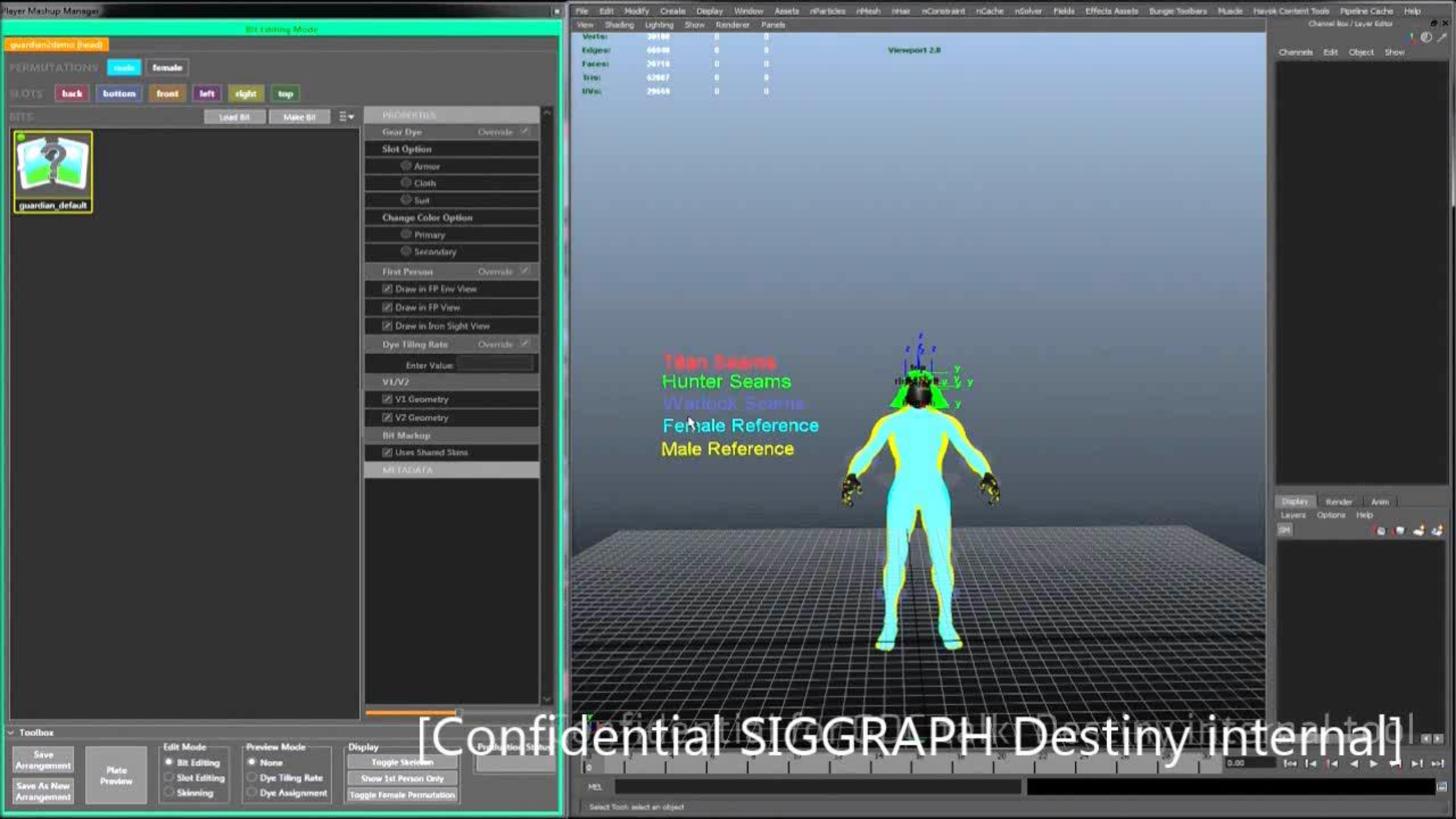




Gear Arrangement and Gear Bits







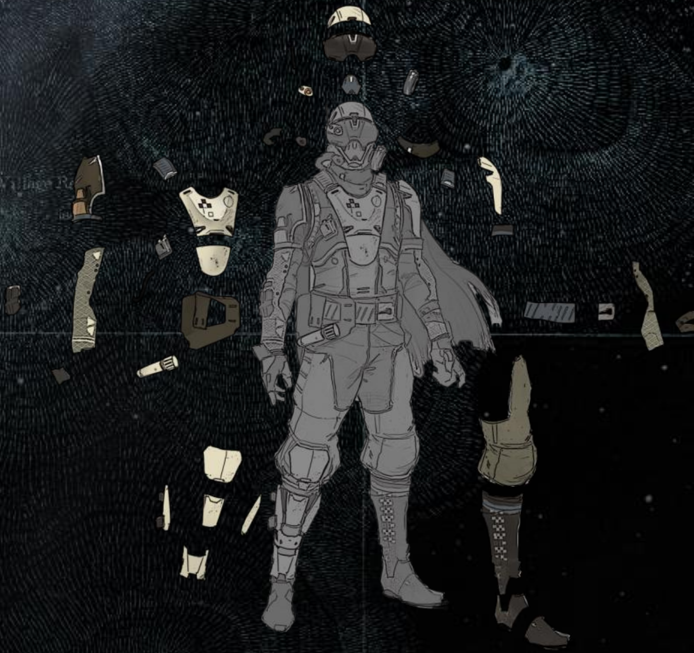
[Confidential SIGGRAPH Destiny internal]

Console Generations

- V1: PlayStation 3 and Xbox 360
- V2: PlayStation 4 and Xbox One

Gear Bits: From DCC to Runtime

- Artist-perspective:
  - Regular authoring process
  - Normal object-space modeling and texturing
- Artists assign properties to bits
  - Shaders
  - Gear dyes
  - Textures
  - Skinning
  - …



Gear Bits: From DCC to Runtime

- Tech ensures bits assembly offline
- Don’t make artists worry about drawcall or shader count
- Make the process transparent to content creators

Gear Bits: From DCC to Runtime

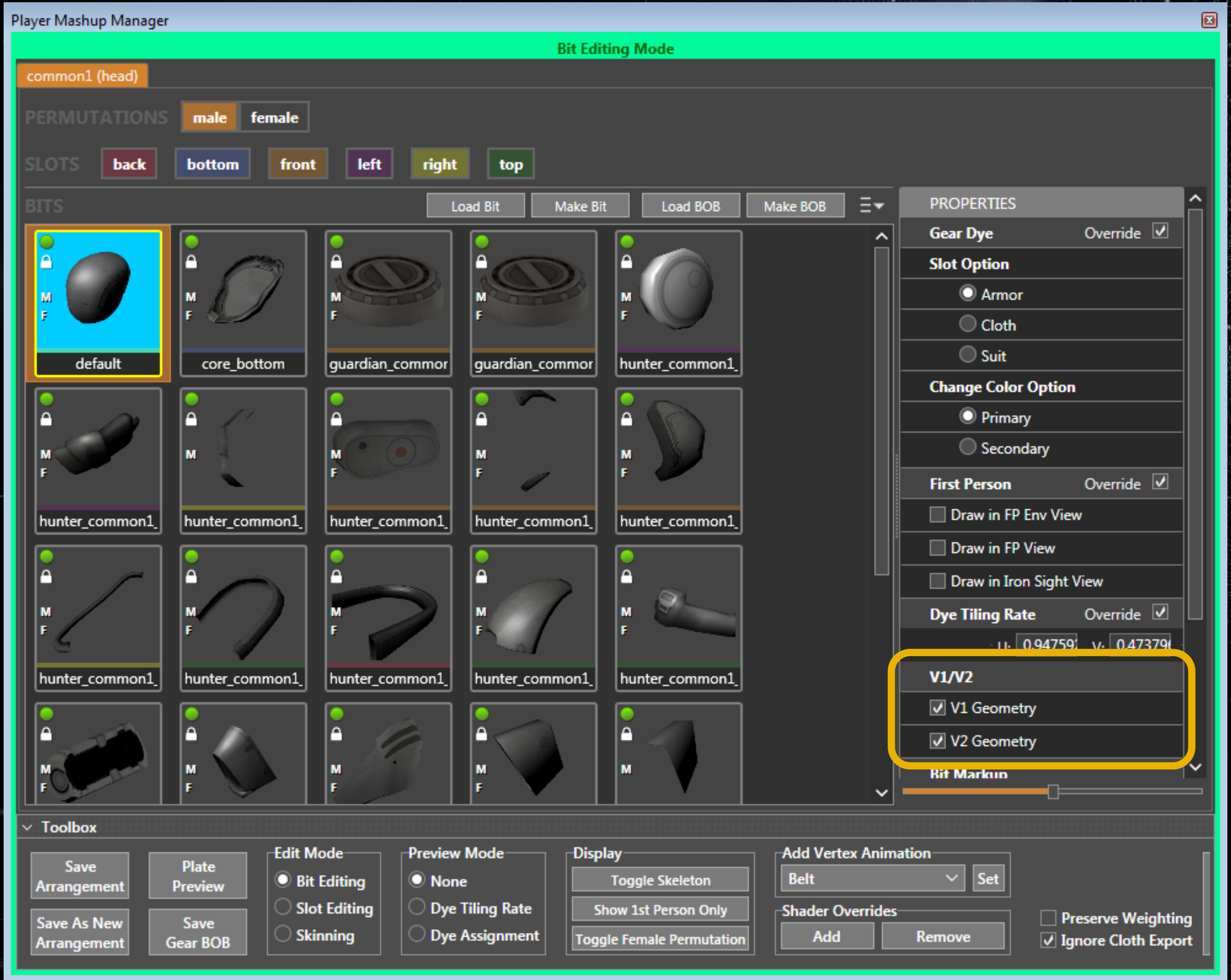
- Merge the bits during import

Gear Bits: From DCC to Runtime

- Merge the bits during import
- Sort by state and merge geometry that has the same state into same drawcall containers

Future-Proofing Gear Geometry

- Artists author high-resolution detailed geometry for bits
- Scalability ensured via gear arrangement tagging and LOD system



Gear Dye Override ☑

Slot Option
● Armor
○ Cloth
○ Suit

Change Color Option
● Primary
○ Secondary

First Person Override ☑
□ Draw in FP Env View
□ Draw in FP View
□ Draw in Iron Sight View

Dye Tiling Rate Override ☑
U: 0.94759; V: 0.47379(

V1/V2
☑ V1 Geometry
☑ V2 Geometry

Edit Mode
● Bit Editing
○ Slot Editing
○ Skinning

Preview Mode
● None
○ Dye Tiling Rate
○ Dye Assignment

Display

Add Vertex Animation
Belt

Shader Overrides

□ Preserve Weighting
☑ Ignore Cloth Export

First-Person Geo to Third-Person

- Base LOD authored to first-person quality
- LOD system automatically drops down in third-person from base geo

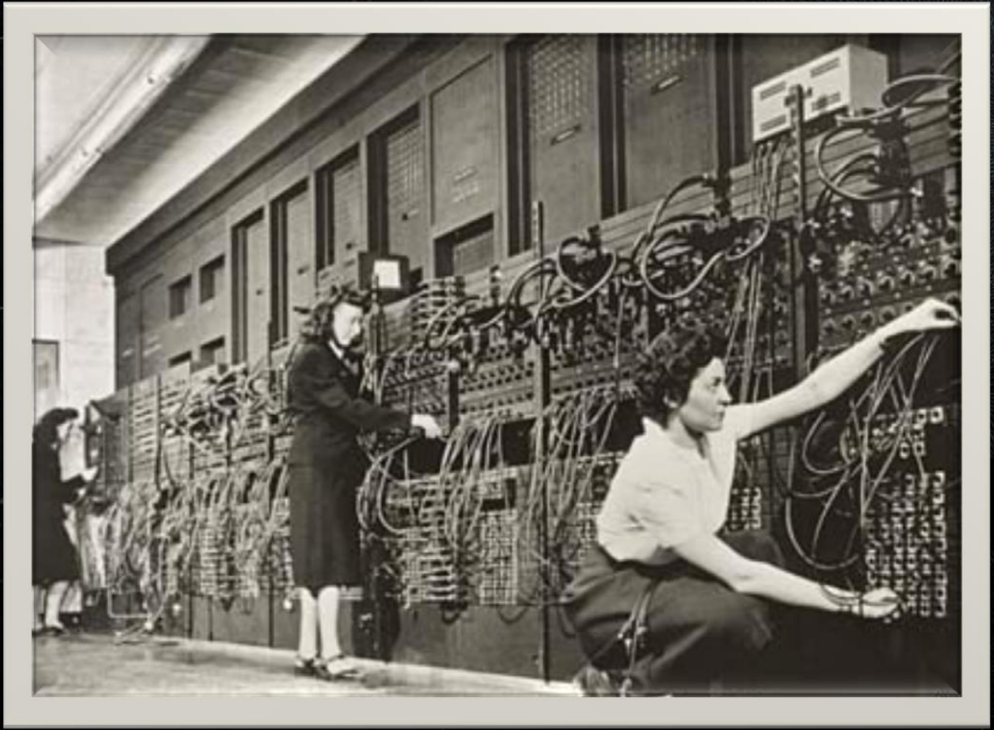
Gear LOD Generation

- Auto-generate 2 additional LOD meshes offline from base
  - Separate geometry for each platform generation
- Medium LOD mesh memory is shared with base LOD (~30% base)
  - Same vertex buffer
  - Different index buffers

Gear LOD Generation

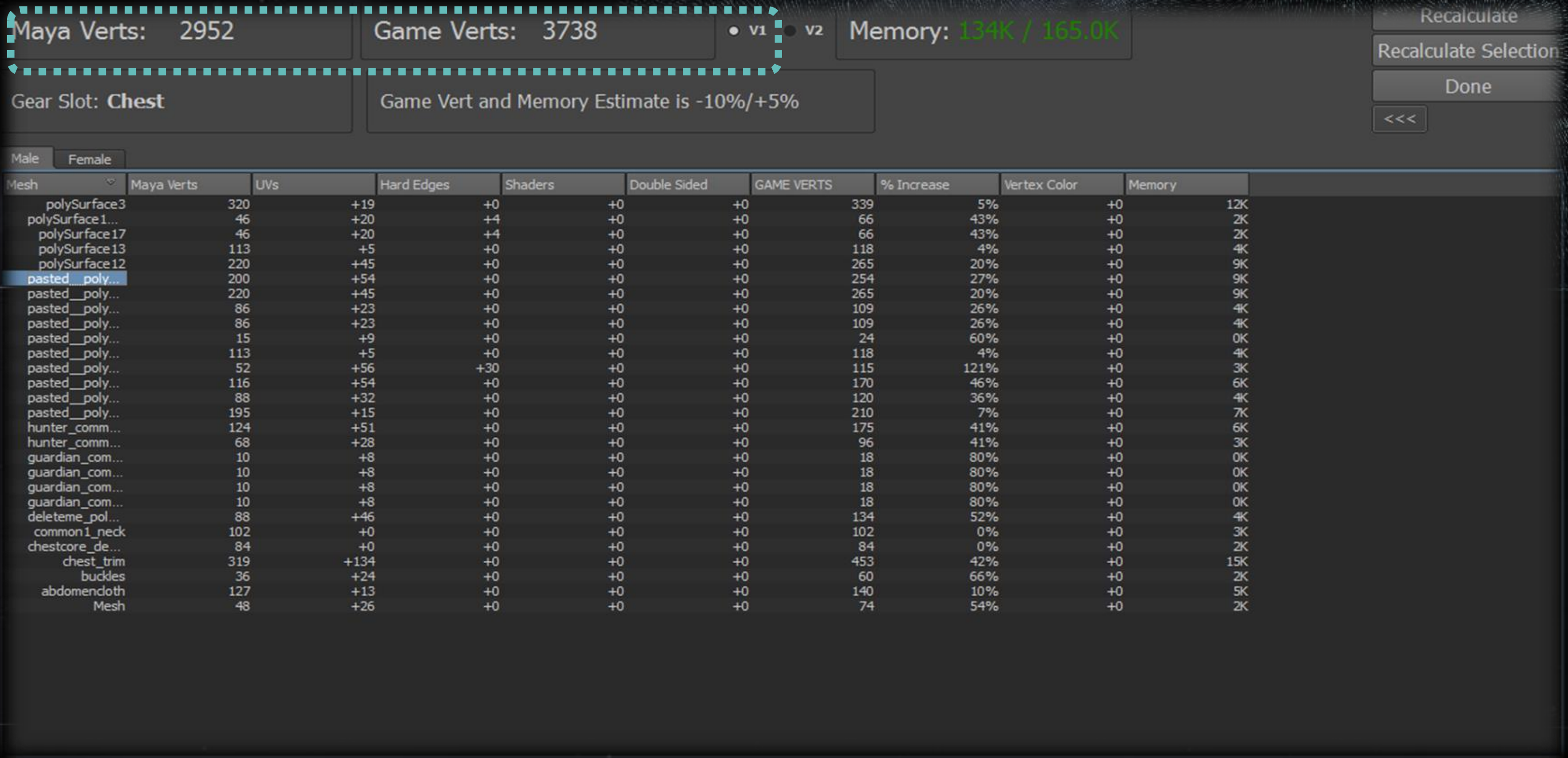
- Auto-generate 2 additional LOD meshes offline from base
- Medium LOD mesh memory is shared with base LOD (~30% base)
  - Same vertex buffer
  - Different index buffers
- Imposter LOD uses very simplified mesh formats (~10% base)
  - One to two bones per vertex only
  - No tangent space
  - Reduced skeleton

Budget and Budget Reporting



+

Gear Arrangement Geometry Profiler



Maya Verts: 2952

Game Verts: 3738

Memory: 134K / 165.0K

Gear Slot: **Chest**

Game Vert and Memory Estimate is -10%/+5%

Mesh	Maya Verts	UVs	Hard Edges	Shaders	Double Sided	GAME VERTS	% Increase	Vertex Color	Memory
polySurface3	320	+19	+0	+0	+0	339	5%	+0	12K
polySurface1...	46	+20	+4	+0	+0	66	43%	+0	2K
polySurface17	46	+20	+4	+0	+0	66	43%	+0	2K
polySurface13	113	+5	+0	+0	+0	118	4%	+0	4K
polySurface12	220	+45	+0	+0	+0	265	20%	+0	9K
pasted__poly...	200	+54	+0	+0	+0	254	27%	+0	9K
pasted__poly...	220	+45	+0	+0	+0	265	20%	+0	9K
pasted__poly...	86	+23	+0	+0	+0	109	26%	+0	4K
pasted__poly...	86	+23	+0	+0	+0	109	26%	+0	4K
pasted__poly...	15	+9	+0	+0	+0	24	60%	+0	0K
pasted__poly...	113	+5	+0	+0	+0	118	4%	+0	4K
pasted__poly...	52	+56	+30	+0	+0	115	121%	+0	3K
pasted__poly...	116	+54	+0	+0	+0	170	46%	+0	6K
pasted__poly...	88	+32	+0	+0	+0	120	36%	+0	4K
pasted__poly...	195	+15	+0	+0	+0	210	7%	+0	7K
hunter_comm...	124	+51	+0	+0	+0	175	41%	+0	6K
hunter_comm...	68	+28	+0	+0	+0	96	41%	+0	3K
guardian_com...	10	+8	+0	+0	+0	18	80%	+0	0K
guardian_com...	10	+8	+0	+0	+0	18	80%	+0	0K
guardian_com...	10	+8	+0	+0	+0	18	80%	+0	0K
guardian_com...	10	+8	+0	+0	+0	18	80%	+0	0K
deleteme_pol...	88	+46	+0	+0	+0	134	52%	+0	4K
common1_neck	102	+0	+0	+0	+0	102	0%	+0	3K
chestcore_de...	84	+0	+0	+0	+0	84	0%	+0	2K
chest_trim	319	+134	+0	+0	+0	453	42%	+0	15K
buckles	36	+24	+0	+0	+0	60	66%	+0	2K
abdomencloth	127	+13	+0	+0	+0	140	10%	+0	5K
Mesh	48	+26	+0	+0	+0	74	54%	+0	2K

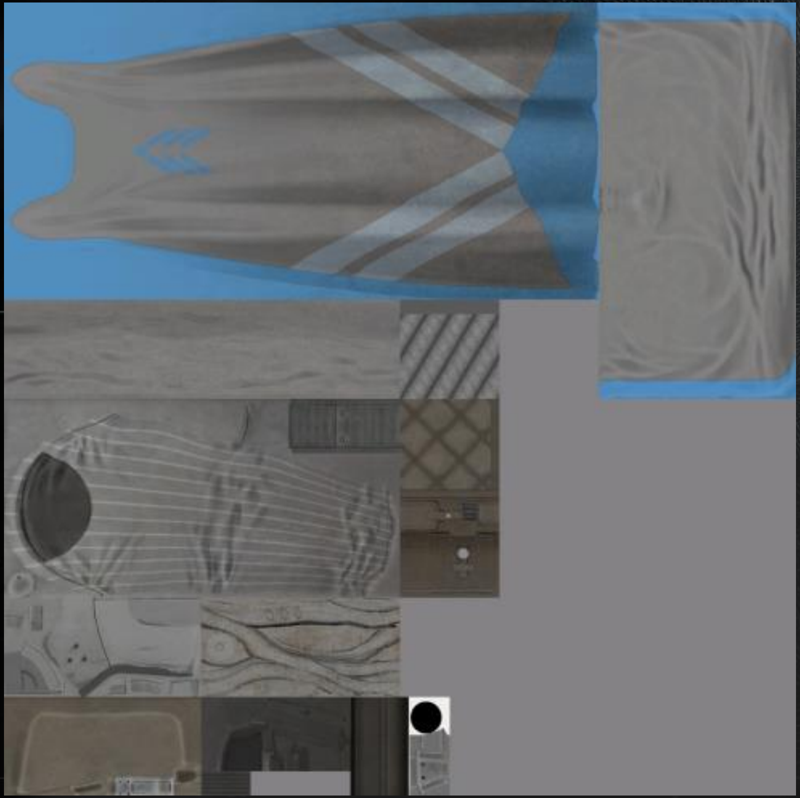
Our Solution: Gear System

1. Gear Slots
2. Arrangements and Bits
3. Plated Textures
4. Gear Dyes

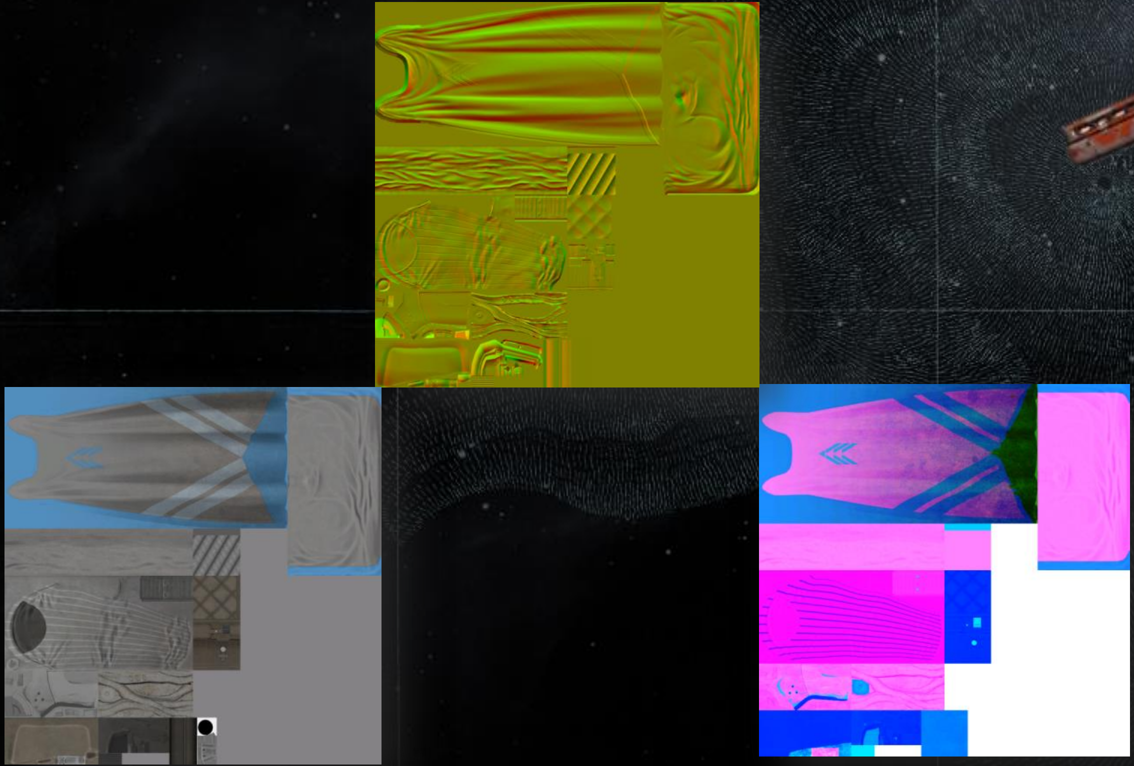
What Is a Gear Plated Texture?

- A special gear texture atlas
- All bits in an arrangement plate to this atlas

What Is a Gear Plated Texture?



Gear Plates



Why Plate?

- Artists are freed up to author bits in intuitive UV layout for each bit
- No need to create custom UV layout for each gear arrangement

Why Plate?

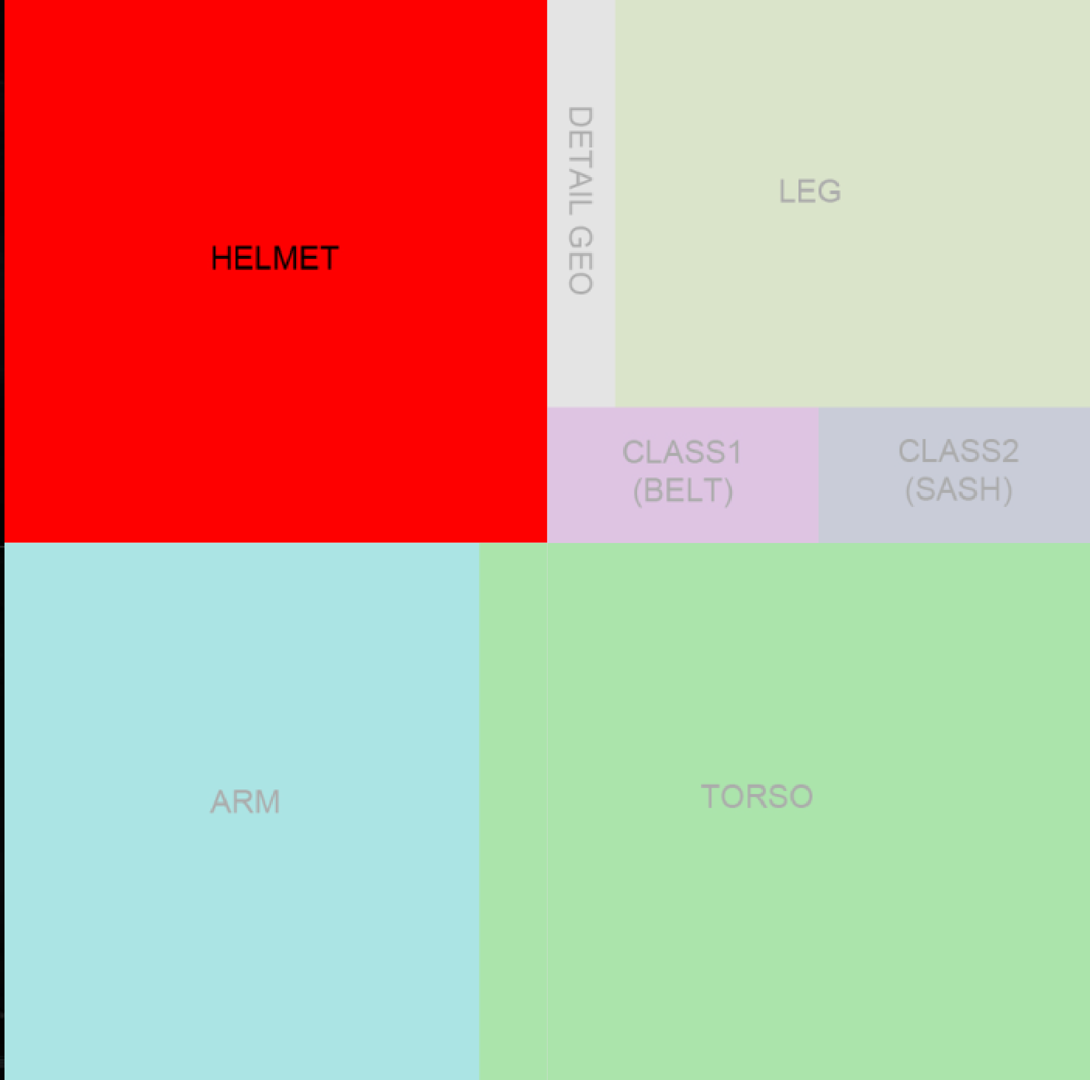
- Artists are freed up to author bits in intuitive UV layout for each bit
- No need to create custom UV layout for each gear arrangement separately
- No need to switch textures at runtime within arrangement
  - Just setup the plate textures and render the whole gear arrangement

Constant Memory Footprint

- All bit textures must fit into a plate for each gear arrangements
- Artists get visual errors when bits don’t fit – in DCC directly
  - Quick interactive preview

Constant Memory Footprint

- All bit textures must fit into a plate for each gear arrangements
- Artists get visual errors when bits don't fit – in DCC directly
  - Quick interactive preview
- Plates automatically scale across v1/v2



HELMET

DETAIL GEO

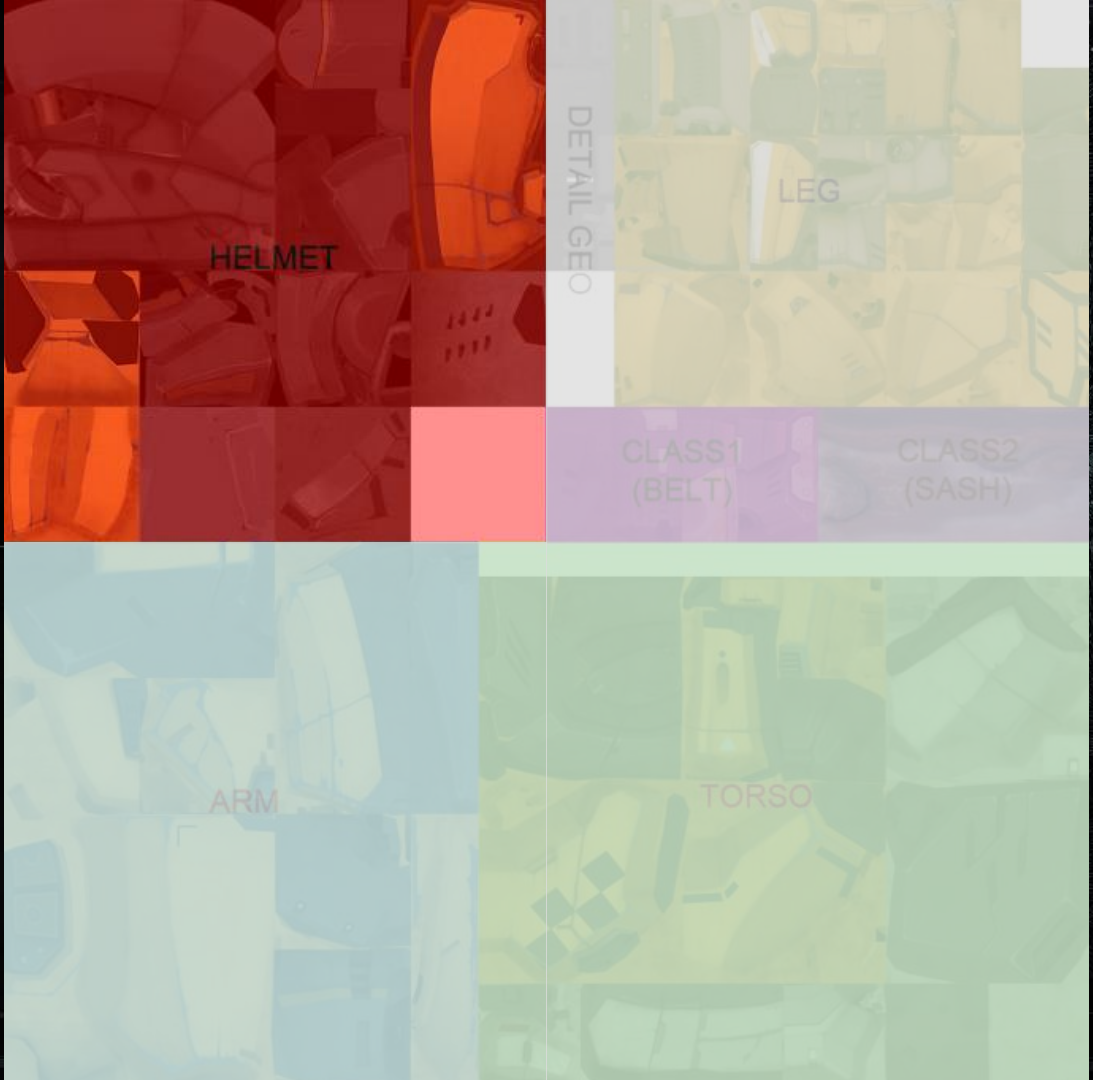
LEG

CLASS1 (BELT)

CLASS2 (SASH)

ARM

TORSO


HELMET
DETAIL GEO
LEG
CLASS1
(BELT)
CLASS2
(SASH)
ARM
TORSO

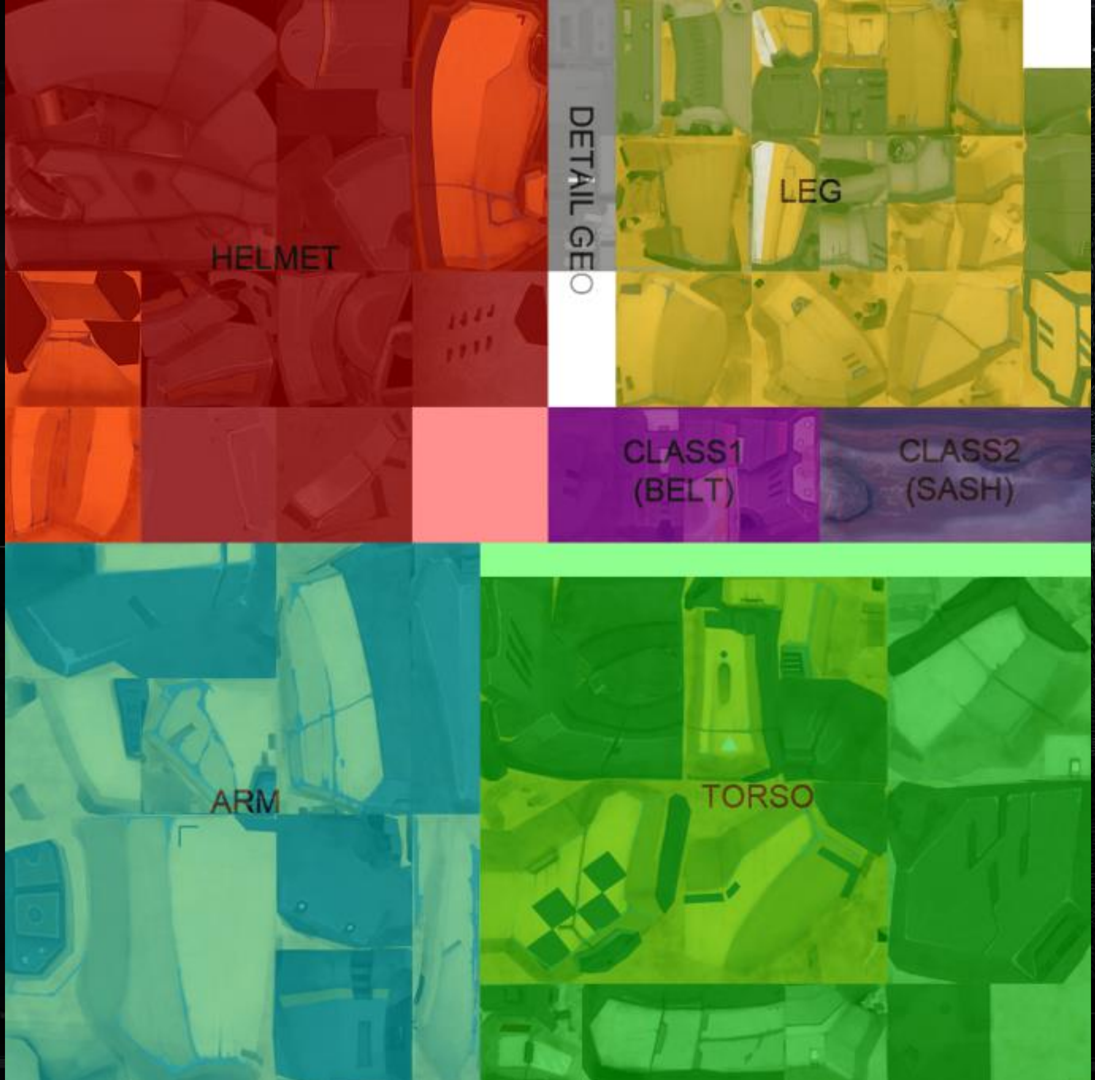
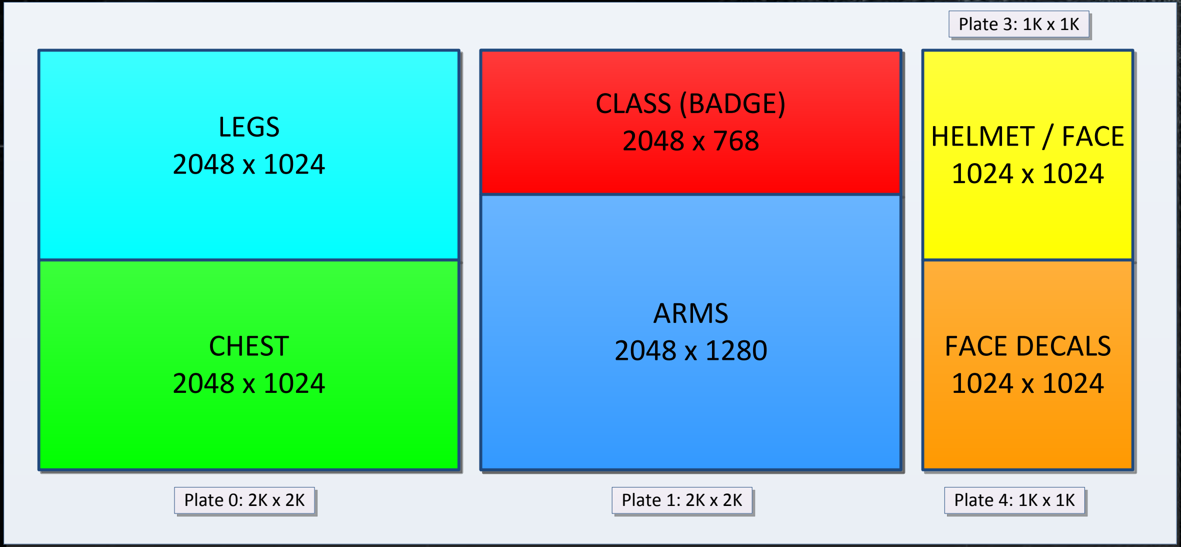

HELMET
DETAIL GEO
LEG
CLASS1
(BELT)
CLASS2
(SASH)
ARM
TORSO

Plate Set: Titan



LEGS
2048 x 1024

CHEST
2048 x 1024

Plate 0: 2K x 2K

CLASS (BADGE)
2048 x 768

ARMS
2048 x 1280

Plate 1: 2K x 2K

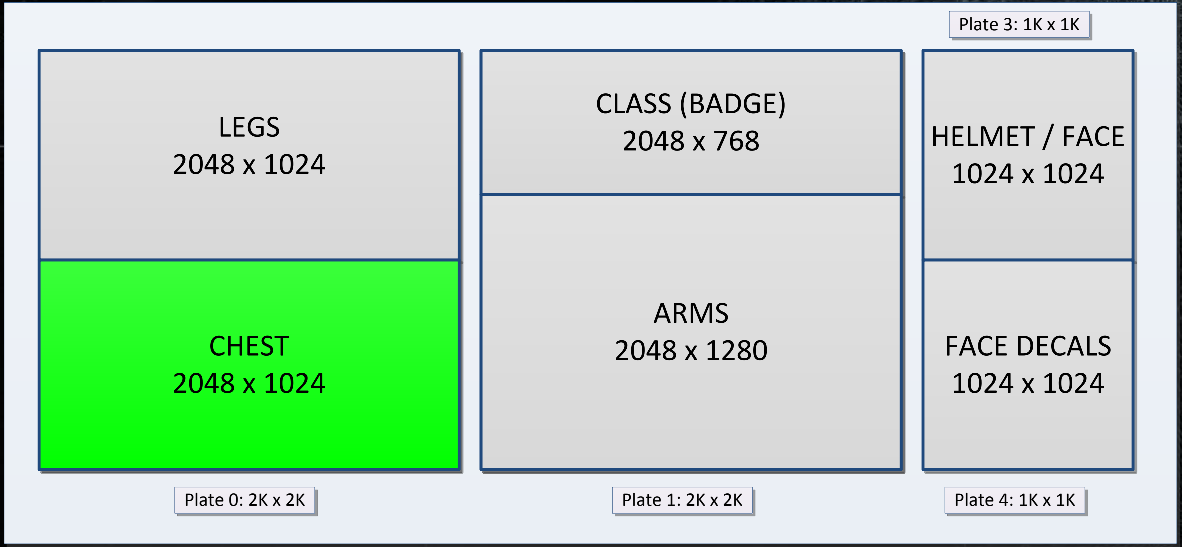
Plate 3: 1K x 1K

HELMET / FACE
1024 x 1024

FACE DECALS
1024 x 1024

Plate 4: 1K x 1K

Plate Set: Titan



LEGS
2048 x 1024

CHEST
2048 x 1024

Plate 0: 2K x 2K

CLASS (BADGE)
2048 x 768

ARMS
2048 x 1280

Plate 1: 2K x 2K

Plate 3: 1K x 1K

HELMET / FACE
1024 x 1024

FACE DECALS
1024 x 1024

Plate 4: 1K x 1K

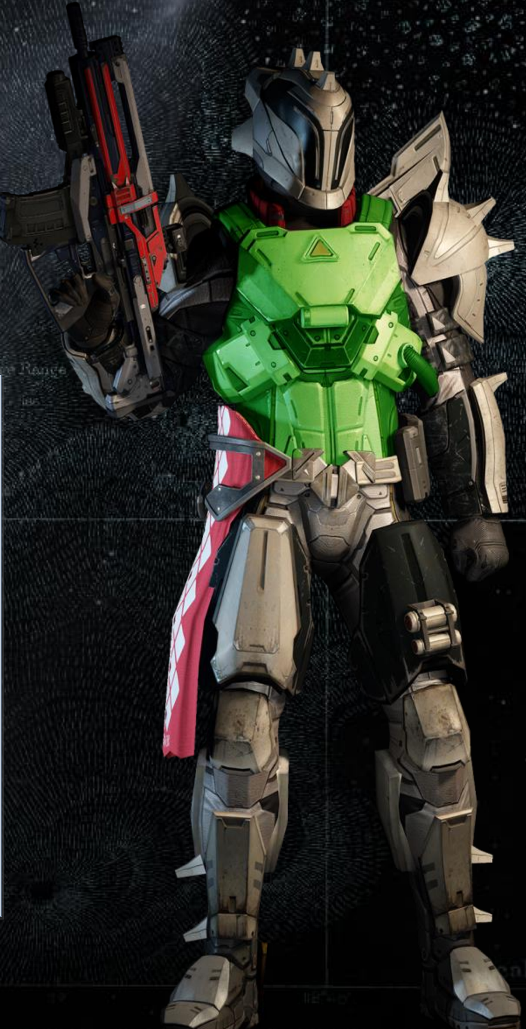
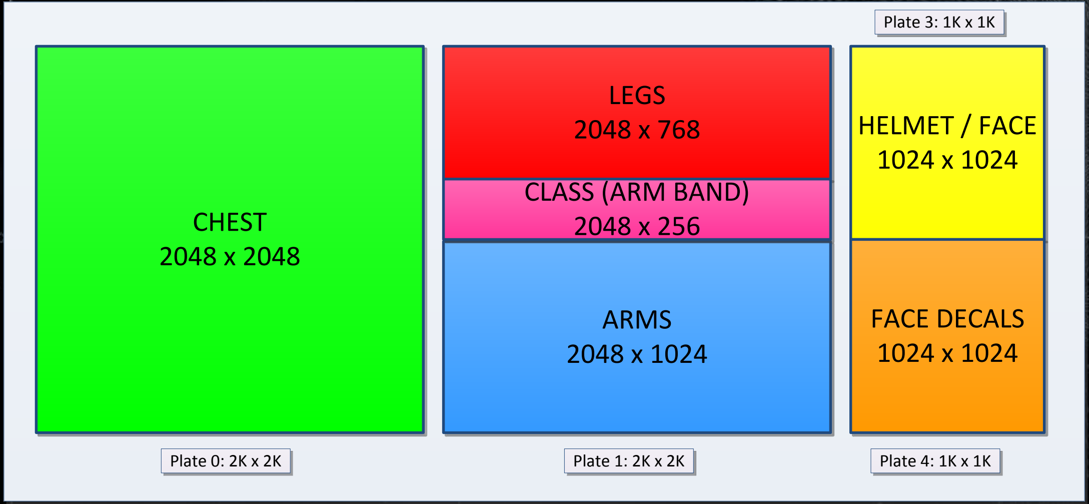


Plate Set: Warlock



CHEST
2048 x 2048

Plate 0: 2K x 2K

LEGS
2048 x 768

CLASS (ARM BAND)
2048 x 256

ARMS
2048 x 1024

Plate 1: 2K x 2K

Plate 3: 1K x 1K

HELMET / FACE
1024 x 1024

FACE DECALS
1024 x 1024

Plate 4: 1K x 1K

Plate Set: Warlock

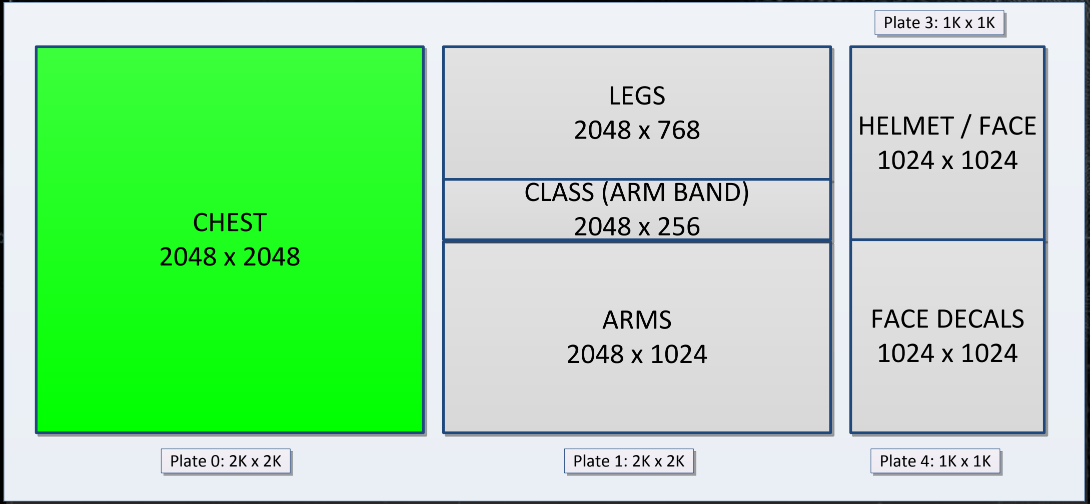


Plate 3: 1K x 1K

CHEST
2048 x 2048

LEGS
2048 x 768

CLASS (ARM BAND)
2048 x 256

ARMS
2048 x 1024

HELMET / FACE
1024 x 1024

FACE DECALS
1024 x 1024

Plate 0: 2K x 2K

Plate 1: 2K x 2K

Plate 4: 1K x 1K

Gear Plating Process: Import

- Walk all bits in in an arrangement
- Pull out the textures
- Create texture descriptors structures (references + plated placements)
  - Do not bake off plate textures offline
- Modify geometry UVs to plated UV locations

Gear Plating Process: Runtime

- Load gear components
- Async stream in texture descriptors and gear bit textures
- Dynamically composite the plates at runtime
- Copy compressed bit textures blocks into plates
- Simple plating rules: no scale or rotation

Our Solution: Gear System

1. Gear Slots
2. Arrangements and Bits
3. Plated Textures
4. Gear Dyes

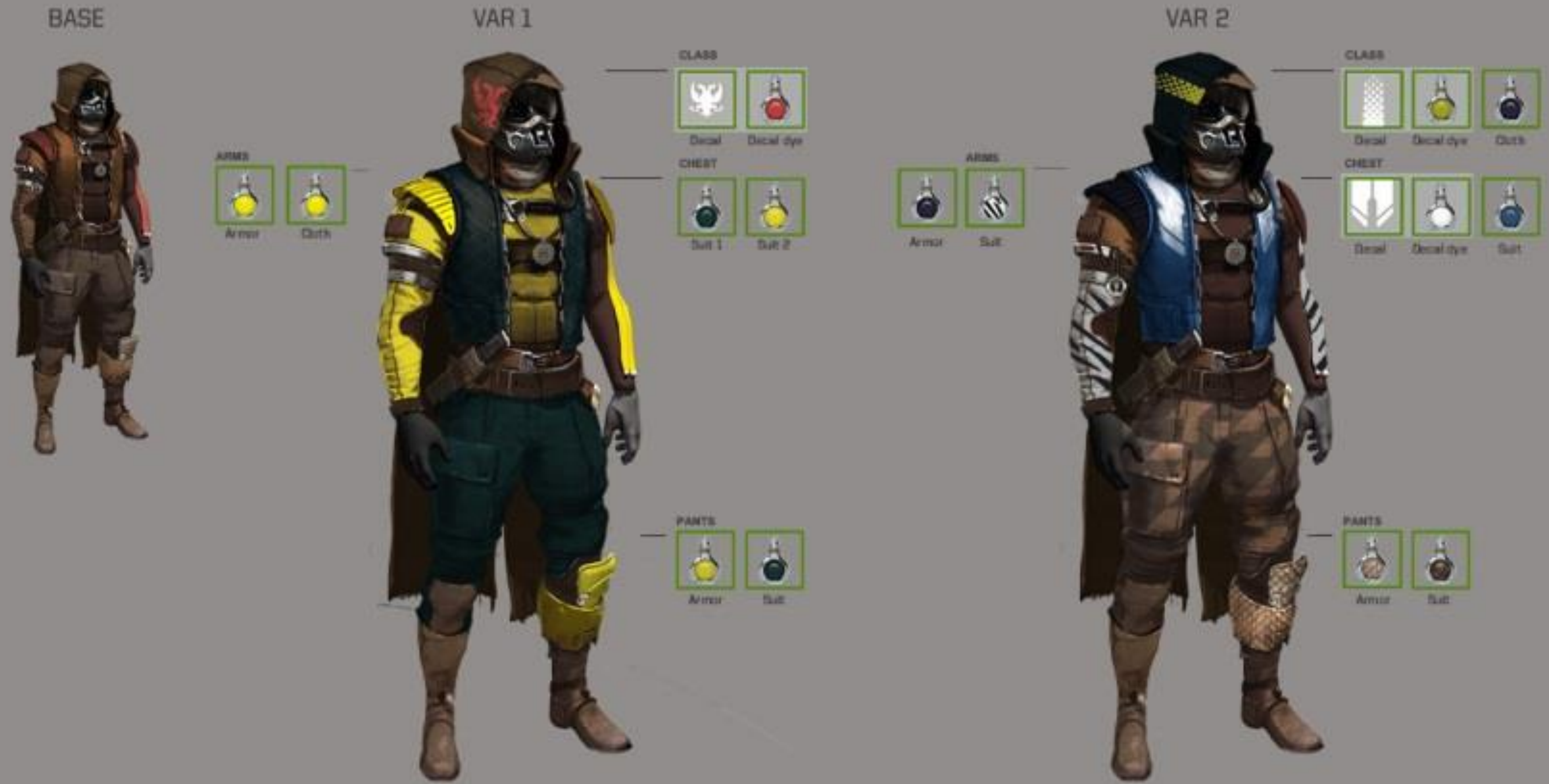
Our Solution: Gear System

1. Gear Slots
2. Arrangements and Bits
3. Plated Textures
4. Gear Dyes
    1. Goals
    2. Tinting
    3. Material response
    4. Gearstack usage
    5. Advanced dyes and materials

Our Solution: Gear System

1. Gear Slots
2. Arrangements and Bits
3. Plated Textures
4. Gear Dyes
    1. Goals
    2. Tinting
    3. Material response
    4. Gearstack usage
    5. Advanced dyes and materials

DYES AND DECALS

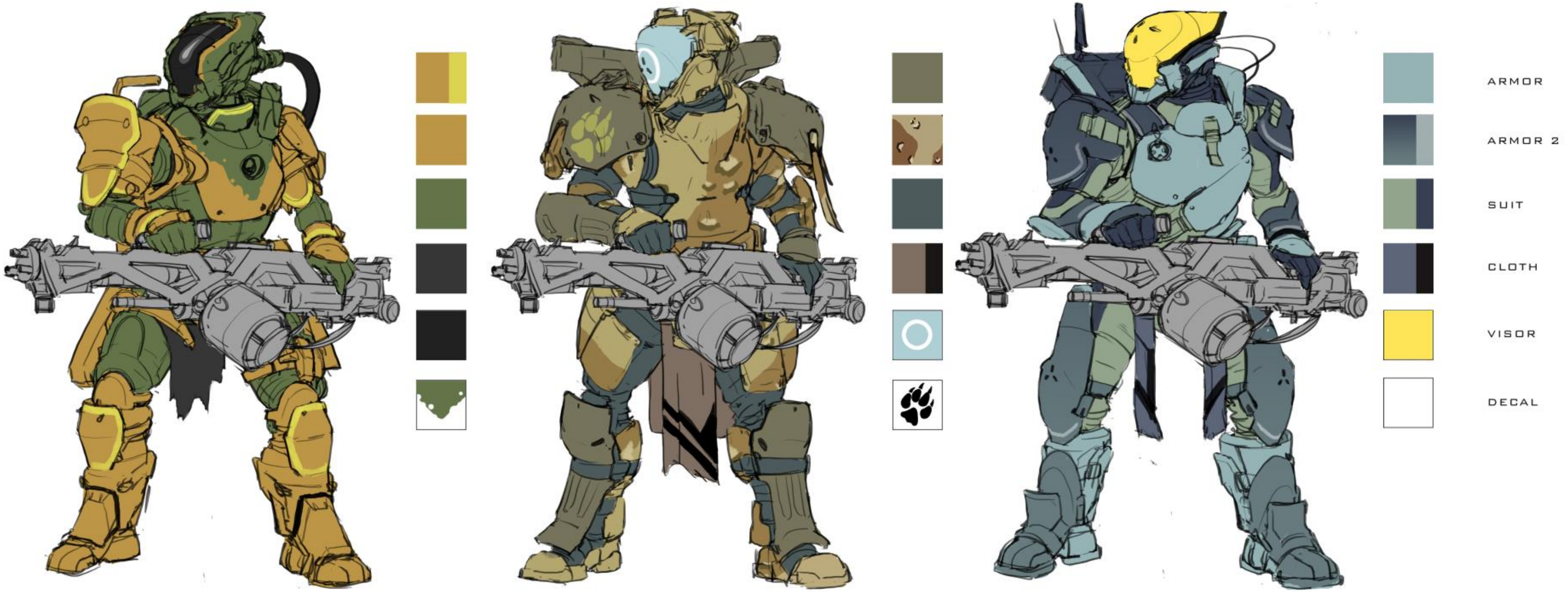


Content Gear Dyes Goals

- Same content supports run-time material and tinting modifications
- Extensible and expressive artist controls
- A library of sharable materials
- A dye is a constant memory material container

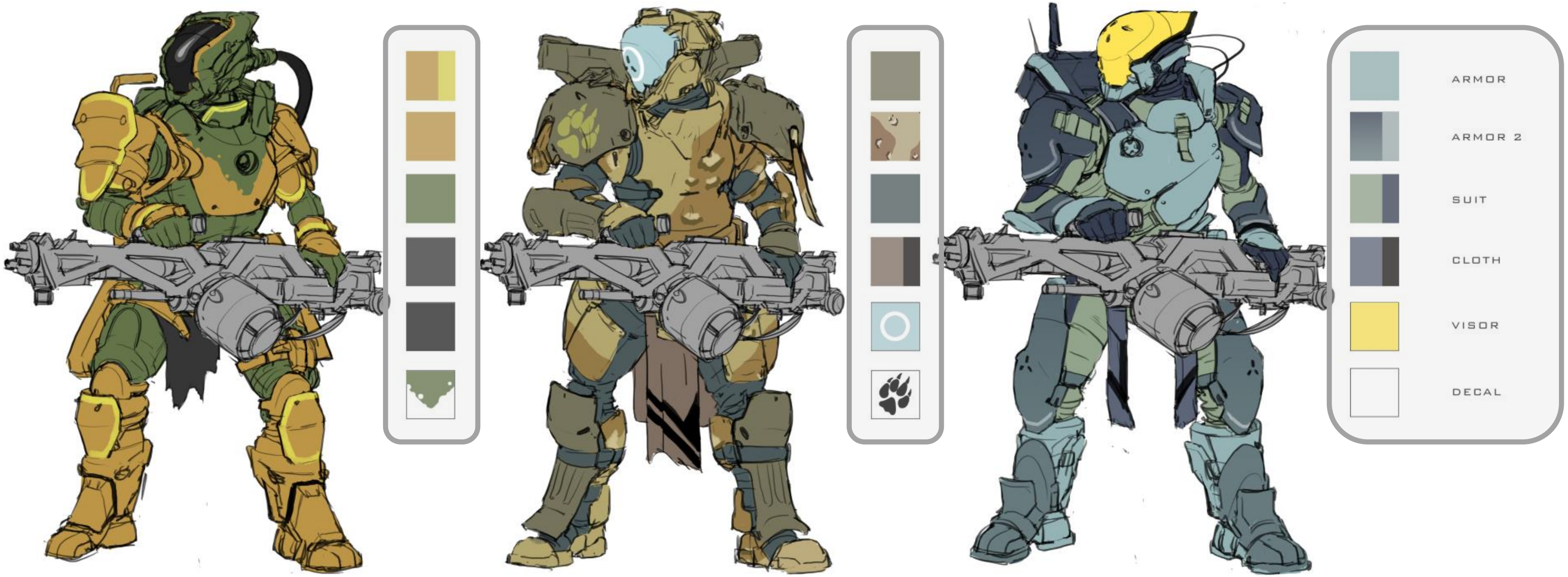
Content Gear Dyes Goals

- Consistent look across the entire character
- Sharable material settings



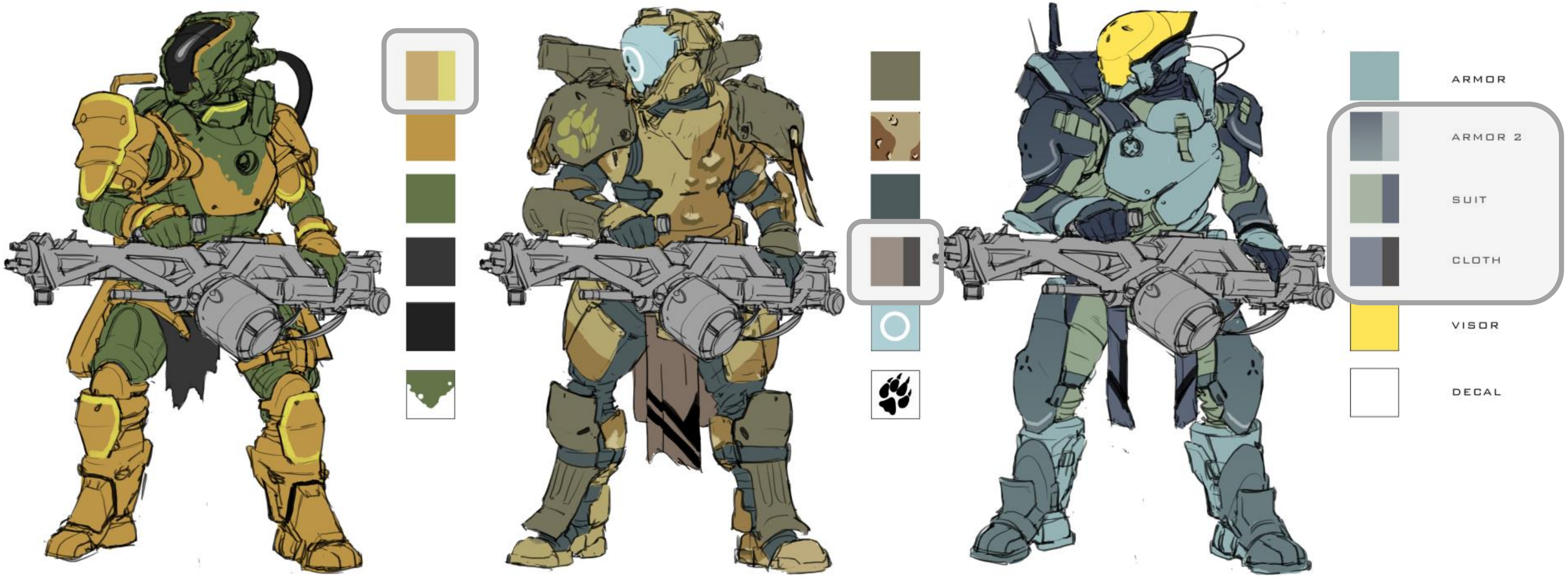
Content Gear Dyes Goals

- Consistent look across the entire character
- Sharable material settings



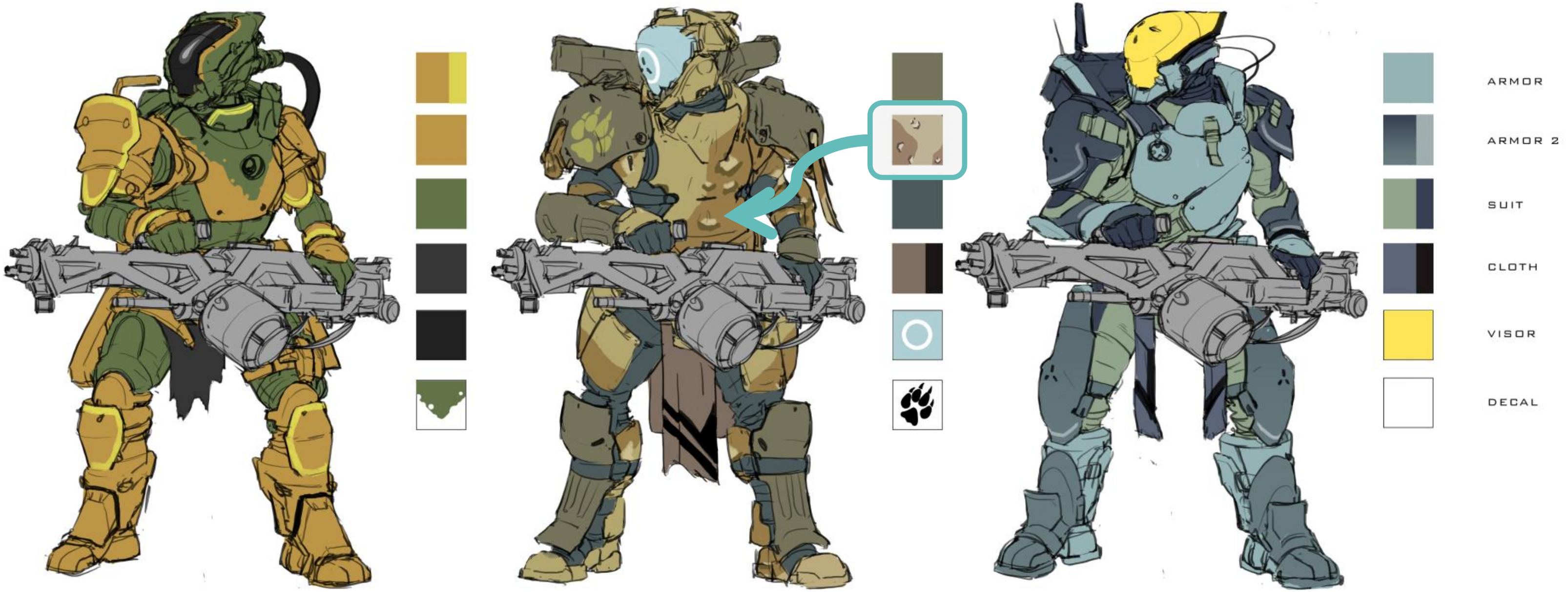
Content Gear Dyes Goals

- Consistent look across the entire character
- Sharable material settings



Content Gear Dyes Goals

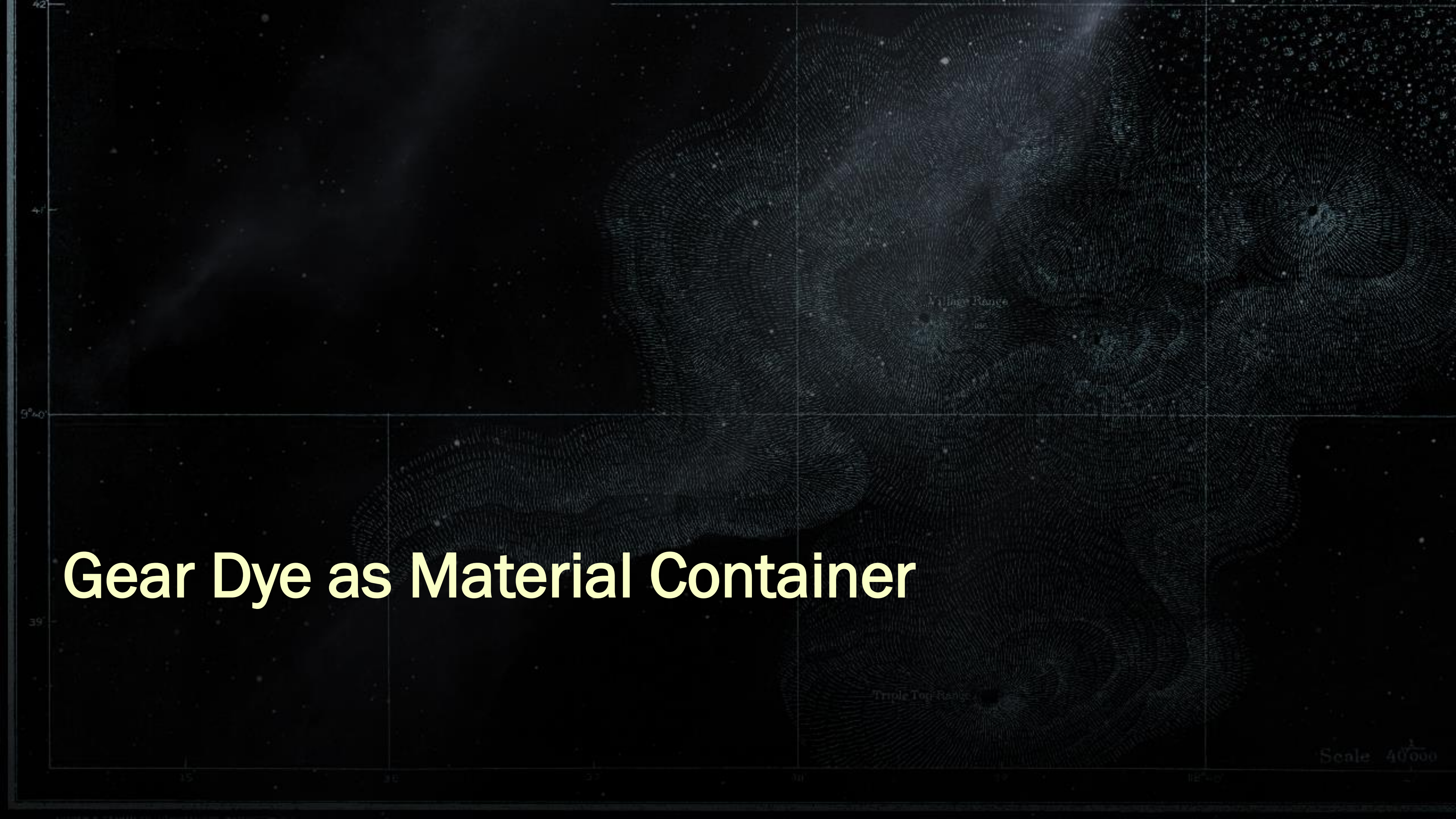
- Consistent look across the entire character
- Sharable material settings



Gear Dye Versatility

- Dyes are always the same data format
- Consistent materials across the game
  - Can be applied to any arrangement / any race / any class
  - Armor / Weapon / Ships, etc.
- Easy hook up for investment

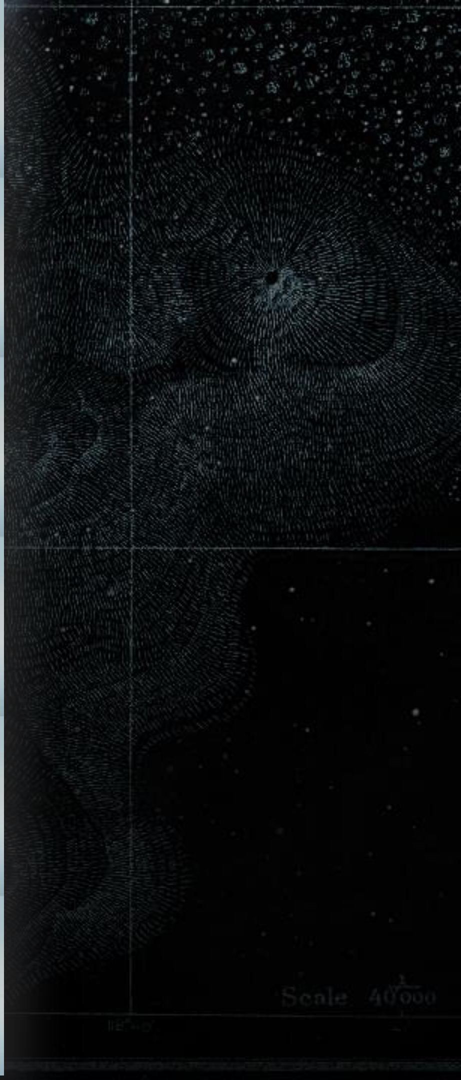
Gear Dye as Material Container



Gear Dye

- A collection of material parameters
  - Tinting
  - Detail Textures
  - Material response properties

42
41'
5°40'
39'
15

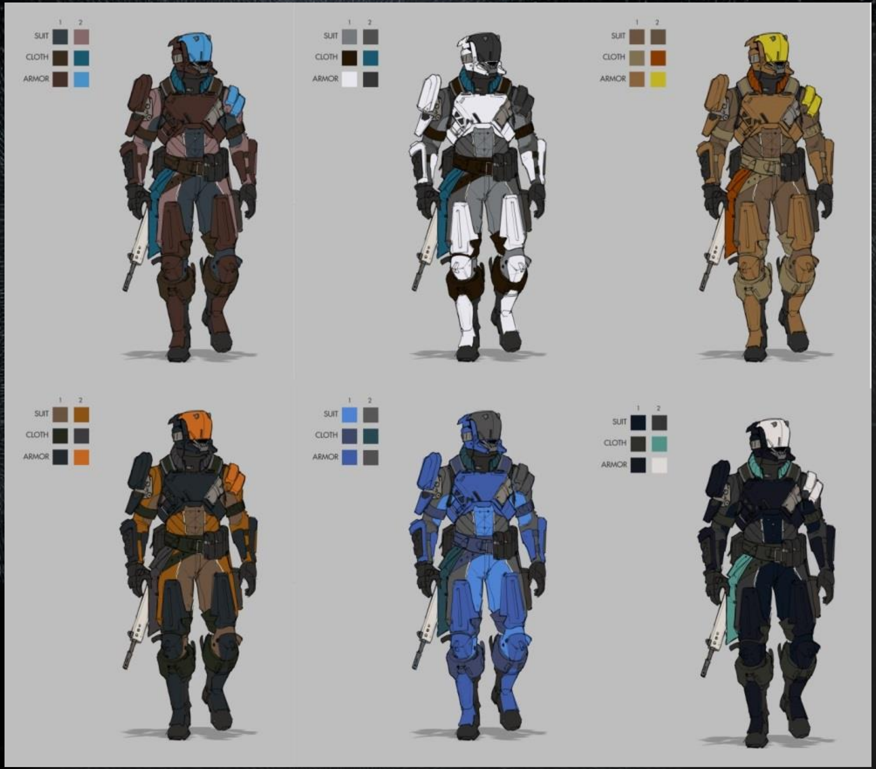

Scale 1/40000

Our Solution: Gear System

1. Gear Slots
2. Arrangements and Bits
3. Plated Textures
4. Gear Dyes
    1. Goals
    2. Tinting
    3. Material response
    4. Gearstack usage
    5. Advanced dyes and materials

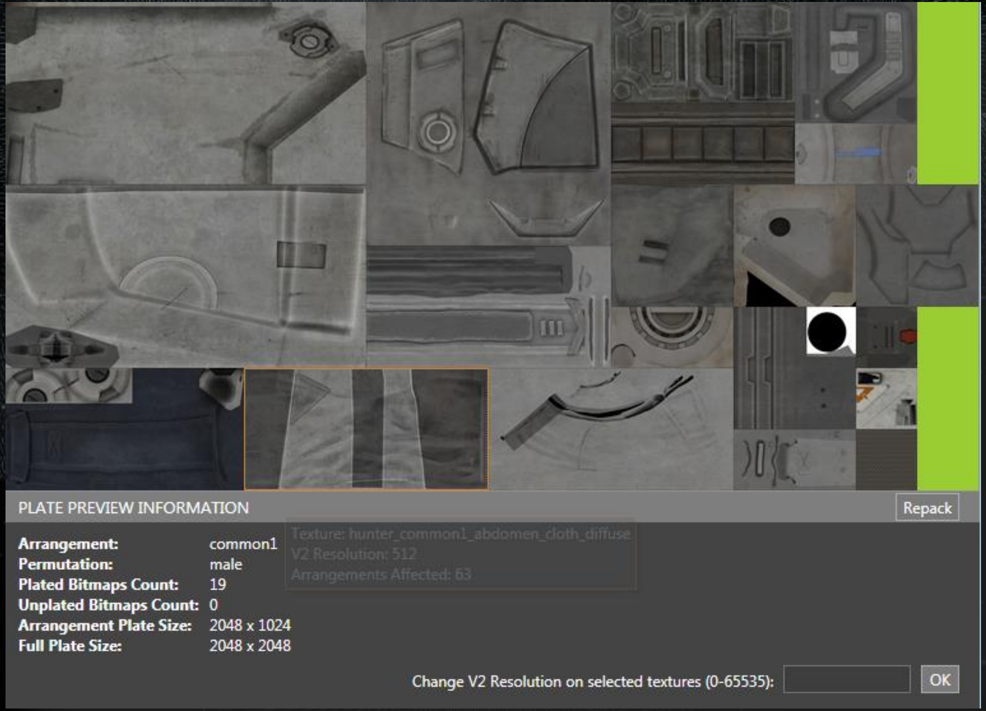
Tinting Gear

- Runtime colorization with artist-friendly Photoshop-like controls



Gear Dye Tint Compositing

- Start with base gear textures



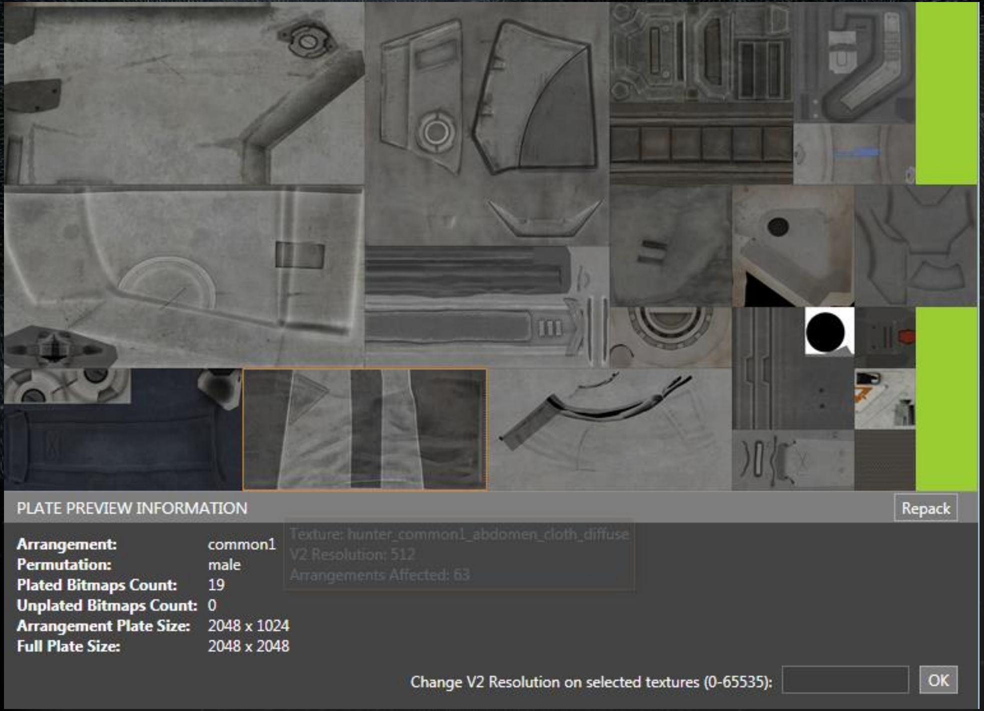
Arrangement: common1
Permutation: male
Plated Bitmaps Count: 19
Unplated Bitmaps Count: 0
Arrangement Plate Size: 2048 x 1024
Full Plate Size: 2048 x 2048

Texture: hunter_common1_abdomen_cloth_diffuse
V2 Resolution: 512
Arrangements Affected: 63

Change V2 Resolution on selected textures (0-65535):

Gear Dye Tint Compositing

- Start with base gear textures
- Greyscale offers most freedom for dyes



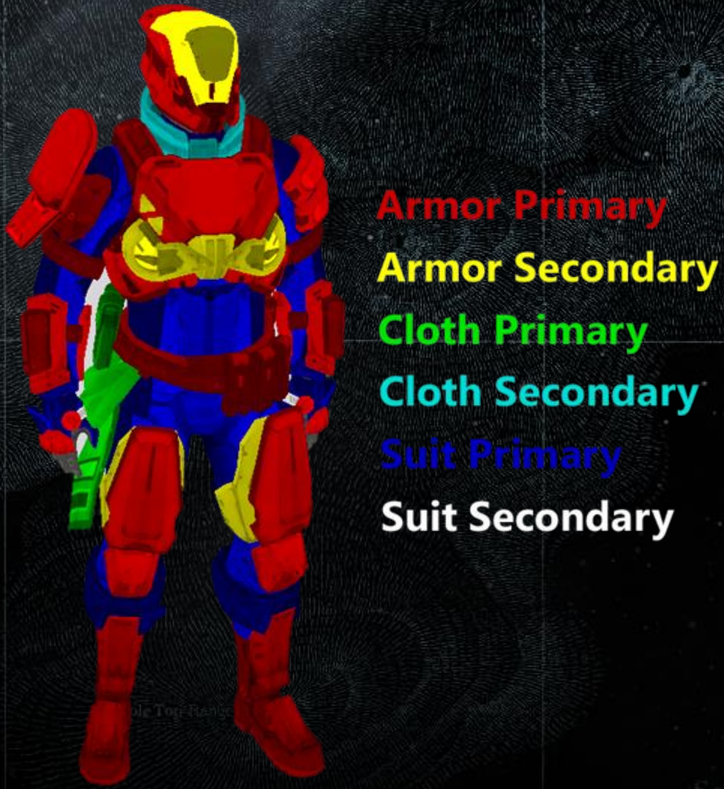
Texture: hunter_common1_abdomen_cloth_diffuse
V2 Resolution: 512
Arrangements Affected: 63

Arrangement: common1
Permutation: male
Plated Bitmaps Count: 19
Unplated Bitmaps Count: 0
Arrangement Plate Size: 2048 x 1024
Full Plate Size: 2048 x 2048

Change V2 Resolution on selected textures (0-65535):

Dye Slot Setup For Gear Arrangements

- Unique dyes per gear arrangement
  - 3 regular dye slots
  - 2 change color options per dye Primary and secondary
  - One special dye for emblem decals usage

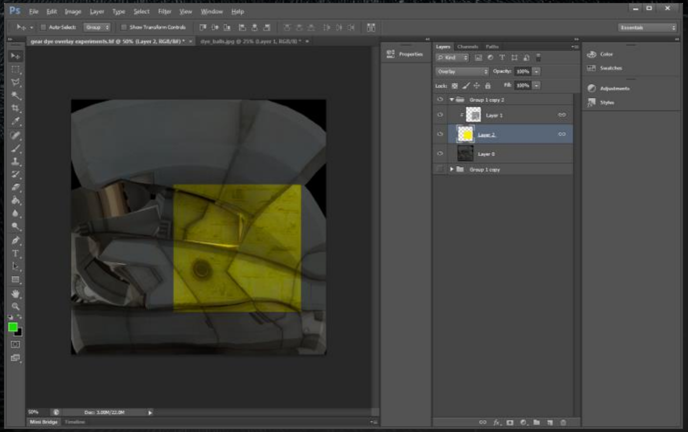


Gear Fixed-Function Shaders

- One unique gear dye per drawcall
  - Some plats didn't like shader flow control
- Except for heads
  - More on that later
- Constant amount of dye parameters

Use Photoshop Blend Modes

- Gear shaders always used the same blends for dyes
- Detail textures use *hard light* to blend with dye change color
- Dye change color uses *overlay* to blend with base textures



Overlay

Opacity: 100%

Lock:

Fill: 100%

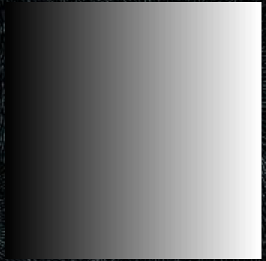
Photoshop Shader Math: Hard Light

- Non-commutative
- if $Blend > \frac{1}{2}$

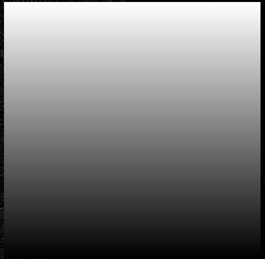
$$R = 1 - (1 - Base) * (1 - 2 * (Blend - \frac{1}{2}))$$

- if $Base \leq \frac{1}{2}$

$$R = Base * (2 * Blend)$$

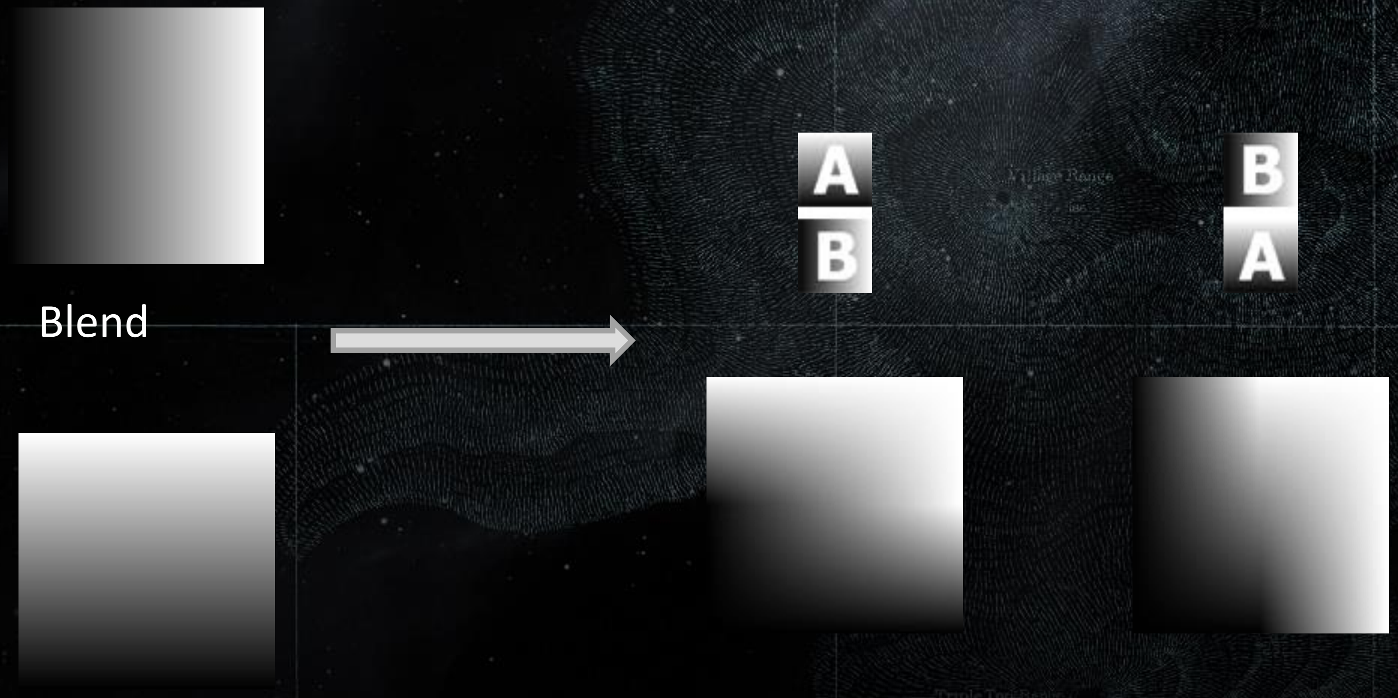


Blend



Base

Photoshop Shader Math: Hard Light



Blend

Base

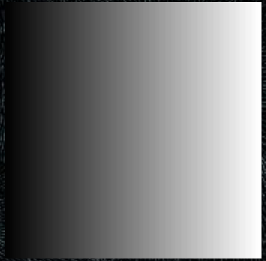
Photoshop Shader Math: Overlay

- Non-commutative
- if $Base > \frac{1}{2}$

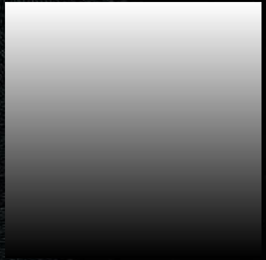
$$R = 1 - \left(1 - 2 * \left(Base - \frac{1}{2}\right)\right) * (1 - Blend)$$

- if $Base \leq \frac{1}{2}$

$$R = (2 * Base) * Blend$$

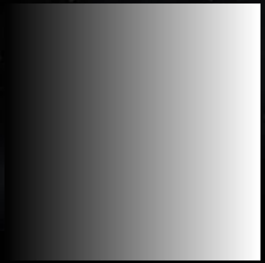


Blend

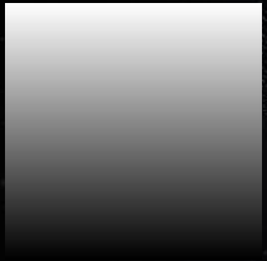


Base

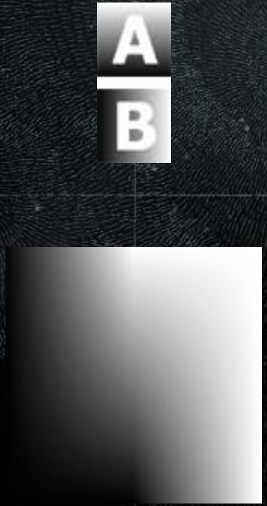
Photoshop Shader Math: Overlay

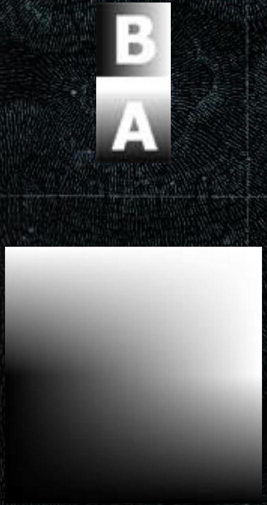


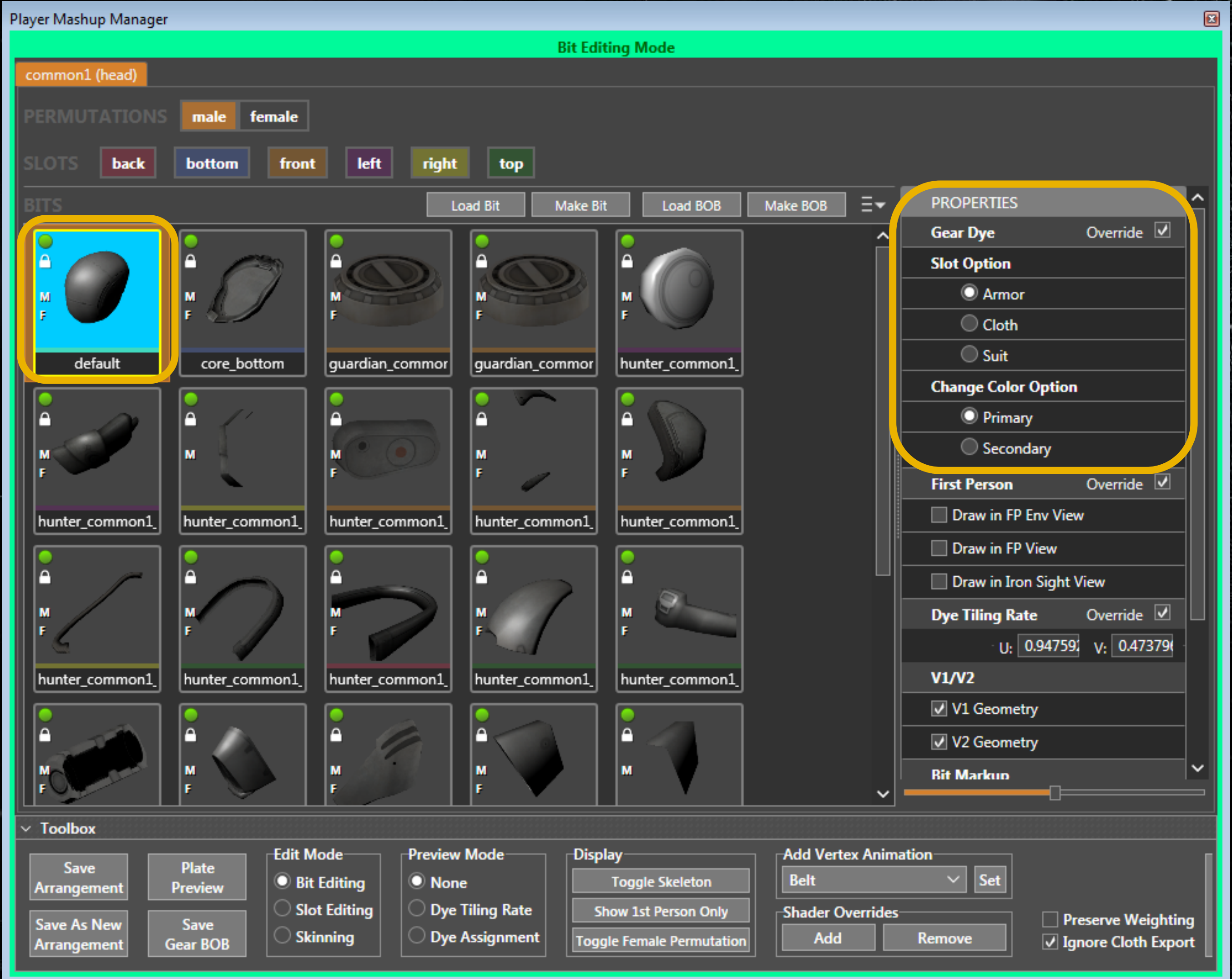
Blend



Base







Player Mashup Manager

Bit Editing Mode

common1 (head)

PERMUTATIONS: male | female

SLOTS: back | bottom | front | left | right | top

BITS

PROPERTIES

Gear Dye — Override ☑

Slot Option
- ● Armor
- ○ Cloth
- ○ Suit

Change Color Option
- ● Primary
- ○ Secondary

First Person — Override ☑
- □ Draw in FP Env View
- □ Draw in FP View
- □ Draw in Iron Sight View

Dye Tiling Rate — Override ☑

U: 0.94759ž V: 0.47379(

V1/V2
- ☑ V1 Geometry
- ☑ V2 Geometry

Bit Markup

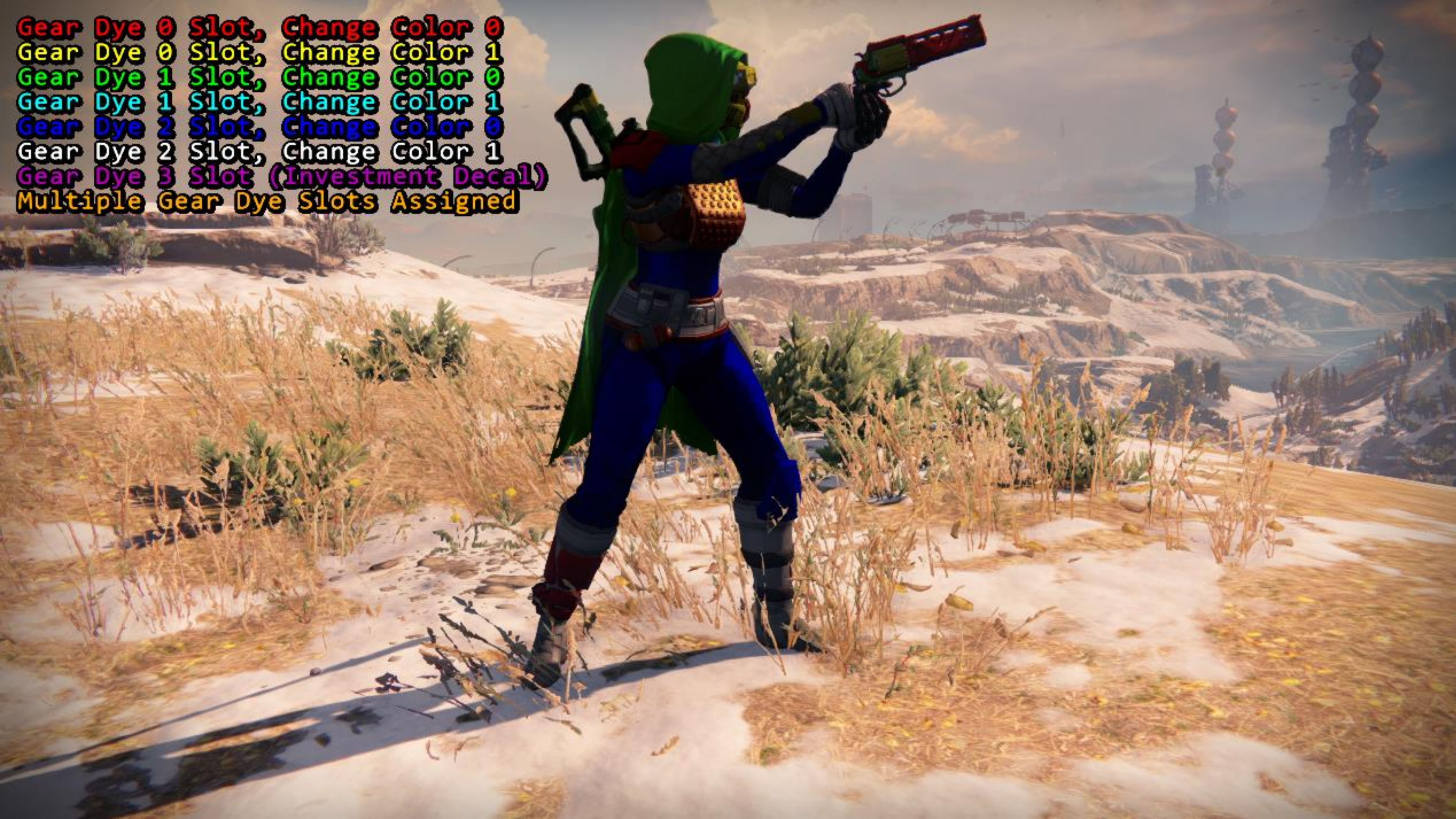
Toolbox

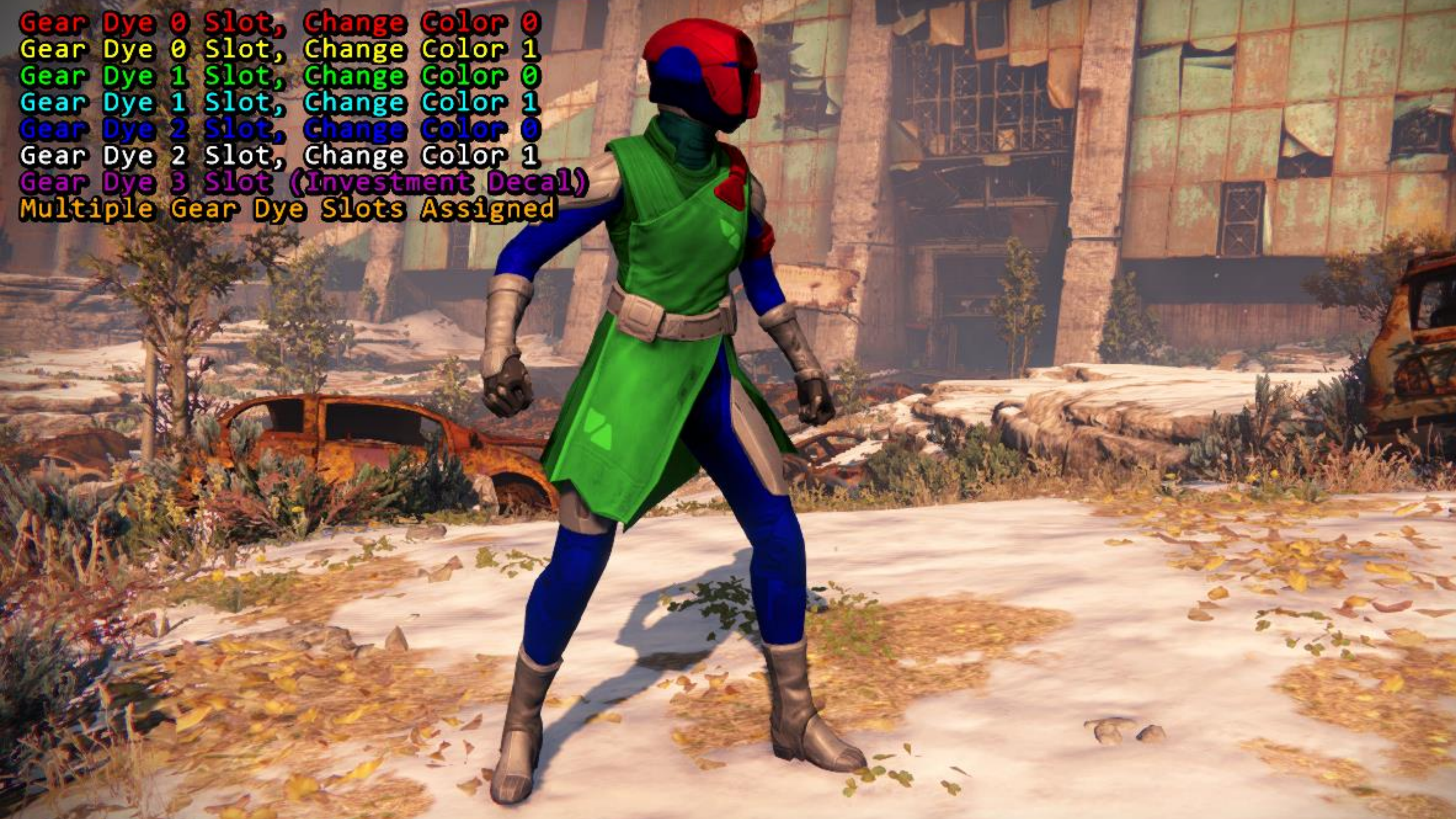
Edit Mode
- ● Bit Editing
- ○ Slot Editing
- ○ Skinning

Preview Mode
- ● None
- ○ Dye Tiling Rate
- ○ Dye Assignment

Add Vertex Animation: Belt

- □ Preserve Weighting
- ☑ Ignore Cloth Export

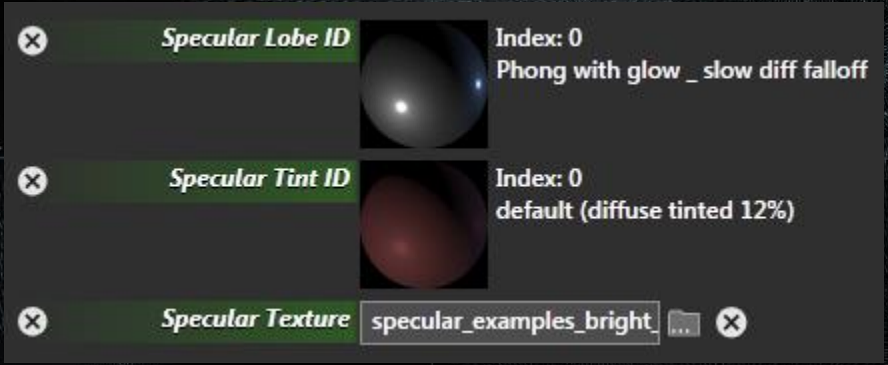

Gear Dye 0 Slot, Change Color 0
Gear Dye 0 Slot, Change Color 1
Gear Dye 1 Slot, Change Color 0
Gear Dye 1 Slot, Change Color 1
Gear Dye 2 Slot, Change Color 0
Gear Dye 2 Slot, Change Color 1
Gear Dye 3 Slot (Investment Decal)
Multiple Gear Dye Slots Assigned


Gear Dye 0 Slot, Change Color 0
Gear Dye 0 Slot, Change Color 1
Gear Dye 1 Slot, Change Color 0
Gear Dye 1 Slot, Change Color 1
Gear Dye 2 Slot, Change Color 0
Gear Dye 2 Slot, Change Color 1
Gear Dye 3 Slot (Investment Decal)
Multiple Gear Dye Slots Assigned

Our Solution: Gear System

1. Gear Slots
2. Arrangements and Bits
3. Plated Textures
4. Gear Dyes
    1. Goals
    2. Tinting
    3. Material response
    4. Gearstack usage
    5. Advanced dyes and materials

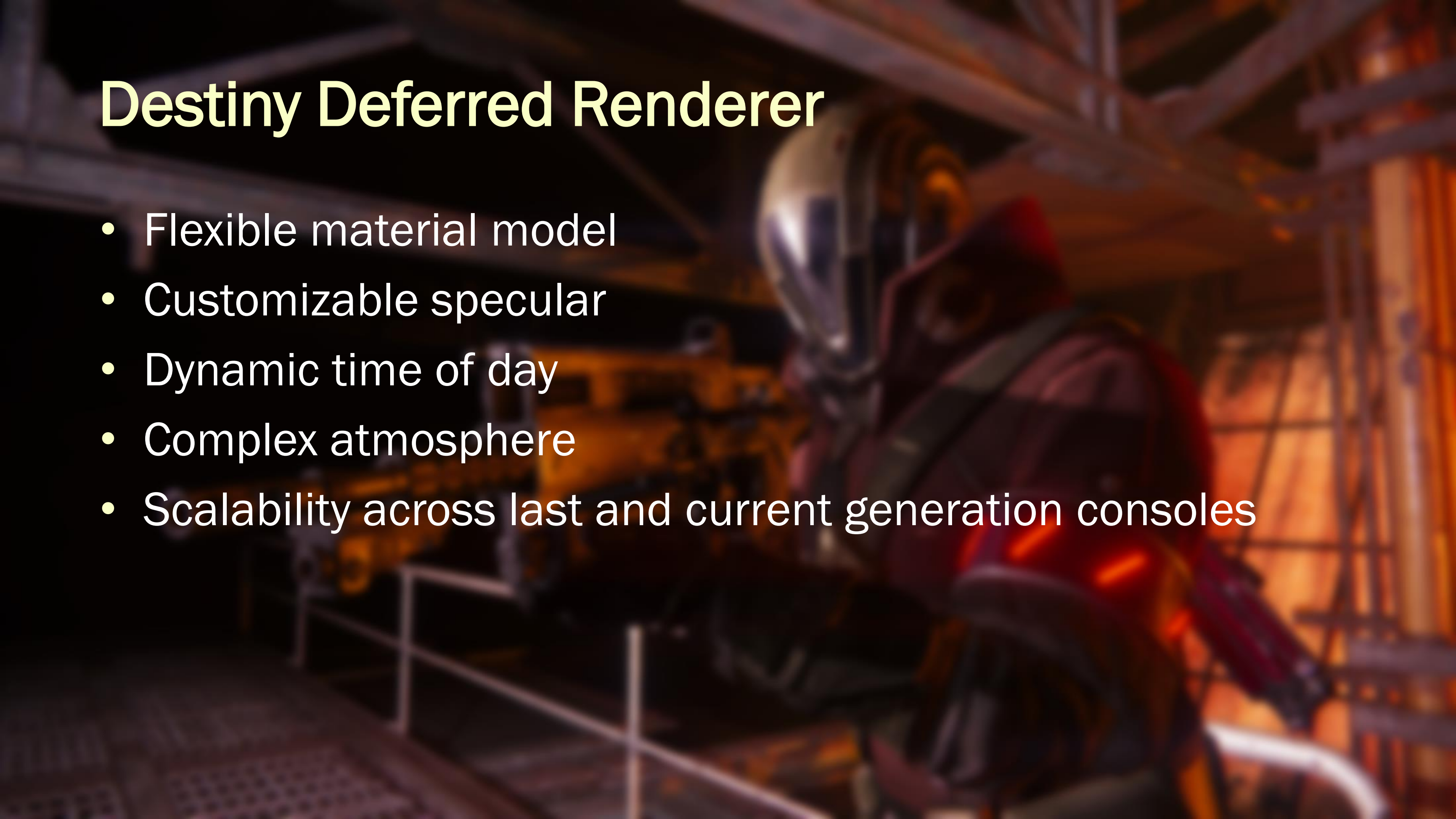
Gear and Dye Specular Response



Specular Lobe ID: Index: 0
Phong with glow _ slow diff falloff

Specular Tint ID: Index: 0
default (diffuse tinted 12%)

Specular Texture: specular_examples_bright_

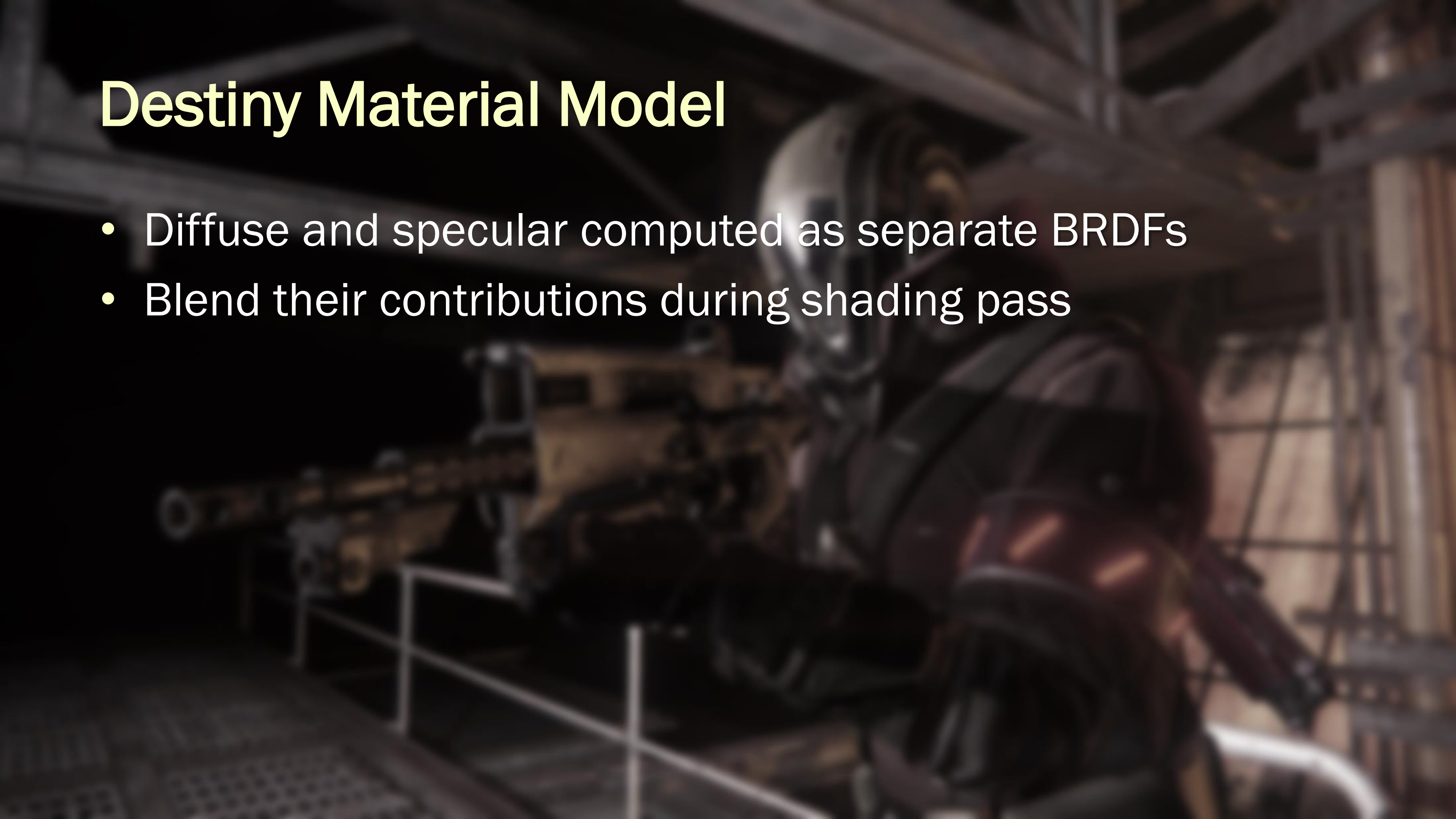


Destiny Deferred Renderer

- Flexible material model
- Customizable specular
- Dynamic time of day
- Complex atmosphere
- Scalability across last and current generation consoles

Destiny Material Model

- Diffuse and specular computed as separate BRDFs
- Blend their contributions during shading pass



Destiny Material Model

- Nearly-Microfacet BRDFs:

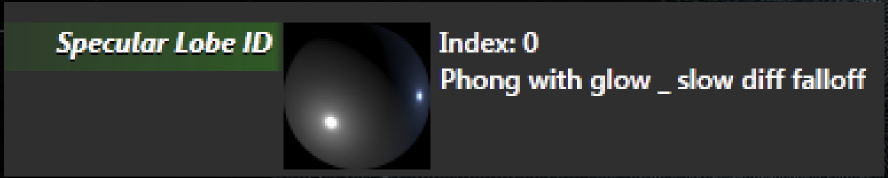
$$f(l,v) = \frac{D(h)F(v \cdot \mathrm{h})G(l,v,h)}{4(n \cdot l)(\mathrm{n} \cdot \mathrm{v})}$$

l	Normalized Light Vector
v	Normalized View Vector
n	Normalized Surface Normal
h	Normalized Half Vector (l + v)
F	microfacet BRDF
G	Geometry term
$D(h)$	Normal distribution term

Destiny Material Model

- Nearly-Microfacet BRDFs:

$$f(l,v)=\frac{D(h)F(v\cdot \mathrm{h})G(l,v,h)}{4(n\cdot l)(\mathrm{n}\cdot \mathrm{v})}$$



Specular Lobe ID

Index: 0

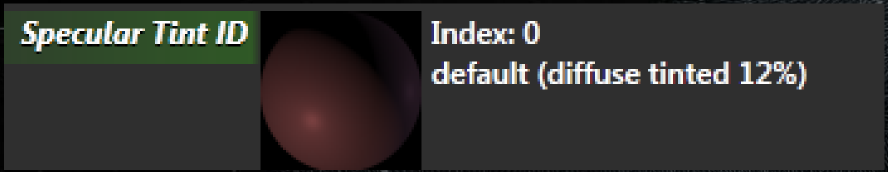
Phong with glow _ slow diff falloff

l	Normalized Light Vector
v	Normalized View Vector
n	Normalized Surface Normal
h	Normalized Half Vector (l + v)
F	microfacet BRDF
G	Geometry term
$D(h)$	Normal distribution term

Destiny Material Model

- Nearly-Microfacet BRDFs:

$$f(l,v) = \frac{D(h)F(v \cdot \mathrm{h})G(l,v,h)}{4(n \cdot l)(\mathrm{n} \cdot \mathrm{v})}$$



Specular Tint ID

Index: 0
default (diffuse tinted 12%)

l	Normalized Light Vector
v	Normalized View Vector
n	Normalized Surface Normal
h	Normalized Half Vector (l + v)
F	microfacet BRDF
G	Geometry term
$D(h)$	Normal distribution term

Destiny Material Model

- Nearly-Microfacet BRDFs:

$$f(l,v)=\frac{D(h)F(v\cdot \mathrm{h})G(l,v,h)}{4(n\cdot l)(\mathrm{n}\cdot \mathrm{v})}$$

l	Normalized Light Vector
v	Normalized View Vector
n	Normalized Surface Normal
h	Normalized Half Vector (l + v)
F	microfacet BRDF
G	Geometry term
$D(h)$	Normal distribution term

Destiny Material Model

- Nearly-Microfacet BRDFs:

$$f(l, v) = \frac{D(h)F(v \cdot \mathrm{h})G(l, v, h)}{4(n \cdot l)(\mathrm{n} \cdot \mathrm{v})}$$

l	Normalized Light Vector
v	Normalized View Vector
n	Normalized Surface Normal
h	Normalized Half Vector (l + v)
F	microfacet BRDF
G	Geometry term
$D(h)$	Normal distribution term

Destiny Material Model

- Our Fresnel term is different
  - We use $\mathrm{F}(v \cdot n)$ not $\mathrm{F}(v \cdot h)$ or $\mathrm{F}(\mathrm{l} \cdot h)$

$$f(l, v) = \frac{D(h)F(v \cdot \mathrm{n})G(l, v, h)}{4(n \cdot l)(\mathrm{n} \cdot \mathrm{v})}$$

Destiny Material Model

- Our Fresnel term is different

$$f(l, v) = \frac{D(h)F(v \cdot \mathrm{n})G(l, v, h)}{4(n \cdot l)(\mathrm{n} \cdot \mathrm{v})}$$

  - We use $\mathrm{F}(v \cdot n)$ not $\mathrm{F}(v \cdot h)$ or $\mathrm{F}(\mathrm{l} \cdot h)$
  - $\mathrm{F}(\mathrm{l} \cdot h)$ is a per-light, per-fragment calculation

Destiny Material Model

- Our Fresnel term is different

$$f(l,v) = \frac{D(h)F(v \cdot \mathrm{n})G(l,v,h)}{4(n \cdot l)(\mathrm{n} \cdot \mathrm{v})}$$

  - We use $\mathrm{F}(v \cdot n)$ not $\mathrm{F}(v \cdot h)$ or $\mathrm{F}(\mathrm{l} \cdot h)$
  - $\mathrm{F}(\mathrm{l} \cdot h)$ is a per-light, per-fragment calculation
  - $\mathrm{F}(v \cdot n)$ drops the light dependency

Destiny Material Model

- Our Fresnel term is different

$$f(l,v) = \frac{D(h)F(v \cdot \mathrm{n})G(l,v,h)}{4(n \cdot l)(\mathrm{n} \cdot \mathrm{v})}$$

  - We use $\mathrm{F}(v \cdot n)$ not $\mathrm{F}(v \cdot h)$ or $\mathrm{F}(\mathrm{l} \cdot h)$
  - $\mathrm{F}(\mathrm{l} \cdot h)$ is a per-light, per-fragment calculation
  - $\mathrm{F}(v \cdot n)$ drops the light dependency
  - Equivalent when specular is a perfect mirror

Destiny Material Model: Fresnel

- Our Fresnel term is different

$$f(l,v) = \frac{D(h)F(v\cdot \mathrm{n})G(l,v,h)}{4(n\cdot l)(\mathrm{n}\cdot \mathrm{v})}$$

  - We use $\mathrm{F}(v\cdot n)$ not $\mathrm{F}(v\cdot h)$ or $\mathrm{F}(\mathrm{l}\cdot h)$
  - $\mathrm{F}(\mathrm{l}\cdot h)$ is a per-light, per-fragment calculation
  - $\mathrm{F}(v\cdot n)$ drops the light dependency
  - Equivalent when specular is a perfect mirror
  - NOT equivalent for rough specularity
    - Not ‘physical’ either -- breaks reciprocity

Destiny Material Model: Fresnel

$$f(l, v) = \frac{D(h)F(v \cdot \mathrm{n})G(l, v, h)}{4(n \cdot l)(\mathrm{n} \cdot \mathrm{v})}$$

- Table-based Fresnel:
- $F(v \cdot n) = Combine(\ LUT(n \cdot v, tint\_id),\ \ RefColor)$

Destiny Material Model: Fresnel

$$f(l, v) = \frac{D(h)F(v \cdot \mathrm{n})G(l, v, h)}{4(n \cdot l)(\mathrm{n} \cdot \mathrm{v})}$$

- Table-based Fresnel:
- $F(v \cdot n) = Combine(\, LUT(n \cdot v, tint\_id),\ \ RefColor)$
    - *Reference color* is the per-pixel color produced by the G-buffer shaders

РФБА

Destiny Material Model: Fresnel

$$f(l,v) = \frac{D(h)F(v \cdot \mathrm{n})G(l,v,h)}{4(n \cdot l)(\mathrm{n} \cdot \mathrm{v})}$$

- Table-based Fresnel:
- $F(v \cdot n) = Combine(\ LUT(n \cdot v, tint\_id),\ \ RefColor)$
    - *Reference color* is the per-pixel color produced by the G-buffer shaders

Destiny Material Model: Fresnel

$$f(l,v) = \frac{D(h)F(v \cdot \mathrm{n})G(l,v,h)}{4(n \cdot l)(\mathrm{n} \cdot \mathrm{v})}$$

- Table-based Fresnel:
- $F(v \cdot n) = Combine(\,LUT(n \cdot v, tint\_id),\ \ RefColor)$
  - Combine function (during shading) mixes the reference color with LUT-provided colors
  - Combine is different for specular and diffuse BRDFs
  - LUT table is not limited to ‘standard’ Fresnel approximations

Destiny Material Model: Distribution

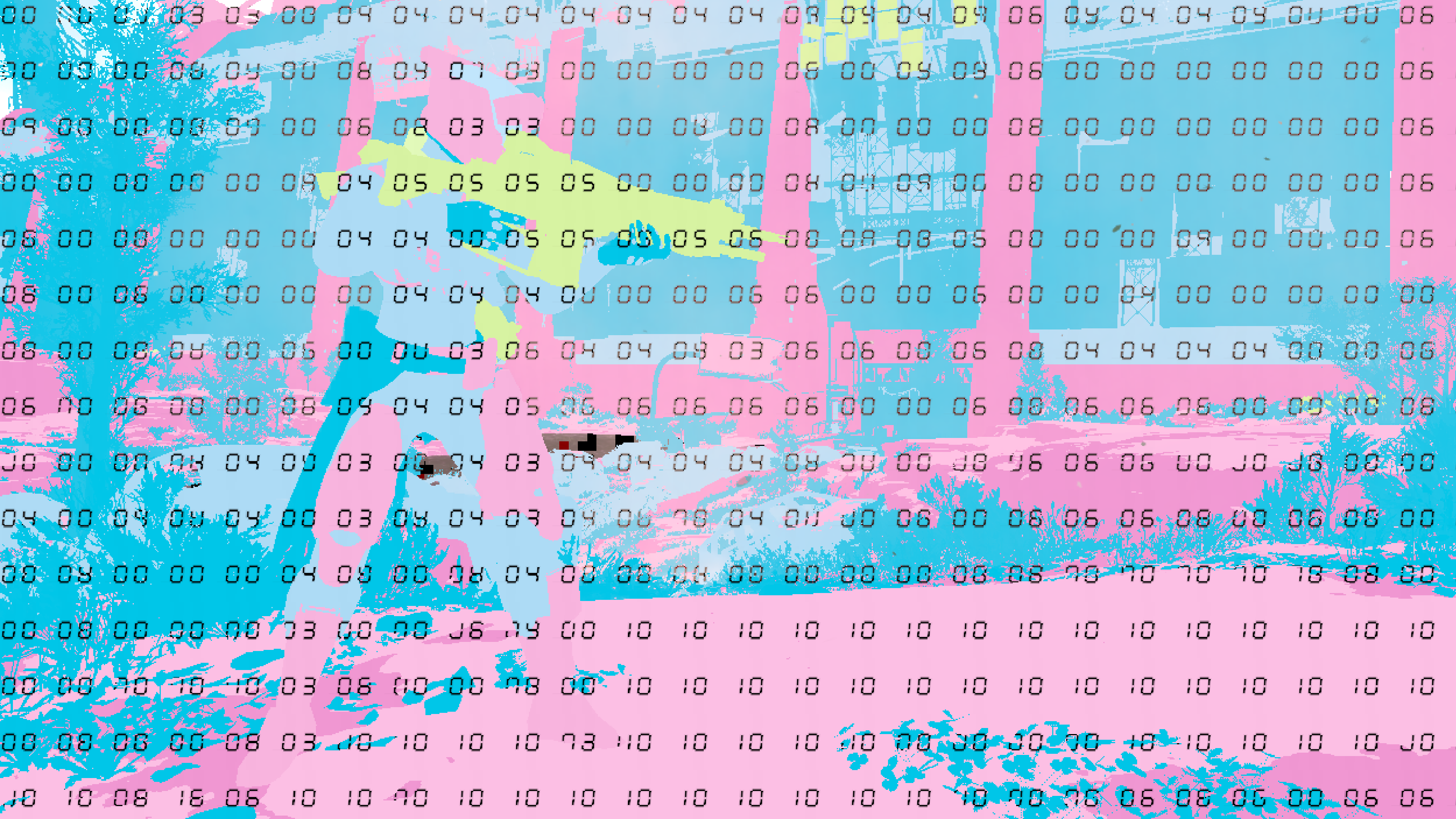
$$f(l,v)=\frac{D(h)F(v\cdot \mathrm{n})G(l,v,h)}{4(n\cdot l)(\mathrm{n}\cdot \mathrm{v})}$$

- Table-based distribution:
- $D(h)=LUT(n\cdot h, roughness, lobe\_id)$
  - Artist-supplied distributions for (roughness = 0) per material ID
    - Spherically blurred to generate roughness variations
    - Normalized to be energy preserving with white Fresnel

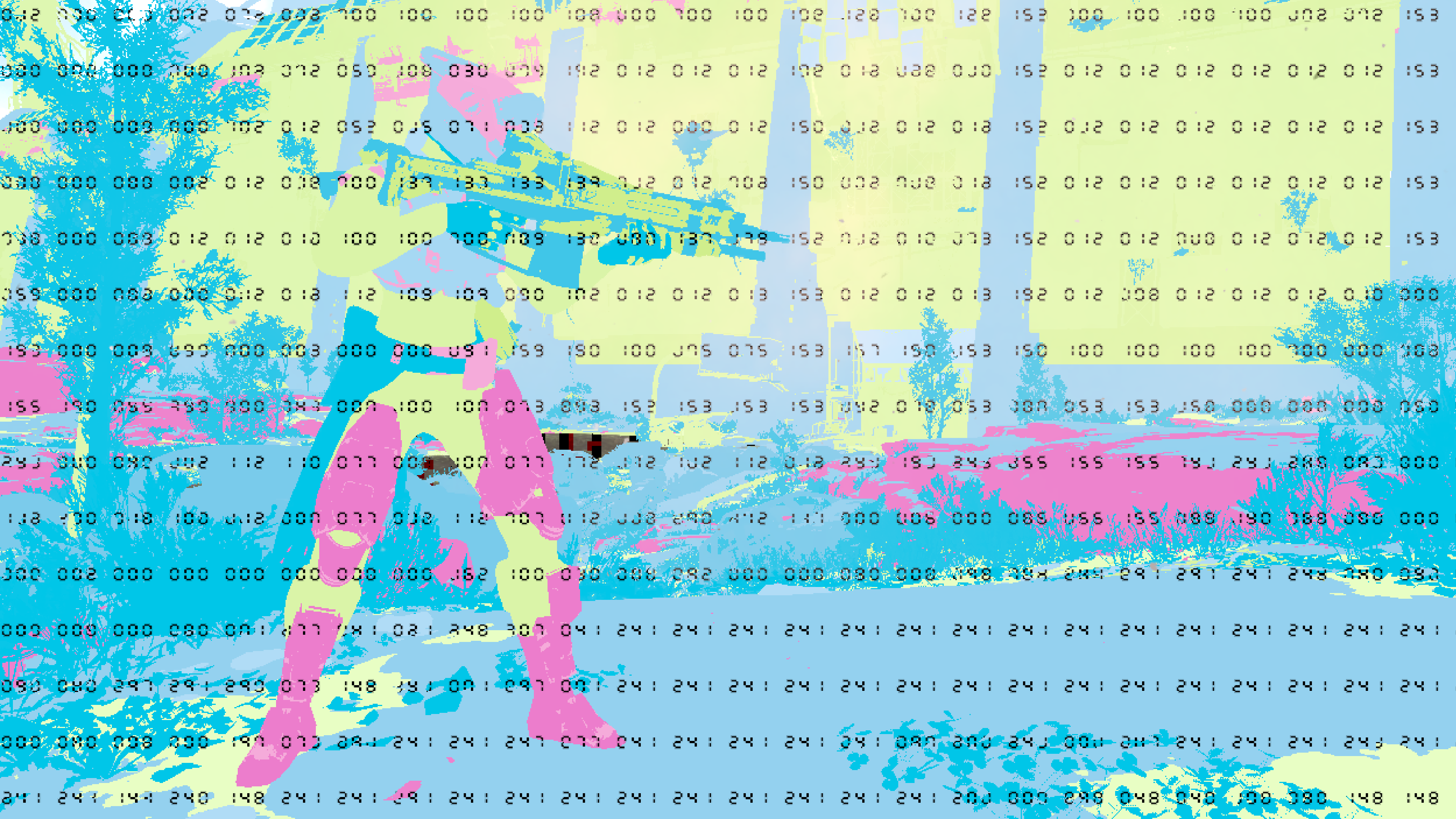
Destiny Material Model: Distribution

$$f(l,v) = \frac{D(h)F(v\cdot \mathrm{n})G(l,v,h)}{4(n\cdot l)(\mathrm{n}\cdot \mathrm{v})}$$

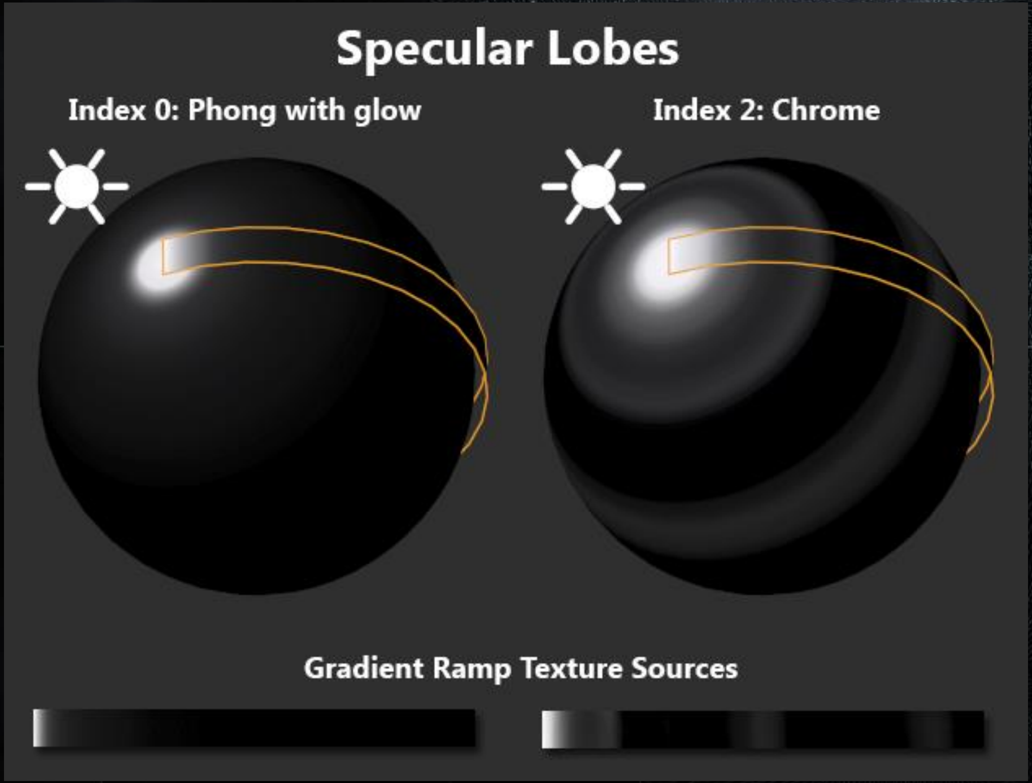
- Table-based distribution:
- $D(h) = LUT(n\cdot h, roughness, lobe\_id)$
  - We store diffuse and specular $D(h)$ in the same LUT
    - ‘Standard’ Lambertian diffuse is: $D(h) = \frac{1.0}{\pi}$
  - But we don’t limit the artists to this



```
00 00 04 03 03 00 04 04 04 04 04 04 04 04 08 09 04 09 06 09 04 04 09 04 00 06
00 03 00 00 04 00 08 04 07 03 00 00 00 00 08 00 04 03 06 00 00 00 00 00 00 06
09 00 00 00 03 00 06 08 03 03 00 00 00 00 08 00 00 00 08 00 00 00 00 00 00 06
00 00 00 00 00 04 04 05 05 05 05 00 00 00 08 04 09 00 08 00 00 00 00 00 00 06
06 00 00 00 00 00 04 04 00 05 05 00 05 05 00 08 03 05 00 00 00 07 00 00 00 06
06 00 08 00 00 00 00 04 04 04 00 00 00 06 06 00 00 06 00 00 04 00 00 00 00 00
08 00 00 00 00 05 00 00 03 06 04 04 08 03 06 06 00 05 08 04 04 04 04 00 00 06
06 00 06 08 00 08 09 04 04 05 06 06 06 06 06 00 00 06 00 06 06 06 00 00 00 08
00 00 00 04 04 00 03 00 04 03 04 04 04 04 08 00 00 08 06 06 06 00 00 08 00 00
04 00 04 04 04 00 03 04 04 03 04 06 08 04 04 00 06 00 06 06 06 08 00 06 08 00
00 03 00 00 00 04 00 00 08 04 00 08 04 00 00 00 00 08 06 10 10 10 10 10 08 00
00 08 00 00 00 03 00 00 06 09 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
00 00 10 10 10 03 06 10 00 08 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
08 08 08 00 08 03 10 10 10 10 13 10 10 10 10 10 10 00 00 10 10 10 10 10 10 10
10 10 08 16 06 10 10 10 10 10 10 10 10 10 10 10 10 10 10 16 06 08 06 00 06 06
```


012 000 000 072 074 008 100 100 100 100 108 000 100 100 108 120 100 128 153 100 100 100 100 008 072 153
080 080 000 100 108 072 050 108 030 079 192 012 012 012 192 018 080 010 152 012 012 012 012 012 012 153
000 000 003 000 102 012 052 015 071 008 112 012 000 012 150 012 012 018 152 012 012 012 012 012 012 153
030 000 080 002 012 018 100 137 133 133 139 012 012 108 150 002 008 018 152 012 012 012 012 012 012 153
038 000 083 012 012 010 100 100 100 089 130 080 137 179 152 012 010 073 152 012 012 000 012 072 012 153
059 000 080 000 012 018 112 109 109 090 102 012 012 013 153 012 012 013 192 012 008 012 012 012 010 000
153 000 008 090 000 003 000 000 091 159 190 100 075 075 153 157 190 153 150 100 100 100 100 100 000 008
155 190 055 790 000 047 000 100 100 073 098 152 153 153 153 092 010 053 080 053 153 150 000 000 000 060
243 000 080 002 112 110 077 008 100 077 172 112 102 112 018 243 190 243 355 155 155 181 241 280 080 000
118 400 018 100 012 000 077 010 112 707 012 008 090 412 111 000 006 000 085 056 155 080 190 080 080 000
000 002 000 000 000 000 000 000 162 100 030 000 092 000 000 080 000 148 784 244 291 291 241 248 080 080
000 000 000 080 041 477 241 021 248 707 041 241 241 241 241 241 241 241 241 241 241 241 241 241 241 241
090 080 297 291 290 073 148 781 001 297 001 241 241 241 241 241 241 241 241 241 241 241 241 241 241 241
000 000 008 000 191 073 241 241 241 247 073 241 241 241 241 241 080 200 242 000 247 241 241 241 243 241
241 241 144 240 148 241 241 241 241 241 241 241 241 241 241 241 200 000 298 048 040 100 080 148 148

Gear and Dye Specular Response



Destiny Material Model: Geometry Terms

$$f(l,v) = \frac{D(h)F(v \cdot \mathrm{n})G(l,v,h)}{4(n \cdot l)(\mathrm{n} \cdot \mathrm{v})}$$

Diffuse Geometry term:

- ‘Implicit’ geometry term:

$$G(L,V,H) = 4.0 * (n \ \cdot l) * (n \cdot v)$$

- Note: the Microfacet denominator cancels with parts of G()

Destiny Material Model: Geometry Terms

$$f(l,v) = \frac{D(h)F(v \cdot \mathrm{n})G(l,v,h)}{4(n \cdot l)(\mathrm{n} \cdot \mathrm{v})}$$

Specular Geometry term:

- A modified Kelemin-Szirmay-Kalos approximation:

$$G(L,V,H) = 4.0 * \frac{(n \ \cdot l) * (n \cdot v)}{0.3 + (l + v)^2}$$

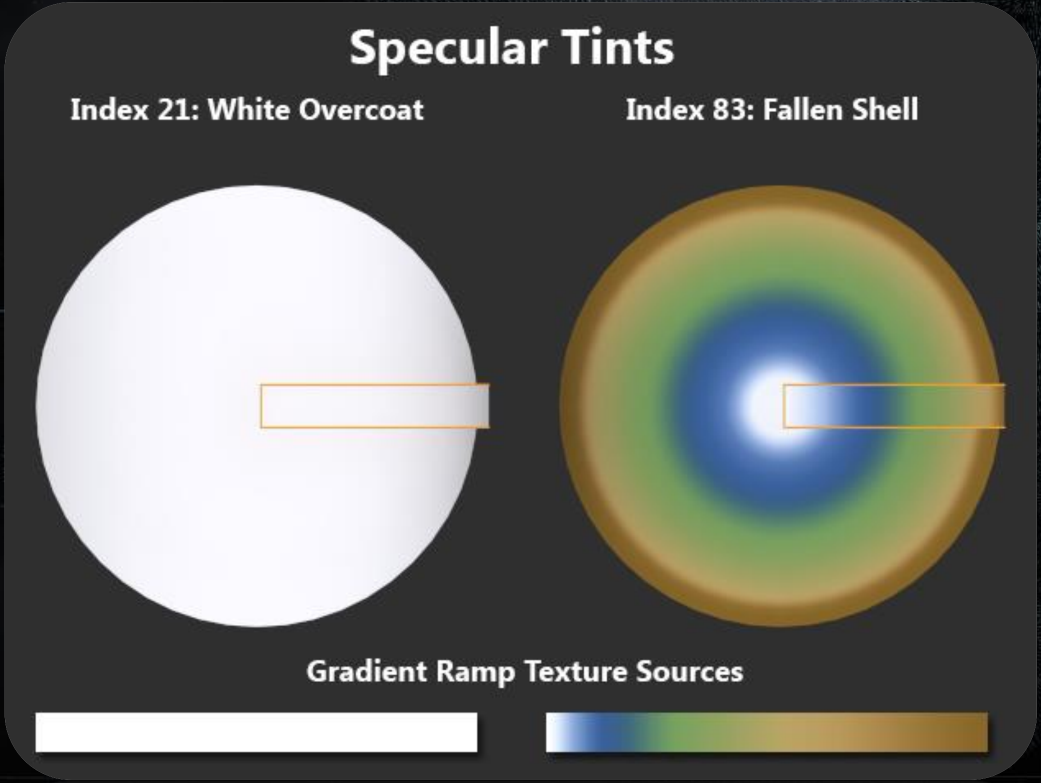
Destiny Material Model

- Initially energy-preserving
  - By default we normalize D() and F() to ensure this

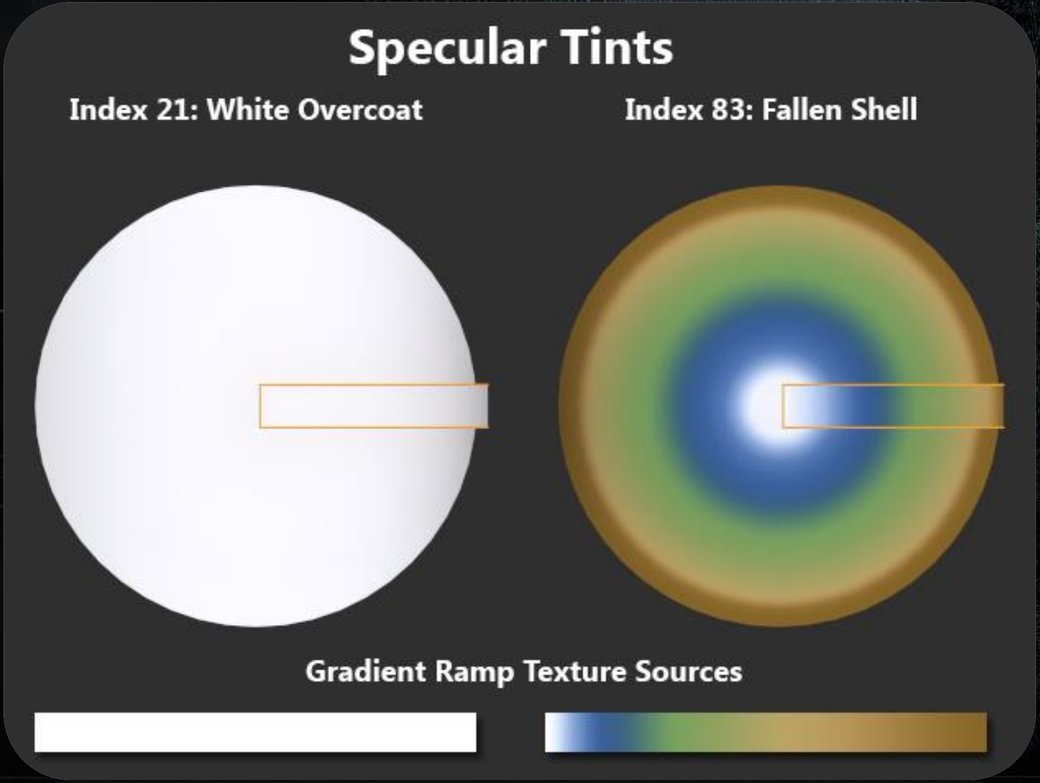
Destiny Material Model

- Artists can override to modify the results beyond that
- Use spec roughness to improve per-pixel specular variation

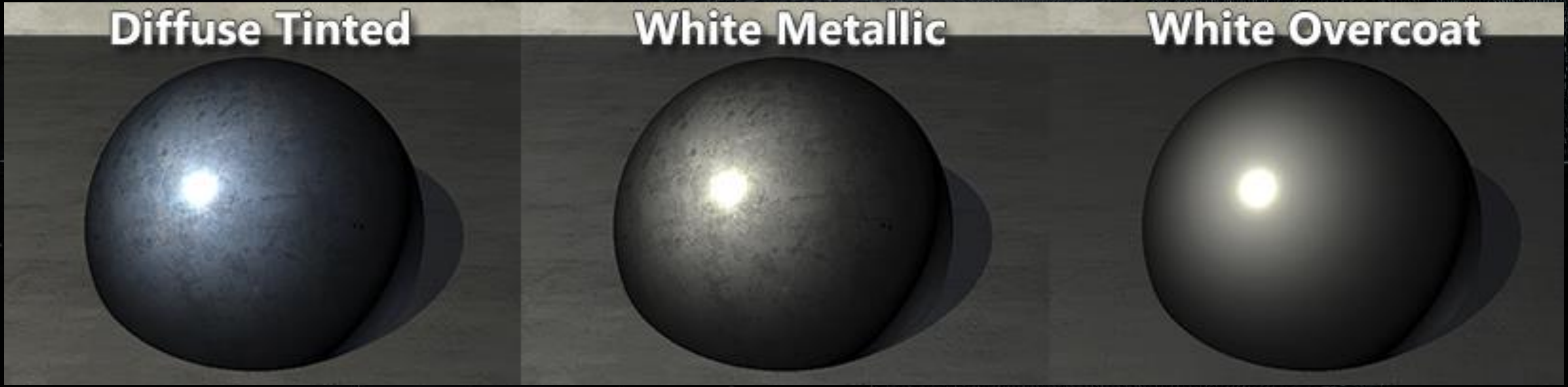
Gear and Dye Spec Tint



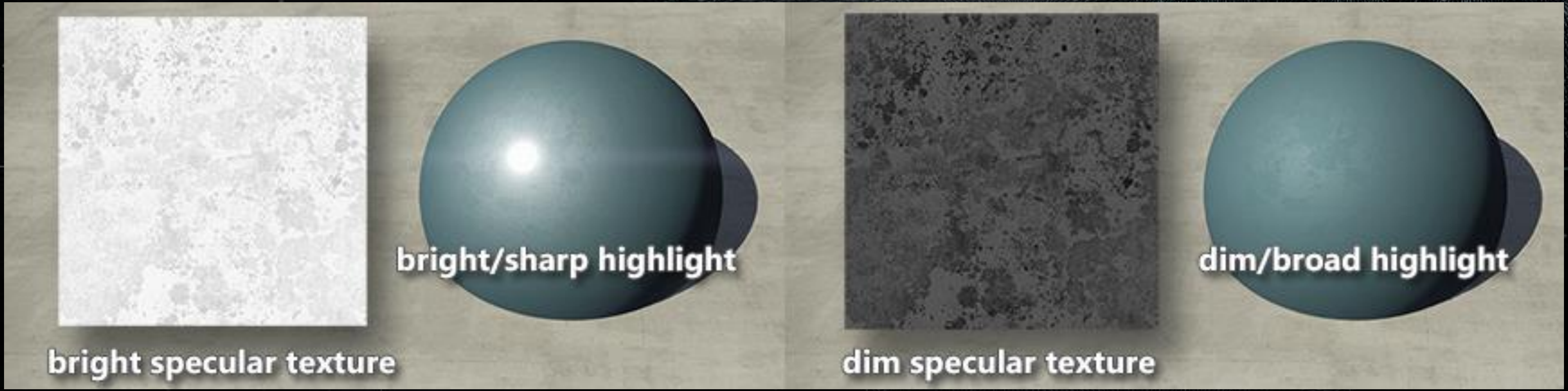
Gear and Dye Spec Tint



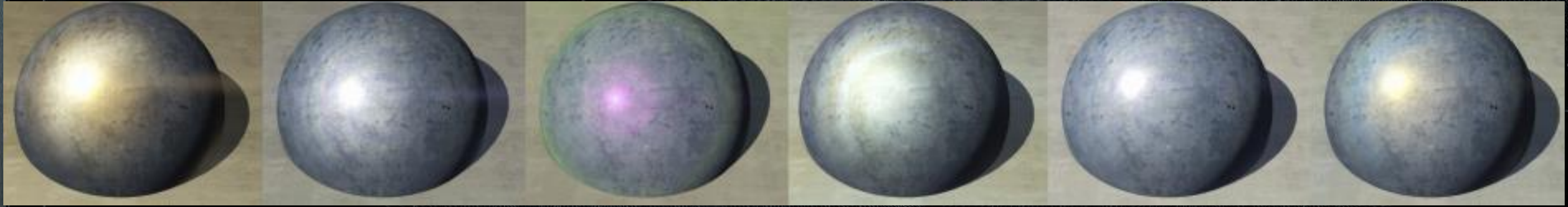
Gear and Dye Specular Response



Gear and Dye Specular Response

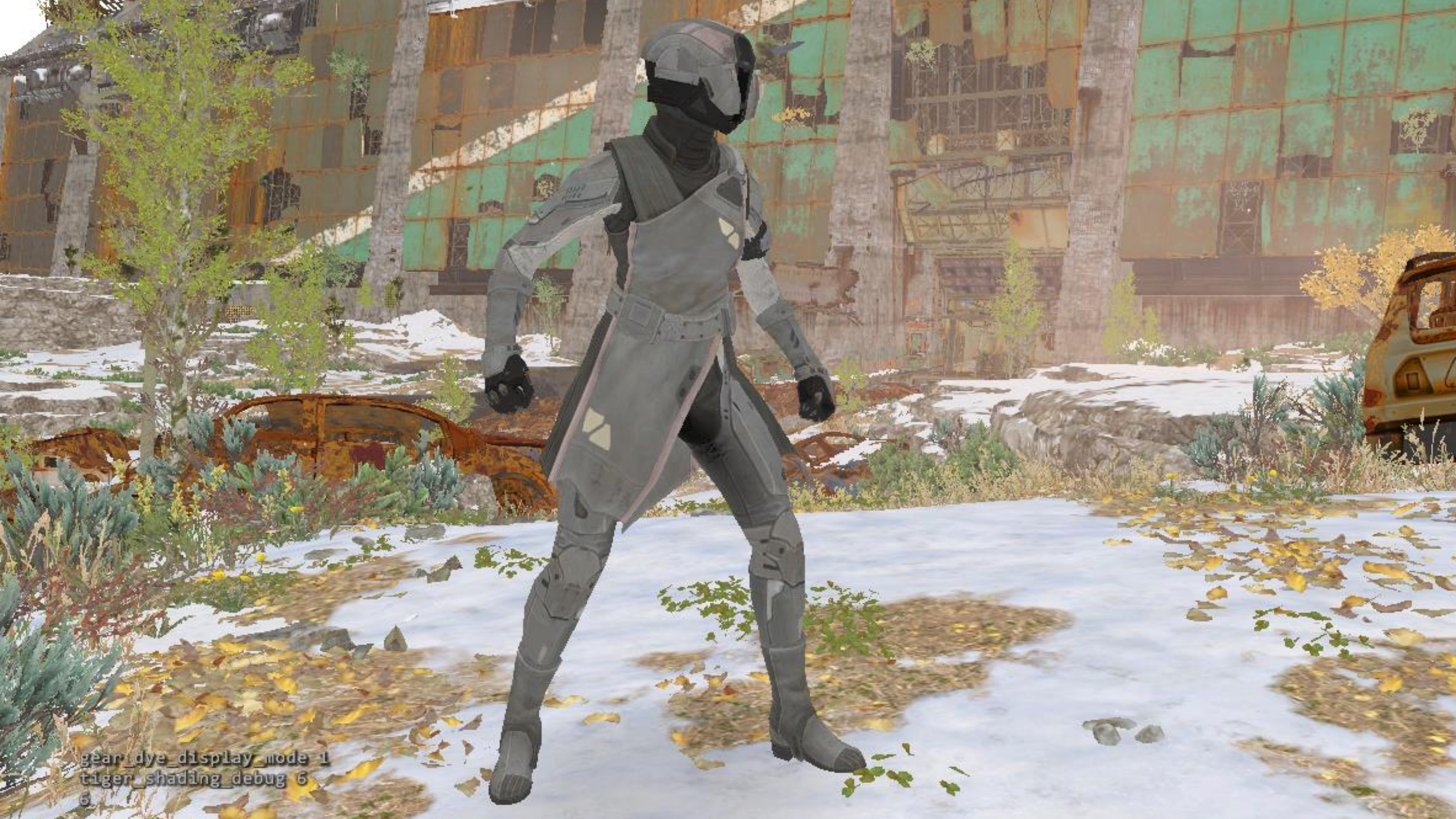


Gear and Dye Specular Response

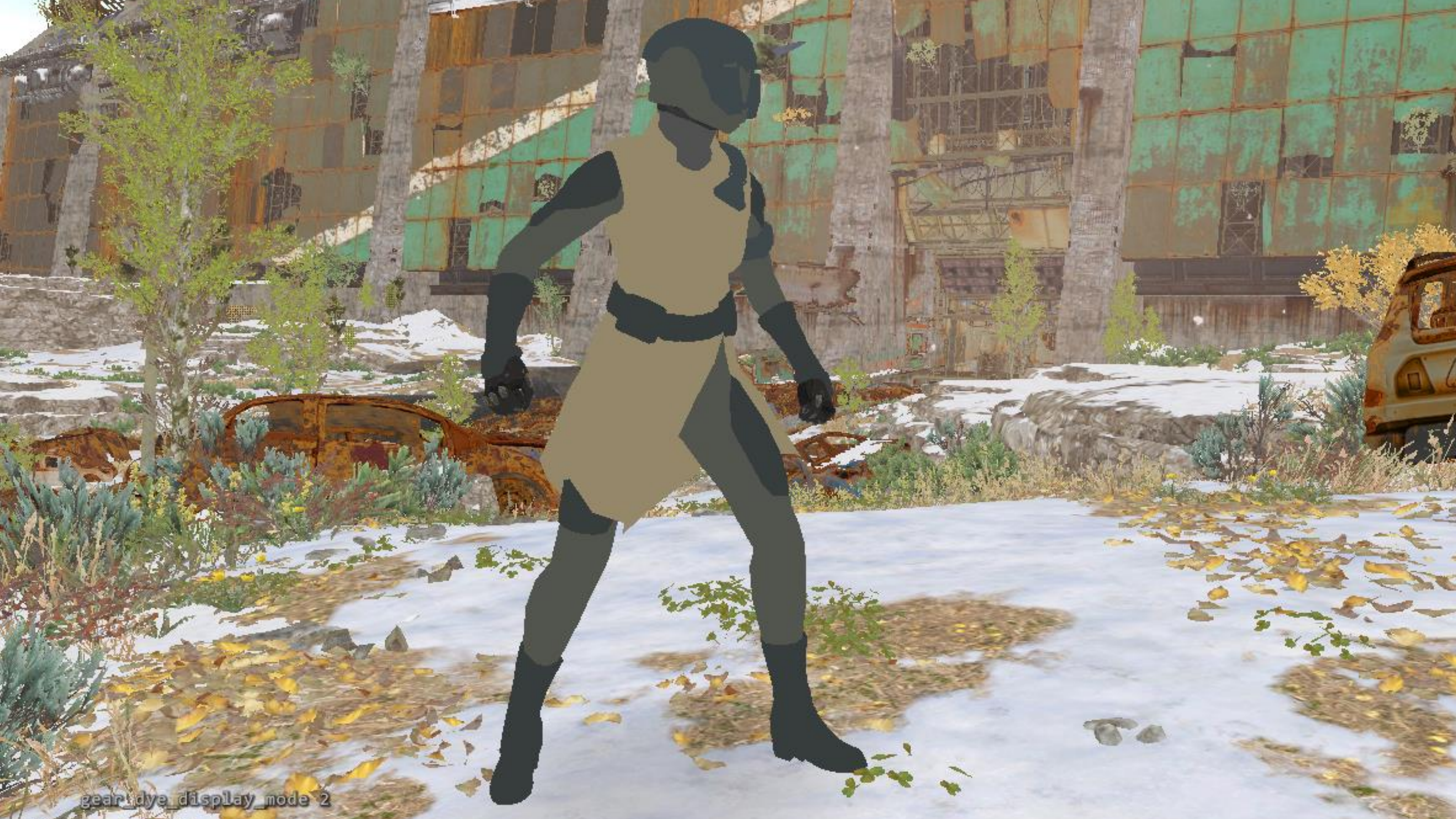


Our Solution: Gear System

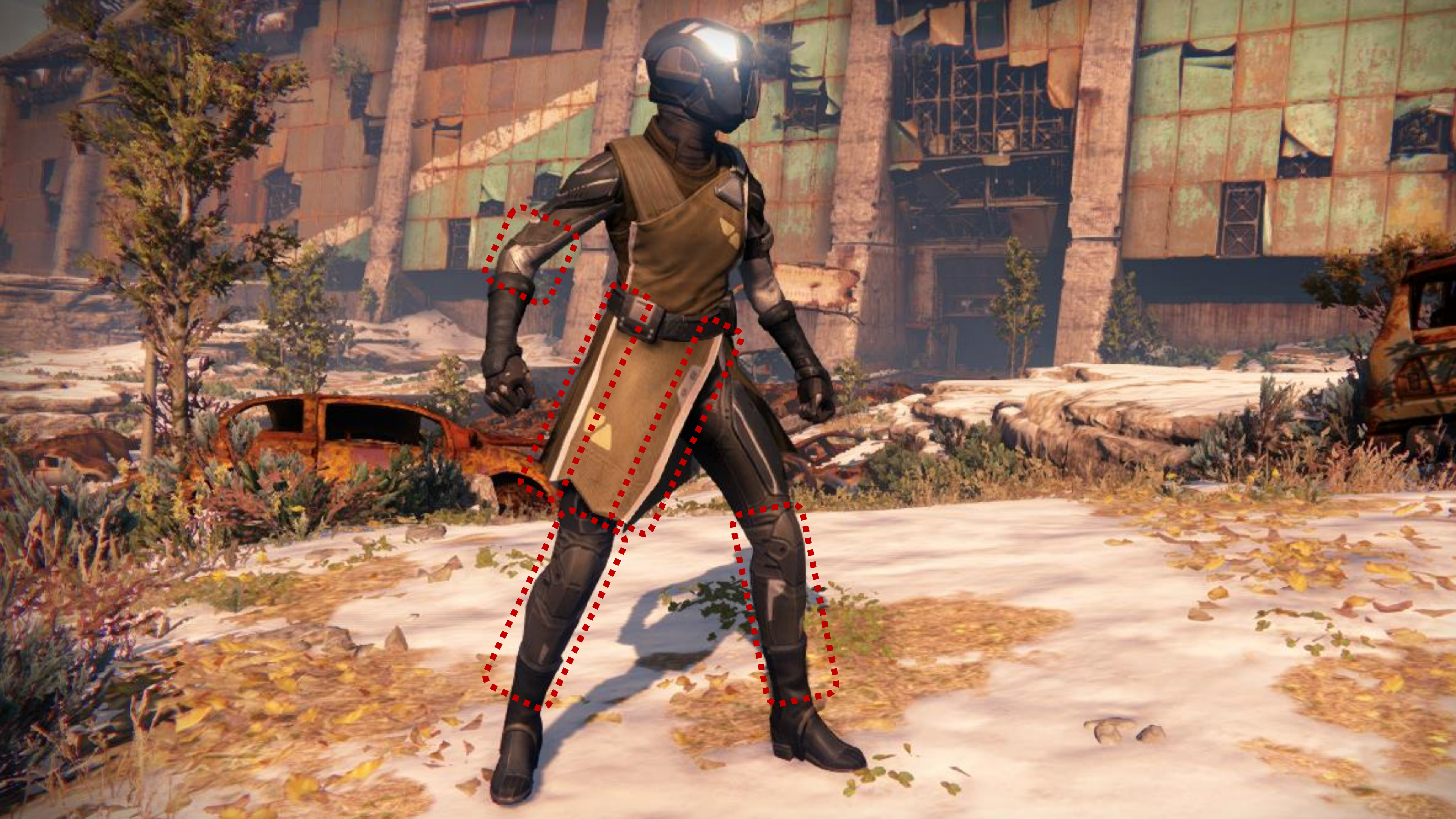
1. Gear Slots
2. Arrangements and Bits
3. Plated Textures
4. Gear Dyes
    1. Goals
    2. Tinting
    3. Material response
    4. Gearstack usage
    5. Advanced dyes and materials

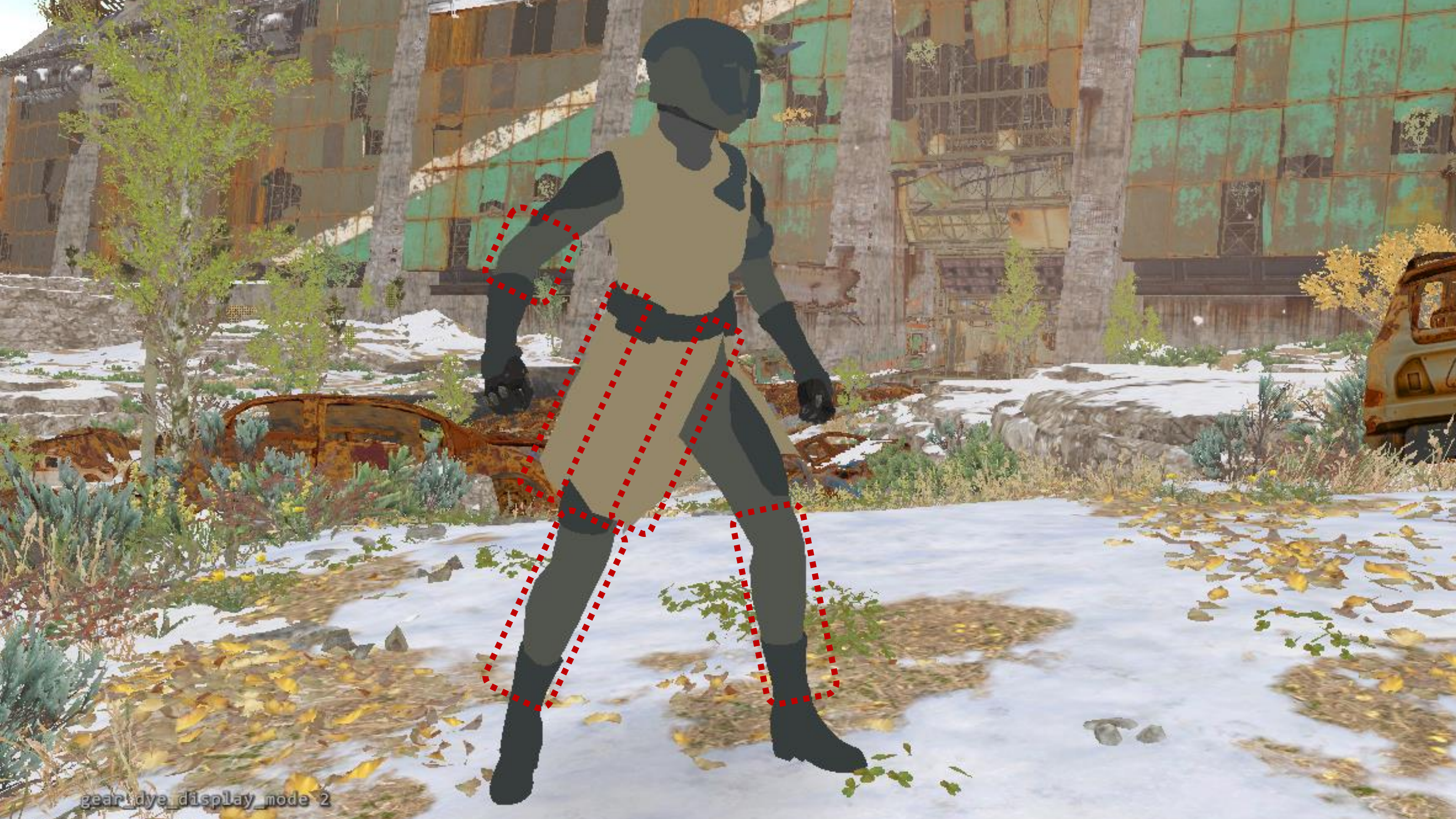


```
gear_dye_display_mode 1
tiger_shading_debug 6
6
```

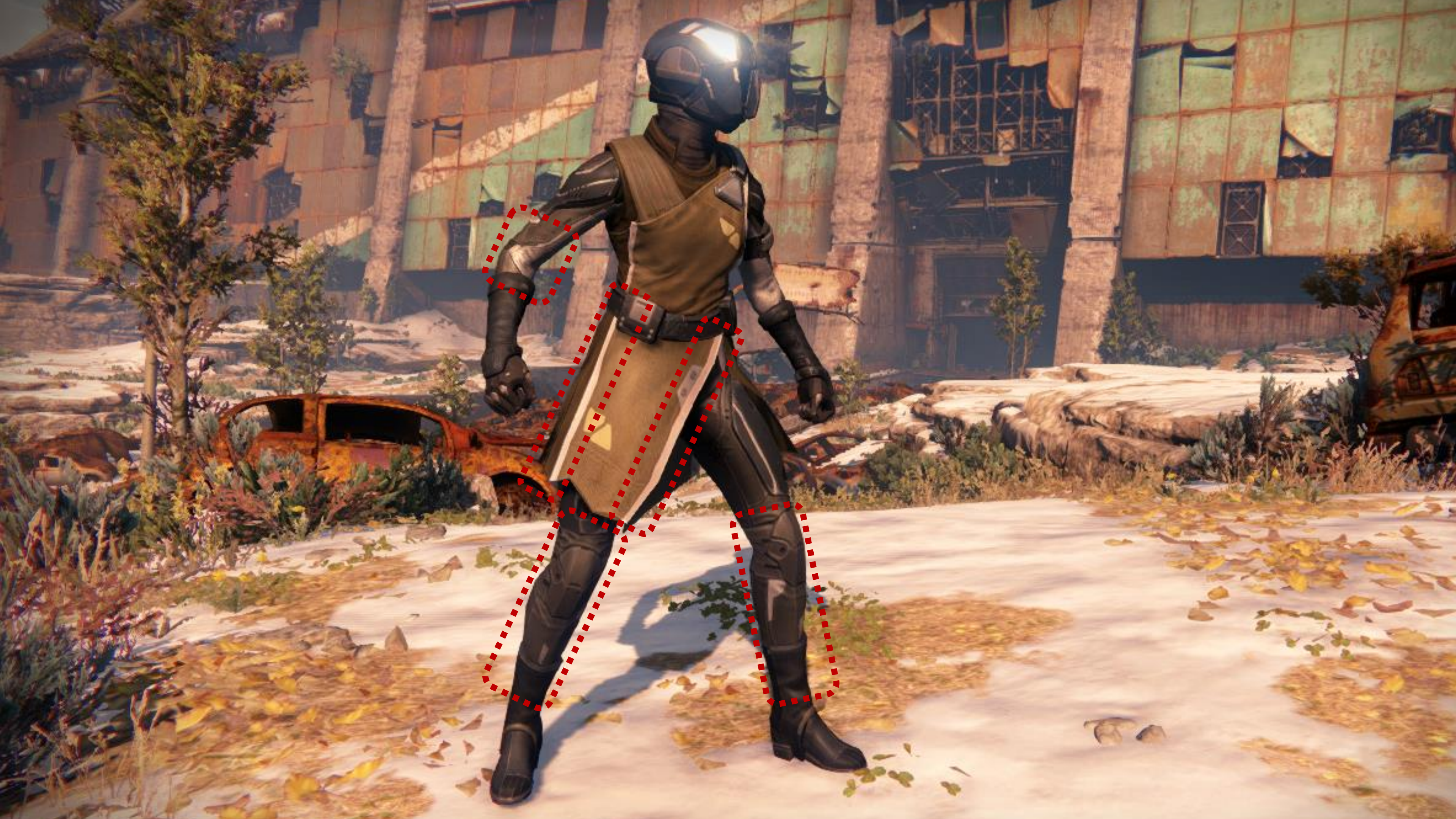


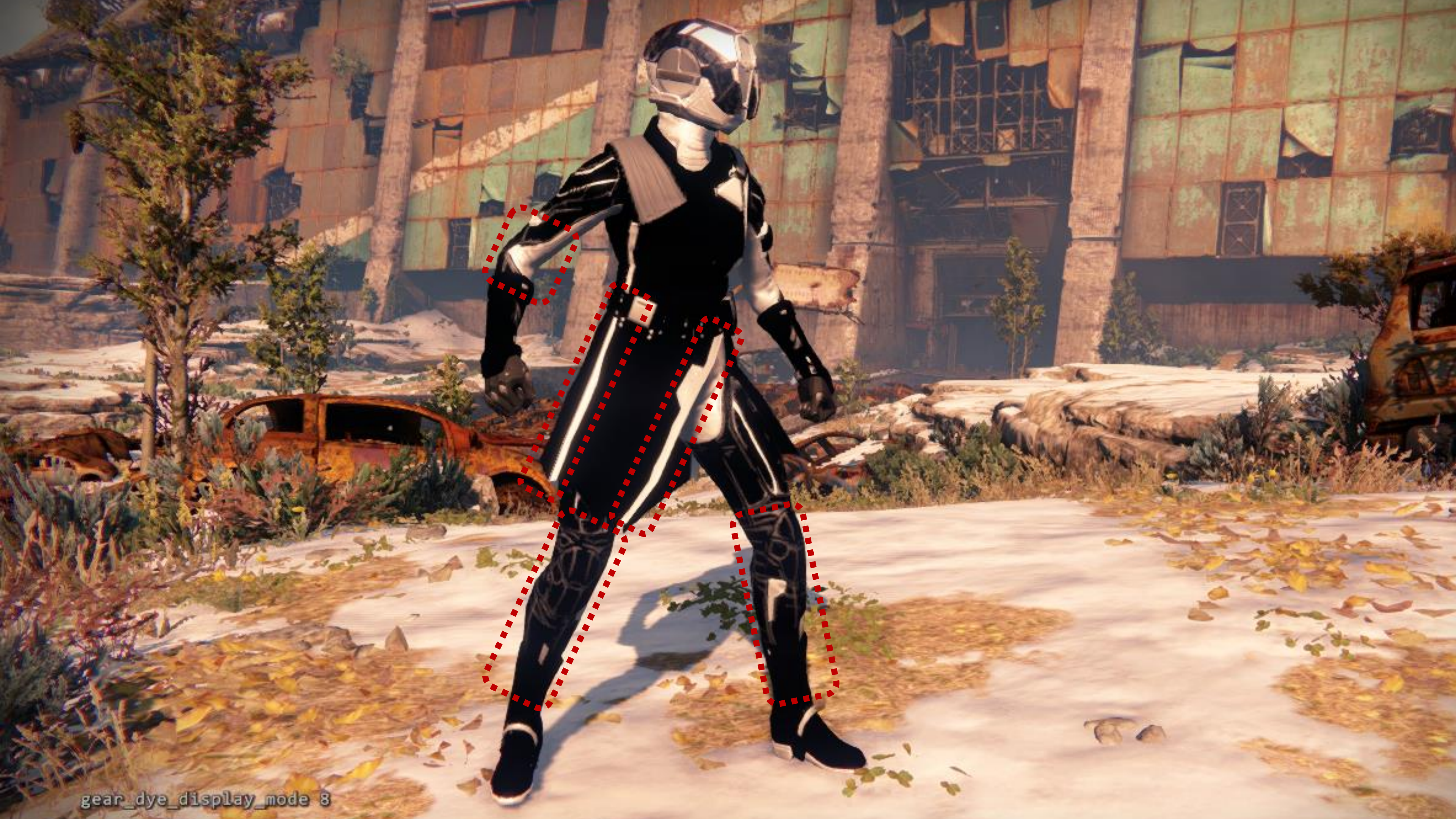
```
gear_dye_display_mode 2
```





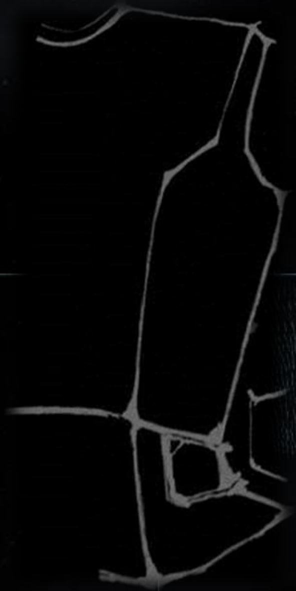
```
gear_dye_display_mode 2
```





gear_dye_display_mode 8

Scratching the Surface

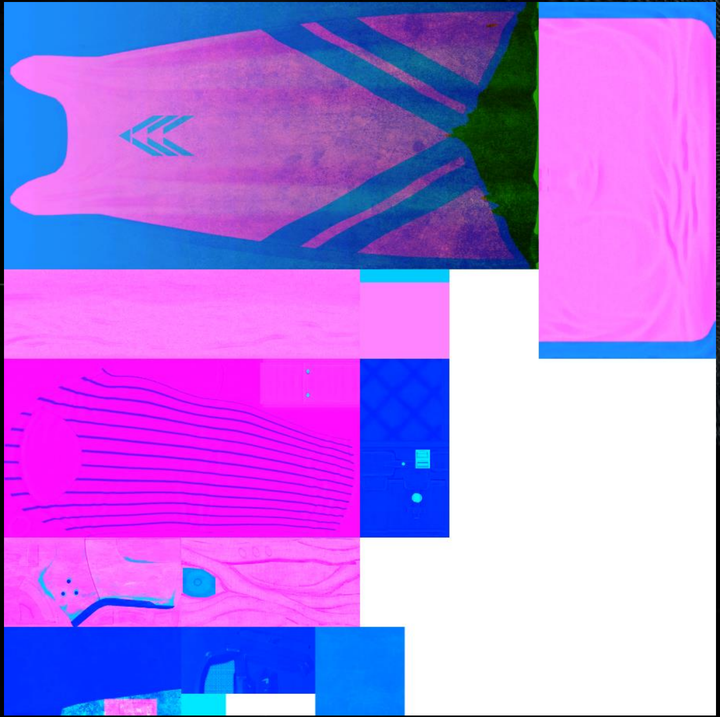


Many Visual Needs…

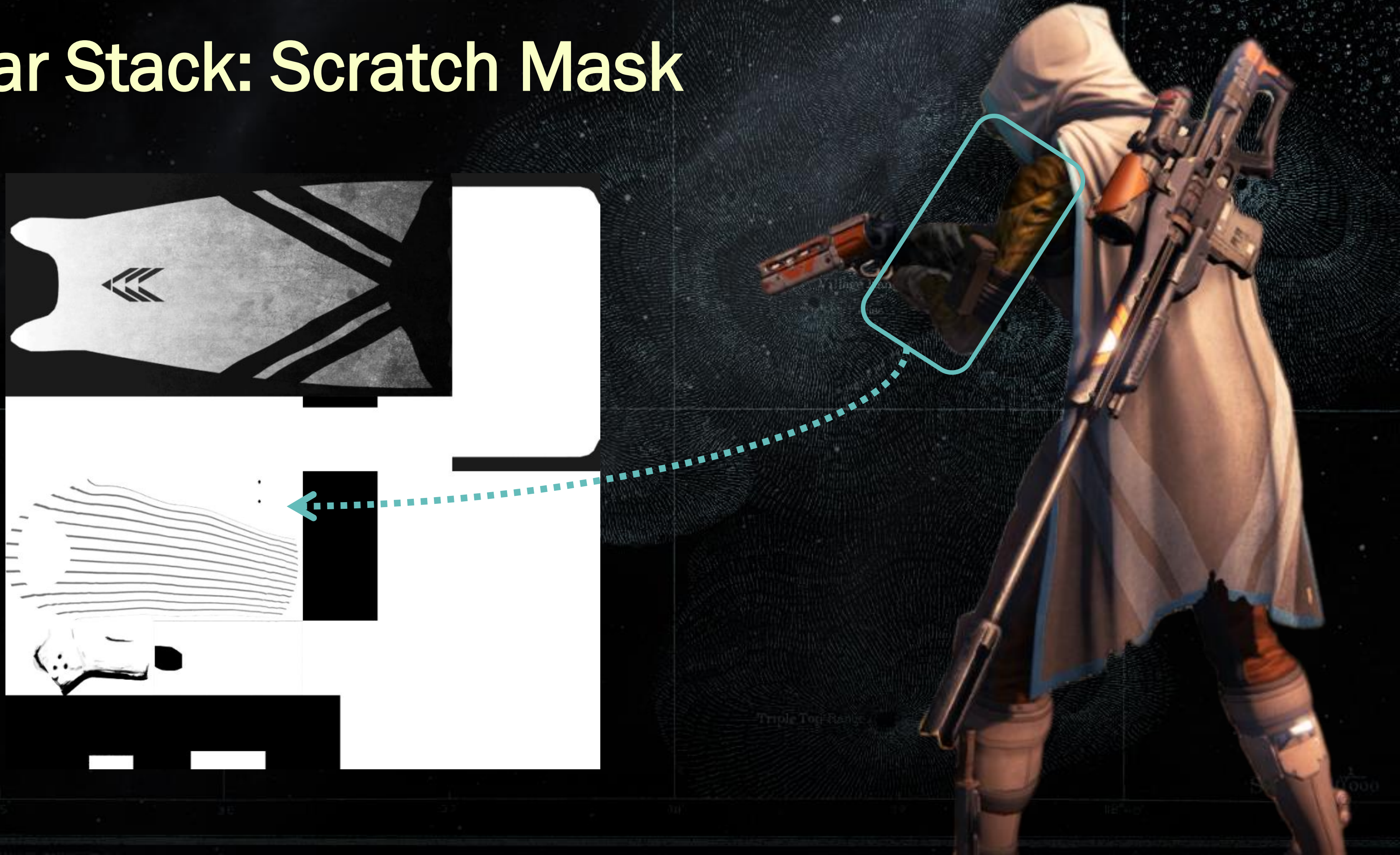
- Worn & varied materials
- Fringed, torn materials
- Varied specular response
- Transparent materials
- Subsurface scattering
- Custom masks
- …

- *Yet - constant memory footprint*

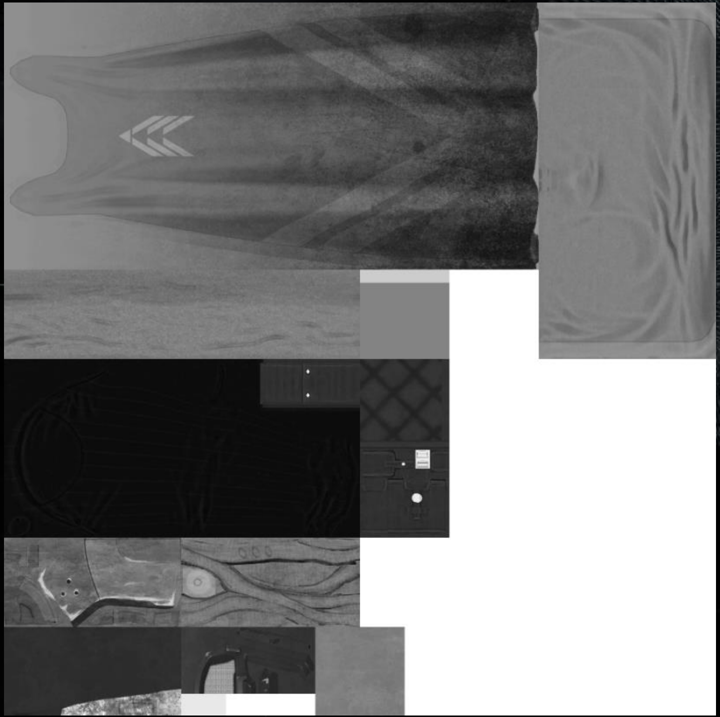
Gear Stack Texture



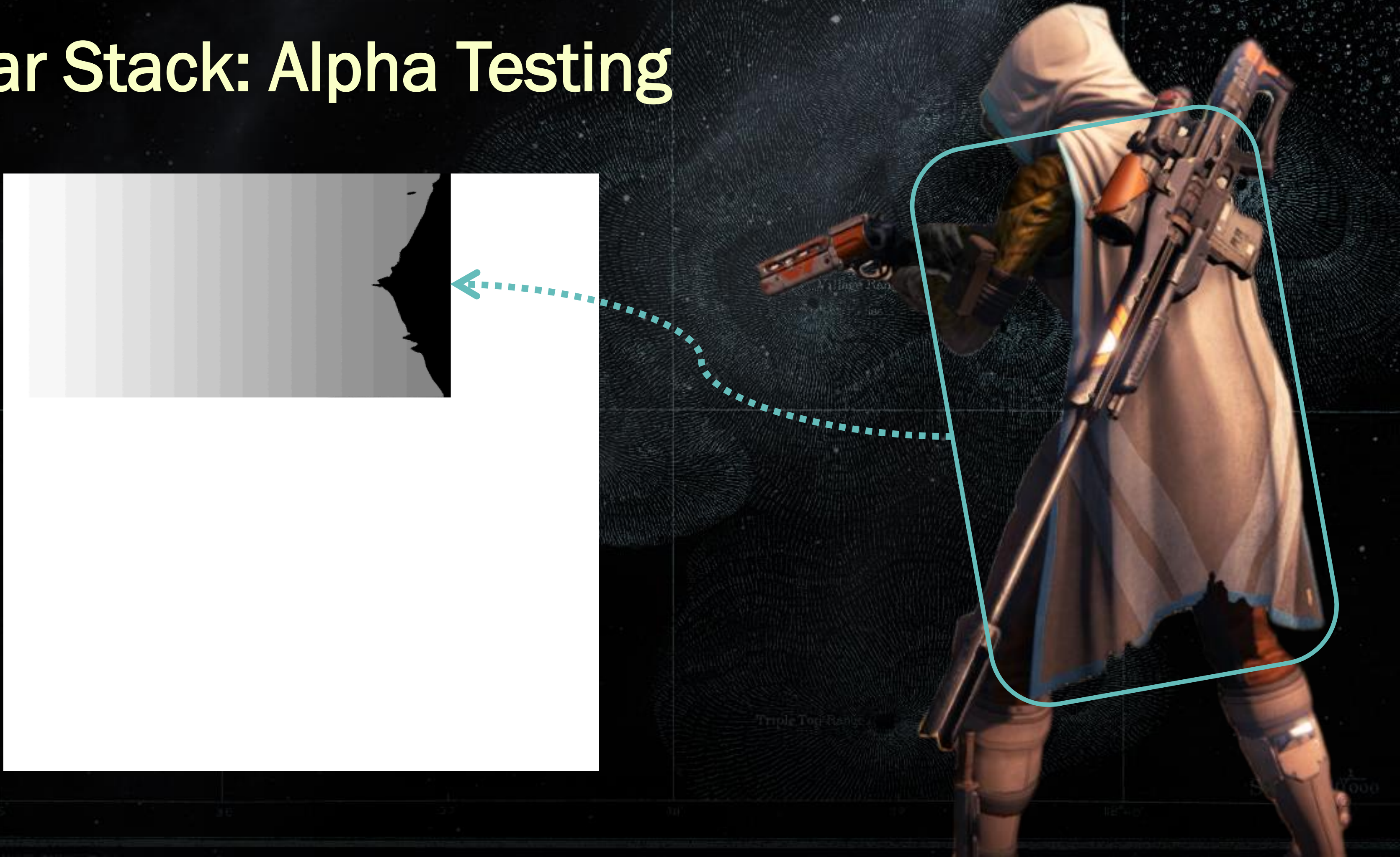
Gear Stack: Scratch Mask



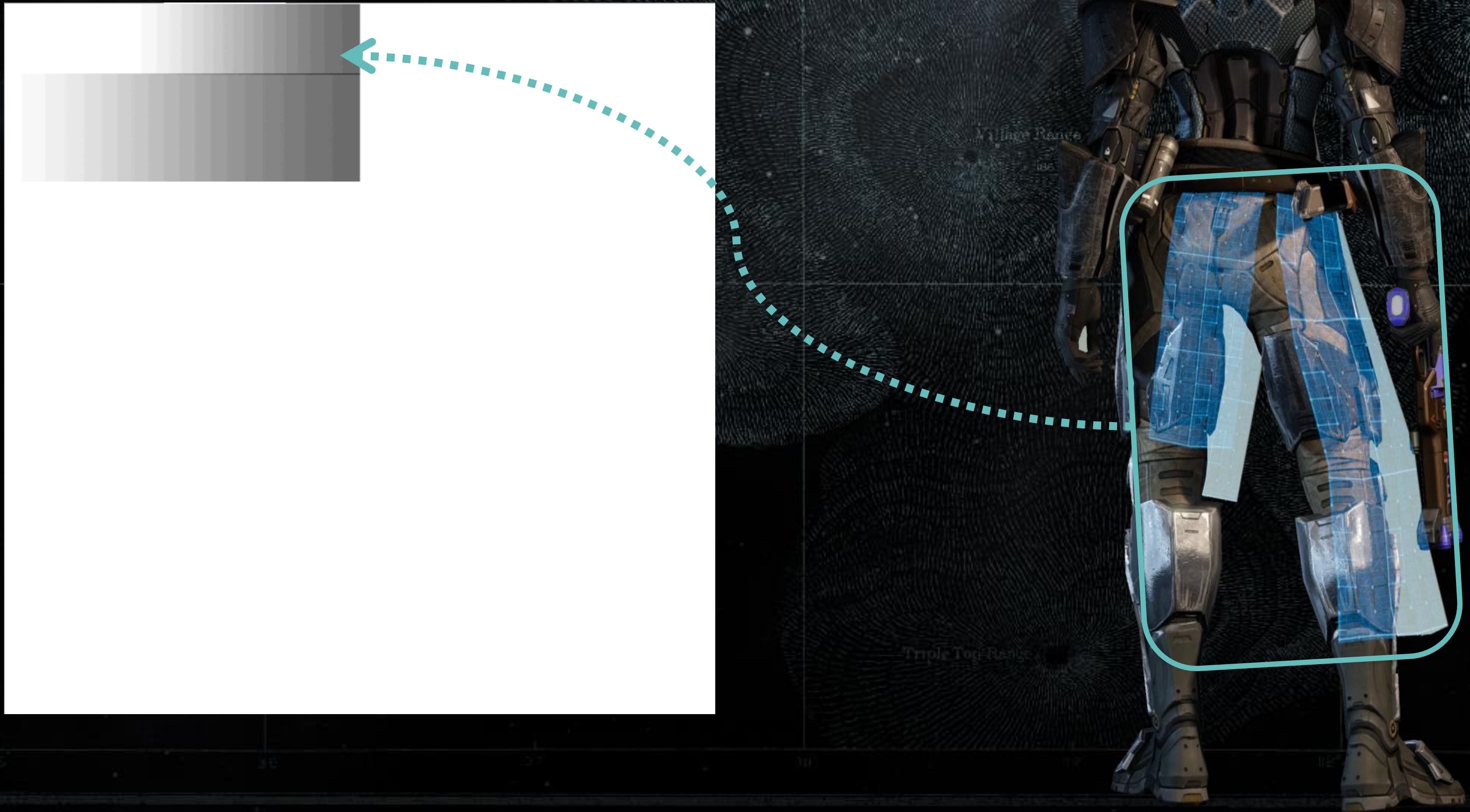
Gear Stack: Specular Roughness



Gear Stack: Alpha Testing



Gear Stack: Transparency

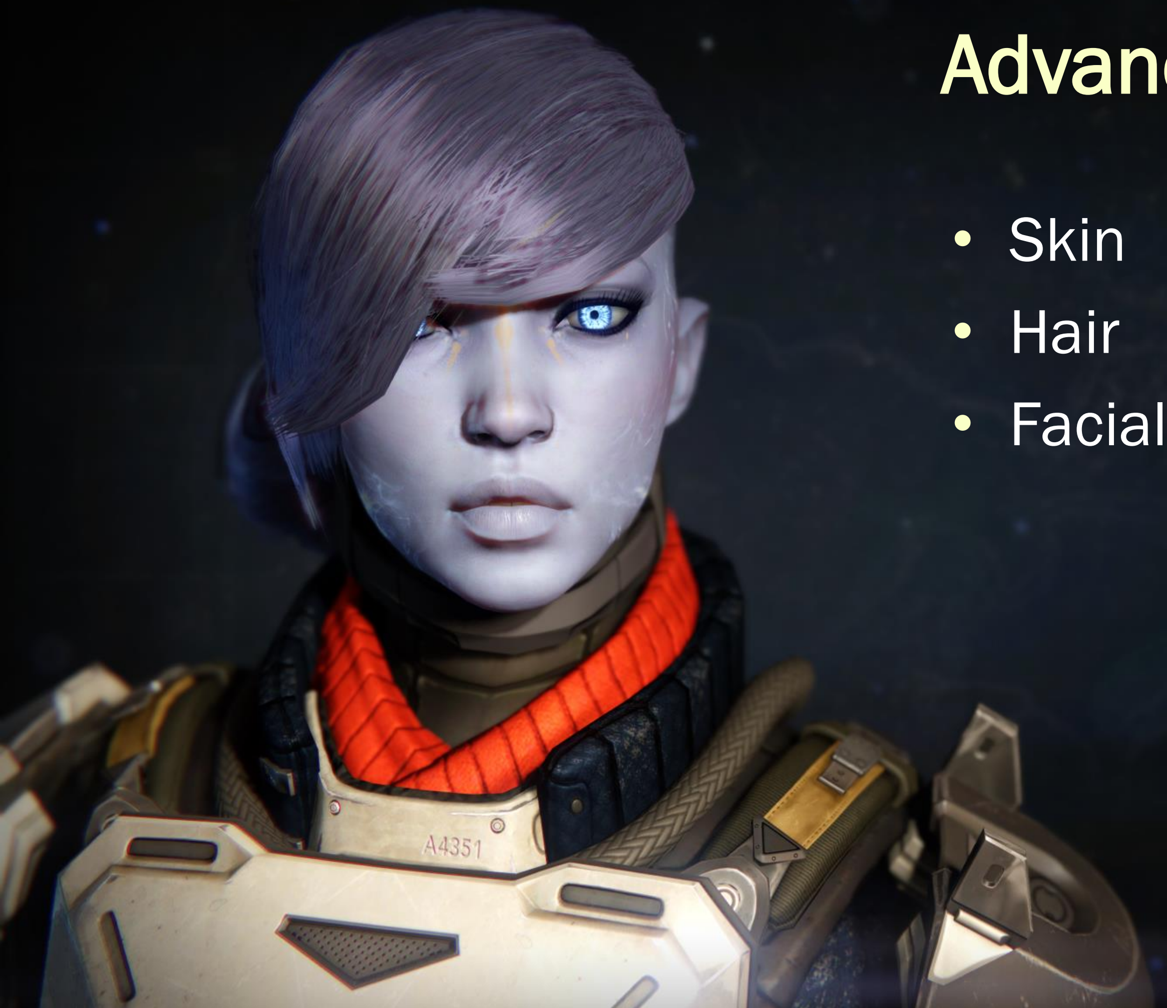


Our Solution: Gear System

1. Gear Slots
2. Arrangements and Bits
3. Plated Textures
4. Gear Dyes
   1. Goals
   2. Tinting
   3. Material response
   4. Gearstack usage
   5. Advanced dyes and materials

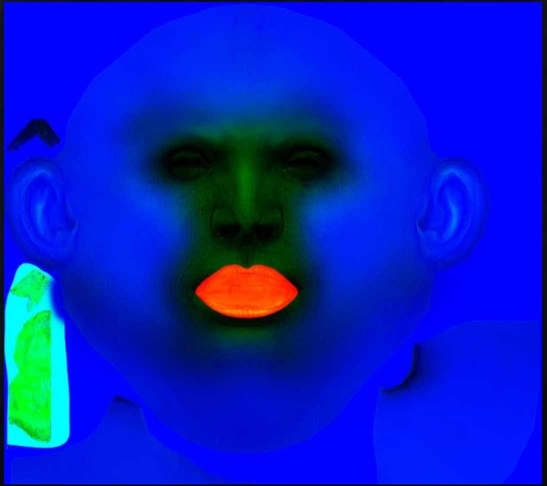
Advanced Dye Materials

- Skin
- Hair
- Facial markings



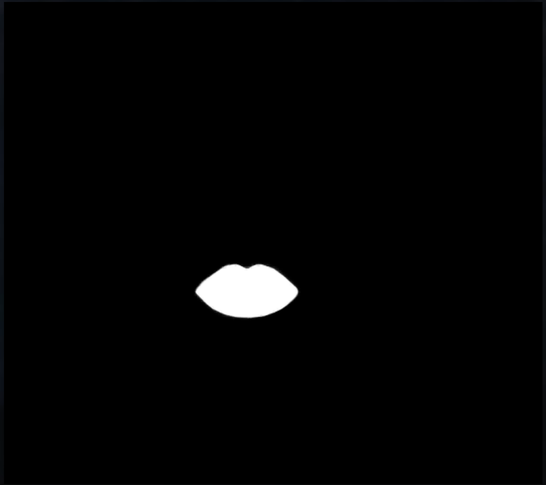
Gear Stack: Lip Customization

- Gear Skin has custom gear stack



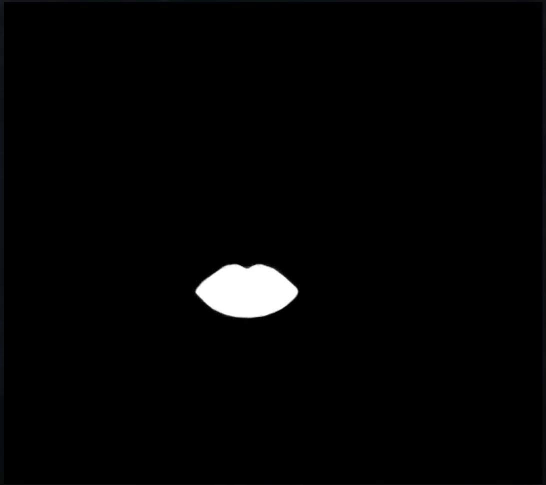
Gear Stack: Lip Customization

- Red: lips dye region



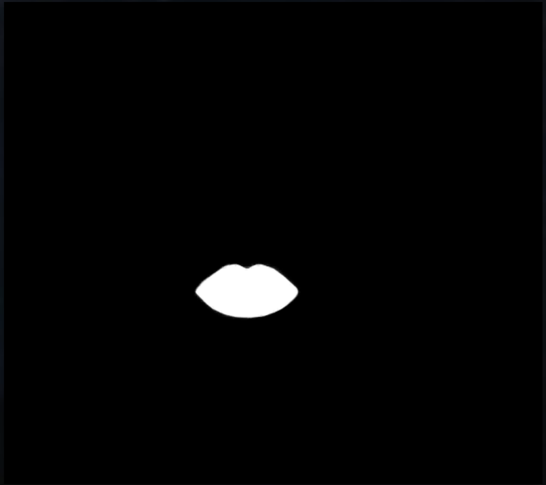
Gear Stack: Lip Customization

- Red: lips dye region



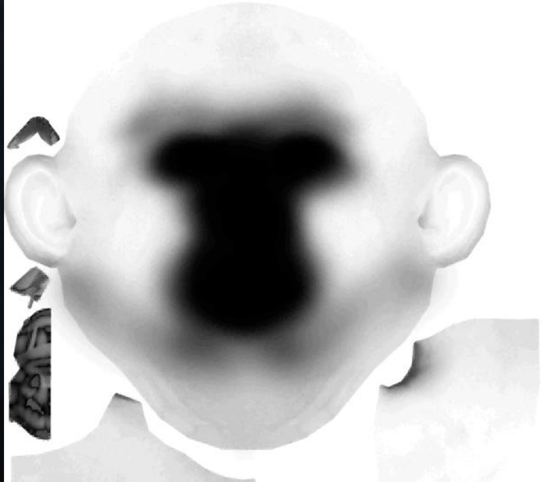
Gear Stack: Lip Customization

- Red: lips dye region



Gear Stack: Subsurface Scattering

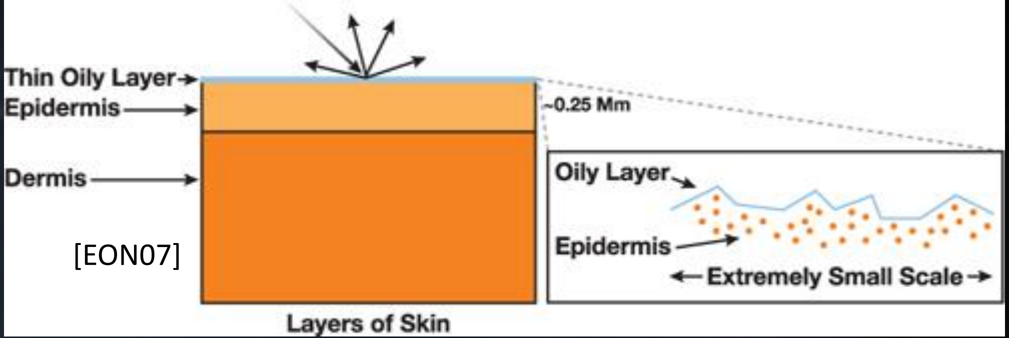
- Per-face subsurface scattering control



Gear Stack: Subsurface Scattering

- Stream for each face independently
- Account for bone structure
- Awoken / Human differences

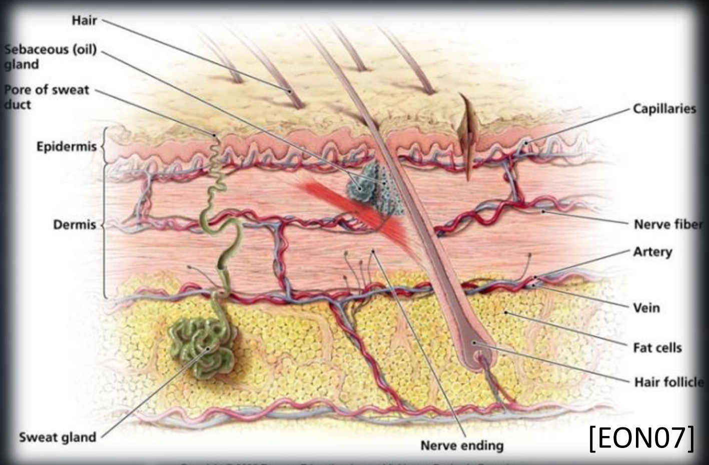
Subsurface Scattering



[EON07]

BRDF

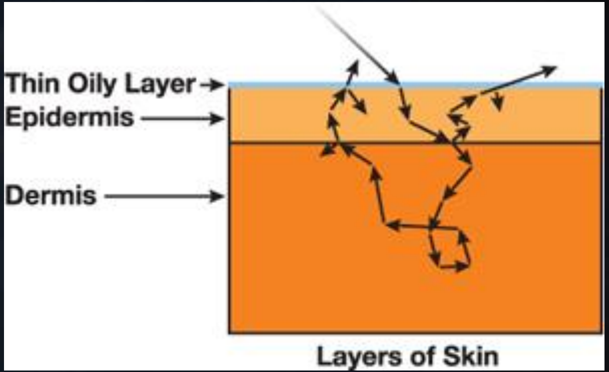
Subsurface Scattering



[EON07]

Copyright © 2006 Pearson Education, Inc., publishing as Benjamin Cummings.

Subsurface Scattering



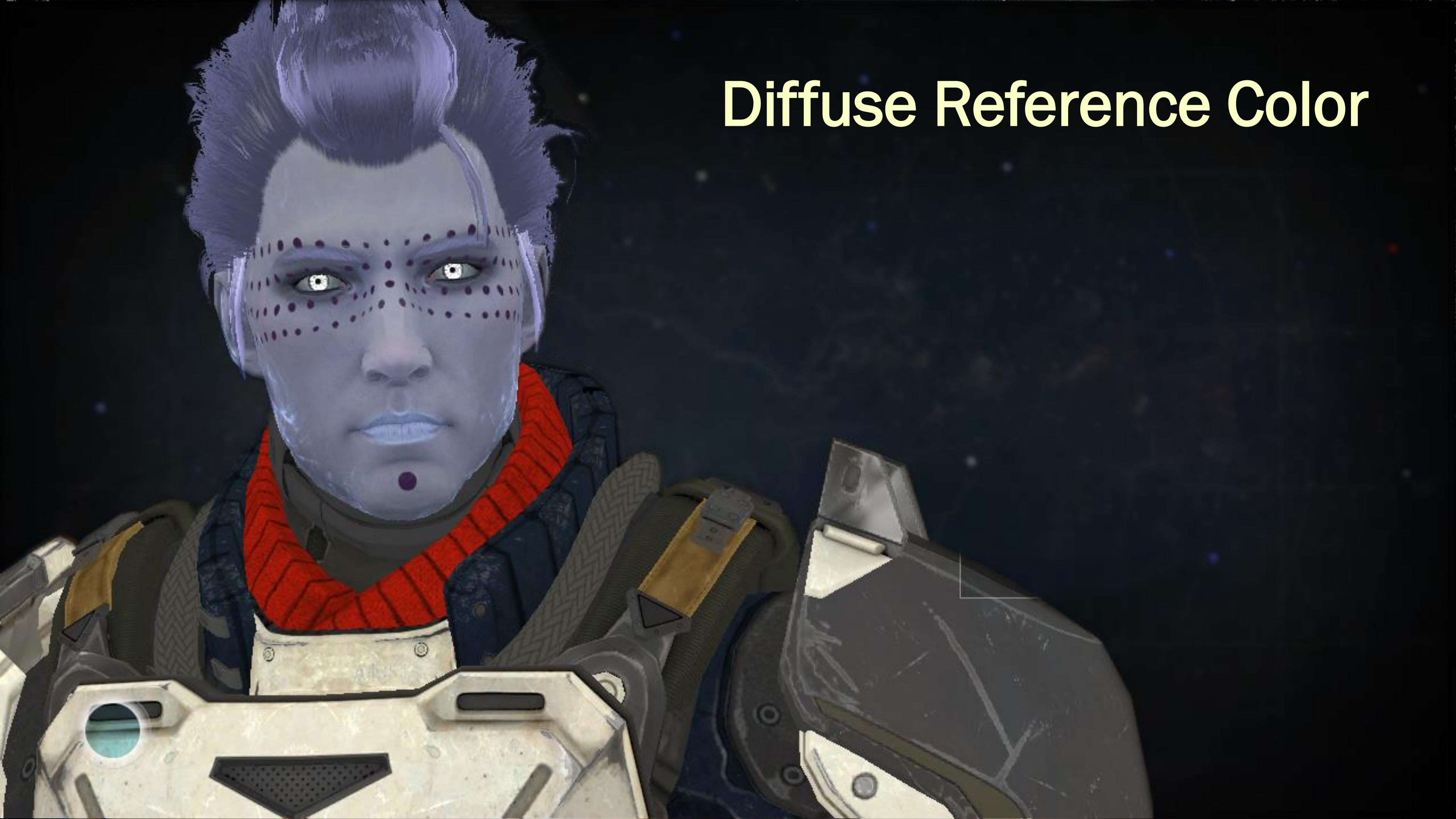
[EON07]

BSSRDF

Subsurface Scattering Breakdown

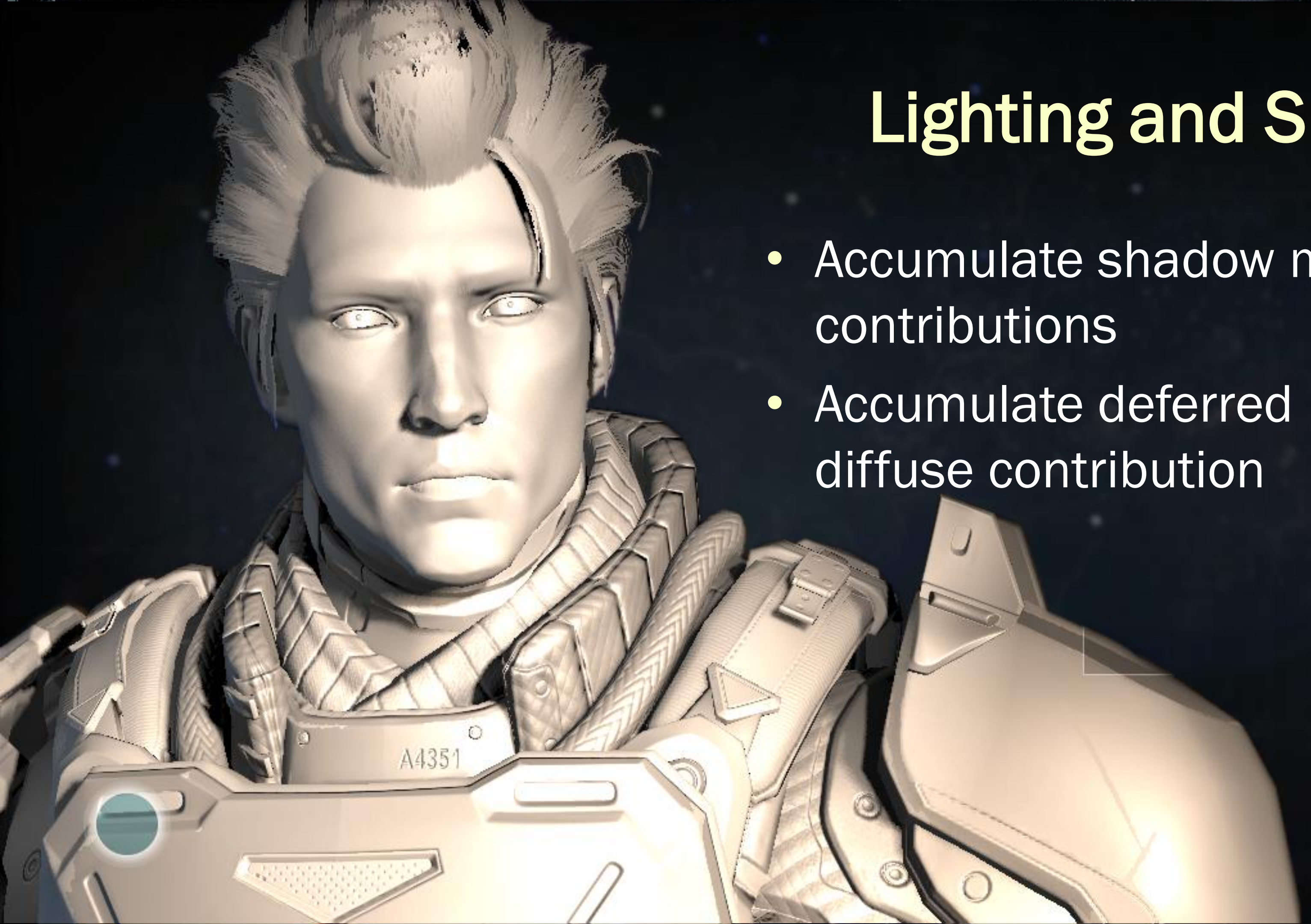
- Accumulate lighting and shadows
- Render subsurface prepass
- Scatter diffuse lighting
- Shade

Diffuse Reference Color



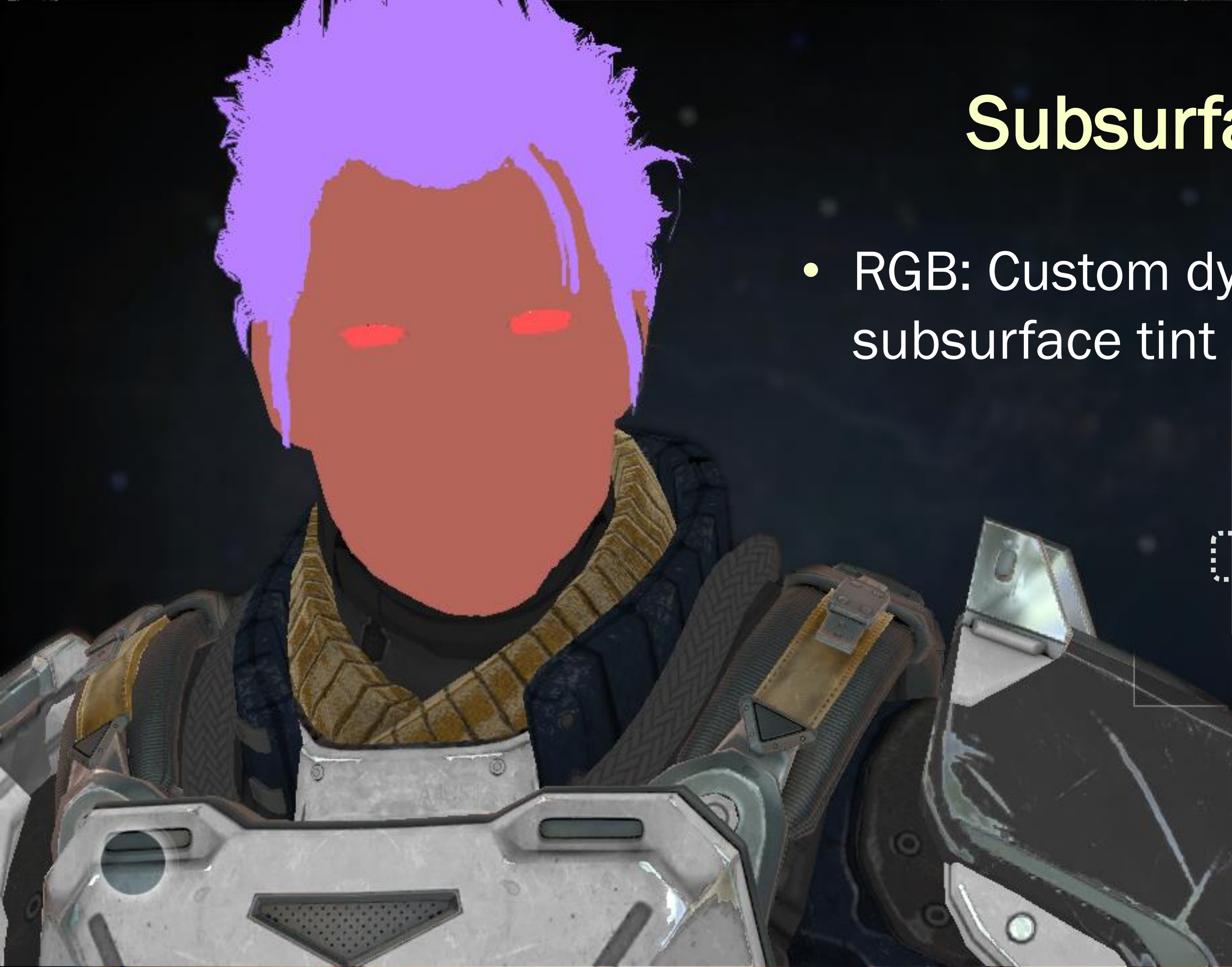
Lighting and Shadows

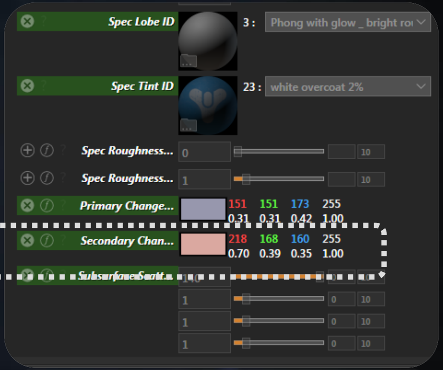
- Accumulate shadow maps contributions
- Accumulate deferred lights diffuse contribution



Subsurface Prepass

- RGB: Custom dye-driven subsurface tint





Spec Lobe ID 3 : Phong with glow _ bright ro

Spec Tint ID 23 : white overcoat 2%

Spec Roughness... 0 10

Spec Roughness... 1 10

Primary Change... 151 151 173 255
0.31 0.31 0.42 1.00

Secondary Chan... 218 168 160 255
0.70 0.39 0.35 1.00

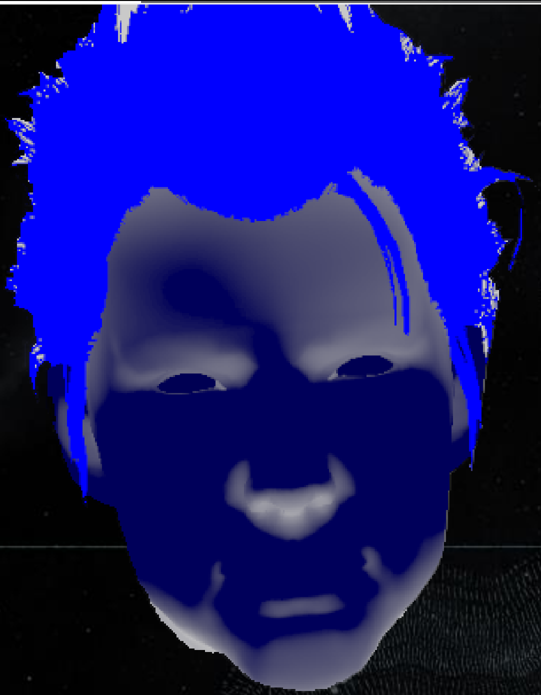
Subsurface Scatt... 140 0 10

1 0 10

1 0 10

1 0 10

Awoken skin dye

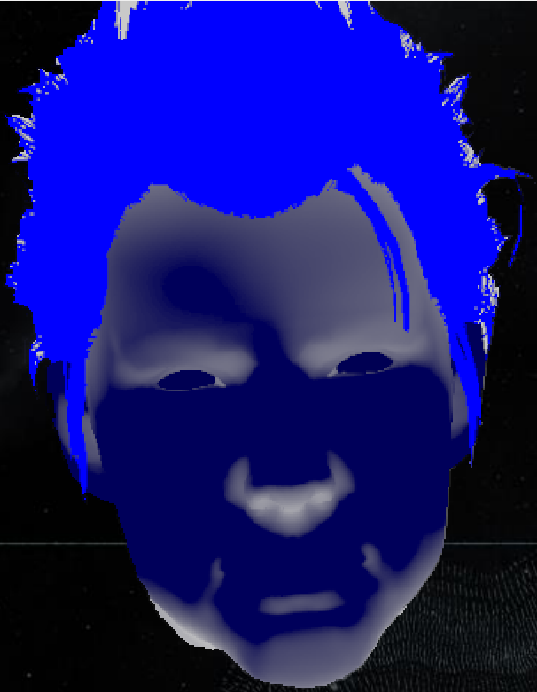


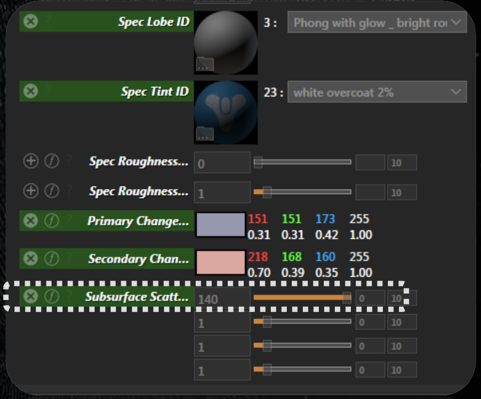
Subsurface Prepass

- Alpha: Per-Pixel dye- and gear-stack driven subsurface strength

Subsurface Prepass

- Alpha: Per-Pixel dye- and gear-stack driven subsurface strength amount





Spec Lobe ID 3 : Phong with glow _ bright ro

Spec Tint ID 23 : white overcoat 2%

Spec Roughness... 0 10

Spec Roughness... 1 10

Primary Change... 151 151 173 255
0.31 0.31 0.42 1.00

Secondary Chan... 218 168 160 255
0.70 0.39 0.35 1.00

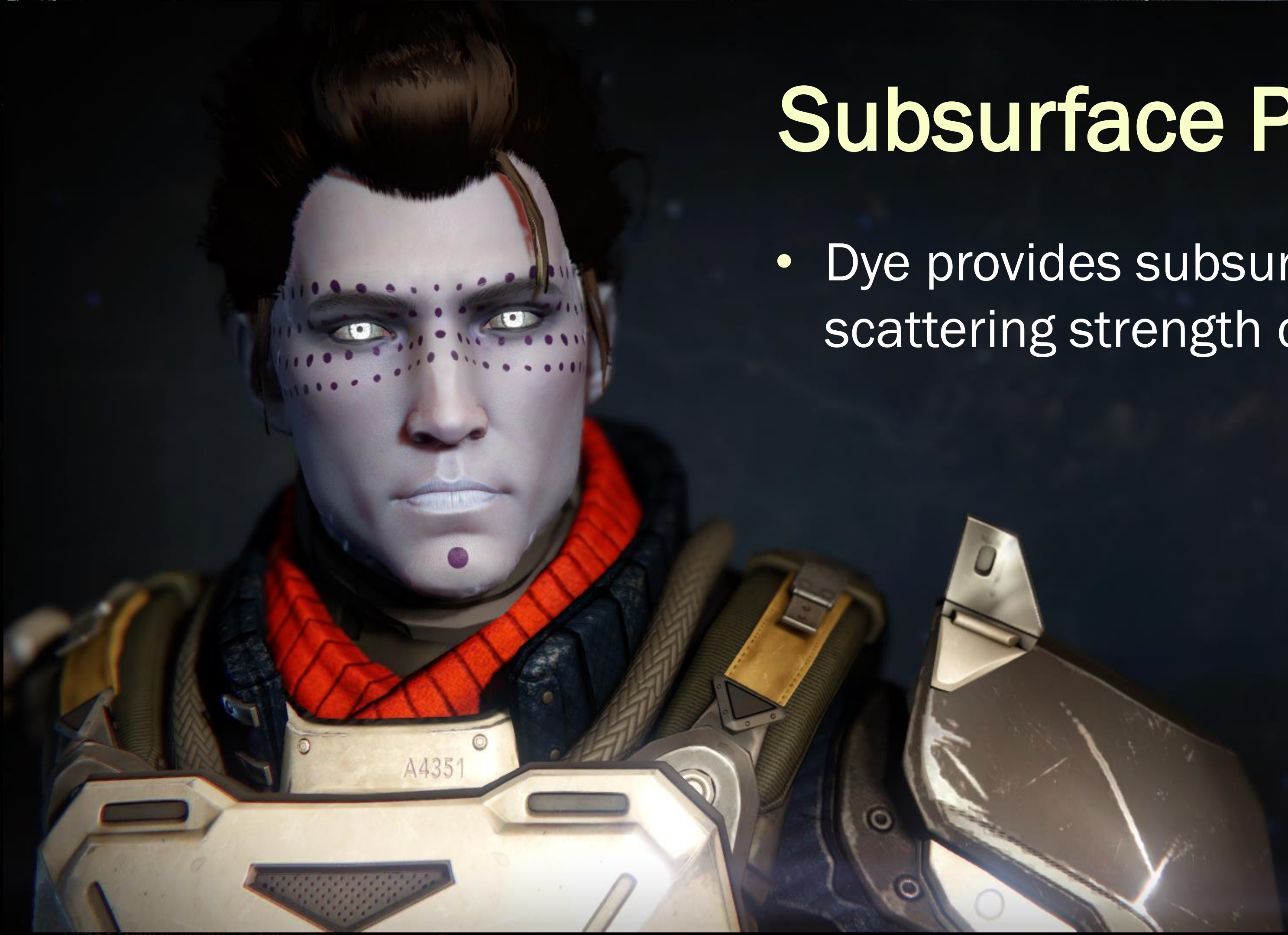
Subsurface Scatt... 140 0 10
1 0 10
1 0 10
1 0 10

Subsurface Prepass

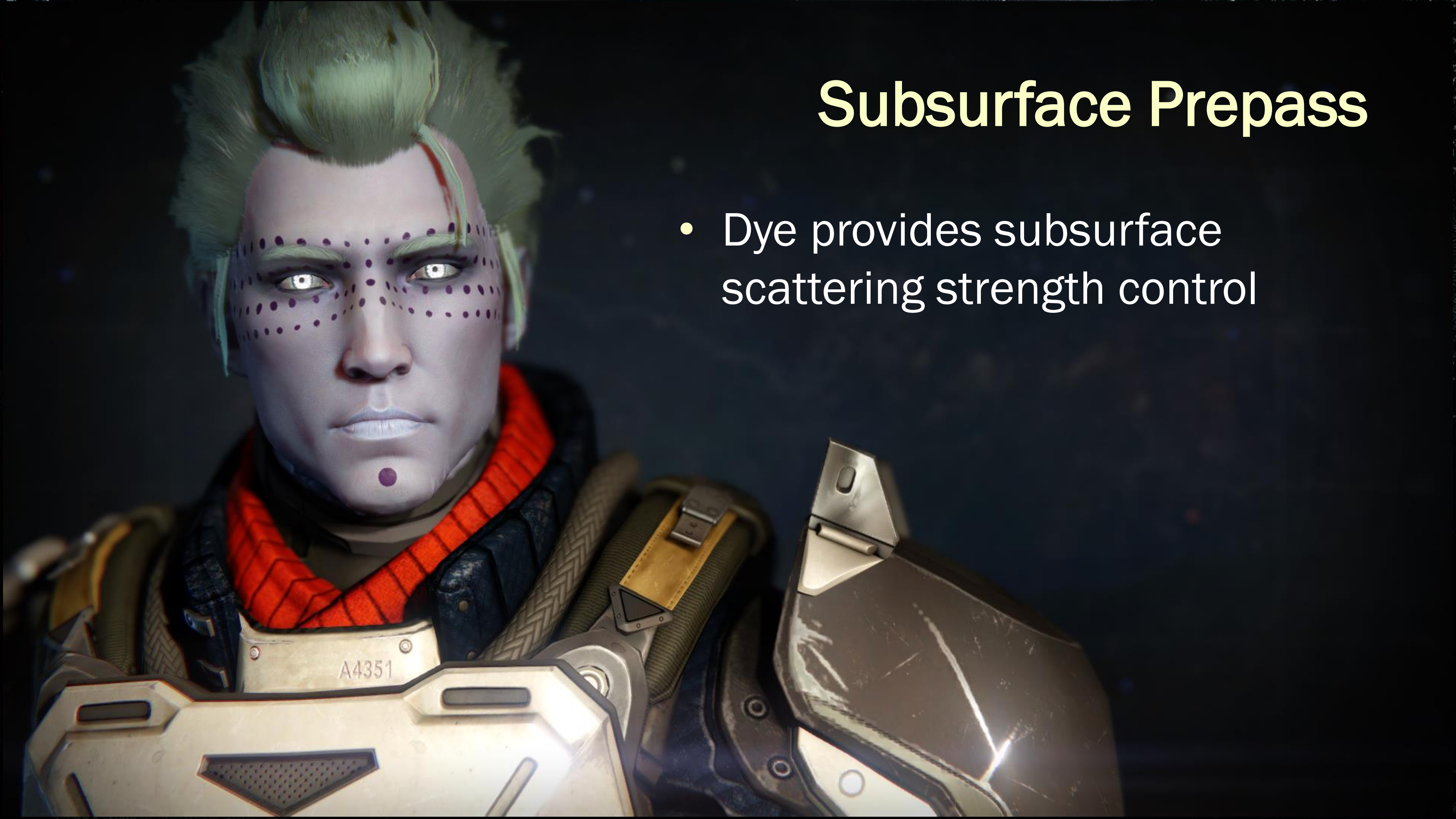
- Dye provides subsurface scattering strength control

Subsurface Prepass

- Dye provides subsurface scattering strength control



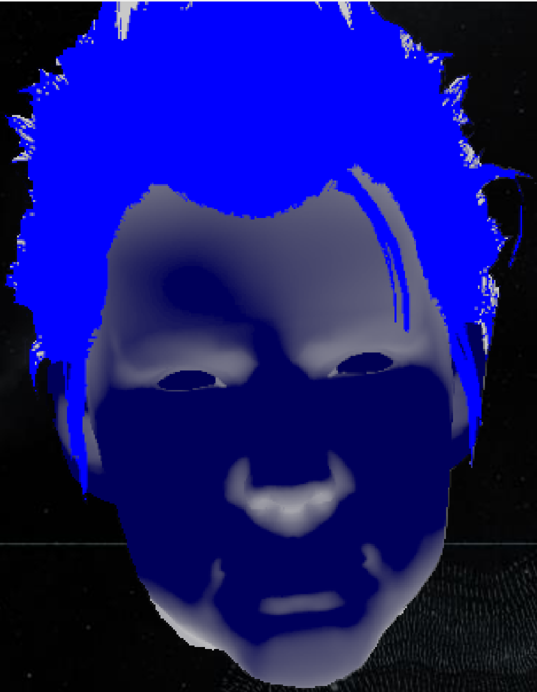
Subsurface Prepass

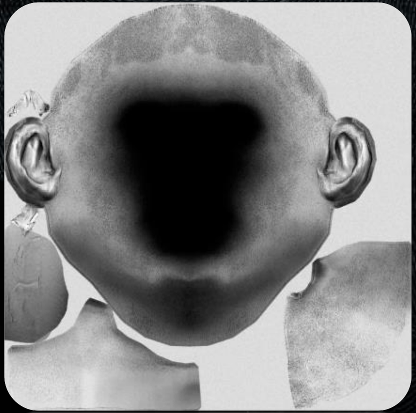


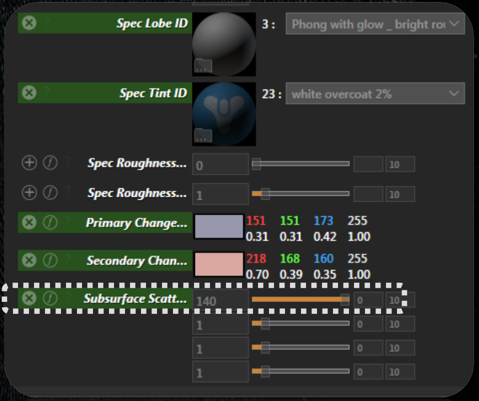
- Dye provides subsurface scattering strength control

Subsurface Prepass

- Alpha: Per-Pixel dye- and gear-stack driven subsurface strength amount







Spec Lobe ID 3 : Phong with glow _ bright roı

Spec Tint ID 23 : white overcoat 2%

Spec Roughness... 0 10

Spec Roughness... 1 10

Primary Change... 151 151 173 255 0.31 0.31 0.42 1.00

Secondary Chan... 218 168 160 255 0.70 0.39 0.35 1.00

Subsurface Scatt... 140 0 10

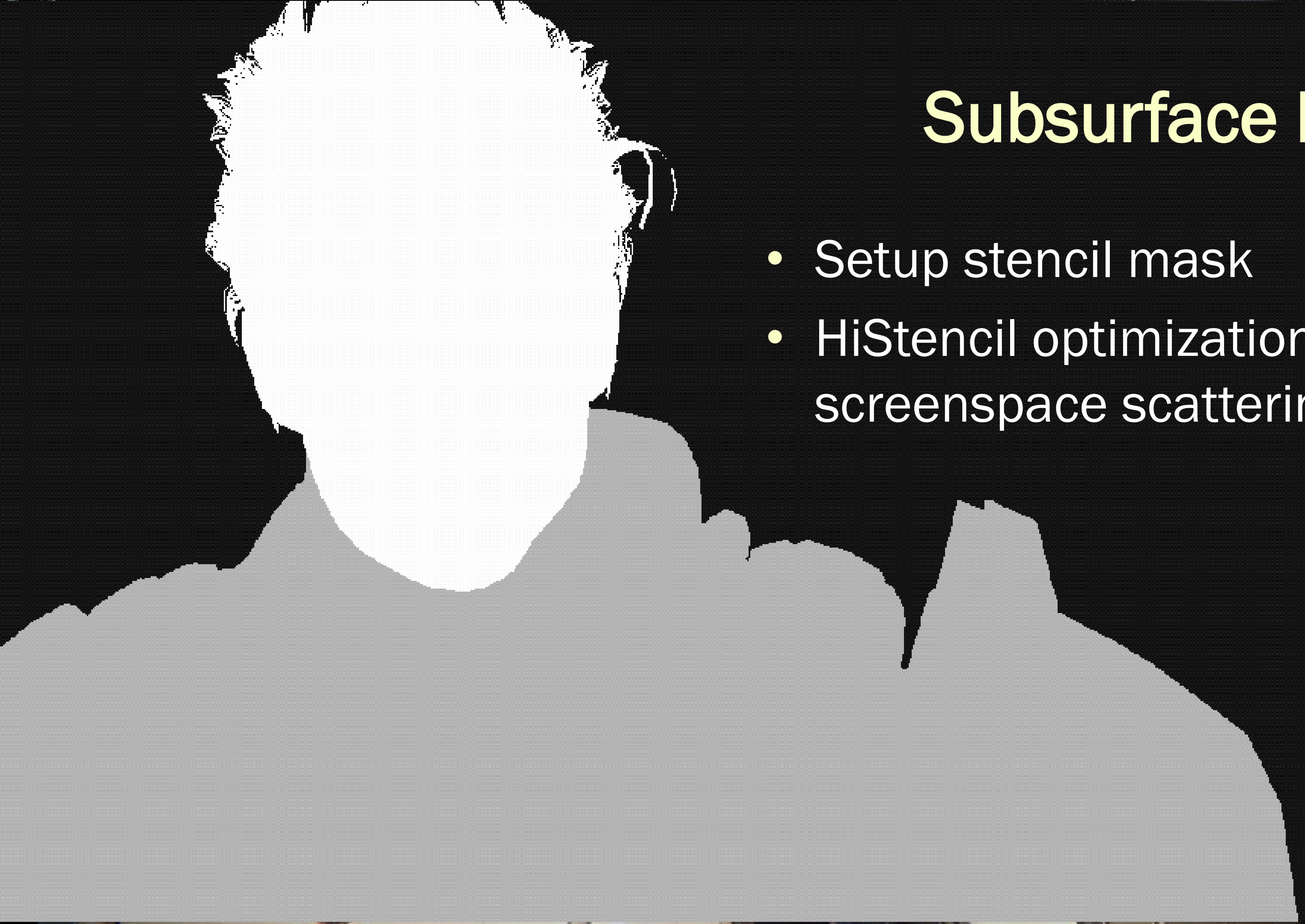
1 0 10

1 0 10

1 0 10

Subsurface Prepass

- Setup stencil mask
- HiStencil optimization during screenspace scattering passes

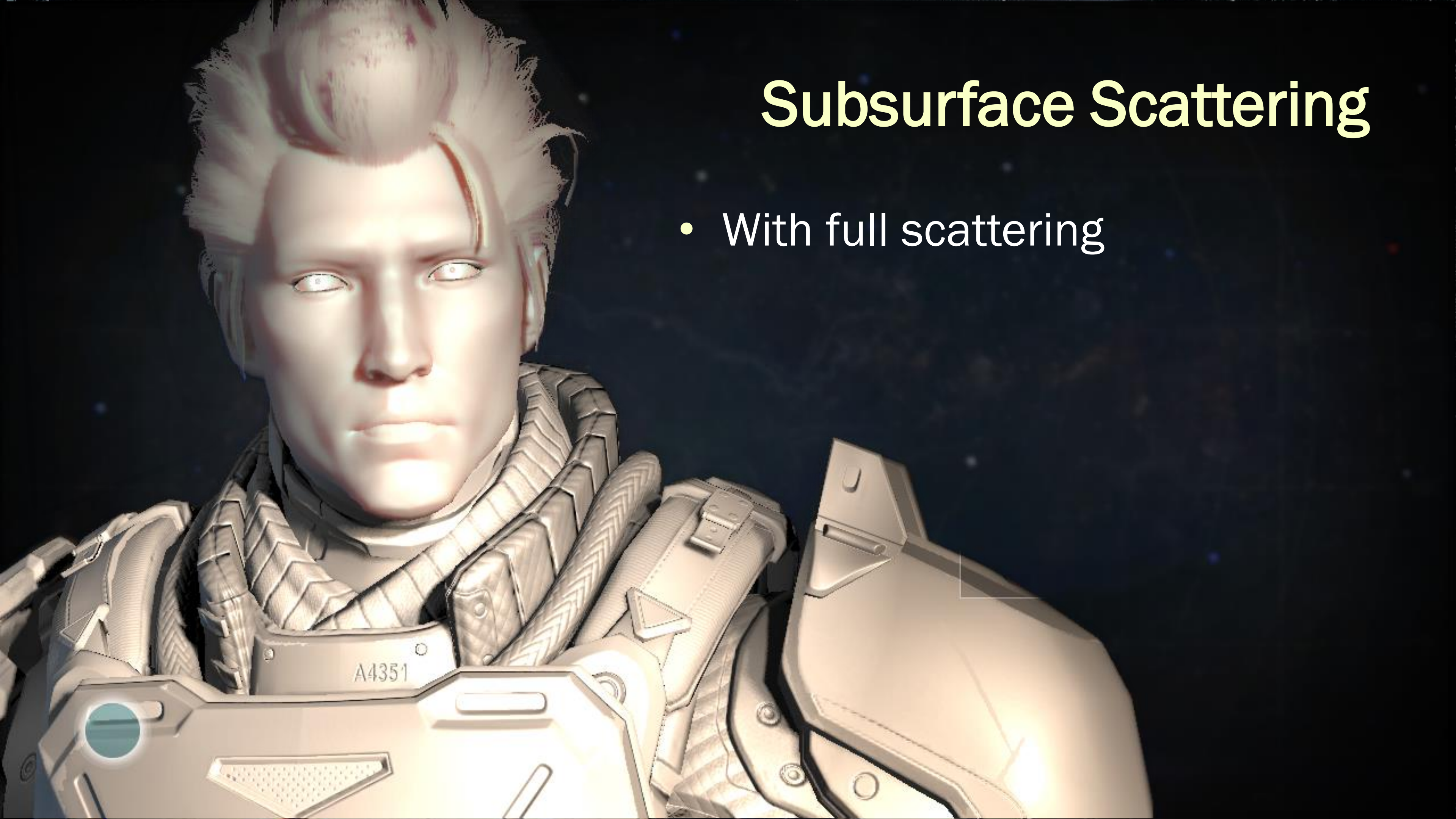


Subsurface Scattering

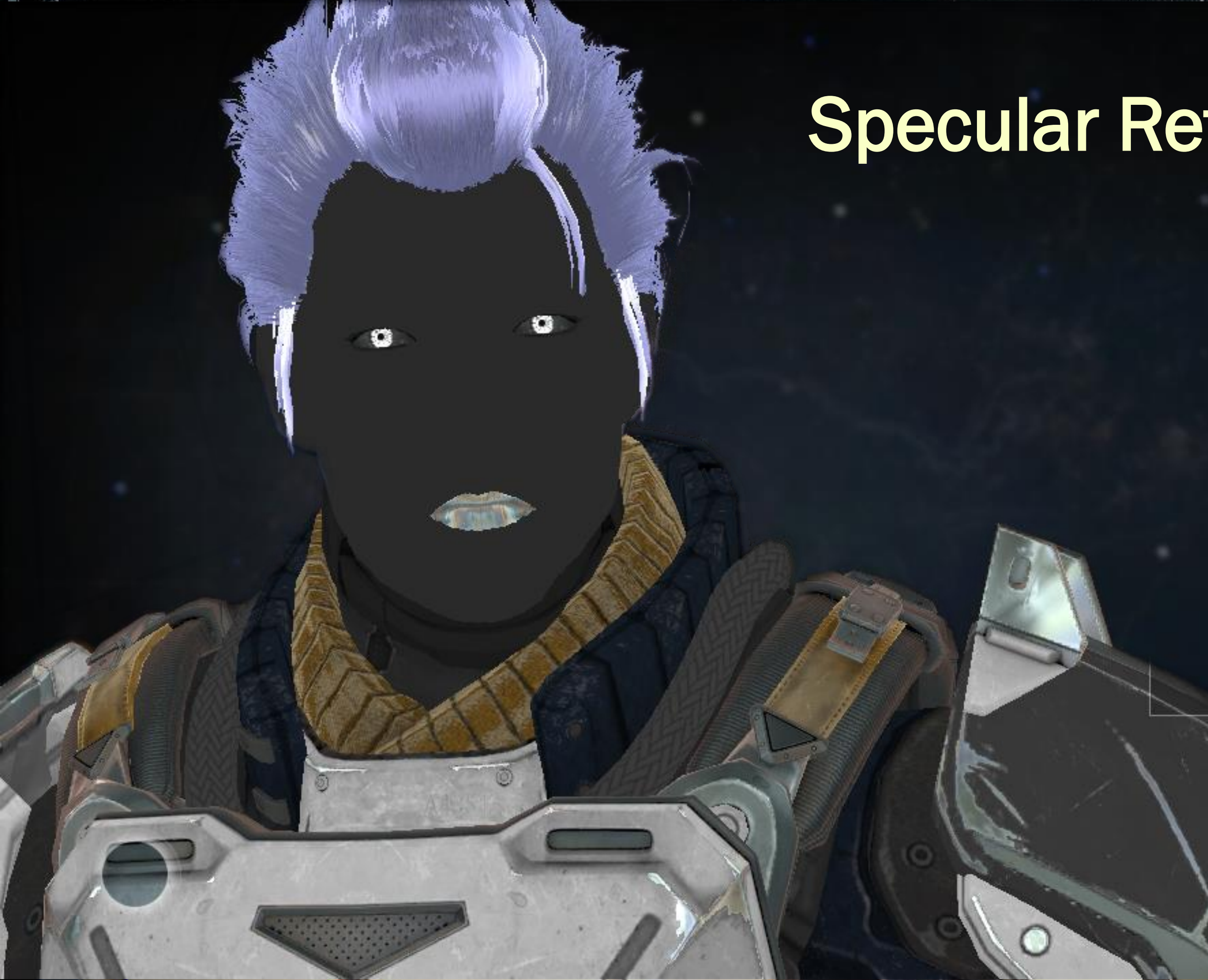
- Run two screenspace separable scattering approximation passes
- Previous work
  - [D'Eon 07]
  - [Jimenez 12]
- Implemented custom extended version

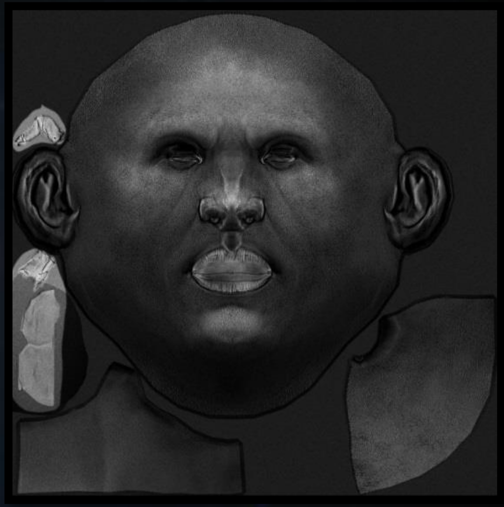
Subsurface Scattering

- With full scattering



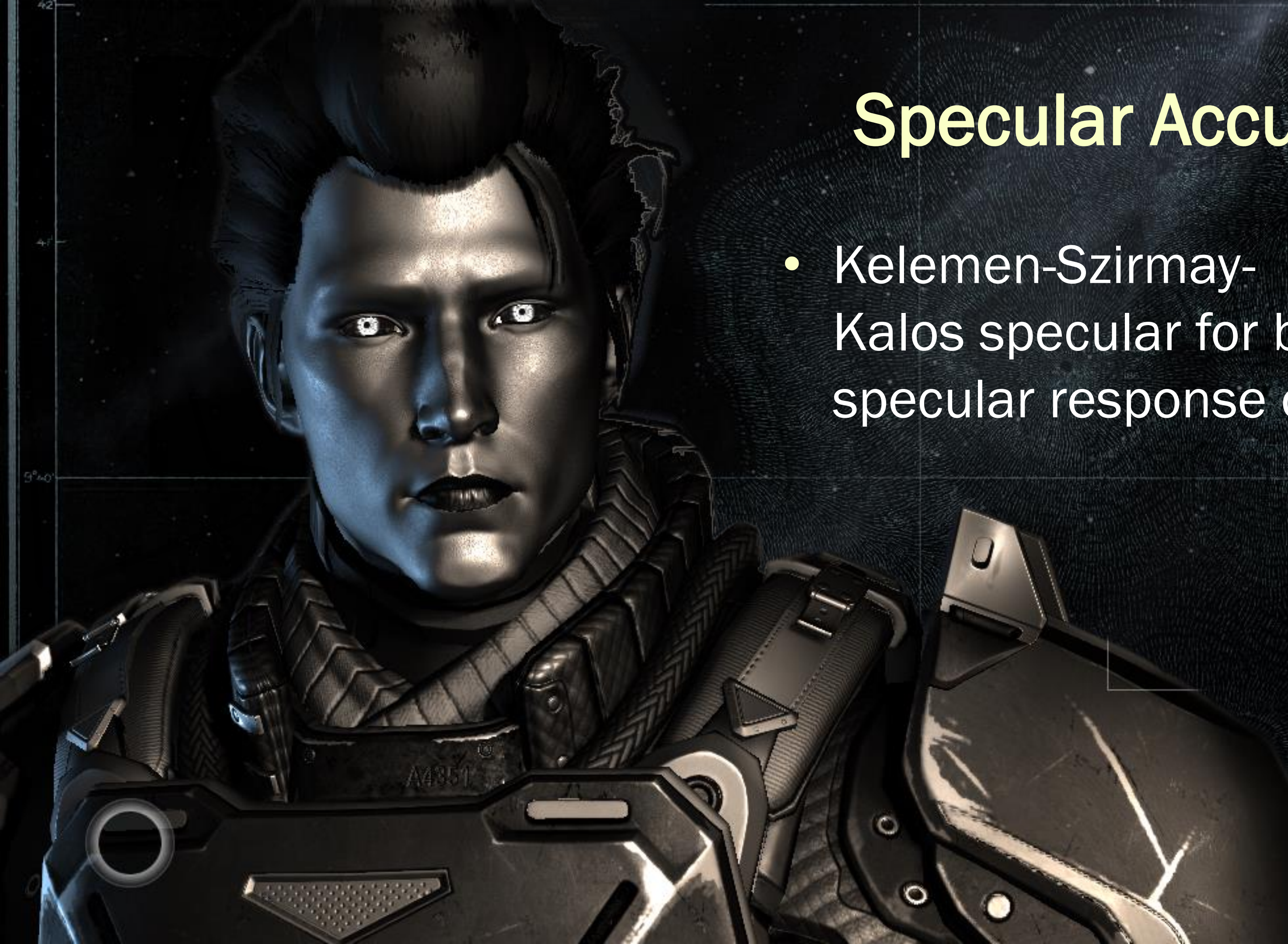
Specular Reference Color



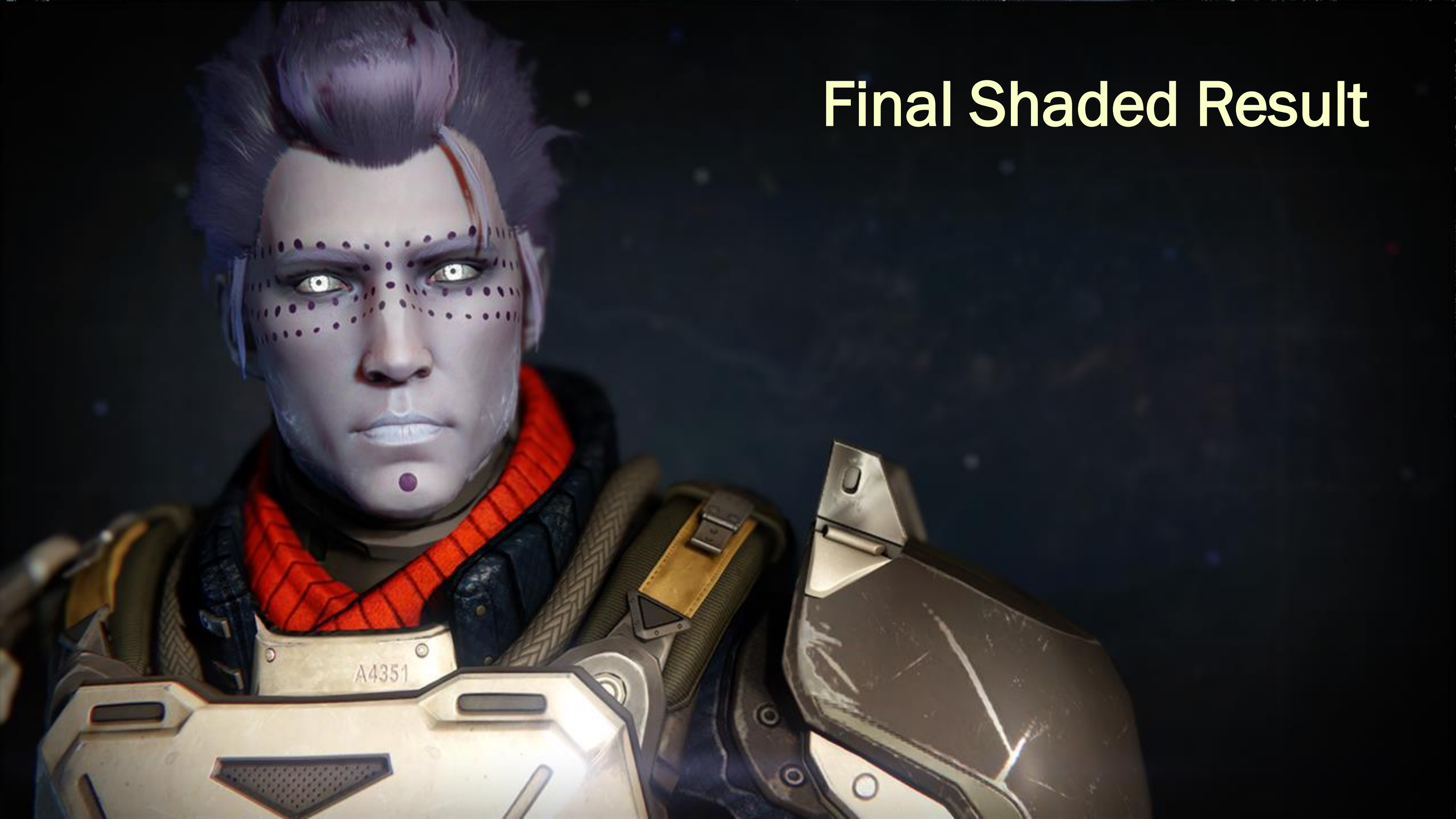


Specular Accumulation

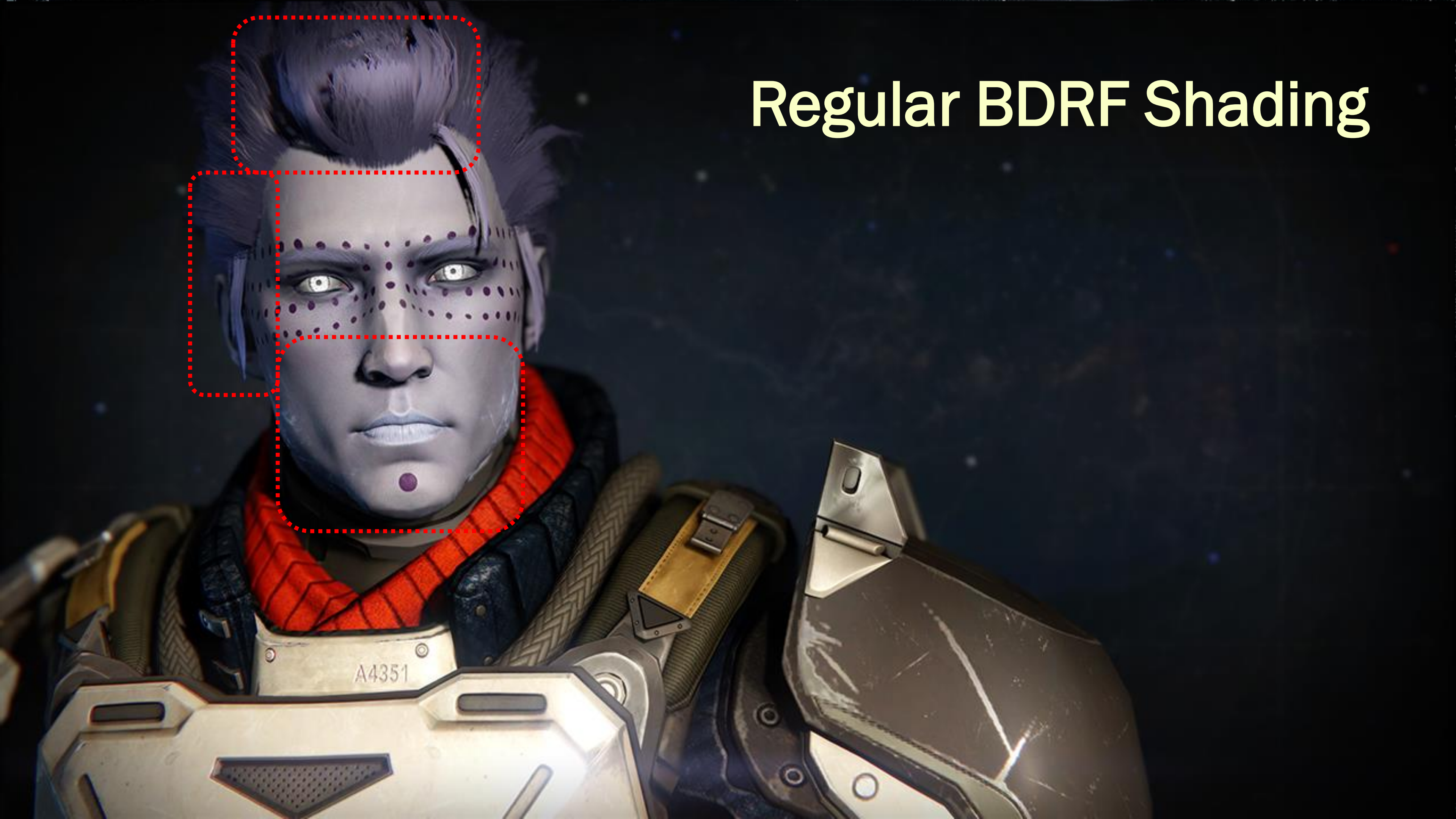
- Kelemen-Szirmay-Kalos specular for bright specular response on rim lights



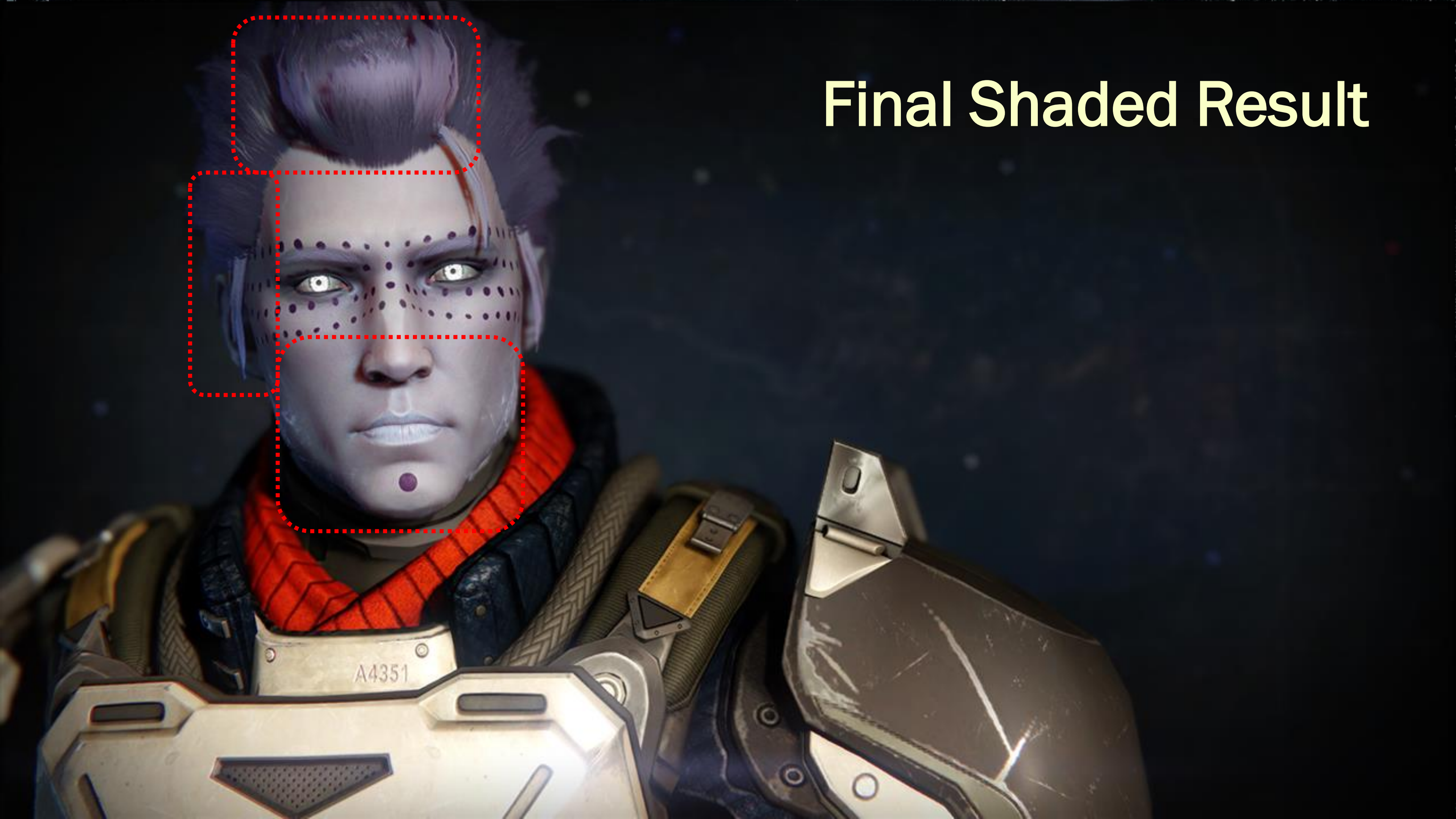
Final Shaded Result



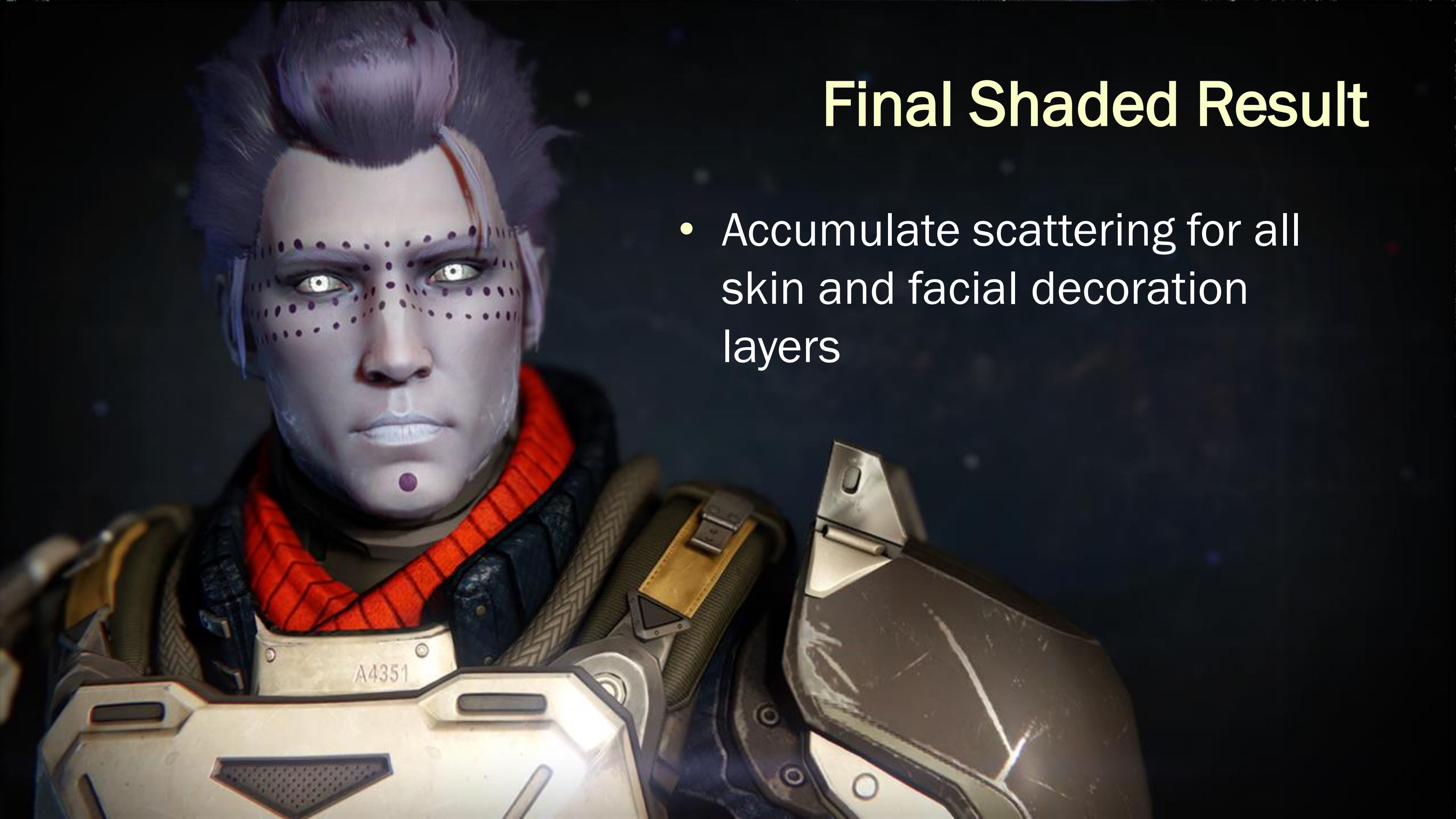
Regular BDRF Shading



Final Shaded Result



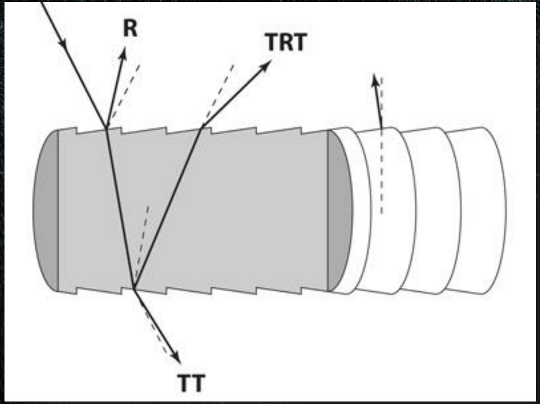
Final Shaded Result



- Accumulate scattering for all skin and facial decoration layers

Deferred Hair Rendering Challenges

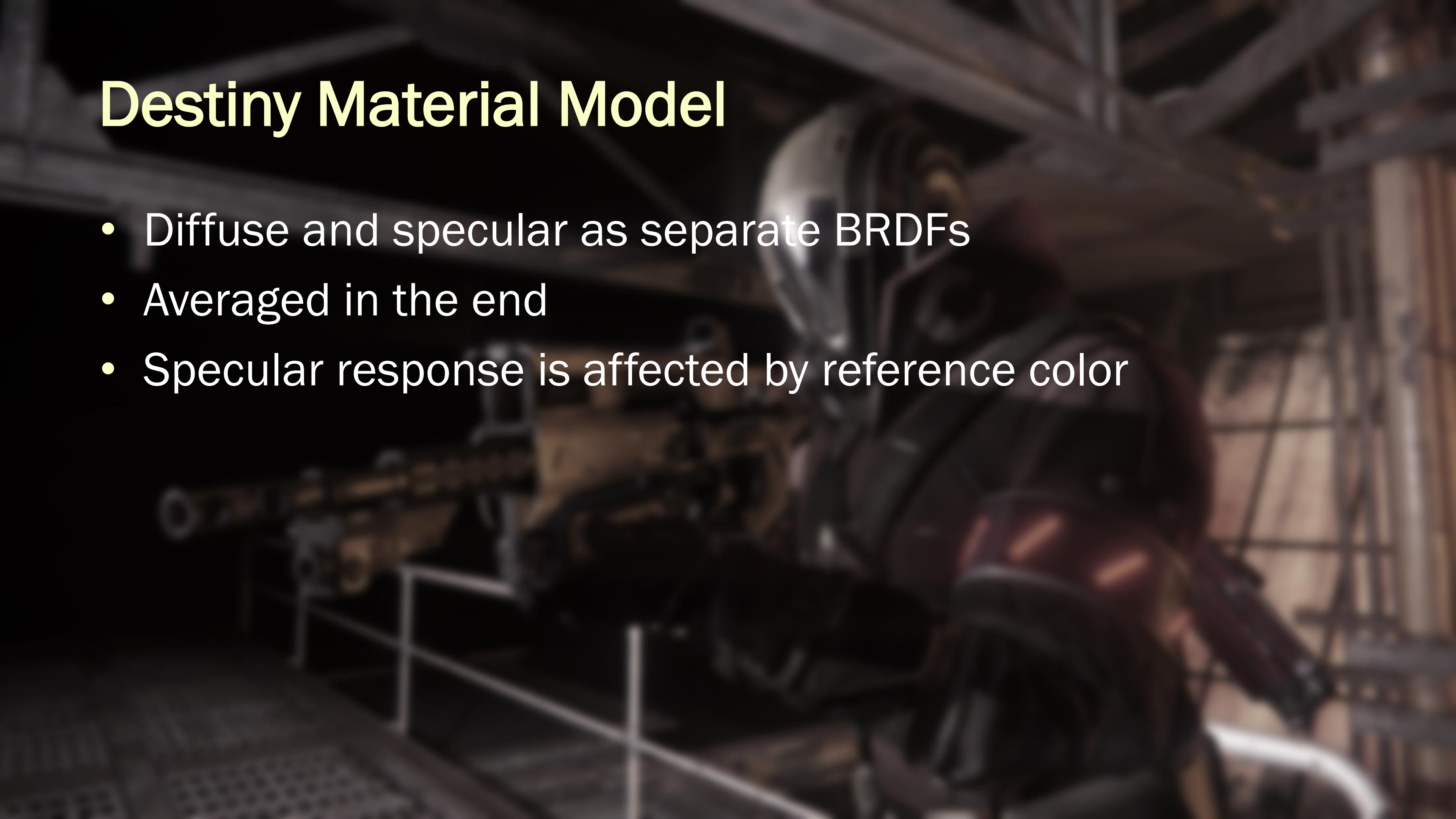
- Anisotropy depends on light direction
- Typically evaluated as forward-lit
- Problematic for production
  - Inconsistent lighting and shadowing
  - Requires custom lit materials
  - Complex to maintain and polish



[Marschner et al 03]

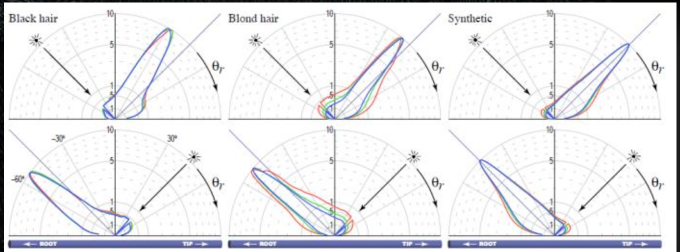
Destiny Material Model

- Diffuse and specular as separate BRDFs
- Averaged in the end
- Specular response is affected by reference color



Deferred Anisotropy

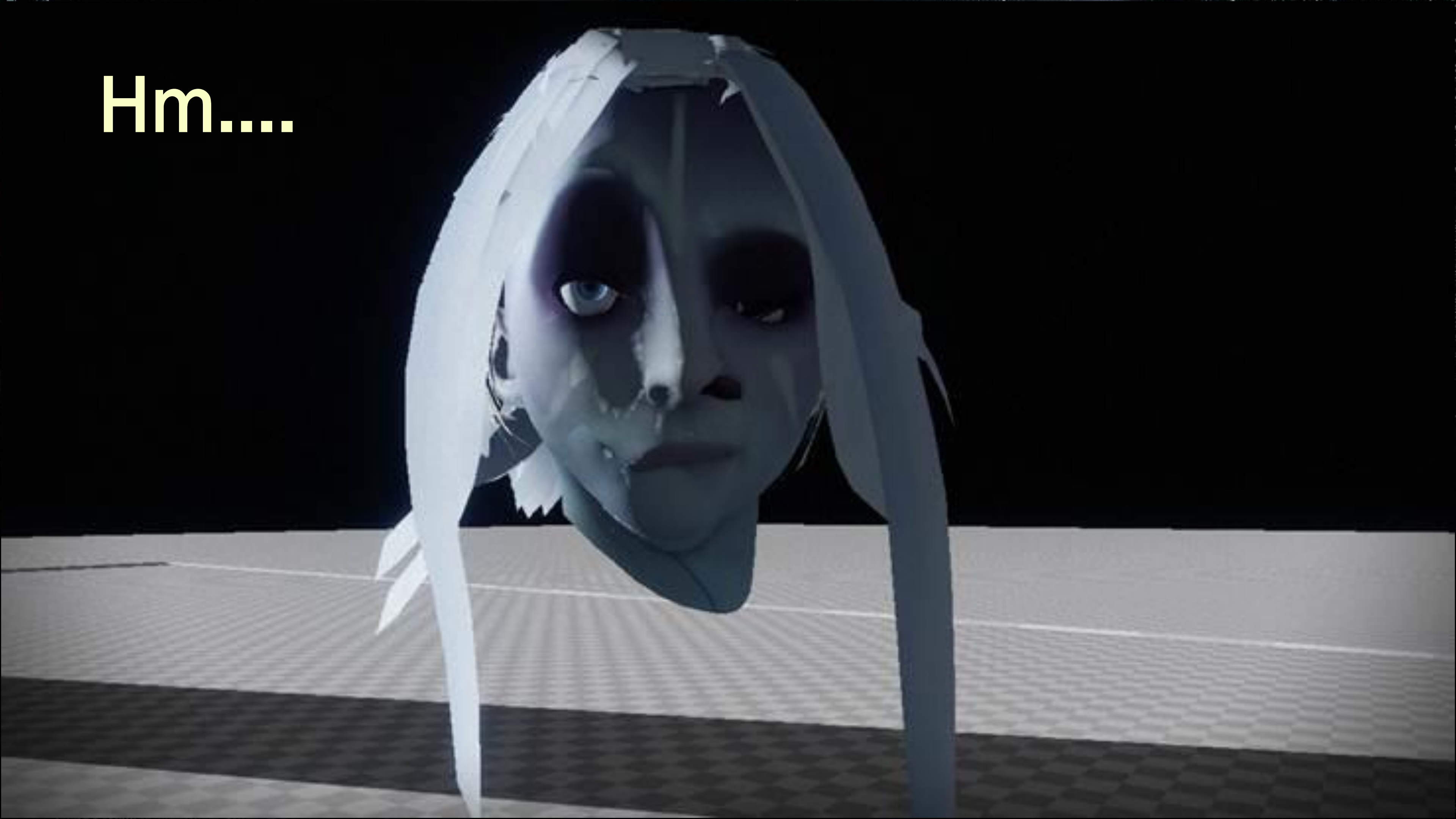
- Compute modified Marshner03 / Scheuermann04 terms
  - Extended the shifted primary and secondary specular highlights terms computation



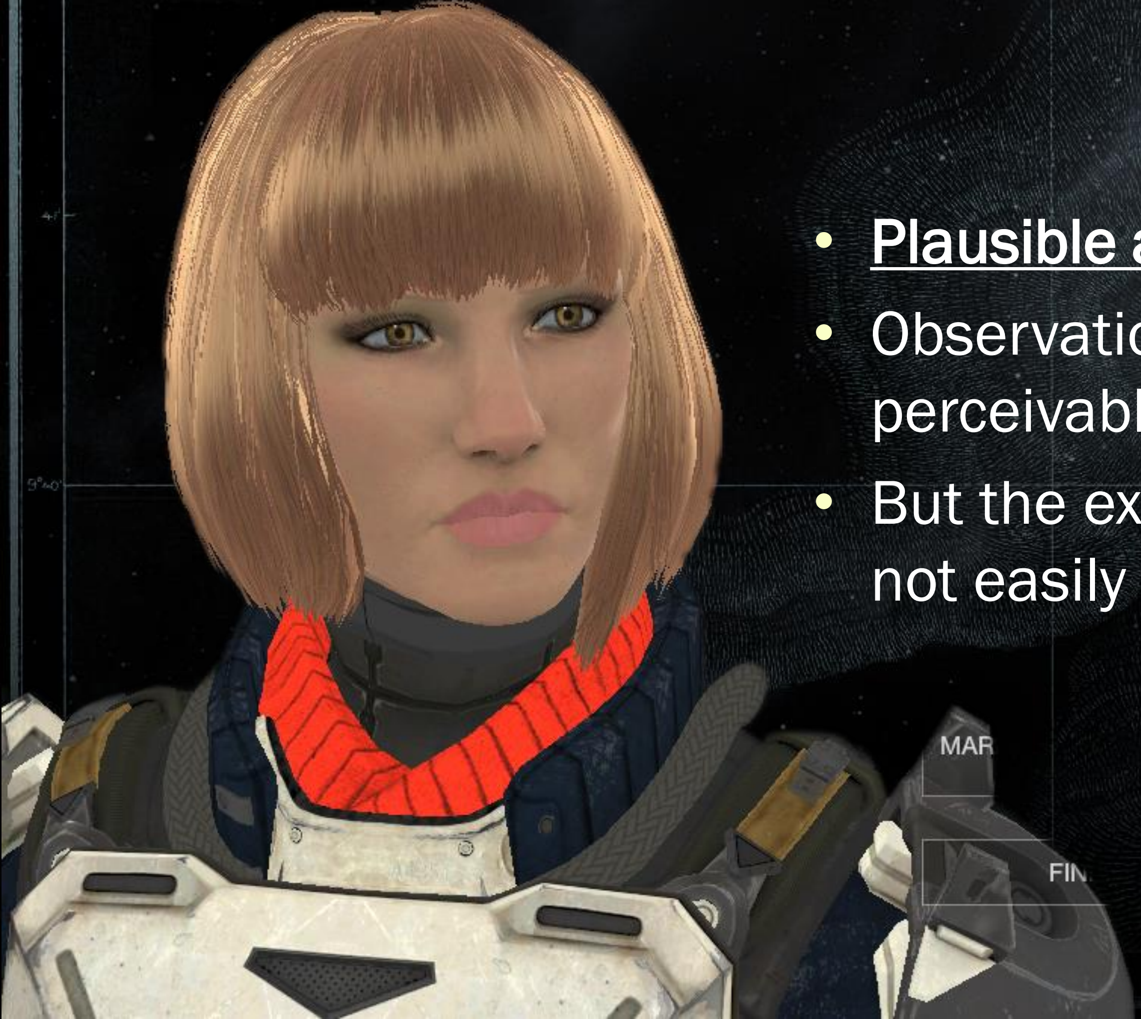
But… Light Direction?

- Need information about the light source
- But we're in G-buffer pass..
  - No light information available?
- **<u>Houston, we have a problem!</u>**

Hm....



Aha!



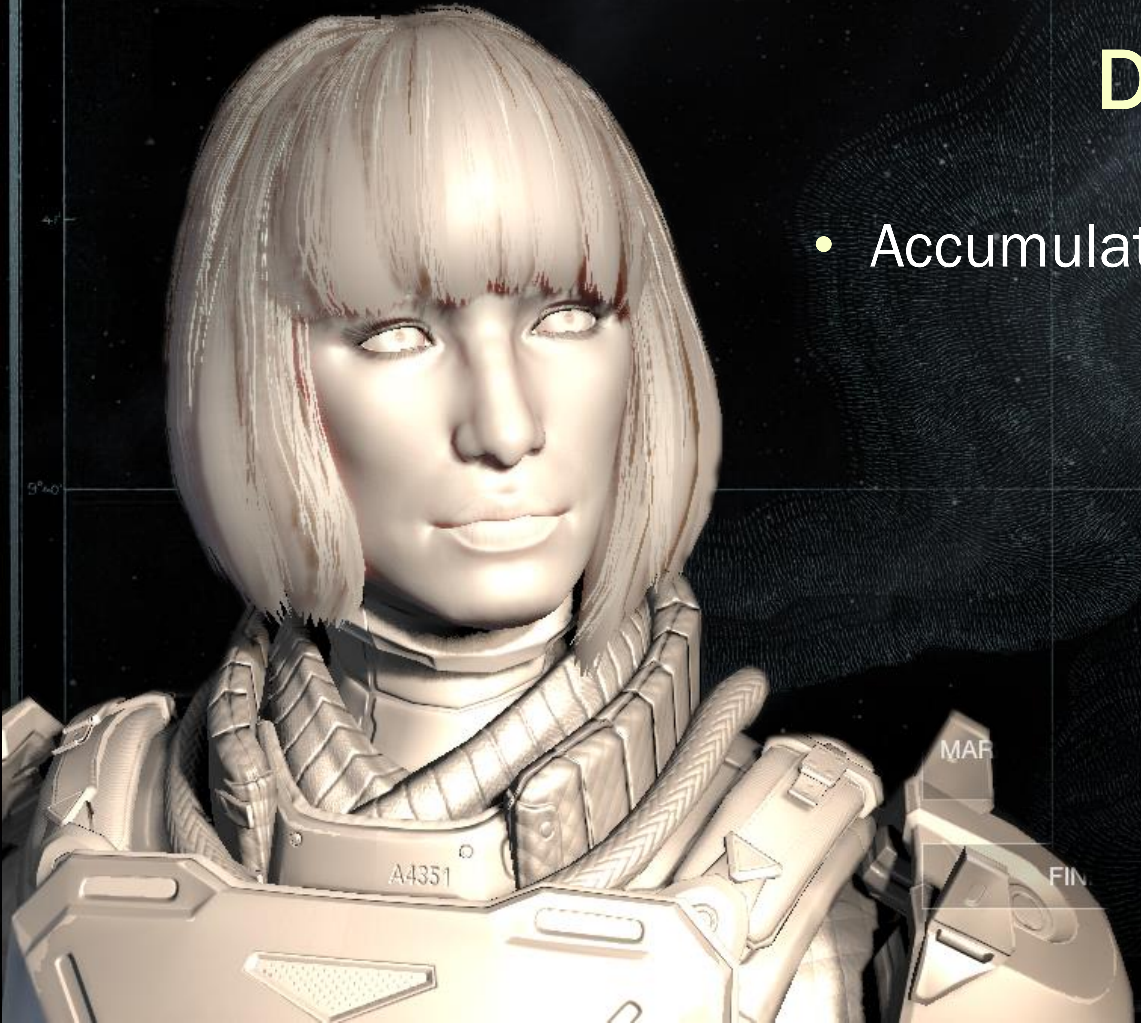
- **Plausible and Believable**
- Observation: anisotropy is perceivable in its presence
- But the exact anisotropy *source* is not easily distinguishable

Plausible and Believable!

- Dominant light source drives anisotropy
- Outdoors: sunlight
- Indoors: dynamic selection
- Feed to G-buffer shader via channels

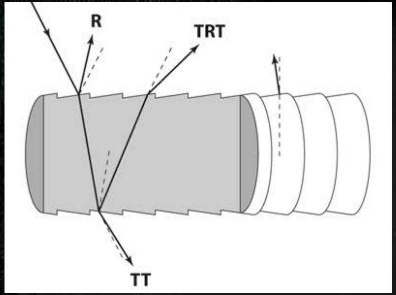
Deferred Anisotropy

- Accumulate diffuse lighting

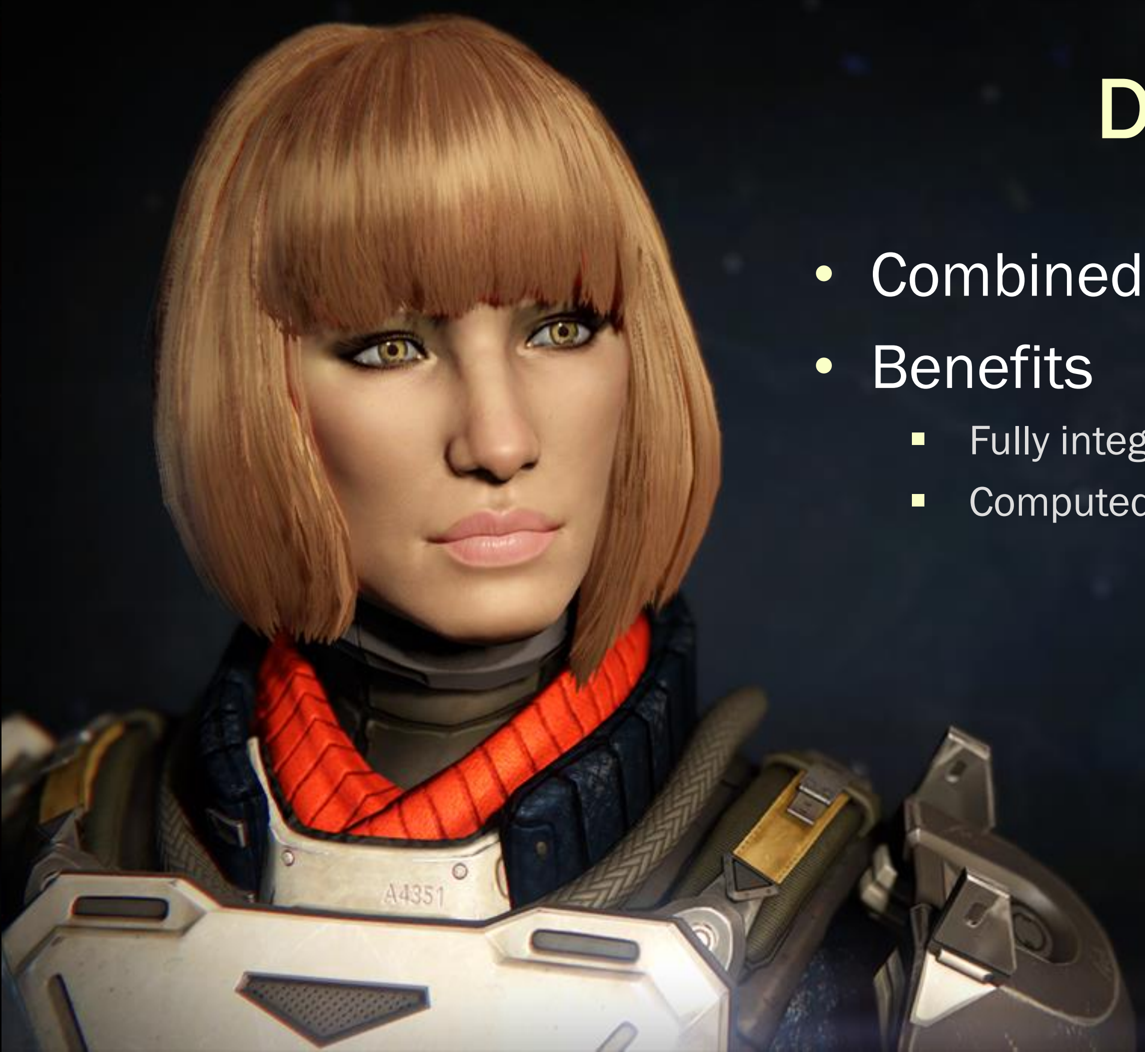


Deferred Anisotropy

- Accumulate specular lighting



Deferred Anisotropy



- Combined final result
- Benefits
  - Fully integrated lighting and shadowing
  - Computed pre-subsurface scattering

Putting It All Together: Armor

How Did We Do?

Lessons Learned: What Went Right

- Flexible, diverse and scalable system
- Achieved art bar within design, production and engineering constraints

Challenges Encountered

- The complexity of the system grew as the project matured
- Tools matured toward the end
- Managing the permutations of features was challenging

GE
DESTINY

Thanks for attending our talk!

- Hope the talk was useful, and you guys got something out of it
- Thanks to our studio’s talented engineers, tools, tech and art team for making this all possible.
- Please stay for more Destiny talks after the Q&A

Questions?

Destiny @ SIGGRAPH

Character Heads Creation Pipeline and Rendering in Destiny

Tuesday, 12 August 10:45 AM - 12:15 PM
West Building, Ballroom C/D

Destiny Character-Animation System and Lessons Learned

Tuesday, 12 August 2:00 PM - 5:15 PM
West Building, Rooms 211-214

Bungie was founded in 1991 with two goals:

- *Combine brilliant technology and beautiful art with captivating gameplay*
- *Achieve World Domination*

We're Hiring. Apply Now.

www.bungie.net/careers

careers@bungie.com