



Unreal Fest 2024 Seoul

# <철권 8> 출시와 동시에 12,000개 아이템 공개! 캐릭터 커스터마이징 아이템 제작과 차세대 콘솔용 렌더링 표현에 대하여

**Wataru Ogasa**  
Engineer  
Bandai Namco Studios Inc.

# 목차

## Section 1

발표자 소개

## Section 2

『철권 8』의 캐릭터 커스터마이징 기능

## Section 3

아이템 제작 · 양산 환경의 정비

## Section 4

아이템 렌더링 처리 구현

## Section 5

요약 및 정리

# 1. 발표자 소개

## 1. 발표자 소개

**Wataru Ogasa**

Engineer  
Bandai Namco Studio Inc.

Author / Presenter

**Tsuyoshi Fukutomi**

Engineer  
Bandai Namco Studio Inc.

Co-Author

**Hirotaka Kawakami**

Technical Artist  
Bandai Namco Studio Inc.

Co-Author

## 2. 『철권8』의 캐릭터 커스터마이징 기능



## 『철권 8』이란 ?

출시 1개월 만에 전 세계 누적 출하량 **200 만장** 돌파 !

장르 3D 대전 격투 액션 게임

출시일 2024년 1월 26일

대응 플랫폼 PC (Steam)  
PlayStation 5  
Xbox Series X / S

Unreal Engine Ver. 5.2.1

개발원 반다이남코 스튜디오

발매원 반다이남코 엔터테인먼트



『어그레시브』한 불꽃 튀는 격투 !

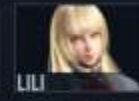
## 캐릭터 커스터마이징 기능 개요

### ◆ 플레이어블 캐릭터의 파트(부위)별로 취향에 따라 커스터마이징

- 총 25종의 부위, 1 캐릭터당 약 400개의 아이템이 존재
- 캐릭터별 **고유 아이템** 외에 모든 캐릭터가 장착할 수 있는 **공유 아이템**이 존재
- 컬러 변경 및 액세서리 부위는 사이즈와 위치도 변경 가능
- 생성한 커스터마이징은 **배틀에서 사용** 가능



# CHARACTER CUSTOMIZE



## 캐릭터 커스터마이징 구현에 대한 요구사항

---

런칭 시점에 약 **12,000여 종**의 아이템  
**제작 · 양산** 체제 구축

차세대 감성이 느껴지는 캐릭터 표현 +  
**쾌적한 배틀이 가능한 드로잉 처리** 구현

### 3. 아이템 제작 · 양산 환경의 정비

## 아이템 제작 환경에 대한 요구사항

---

- ◆ 아이템 메시만으로는 커스터마이징 아이템으로 사용이 불가함

➡ 커스터마이징 아이템으로 **핸들링하기 위한 정보 에셋이** 필요함

- ◆ 모든 캐릭터에 장착 가능한 돌려쓰는 아이템(공유 아이템)이 존재함

➡ 공유 가능한 설정은 하나의 에셋으로 정리해둘 필요가 있음

- ◆ 아이템을 손쉽게 확인할 수 있는 환경이 필요

➡ **전용 레벨을 구축**하여 대응, 정보 에셋도 사용 가능하도록 지원

## 아이템 제작 환경에 대한 요구사항

---

- ◆ 아이템 메시만으로는 커스터마이징 아이템으로 사용이 불가함

➡ 커스터마이징 아이템으로 **핸들링하기 위한 정보 에셋이** 필요함

- ◆ 모든 캐릭터에 장착 가능한 돌 **아이템 관리 에셋**

➡ 공유 가능한 설정은 하나의 에셋으로 정리해줄 필요가 있음

- ◆ 아이템을 손쉽게 확인할 수 있는 환경이 필요

➡ 전용 레벨을 구축하여 대응, 정보 에셋도 사용 가능하도록 지원

# 아이템 관리 에셋

## ItemPrefab (IP)

BI, BCI, CI, DI의 각 에셋에 대한 참조만을 가진다.  
각 아이템에 대해 1개씩 존재

## Baseltem (BI)

아이템의 기본 정보를 가진다.  
아이템 부위, 메시, 관절, 기술 아이템 정보 등...  
각 아이템의 베이스로 1개 존재

## BaseCharacterItem (BCI)

각 캐릭터별 설정 정보를 가진다.  
장착 조합 정보, 캐릭터별 오프셋 정보 등...  
추가 설정이 필요한 캐릭터만큼 존재

## CustomizeItem (CI)

아이템 외형에 관한 정보를 가진다.  
베이크 정보(컬러, 패턴 변경 등) 등...  
아이템의 디버깅에 관한 정보를 가지고 있다. 기본 컬러 1개, 추가 컬러 변경  
아이템만큼 존재

## DebugItem (DI)

아이템 디버깅에 관한 정보를 가진다.  
썸네일 촬영 정보 등...  
기본적으로 아이템별로 존재하지만, 재사용이 가능



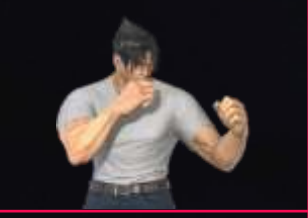

# 아이템 관리 에셋

	폴 티셔츠 아이템 무지
	폴 티셔츠 아이템 호피무늬
	진 티셔츠 아이템 무지
	진 티셔츠 아이템 호피무늬

# 아이템 관리 에셋

	<b>폴 티셔츠 아이템 무지</b>	IP_pau_tshirt
	<b>폴 티셔츠 아이템 호피무늬</b>	IP_pau_tshirt_tiger
	<b>진 티셔츠 아이템 무지</b>	IP_jin_tshirt
	<b>진 티셔츠 아이템 호피무늬</b>	IP_jin_tshirt_tiger

## 아이템 관리 에셋

	폴 티셔츠 아이템 무지	IP_pau_tshirt	BI_cmn_tshirt BCI_pau_tshirt CI_cmn_tshirt DI_pau_tshirt
	폴 티셔츠 아이템 호피무늬	IP_pau_tshirt_tiger	BI_cmn_tshirt BCI_pau_tshirt CI_cmn_tshirt_tiger DI_pau_tshirt
	진 티셔츠 아이템 무지	IP_jin_tshirt	BI_cmn_tshirt BCI_jin_tshirt CI_cmn_tshirt DI_jin_tshirt
	진 티셔츠 아이템 호피무늬	IP_jin_tshirt_tiger	BI_cmn_tshirt BCI_jin_tshirt CI_cmn_tshirt_tiger DI_jin_tshirt

## 아이템 관리 예셋

	폴 티셔츠 아이템 무지	IP_pau_tshirt	BI_cmn_tshirt BCI_pau_tshirt CI_cmn_tshirt DI_pau_tshirt
	폴 티셔츠 아이템 호피무늬	IP_pau_tshirt_tiger	BI_cmn_tshirt BCI_pau_tshirt CI_cmn_tshirt_tiger DI_pau_tshirt
	진 티셔츠 아이템 무지	IP_jin_tshirt	BI_cmn_tshirt BCI_jin_tshirt CI_cmn_tshirt DI_jin_tshirt
	진 티셔츠 아이템 호피무늬	IP_jin_tshirt_tiger	BI_cmn_tshirt BCI_jin_tshirt CI_cmn_tshirt_tiger DI_jin_tshirt

같은 메시이므로  
재사용 가능!





# 아이템 관리 에셋

	<p>폴 티셔츠 아이템 무지</p>	<p>IP_pau_tshirt</p>	<p>BI_cmn_tshirt</p>
	<p>폴 티셔츠 아이템 호피무늬</p>	<p>IP_pau_tshirt_tiger</p>	<p>BCI_pau_tshirt</p>
	<p>진 티셔츠 아이템 무지</p>	<p>IP_jin_tshirt</p>	<p>CI_cmn_tshirt</p>
	<p>진 티셔츠 아이템 호피무늬</p>	<p>IP_jin_tshirt_tiger</p>	<p>DI_pau_tshirt</p>

엄밀히 말하면 동일한 메시에 체계 변경 + 피부 텍스처를 변경한 것





같은 메시이므로 재사용 가능!

## 아이템 관리 예셋

	폴 티셔츠 아이템 무지	IP_pau_tshirt	BI_cmn_tshirt BCI_pau_tshirt CI_cmn_tshirt DI_pau_tshirt
	폴 티셔츠 아이템 호피무늬	IP_pau_tshirt_tiger	BI_cmn_tshirt BCI_pau_tshirt CI_cmn_tshirt_tiger DI_pau_tshirt
	진 티셔츠 아이템 무지	IP_jin_tshirt	BI_cmn_tshirt BCI_jin_tshirt CI_cmn_tshirt DI_jin_tshirt
	진 티셔츠 아이템 호피무늬	IP_jin_tshirt_tiger	BI_cmn_tshirt BCI_jin_tshirt CI_cmn_tshirt_tiger DI_jin_tshirt

각각의 캐릭터 설정을  
재사용 가능!

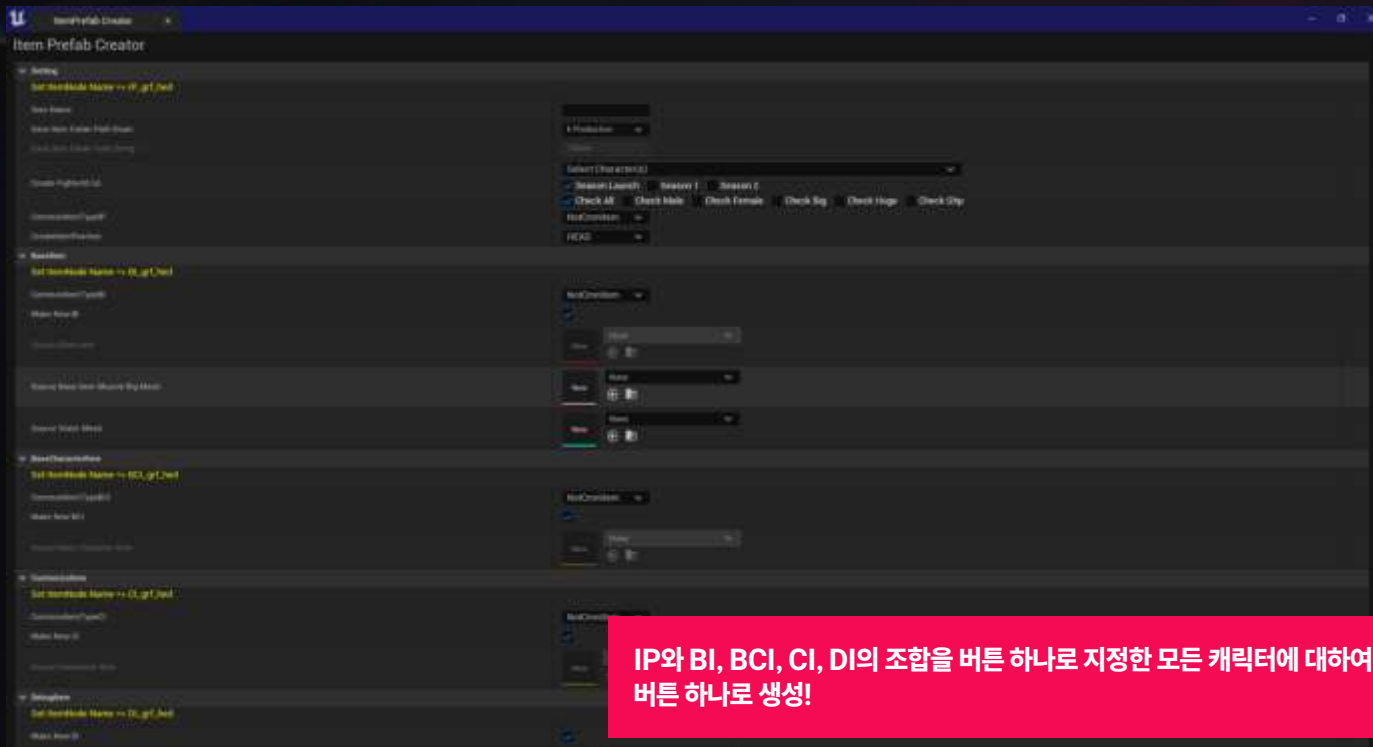
# 아이템 관리 예셋

	<p>폴 티셔츠 아이템 무지</p>	<p>IP_pau_tshirt</p>	<ul style="list-style-type: none"> <li>BI_cmn_tshirt</li> <li>BCI_pau_tshirt</li> <li><b>CI_cmn_tshirt</b></li> <li>DI_pau_tshirt</li> </ul>
	<p>폴 티셔츠 아이템 호피무늬</p>	<p>IP_pau_tshirt_tiger</p>	<ul style="list-style-type: none"> <li>BI_cmn_tshirt</li> <li>BCI_pau_tshirt</li> <li><b>CI_cmn_tshirt_tiger</b></li> <li>DI_pau_tshirt</li> </ul>
	<p>진 티셔츠 아이템 무지</p>	<p>IP_jin_tshirt</p>	<ul style="list-style-type: none"> <li>BI_cmn_tshirt</li> <li>BCI_jin_tshirt</li> <li><b>CI_cmn_tshirt</b></li> <li>DI_jin_tshirt</li> </ul>
	<p>진 티셔츠 아이템 호피무늬</p>	<p>IP_jin_tshirt_tiger</p>	<ul style="list-style-type: none"> <li>BI_cmn_tshirt</li> <li>BCI_jin_tshirt</li> <li><b>CI_cmn_tshirt_tiger</b></li> <li>DI_jin_tshirt</li> </ul>

**컬러, 패턴 변경 정보는 다른 캐릭터에도 재사용 가능!**

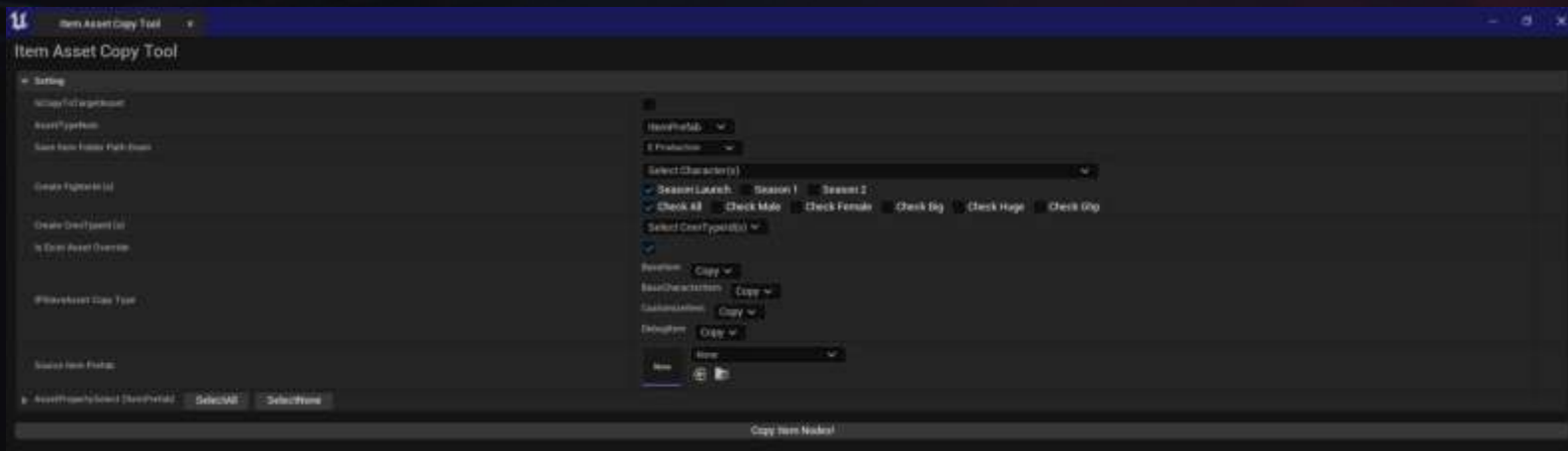
# 아이템 관리 에셋 툴

## ◆ 아이템 관리 에셋 툴을 구축



# 아이템 관리 에셋 툴

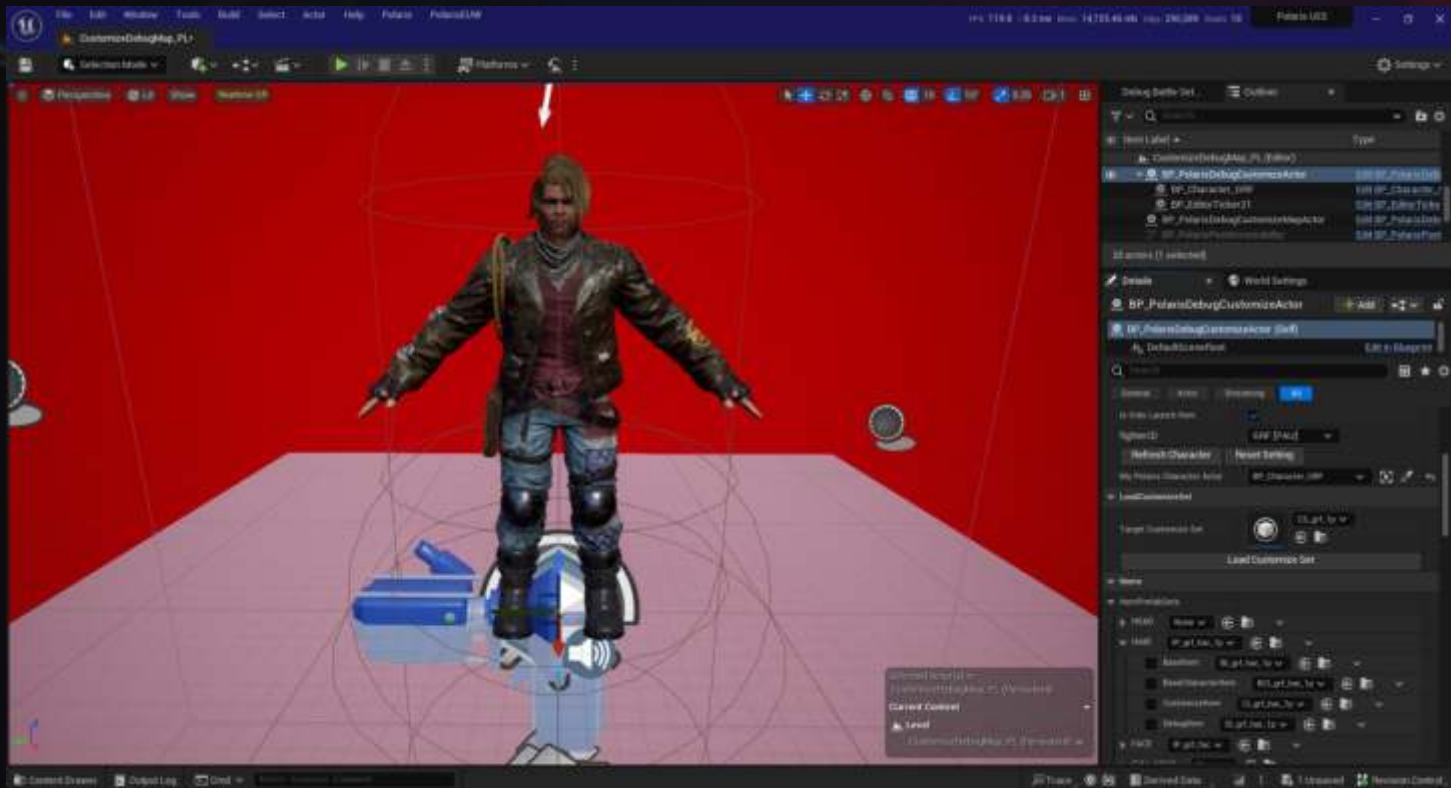
## ◆ 아이템 관리 에셋 카피 툴도 구축

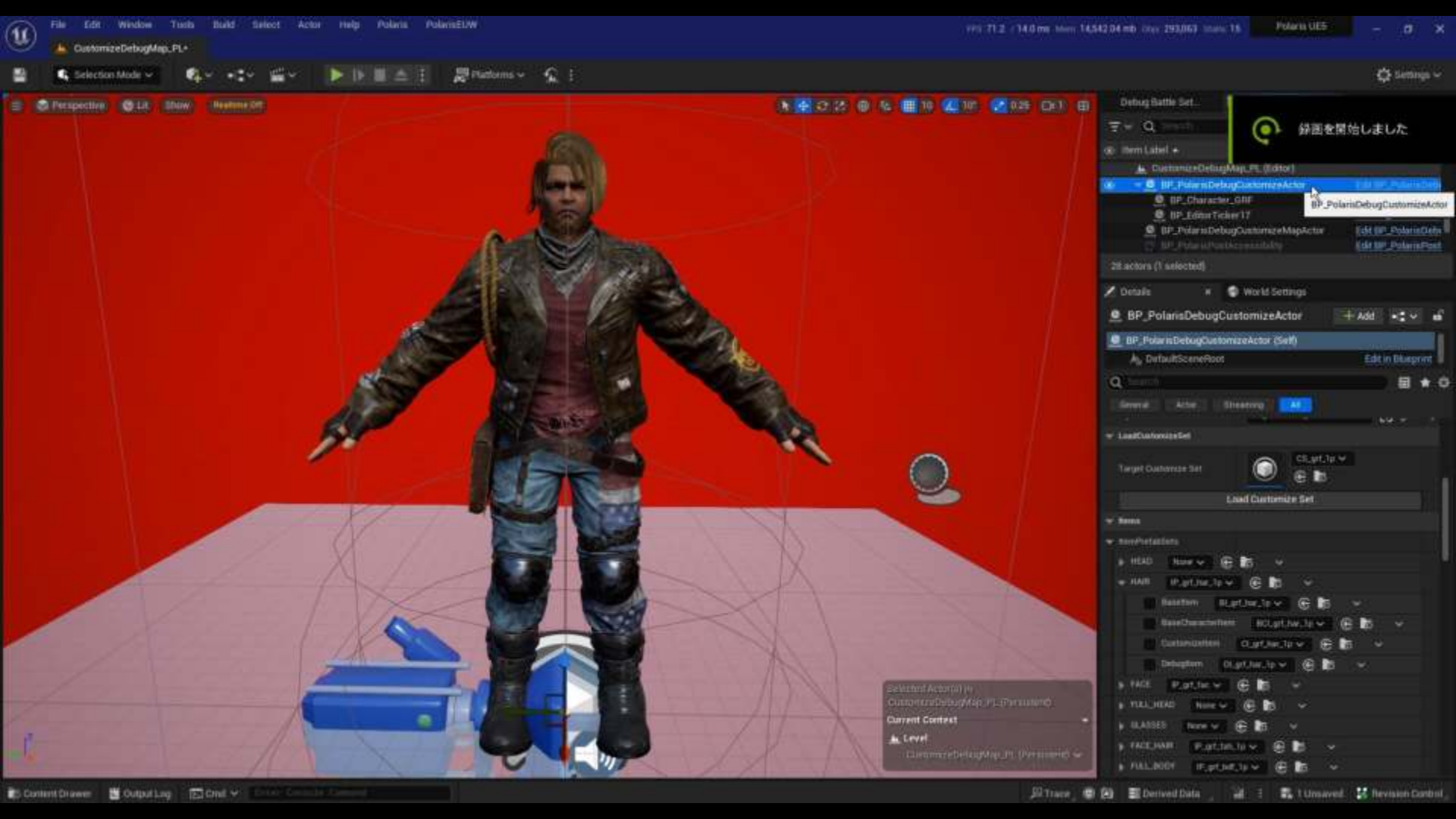


기존 아이템 관리 에셋을 카피해 새로운 아이템에 대한 에셋을 구축

# 아이템 확인 환경

## ◆ CustomizeDebugMap





CustomizeDebugMap\_PL\*

Selection Mode

Settings

Perspective LA Show Restone On

70 30 0.25 1

録画を開始しました

- CustomizeDebugMap\_PL (Editor)
- BP\_PolarisDebugCustomizeActor (Edit BP\_PolarisDe...)
- BP\_Character\_GIF
- BP\_EditorTicker17
- BP\_PolarisDebugCustomizeMapActor (Edit BP\_PolarisDeb...)
- BP\_PolarisPostAccessibility (Edit BP\_PolarisPost...)

28 actors (1 selected)

Details World Settings

BP\_PolarisDebugCustomizeActor + Add

BP\_PolarisDebugCustomizeActor (Self) Edit in Blueprint

DefaultSceneRoot

General Actor Streaming All

LoadCustomizeSet

Target Customize Set CL\_gt\_Lip

Load Customize Set

Bones

nonPhotobable

HEAD None

RIB R\_gt\_rse\_lp

BaseItem R\_gt\_rse\_lp

BaseCharacterItem RCL\_gt\_rse\_lp

Customization CL\_gt\_rse\_lp

DebugItem DL\_gt\_rse\_lp

FACE F\_gt\_fac

YAKL\_HEAD None

GLASSES None

FACE\_HAIR F\_gt\_fac\_lp

FULL\_BODY F\_gt\_fac\_lp

Select Actor in CustomizeDebugMap\_PL (Polaris) Current Context Level CustomizeDebugMap\_PL (Polaris)



DebugBattleSystem (BattleScene) battle\_mode = 4, 909 / drop = 0 / r = 029c052c, 33263 / pl = 0xaffda12a m41 s0, p2 = 0xaffda12a m41 s0  
Frame: 14.57 / 1000 (1.45%) Game: 1.81 / 11.83 (70%) Draw: 4.0 / 4.74 (72%) BHFT: 1.0 / 1.04 (30%) ViewWin: 3320x1040  
DPO.DT: 0.016 / 0.000000 I-Kamu: 2.28 / 2.28 (100%) NetIO GB: 0 / 0 (0%) GC: 0.00 / 0.00 (0%) Key: ----- Mouse: 1710, 471

NVIDIA GeForce RTX 3090  
RenderRes: 1920x1040 Memory: 19,494,30MB used / 19,727,60MB free. VRAM: 6,973 / 11,361,00MB (61%)  
el: 999999(G/71) text: 7

PAUL

NOT LOGGED IN

入門生

トータルダメージ	0
最大コンボダメージ	0
攻撃判定	
ダメージ	0 (0%)
回復可能ダメージ	0 (0%)

CLEAN COUNTER PUNISH WALL

攻撃発生フレーム	- (0 - 0)
硬直差	- (-)
状態	立ち
相手との距離	2.00



PAUL

▼ DebugCustomizeItemChanger (cws.jp)

• Change Fighter

GRF [PAU] ▼ Fighter

CharacterSelectWindow

ChangeTargetPositionWindow

IsFembaSqueeze

IsFembaExclusive

Change Fighter: Reload Fighter

Before Fighter: Next Fighter

Change All Fighter

• Change Costume

CL\_grf\_ip : Costume\_Default0 ▼ Costumes

OnlyJustTimeCostume

KeepCostume

Costume\_Change

Before Costume: Next Costume

IsTesting

• Change Item

▼ Equip\_HEAD...

HEAD

None ▼

Equip: Before: Next

ColorDebug

Visible

HAIR

IP\_grf\_her\_allback ▼

Equip: Before: Next

ColorDebug

Visible

FACE

IP\_grf\_fac ▼

Equip: Before: Next

ColorDebug

Visible

FULL\_HEAD

None ▼

Equip: Before: Next

ColorDebug

Visible

GLASSES

None ▼

Equip: Before: Next

ColorDebug

Visible

FACE\_HAIR

IP\_grf\_fah\_3p ▼

Equip: Before: Next

ColorDebug

Visible

Item\_BODY

FULL\_BODY

IP\_grf\_btf\_ip\_v2 ▼

Equip: Before: Next

ColorDebug

Visible

UPPER

None ▼

Equip: Before: Next

ColorDebug

Visible

## 제작 · 양산 환경 정비 소감

### 좋았던 점

- ◆ 아이템 제작 비용을 대폭 절감할 수 있었음
  - 공유 아이템으로 에셋 재사용이 가능해지면서 양산이 용이해짐
    - 아이템 총 12,000개 중 약 10,000개가 공유 아이템
  - 에셋 제작 후 유지보수 시 수정하는 에셋 수도 줄일 수 있었음
- ◆ 아이템 확인 환경도 호평을 받음
  - VA의 요구사항에 세밀하게 대응, 구현 확인보다 모델 제작에 집중할 수 있게 됨
  - 레벨에 의존하지 않게 되어 확장성도 보장됨

## 제작 · 양산 환경 정비 소감

---

### 아쉬웠던 점

- ◆ 에셋의 절대 수가 늘어남
  - 에셋 추가를 툴로 서포트하여 커버
    - 버튼 하나로 한 번에 32 캐릭터 분량의 에셋 구축이 가능
  
- ◆ 러닝 코스트가 다소 높아짐
  - 프로젝트 자체 에셋이 늘어나버림
    - 꼼꼼한 도큐먼트 준비를 통해 커버
    - 익숙해지면 양산 시 든든한 무기가 됨

## 4. 아이템 렌더링 처리 구현

## 아이템 렌더링 처리에 대한 요구사항

---

- ◆ 철권 7보다 더 많은 곳에서 컬러 변경을 희망함(1메시에 대하여 30개 이상)

➡ 다영역 표현이 가능한 IDMap을 고려할 필요가 있음

- ◆ 캐릭터 표현의 향상과 IDMap의 디코딩 등으로 인해 셰이더 부하가 높아질 우려

➡ 베이크 처리를 통해 런타임 머티리얼 계산 부하를 경감

- ◆ 차세대 철권의 감각이 느껴지는 머티리얼 표현 구현

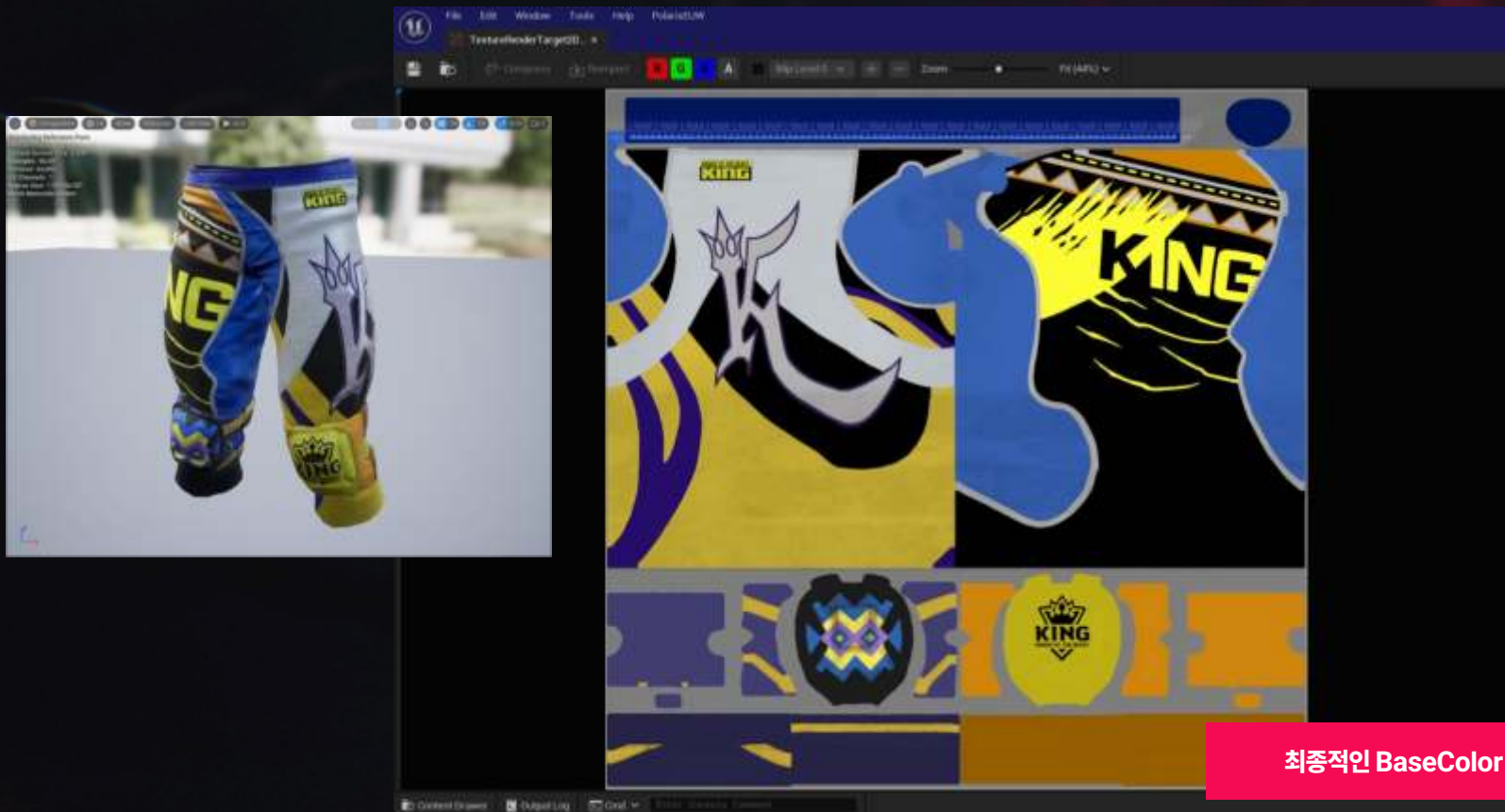
➡ 각 사양에 맞는 머티리얼 구축

## IDMap에 대하여

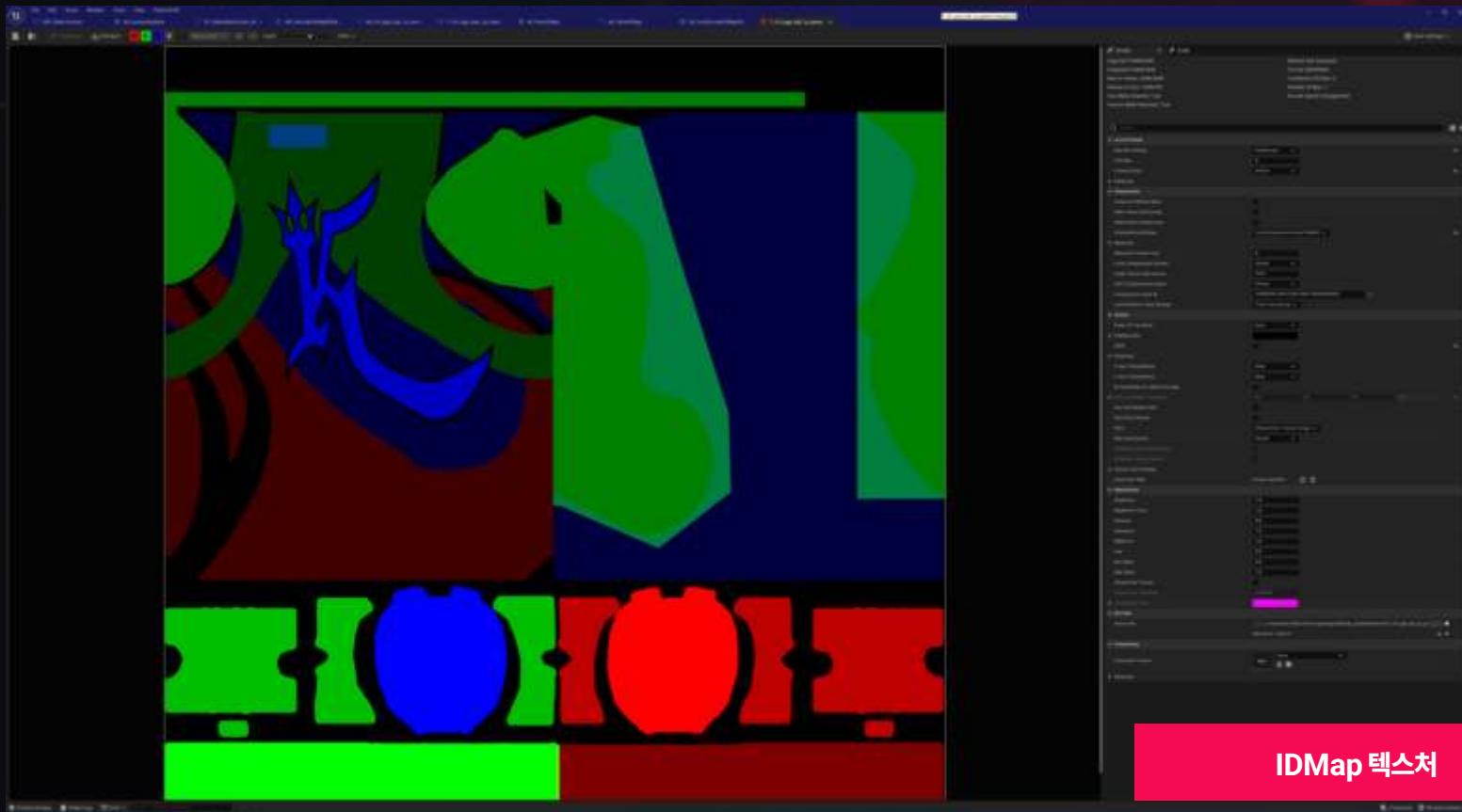
- ◆ RGBA의 각 채널을 값 영역으로 분할하고, 각 값 영역을 별도의 ID로 설정하여 다중 ID화에 대응
  - 값 영역의 폭과 추출할 맵은 머티리얼 파라미터로 지정
    - SplitNumberPerChannel, SplitIndex, ChannelMask
  - 머티리얼 측에서는 각 채널당 4개까지의 범위를 취급, 4 범위 × 4 채널로 16개의 ID를 할당할 수 있도록 함



# IDMap에 대하여



# IDMap에 대하여



IDMap 텍스처

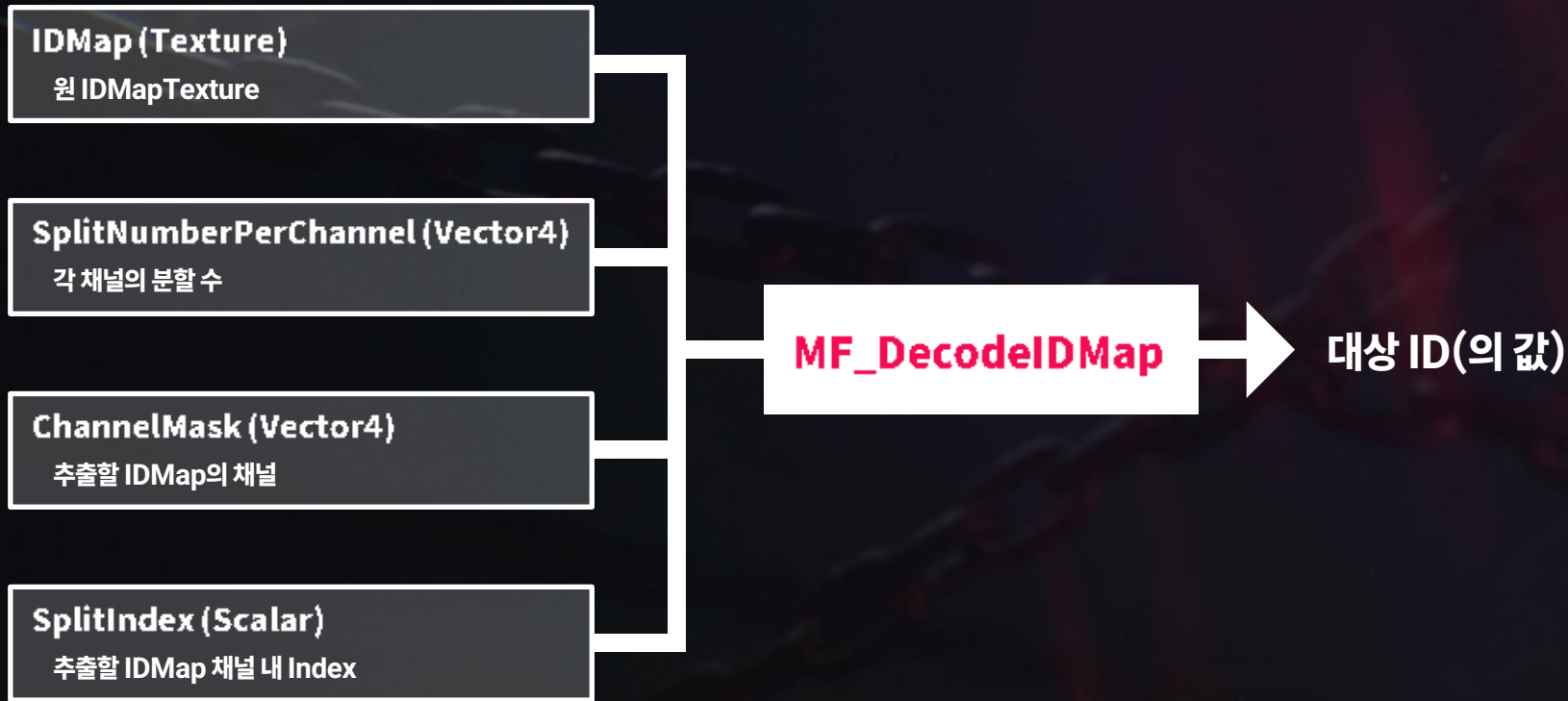
# IDMap에 대하여



ID\_3로 추출

# IDMap에 대하여

## ◆ IDMap의 Decode



# IDMap에 대하여

## ◆ SplitNumberPerChannel : 텍스처의 각 채널을 몇 개의 ID로 분할하여 사용할 것인가?

- 예시: SplitNumberPerChannel (4, 4, 4, 4)

R	G	B	A
255 ID_3	255 ID_7	255 ID_11	255 ID_15
192	192	192	192
191 ID_2	191 ID_6	191 ID_10	191 ID_14
128	128	128	128
127 ID_1	127 ID_5	127 ID_9	127 ID_13
64	64	64	64
63 ID_0	63 ID_4	63 ID_8	63 ID_12
0	0	0	0

# IDMap에 대하여

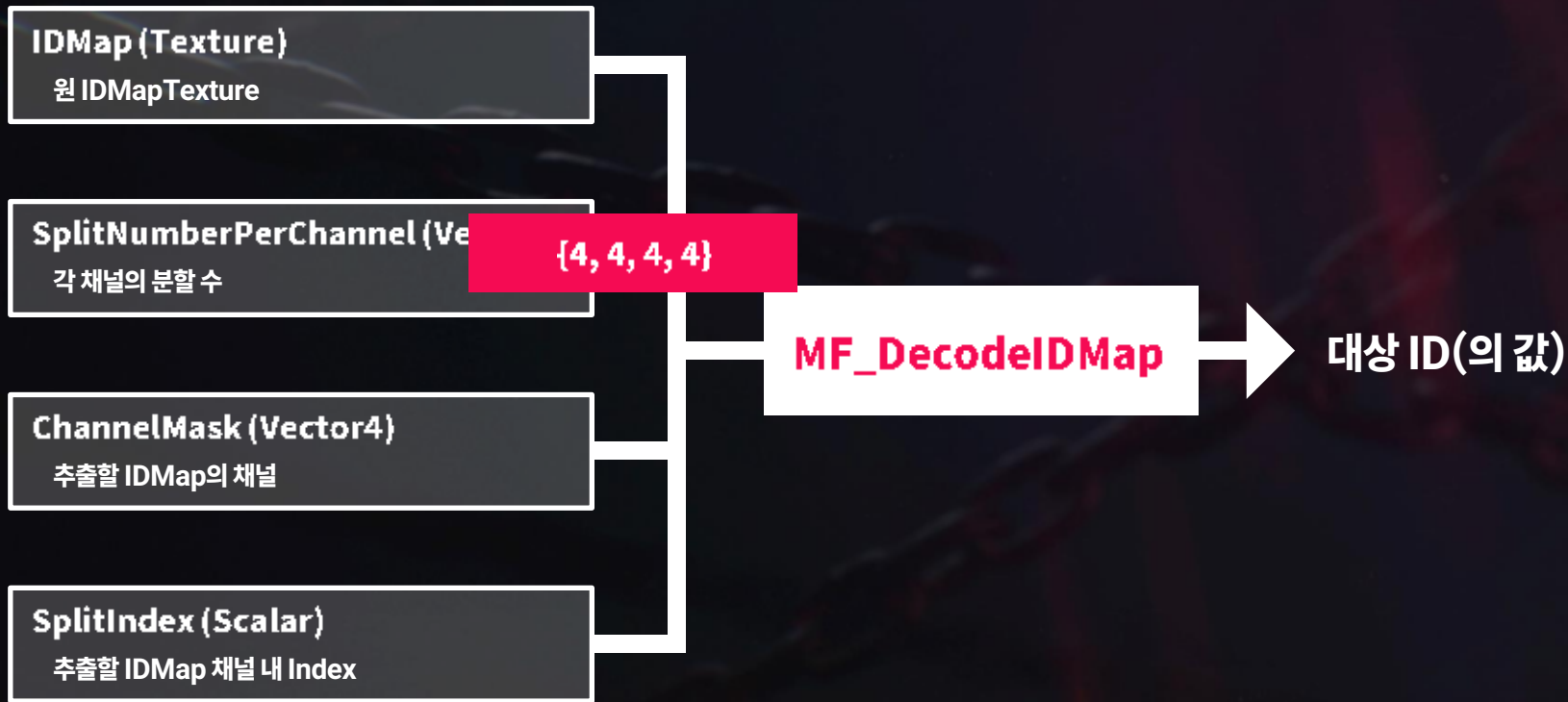
◆ SplitNumberPerChannel : 텍스처의 각 채널을 몇 개의 ID로 분할하여 사용할 것인가?

● 예시: SplitNumberPerChannel (2, 3, 4, 1)

R	G	B	A
255	254	255	ID_12
ID_1	ID_6	ID_11	
	170	192	
128	176	ID_10	
ID_0	ID_5	128	
	85	ID_9	
	84	127	
	ID_4	64	
0	0	ID_8	
		63	
		0	

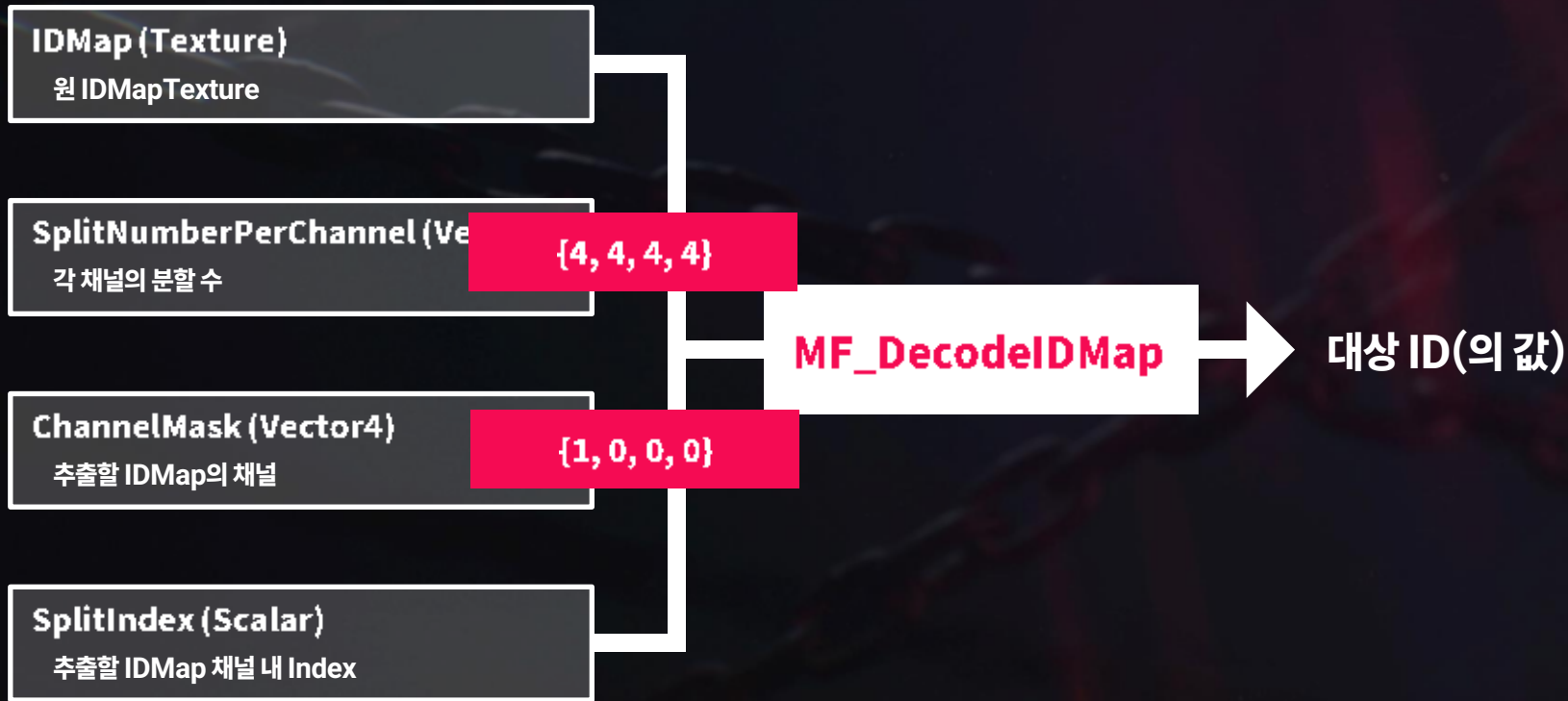
# IDMap에 대하여

## ◆ IDMap의 Decode



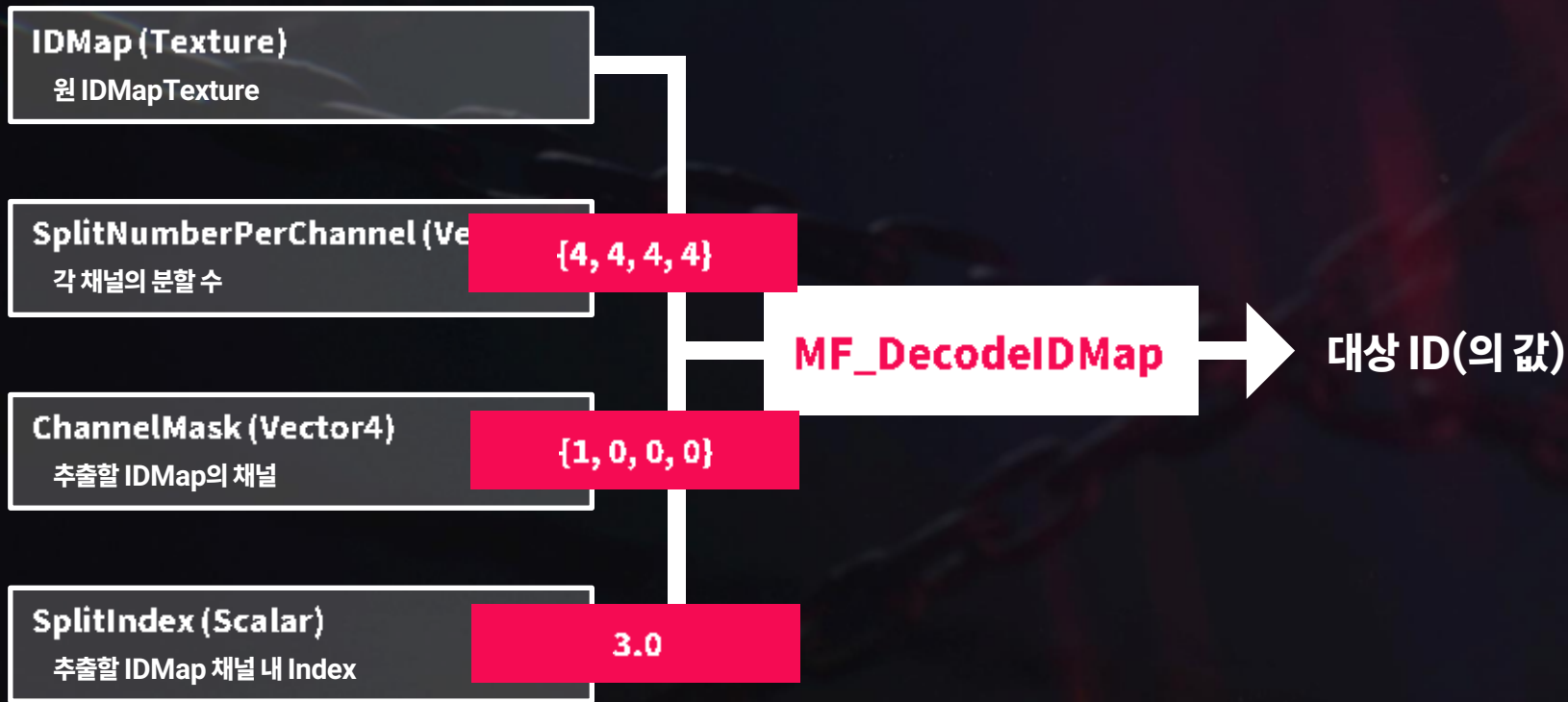
# IDMap에 대하여

## ◆ IDMap의 Decode



# IDMap에 대하여

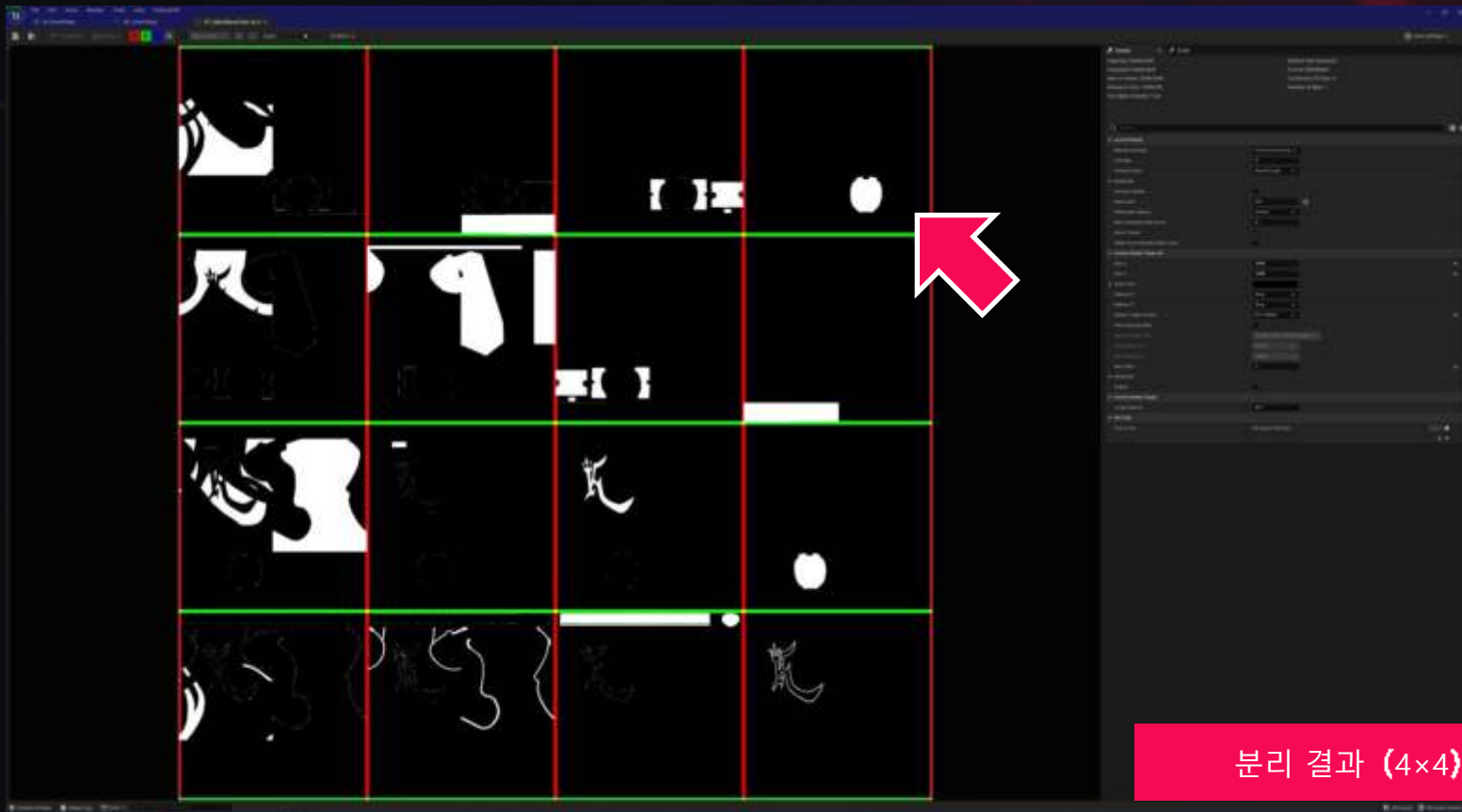
## ◆ IDMap의 Decode



# IDMap에 대하여



# IDMap에 대하여



분리 결과 (4x4)

# IDMap 소감

---

## 좋았던 점

- ◆ 1장의 IDMap과 1개의 Vector4(분할 정보)로 다영역 표현이 가능해짐
  - 철권 7은 4곳 → 철권 8은 16곳이 하나의 텍스처로 변경 가능해짐
- ◆ 분할 수에 좌우되기는 하지만 강도 표현도 가능함

# IDMap 소감

## 아쉬웠던 점

- ◆ 최종적으로 런타임 머티리얼에서도 사용하게 된 점
  - IDMap은 베이크 시에만 사용하려고 했으나 DetailMap의 적용 범위를 ID별로 설정할 필요가 생기면서 런타임 머티리얼에서도 IDMap이 필요해짐
    - 런타임 머티리얼의 부하에 IDMap의 디코딩이 추가됨
- ◆ Mipmap이 현실적으로 사용 불가했음
  - 영역 경계 부근에서 서로 다른 영역을 표현하는 값이 되어 버림
- ◆ 강도 표현이 거칠어질 가능성
  - 분할 수가 많은 채널에서는 강도 표현이 거칠어지게 됨



## 텍스처 베이킹에 대하여

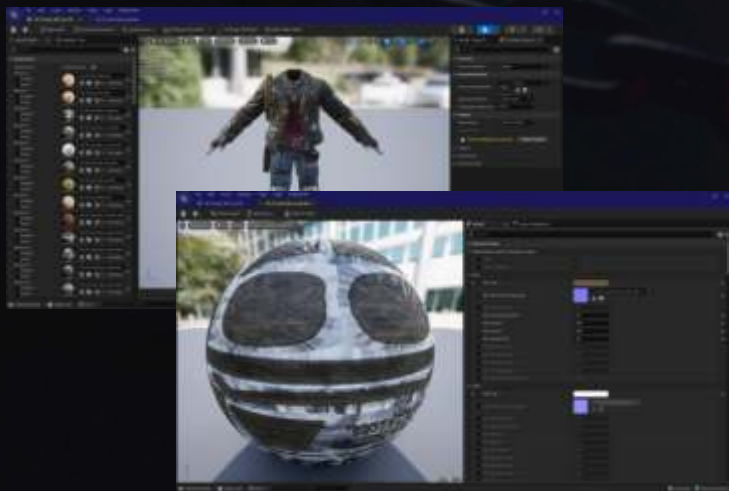
- ◆ 피부 계산 결과도 텍스처에 베이킹하여 사용
  - 얼굴용 커스터마이징이 다수 존재하여, 표현 품질 향상을 위해 베이킹 처리를 채택함
    - 메이크업(페이스 페인트, 아이메이크업, 립, 블러셔), 태닝이 존재



# 텍스처 베이킹에 대하여

## ◆ 프리뷰 머티리얼과 런타임 머티리얼

프리뷰 머티리얼



➤ 에디터 작업용 머티리얼

런타임 머티리얼



➤ 실제 인게임에서 사용되는 머티리얼

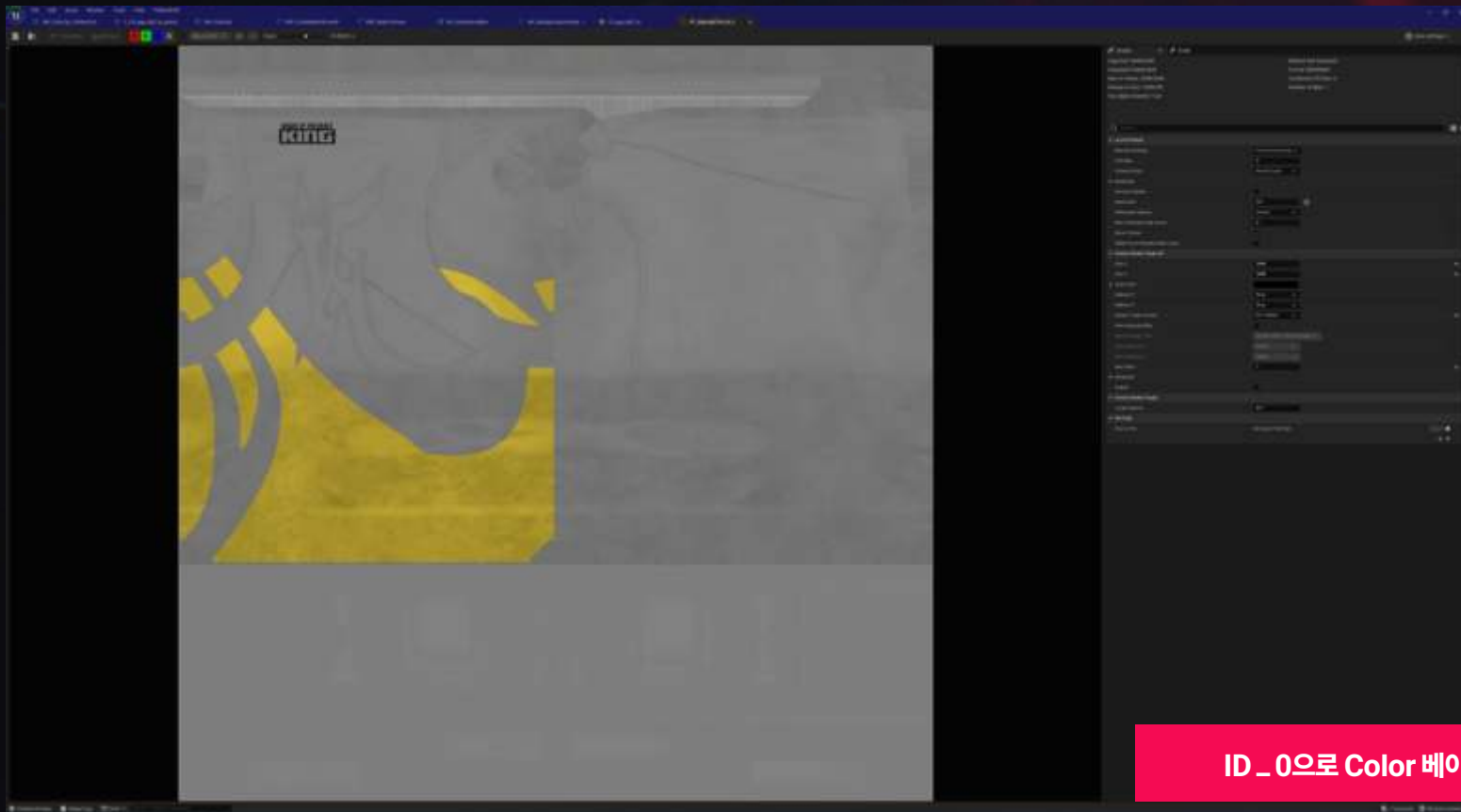
베이킹 처리

## 텍스처 베이킹에 대하여



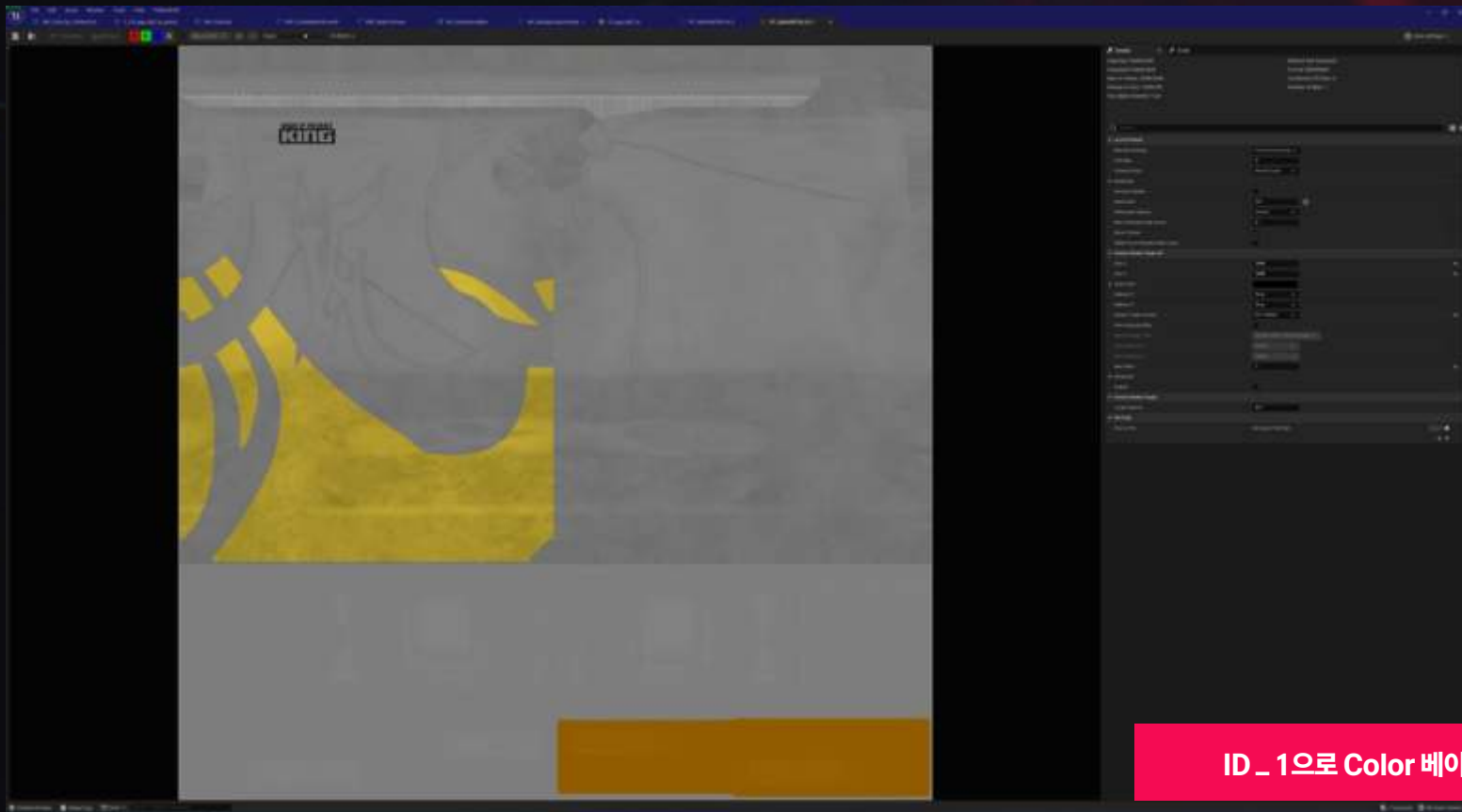
원래의 BaseColor 텍스처

## 텍스처 베이킹에 대하여

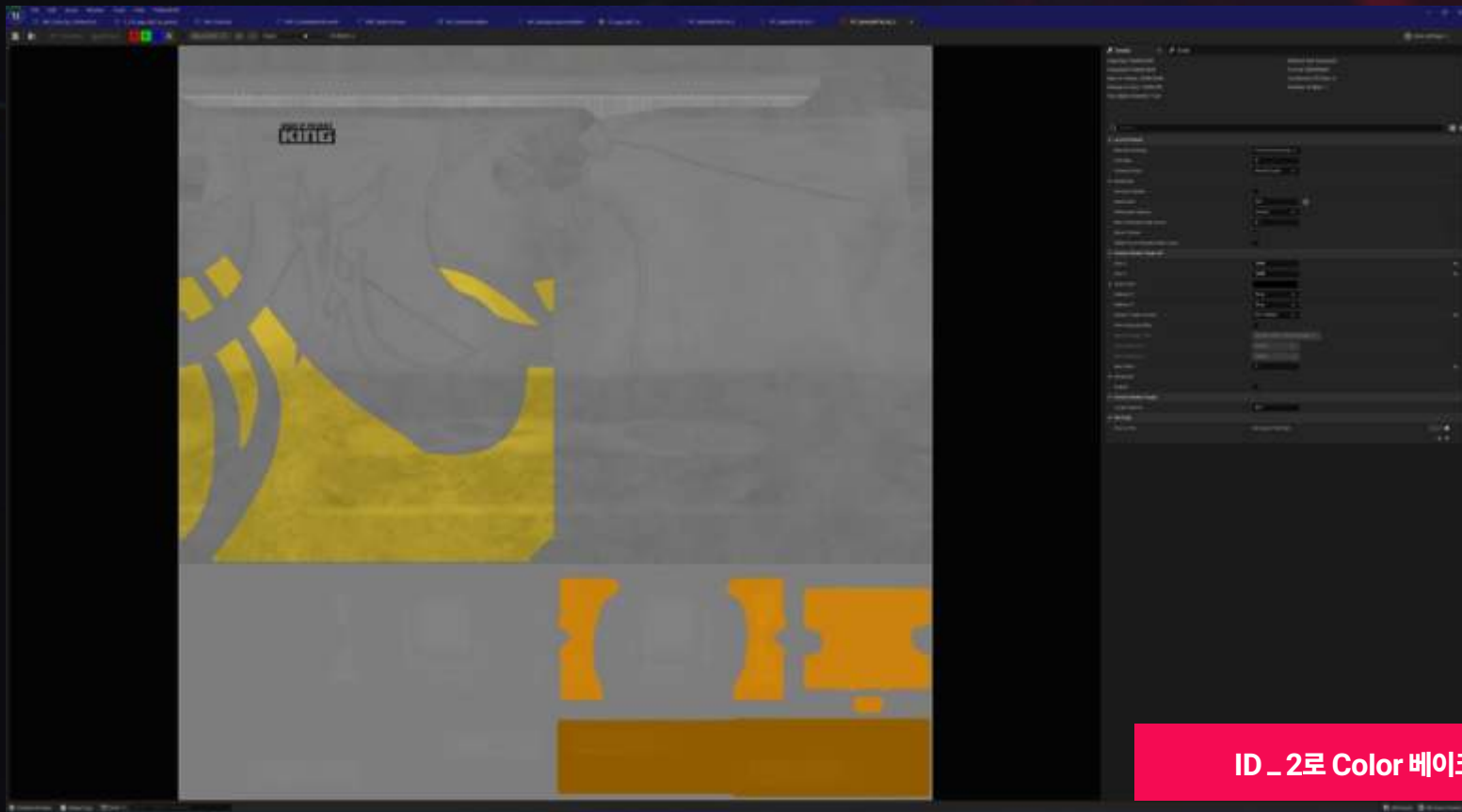


ID \_ 0으로 Color 베이킹

## 텍스처 베이킹에 대하여

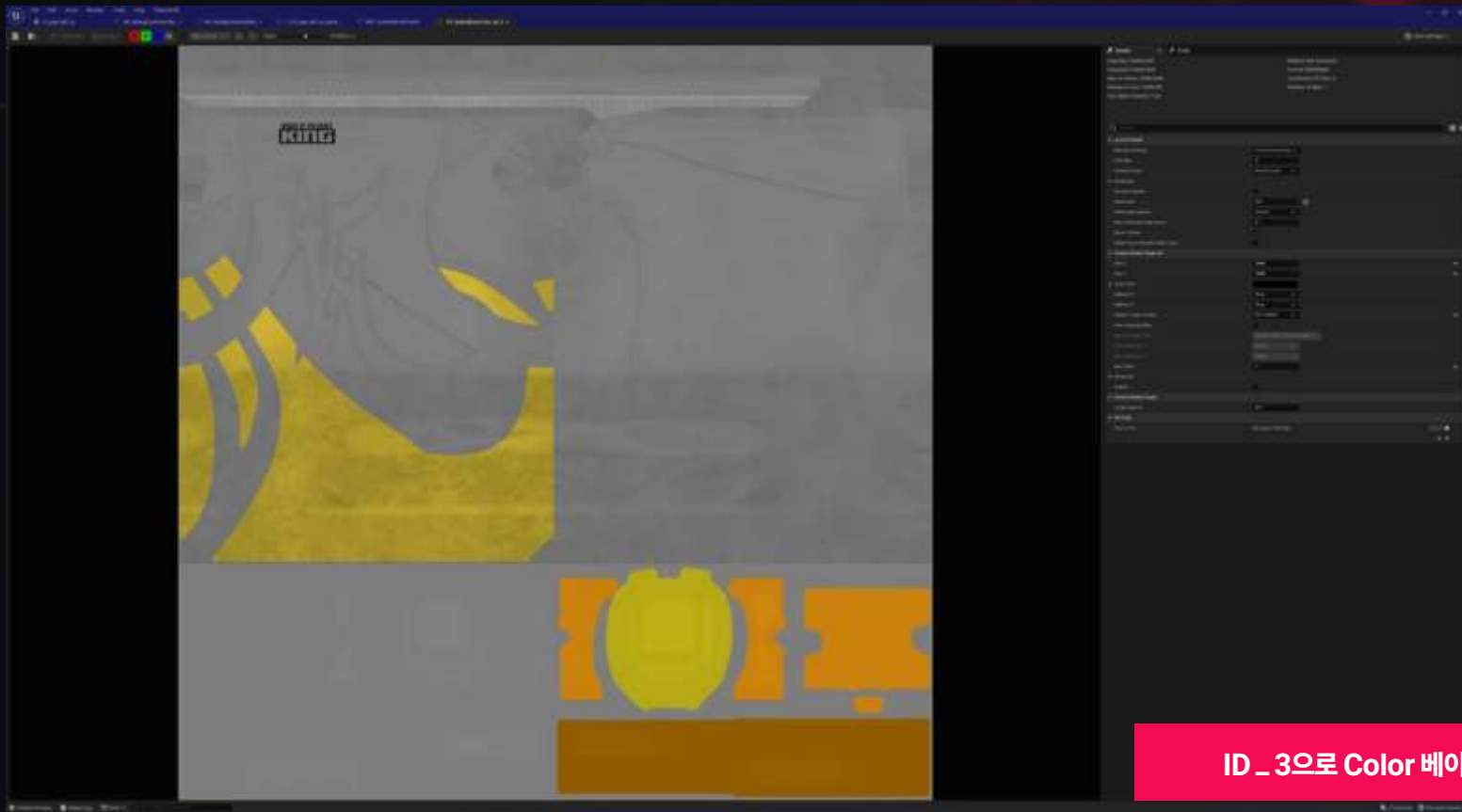


## 텍스처 베이킹에 대하여



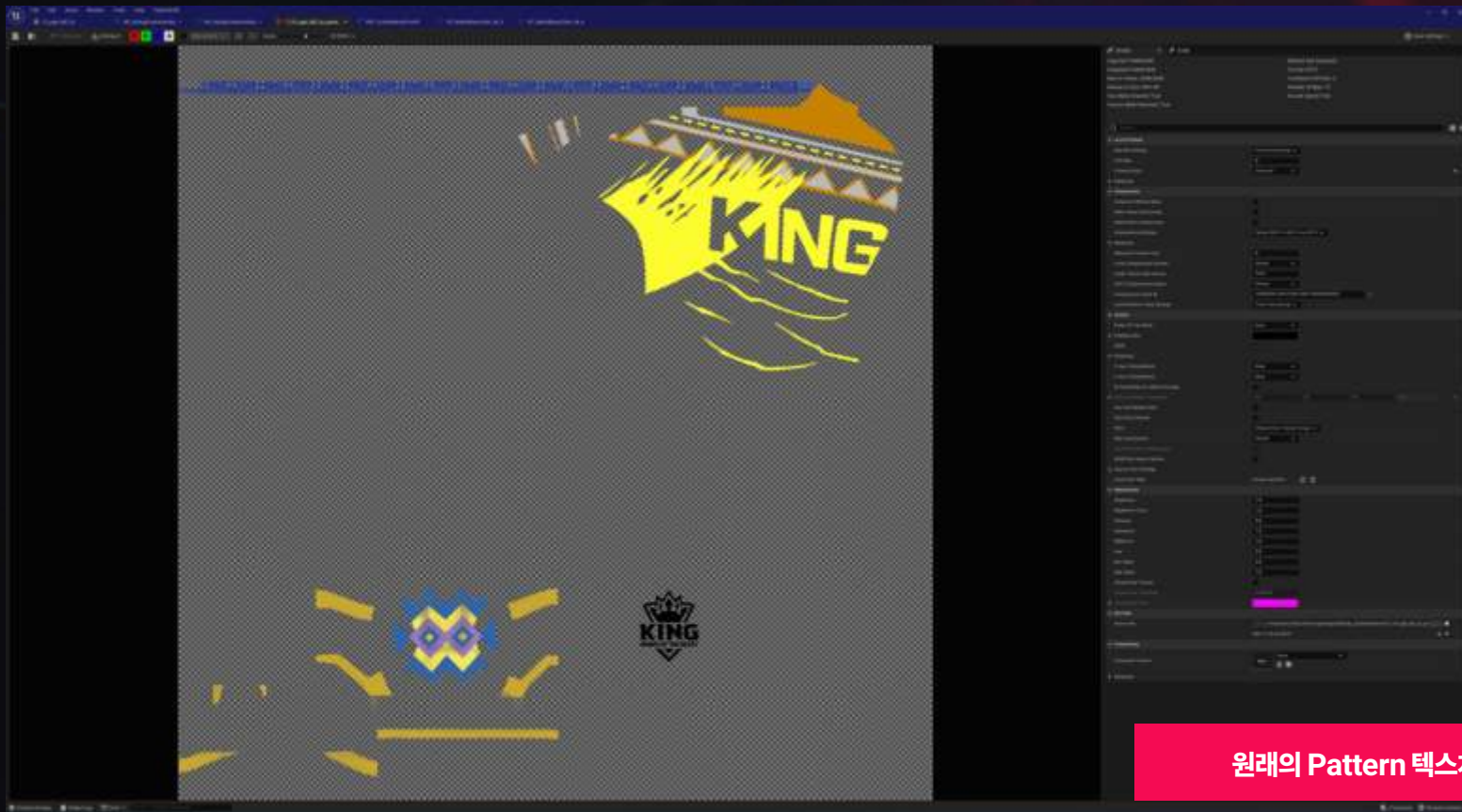
ID\_2로 Color 베이킹

## 텍스처 베이킹에 대하여



ID \_ 3으로 Color 베이킹

# 텍스처 베이킹에 대하여



원래의 Pattern 텍스처

## 텍스처 베이킹에 대하여



Color ID 3으로 Pattern 베이킹

## 텍스처 베이킹에 대하여



베이킹 처리 최종 결과

## 텍스처 베이크 소감

---

### 좋았던 점

- ◆ 런타임 부하 경감은 계획대로 성공
  - 런타임 머티리얼에 결과가 저장되어 있기 때문에 런타임에 컬러, 패턴 계산이 불필요
- ◆ 다양한 표현에 대응 가능
  - 컬러, 패턴 커스터마이징 외에 메이크업 등의 표현이 가능해짐
- ◆ 계산 결과의 재사용
  - 컬러, 패턴 커스터마이징을 하고 계산한 결과를 그대로 텍스처로 취급할 수 있기 때문에 pub 모델용으로 출력하여 재사용이 가능

## 텍스처 베이크 소감

---

### 아쉬웠던 점

#### ◆ 프리뷰 머티리얼과의 외형적 차이

- 차이를 베이크 쪽에서 맞췄기 때문에 베이크 횟수가 증가함

#### ◆ RenderTarget을 따로 가질 필요가 있음

- RGBA8 등 RenderTarget화된 텍스처만큼의 메모리 사용량이 증가함

#### ◆ 베이크 처리에 시간이 소요됨

- 캐릭터 로딩 시간에 영향
- 캐릭터 커스터마이징 컬러 변경 시에도 영향

# 머티리얼 표현

## ◆ 아이템 머티리얼

- 캐릭터가 입는 옷이나 장비 등에 사용되는 기본적인 머티리얼
- 베이크 처리의 대상
  - BaseColor
  - Normal
  - RMA (Roughness, Metalness, AmbientOcclusion)
- 기능
  - Iridescent(무지갯빛) 표현
  - 스타킹 표현
  - SSS
  - 환경 맵
  - 이미시브(애니메이션 가능)
  - 레이지 처리
  - 젖음, 오염

등 Base 머티리얼에서 파생된 약 30여 종



## 머티리얼 표현

### ◆ 아이템 머티리얼

- **오염 표현**
  - 마스크를 사용한 오염 기능
  - 유사 프로시저럴로 오염 방식을 파라미터로 변경 가능
- **젖음 표현**
  - BaseColor에 1.0보다 낮은 수치를 곱하여 어둡게 하고, Roughness를 낮춰서 반사 강도를 높여서 표현



# 머티리얼 표현

## ◆ 피부 머티리얼

- 캐릭터의 피부에 사용되는 머티리얼
- 베이크 처리 대상
  - BaseColor x2
  - Normal, Normal (NoWrinkle)
  - RAD (Roughness, AmbientOcclusion, Dirt)
  - MTM (Metalness, Translucency, Mask)
  - ScalpMask
- 기능
  - 표정 주름, 근육 주름
  - 레이지 중 Fresnel 처리
  - 태닝 처리
  - 땀, 습기, 얼룩
  - 두피 마스크
  - 데칼 기능



# 머티리얼 표현

## ◆ 피부 머티리얼

### ● 얼굴 모드

- 표정 주름, 근육 주름이 유효해짐
- 텍스처 수가 늘고 부하가 커짐

### ● 얼굴 모드의 NormalMap에 대하여

- 일반 주름, 무표정 주름, 깊은 주름으로 분류
- Facial Mask 맵과 웨이트값을 사용하여 얼굴의 부위별로 주름의 정도를 컨트롤



×



=



## 머티리얼 표현

### ◆ 피부 머티리얼

- **땀 처리**
  - UV 스크롤 사용
- **격투 자세 때 땀이 팔꿈치에서 손 방향으로 올라감**
  - 프로그램 측에서 팔뚝의 각도를 보고 UV의 속도를 0으로 설정
  - 땀의 속도를 서서히 늦추기 위해 캐릭터 BP 측에서 Tick으로 증가하는 시간변수를 만들어 알림



## 머티리얼 표현

### ◆ 헤어 머티리얼

- 캐릭터의 머리카락에 사용되는 머티리얼
  - Alpha, Depth, ID를 한 장으로 묶어 Root + ADI 맵 2장의 텍스처로 운영
- 헤어 컬러 지정
  - 기본은 뿌리, 끝 2가지 컬러 지정 가능, 예외적으로 중간을 추가하여 최대 3가지 컬러 지정 가능
  - 1색을 지정하면 다른 1색을 잘 어울리는 느낌의 컬러로 만들어주는 SubColor 생성 기능
- ScalpMask
  - 헤어 컬러에 연동하여 두피 부분의 컬러를 바꾸기 위해 스킨 베이크를 실행



## 머티리얼 표현

### ◆ 안구 머티리얼

#### ● 캐릭터의 안구에 사용되는 머티리얼

- 안구 머티리얼
- 눈물 머티리얼
- BlurAO 머티리얼

#### ● 기능

- 2D 텍스처를 이용한 페이크 스펙큘러
- Iris(홍채) 컬러 변경 기능
- 펑 웨이의 레이저 아츠 시 세로 눈동자에 대응 (Iris, Sclera 별도의 UV 컨트롤 추가)



## 머티리얼 표현

### ◆ 털 머티리얼

- 캐릭터의 털 표현에 사용되는 머티리얼
- gFur 플러그인
  - 주로 동물의 털에 사용
  - 플러그인 제공원은 부하가 높기 때문에 경량화 실시 + Mask 처리 및 Depth 패스 최적화
  - UE5 대응이 늦어 사내에서 Integrate 대응 (현재는 공식 대응 완료)



## 머티리얼 표현

### ◆ 털 머티리얼

- 캐릭터의 털 표현에 사용되는 머티리얼
- 간이 BlurFur
  - 플로우맵과 방사형 블러(Radial Blur)를 사용
  - 처리 부하는 가벼운 편



## 머티리얼 표현

### ◆ 털 머티리얼

- 캐릭터의 털 표현에 사용되는 머티리얼
- Parallax Occlusion Mapping Fur
  - Parallax Occlusion Mapping을 사용
  - Shell 기법을 POM으로 유사하게 표현
  - 처리부하는 단순 Blur Fur보다 약간 무거운 편



# 머티리얼 표현 소감

## 좋았던 점

- ◆ 소재별 다양한 표현을 도입함
  - 텍스처 소재 등을 물리적으로 올바르게 구현할 수 있게 됨 = 라이팅 표현도 올바르게 구현함
  - 소재가 구비되어 있어 캐릭터별 추가 사양 및 변경 대응이 용이해짐
- ◆ Unreal Engine의 기존 기능이 풍부함 + 강력함
  - 기본적으로 제공되는 기능의 조합으로 표현이 가능했음
  - 엔진 커스터마이징은 거의 하지 않음
  - 철권 7 시절부터 개발하여 Unreal Engine 자체에 대한 지식도 풍부했음

## 머티리얼 표현 소감

---

### 아쉬운 점

- ◆ 다양한 기능이 포함된 머티리얼의 자식이 되는 MI가 많아짐
  - 기능이 풍부해 돌려쓰기가 용이해졌기 때문
  - 자식이 되는 MI의 수에 비례하여 수정 시 ShaderCompile 시간이 길어짐
- ◆ 부하가 큰 머티리얼이 많아짐
  - 표현에 대한 고도의 요구사항에 비례하여 부하가 증가
  - 프로젝트의 드로잉 엔지니어와 상의

## 5. 요약과 정리

## 제작 · 양산 환경 요약 및 정리

---

- ◆ 『커스터마이징 아이템』에 필요한 정보를 세분화하여 공유 가능한 부분은 공유
  - **파라미터를 가진 에셋 수 줄이기**를 의식함
    - 아이템 개수만큼 있는 ItemPrefab은 실제 값을 가진 각 관리 에셋에 대한 참조만 가지도록 함
    - 같은 설정을 가급적 재사용함으로써 수정할 대상을 줄이고자 함
  - **에셋 추가는 가급적 수동으로 하지 않는다**
    - 수동으로 추가하거나 복사하면 실수가 발생하기 쉬움
- ◆ 아이템의 확인 환경은 철저한 의견 청취 & 가능한 한 오더에 대응하고자 함
  - **좋은 확인 환경이 제작 가능한 아이템 개수로 연결됨**

## 렌더링 처리 요약과 정리

---

### ◆ 1개의 머티리얼에서 다영역의 컬러 변경이 가능한 IDMap 구축

- 철권 7 시대의 4배에 달하는 **16개의 영역을 확보**할 수 있게 됨

### ◆ 런타임 상의 머티리얼 처리 부하 경감을 위해 텍스처 베이크를 실시

- **부하 경감은 성공**, 피부 머티리얼에 대한 베이크는 표현력 향상으로 이어짐

### ◆ 소재에 부합한 텍스처 · 머티리얼 조합이 올바른 표현으로 연결됨

- **올바른 머티리얼 표현 = 올바른 라이팅으로 연결됨**
  - 보다 차세대 감각의 머티리얼 표현으로
- 캐릭터 추가에 따른 표현 추가 및 변경에도 쉽게 대응 가능



BANDAI NAMCO

# Thank you!

— Team Tekken

