

언리얼 엔진 4 게임 최신 기능과 최적화

에픽게임즈 코리아
서포트 엔지니어 박창현



목차

프로파일링

- Stat 명령어
- ProfileGPU 명령어

최적화

- 기본 점검 - 렌더러, 피쳐레벨, 엔진 퀄리티
- 플랫폼별 다른 프로퍼티 설정
- 블루프린트 네이티브화
- 애니메이션 최적화
- 템포럴 업샘플 & 동적 해상도

최신 기능

- 라이브 코딩 / 빌드 타임 단축(3x 빠름)
- 카오스
- 버추얼 텍스처링
- 리얼타임 레이트레이싱
- 언리얼 인사이트



프로파일링

Stat 명령어 / ProfileGPU



Stat 명령어

- Stat Unit / UnitGraph

```
Frame: 3.61 ms
Game: 5.68 ms
Draw: 3.21 ms
GPU: 3.61 ms
RHIT: 3.49 ms
DynRes: Unsupported
```

Frame: 하나의 프레임 생성에 걸린 총 시간.

- Game: Stat Game
- Draw: Stat SceneRendering
- GPU: Stat GPU
- RHIT: Stat RHI

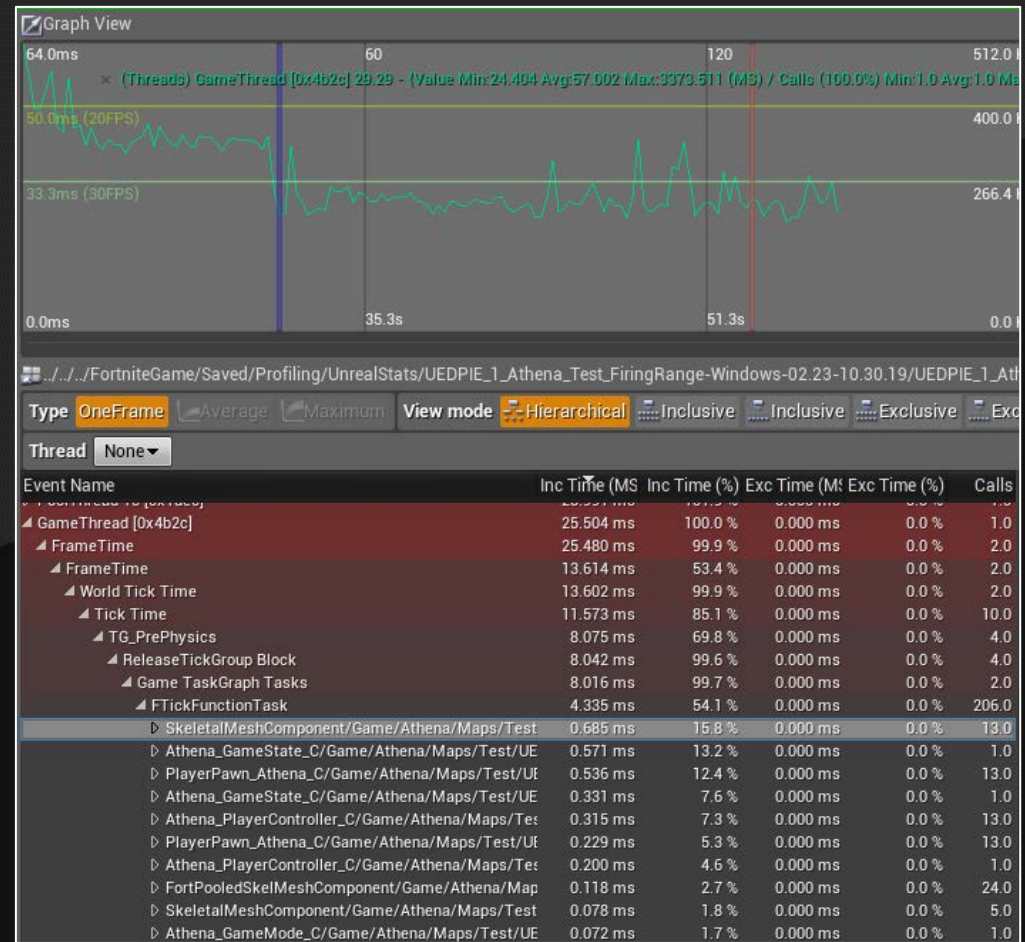
Game [STATGROUP_game]					
Cycle counters (flat)					
	Call Count	InclusiveAvg	InclusiveMax	ExclusiveAvg	ExclusiveMax
World Tick Time	1	0.24 ms	0.29 ms	0.02 ms	0.02 ms
Tick Time	6	0.19 ms	0.24 ms	0.00 ms	0.00 ms
Post Tick Component Update	6	0.07 ms	0.07 ms	0.00 ms	0.00 ms
Transform or RenderData	7	0.06 ms	0.06 ms	0.00 ms	0.00 ms
Queue Ticks	2	0.01 ms	0.01 ms	0.01 ms	0.01 ms
TickableGameObjects Time	2	0.01 ms	0.01 ms	0.01 ms	0.01 ms
GT Tickable Time	1	0.01 ms	0.01 ms	0.00 ms	0.00 ms
Nav Tick Time	1	0.01 ms	0.01 ms	0.00 ms	0.00 ms
Finish Async Trace Time	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Reset Async Trace Time	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Net Tick Time	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Cooldown Dequeuing	4	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Blueprint Latent Actions	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Net Broadcast Tick Time	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Update Camera Time	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
MoveComponent(Primitive) Time					
Recreate					
Counters		Average	Max	Min	
Ticks Queued		21.00	21.00	21.00	
TimerManager Heap Size		2.00	2.00	2.00	



Stat 명령어 – Stat Startfile / Stopfile

언리얼 엔진의 여러 스탯 기록

- 스탯 파일 생성.
- 세션 프론트엔드의 Stat Viewer로 확인.
- 어디에 유용?
 - 일반적인 성능 이슈 파악
 - 히치 추적
 - 불필요하게 Tick된 객체 찾기

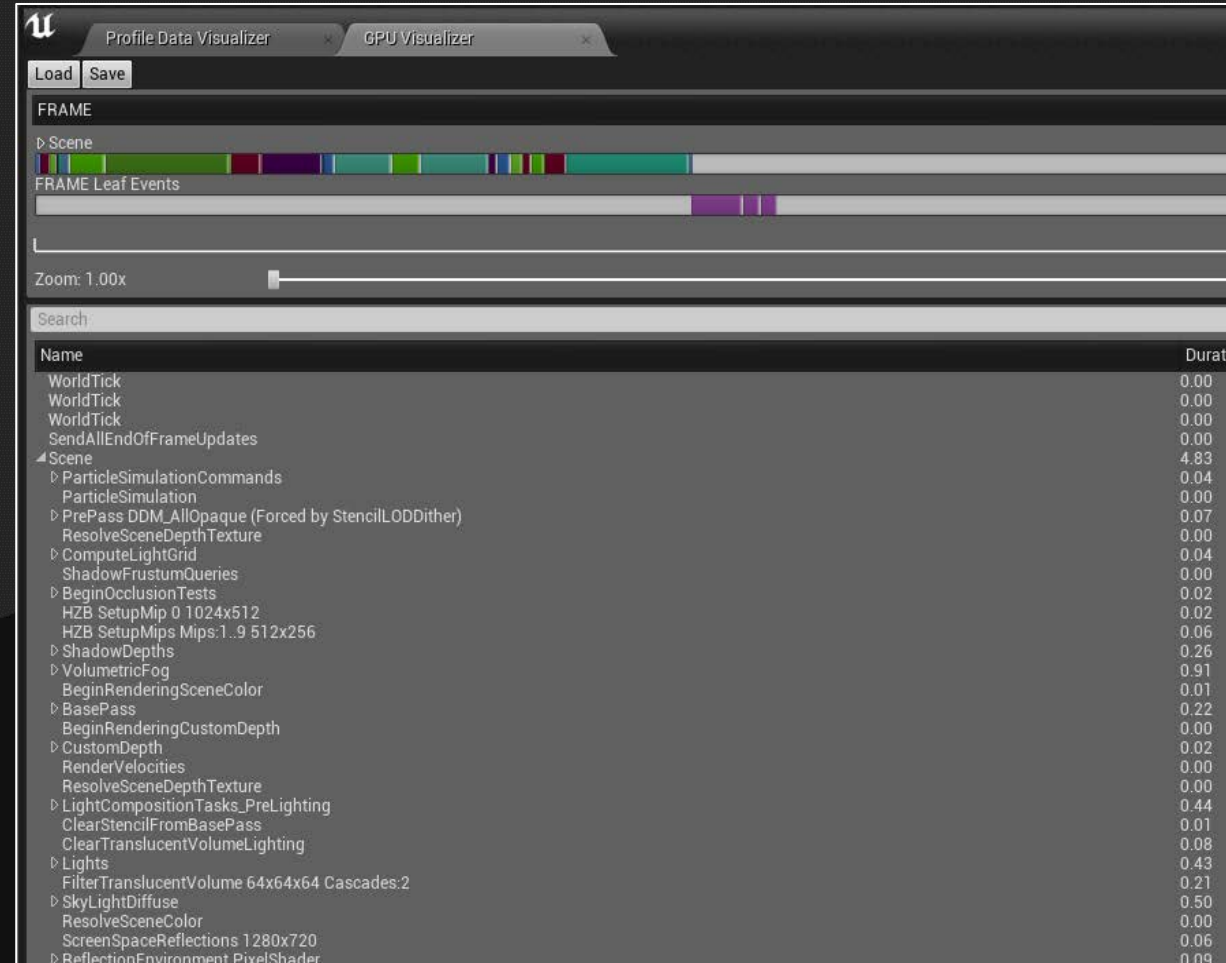


※ 드로우콜 최적화시 참고 = <https://docs.unrealengine.com/en-US/Engine/Performance/CPU>

ProfileGPU

GPU 타임 세부를 보여주는 빌트인 툴

- GPU Timestamp 활용
- GUI / 로그 형태: `r.ProfileGPU.ShowUI`
- 병목 추적하기: `r.ScreenPercentage` 사용
 - 엄청난 성능 향상이 있다면 Pixel bound
 - 변경 후 큰 성능 변화가 없다면 Vertex bound



※ 렌더 패스별 최적화시 참고 = <https://docs.unrealengine.com/en-US/Engine/Performance/GPU>

최적화

기본 점검 / 플랫폼간 다른 프로퍼티 설정
블루프린트 네이티브화 / 애니메이션 최적화
템포럴 업샘플 & 동적 해상도



최적화 – 기본 점검

어떤 렌더러를 사용하나

- 디퍼드 렌더러
- 포워드 렌더러
- VR 포워드 렌더러

어떤 피처레벨을 사용하나

- ES2 / ES3_1 / SM4 / SM5
- ※ ES3_1 = OpenGL ES3.1 + Vulkan + Metal

엔진 퀄리티를 어떻게 설정하였는지

- Low / Medium / High / Epic / Cinematic / Auto
- FPS / Resolution / Quality / Motion Blur / Texture details / Anti-aliasing



최적화 – 플랫폼간 다른 프로퍼티 설정

Level Of Detail (이하 LOD) 관련

- 에디터내 StaticMesh / SkeletalMesh LOD 생성 지원
- 플랫폼간 다른 설정을 적용할 수 있음
- 애셋간 설정을 재사용할 수 있음



최적화 – 플랫폼간 다른 프로퍼티 설정

플랫폼별 다른 LOD 설정

- 플랫폼마다 대표 하드웨어가 제각각, 특히 콘솔
- 플랫폼간 다른 설정을 편하게 설정 가능
- 원소스 멀티 유즈

[시연] LOD 생성 + 플랫폼별 + 애셋별



최적화 – 블루프린트 네이티브화

블루프린트 vs. C++

- VM을 통해야 하는 블루프린트보다 네이티브 호출이 훨씬 빠르다
- 블루프린트는 제작시 Iteration이 훨씬 편하고 빠르다

블루프린트 네이티브화

- 블루프린트를 네이티브 코드로 해석/변환
- 프로젝트 세팅 / 패키징 / Blueprints 섹션에 설정 존재

Blueprints

Blueprint Nativization Method

Inclusive

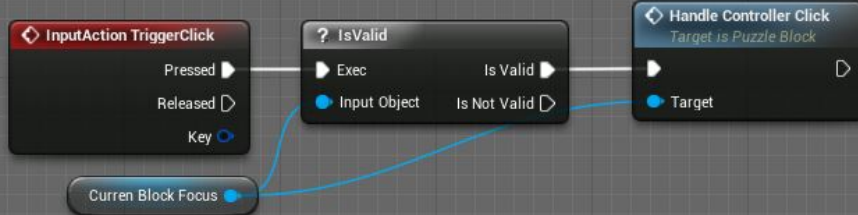
Warn if Packaged Without Nativization Flag

Disabled

Inclusive

Exclusive

Use R Controller Trigger to process click events



```
461 void APuzzlePlayerCharacter_C_pf2520442763::bpf__InpActEvt_TriggerClick_K2Node_InputActionEvent_1
462 {
463     b01_K2Node_InputActionEvent_Key_pf = bpp_Key_pf;
464     bpf__ExecuteUbergraph_PuzzlePlayerCharacter_pf_1(7);
465 }
466 void APuzzlePlayerCharacter_C_pf2520442763::bpf__InpActEvt_ResetVR_K2Node_InputActionEvent_1
467 {
468     b01_K2Node_InputActionEvent_Key2_pf = bpp_Key_pf;
469     bpf__ExecuteUbergraph_PuzzlePlayerCharacter_pf_2(1);
470 }
```



최적화 – 애니메이션 최적화

Update Rate Optimization (URO)

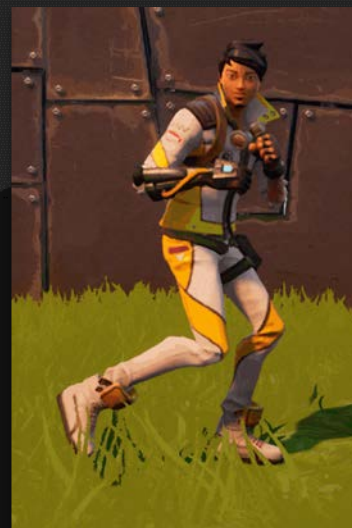
- 중요도에 따라 애니메이션 업데이트 빈도를 조절
- Ex. 중요도가 낮은 원거리, 혹은 작은 스크린사이즈 객체는 업데이트를 스킵
- 퀄리티 보완을 위해 보간 지원



No URO



URO 4



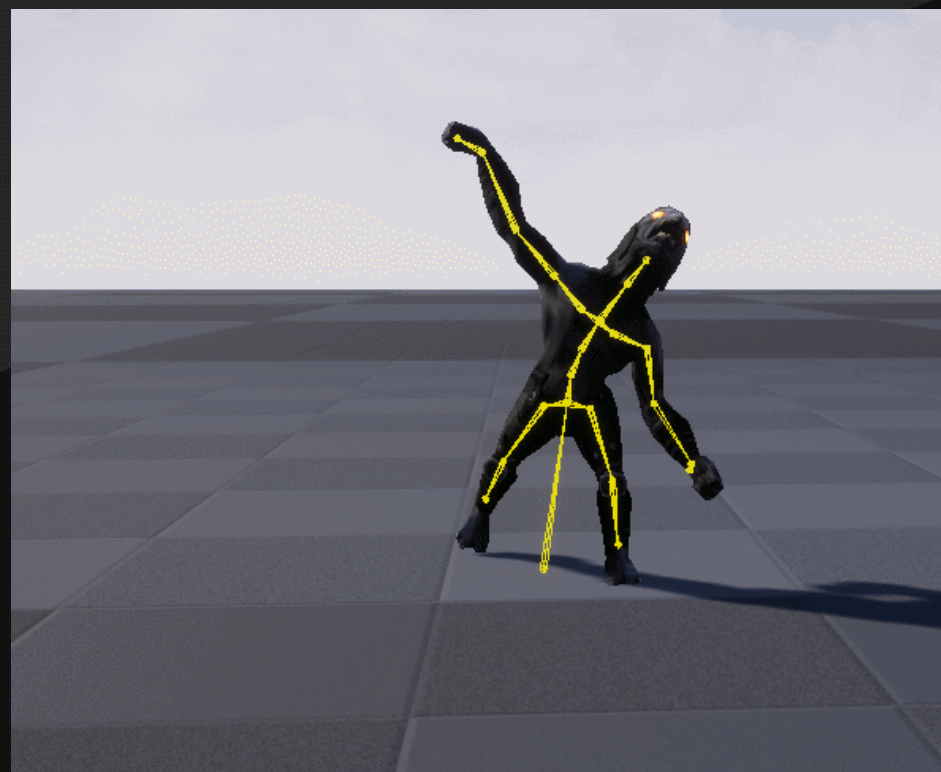
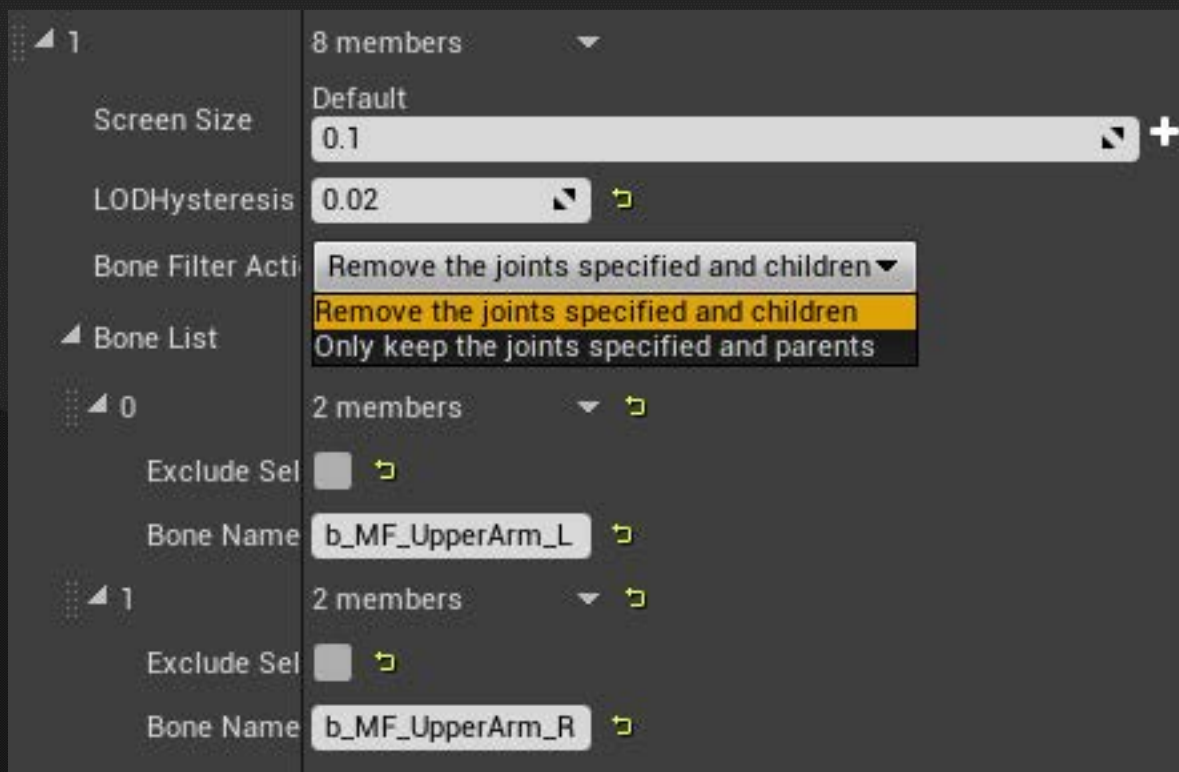
URO 10



URO 4 – No Interp

최적화 – 애니메이션 최적화

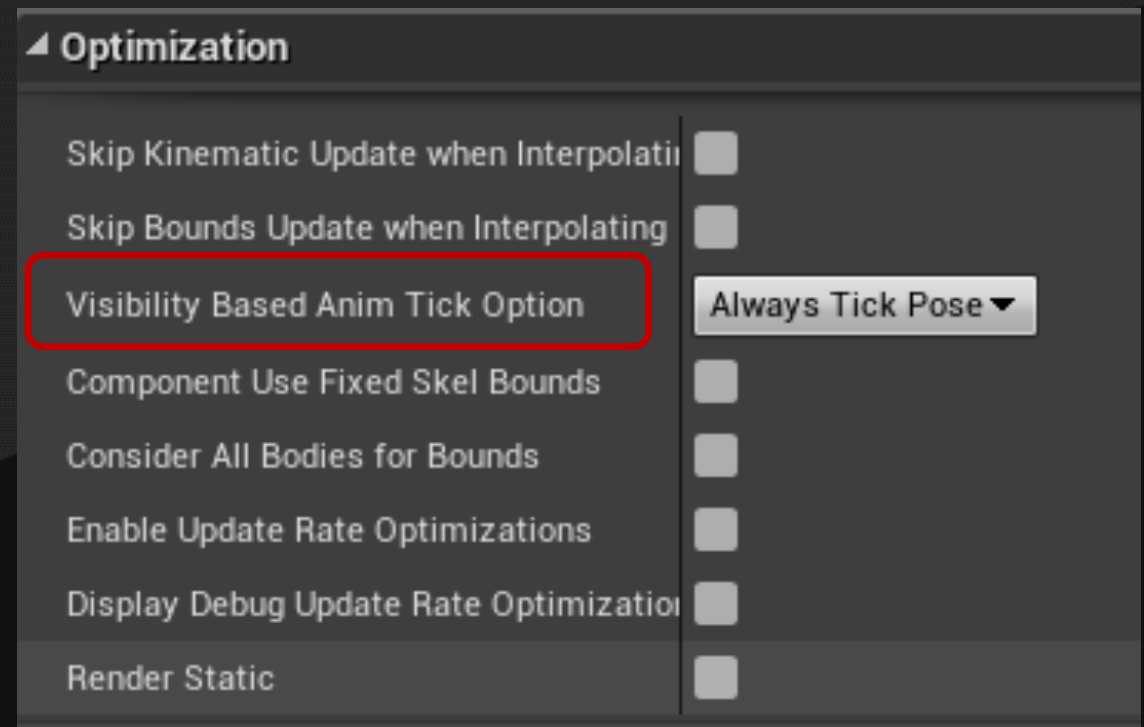
LOD별로 특정 본 이하 애니메이션 업데이트 무시하기



최적화 – 애니메이션 최적화

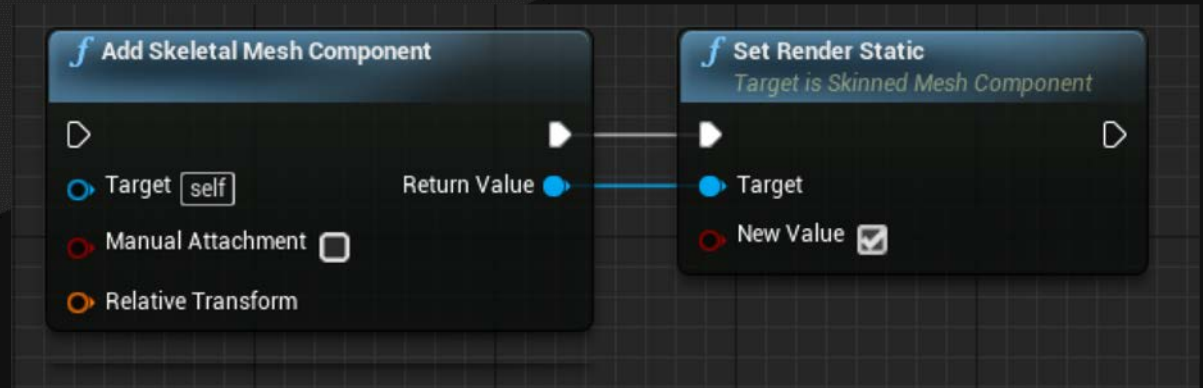
SkeletalMesh가 화면에 보이는 경우에만 업데이트 하기

- AlwaysTickPoseAndRefreshBones
- AlwaysTickPose
- OnlyTickMontagesWhenNotRendered
- OnlyTickPoseWhenRendered



최적화 – 애니메이션 최적화

애니메이션하지 않는 SkeletalMesh는 StaticMesh 렌더패스로 그리기



최적화 – 애니메이션 최적화

애니메이션 버짓을 정하고, 동적으로 SkeletalMesh 틱 제어

- 플랫폼 별로 (게임스레드에서 수행할 ms 단위 작업량으로) 고정 예산을 설정
- 현재 작업이 그 예산에 맞는지 아니면 작업량을 줄여야 하는지 결정



최적화 – 애니메이션 최적화

애니메이션 셰어링: 군중 연출을 위해, 업데이트된 포즈 공유하여 사용하기

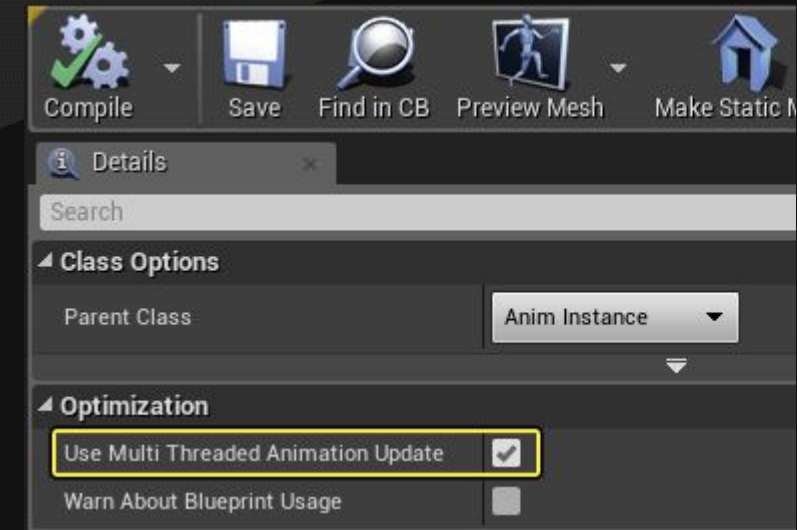
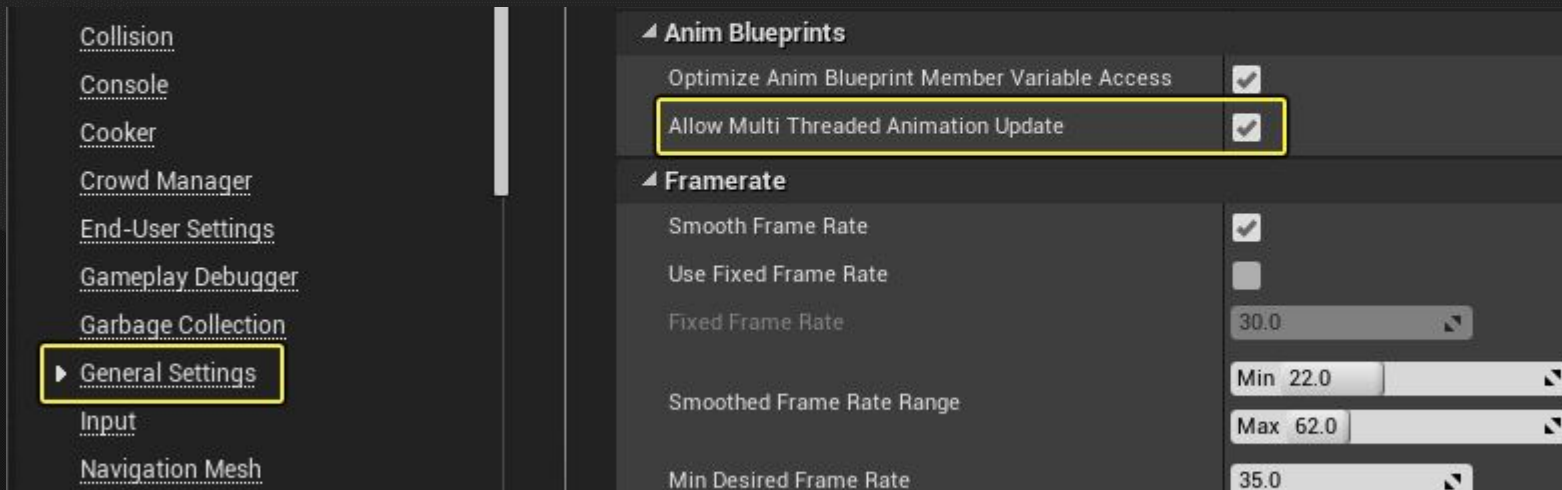
- 애니메이션 스테이트 버킷 세트를 활용하여 애니메이션 인스턴스를 평가
- 최종 포즈를 모든 자손 컴포넌트에 전송



최적화 – 애니메이션 최적화

부하가 큰 작업 나누기

- 멀티 스레드로 업데이트 작업 나누기
- [필수] 프로젝트 세팅 설정 / [필수] 애니 블루프린트에 지정



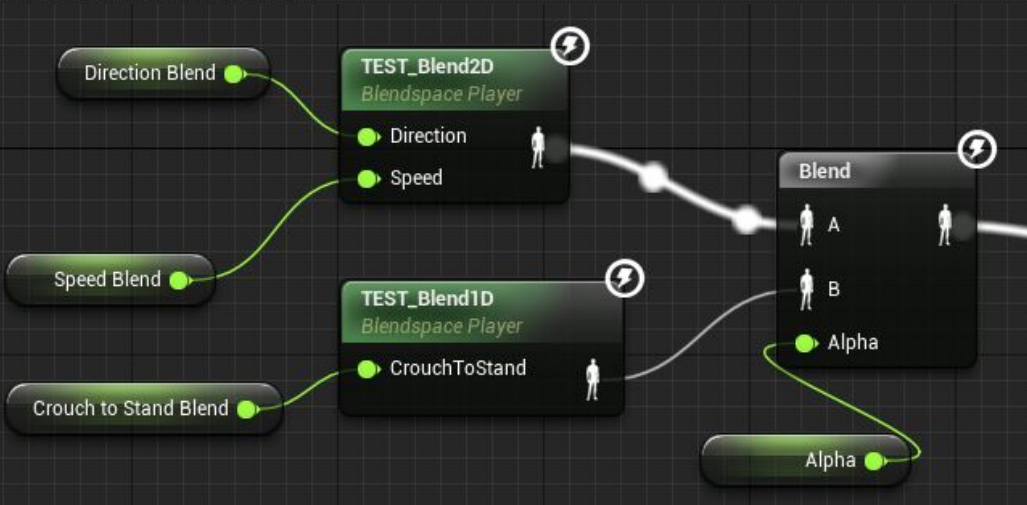
최적화 – 애니메이션 최적화

블루프린트 VM사용 최대한 피하기

- 애니메이션에서 무거운 블루프린트 연산을 하지 않는 경우, 빠른 경로 Fast Path를 사용.
- 빠른 경로일 경우, 애니 블루프린트에서 번개 아이콘으로 표시
- [필수] 프로젝트 세팅 / [옵셔널] 애니 블루프린트에 경고 표시

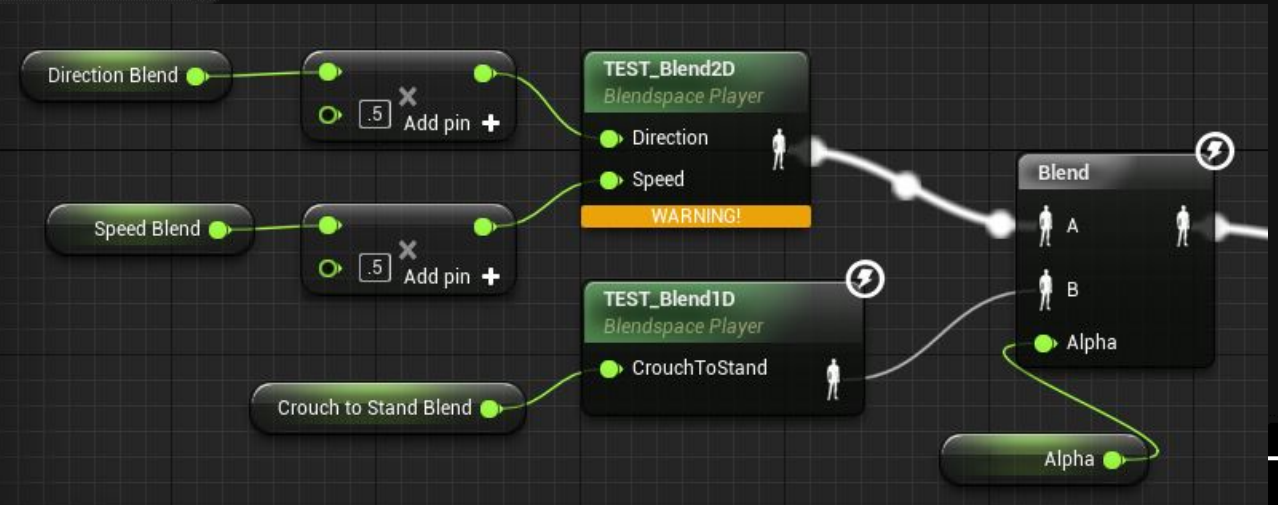
Anim Blueprints

- ☒ Optimize Anim Blueprint Member Variable Access
- ☒ Allow Multi Threaded Animation Update



Optimization

- ☒ Use Multi Threaded Animation Update
- ☒ Warn About Blueprint Usage



최적화 – 해상도 관련

템포럴 업샘플 & 동적 해상도

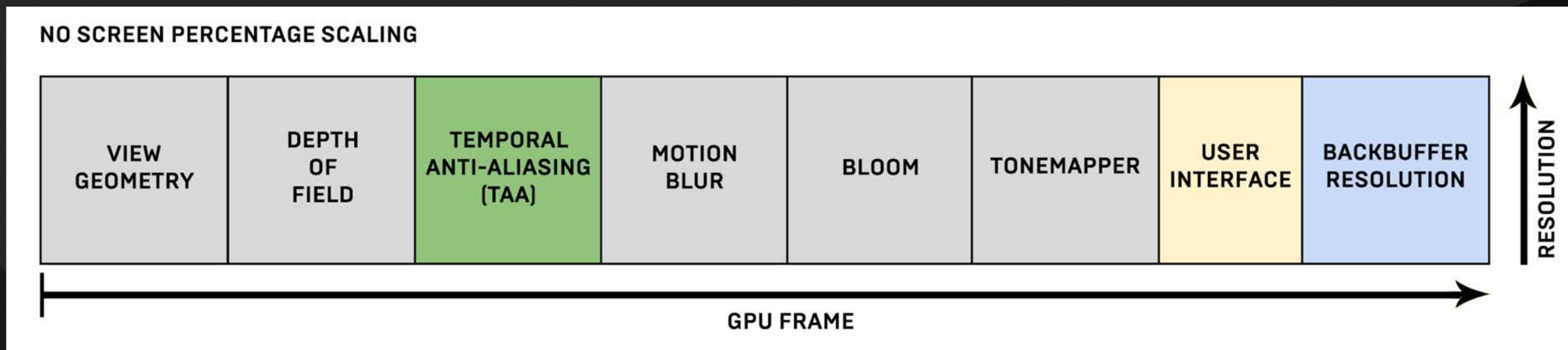


UNREAL ENGINE

언리얼 서밋 2019 전국투어

최적화 – 템포럴 업샘플링

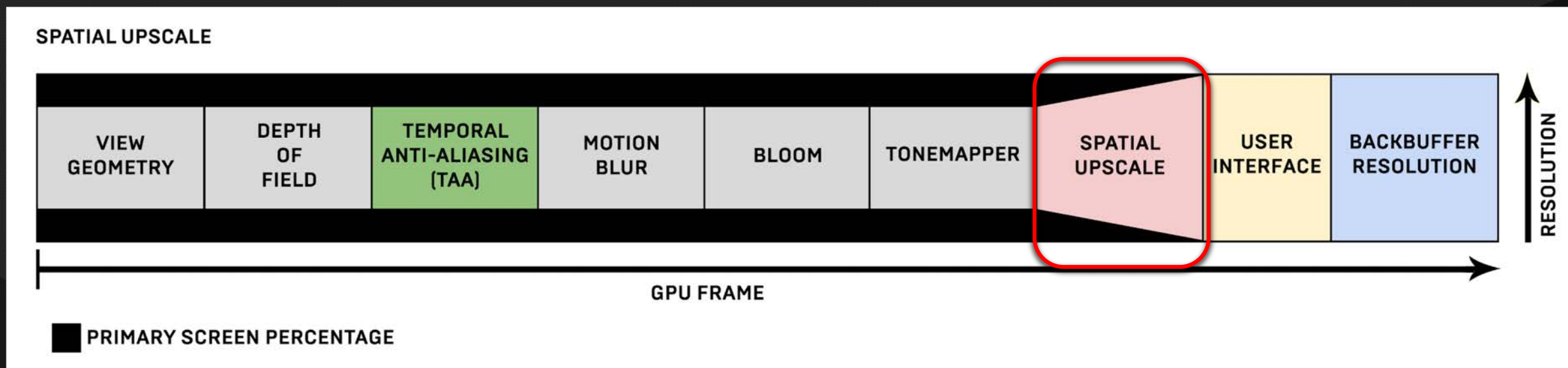
프레임 렌더링 중 스크린 해상도



기본값 = Scaling 없음

최적화 – 템포럴 업샘플링

Spatial Upscale [기존]



- UI 그리기 직전 업-샘플링
- UI는 최상단에 그려져, 퀄리티 하락이 가장 눈에 보임.

최적화 – 템포럴 업샘플링

Spatial Upscale Settings

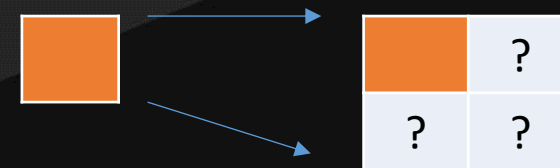
r.ScreenPercentage

- 100: 기본 해상도
- 실기기에서 납득할만한 퀄리티까지 낮추며 적절한 값 결정

r.Upscale.Quality

- 0: Nearest Filtering / 1: Simple Bilinear / 2: Directional blur with unsharp mask upsample...
- 3: 5-tap Catmull-Rom bicubic, approximating Lanczos 2 (default)
- 4: 13-tap Lanczos 3
- 5: 36-tap Gaussian-filtered unsharp mask (very expensive, but good for extreme upsampling)

※ 가로 x2 세로 x2 하는 경우



Screen Percentage: 100
Temporal Upsample: Disabled

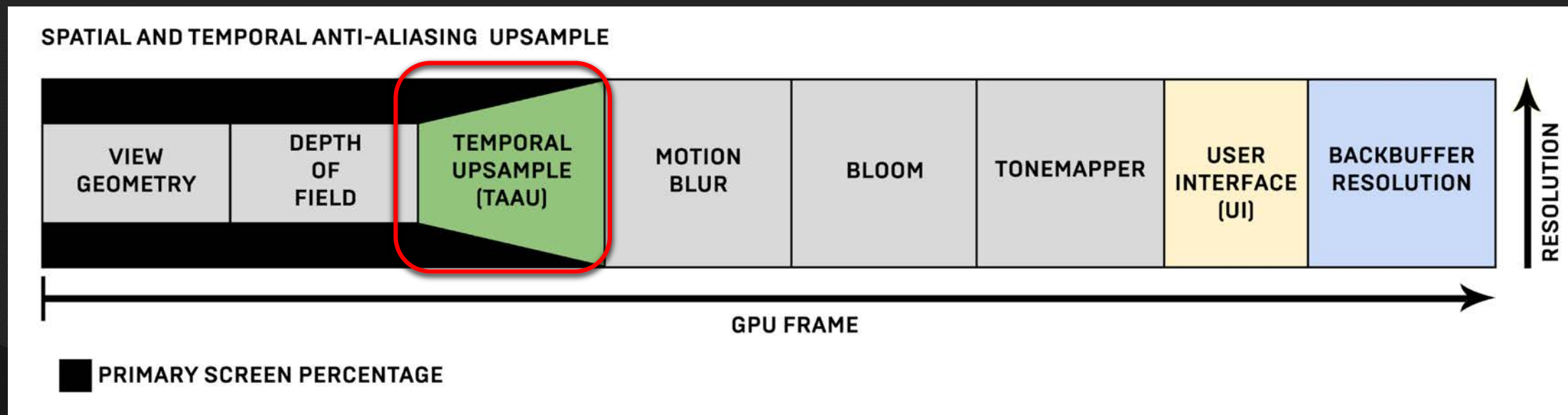


Screen Percentage: 70
Temporal Upsample: Disabled



최적화 – 템포럴 업샘플링

Temporal Anti-Aliasing Upsample (TAAU) [4.19+]

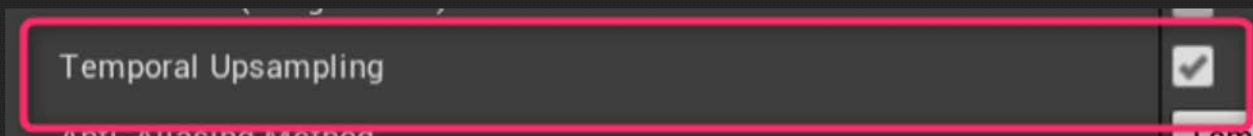


- TemporalAA에서 업샘플링
- 모션 블러, 블룸을 높은 해상도로 = Spatial Upscale보다 부하가 크다.
- 동적 해상도 Dynamic Resolution은 여기서 해상도를 변경

최적화 – 템포럴 업샘플링

Temporal AA Upscale Settings

“프로젝트 세팅/엔진/렌더링”에서 TAAU 켜고/끄기



r.ScreenPercentage

- Spatial Upscale과 같은 커맨드 사용

r.PostProcessAAQuality

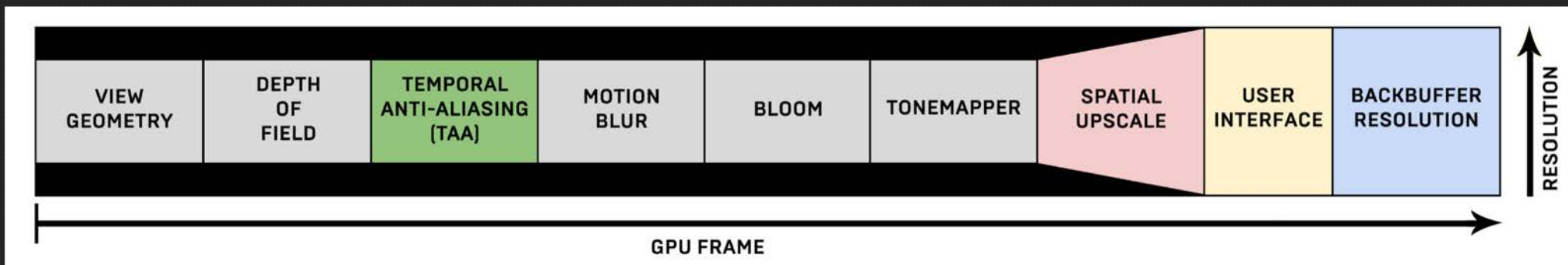
- TAAU 품질 세팅 = TemporalAA 품질 세팅
- ~~0:off / 1:very low (faster FXAA) / 2:low (FXAA)~~
- 3:medium (faster TemporalAA) / 4:high (default TemporalAA) / 5:very high / 6:max

Screen Percentage: 100
Temporal Upsample: Disabled

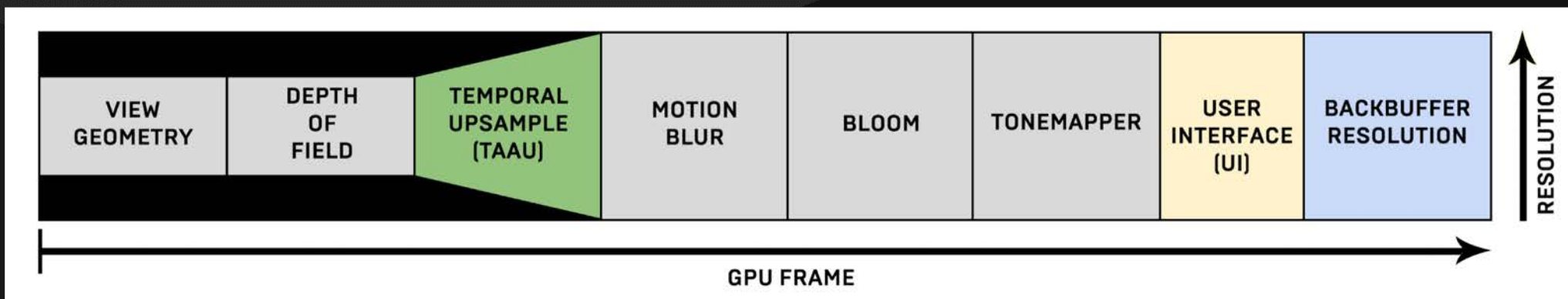


Screen Percentage: 70
Temporal Upsample: Enabled





Spatial Upscale VS Temporal Upsampling



Screen Percentage: 70
Temporal Upsample: Disabled



Screen Percentage: 70
Temporal Upsample: Enabled



Screen Percentage: 100
Temporal Upsample: Disabled



Screen Percentage: 70
Temporal Upsample: Disabled

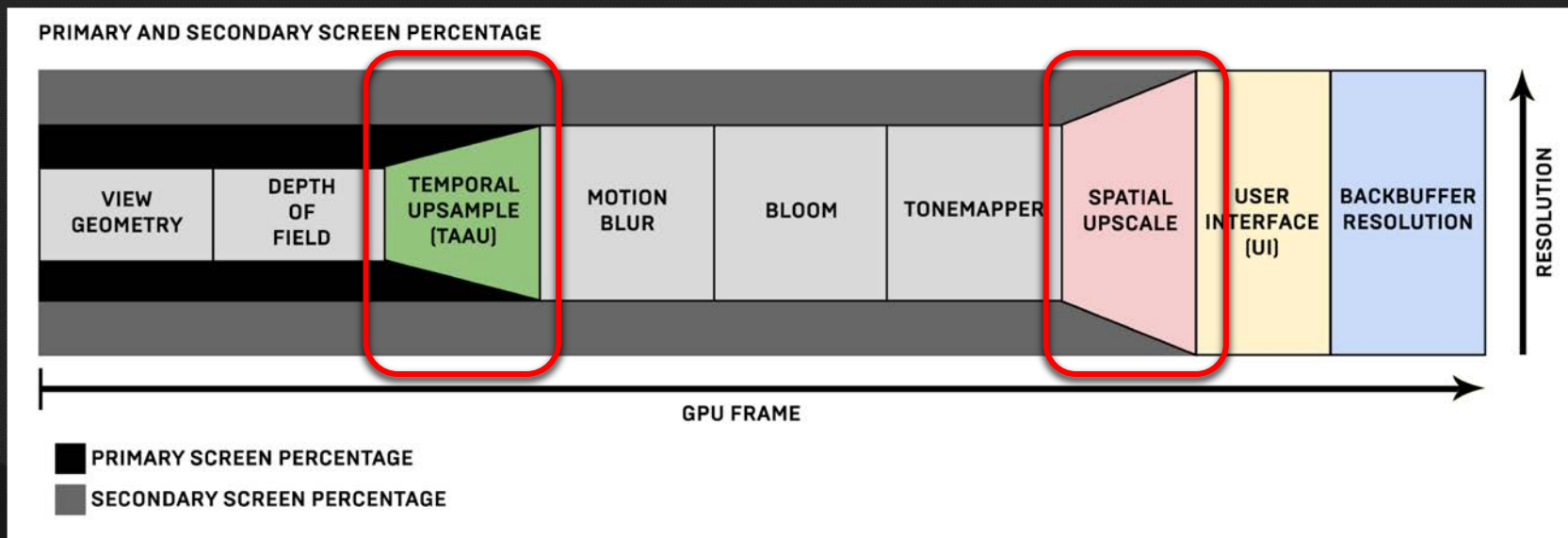


Screen Percentage: 70
Temporal Upsample: Enabled



최적화 – 템포럴 업샘플링

Secondary Spatial Upscale

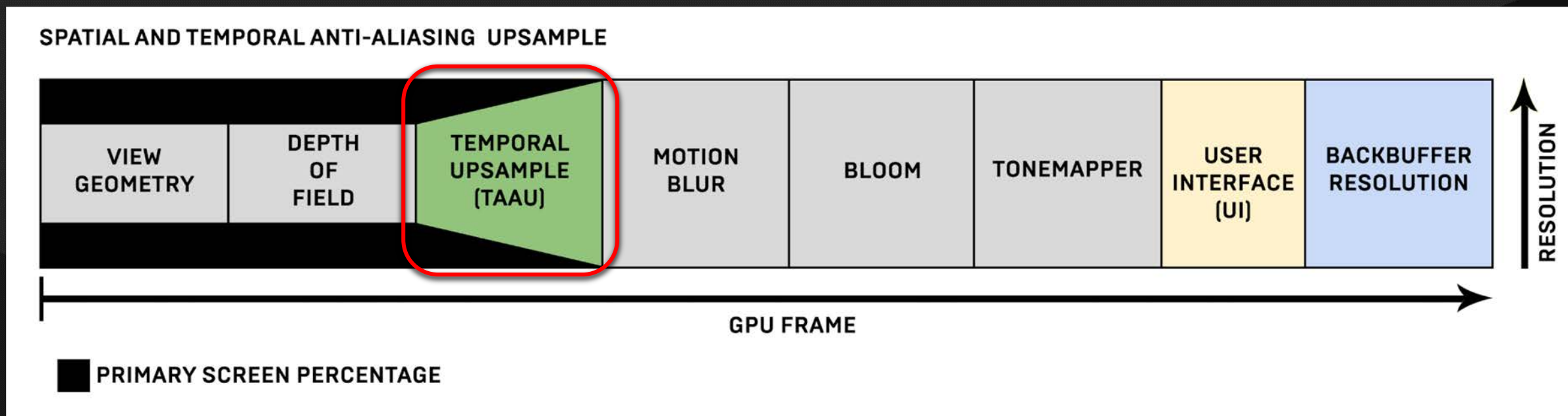


동시 사용 가능 : Spatial Upscale & TAAU

- `r.ScreenPercentage` : TAAU
- `r.SecondaryScreenPercentage.GameViewport` : Spatial Upscale

최적화 – 동적 해상도

Dynamic Resolution



- Temporal Upsample을 사용하여, GPU의 부하에 따라 자동으로 해상도를 변경
- TAAU는 동적으로 변경 가능

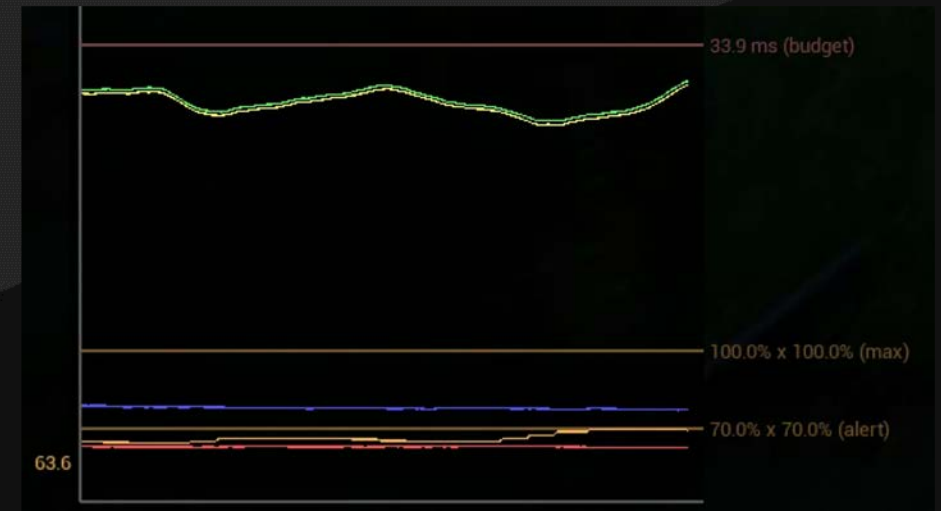
최적화 – 동적 해상도

Dynamic Resolution Settigs

콘솔커맨드 : “r.DynamicRes.~~~”로 시작

최대/최소 해상도 세팅

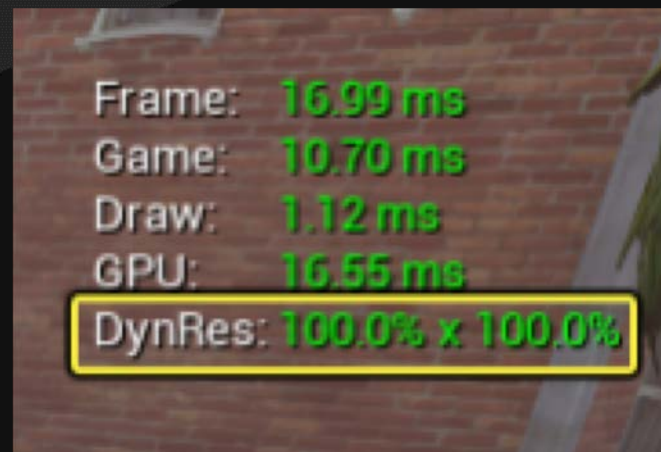
- r.DynamicRes.MinScreenPercentage
- r.DynamicRes.MaxScreenPercentage



최적화 – 동적 해상도

Enabling Dynamic Resolution

- r.DynamicRes.OperationMode
 - 0: 끄기 (기본값)
 - 1: “Game User Settings”을 따릅니다.
 - 2: 항상 켜기



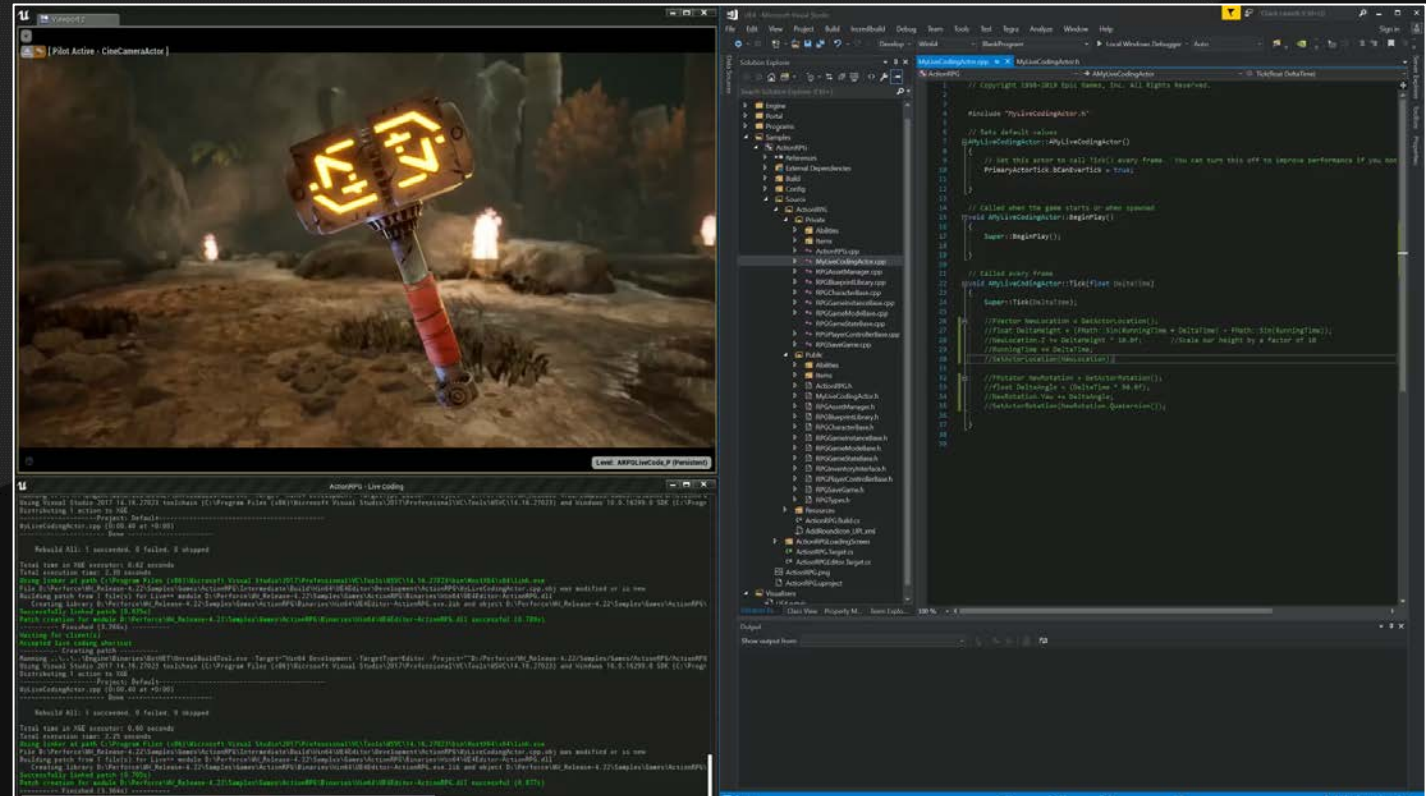
게임 최신 기능

라이브 코딩 / 빌드 시간 단축

카오스 / 버추얼 텍스처 / 리얼타임 레이트레이싱 / 언리얼 인사이트

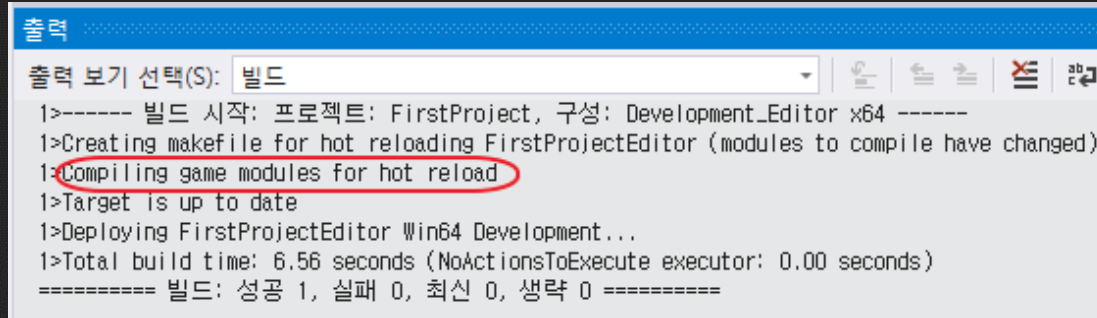
라이브 코딩

- 라이브 코딩(Live coding) 기능
 - 핫 리로드와 달리 함수 단위 패치를 통해 변경 적용
 - 초 단위로 수정 반영



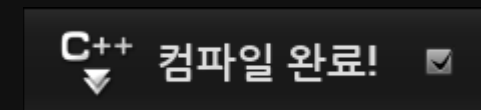
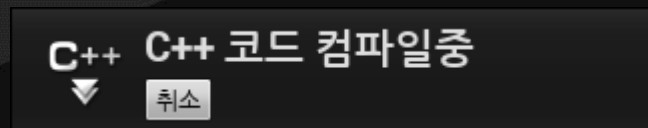
라이브 코딩 vs. 핫 리로드

- 핫 리로드 기능
 - 에디터를 닫지 않고, C++ 변경 내용을 바로 반영
 - 평소처럼 Visual Studio에서 빌드



```
출력
출력 보기 선택(S): 빌드
1>----- 빌드 시작: 프로젝트: FirstProject, 구성: Development_Editor x64 -----
1>Creating makefile for hot reloading FirstProjectEditor (modules to compile have changed)
1>Compiling game modules for hot reload
1>Target is up to date
1>Deploying FirstProjectEditor Win64 Development...
1>Total build time: 6.56 seconds (NoActionsToExecute executor: 0.00 seconds)
===== 빌드: 성공 1, 실패 0, 최신 0, 생략 0 =====
```

- 에디터 톨바 “Compile” 클릭



⇒ 컴파일된 새로운 DLL을 즉시 리로드(Hot-reload)

빌드 시간 단축

4.22 기준, UBT/UHT 최적화로 C++ 이터레이션 시간을 최대 3배 단축

전체 빌드 (UE4Editor Win64 Development)

	언리얼 엔진 4.21	언리얼 엔진 4.22	개선
총 빌드 시간:	436.90	326.81	30% 개선
UnrealHeaderTool 컴파일:	46.12	46.30	
헤더 생성	25.05	15.50	60% 개선
UE4Editor 컴파일	323.15	257.97	25% 개선
UnrealBuildTool 오버헤드	42.58	7.04	600% 개선

증분 빌드 (UE4Editor Win64 Development)

	언리얼 엔진 4.21	언리얼 엔진 4.22	개선
총 빌드 시간:	7.47	2.14	340% 개선
Compiling UE4Editor	1.19	1.08	
UnrealBuildTool overhead	6.28	1.06	590% 개선

코드 변경 없음 (UE4Editor Win64 Development)

	언리얼 엔진 4.21	언리얼 엔진 4.22	개선
UnrealBuildTool 오버헤드	5.38	1.03	520% 개선

카오스



언리얼 서밋 2019 전국투어

카오스

차세대 피직스 및 디스트럭션 시스템

- 피직스 – PhysX(충돌 / 랙돌 / 클로스 / 비히클) 대체 목표
- 디스트럭션 – 디스트럭터블 메시 대체

디스트럭터블 메시

- 스테틱 메시지를 “변환”한 후 파티클, 사운드, (내부/외부) 머티리얼 설정 => 충분히 빠르게 작업 가능
- 복잡한 파괴를 위해 외부 DCC툴, 플러그인 활용 => 써드파티 API에 의존적
- 더 높은 수준의 파괴 묘사를 위해 카오스 개발

카오스

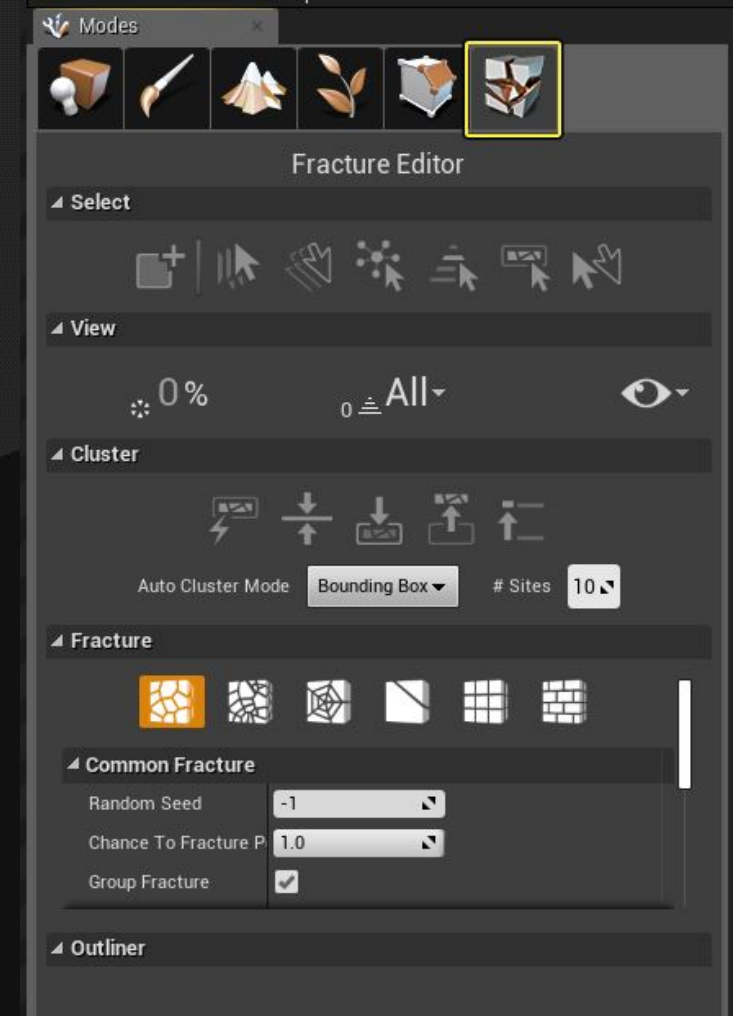
차세대 피직스 및 디스트럭션 시스템

- 현재 피직스는 작업 중
- 4.23에 공개된 카오스는 디스트럭션 시스템에 초점.

디스트럭션

- 지오메트리 콜렉션 / 클러스터 / 프랙처
- 필드 / 연결 그래프 / 시뮬레이션 캐시 / 나이가라 통합

[시연] 카오스



버추얼 텍스처링



버추얼 텍스처링

고품질 텍스처 사용과 메모리 절약을 동시에

스트리밍 버추얼 텍스처

- 일반 텍스처를 버추얼 텍스처로 “변환”
- Ex. 라이트맵, 디테일 노멀 맵,

런타임 버추얼 텍스처

- 프로시저럴 텍스처에 사용
- Ex. 랜드스케이프에 추가한 데칼, 스플라인



버추얼 텍스처링

Stat VirtualTexturing

Virtual Texturing [STATGROUP_virtualtexturing]					
Cycle counters (flat)					
	CallCount	InclusiveAvg	InclusiveMax	ExclusiveAvg	ExclusiveMax
VirtualTextureSystem Update	1	0.22 ms	0.35 ms	0.03 ms	0.08 ms
Finalize Requests	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Gather Requests	1	0.11 ms	0.24 ms	0.09 ms	0.11 ms
Feedback Analysis	1	0.04 ms	0.08 ms	0.01 ms	0.06 ms
Submit Tasks	2	0.03 ms	0.03 ms	0.00 ms	0.01 ms
Page Table Updates	1	0.00 ms	0.01 ms	0.00 ms	0.01 ms
Submit Requests	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Merge Unique Pages	1	0.00 ms	0.01 ms	0.00 ms	0.01 ms
Map Requests	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Sort Requests	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Merge Requests	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms

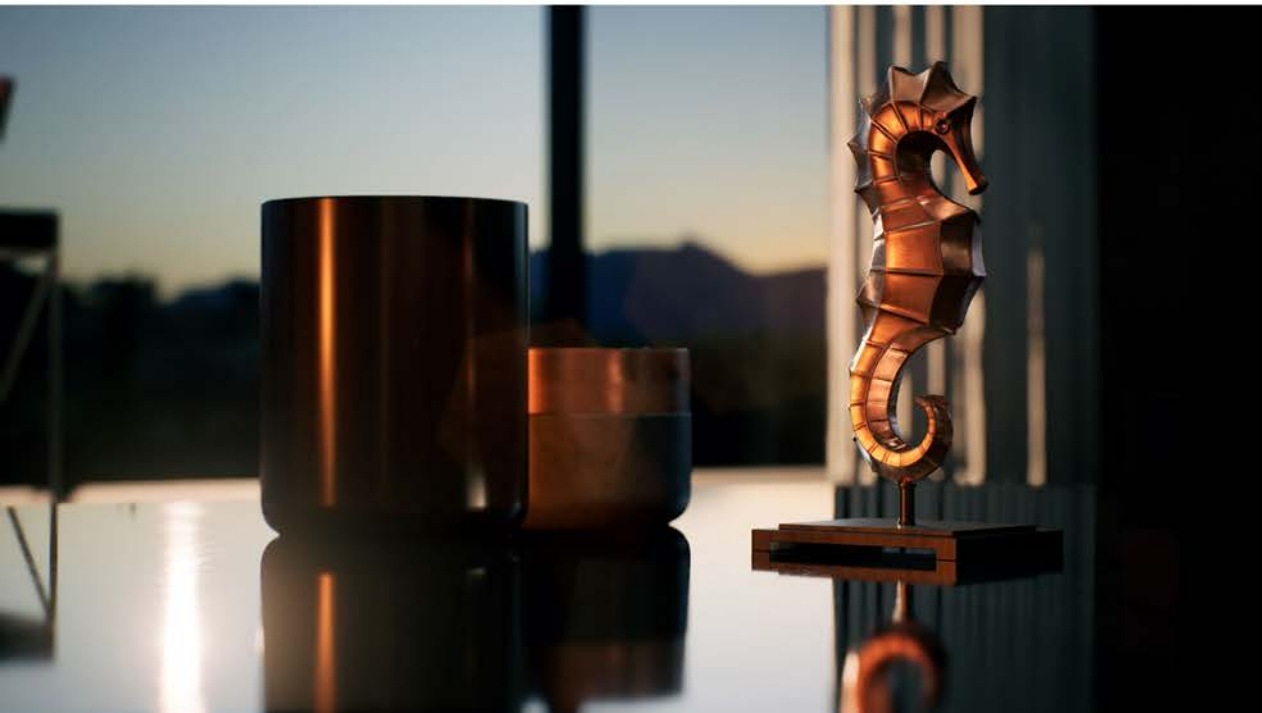
Stat VirtualTextureMemory

Memory Counters				
	UsedMax	Mem%	MemPool	Pool Capacity
Total Physical Memory	451.42 MB		GPU	
Total Pagetable Memory	128.04 MB		GPU	
Total GPU Upload Memory	24.62 MB		GPU	
Total Disk Size	21.90 MB		Physical	
Total CPU Upload Memory	10.80 MB		Physical	
Total Header Size	0.03 MB		Physical	
Tile Header Size	0.02 MB		Physical	
Blinking Blue: NAN or Inf				
Counters	Average	Max	Min	

[시연] 버추얼 텍스처링



리얼타임 레이트레이싱



언리얼 서밋 2019 전국투어

리얼타임 레이트레이싱

얻을 수 있는 효과

- Soft Shadows / Reflection / Translucency / Ambient Occlusion / Global Illumination

지원 타입

- Soft Shadows / Reflections / Translucency
- AO and a first pass GI / Image Based Lighting
- Static and Skeletal meshes, basic BSP, Niagara particles
- World Position Offset / Landscapes
- Hierarchical Instanced Static Mesh / Instanced Static Mesh
- Procedural meshes
- Multi-bounce Ray Traced Reflection



리얼타임 레이트레이싱

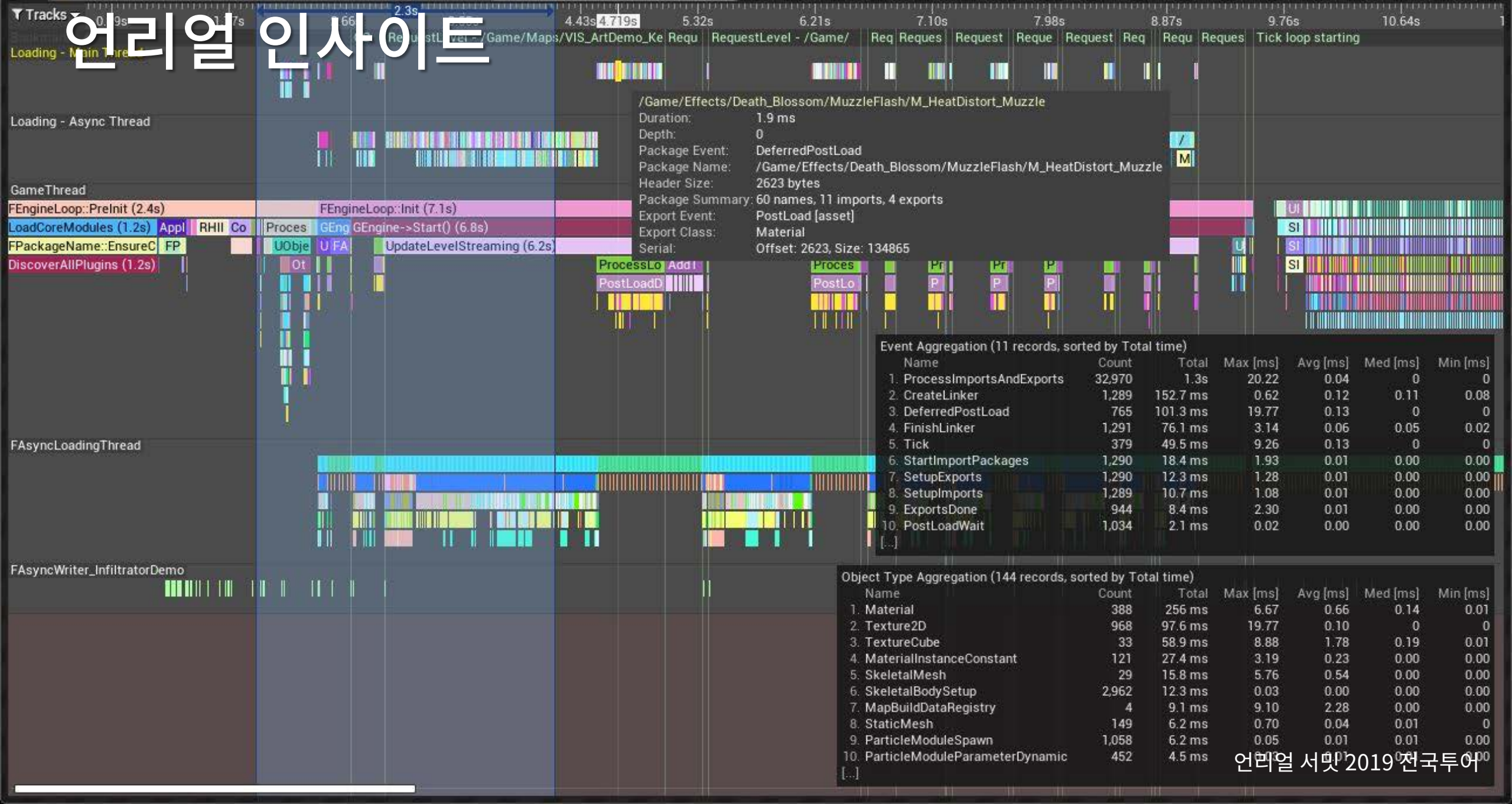


리얼타임 레이트레이싱

[시연] 리얼타임 레이트레이싱 데모

- Soft Shadow
- Ray Traced Reflection
- Material switch – Ray traced vs. Normal





언리얼 인사이드

언리얼 인사이트

엔진 내부 동작을 유저에게 더 보여주자

- 초기 버전에는 성능 분석에 초점
- 프로그래머, 기술에 익숙한 유저 대상 툴

툴 제작 목표

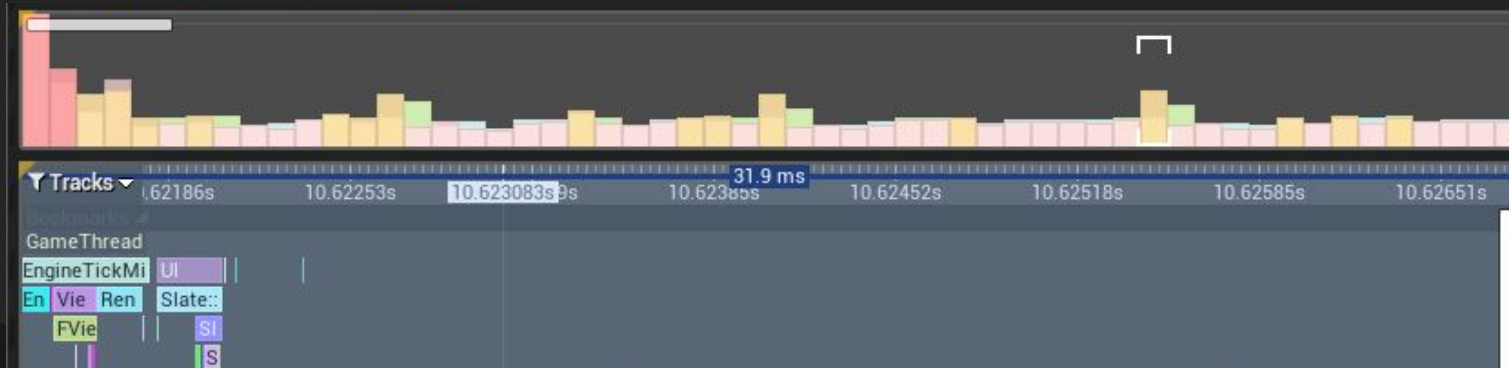
- 빠른 탐색: 빠르게 분석이 가능한 Responsive UI (FPS60)
- Scalability: 세션이 길고 데이터가 많아져도 문제없음
- 확장성: 새로운 정보 노출과 툴 확장이 쉽다
- 오프라인 분석: 자동화 리포트



언리얼 인사이트

지원 기능(초기 버전)

- CPU-timing 정보 (스코프 기반 CPU 프로파일링)
- 로그 출력 – CPU 이벤트와 연관 / 애셋 로딩 분석(실험기능)



- 타겟 무관: Game / Editor / 모든 Commandlet(ex. Cooking) / Server
- 추적 Tracing 지원 플랫폼: Windows / Linux / Consoles
- 분석툴 GUI 지원 플랫폼: Windows



언리얼 인사이트

사용법

- 새로 추가: TRACE_CPUPROFILER_EVENT_SCOPE
- 기존 stat용 후킹: SCOPE_CYCLE_COUNTER
- 타임라인 표시용 북마크 추가: TRACE_BOOKMARK

[시연] 언리얼 인사이트



감사합니다