Causing Chaos

The Future of Physics and Destruction in UE4

Session Overview

Jack Oakman
Epic Games - Senior Technical Artist

Benn Gallagher
Epic Games - Senior Physics Programmer

Kelly D Gawne
Intel - Game Performance Engineer

Chaos

UE4's new upcoming physics system

Chaos

UE4's new upcoming physics system

Preview ships in **4.23**

Chaos

UE4's new upcoming physics system

Preview ships in **4.23**

Great at destruction!

Designing
with Chaos

Chaos Demo

Completely created in Unreal Engine
Chaos is fully integrated with UE4

Chaos Demo

Completely created in Unreal Engine
Chaos is fully integrated with UE4

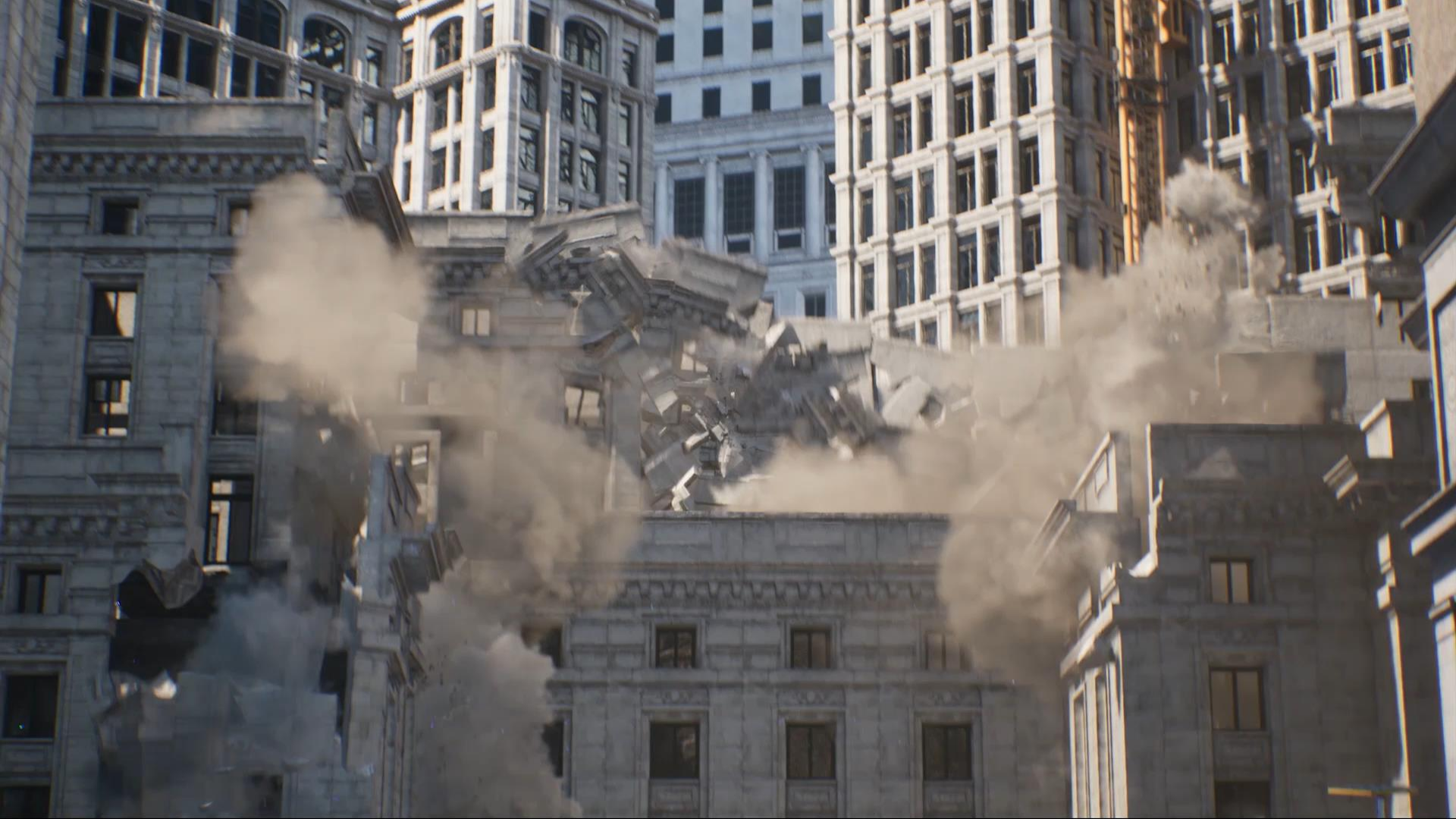
Rendered In Real-Time

Chaos Demo

Completely created in Unreal Engine
Chaos is fully integrated with UE4

Rendered In Real-Time

Simulated In Real-Time
When Needed



Realtime/Prebaked

Destruction can be pre-simulated, saved out and played back

- Allows us to art-direct
- Mass destruction at perf

Realtime/Prebaked

Destruction can be pre-simulated, saved out and played back

- Allows us to art-direct
- Mass destruction at perf

Created using exactly the same tools

- All are GeometryCollections

Realtime/Prebaked

Destruction can be pre-simulated, saved out and played back

- Allows us to art-direct
- Mass destruction at perf

Created using exactly the same tools

- All are GeometryCollections

Any part of a prebaked simulation can become active at any time!

- Destruction is never static!
- Chaos reacts to dynamic world state
- Extremely performant

Chaos is a System

Chaos Is A First Class Citizen of UE4

Lives alongside all other systems
that simulate your world

Chaos is a System

Chaos Is A First Class Citizen of UE4

Lives alongside all other systems
that simulate your world

Chaos can:

- Be Triggered
- Be Queried
- Send events

All gameplay systems
can tightly integrate

Managing Chaos

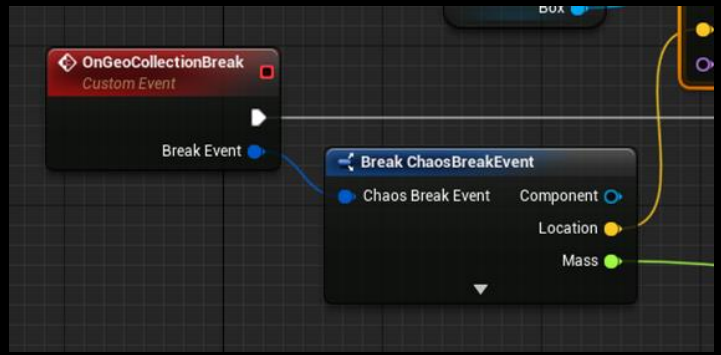
Chaos lets you create systems that account for runtime Chaos:

- Modify nav mesh
- AI can react to debris
- AI can detect debris

Managing Chaos

Chaos lets you create systems that account for runtime Chaos:

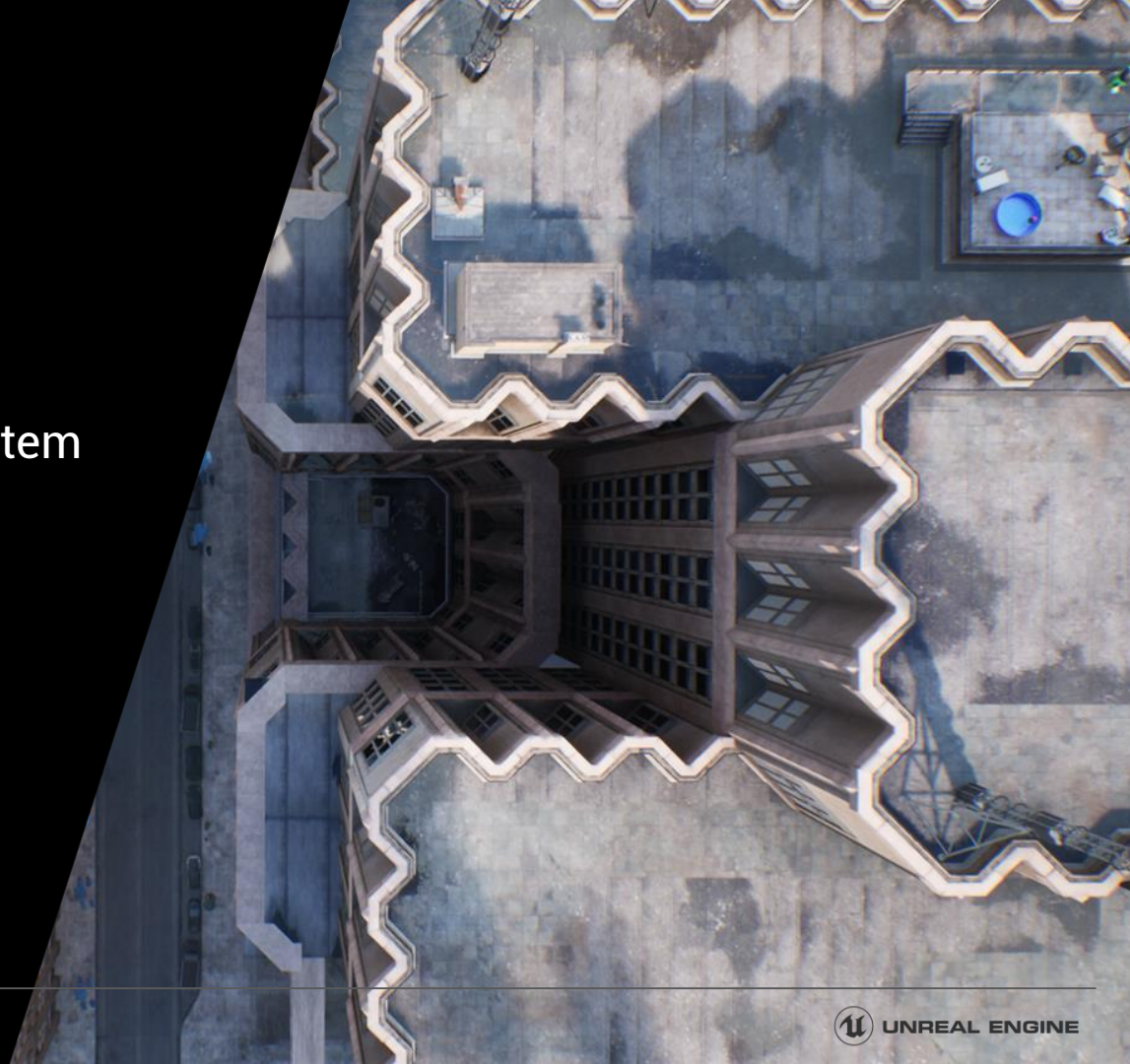
- Modify nav mesh
- AI can react to debris
- AI can detect debris
- Designers can detect destruction events



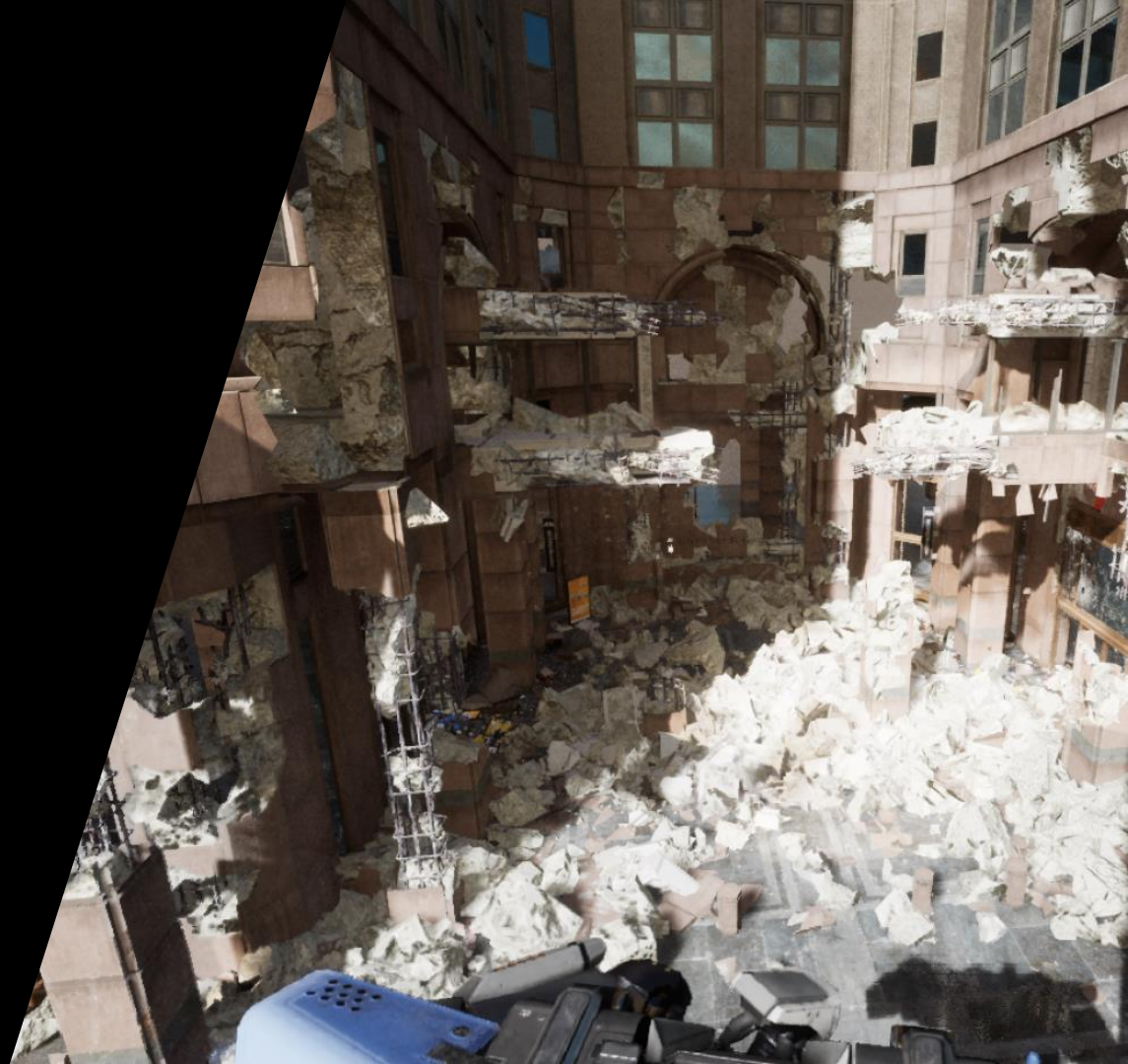
NETWORKING

Chaos was built with networking in mind

Chaos will be Epic's physics system of the future



The Art of **Destruction**



Planned Chaos...

Looks like total **Chaos**

- Random **Geo**
- Random **Fracture**
- Random **Forces**
- Random **User**

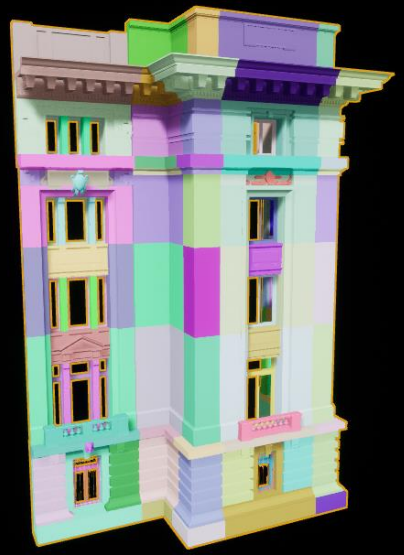
Control

Geometry Collections

A new type of asset in Unreal for destructible objects

Modular Blueprint

Single Geometry Collection



Individual Parts Transfer

Geometry Collections

One Pillar: 3 Geometry Collections

Front / Interior of Building: 200+ Geometry Collections

Entire Building: 12 Geometry Collections

Fracture in Unreal

Conversion to Geometry Collection:

- Water-tight

Original Geometry Collection

Fracture in Unreal

Conversion to Geometry Collection:

- Water-tight
- Non-Intersecting
- Cheap

Original Geometry Collection

Fracture in Unreal

Conversion to Geometry Collection:

- Water-tight
- Non-Intersecting
- Cheap
- Modular
- Gridded

Original Geometry Collection

Fracture in Unreal

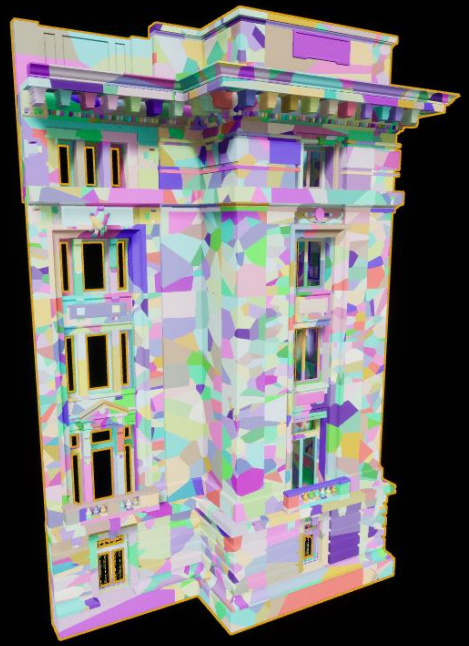
Conversion to Geometry Collection:

- Water-tight
- Non-Intersecting
- Cheap
- Modular
- Gridded
- Materials Assigned to Support Interiors

Fracture in Unreal

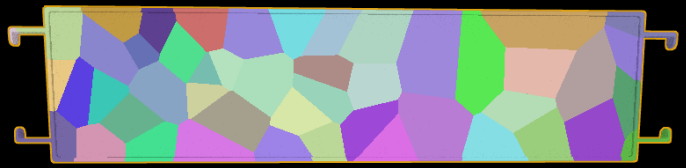
Original Geometry Collection

Fracture across entire mesh

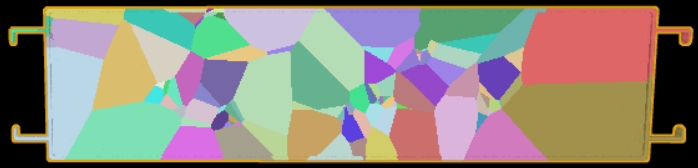


Sub-fracturing only large pieces

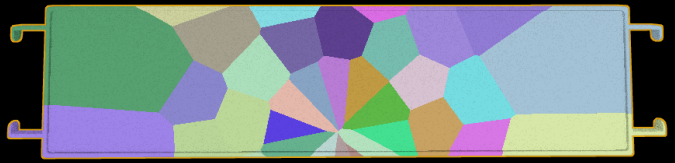
Fracture in Unreal



Standard Voronoi



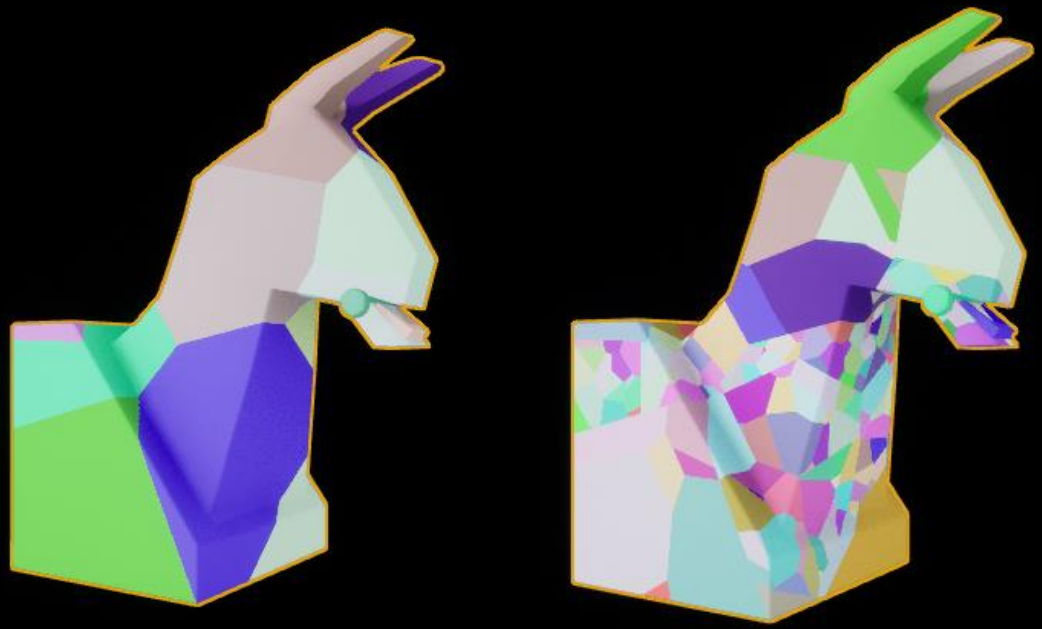
Clustered Voronoi

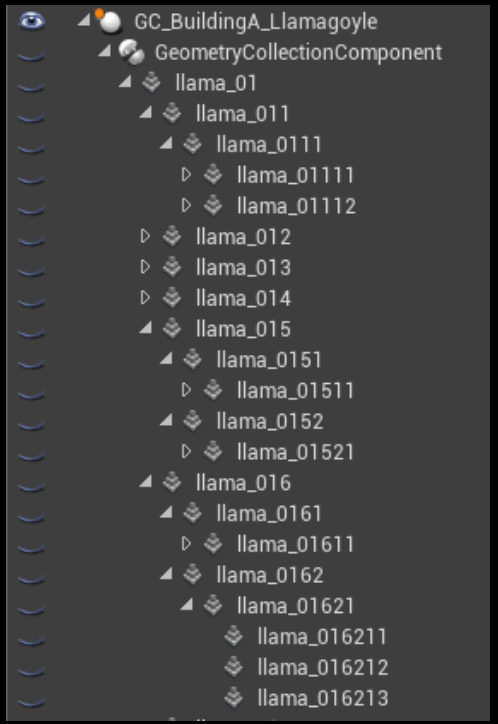


Radial Fracture

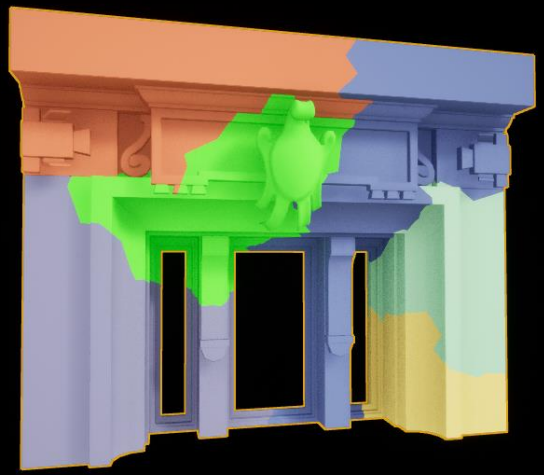
Planar Fracture

Sub-Fracturing and Levels

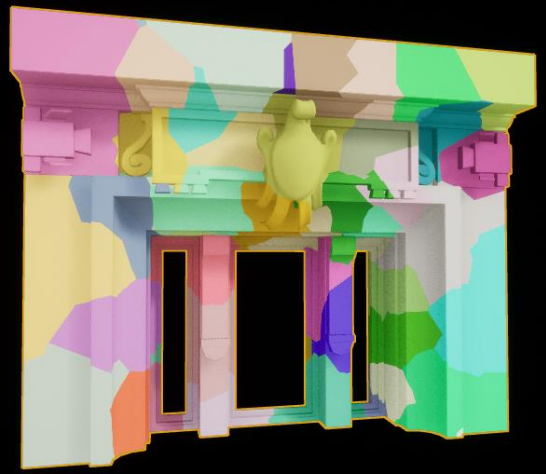




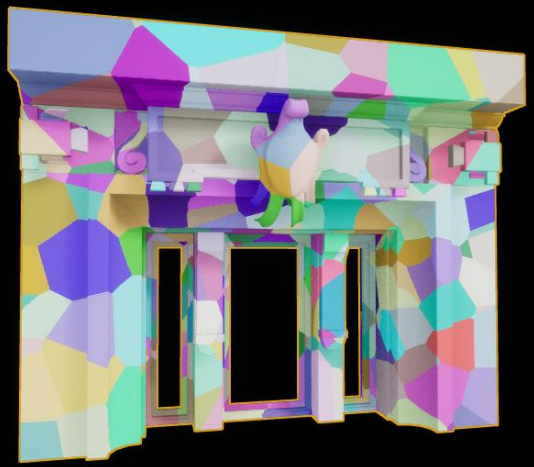
Performant Simulation Through **Clustering**



Level 1: 6 Objects



Level 3: 50 Objects



Level 5: 513 Objects

Performant Simulation Through **Clustering**

Level 1

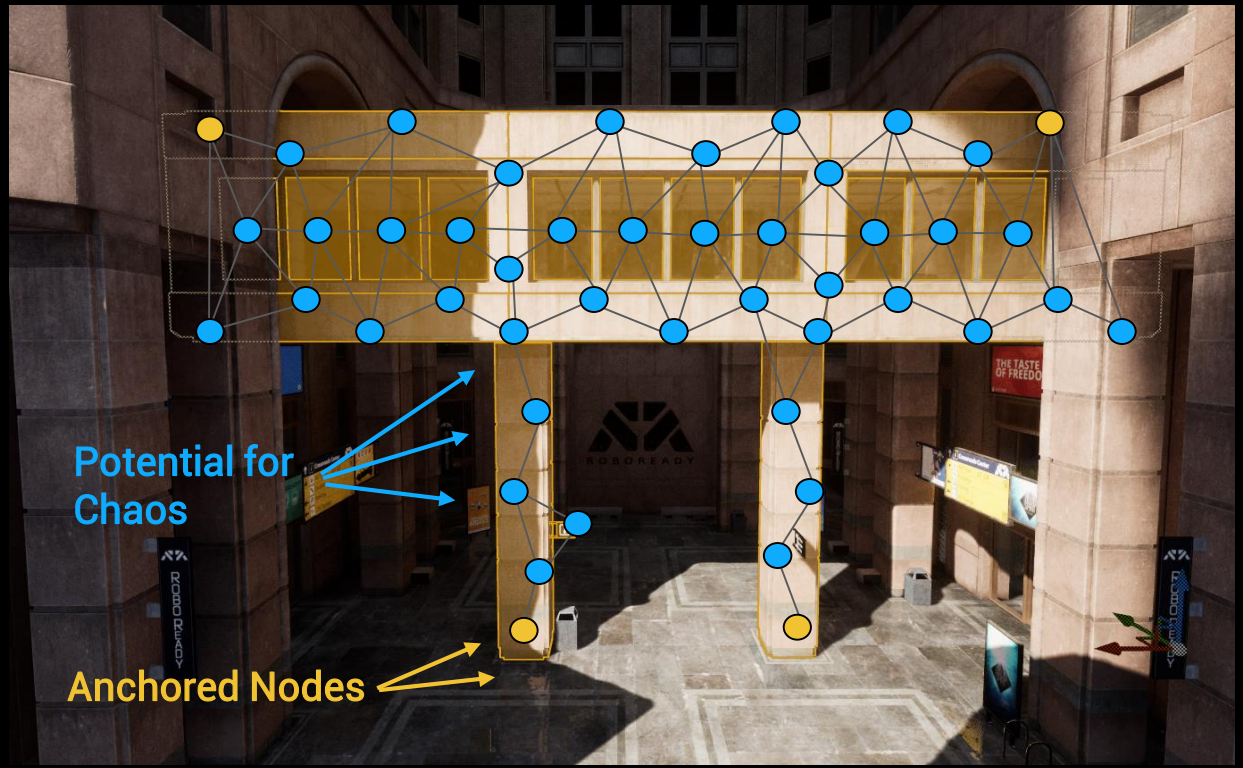
Level 5

All levels together

Performant Simulation Through **Clustering**

Clustering with **Auto-Cluster**

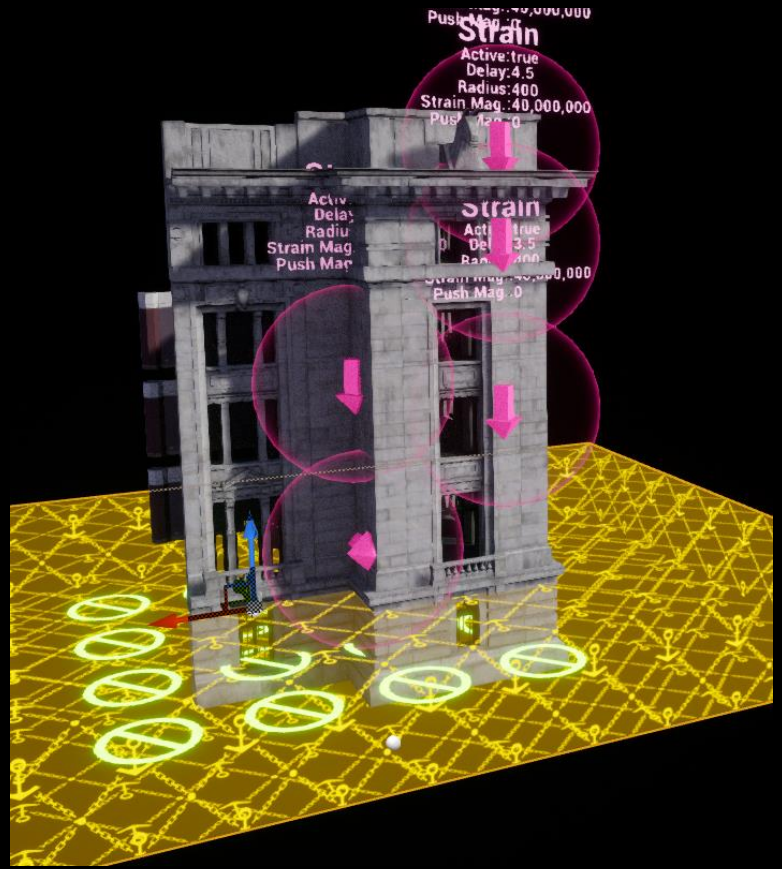
Connection Graph



Connection Graph

Chaos Field System

Fields put the **power** in the hands of **Artists**



Large Scale Simulation **Caching**

High fidelity simulations timed and driven using fields.

The result can be tweaked...

...then Pre-cached, and then played back in realtime...

Other fields can then be introduced to interact with the result.

The Tech **Behind Chaos**

Benn Gallagher
Senior Physics Programmer
Epic Games

The Tech Behind Chaos

Give the Power to Content Creators

Scalable

Physics

- Non Convex Collisions
- Fields
- Niagara Integration
- Interactive Caching
- Dedicated Physics Thread

Destruction

- Geometry Collection
- Cutting Tools
- Destructible LODs
- Dynamic Strain Evaluation

Non-Convex Collisions

Art directed

Accurate

Fast

Fields

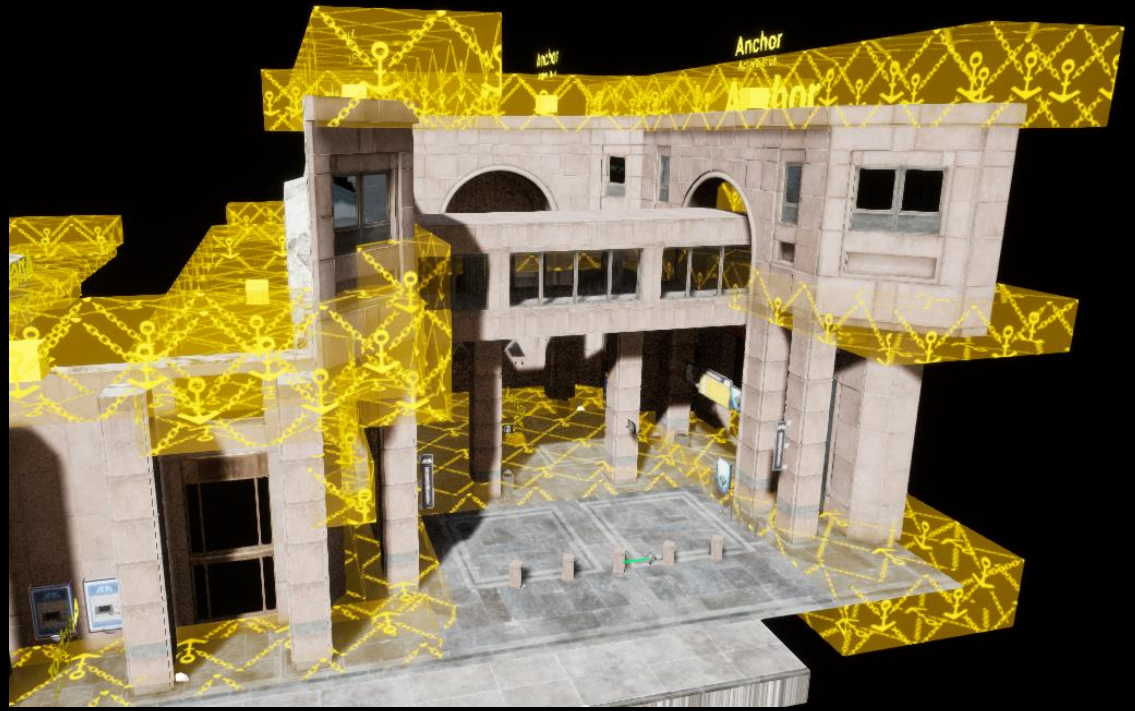
Can set physics parameters for a lot of objects at once

Strain, Force, Friction, etc.

Anchor Fields

Anchoring

Lock part of a geometry collection in place with an anchor field



Chaos Field System: **Strain**

Strain – Force – Decay

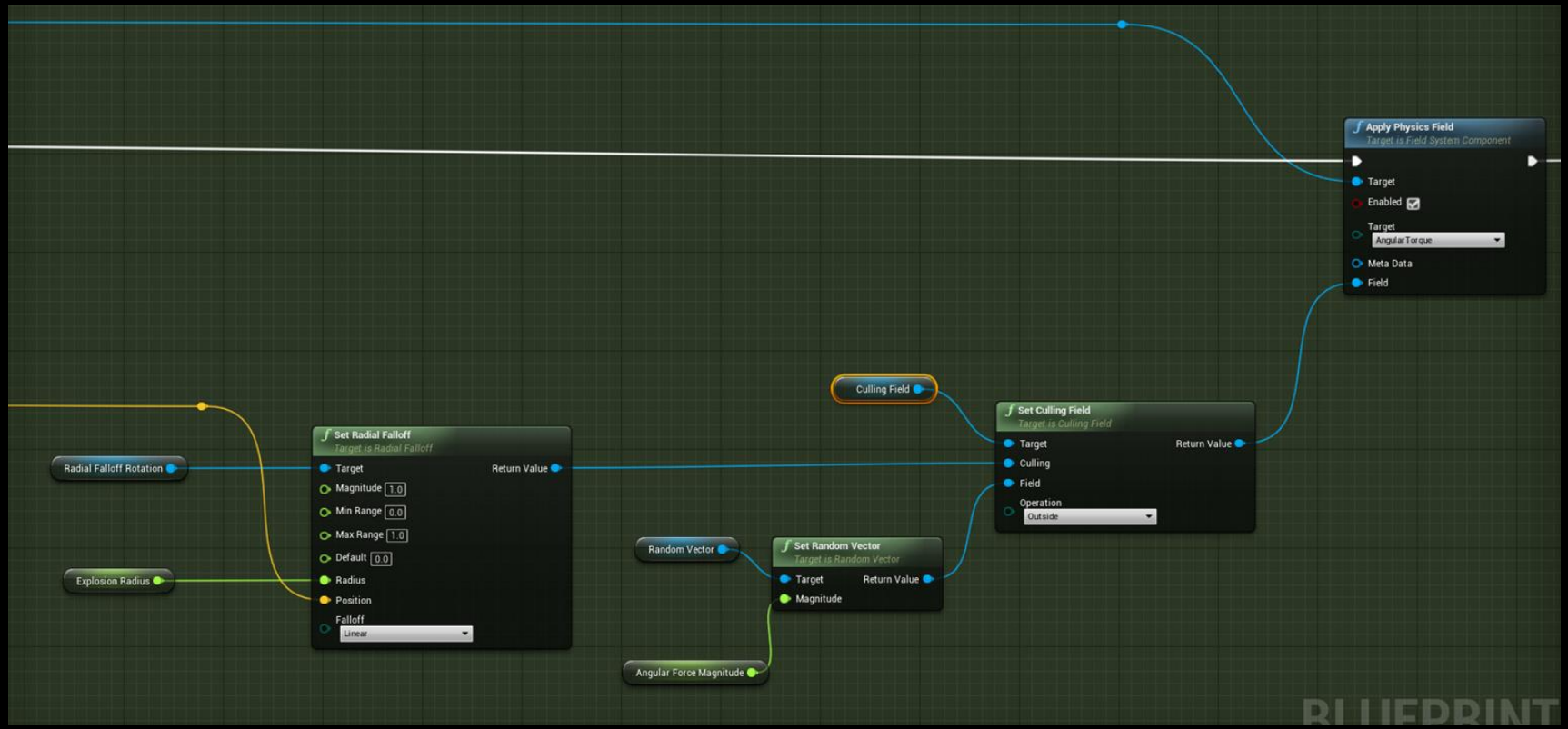
Chaos Field System: **Sleep, Disable**

Sleep fields control dynamic objects on the ground or outside of your gameplay area

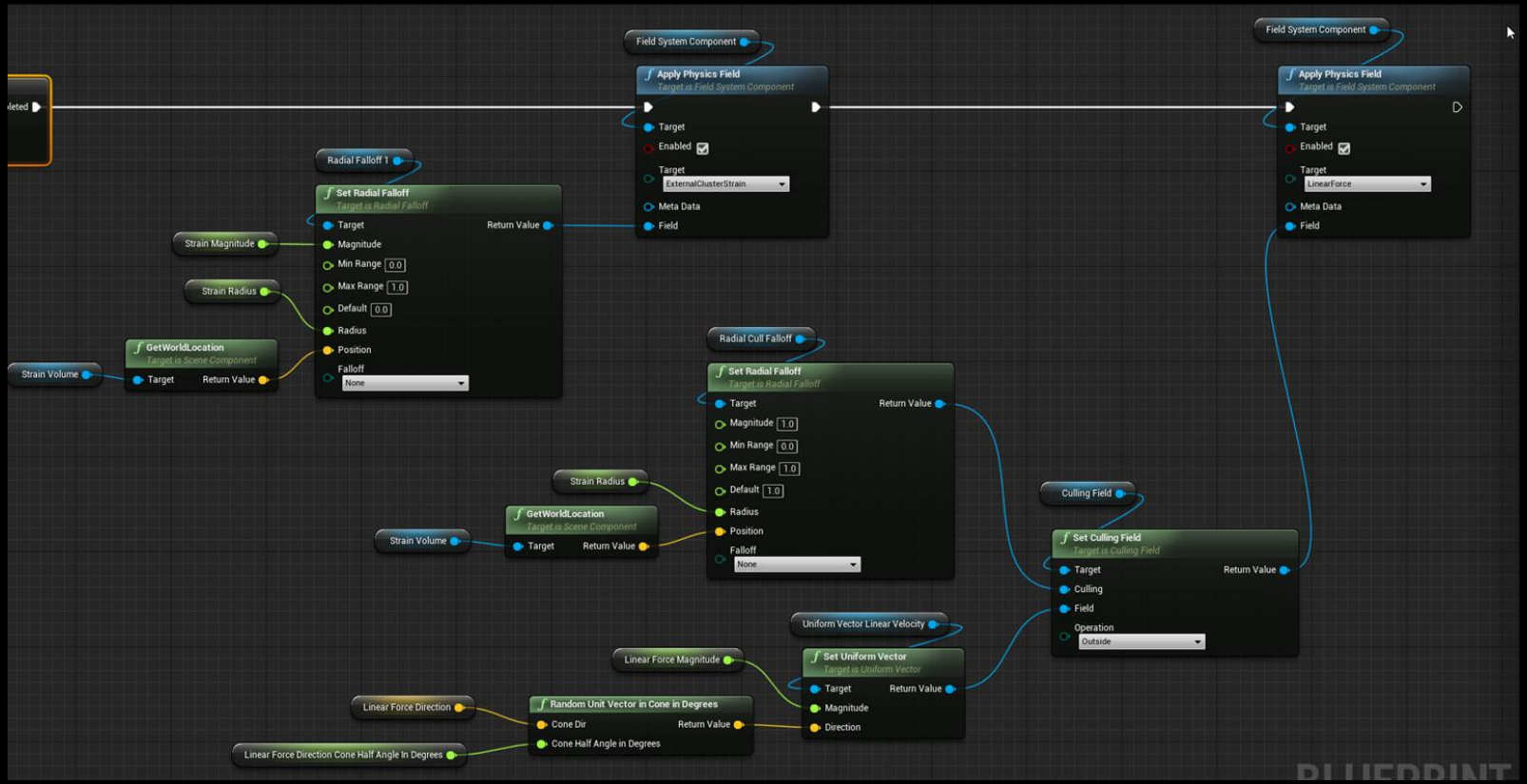
Sleep / Disable Field

- Still interactive
- Looks better
- Performance Control

Fields


Apply Physics Field
Target is Field System Component
Target
Enabled
Target
Angular Torque
Meta Data
Field
Culling Field
Set Culling Field
Target is Culling Field
Target
Culling
Field
Operation
Outside
Return Value
Set Radial Falloff
Target is Radial Falloff
Target
Magnitude 1.0
Min Range 0.0
Max Range 1.0
Default 0.0
Radius
Position
Falloff
Linear
Return Value
Radial Falloff Rotation
Explosion Radius
Random Vector
Set Random Vector
Target is Random Vector
Target
Magnitude
Return Value
Angular Force Magnitude
BLUEPRINT

Fields


Field System Component
Apply Physics Field
Target is Field System Component
Target
Enabled
Target
ExternalClusterStrain
Meta Data
Field
Field System Component
Apply Physics Field
Target is Field System Component
Target
Enabled
Target
LinearForce
Meta Data
Field
Radial Falloff 1
Set Radial Falloff
Target is Radial Falloff
Target
Return Value
Magnitude
Min Range 0.0
Max Range 1.0
Default 0.0
Radius
Position
Falloff
None
Strain Magnitude
Strain Radius
Strain Volume
GetWorldLocation
Target is Scene Component
Target
Return Value
Radial Cull Falloff
Set Radial Falloff
Target is Radial Falloff
Target
Return Value
Magnitude 1.0
Min Range 0.0
Max Range 1.0
Default 1.0
Radius
Position
Falloff
None
Strain Radius
Strain Volume
GetWorldLocation
Target is Scene Component
Target
Return Value
Culling Field
Set Culling Field
Target is Culling Field
Target
Return Value
Culling
Field
Operation
Outside
Uniform Vector Linear Velocity
Linear Force Magnitude
Set Uniform Vector
Target is Uniform Vector
Target
Return Value
Magnitude
Direction
Linear Force Direction
Random Unit Vector in Cone in Degrees
Cone Dir
Return Value
Cone Half Angle in Degrees
Linear Force Direction Cone Half Angle In Degrees

Niagara

Collision and breaking events are transferred to niagara for dust and debris simulations

Can also be used by other subsystems

Interactive Caching

Simulations can be recorded and played back in engine

When playing back they can be activated and start interacting with characters

Amount of computation is low but still have a completely interactive environment

Physics Thread

Physics has the option to completely decouple from the rest of the game

Geometry Collections

All destructible objects are represented by geometry collection actors

Single draw call per material across entire geometry colleciton object

Cutting Tools

Can break any static mesh

Voronoi fracture

Plane cutting

Image based fracture

Destructible LODs

Destructible Objects are authored in Levels

Secondary breaking

Scalability

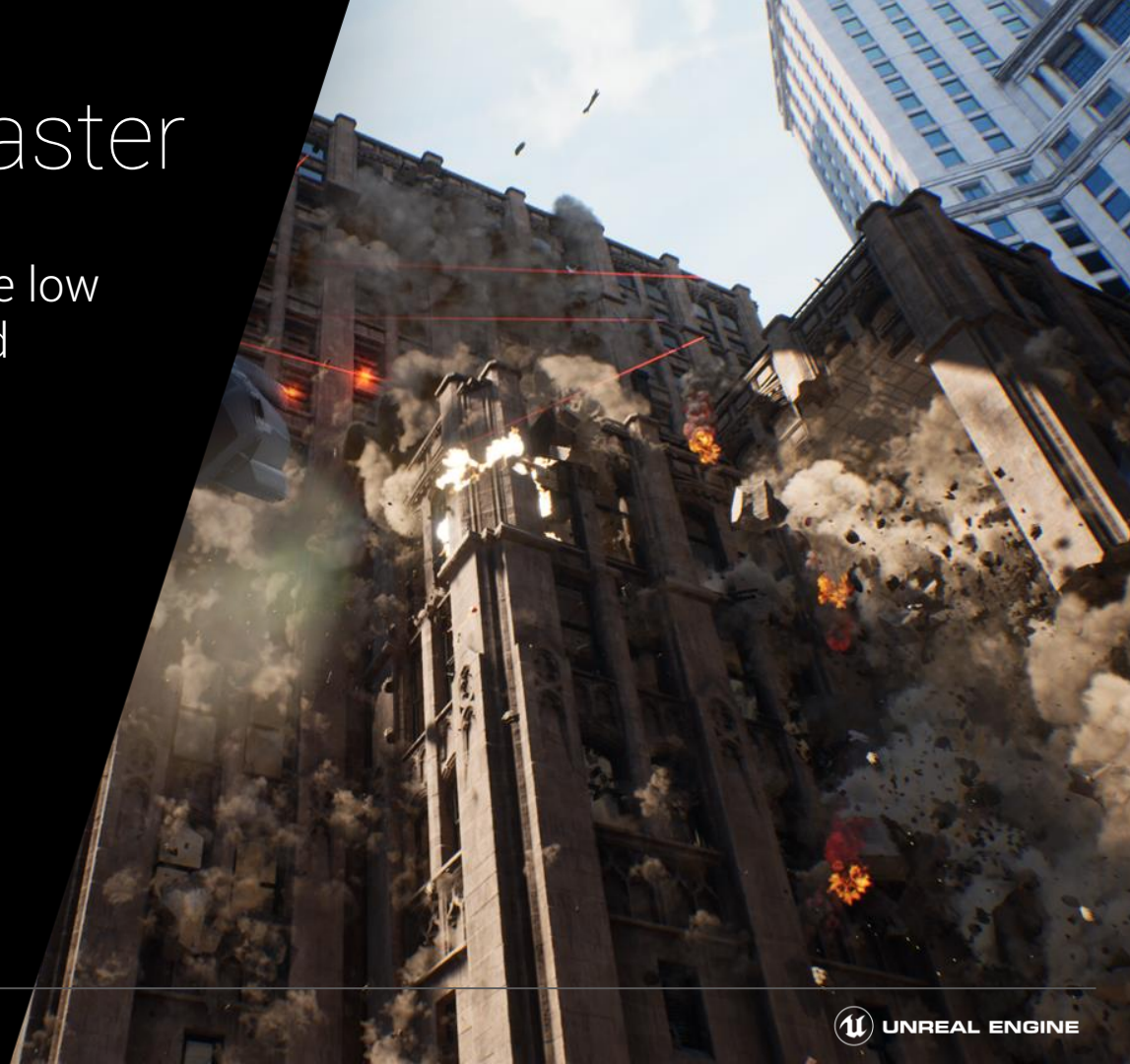

Pull Left Trigger at Bottom Left of Screen
Switched to camera CineCameraActor4
Letterbox LetterBox screenSize: 2,560x1,440 Calculated height has been set to 0px per half. ViewportScale 1.333
BP_ChaosPlayerCharacter_Vignette1
1
Rotating_sun
ROCKETS
Letterbox LetterBox screenSize: 2,560x1,440 Calculated height has been set to 0px per half. ViewportScale 1.333
Letterbox LetterBox screenSize: 2,560x1,440 Calculated height has been set to 0px per half. ViewportScale 1.333
Letterbox LetterBox screenSize: 2,560x1,440 Calculated height has been set to 0px per half. ViewportScale 1.333
Letterbox LetterBox screenSize: 2,560x1,440 Calculated height has been set to 0px per half. ViewportScale 1.333
Letterbox LetterBox screenSize: 2,560x1,440 Calculated height has been set to 0px per half. ViewportScale 1.333
Letterbox LetterBox screenSize: 2,560x1,440 Calculated height has been set to 0px per half. ViewportScale 1.333
ROBO READY

Making **Chaos Faster**

Kelly D Gawne
Game Performance Engineer
Intel

Making Chaos Faster

Intel worked with Epic to optimize low level solvers, data structures, and thread parallelism.



Making Chaos Faster

Key Learnings:

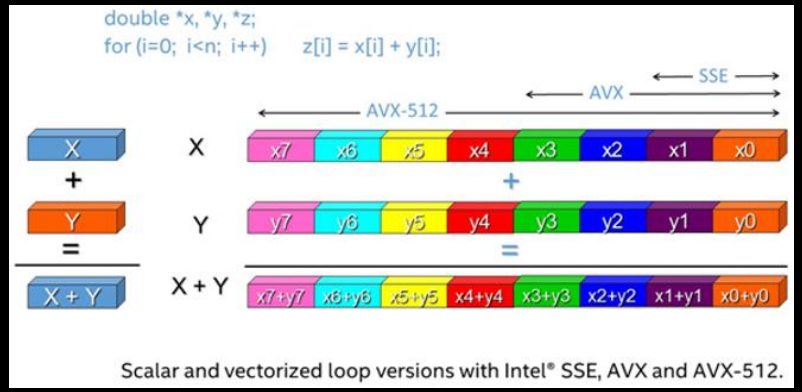
- **C++ is bad at SIMD**. Integrated Intel ISPC for ~3x gains in perf critical areas.
- **TSet can be bad in high performance situations.** Use TArray, Sort and RemoveSwap rather than guard dupes with TSet.Contains.
- **ParallelFor can be overused!** Some oversubscription is good but don't go overboard with 1000s of jobs. Batch!

Intel SPMD Program Compiler (ISPC)

Problem: In perf critical code, C++ often doesn't cut it. Usual solution is intrinsics.
Not anymore!

- **Implicit parallelism**: SIMD lanes act similar to GPU shader invocations
- **Write Once**, Compile to many vectorized instruction sets (SSE4, AVX, AVX2)
- Used in Chaos, available in UE4 soon!

- Include ISPC module in your build.cs
- Add ispc files to your project
- Include a generated C++ header
- Unreal Build Tool handles the rest



Scalar and vectorized loop versions with Intel® SSE, AVX and AVX-512.

When to Use ISPC?

- Dense compute-bound workloads. Heavy math like physics intersection testing, cloth or CPU vertex transformations
- Best with contiguous memory load, manipulate, store ie TArray
- Best when no data dependencies between operations. Especially useful when combined with ParallelFor and batching

```
export void rgb2grey(uniform int N, uniform float r[],
uniform float g[], uniform float b[], uniform float grey[])
{
      foreach (idx = 0 ... N)     {
            grey[idx] = 0.3f * r[idx] +
            0.59f * g[idx] +
            0.11f * b[idx];
      }
}
```

```
…
vmulps (%rdx,%rax), %ymm1, %ymm3
vfmadd231ps (%rsi,%rax), %ymm0, %ymm3
vfmadd231ps (%rcx,%rax), %ymm2, %ymm3
vmovups %ymm3, (%r8,%rax)
…
```

UNREAL
ENGINE


09