
Unreal Engine
From Shooter to Simulation

About

```
$ whoami
┌─[ pascal@roguesheep.nl ]──────────────────────────┐
| role  :: software engineer | Unreal developer     |
| stack :: VR · UE · C++ · Android                  |
|                                                   |
| toolchain :: Unreal · Rider · Git · Blender       |
| traits :: pragmatic && autonomous                 |
| status :: Available for complex problems          |
|           and fun ideas                           |
| required :: Clear specifications                  |
└───────────────────────────────────────────────────┘
```

About

```
$ history
1988 Computer sciences at RU Nijmegen
1994 Consultancy work
1997 Co Founder and developer at <GX> Software
2004 Freelance web developer
2017 Co Founder and developer at VR Lab
2023 Freelance Unreal developer and trainer
```

Unreal Engine?

Origins

Founder: Tim Sweeney (Epic Games).

Debut: Launched with the FPS game *Unreal* in 1998.

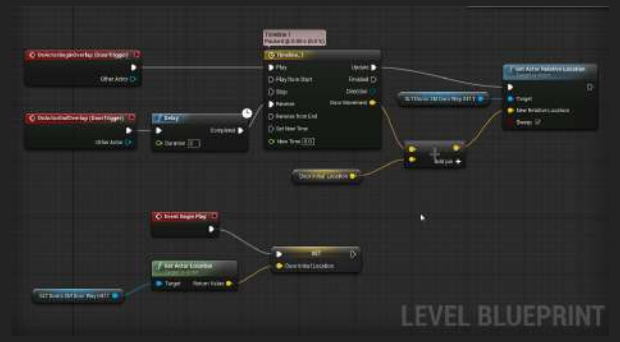
Innovation: Modular architecture separating game logic from rendering.

Pivot: Licensing model democratized high-end development.

Why Unreal?

Visual Fidelity

Unreal Engine 5 (UE5) bridges the gap between movie-quality CGI and **real-time** performance.



Accelerated Workflow

Build faster with tools designed to empower both engineers and artists.

Blueprints: Visual Scripting

Quixel Megascans

MetaHuman Creator

Full Control

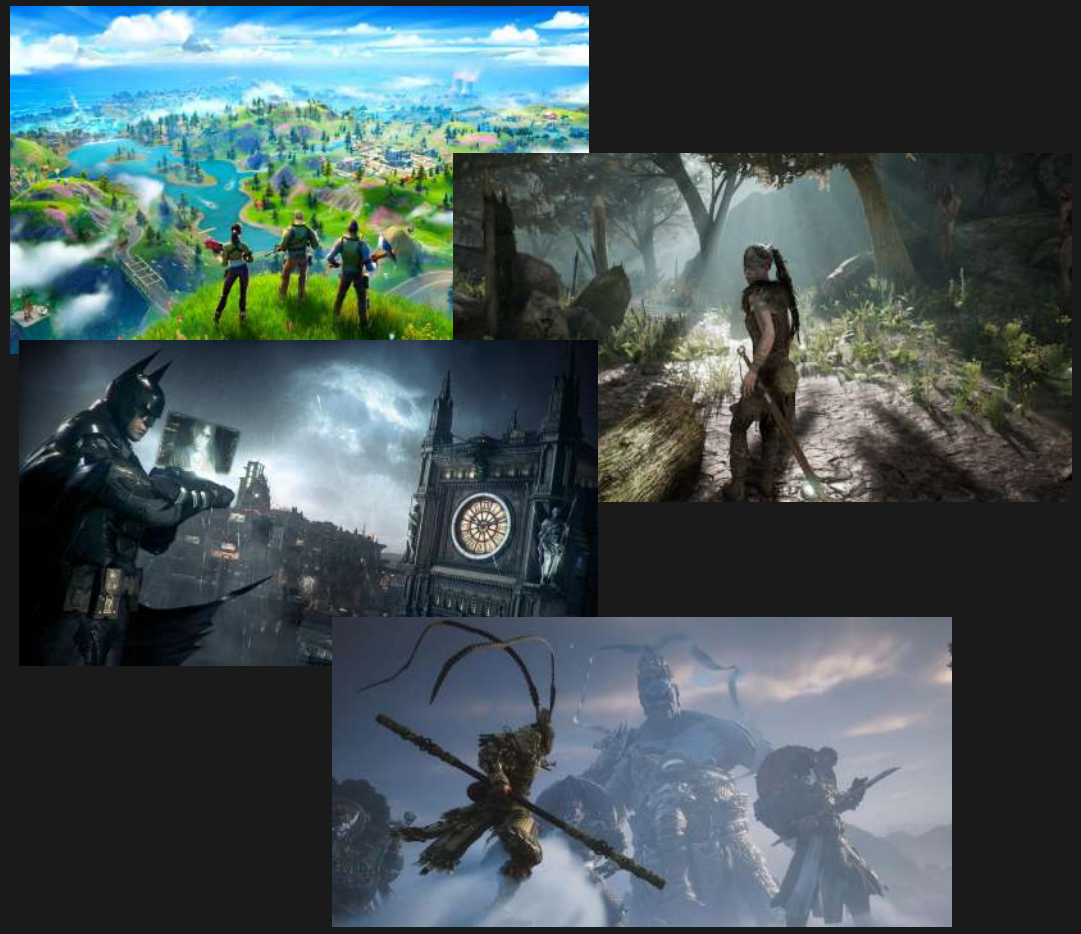
Open Source

Full C++ Source Code Access

Extendable through plugins

Cross-Platform Scalability

The Gold Standard in Gaming



Examples

Unreal Tournament

***Hellblade**: Senua's Sacrifice*

***Batman**: Arkham series*

Gears of War

Fortnite (1-3 million concurrent players)

Valorant

Black Myth: Wukong

Film & TV

Virtual Production & ICVFX

Tech: In-Camera Visual Effects (ICVFX) on LED walls.

Benefit: Real-time lighting and "in-camera" backgrounds; reduces post-production.

Examples:

- ***The Mandalorian***
- ***Westworld***
- ***Ford v Ferrari***
- ***Dune 2***
- ***1899***

Automotive & Manufacturing

Digital Twins & HMI

Configurators: Real-time customization (ray tracing).

HMI: In-car dashboards and infotainment.

Simulation: Autonomous driving training.

Design: customer experience

Examples

- **Rivian** & GM **Hummer** EV: Dashboard UI running on Unreal.
- **BMW** & **Audi**: Digital factory planning. Design & engineering

Architecture & Simulation

ArchViz

VR walkthroughs

Twinmotion integration

BIM visualisation

Simulation

Flight simulators

Engineering

Smart cities

Virtual Reality

Engineering

Visual and Spatial testing

History/future

Flight simulators

Exposure therapy

smart cities

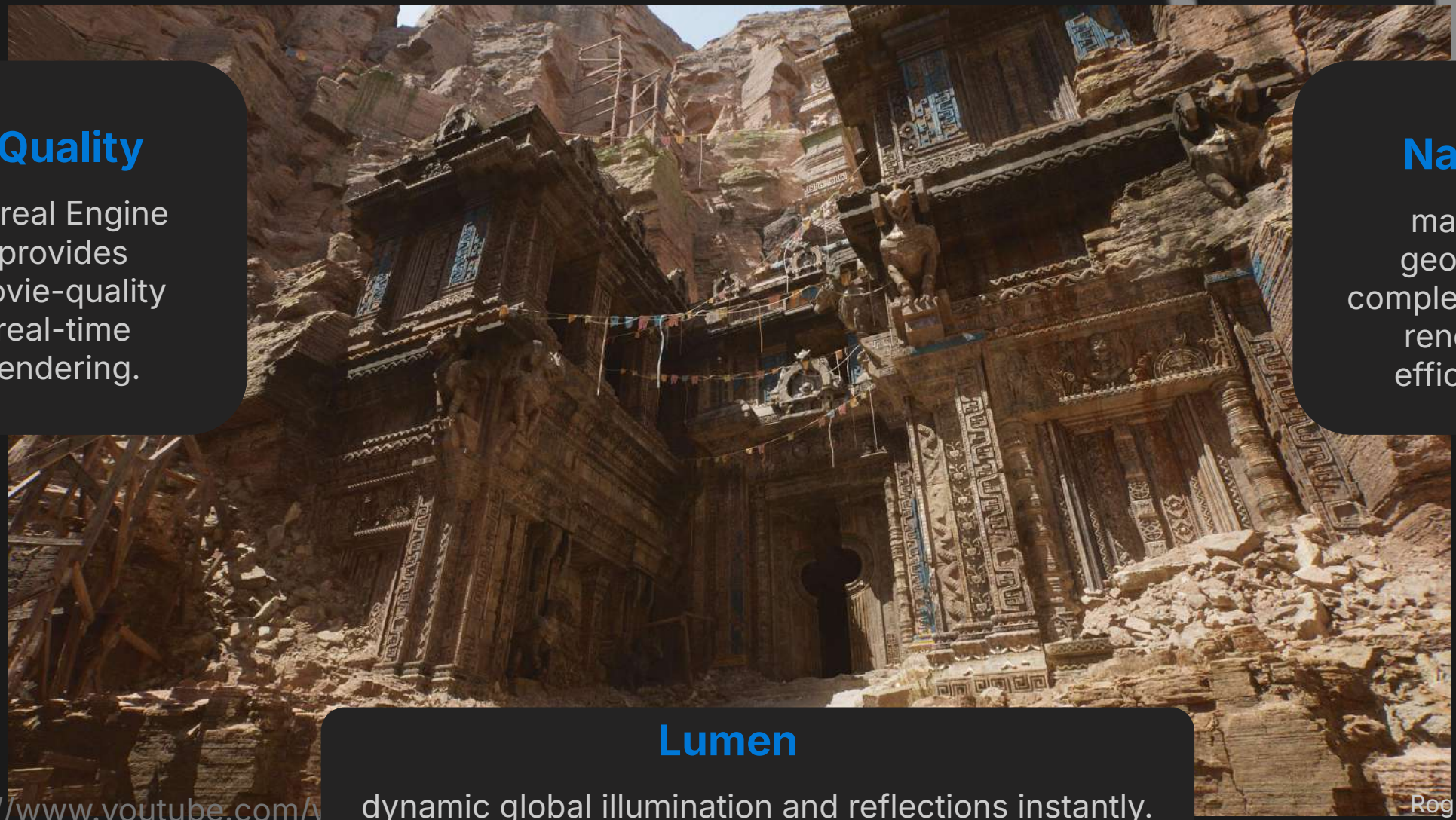
Training

Flight simulators

Exposure therapy

smart cities

Real-Time Rendering



Quality

Unreal Engine provides movie-quality real-time rendering.

Nanite

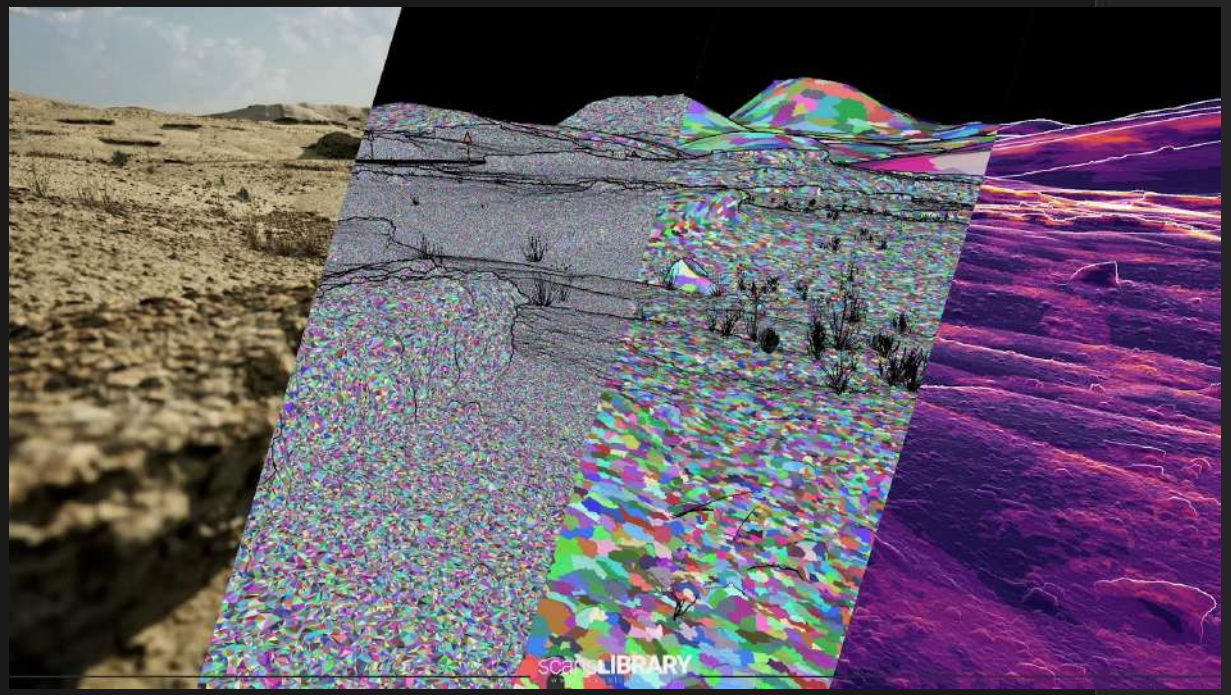
massive geometry complexity to be rendered efficiently.

Lumen

dynamic global illumination and reflections instantly.

https://www.youtube.com/v

Key Features: Nanite



- Nanite uses a virtualized micro-polygon geometry system.
- It allows importing cinema-quality assets with billions of triangles.
- Meshes scale efficiently without major loss in visual fidelity.
- Performance is boosted by only rendering necessary detail.

Key Features: Lumen

- Lumen provides dynamic global illumination and reflections.
- Indirect lighting adapts instantly to changes in geometry or light.
- It is a fully software-based ray tracing solution.
- Lumen replaces screen space reflections and complex light bakes.

Meta Humans

- MetaHuman Creator produces realistic digital humans fast.
- Quickly design, customize, and animate unique characters.
- High-quality models are compatible with Unreal Engine 5.
- Free cloud-based application integrates into your workflow.

https://www.youtube.com/watch?v=T38ZURD768w

Strategic Acquisitions

Photorealism & Assets

- Quixel (2019)
- Capturing Reality (2021)
- Sketchfab (2021) & ArtStation (2021)

Digital Humans

- 3Lateral (2019) (digital human creation)
- Cubic Motion (2020) (real-time facial animation)
- Hyprsense (2020) (webcam face tracking)
- Shave and a Haircut (2019)

Visualization

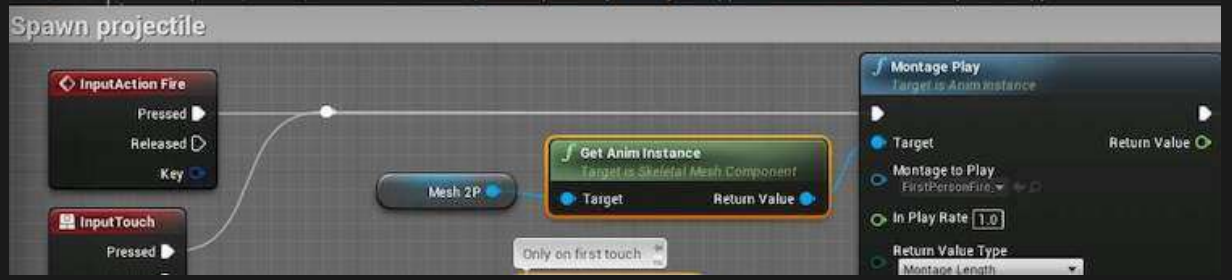
- Twinmotion (2019)

Technical Infrastructure & Performance

- RAD Game Tools (2021) (data compression)
- Loci (2025): (AI-powered 3D content tagging)

Classes, Blueprints, Assets and Actors

```
62      PlayerInputComponent->BindAxis("MoveForward", this, &ACppIntroCharacter::MoveForward);
63      PlayerInputComponent->BindAxis("MoveRight", this, &ACppIntroCharacter::MoveRight);
64
65      // We have 2 versions of the rotation bindings to handle different kinds of devices differently
66      // "turn" handles devices that provide an absolute delta, such as a mouse.
67      // "turnrate" is for devices that we choose to treat as a rate of change, such as an analog joystick
68      PlayerInputComponent->BindAxis("Turn", this, &APawn::AddControllerYawInput);
69      PlayerInputComponent->BindAxis("TurnRate", this, &ACppIntroCharacter::TurnAtRate);
70      PlayerInputComponent->BindAxis("LookUp", this, &APawn::AddControllerPitchInput);
71      PlayerInputComponent->BindAxis("LookUpRate", this, &ACppIntroCharacter::LookUpAtRate);
```



1. The Foundation: C++ Native Class
2. Unreal C++ Code adds specialized engine functionalities.
3. The Bridge: Blueprint Asset

   A serialized binary file on disk
4. The Result: Blueprint Generated Class
5. Visual scripting allows fast iteration without C++ access.

The Editor

1. Main Menu
2. Top Menu
3. Viewport - *The Stage*
4. Outliner - *Cast & Crew list*
5. Details - *The Clipboard*
6. Content Drawer - *The Warehouse*

Let’s dive in!

Questions?