Game Engine

Architecture, Game Loop, Unity

Tomáš Polášek ipolasek@fit.vutbr.cz

Brno University of Technology

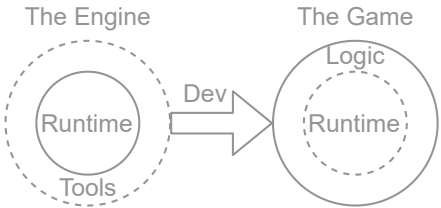
Faculty of Information Technology
DCGM, cPhoto@FIT
Faculty of Fine Arts
Game Media Studio

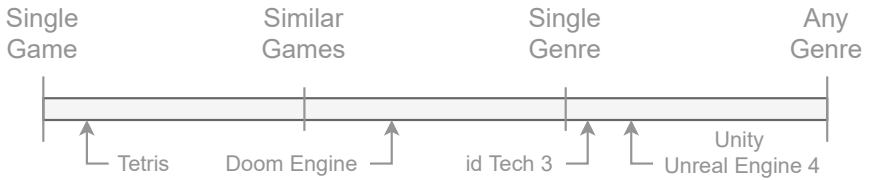
3

What is a Game Engine?

- Reusable Software → Platform
- **Goal**: Simplify Game Development
- There is **One**There are **Many**
- Build Your Own?
- "Choosing the Right Tool for the Job"
  - Licensing & Royalties
  - Internal tools
  - Ecosystem Integration
  - Target Platform
  - Game Genre, Style, Gameplay
  - Community

- "Fun Activity" vs "Soft Real-Time Simulation"
- Game Engine = **Runtime** + **Tools**
- Gamut of **Reusability**
- Generality × Optimality



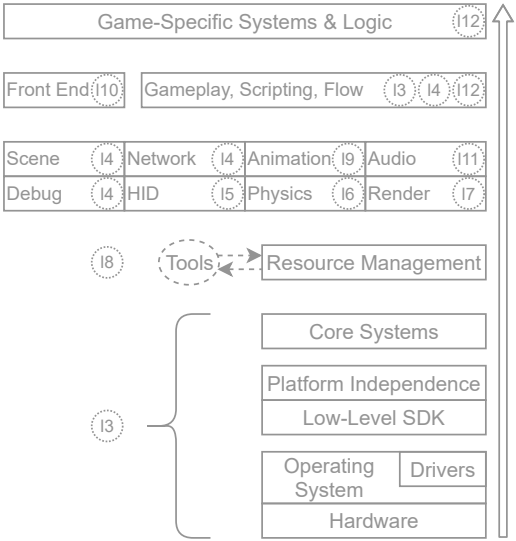


Source: Game Engine Architecture [1]

Engine Architecture

Game Engine Overview

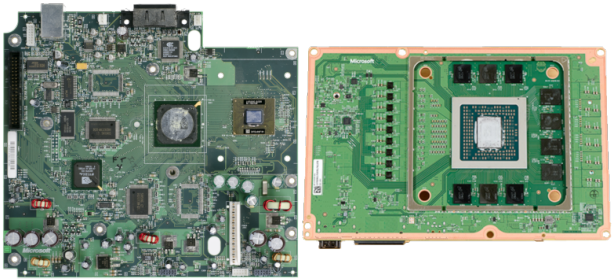
- **Complex** Software Architecture [1]
- Layers & Dependencies
- System Overview



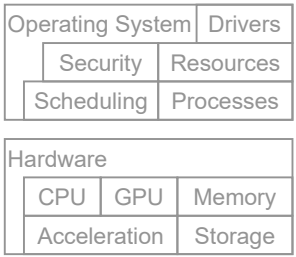
[1] Jason Gregory – Game Engine Architecture [1]

Hardware & OS

- Architecture, Optimization
- Varied → Uniform
- Operating System = Software Layer
- Drivers = Hardware Interface

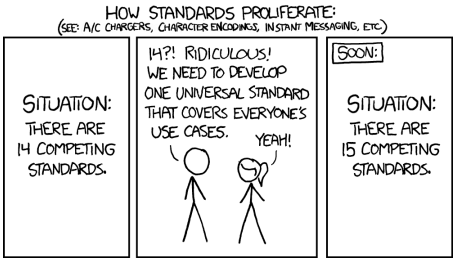


Source: Xbox 2001, Xbox Series S 2020



- APIs, SDKs, Standard Library, ...

Low-Level Libraries		
Middleware	SDKs	APIs

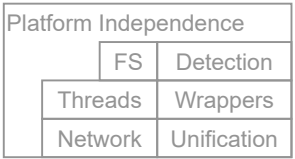


Source: XKCD – Standards

Platform Independence Layer

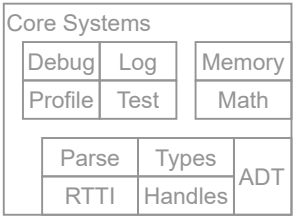
- Platform Independence
- Unified Interface
- Multiplatform Development

Source: Unreal, Unreal Engine 5



Core Systems

- Library of Utilities
- Base Debugging & Logging
- Memory Management
- Data Types
- Serialization, RTTI, Parsing
- Mathematics – Transform, Geometric, Solvers
- ...

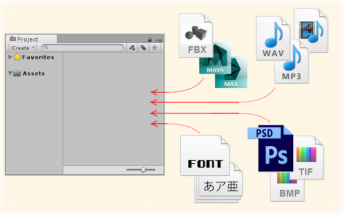



```
Player player = new Player();
player.level = 1;
player.health = 100.0f;
player.name = "Thomas";
```

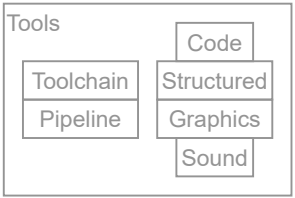
```
{
    "level": 1,
    "health": 100.0,
    "name": "Thomas"
}
```

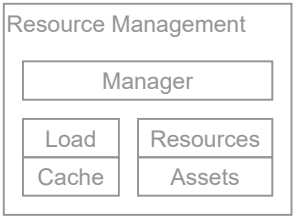
Resource Management

- Unified Data Access
- Resources & Assets
- **Toolchain → Pipeline → Assets**
- “Mark of Style”



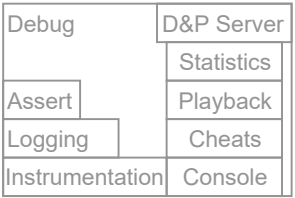
Source: Unity Manual – Asset Workflow

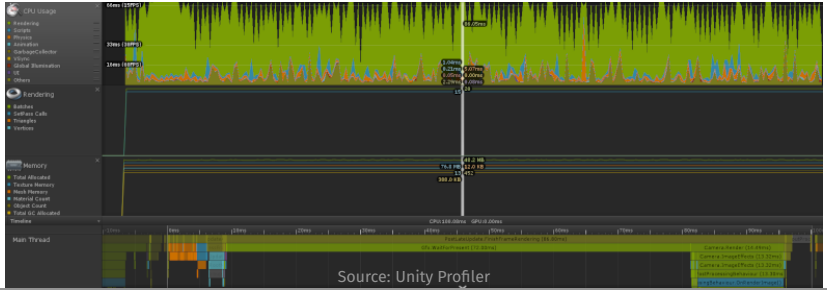




Debugging & Profiling

- Complexity → Bugs
- What is Wrong?
- **Instrumentation** & **Logging**
- Playtime Statistics, Core Dumps
- Remote Debugging & Profiling

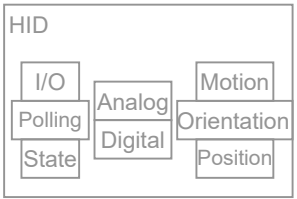




Source: Unity Profiler

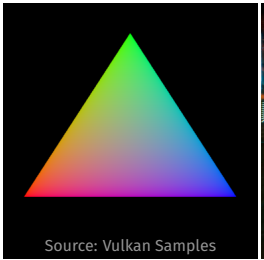
Human Interface Devices

- **Input** & **Output**
- Raw I/O → API
- Wide Range
- Device Sensors
- User Feedback



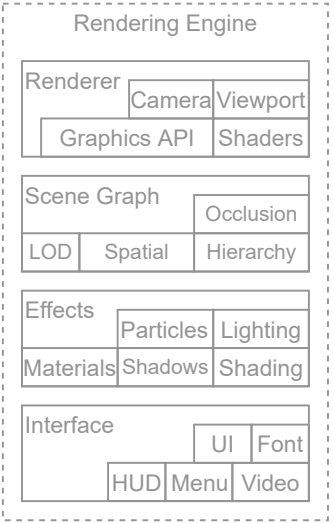
Rendering Engine

- Visualization of the Game
- “From Zero to Mirrors”
- Scene Graph
- Materials, Effects, Lighting
- Interface Rendering



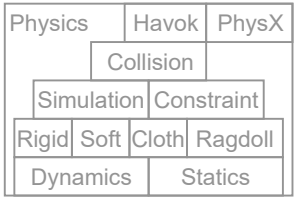
Source: Vulkan Samples

Source: Cyberpunk 2077



Physics

- Collision Detection → Resolution
- Physical Simulation
- Simplified Models
- Static & Dynamic Objects



Animation

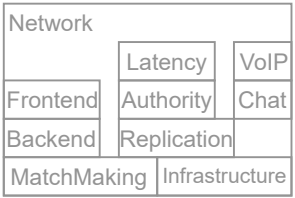
- Making Things Move
- Animation vs Physics
- Kinematics & Dynamics
- Rigging, Skinning

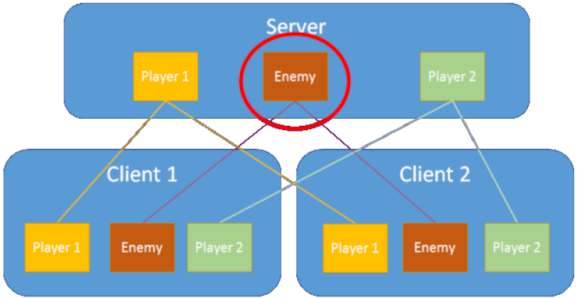
Source: Unreal, Unity

Animation		
Texture	Rigid	Skeleton
Sprite	Vertex	Morph
Kinematics		Skinning
Dynamics		Rigging

Network

- Network Stack
- Frontend × Backend
- Latency, Replication, Authority

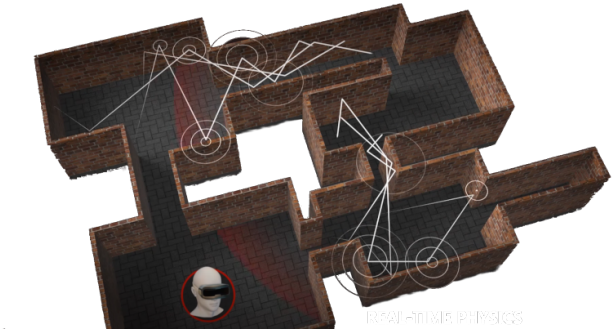




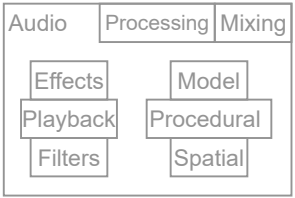
Source: Unity

Audio

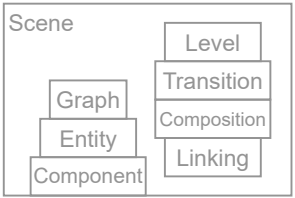
- Often Neglected
- Fidelity & Procedural
- Realistic Modeling

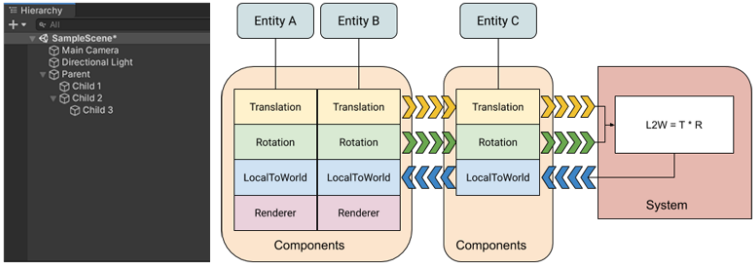


Source: AMD TrueAudio Next



- Graph Data Structure
- World Hierarchy
- Entity-Component-System
- Composition & Linking

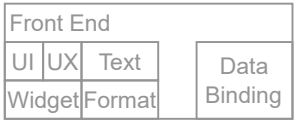




Source: Unity

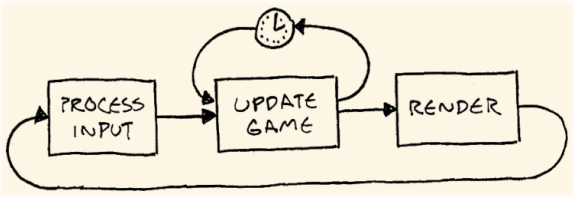
Front End

- UI & UX
- Aural Feedback
- Widgets, Layouts
- Data Binding

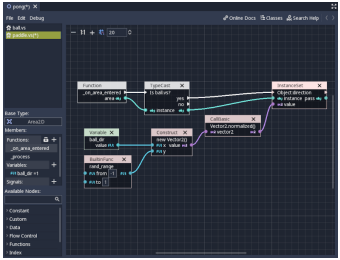


Source: Unity UI Builder

- Framework for Building the Game
- Scripting Utilities
- Low-Level $\Longleftrightarrow$ Gameplay
- Game Object Hierarchy
- Game Loop [2]



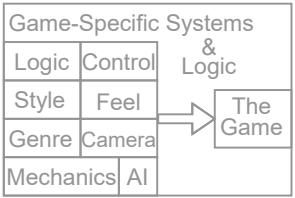
Source: Robert Nystrom – Game Programming Patterns



Source: Godot Visual Scripting

- This is **The Game**
- Gameplay Implementation

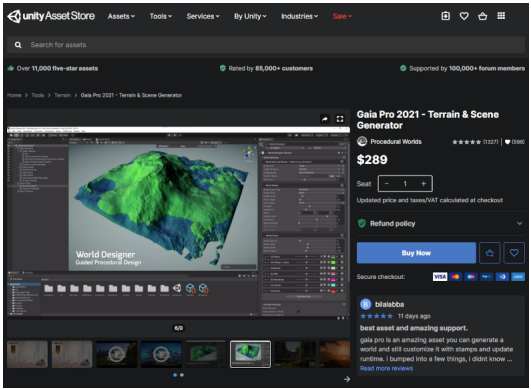
Source: Unity Bolt

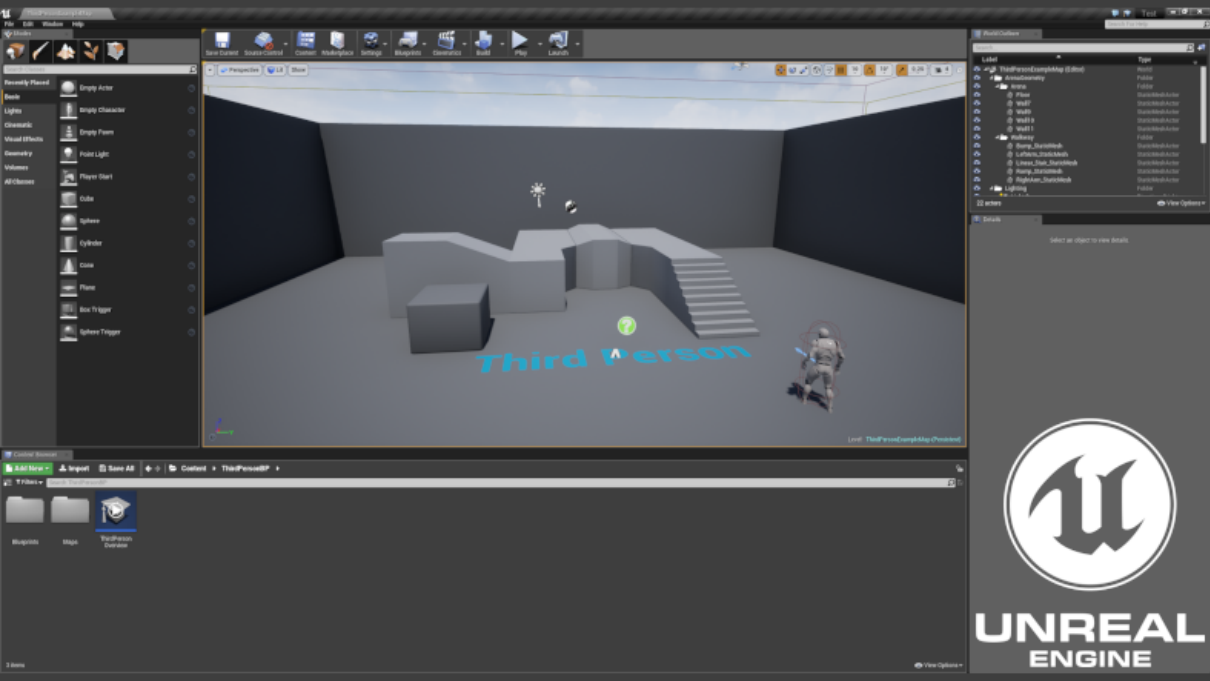


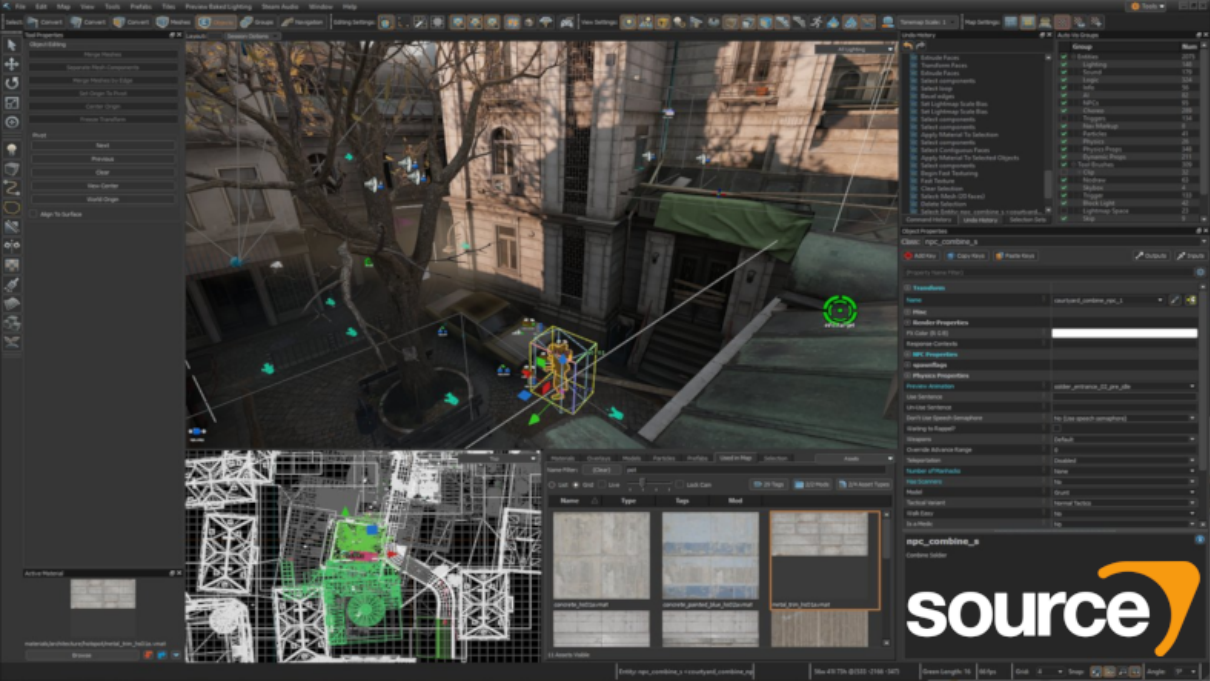
Unity Engine

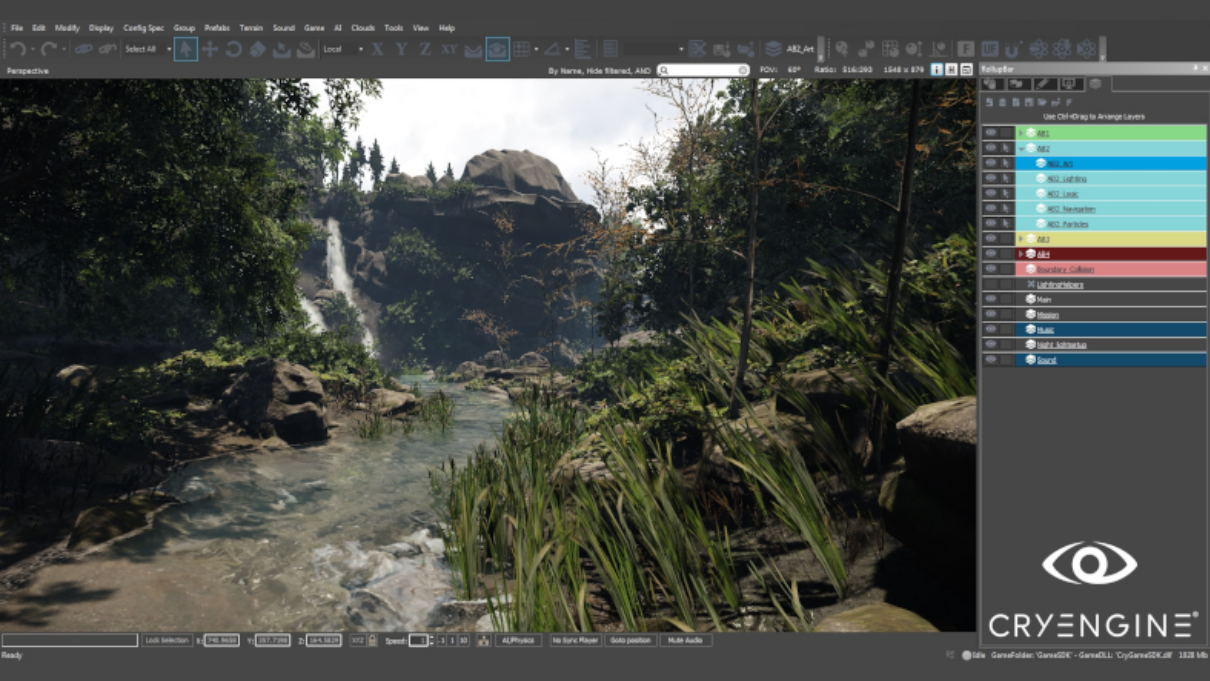
What is Unity?

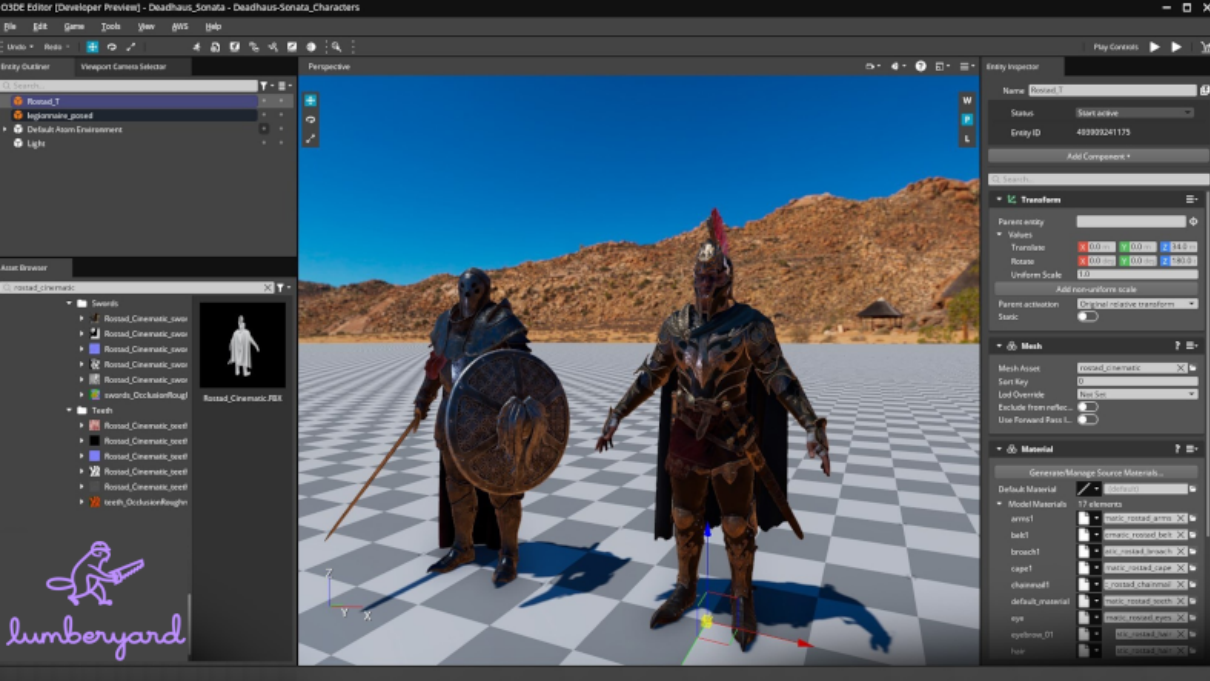
- Game Engine
- Development Platform
- Documentation & Community
- **Why Unity?**

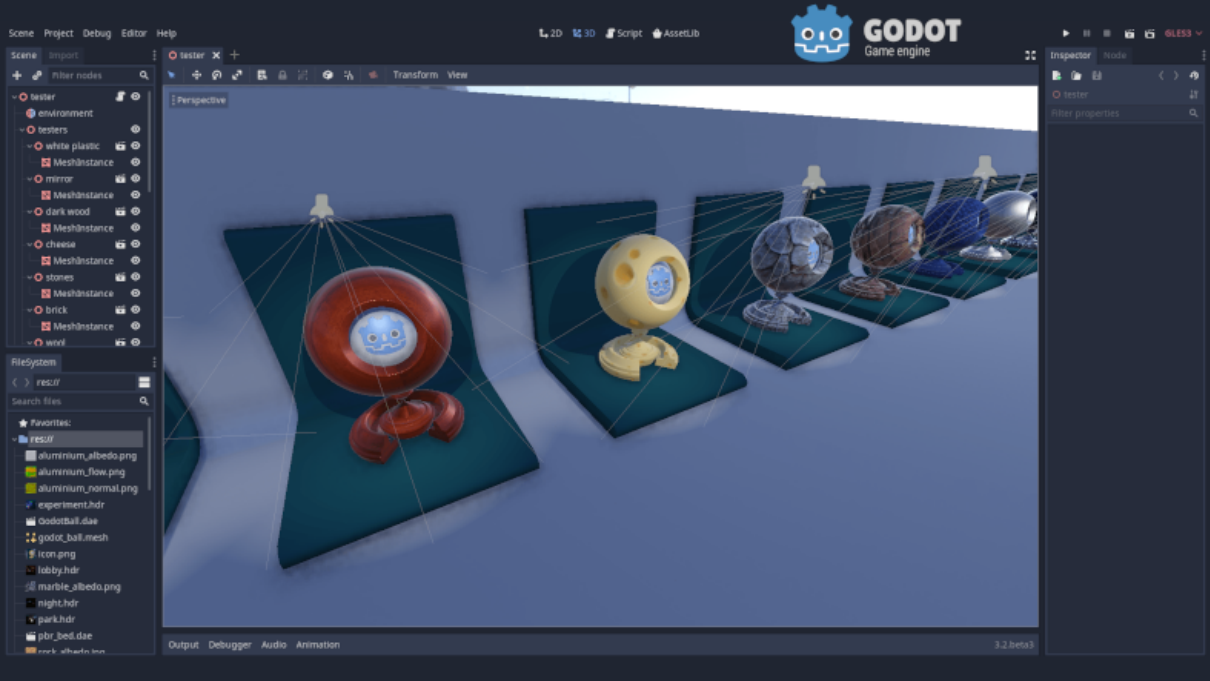


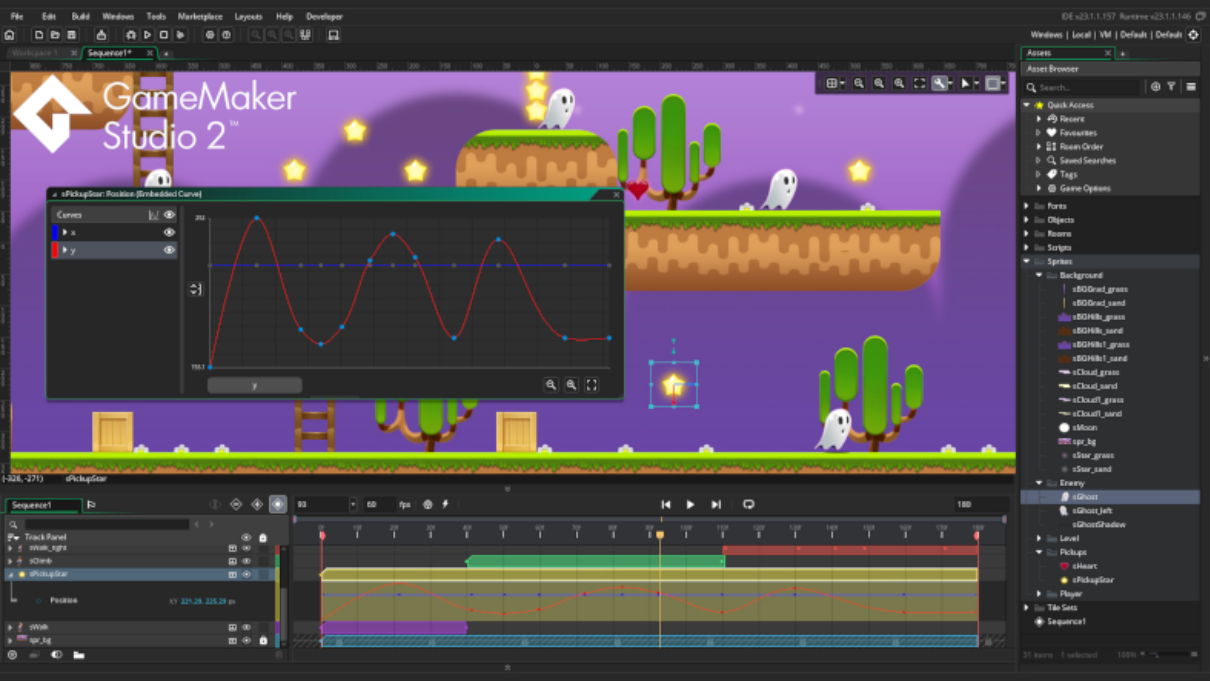

Third Person
UNREAL
ENGINE

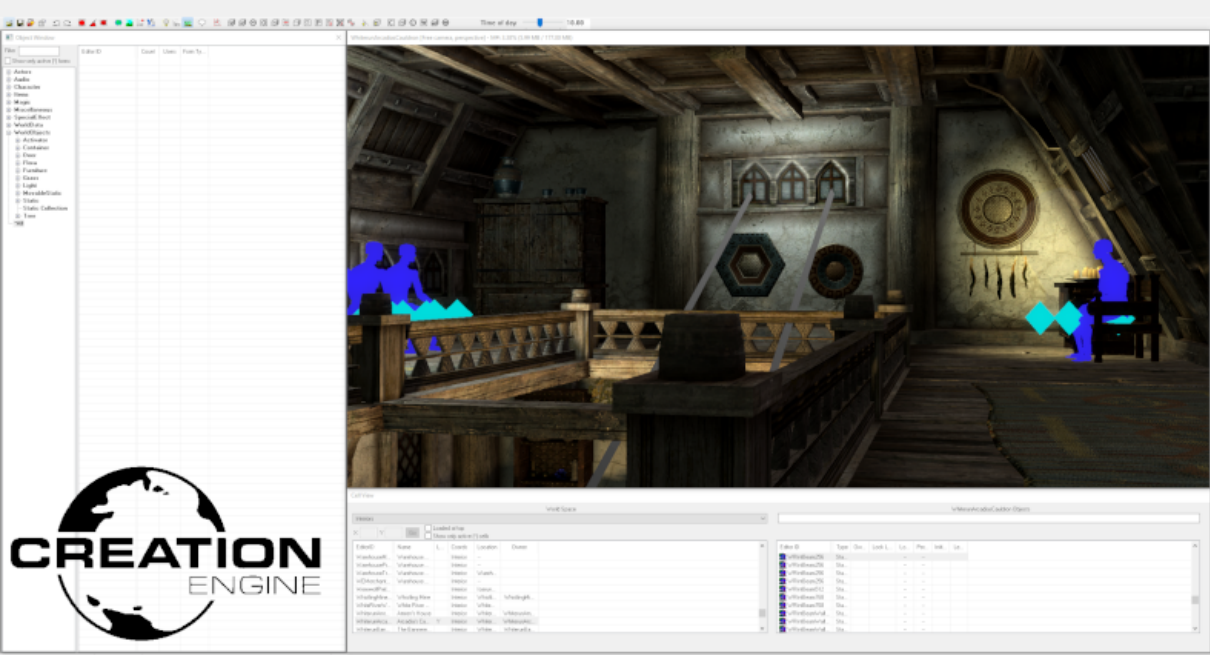

source

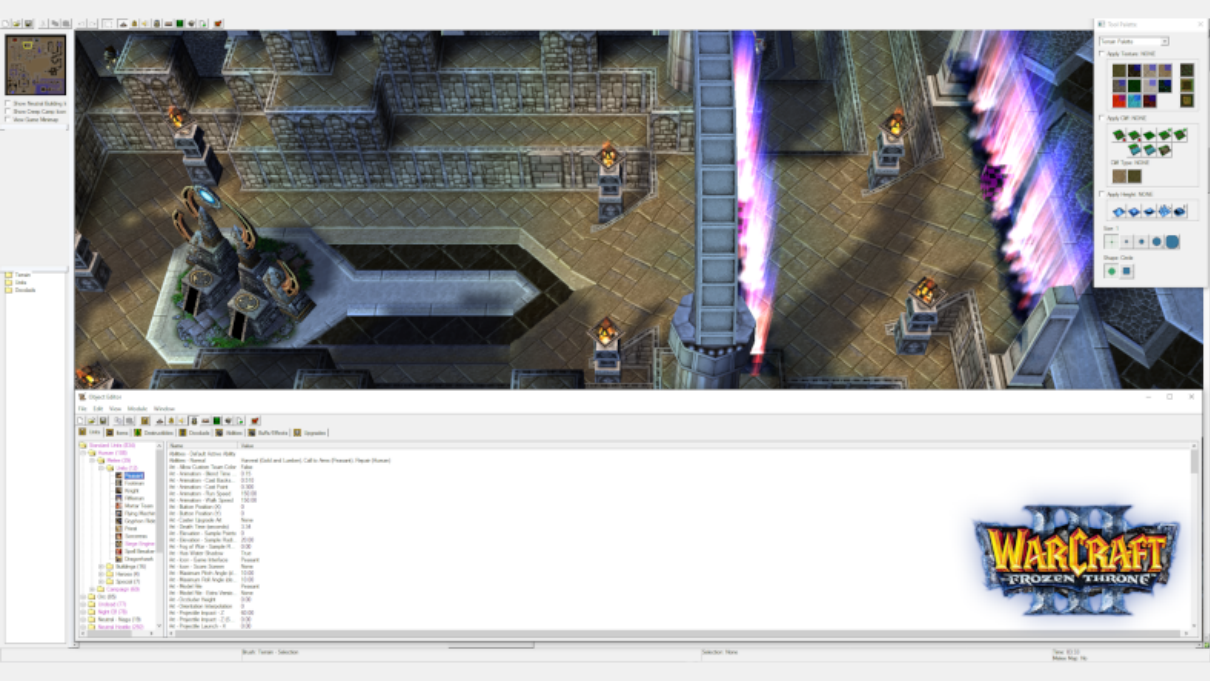

File Edit Modify Display Config Spec Group Prefabs Terrain Sound Game AI Clouds Tools View Help
Perspective
Rollupbar
Use Ctrl+Drag to Arrange Layers
CRYENGINE
Ready

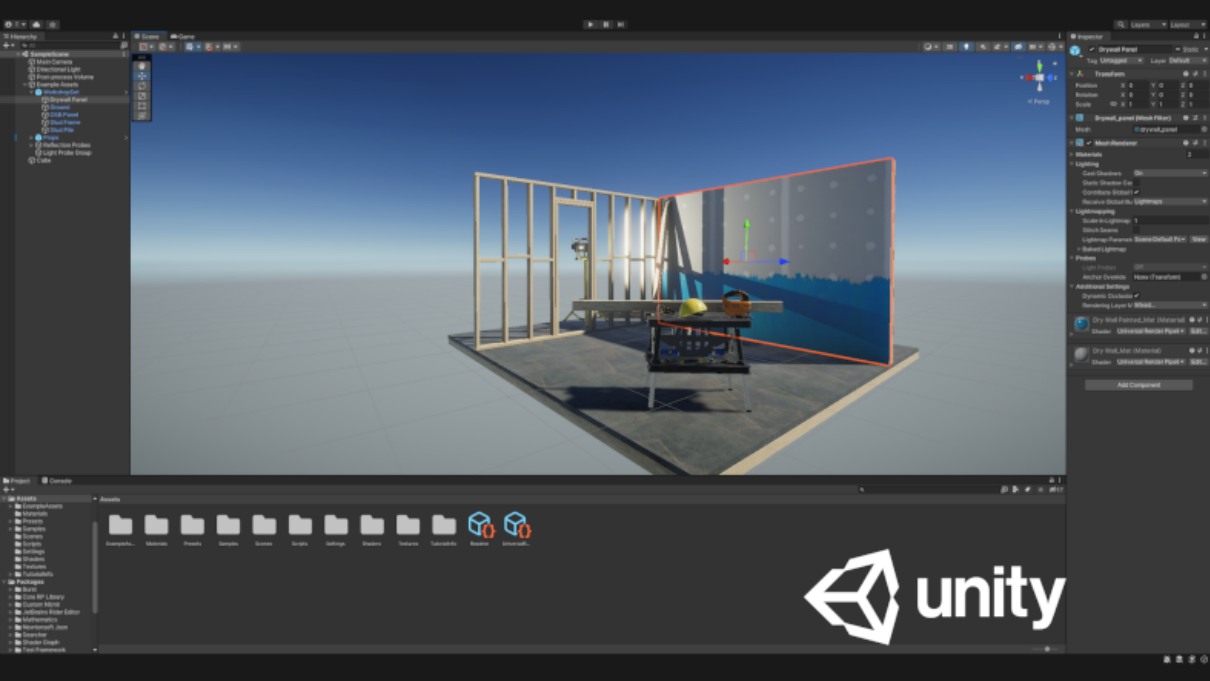

O3DE Editor [Developer Preview] - Deadhaus_Sonata - Deadhaus-Sonata_Characters
File Edit Game Tools View AWS Help
Undo Redo
Play Controls
Entity Outliner
Viewport Camera Selector
Perspective
Entity Inspector
Rostad_T
legionnaire_posed
Default Atom Environment
Light
Name
Rostad_T
Status
Start active
Entity ID
483093241175
Add Component +
Transform
Parent entity
Values
Translate
Rotate
Uniform Scale
1.0
Add non-uniform scale
Parent activation
Original relative transform
Static
Mesh
Mesh Asset
rostad_cinematic
Sort Key
0
Lod Override
Not Set
Exclude from reflec...
Use Forward Pass I...
Material
Generate/Manage Source Materials...
Default Material
Model Materials
17 elements
arms1
belt1
brooch1
cape1
chainmail1
default_material
eye
eyebrow_01
hair
Asset Browser
rostad_cinematic
Swords
Rostad_Cinematic_swor
Rostad_Cinematic_swor
Rostad_Cinematic_swor
Rostad_Cinematic_swor
Rostad_Cinematic_swor
swords_OcclusionRough
Teeth
Rostad_Cinematic_teeth
Rostad_Cinematic_teeth
Rostad_Cinematic_teeth
Rostad_Cinematic_teeth
Rostad_Cinematic_teeth
teeth_OcclusionRoughn
Rostad_Cinematic.FBX
lumberyard


Scene Project Debug Editor Help
2D 3D Script AssetLib
GODOT
Game engine
GLES3
Scene Import
Filter nodes
tester
environment
testers
white plastic
MeshInstance
mirror
MeshInstance
dark wood
MeshInstance
cheese
MeshInstance
stones
MeshInstance
brick
MeshInstance
FileSystem
res://
Search files
Favorites:
res://
aluminium_albedo.png
aluminium_flow.png
aluminium_normal.png
experiment.hdr
GodotBall.dae
godot_ball.mesh
icon.png
lobby.hdr
marble_albedo.png
night.hdr
park.hdr
pbr_bed.dae
tester
Transform View
Perspective
Inspector Node
tester
Filter properties
Output Debugger Audio Animation
3.2.beta3


File
Edit
Build
Windows
Tools
Marketplace
Layouts
Help
Developer
Sequence1*
GameMaker
Studio 2
sPickupStar: Position (Embedded Curve)
Curves
x
y
Assets
Asset Browser
Search...
Quick Access
Recent
Favourites
Room Order
Saved Searches
Tags
Game Options
Fonts
Objects
Rooms
Scripts
Sprites
Background
sBGGrad_grass
sBGGrad_sand
sBGHills_grass
sBGHills_sand
sBGHills1_grass
sBGHills1_sand
sCloud_grass
sCloud_sand
sCloud1_grass
sCloud1_sand
sMoon
spr_bg
sStar_grass
sStar_sand
Enemy
sGhost
sGhost_left
sGhostShadow
Level
Pickups
sHeart
sPickupStar
Player
Tile Sets
Sequence1
Sequence1
Track Panel
sHumb
sPickupStar
Position
sWalk
spr_bg


CREATION
ENGINE


WARCRAFT
FROZEN THRONE


unity

Which to Choose?

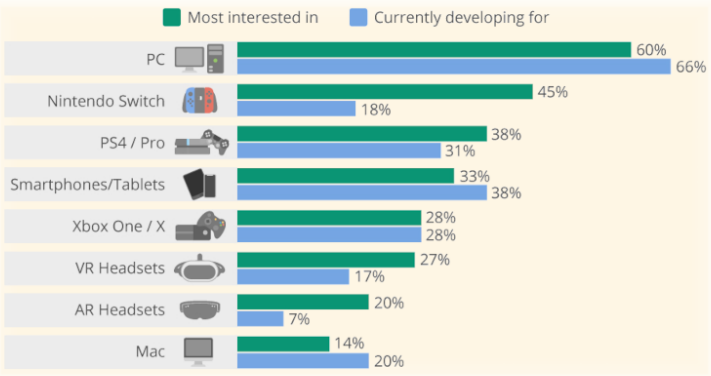
- AAA & Realistic → **Unreal Engine**
- Agile & Indie → **Unity Engine**

- Non-Programmer → **Game Maker**
- Open & Free → **Godot Engine**
- Learn Engine-ering → **Roll Your Own**

- Learn GameDev → **Unity Engine**

Additional Resources

- [Thesis] James Lear: The Video Game Asset Pipeline
- [Online] Robert Nystrom: Game Loop
- [Online] Unity: Order of Execution for event functions



Source: Statista – The Most Important Gaming Platforms


MENTAT
OPTIONS
Credits
Thanks For
Your Attention!
TANK
DMG
ATTACK
MOVE
RETREAT
GUARD
Dune 2 : The Building of a Dynasty

References I

[1] Jason Gregory. ***Game Engine Architecture, Second Edition.*** 3rd. USA: A. K. Peters, Ltd., CRC Press, 2018. ISBN: 1351974288.

[2] R. Nystrom. ***Game Programming Patterns.*** UK: Genever Benning, 2014. ISBN: 0990582906.