



LECTURE 1

# OVERVIEW OF GAME ENGINE

Modern Game Engine – Theory and Practice

WANG XI

GAMES 104

2022



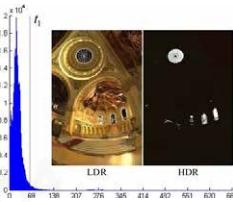
NOW



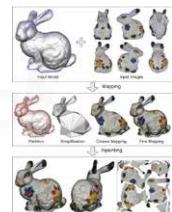
2011



2006



2004



2001



1997

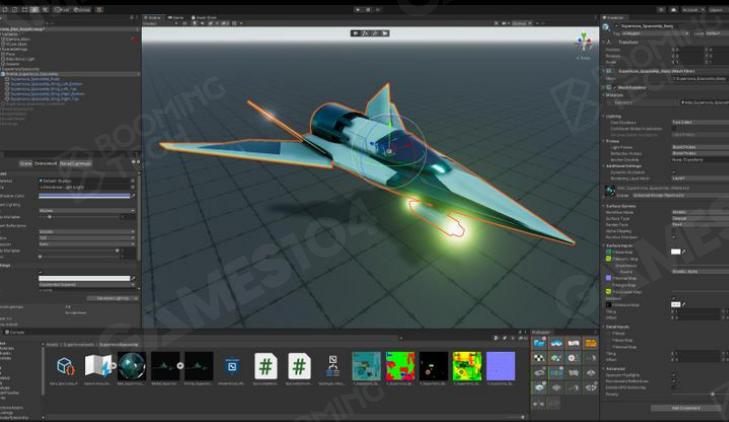
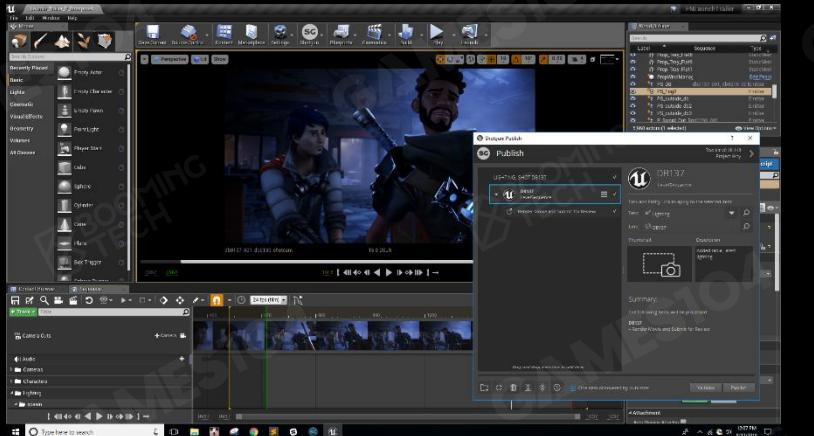
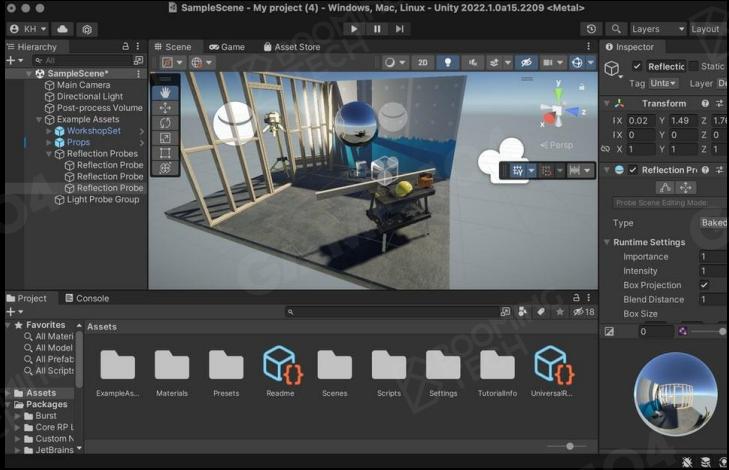
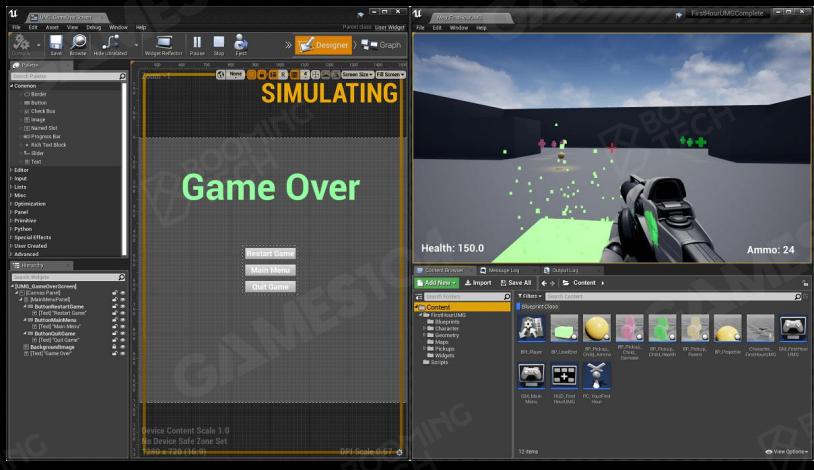
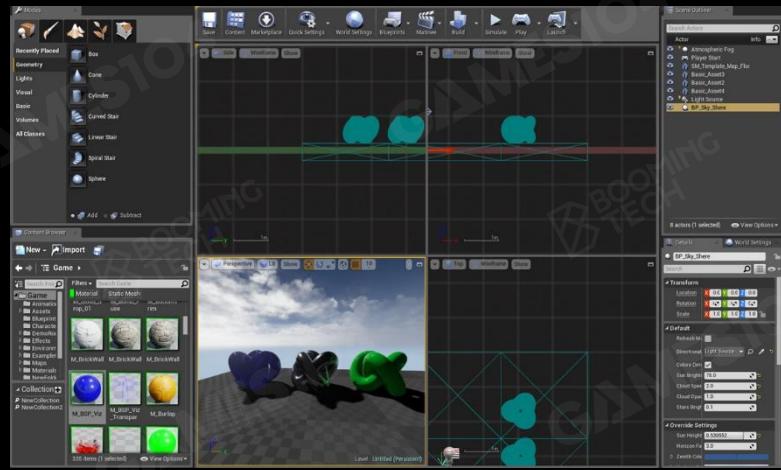


**Fun facts:** video games, swimming, trouble maker, etc.





**Game : Miracle of Modern Computer Technology**



# Game Engine : The Diamond on the Crown

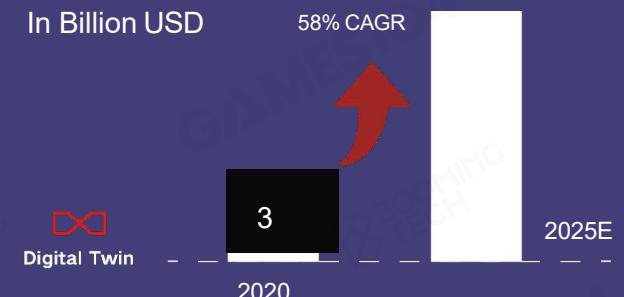
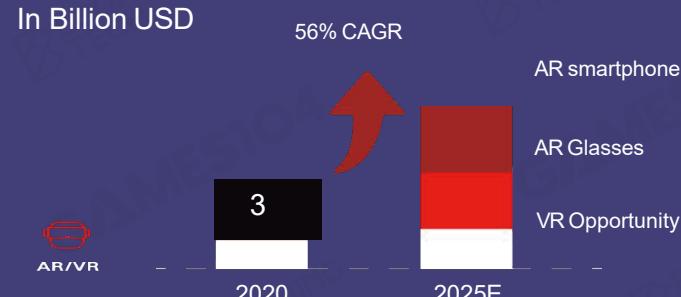
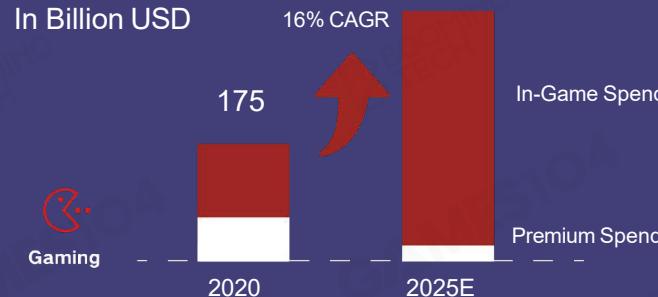


## Topic 1

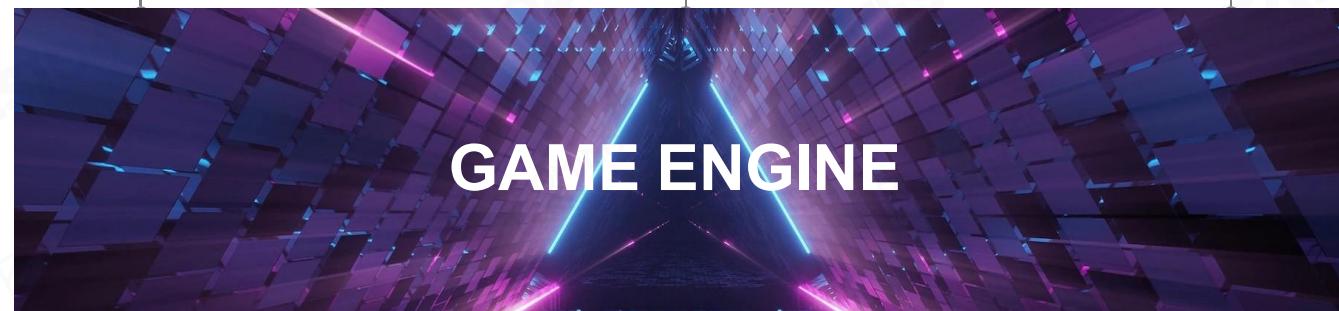
# WHY WE NEED TO LEARN



## Game Engine is the Foundation of Virtual World



Data source: Newzoo, Markets&Markets, Microsoft; AR/VR market size includes AR/VR hardware devices and AR/VR games

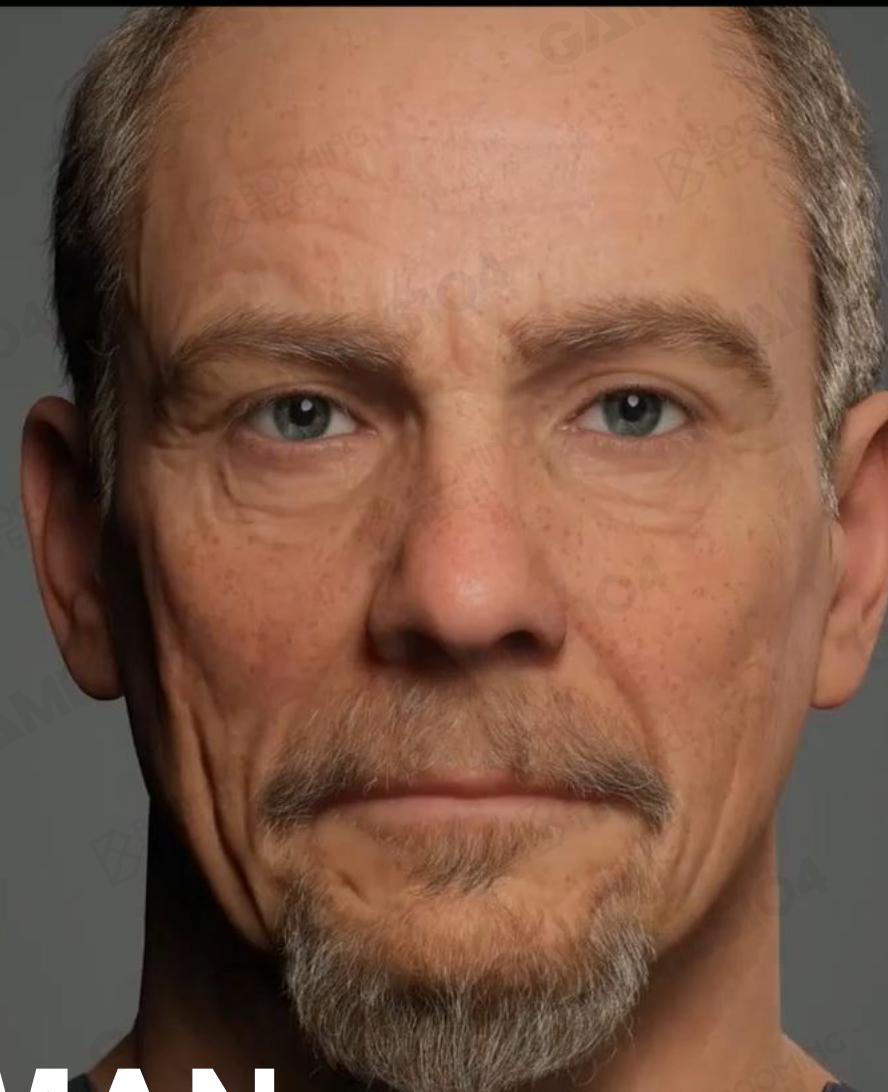




Teeth  
HAIR  
Head  
Beard  
Mustache

EYES  
Eye Color  
Eyelashes  
Eyebrows

BODY  
Body Shape  
Tops  
Bottoms  
Shoes



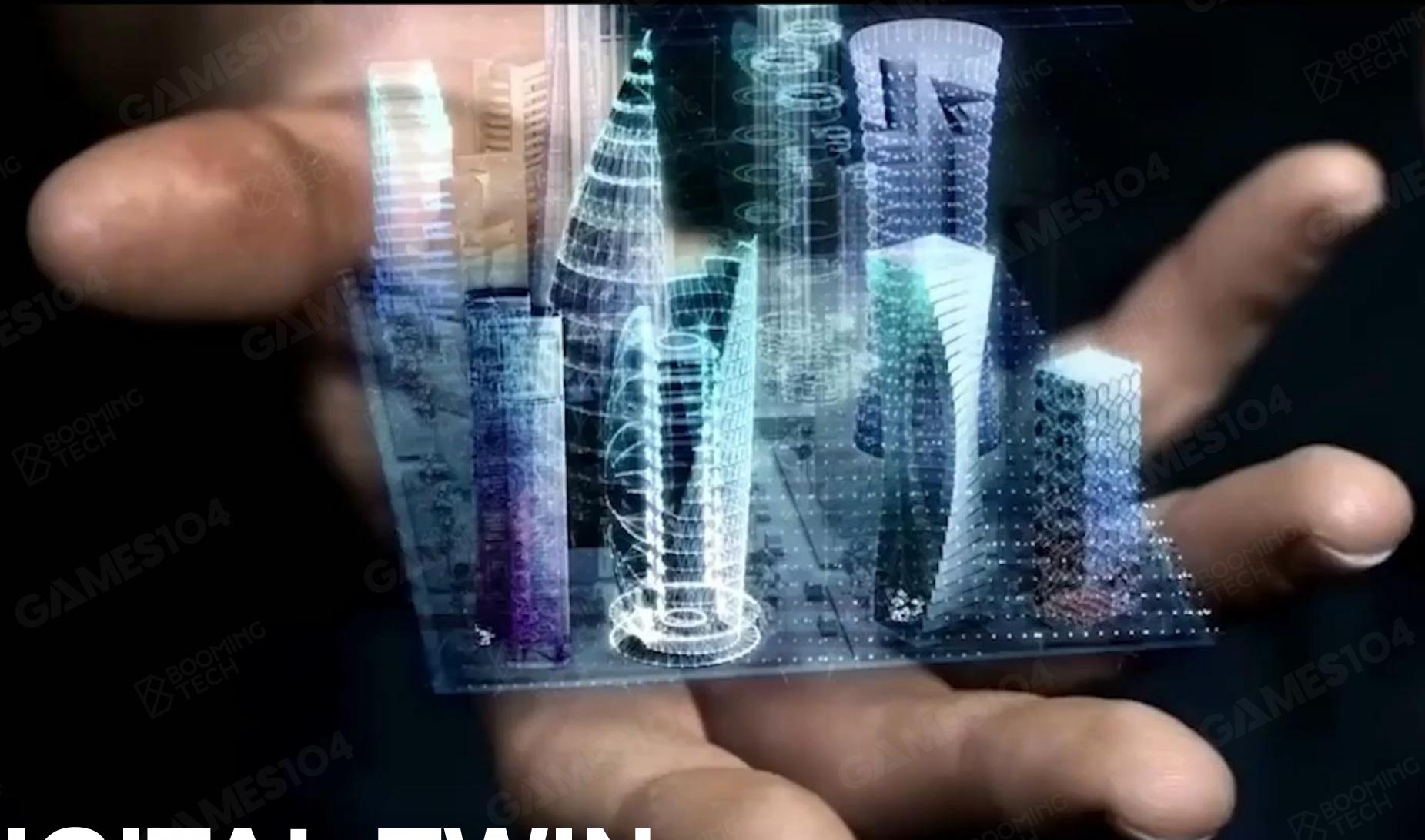
# METAHUMAN



CINEMATIC & ANIMATION



# SIMULATION



# DIGITAL TWIN



Topic 2

# HISTORY OF GAME ENGINE



# Early Age of Video Games



Famicom (FC)





# Father of Game Engine

- **John Carmack**

John conceived and executed a new way of organizing the components of computer games by separating execution of core functionality by the game engine from the creative assets that filled the play space and content of a specific game title.

- **Wolfenstein 3D (1992)**

“Father of 3D shooters” and first demonstration of game engine application



John Carmack



Wolfenstein 3D



# Father of Game Engine

- **Doom**

Along with its predecessor Wolfenstein 3D, Doom defined the FPS genre and inspired numerous similar games, often called the Doom clones. It was the first online distribution game, and it pioneered technologies including 3D graphics, networked multiplayer gaming, and support for custom modifications via packaged WAD files.



Doom

- **Engine License**

1994, ID Software license Doom engine to Raven, which built a successful game ShadowCaster based on it.

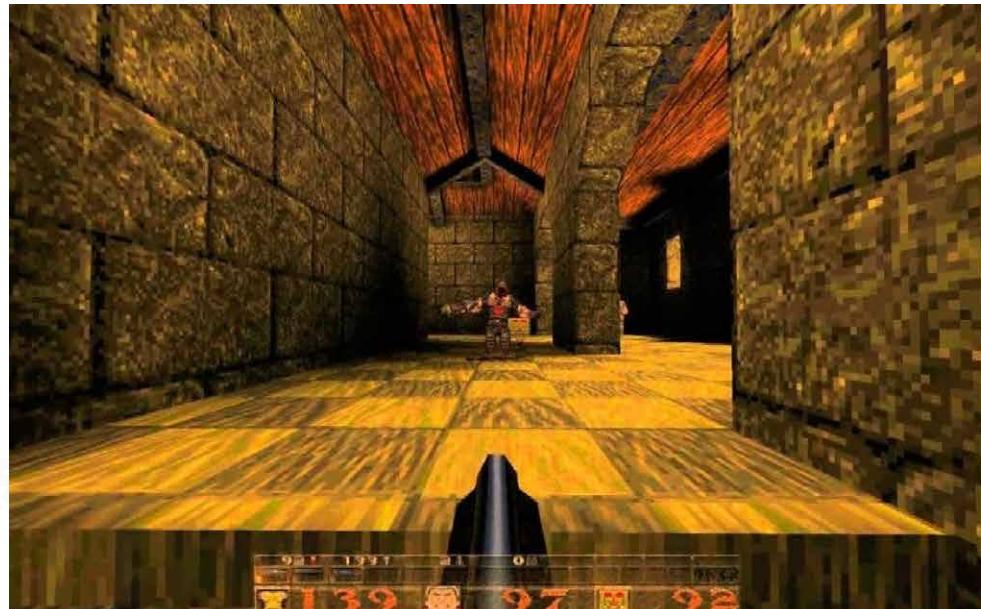


ShadowCaster

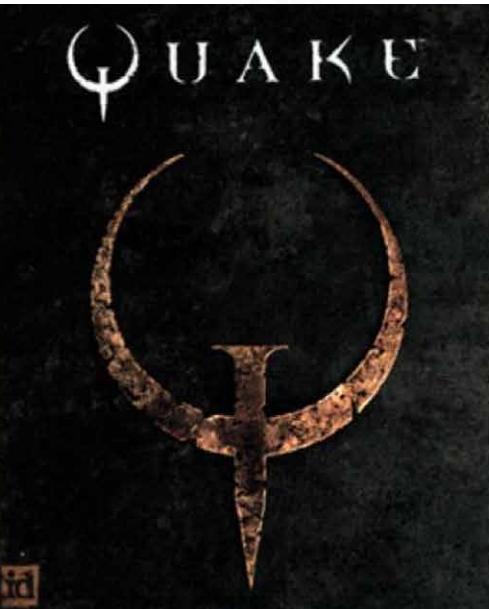


# Early Age of Modern Game Engine

- Quake
- Unlike the Doom Engine, the Quake engine offered full real-time 3D rendering and supported early 3D acceleration through OpenGL.



Quake



Diamond Multimedia的Monster 3D (3dfx Voodoo1 4MB PCI)



A game engine is a software framework primarily designed for the development of video games, which normally includes relevant libraries and support programs.

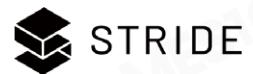




# Family of Game Engines



UNREAL  
ENGINE



STRIDE



GODOT  
Game engine



Commercial Engine

In-house Engine

Free Engine





# Middleware of Game Engine



Physics&Animation

Audio

Rendering

Other



# Q&A



# Topic 3

# WHAT IS GAME ENGINE



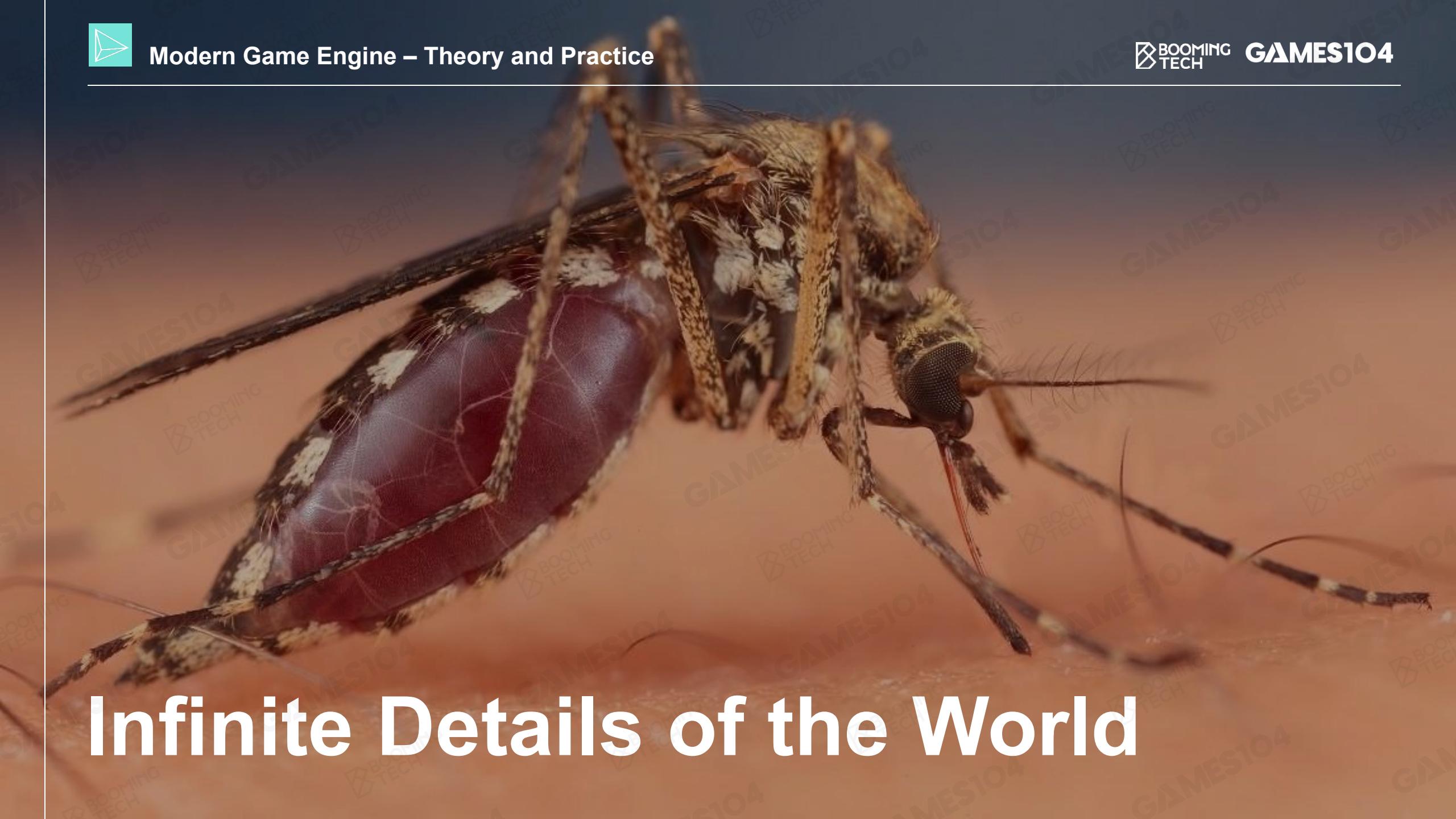
# What's Game Engine?

- A game engine **is a software framework primarily designed for the development of video games, and generally includes relevant libraries and support programs.**<sup>[1]</sup> The "engine" terminology is similar to the term "software engine" used in the software industry.
- Game engine can also refer to **the development software utilizing this framework**, typically offering a suite of tools and features for developing games.<sup>[2][3]</sup>
- Developers can use game engines **to construct games for video game consoles and other types of computers.** The core functionality typically provided by a game engine may include a rendering engine ("renderer") for 2D or 3D graphics, a physics engine or collision detection (and collision response), sound, scripting, animation, artificial intelligence, networking, streaming, memory management, threading, localization support, scene graph, and video support for cinematics. Game engine implementers often economize on the process of game development by reusing/adapting, in large part, the same game engine to produce different games<sup>[4]</sup> or to aid in porting games to multiple platforms.



# Our Definition : What's Game Engine?

- Technology foundation of the Matrix
- Productivity tools of creation
- The Art of complexity



# Infinite Details of the World

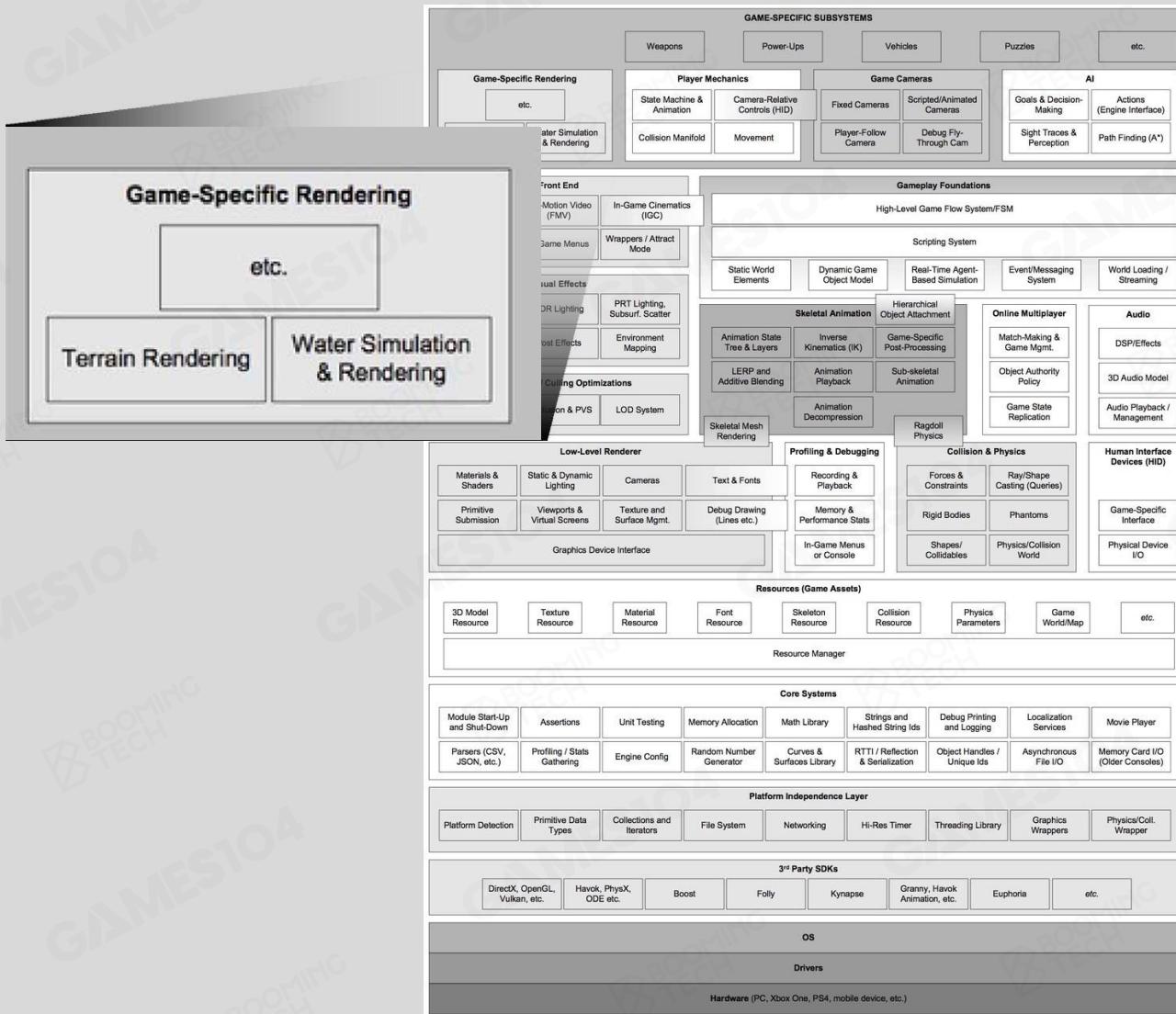


# Complexity of Simulation by 0/1





# Game Engine is Far Beyond Rendering

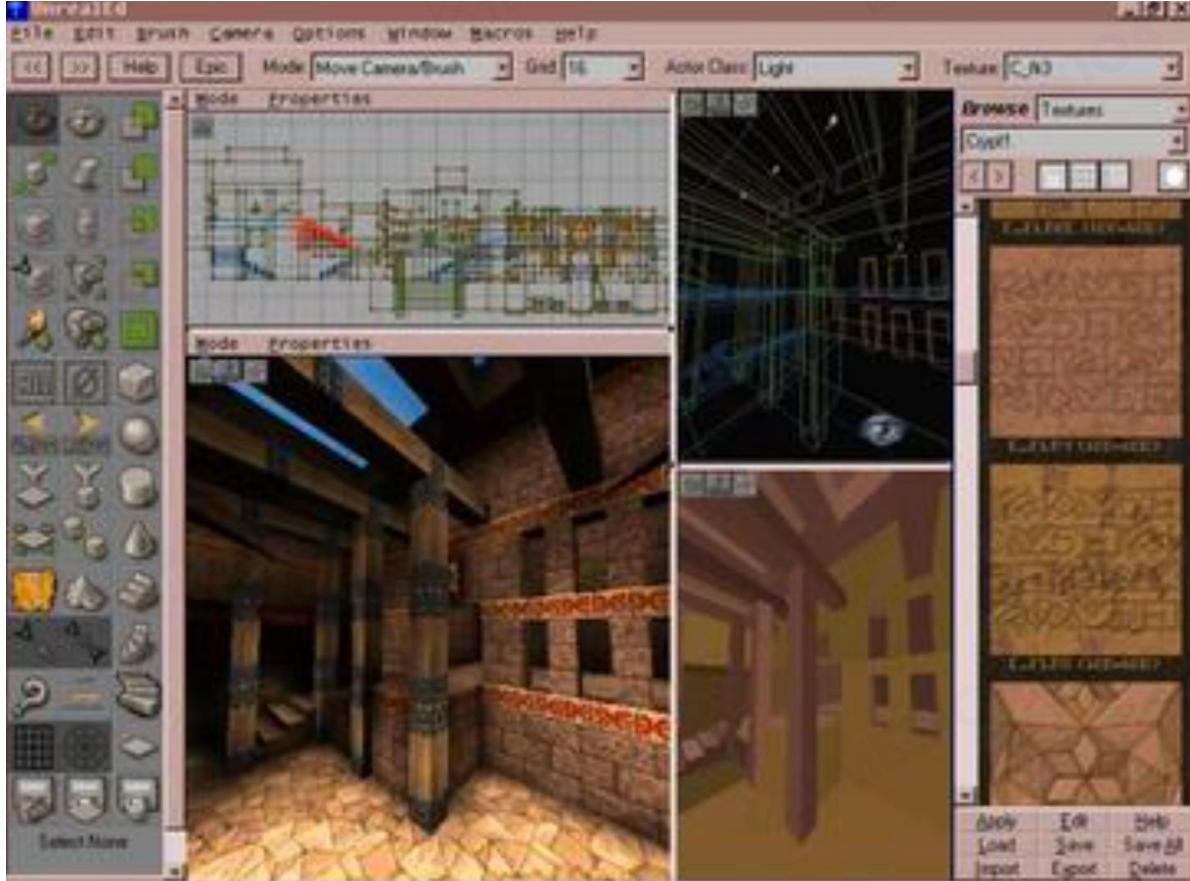




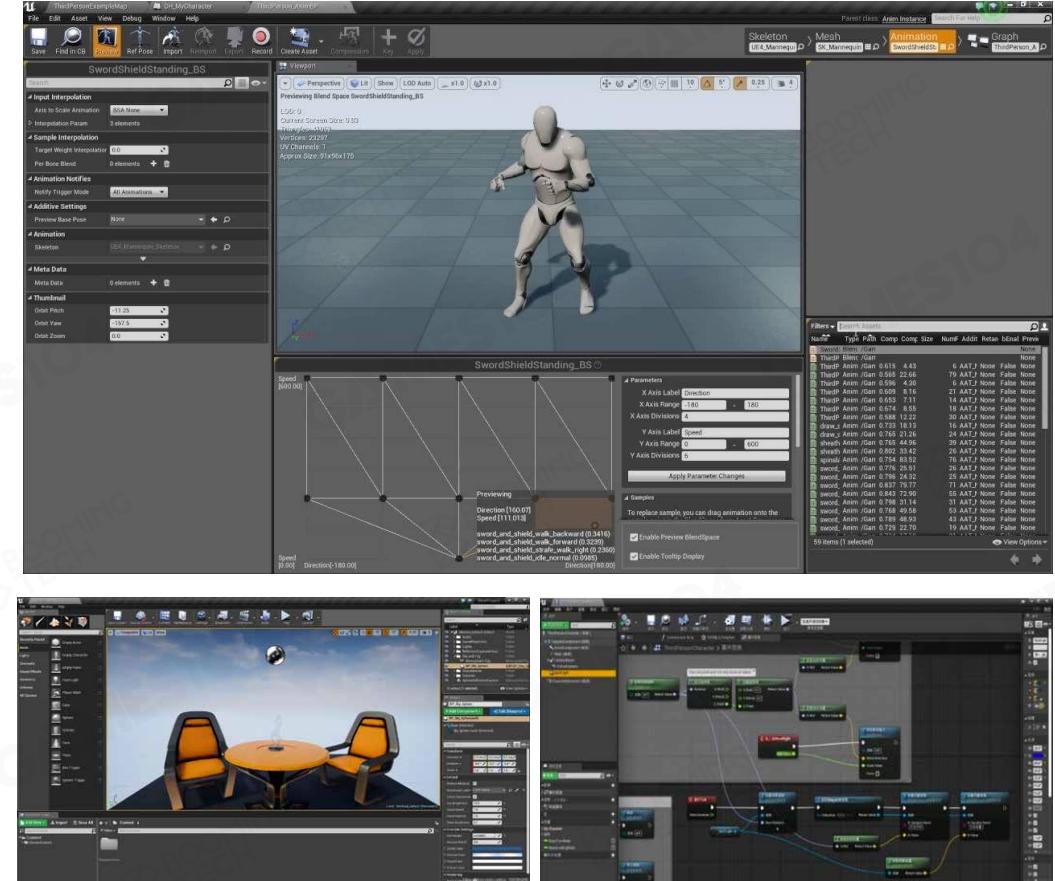
# God with Limited Power in Realtime



# Toolchain for Creators



Engine in 1990s



Engine Nowadays



# Developer Platform

- **For Programmer**

- Expandable API interfaces allow programmers to define various of gameplay without changing the core.

- **For Studio**

- Collaborate hundreds of developers with different work streams smoothly together.



# Update the Engine in the Air



**Yes, We Are...  
The Creator and Operator of This Ugly Monster.  
But Future Will be Even More Difficult...**



# Topic 4

# HOW TO STUDY



## Game Engine Technology Covers All Major Area of Computer Science





# Focus on the Main Road by Building the Framework



# Topic 5

# COURSE CONTENT

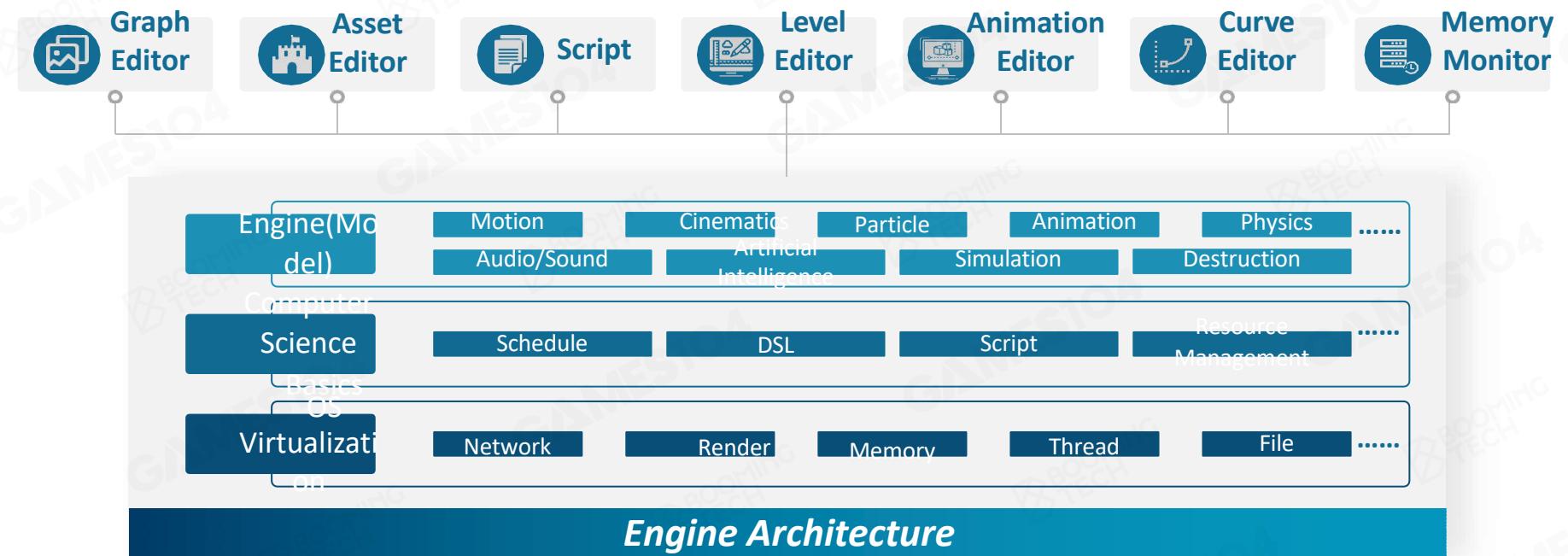


# Basic Elements

- Engine structure and layers
- Data organization and management



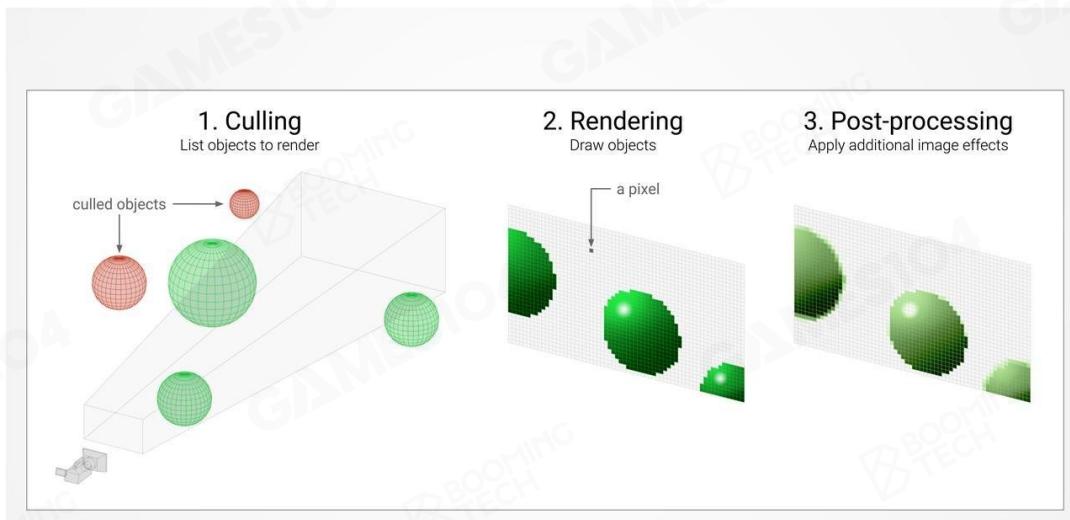
**MVVM**  
MODEL VIEW VIEWMODEL





# Rendering

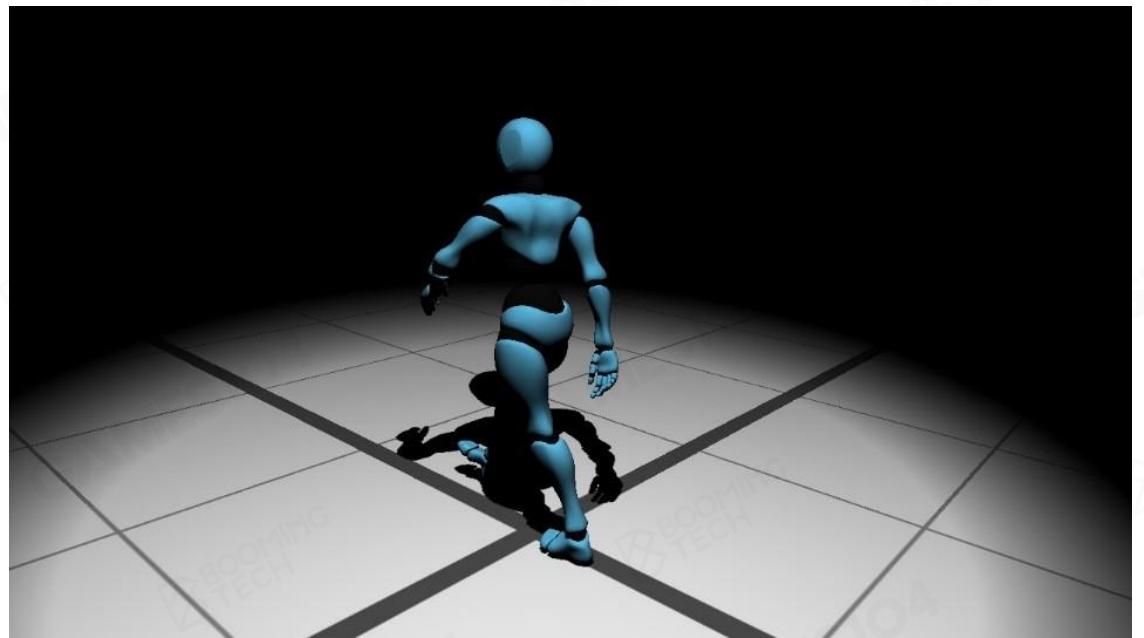
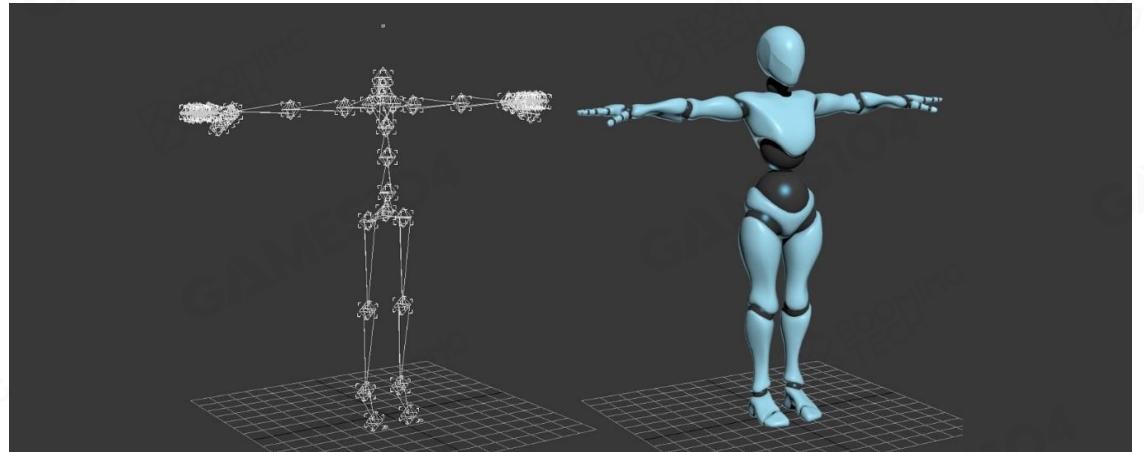
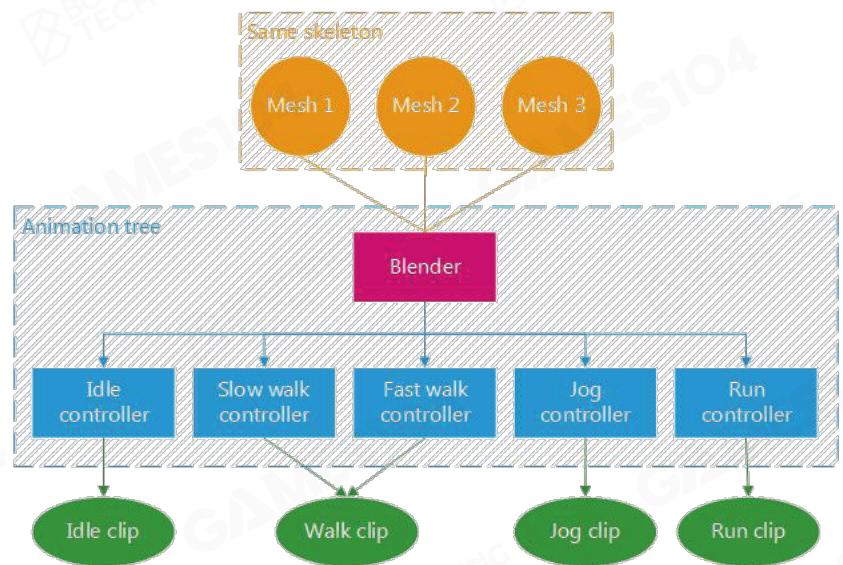
- Model, material, shader, texture
- Light and shadow
- Render pipeline
- Sky, terrain, etc





# Animation

- Basic concepts of animation
- Animation structure and pipeline





# Physics

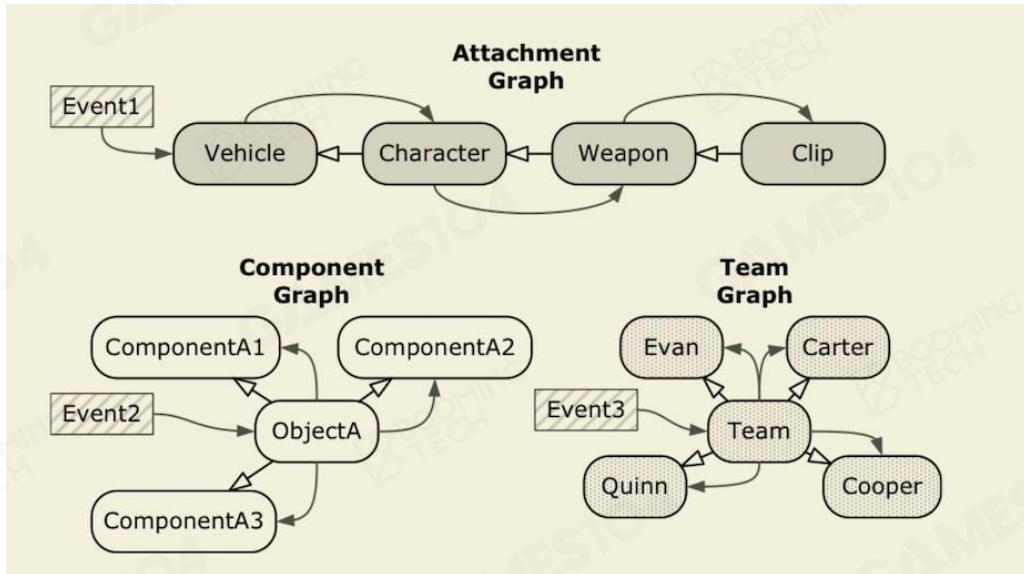
- Basic concepts of Physics System
- Gameplay applications
- Performance optimization





# Gameplay

- Event System
- Scripts System
- Graph Driven

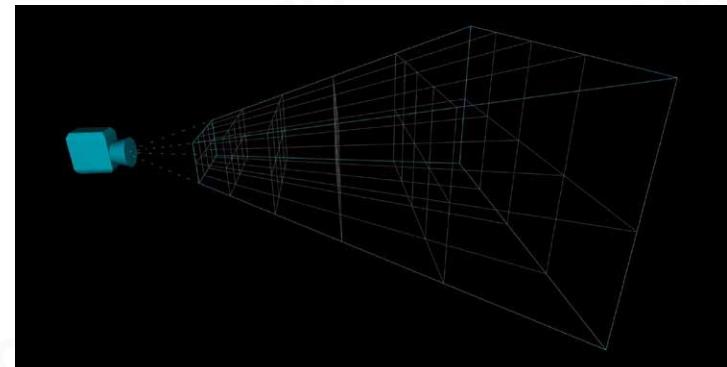


```
1  ---+LUA Sample Script Version 1.0.0.0+---  
2  ---+Author: ViperGTS96+---  
3  -----+The simplest design is always the best design.+---  
4  -----  
5  myLUAScript = {};  
6  myLUAScript.myLUAScript = g_currentModDirectory; -- Gets the Mod's base directory which can only be obtained at registration. Here it is transferred to this main script.  
7  --myLUAScript.myLUAScript = now equals the base Directory of the mod for importing files.  
8  -----  
9  myLUAScript = {};  
10 myLUAScript.myLUAScript = g_currentModDirectory; -- Gets the Mod's base directory which can only be obtained at registration. Here it is transferred to this main script.  
11 --myLUAScript.myLUAScript = now equals the base Directory of the mod for importing files.  
12 -----  
13 if not myLUAScript then return true; end;  
14 -----  
15 function myLUAScript:processUpdateEvent(specifications)  
16 end;  
17 -----  
18 function myLUAScript:load(savegame)  
19 end;  
20 myLUAScript.renderHelp = false;--Boolean used for toggling the Help Menu controls.  
21 -----  
22 end;  
23 -----  
24 function myLUAScript:help()  
25 end;  
26 -----  
27 function myLUAScript:mouseEvent(posX, posY, isDown, isUp, button)  
28 end;  
29 -----  
30 function myLUAScript:keyboard(unicode, sym, modifier, isDown)  
31 end;  
32 -----  
33 function myLUAScript:update(dt)  
34 end;  
35 -----  
36 function myLUAScript:quit(dt)  
37 end;  
38 if g_currentMission.controlledVehicle == self then --Is in a vehicle  
39 if not self.isBroken then -- Vehicle is not broken  
40 if InputBinding.nextEvent(InputBinding.yourInputNameKey1) then  
41 -- Code for Inputbinding 1  
42 end;  
43 if InputBinding.nextEvent(InputBinding.yourInputNameKey2) then  
44 -- Code for Inputbinding 2  
45 end;  
46 if InputBinding.nextEvent(InputBinding.yourInputNameKey3) then  
47 -- Code for Inputbinding 3  
48 end;
```



# Misc. Systems

- Effects
- Navigation
- Camera
- ...





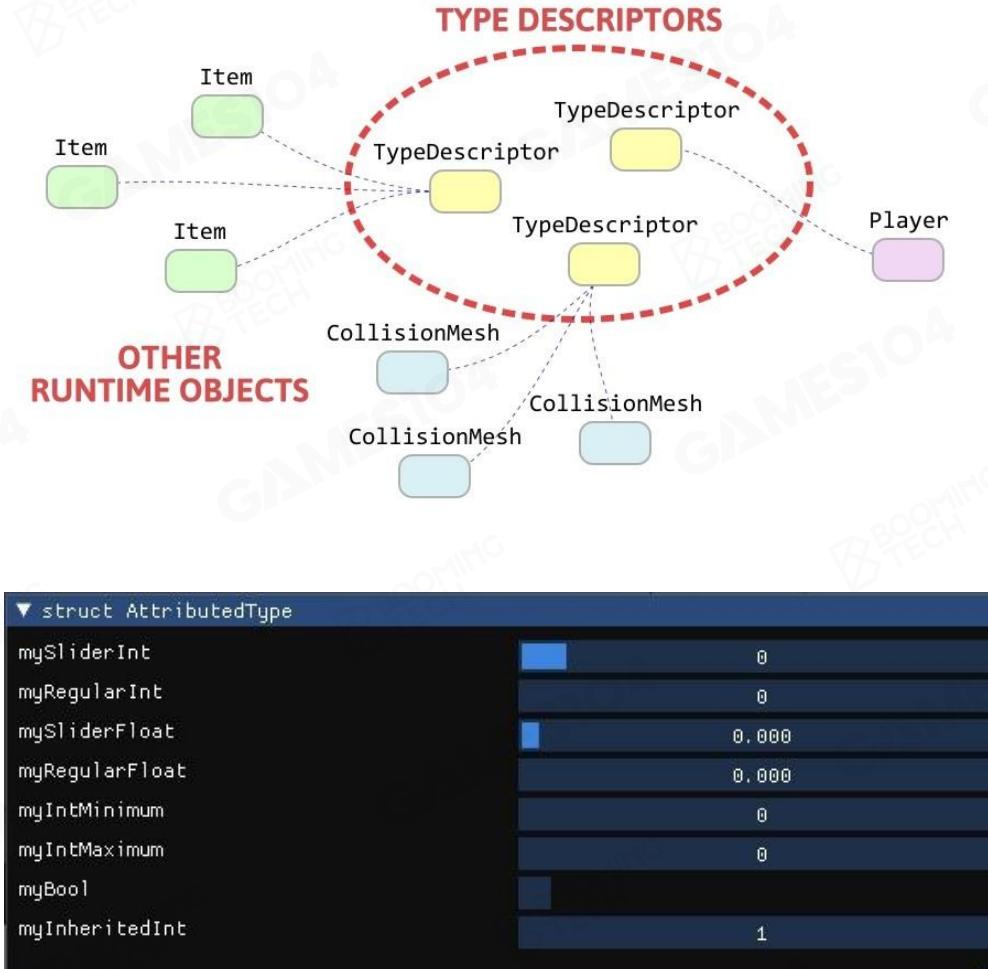
# Toolchain

- **C++ Reflection**

Expose variables and functions used in the editor. That is, the game creation tool will use a form of reflection (or similar) on the code provided by the developers, which then allows it to expose parts of it in editors for designers.

- **Data Schema**

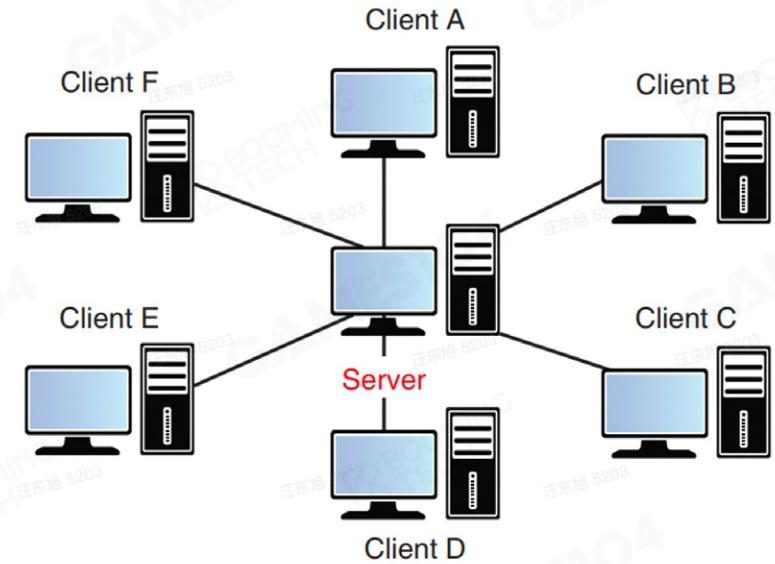
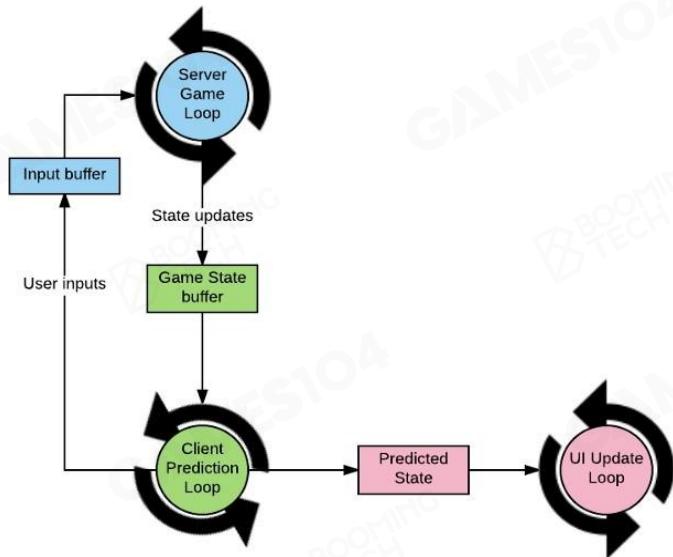
A data schema is the formal description of the structures which a system is working with.





# Online Gaming

- Lockstep synchronization
- State synchronization
- Consistency





# Advanced Technology

## · Motion Matching

Motion Matching is a simple yet powerful way of animating characters in games. Compared to other methods, it doesn't require very much manual work once you have a basic set-up: there is no need to structure clips in graphs, to carefully cut or synchronize them, or to explicitly create new transitions between status.

## · Procedural Content Generation (PCG)

PCG is a method of creating data algorithmically as opposed to manually, typically through a combination of human-generated assets and algorithms coupled with computer-generated randomness and processing power.





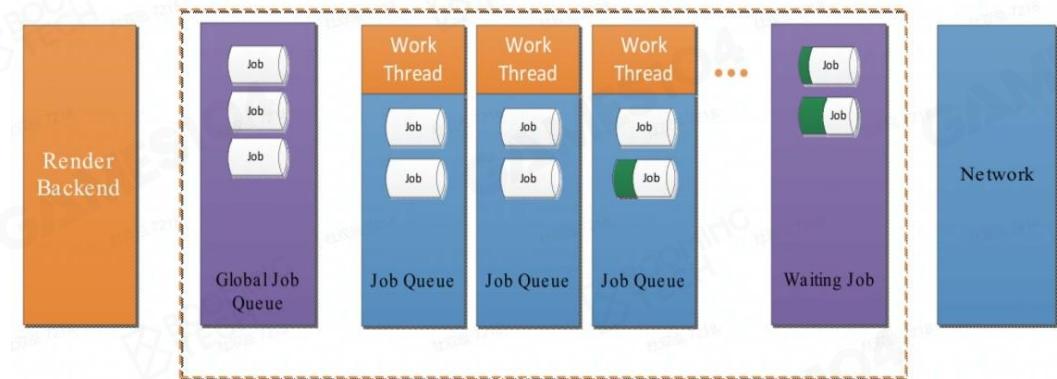
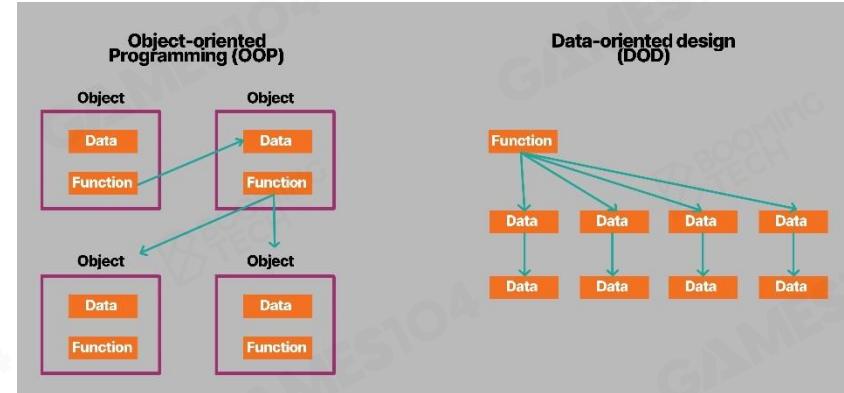
# Advanced Technology

## · Data-Oriented Programming (DOP)

DOP is an exciting new paradigm that eliminates the usual complexity caused by combining data and code into objects and classes. In DOP, you maintain application data in persistent generic data structures separated from the program's code. You use general-purpose functions to manipulate the data without mutating it. This approach rids your applications of state-related bugs and makes your code much easier to understand and maintain.

## · Job System

A job system manages multithreaded code by creating jobs instead of threads.





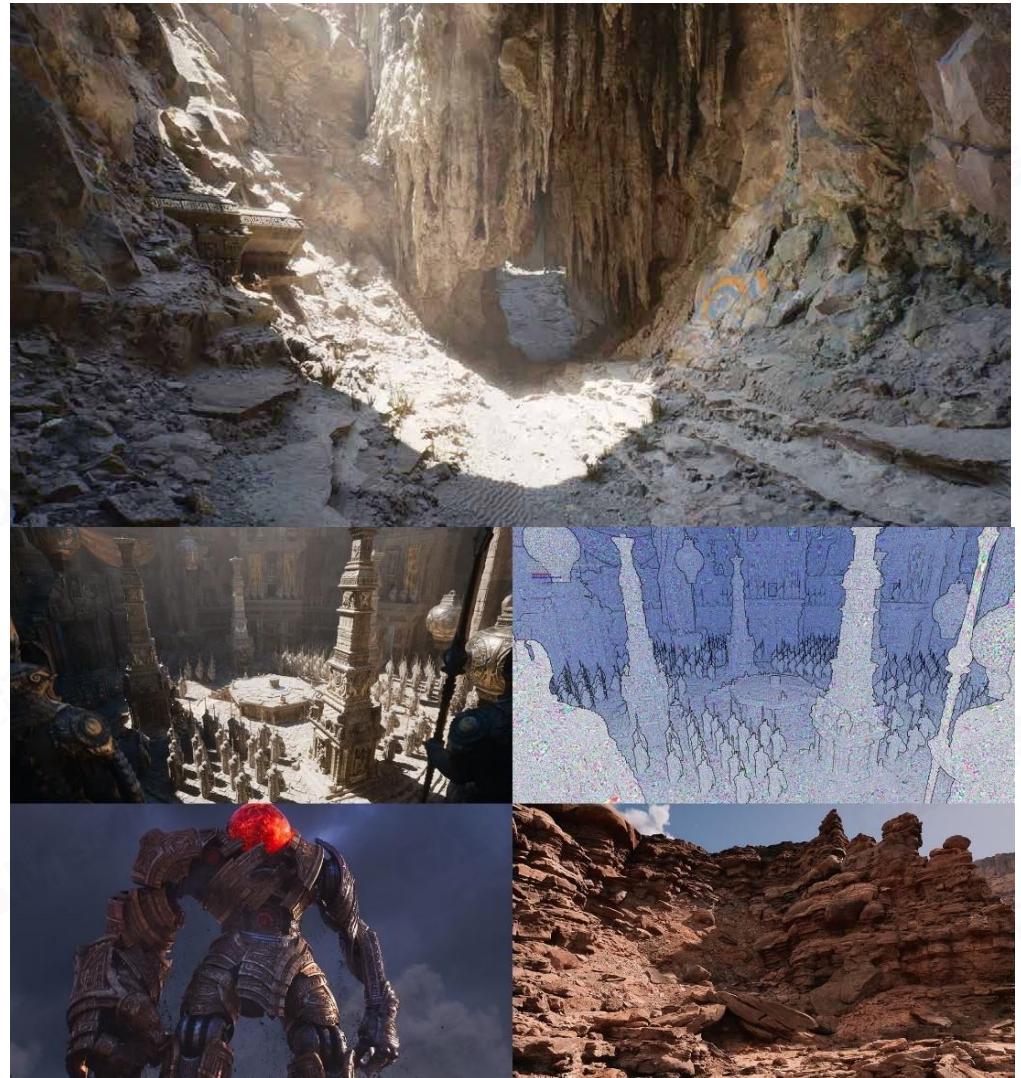
# Advanced Technology

## · Lumen

Unreal Engine 5's new fully dynamic global illumination and reflections system that is designed for next-generation consoles. It renders diffuse interreflection with infinite bounces and indirect specular reflections in large, detailed environments at scales ranging from millimeters to kilometers.

## · Nanite

Unreal Engine 5's new virtualized geometry system which uses a new internal mesh format and rendering technology to render pixel scale detail and high object counts.





# Topic 6

# COURSE LOGISTICS



# General Information

## · Course Website

- <https://games104.boomingtech.com>
- course slides

## · Q & A

- Sign up on our BBS for discussion  
[games-cn.org/forums/forum/games104-forum](http://games-cn.org/forums/forum/games104-forum)

## · Course Wechat



The screenshot shows the GAMES104 website homepage. At the top, there's a navigation bar with links for 'GAMES104介绍' (Introduction), '课程内容' (Course Content), 'GAMES官网' (GAMES Official Site), and 'BOOMING TECH'. Below the navigation are three blue buttons: '课程资料' (Course Materials), 'BBS论坛' (BBS Forum), and '作业提交' (Assignment Submission). The main content area features a large yellow sphere and an orange rectangular prism. A central banner with the text '从0到1搭建起一个完整的迷你游戏引擎' (Build a complete mini game engine from scratch) is displayed. Below the banner are two buttons: '了解更多关于GAMES' (Learn more about GAMES) and '关于我们' (About us). Further down, there's a screenshot of a 3D game engine interface titled 'MINI 引擎' (MINI Engine), showing a 3D scene with a character and various engine settings. A descriptive paragraph at the bottom right explains the course's goal: '本课程将介绍现代游戏引擎所涉及的系统架构, 技术点, 引擎系统相关的知识。通过该课程, 你能够对游戏引擎建立起一个全面且完整的了解。如果你动手能力足够强, 你将能够跟随课程, 从0到1搭建起一个完整的迷你游戏引擎。如本课程适合相关专业领域的学生、研究者, 以及所有对游戏引擎设计和开发感兴趣的人。'



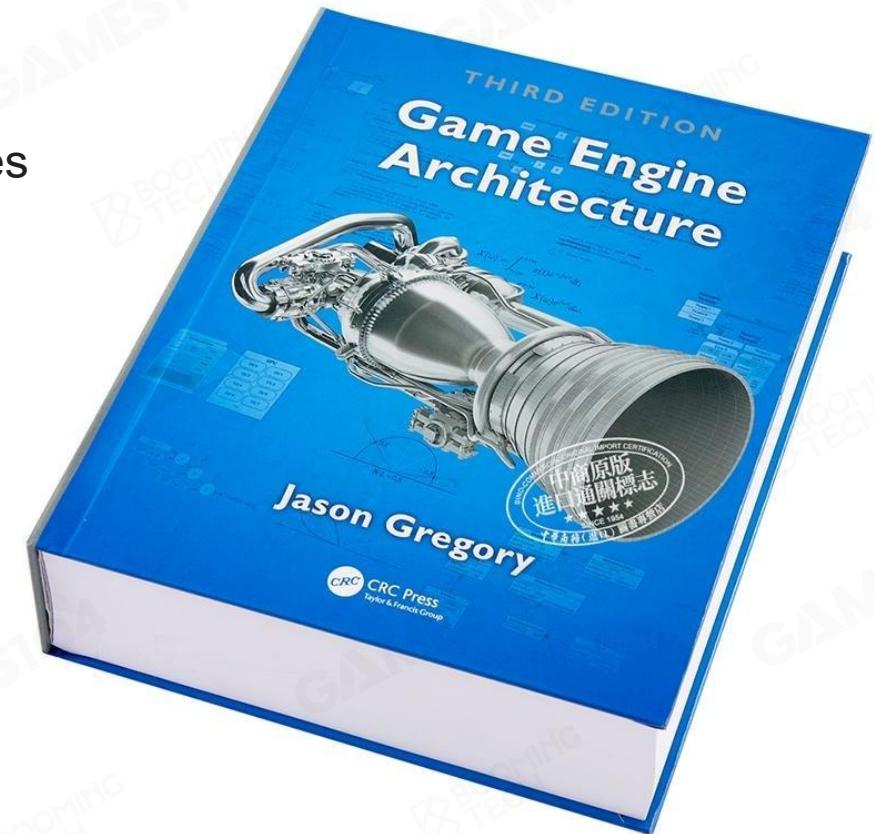
# References

- **No Textbooks Required**

- Reading materials (if any) will available online before lectures
- Lecture slides will be available after each class

- **Most Recommended Reference**

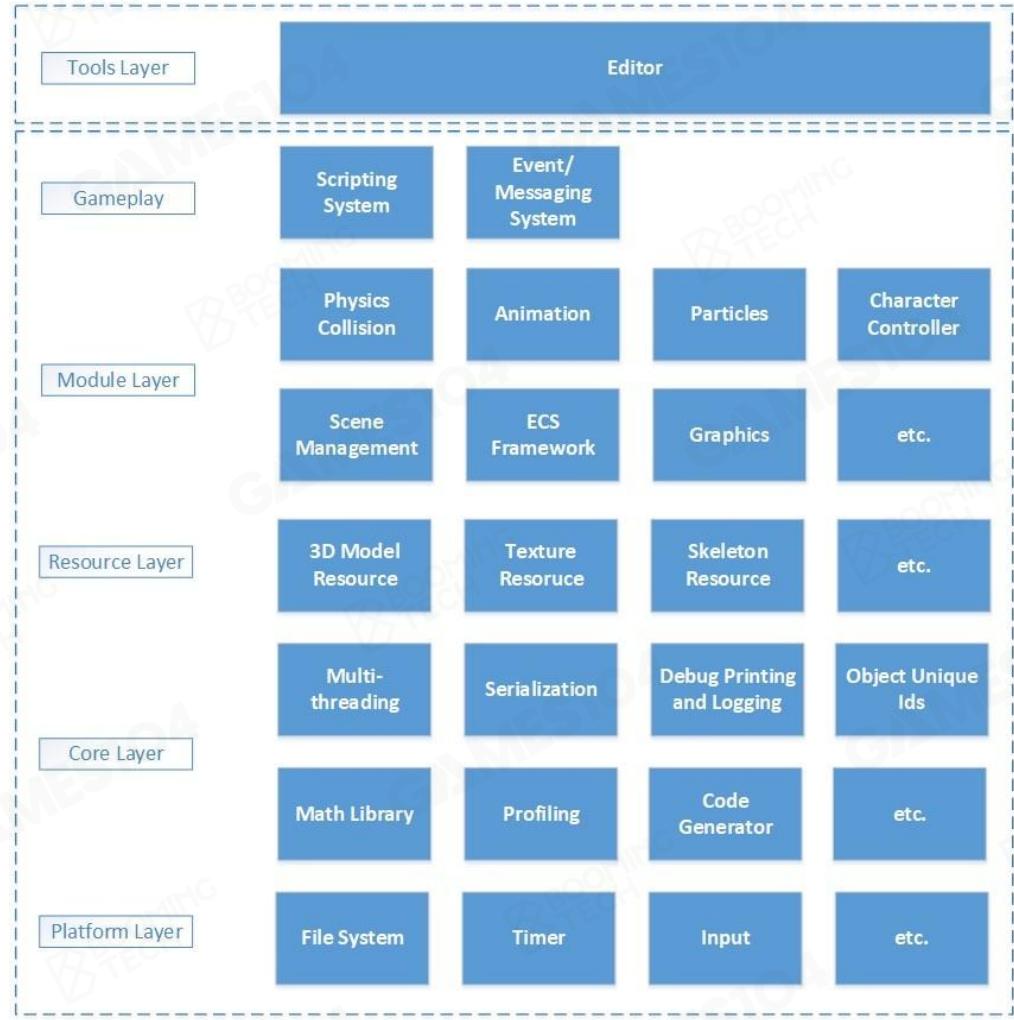
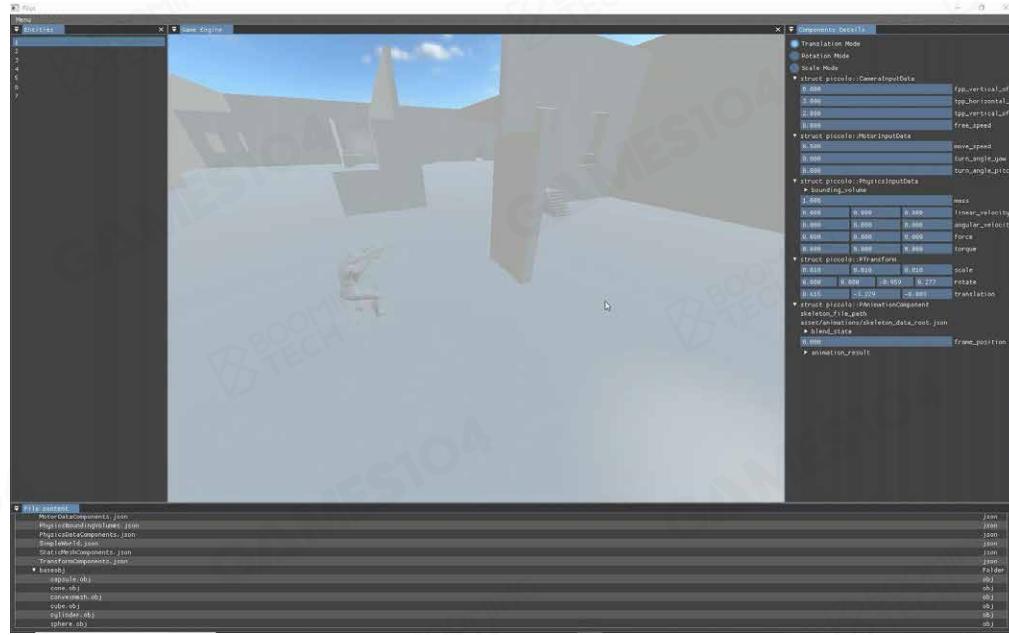
- Jason Gregroy, “Game Engine Architecture”, 3rd or later editions





# Mini Engine

- Mini runtime framework
- Mini editor
- Building basic knowledge system of game engine





# Assignments

## · Assignments

- Mostly programming tasks with code skeletons and virtual machine image provided
- Monthly (usually no more than 200 lines of code per week)
- Language C++

## · Submission

- Every four weeks for homework
- Feedback will be given in a week



# Assignments

- **Assignment Submission Website**

[www.smartchair.org/GAMES104](http://www.smartchair.org/GAMES104)

- **No Exams**

- **Course Project / Final Project**

- Midway of this course
- References will be provided, but you decide the topic
- Best work will be posted online or demonstrated in other ways



# Course Contributor

- Claire (清华大学)
- Bear (游戏引擎开发者)
- 彭渊 (《实时渲染 4th》译者)
- 常楠 Alfred (英雄游戏)
- 严昊 Eric (动视Activision)
- 唐声福 (腾讯游戏)
- 陈文博 (中国科技大学GCL实验室)
- 李效良 (网易游戏)
- 施祺 (Gameplay研发&游戏设计师)
- 毛彦凯 (上海交大)
- 孟本源 (清华大学)
- 陈庆 (图形引擎工程师)
- 鲁瀚洋 (东南大学)
- 黄高乐 (独立游戏制作人)
- 罗思源 (西安交大)
- 张嘉瑶 (阿卜杜拉国王科技大学)
- 曾添 (Sumo Digital)
- 曹令鑫 (山东大学)
- 黄琦 papalqi (腾讯)
- 梁讯同 (宾夕法尼亚大学)

## Course Team

- |      |         |         |        |
|------|---------|---------|--------|
| - 大喷 | - 胖丁    | - BOOK  | - 金大壮  |
| - 爵爷 | - 东旭    | - MANDY | - LEON |
| - 德辉 | - 砚书    | - 吴俗    | - 梨叔   |
| - 靓靓 | - SHINE | - 邓导    | - KAI  |
| - 之栋 | - JUDY  | - QIUU  | - C佬   |
|      |         |         | - 阿乐   |

More will be recruited soon after  
this lecture(based on need).



# Enjoy ;) Coding



Thankyou for attending this course  
Please scan here to let me know, how you think of our course



Please note that all videos and images and other media  
are cited from the Internet for demonstration only.