



SIGGRAPH 2026
Los Angeles 19–23 JUL

The Premier Conference & Exhibition on
Computer Graphics & Interactive Techniques

Teaching Generative Graphics for Games

Arnav Jhala

NC STATE
UNIVERSITY

Duke

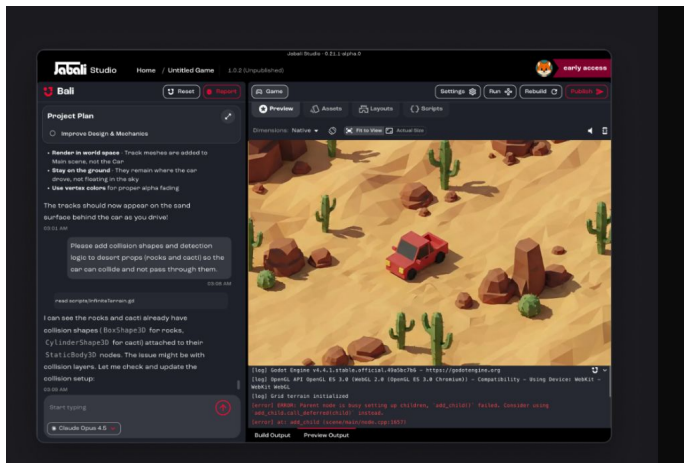


2026



Inspiration for this presentation

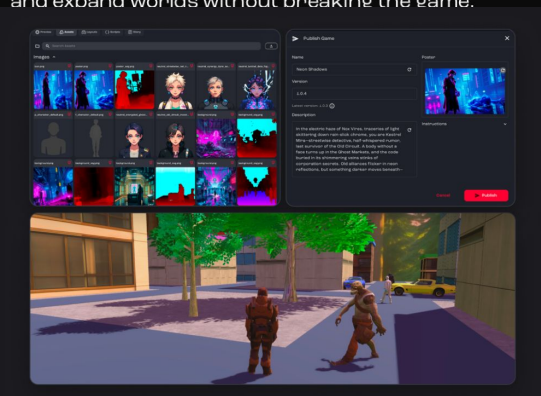
SIGGRAPH



Stop Generating, Start Producing

Most AI tools stop at generation. Jabali is built for producing games.

Start with a playable prototype, then evolve it into a real game. Change mechanics, generate assets, refine systems, and expand worlds without breaking the game.



Iterate Until It's Right

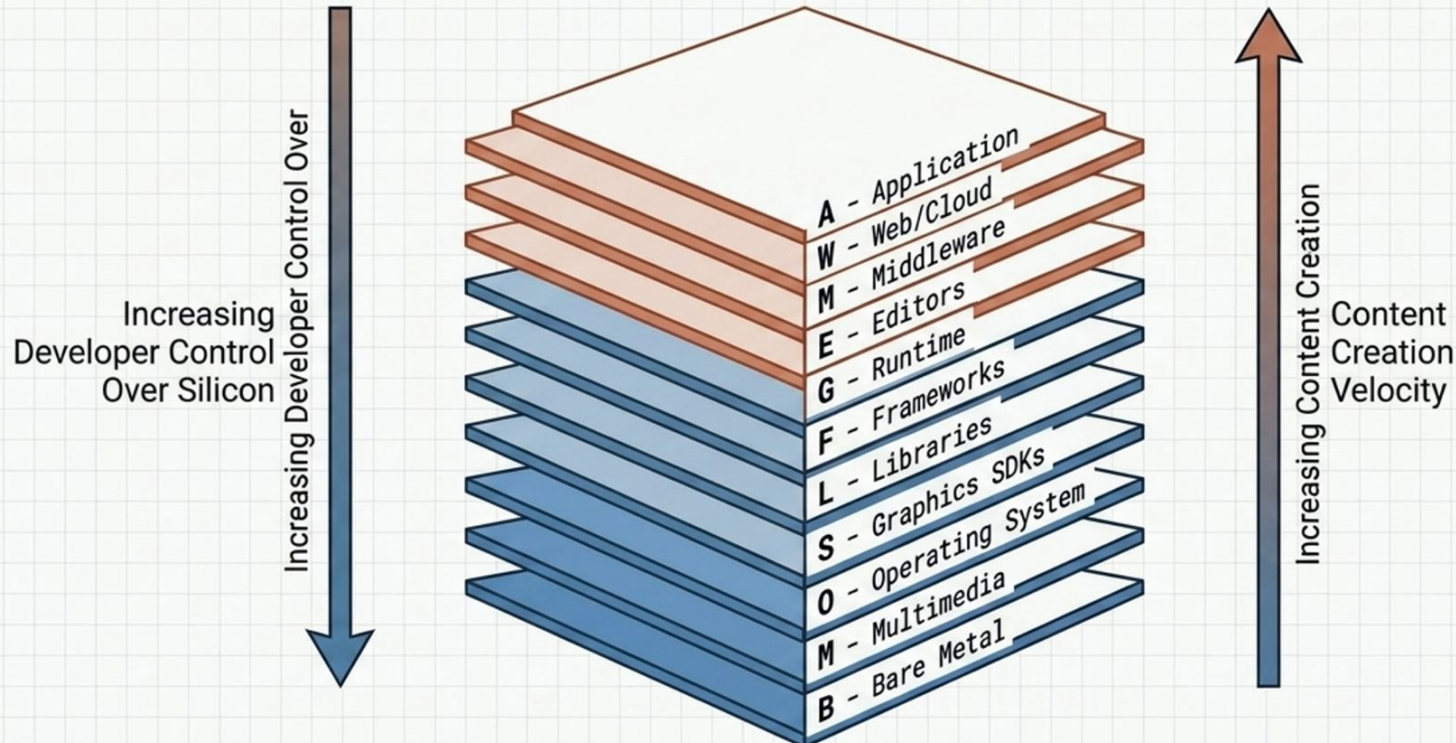
Great games are discovered through iteration. Jabali lets you refine your game through structured changes and controlled improvements.

Tune mechanics. Adjust progression. Generate new assets. Improve systems. Each change builds on the last so your prototype evolves into something worth shipping.

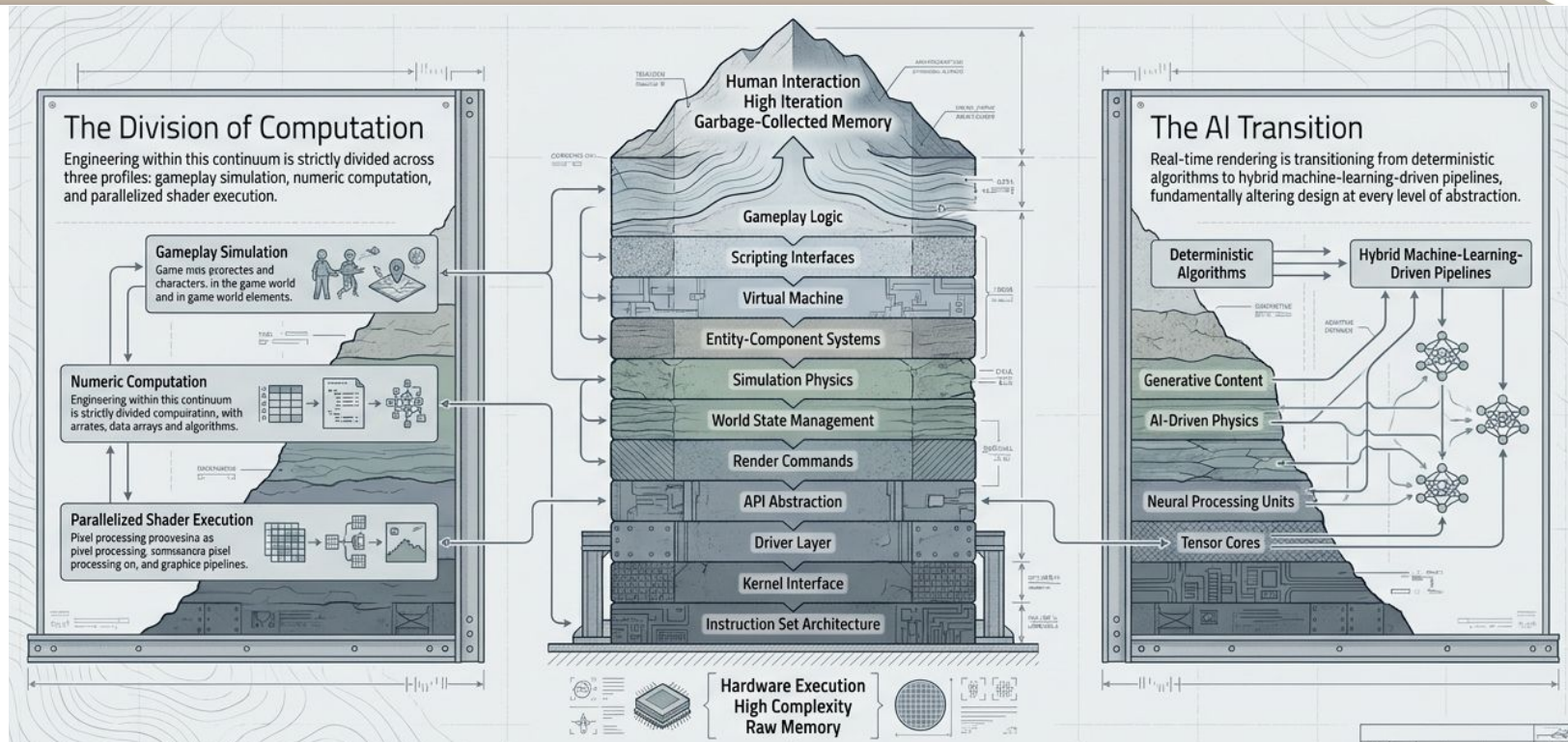
NC STATE
UNIVERSITY

Duke

The Spectrum of Game Engines



Engine architecture perspective



Examples of changes at different tiers



Tier 1: Bare Metal

Mechanical Interface:

Direct silicon execution without operating system mediation. Involves manual manipulation of memory-mapped I/O (MMIO) registers, clock timings, and custom instruction scheduling.

Specific Example:

Custom scanline rendering algorithms and manual register management on retro console hardware.

AI Integration: Silicon-Level Compiler Optimization

Machine learning compilers predict instruction latency, register conflicts, and branch execution delays to generate highly optimized assembly binaries, minimizing physical register spill-over.

Tier 2: Multimedia Abstraction

Mechanical Interface:

Lightweight, platform-agnostic layers that abstract window creation, event handling, input polling, and graphics context initialization, while leaving the rendering pipeline custom-tailored.

Specific Example:

The Simple DirectMedia Layer (SDL), GLFW, or Allegro.

AI Integration: Intelligent Input and Display Reconstruction

Predictive models analyze physical input device queues to filter sensor noise, predict future joystick coordinate trajectories, and pre-cache input states to prevent frame drops.

Low-level Graphics Tiers

SIGGRAPH



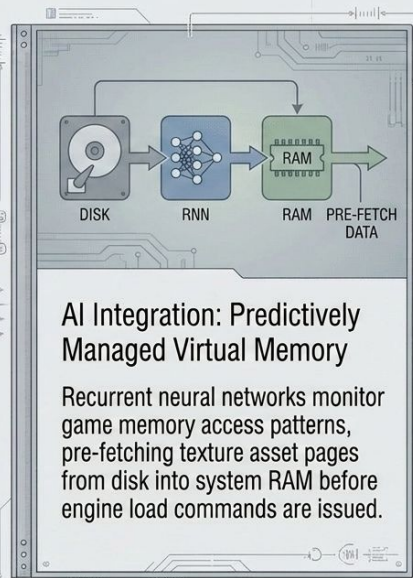
Tier 3: Operating System Kernel

Mechanical Interface:

Management of physical resource allocation, virtual-to-physical memory mapping, thread scheduling, and device driver isolation. Prevents frame-time fluctuations caused by page faults or thread preemption.

Specific Example:

Kernel-mode drivers communicating with user-space graphics drivers.



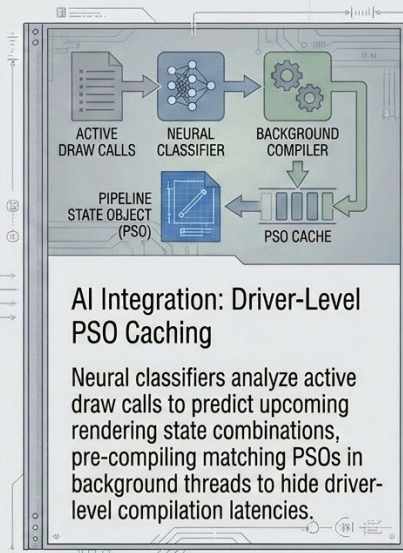
Tier 4: Software Graphics SDKs

Mechanical Interface:

Explicit APIs that map directly to physical GPU structures, requiring manual construction of Pipeline State Objects (PSOs), explicit memory barriers, and semaphore chains.

Specific Example:

Vulkan, DirectX 12, and Metal implementations.



Specialized Libraries and Frameworks



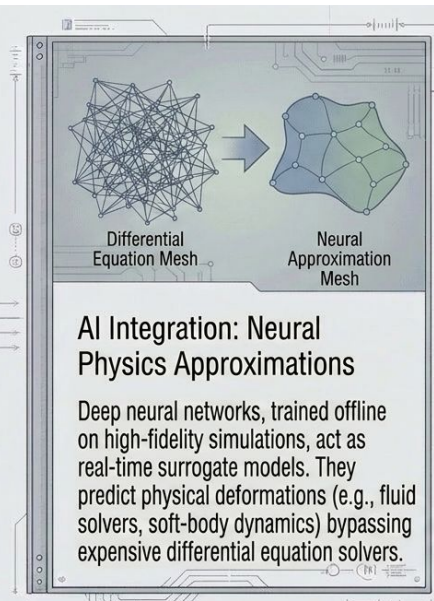
Tier 5: Libraries

Mechanical Interface:

Modular, non-structural utilities designed to solve highly specific mathematical or physical simulation problems via cache-friendly, SIMD-accelerated procedural execution.

Specific Example:

Physics packages like Box2D or Bullet; linear algebra packages like GLM.



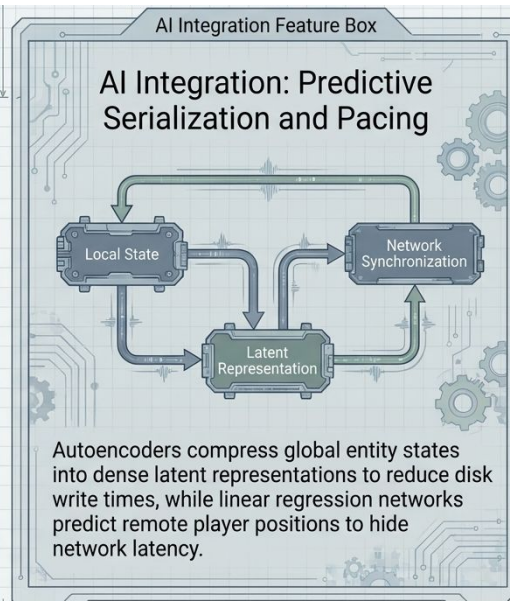
Tier 6: Frameworks

Mechanical Interface:

Code-driven structures that establish the main game loop, basic rendering wrappers, and scene transition logic without visual editors, relying on component-based composition models.

Specific Example:

MonoGame, Raylib, or Phaser frameworks.



Engine Runtimes and Editor Environments



Engine Runtime

Mechanical Interface:

A highly integrated, monolithic software suite managing the global scene graph, culling, material sorting, and multi-threaded frame execution across specialized worker threads.

Specific Example:

Core rendering pipelines of proprietary AAA titles optimized for specific hardware platforms.

AI Integration Feature Box

AI Integration: Neural Rendering and Reconstruction

The runtime computes low-resolution target frames with motion vectors, passing them to neural networks (DLSS/XeSS) on GPU tensor cores to reconstruct high-resolution outputs. This reduces shader instruction compute by up to 70%.

$$C_{\text{shade}} = F_{\text{rate}} \times P_{\text{frame}} \times N_{\text{passes}} \times I_{\text{shade}} \times W_{\text{SIMD}}$$

Reduction in I_{shade}

Tier 8: Editable Environments

Mechanical Interface:

Visual Integrated Development Environments (IDEs) sitting atop the runtime, translating viewport manipulations and node-based graphs into structural runtime data schemas.

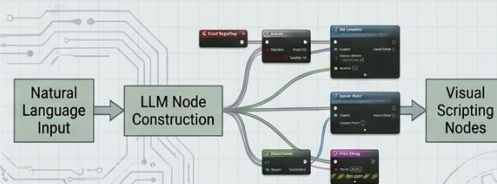
Specific Example:

Unreal Engine, Unity, and Godot editor workspaces.

AI Integration Feature Box

AI Integration: Integrated Level Design Assistants

Generative diffusion models allow designers to modify landscape heightmaps and place static meshes via semantic parameters. Large language models construct and compile visual scripting nodes from natural language.





Tier 9: Third-Party Middleware

Mechanical Interface:
Specialized software packages integrated via custom data ingestion interfaces and shader configurations, solving complex generation or optimization problems without disrupting memory alignment.

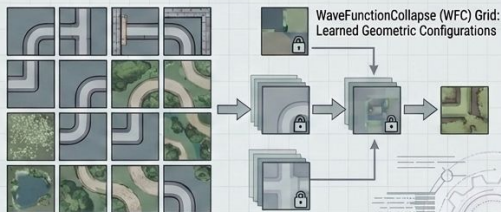
Specific Example:

Simplygon (polygon reduction),
SpeedTree (procedural vegetation),
Wwise (spatial audio).

AI Integration Feature Box

AI Integration: Procedural Content Generation via ML (PCGML)

Deep generative models learn spatial adjacency constraints from small design fragments (e.g., WaveFunctionCollapse). Models define expressively constrained design spaces refined by artist-provided positive/negative examples.



Tier 10: Web Runtimes & Cloud Streaming

Mechanical Interface:

Secure execution environments operating within strict browser sandboxes or remote server farms, managing ultra-low-latency network streaming and restricted file input/output.

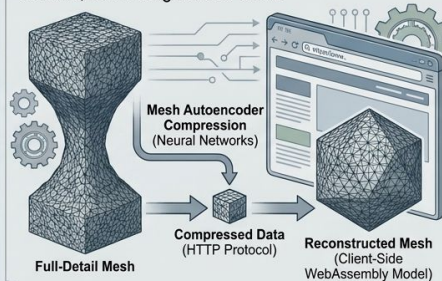
Specific Example:

WebGL/WebGPU browser contexts;
Xbox Cloud Gaming.

AI Integration Feature Box

AI Integration: Dynamic Mesh Autoencoders

Lightweight neural autoencoders compress mesh data over standard HTTP protocols. Client-side WebAssembly-compiled neural models decompress and reconstruct full-detail 3D geometry inside the browser, minimizing transfer sizes.





Tier 11: Application Logic & Simulation Agents

Mechanical Interface:

High-level gameplay simulation, visual scripting logic, and behavioral agent rulesets acting as state generators for the rendering subsystem. Operates on non-coherent, pointer-heavy heap structures.

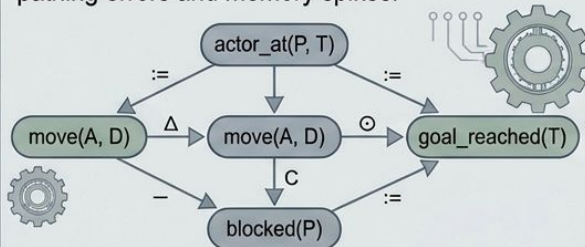
Specific Example:

C++, C#, GDScript, or Blueprints defining actor movements and transitions.

AI Integration Feature Box

AI Integration: Symbolic Design Space Modeling

Declarative specifications in Answer Set Programming (ASP) formally verify puzzle feasibility and gameplay rules. Reinforcement learning agents run automated playtests to identify pathing errors and memory spikes.



```
% Rule verification
reachable(P) :- start(P).
reachable(P2) :- reachable(P1), move(P1, P2).
               :- not goal_reached(T). % Constraint
```

Discussion: Audience/Panelist Perspectives



Assessment and Expectations

- What do individual portfolios look like for different positions?
- How rigorously do we educate (across the board) fundamentals? At what level are fundamentals now necessary to learn for all engine/graphics programmers?
- What are the gaps in educational content that need better organization from educators to keep the current workforce up to date and prepare incoming cohorts better.

Ethical questions

- How does the cost of (generative AI) tools drive access to a wider audience?
- How do we handle the balance between our curiosity to build and explore generative models and our autonomy and individual expression?
- In an educational setting, how to balance opposing perspectives on the ethical use of generative AI models?



2026