



BITM3213 • INTERACTIVE COMPUTER GRAPHICS

Special Topics in CGI

Module 9 — Virtual reality, mixed reality, game engines & the road beyond OpenGL

Where everything you've built goes next — and where the field is heading

Week 15 • Lecture 15 • FTMK • Session 2025/2026

WHAT YOU'LL GET TODAY

Learning Outcomes



Understand VR rendering

Explain stereo rendering, head tracking, and why latency matters so much.



Place AR / MR / VR

Locate them on the reality–virtuality continuum and tell them apart.



See the engine stack

Identify where your renderer sits inside a full game-engine architecture.



Know the road ahead

Compare OpenGL with Vulkan, WebGPU, and ray-tracing APIs.



Connect to your future

Relate these topics to the final project, research, and industry.

Every headset, every engine, every console game runs on the fundamentals you've built.



VR, MR, and game engines don't replace these — they build on top of them.

Today is about seeing the bigger structure your foundations slot into.

AR, MR, VR: One Continuum

The Reality-Virtuality Continuum (Milgram, 1994)



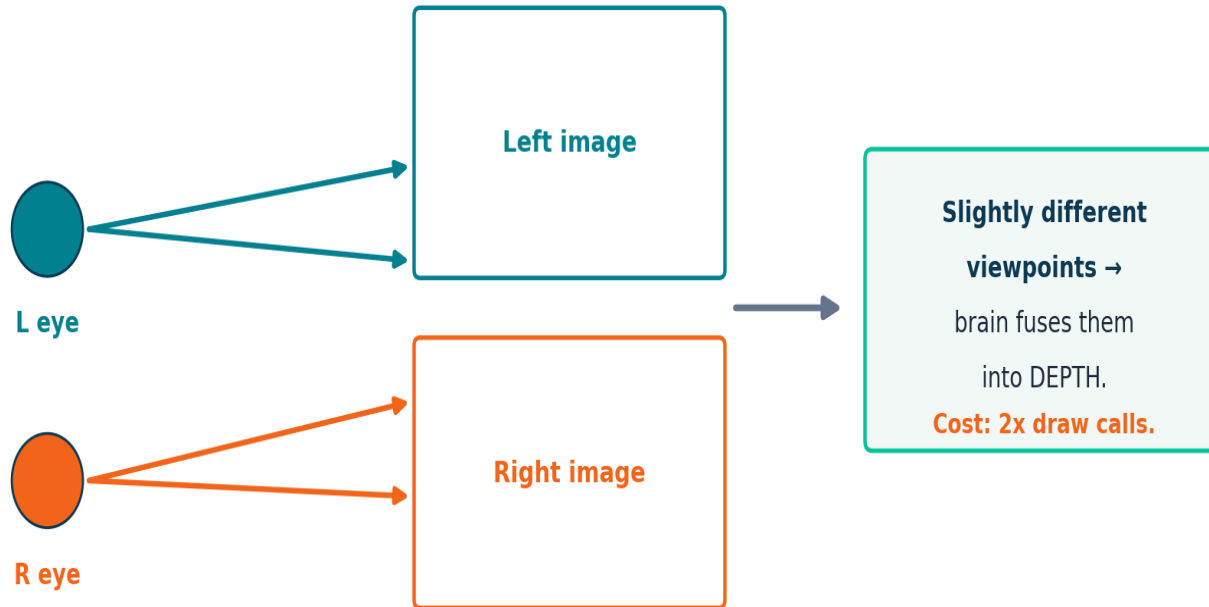
Mixed Reality (MR) = everything between fully real and fully virtual

AR adds to reality. Virtual objects overlaid on the real world (phone filters, HUDs).

VR replaces reality. A fully synthetic world; the real one is shut out by the headset.

Stereo Rendering: One Scene, Twice

VR renders the scene **TWICE** — once per eye



Two eyes = two cameras Render the scene from two slightly offset viewpoints, one per eye. The brain fuses the pair into depth. It's just two view matrices from your Lab 4 camera.

The hidden cost

Two images every frame $\approx$ double the draw calls. VR also demands 90+ FPS, so performance is brutal.

Latency, Tracking & Presence

Head tracking

Sensors measure where your head is and points. The view must update to match, instantly.

Low latency

Motion-to-photon delay must stay tiny (~20 ms). Lag breaks immersion and causes nausea.

Presence

The goal: the brain accepts the virtual world as 'real'. Needs tracking + latency + framerate all holding.



The chain is only as strong as its weakest link Great graphics with laggy tracking still feels broken. VR is a systems problem: rendering, sensing, and timing must all succeed together.

Blending Real and Virtual

Mixed reality makes virtual objects behave as if they share your real space.

That needs the system to UNDERSTAND the real world:

- Track the device's pose in the room
- Sense surfaces (floors, walls, tables)
- Occlude: real objects hide virtual ones
- Light virtual objects to match the room



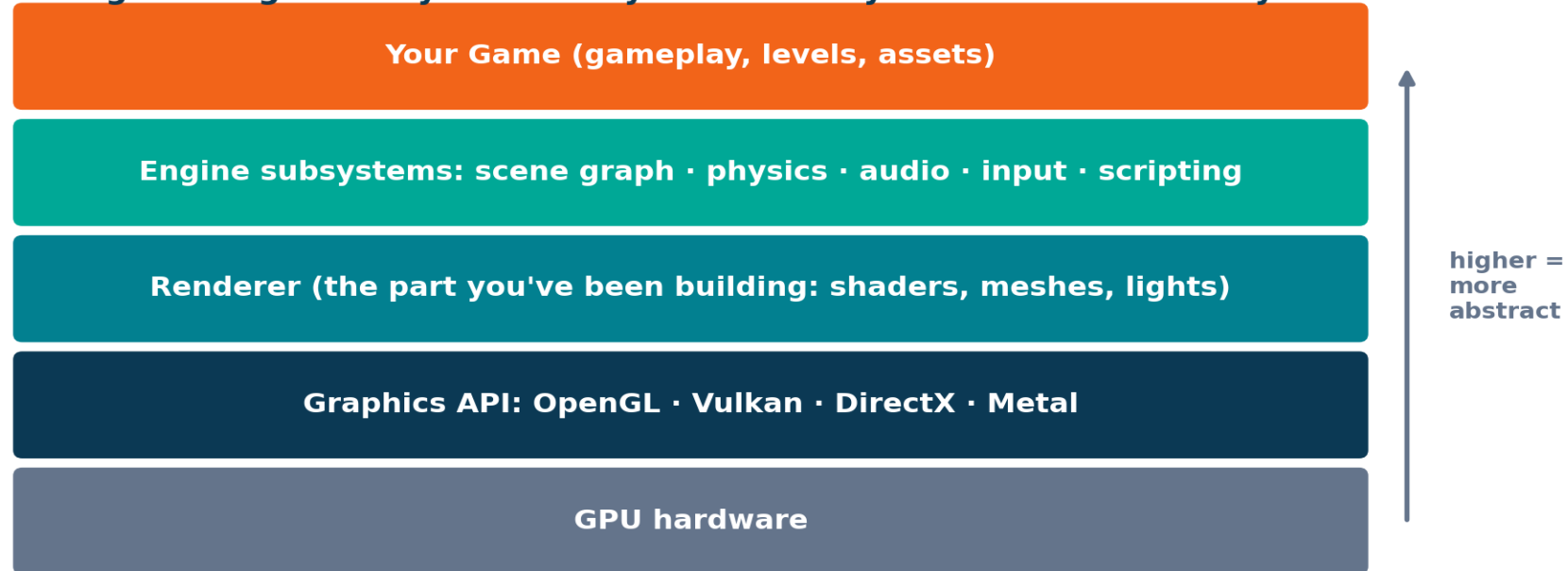
Harder than VR in one way VR controls the whole world. MR must understand a world it didn't create — sensing geometry and lighting in real time so the virtual blends in believably.

Everyday MR / AR

Furniture-preview apps that place a sofa in your room, surgical overlays on a patient, maintenance guides drawn onto machinery, and museum exhibits that come alive through a screen.

Where Your Renderer Fits

A game engine is layers — and you've already built the renderer layer



You built the middle The renderer layer — shaders, meshes, lights, cameras — is exactly what you wrote by hand. An engine adds the layers above and below so you don't rewrite them each game.

What an Engine Gives You for Free



Scene graph

Module 7's tree, built in — parent-child objects ready to use.



Physics

Lab 9's update loop, but robust — collisions, forces, constraints.



Asset pipeline

Import models, textures, audio without writing loaders (recall Lab 8).



Input & scripting

Map controls and write game logic without touching the render loop.



Audio

Positional 3D sound tied to objects in the scene.



Tooling

Editors, profilers, and debuggers to build and ship faster.

Beyond OpenGL

The road beyond OpenGL

OpenGL 3.3
(this course)

Vulkan
(explicit, fast)

WebGPU
(web + native)

RTX / ray
tracing APIs



Same core ideas — meshes, shaders, transforms, lighting — just lower-level control

Vulkan

Explicit, low-level control over the GPU. More code, but maximum performance and parallelism.

WebGPU

Modern GPU access on the web and natively. The likely successor for cross-platform work.

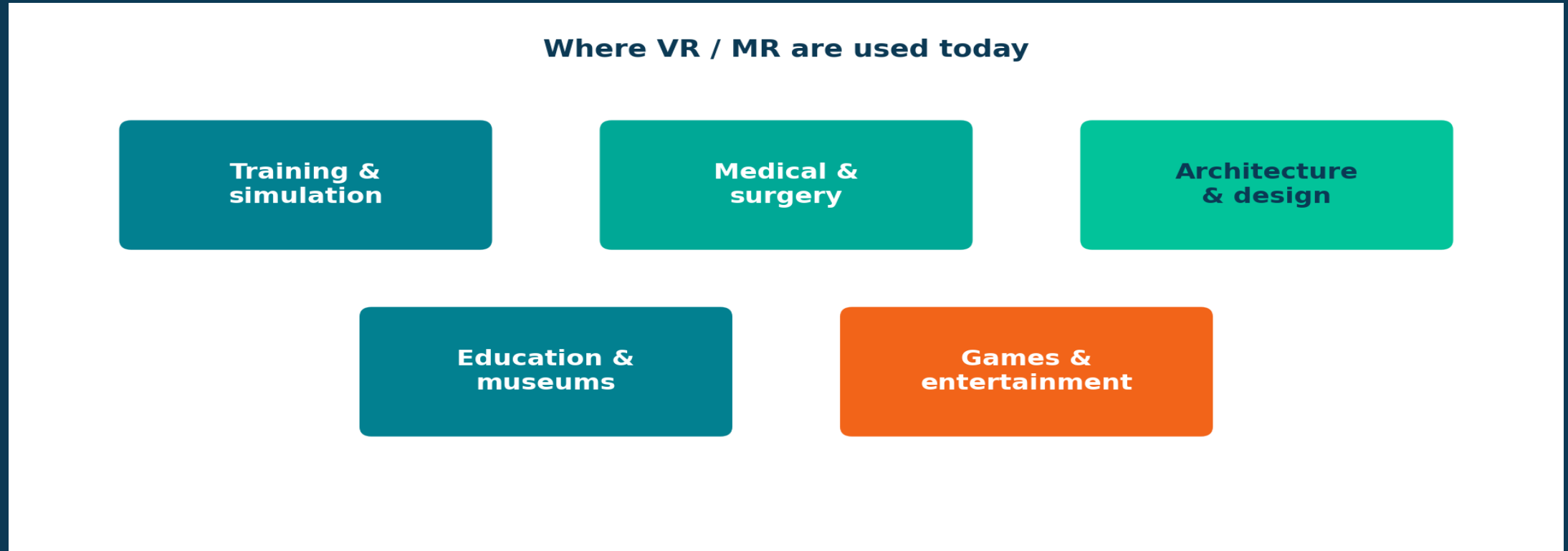
Ray-tracing APIs

Hardware ray tracing (RTX) for true reflections, shadows, and global illumination in real time.

OpenGL vs Vulkan: Why Move Lower?

	OpenGL (this course)	Vulkan / modern
Control	Driver makes many decisions for you	You control memory, sync, threading
Code to draw	Short, forgiving	Verbose, explicit, strict
Performance ceiling	Good	Higher, especially multi-core
Best for	Learning, prototypes, tools	Shipping high-performance engines

Where VR & MR Are Used Today



Far beyond games — training, medicine, design, and education all use real-time 3D.

Your Final Project Is the Proof

Everything connects

Lab 10's interactive world uses:

- Meshes & buffers (Lab 1–2)
- Transforms & hierarchy (M4, M7)
- Camera & input (M5, Lab 4)
- Lighting & materials (M6)
- Textures & models (Lab 5, 8)
- Simulation (Lab 9) + viz (M8)



Present it well

The same skills power the future:

- A walkable world is mini-VR
- Your scene graph is engine-thinking
- Your render loop scales to Vulkan
- Clear presentation = research skill

Defend your design choices with the vocabulary from this whole course.

Keep Going After This Course



Build more

Pick an engine (Unity / Unreal / Godot) or push raw OpenGL further with bigger projects.



Research

Graphics, simulation, animation, and visualization are active research areas — including here at FTMK.



Specialise

Rendering, technical art, XR, GPU computing, scientific viz — pick a direction and go deep.

learnopengl.com • Real-Time Rendering (the field's reference) • [open-source engines on GitHub](#)

Myths to Drop

'VR is just a 3D game.'

VR is a systems problem — tracking and latency matter as much as rendering.

'Engines make graphics knowledge useless.'

Engines hide the renderer, but debugging, performance, and shaders still need it.

'OpenGL is dead.'

Still ideal for learning and many tools; the concepts carry straight into Vulkan.

'Ray tracing replaces everything.'

It's added alongside rasterisation for specific effects, not a total replacement.

Key Terms — Know These Cold

Virtual Reality (VR)

A fully synthetic world replacing the real one, viewed in a headset.

Augmented Reality (AR)

Virtual content overlaid on the real world.

Mixed Reality (MR)

Real and virtual coexisting and interacting; the in-between band.

Stereo rendering

Drawing the scene twice, one slightly offset view per eye, for depth.

Latency

Motion-to-photon delay; must be tiny in VR to avoid sickness.

Presence

The brain accepting a virtual world as real.

Game engine

Layered framework (renderer, physics, audio, tools) for building interactive 3D apps.

Vulkan

Low-level graphics API giving explicit GPU control for performance.

Self-Check Questions

1

Place AR, MR, and VR on the reality–virtuality continuum.

2

Why must VR render the scene twice each frame?

3

Why is low latency so critical in VR specifically?

4

Name two things MR must sense about the real world that VR does not.

5

Which layer of a game engine did you build by hand this term?

6

Give one reason to use Vulkan instead of OpenGL.

Recap — Screenshot This Slide



AR, MR, VR are points on one reality–virtuality continuum.



Latency and tracking decide whether VR feels real (presence).



A game engine is layers; you built the renderer layer.



Same fundamentals everywhere — your skills transfer.



VR = stereo rendering: one scene drawn twice, once per eye.



MR must understand a world it didn't create.

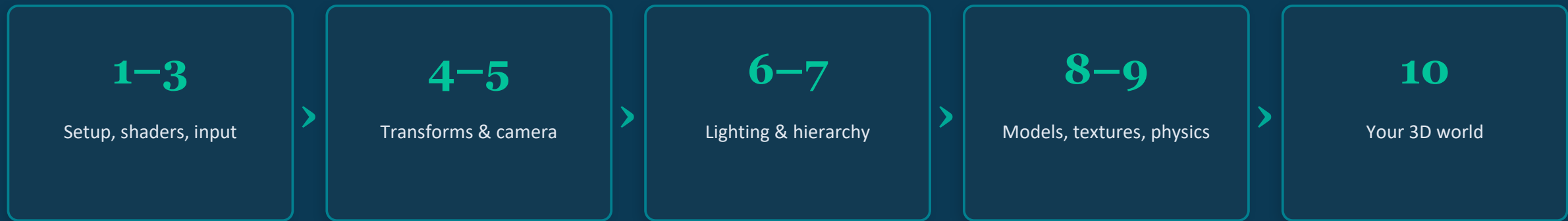


Beyond OpenGL: Vulkan, WebGPU, hardware ray tracing.



Lab 10 final project ties the whole course together.

From a Blank Window to an Interactive World



You started not knowing what a VBO was. Now you can build, light, animate, and present a 3D world.

Terima kasih — go build something brilliant.



Thank You

BITM3213 Interactive Computer Graphics — Module 9 complete

Finish Lab 10. Present with pride. Keep the fundamentals close — they'll take you anywhere in graphics.

learnopengl.com • Real-Time Rendering, 4th Ed • Gordon & Clevenger, OpenGL with C++, 3rd Ed