Console Game Development

Module Information

2022.01, Approved

Summary Information

Module Code	7119COMP
Formal Module Title	Console Game Development
Owning School	Computer Science and Mathematics
Career	Postgraduate Taught
Credits	20
Academic level	FHEQ Level 7
Grading Schema	50

Teaching Responsibility

LJMU Schools involved in Delivery
Computer Science and Mathematics

Learning Methods

Learning Method Type	Hours
Lecture	22
Practical	33

Module Offering(s)

Display Name	Location	Start Month	Duration Number Duration Unit
JAN-CTY	CTY	January	12 Weeks

Aims and Outcomes

Aims	To provide an overview of the hardware, systems architecture and ecosystem of a modern games console and its associated development kit.To explain the processing models, processor taxonomies and parallel and threaded computation models that exist on modern games consoles.To understand the low-level processes and optimise the memory layout and processing tasks of the core game subsystems on both CPU and GPU.To adapt programmatic game and real-time rendering algorithms and tailor them to a console environment.

After completing the module the student should be able to:

Learning Outcomes

Code	Number	Description
MLO1	1	Produce console game applications using game SDKs.
MLO2	2	Use low-level APIs, middleware, and game engines targeting a given console ecosystem in order to solve game software development problems on a games console.
MLO3	3	Critically evaluate the hardware architecture and processing models employed in modern game consoles.
MLO4	4	Explain the theories of parallel processing, GPU processing and memory management strategies in relation to a modern console architecture.

Module Content

Outline Syllabus	Evolution of game console hardware and architecture. Types of game platforms. Game platform languages.Game platform constraintsTask-based decompositionUsing Game Engines and Middleware targeting Game Consoles. Using Close-To-Metal Console SDKs and APIsUsing Console Profiling and Debugging Tools.Game Console Platform Support and OS Ecosystems.Assembly/machine language programming; Instruction formats; Addressing modes; I/O and interrupts.Shared memory multiprocessors/multicore organization.Introduction to SIMD vs. MIMD and the Flynn Taxonomy.Review of physical memory and memory management hardware.Memory hierarchy: importance of temporal and spatial locality.Main memory organization and operations.Buses: bus protocols, arbitration, direct-memory access (DMA).Instruction pipelining.Introduction to instruction-level parallelism (ILP).Example SIMD and MIMD instruction sets and architectures.Symmetric multiprocessing (SMP).Shared multiprocessor memory systems and memory consistency.Multiprocessor cache coherence.Multicore architectures and hardware support for synchronization.Performance figures of merit.Workloads and representative benchmarks, and methods of collecting and analysing performance figures of merit.CPI (Cycles per Instruction) equation as tool for understanding trade-offs in the design of instruction sets, processor pipelines, and memory system organizations.Amdahl’s Law.Managing Latencies: memory vs. disk latencies.Caches and the effects of spatial and temporal locality on performance in processors and systems.Introduction into the processor memory hierarchy and the formula for average memory access time.Superscalar architecture.Vector processors and GPUs.Hardware support for multithreading.Memory/disk management requirements in a real-time environment.Why system performance needs to be evaluated and what is to be evaluated.Implementation strategies such as threads.Data-parallel decomposition.Shared Memory.Maintaining spatial locality.Sequential vs. parallel processing.Parallel programming vs. concurrent programming.Request parallelism vs. Task parallelism.Re-Defining Algorithms for Games using Console Architecture Awareness.
Module Overview	
Additional Information	This module will explain the key architectural and system-level design choices and CPU/GPU processing models used in Triple-A game console design and software development. Students will be exposed to the development kits of a modern game console and will be learning to use the low-level libraries, middleware and game engines of a console in order to build and deploy a console game. Students will learn to collaborate as a development team within simulated industry conditions. The module will cover both algorithms and technologies specific to console development and key principles of processor taxonomies, multi-processing programming, stream and vector programming and memory management techniques that are required to target a constrained architecture that a game console provides.

Assessments

Assignment Category	Assessment Name	Weight	Exam/Test Length (hours)	Module Learning Outcome Mapping
Portfolio	Console Game Application	50	0	MLO1, MLO2
Centralised Exam	Examination	50	2	MLO3, MLO4

Module Contacts

Module Leader

Contact Name	Applies to all offerings	Offerings
Christopher Carter	Yes	N/A

Partner Module Team

Contact Name	Applies to all offerings	Offerings