GAME ENGINES DEVELOPMENT

Code: 26210

Main Scientific Area: Computer Graphics and Multimedia

Lecturer: Eva Ferreira de Oliveira

Language of Instruction: Portuguese

Regime: S1

Contact Hours: 30h Total Workload: 138h

ECTS: 5,0

Objectives

The Digital Games Engines Course aims to present and implement useful techniques, such as advanced concepts of object oriented programming and design patterns, used in the development of efficient, reusable and performant code.

Learning Outcomes

In this Course, the C# programming language and the Monogame framework are used to implement, in practice, the concepts presented in a theoretical way. The main problems and

difficulties that arise in the development of complex software as a team, as well as good practices to avoid these problems and optimize the performance of an interactive program in real time, will be exposed and discussed.

After approval in the Course Unit the student must have acquired the necessary skills to be able to work in coordination with other programmers in the development of complex software with high performance requirements, through the use of design patterns and good practices that facilitate the organization, reuse and code maintenance.

Course Contents

Version control - concept, repositories, working copies, branch / merge, centralized vs distributed, history, conflicts, good practices, workflows.
Object-oriented programming - concept, classes, objects, properties, methods, constructor, instantiation, access modifiers, encapsulation, object relationships, inheritance, virtual methods, abstract classes and methods, polymorphism, overloading, type casting, entity / component.

Game Loop - batch mode vs interactive, components, fixed delta time, variable delta time, free physics / semi-fixed, state machines.
Game engines - concept, history, modern game engines, components, architecture, Monogame.

Memory Management - concept, history, memory related bugs, memory management evolution, stack vs heap, fragmentation, .NET memory management, garbage collector, access speed, cache lines, good practices in games,

design patterns related to memory, implicit allocations.

Design Patterns - concept, command, flyweight, observer, sandbox subclass.
Human Interface Devices - types of HIDs, keyboard, mouse and gamepad in Monogame, input manager.
Error handling - error conditions, indicators / error codes, error handler function, exceptions - throw, try / catch, custom exception.
Generics and Serialization - concept, boxing / unboxing, constraints, binary vs. XML / JSON. Interfaces
Resources - content pipeline, resource manager
Performance - tuning, profilers, cpu-bound vs. gpu-bound, possible optimizations
Networking - Internet, TCP / UDP protocols, sockets, server / client, synchronous vs. asynchronous / blocking vs non-blocking, client and server implementation in C #, network architectures - client / server, MMO, P2P, UDP hole punching, protocols applications.

Recommended Bibliography

Game Programming Patterns, Robert Nystrom, Genever Benning - ISBN: 978-0990582908 Game Engine Architecture, Jason Gregory, A K Peters/CRC Press - ISBN: 978-1138035454

Learning and Teaching Methods

The syllabus contents covered by the Course Unit give the student the knowledge and skills necessary to develop high performance interactive real-time programs, respecting good practices and ensuring that the software is developed based on a solid, scalable and reusable architecture.

Assessment Methods

The evaluation of the course is done according to the following method:

Individual written work, weighing 15% in the final classification
First evaluation of the practical work, in groups of up to 3 students, with a weight of 25% in the final classification
Presentation and defense of the practical work, in groups of up to 3 students, with a weight of 40% in the final classification
Continuous assessment, with a weight of 20% in the final classification

Given the continuous evaluation throughout the semester and the relevant practical component of this Curricular Unit, there will be no examination at the supplementary and special periods.

The classification of continuous assessment is calculated according to:

Commitment in class
Technical ability and intellectual ability to carry out the work Communication and interpretation
Attention, participation, interest and relevance

Organization of work (including digital information) Work methodology
Behavior in class