



Course guide

270152 - VJ - Videogames

Last modified: 30/01/2024

Unit in charge: Barcelona School of Informatics
Teaching unit: 723 - CS - Department of Computer Science.

Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Optional subject).

Academic year: 2023 **ECTS Credits:** 6.0 **Languages:** Catalan, Spanish

LECTURER

Coordinating lecturer: ANTONIO CHICA CALAF

Others:

Primer quadrimestre:
JESUS ALONSO ALONSO - 13
FRANCISCO JOSE ARIAS ARNEDO - 12
ANTONIO CHICA CALAF - 11, 12, 13

Segon quadrimestre:
JESUS ALONSO ALONSO - 13
OSCAR ARGUDO MEDRANO - 12
ANTONIO CHICA CALAF - 11, 12, 13

REQUIREMENTS

- Prerequisite IDI

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:

CCO2.6. To design and implement graphic, virtual reality, augmented reality and video-games applications.

CT1.2A. To interpret, select and value concepts, theories, uses and technological developments related to computer science and its application derived from the needed fundamentals of mathematics, statistics and physics. Capacity to solve the mathematical problems presented in engineering. Talent to apply the knowledge about: algebra, differential and integral calculus and numeric methods; statistics and optimization.

CT4.3. To demonstrate knowledge and capacity to apply the fundamental principles and the basic techniques of the intelligent systems and its practical application.

CT5.3. To design, write, test, refine, document and maintain code in an high level programming language to solve programming problems applying algorithmic schemas and using data structures.

CT5.5. To use the tools of a software development environment to create and develop applications.

CT5.6. To demonstrate knowledge and capacity to apply the fundamental principles and basic techniques of parallel, concurrent, distributed and real-time programming.

Generical:

G5. TEAMWORK: to be capable to work as a team member, being just one more member or performing management tasks, with the finality of contributing to develop projects in a pragmatic way and with responsibility sense; to assume compromises taking into account the available resources.

TEACHING METHODOLOGY

Theory sessions are designed to introduce the concepts of videogame programming, going into detail on the most common algorithms. These classes will be in units of two hours once a week.

Lab sessions will present 2D and 3D tools that will be used to develop the two projects used to grade the practical component of the course. Just as in the case of the theory classes, laboratory classes are given at a rate of two hours per week.

LEARNING OBJECTIVES OF THE SUBJECT

1. Understanding the history of game development.
2. Understanding the internal structure of a computer game development team, understanding what are the tasks assigned to each role.
3. Understand the basic structure of a game, its main loop, as well as the various components that form it.
4. Learn the basic concepts used in 2D game programming: sprites, tiling, scrolling, multiple layers, parallax and isometric view.
5. Assimilate the basic concepts of 3D game programming like the typical visualization pipeline and the use of scene graphs.
6. Understand the main acceleration algorithms for interior rendering.
7. Understand the main acceleration algorithms for exterior rendering.
8. Understand how organic objects (trees, grass, water, núbols) are rendered in a game.
9. Understanding the different techniques applied to the animation of characters in video games.
10. Understand and assimilate the various types of cameras that can be used in a video game.
11. Understand the different concepts that apply to the development of particles systems. Learn the options they provide in the generation of visual effects in real time.
12. Understand the fundamentals of the application of artificial intelligence in videogames.
13. Understand what are the capabilities and limitations of the physical simulation systems in current game engines.
14. Assimilate the concepts behind the design of videogames, in particular the importance of the gameplay and its relationship with the usability of applications.
15. Demonstrate the ability to develop a game using an engine created specifically for this purpose.

STUDY LOAD

Type	Hours	Percentage
Hours large group	30,0	20.00
Hours small group	30,0	20.00
Self study	84,0	56.00
Guided activities	6,0	4.00

Total learning time: 150 h

CONTENTS

Videogame development history

Description:

History of videogame development since its inception, through its consolidation as a mean of entertainment and its extension to different platforms and media.

Basic concepts

Description:

Basic concepts of videogame programming. This includes the composition of a game programming team, the basic structure of a game, as well as the components of a game engine.



2D videogame programming

Description:

Concepts 2D game programming. Sprites, tiling, scrollers, multiple layers, parallax, isometric view.

3D videogame programming

Description:

Basic concepts. Interior and exterior rendering. Organic objects. Character animation. Types of cameras.

Videogame design

Description:

Videogame design concepts. Gameplay. Storyboard of the game. Document design.

Artificial Intelligence for videogames

Description:

Routing, finite state machines and rule systems. Action-oriented intelligence and tactical intelligence.

Particle systems

Description:

Generation, behavior, and extinction of particles and its use to achieve visual effects in a videogame.

Physics

Description:

Collisions. Dynamics of rigid bodies. Restrictions. Springs.

Additional systems

Description:

Scripting systems. Audio management. Network programming.

ACTIVITIES

Introduction to game programming

Description:

Review of the history of game development. Taxonomy of video games. Composition of a videogame programming team, with the description of the role performed by each team member.

Full-or-part-time: 2h

Theory classes: 2h



Basic videogame architecture

Description:

Description of the basic architecture of a videogame. Game Loop: presentation and update. Definition of a game engine and its components.

Full-or-part-time: 4h

Theory classes: 2h

Self study: 2h

2D game programming

Description:

Introduction to 2D game programming, introducing concepts such as sprites, tiling, scrolling, using multiple layers, parallax and isometric view.

Full-or-part-time: 8h

Theory classes: 4h

Self study: 4h

2D game programming tools

Description:

Introduction to the tools to be used for the realization of a 2D game.

Full-or-part-time: 8h

Laboratory classes: 8h

2D game development

Description:

Development of a 2D game individually.

Full-or-part-time: 18h

Self study: 18h

Introduction to 3D game programming

Description:

Review of the graphics pipeline. Description of the use of scene graphs and overlays. Introduction to the need of acceleration techniques. Types of cameras.

Full-or-part-time: 6h

Theory classes: 2h

Self study: 4h



3D game programming tools

Description:

Description of operation and use of the graphics engine to be used to implement a 3D computer game.

Full-or-part-time: 22h

Laboratory classes: 22h

3D game development

Description:

Team-based development of the 3D game.

Specific objectives:

15

Related competencies :

G5. TEAMWORK: to be capable to work as a team member, being just one more member or performing management tasks, with the finality of contributing to develop projects in a pragmatic way and with responsibility sense; to assume compromises taking into account the available resources.

Full-or-part-time: 44h

Self study: 44h

Interior rendering

Description:

Description of the portal rendering algorithm and BSP structures for accelerating the rendering of indoor scenes.

Full-or-part-time: 4h

Theory classes: 2h

Self study: 2h

Exterior rendering

Description:

Description of the data structures types used for exterior rendering (elevation maps, quadtrees). Algorithms for outdoor display (ROAM, geomipmapping). Displaying organic objects.

Full-or-part-time: 8h

Theory classes: 4h

Self study: 4h

Character animation

Description:

Comparison of explicit and implicit models. Description of keyframe-based animation. Use of hierarchies of transformations (skeletons) and transfer to the associated mesh (Skinning). Facial animation.

Full-or-part-time: 8h

Theory classes: 4h

Self study: 4h



Particle systems

Description:

Description of the operation of a particle system. Generation, behavior, extinction and visualization of a particle.

Full-or-part-time: 4h

Theory classes: 2h

Self study: 2h

Artificial intelligence for videogames

Description:

Basic routing algorithms. Finite state machines.

Full-or-part-time: 4h

Theory classes: 2h

Self study: 2h

Videogame physics

Description:

Concepts used in physics engines used in games engines: collisions, rigid bodies dynamics, springs.

Full-or-part-time: 4h

Theory classes: 2h

Self study: 2h

Additional systems

Description:

Description of the capabilities and use of scripting systems, audio and connectivity of game engines.

Full-or-part-time: 4h

Theory classes: 2h

Self study: 2h

Final exam

Description:

Written exam to assess the knowledge gained throughout the course.

Specific objectives:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14

Related competencies :

G5. TEAMWORK: to be capable to work as a team member, being just one more member or performing management tasks, with the finality of contributing to develop projects in a pragmatic way and with responsibility sense; to assume compromises taking into account the available resources.

Full-or-part-time: 2h

Guided activities: 2h



GRADING SYSTEM

Grading:

- 30% Individual 2D game project
- 40% Team-based 3D game project
- 30% Final exam

The competence "teamwork" will be evaluated based on the distribution of tasks during the development of the 3D game project. To achieve the maximum grade (A) the student must prove the ability to distribute roles in the development of the 3D game optimally. It is also necessary to demonstrate ability to collaborate with other group members.

BIBLIOGRAPHY

Basic:

- Watt, A.H.; Policarpo, F. 3D games. Addison-Wesley, 2001-2003. ISBN 0201619210.
- Gregory, J. Game engine architecture. 3rd ed. Taylor & Francis, CRC Press, 2018. ISBN 9781138035454.
- Sánchez-Crespo, D. Core techniques and algorithms in game programming. New Riders, 2004. ISBN 0131020099.

RESOURCES

Hyperlink:

- <http://www.giprogramming.com/red/>- <http://www.lighthouse3d.com>