| | | | | | |
|---|---|---|---|---|---|
| Course Title | Interactive Design | | | | |
| Course Code | ACSC330 | | | | |
| Course Type | Compulsory | | | | |
| Level | BSc (Level 1) | | | | |
| Year / Semester | 3rd (Spring) | | | | |
| Teacher's Name | Dr Stephania Loizidou, Dr Christos Markides | | | | |
| ECTS | 6 | Lectures / week | 2 | Laboratories/week | 2 |
| Course Purpose | This course provides students with an in-depth exploration of 3D game engine architecture, and the underlying. Students will learn state-of-the-art software architecture principles in the context of game engine design, investigate the subsystems typically found in a real production game engine, survey some engine architectures. The courses discusses different interactive applications, simulations, and games, and explore how the differences between interactive applications can affect the application design. Students will participate in individual hands-on lab exercises, and also work together like a real interactive application development team to design and build their own functional interactive application or game by designing and implementing assets, and components, as well as scripting, programming, and integrating 3rd party components. Students will have the opportunity to implement and develop in 2D and 3D computer graphics, user input, and apply the principles and constructs of interactive application design. | | | | |
| Learning Outcomes | Upon successful completion of the course students will be able to:<br>1. Describe in detail the functions and role of an interactive application development team.<br>2. Define game engine runtime engine subsystems, and their purpose.<br>3. Apply both in theory and practice the programmatic use of game engines and game engine architectures for the development of an interactive application.<br>4. Examine the nature of assets, video and audio, graphics, the graphics pipeline and 3rd party integration to produce visually appealing interactive applications and 2D/3D games.<br>5. Describe and compare the basic models and mechanisms for the development of interactive applications.<br>6. Discuss alternative computer graphics and APIs for the development of interactive applications. | | | | |
| Prerequisites | ACSC183, ACSC382 | | Co-requisites | | None |
| Course Content | • **Overview**: Interactive applications and game genres, survey of game engines, overview of the technologies that comprise a typical 3D game engine. | | | | |

| | |
|---|---|
| | • **Game Engine Architecture**: Overview of the technologies that comprise a typical 3D game, survey of runtime engine subsystems, survey of tools and the asset pipeline.<br>• **3D Math for 3D Graphics**: Points, vectors and Cartesian coordinates, vector operations, dot and cross product, 2D and 3D matrices, homogeneous coordinates, hierarchical coordinate frames, change of basis, introduction to quaternions, comparison of rotational representations, spheres, lines, line segments and rays, planes, and Splines.<br>• **3D Rendering**: Triangle meshes and tessellation, coordinate spaces and rendering transformations, lighting, color and texturing basics, the virtual camera and projection.<br>• **The Graphics Pipeline**: Review triangle meshes, materials, texturing, transformation, lighting basics, pipelining concepts, the rendering pipeline, tools stage, asset conditioning stage, application stage, visibility determination and primitive submission, geometry processing and rasterization stages, GPU architecture, introduction to global illumination and programmable shaders.<br>• **Time and the Game Loop**: The rendering loop, the game loop, game loop architectural styles, abstract timelines, measuring and dealing with time.<br>• **Human Interface Devices**: Types of human interface devices, interfacing with a HID, Handling various types of inputs, outputs, game engine HID systems, dead zone, detecting button-up and button-down, chords, sequences and gestures, control remapping, and context-sensitive controls.<br>• **Fundamentals of Character Animation**: Types of character animation, skeletons and poses, clips and the local time line, and skinning,<br>• **Collision Detection**: Collision detection basics, sphere vs sphere, axis-aligned bounding boxes, other collision primitives, and optimization, broad phase, narrow phase, and spatial subdivision. |
| Teaching Methodology | The course will combine theoretical aspects of interactive application and game design with practical work in implementation using the Unity cross-platform framework. Delivery will be based on 2 periods of labs and 2 periods for the theoretical part of the course. Laboratory work will mainly consist of introducing students to the practical and students are expected to complete the practical outside contact hours. Students are expected to find references from the library and on the Internet to complete their practical work. The purpose of the assignment is to gain experience in taking an original interactive application or game through the entire process: from concept to completion. Each student will work on their assignment throughout the semester, culminating in a complete interactive application or game design. |
| Bibliography | Textbooks:<br>• J. Gregory, "***Game Engine Architecture***", 3rd Edition, CRC Press, 2018, ISBN: 9781138035454.<br>• Paris Buttfield-Addison, Jonathon Manning, Tim Nugent, "***Unity Game Development Cookbook***", O'Reilly Media, 2019 (ISBN-13: 978- |

| | |
|---|---|
| | 191999158).<br><br>References:<br>• Joseph Hocking, "***Unity in Action: Multiplatform game development in C#***", 2nd Edition, Manning Publications, 2018.<br>• Harrison Ferrone, "***Learning C# by Developing Games with Unity 2019***", 4th Edition, Packt Publishing, 2019.<br>• ***Unity – Learn Tutorials***, Available [Online]: https://learn.unity.com<br>• ***Unity User Manual***, Available [Online]: https://docs.unity3d.com |
| Assessment | Students are assessed on the theoretical aspects of the course through a group project, test, and the final exam, while lab exercises cover the applied and hand-on aspects of the course. Coursework will comprise of one test, a set of lab exercises, and three-hour closed book exam. The weights for each assessment component are:<br>• Labs: 20%<br>• Group Project: 30%<br>• Test: 10%<br>• Final Exam: 40% |
| Language | English |