Course guides
804260 - PGA - Advanced Graphics Programming

Unit in charge: Image Processing and Multimedia Technology Centre
Teaching unit: 804 - CITM - Image Processing and Multimedia Technology Centre.
Degree: BACHELOR’S DEGREE IN VIDEO GAME DESIGN AND DEVELOPMENT (Syllabus 2014). (Optional subject).
Academic year: 2021 ECTS Credits: 6.0 Languages: English

LECTURER
Coordinating lecturer: Díaz García, Jesús

Others:

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:
CEVJ 5. Use programming languages, algorithmic patterns, data structures, visual programming tools, game engines and libraries for the development and prototyping of video games, in any genre and for any platform and mobile device.
CEVJ 6. Analyse, decide upon and apply graphic programming techniques, physics, artificial intelligence, interaction, augmented reality and networks to a video game project.

Transversal:
04 COE N1. EFFICIENT ORAL AND WRITTEN COMMUNICATION - Level 1. Planning oral communication, answering questions properly and writing straightforward texts that are spelt correctly and are grammatically coherent.
CT4. EFFECTIVE USE OF INFORMATION RESOURCES: Managing the acquisition, structuring, analysis and display of data and information in the chosen area of specialisation and critically assessing the results obtained.
07 AAT N1. SELF-DIRECTED LEARNING - Level 1. Completing set tasks within established deadlines. Working with recommended information sources according to the guidelines set by lecturers.

TEACHING METHODOLOGY
Lectures are divided in 2h sessions. During lectures, the teacher presents the theoretical concepts and explains them by means of examples that are solved in class. Some time is also dedicated to the resolution of the proposed exercises/projects of the subject with the assistance of the teacher (solve the doubts that may appear).

LEARNING OBJECTIVES OF THE SUBJECT
- Acquire the knowledge on the state of the art and the possibilities that computer graphics offer in the field of design and development of video games.
- Complete the knowledge on computer graphics and GPU programming acquired in previous subjects (mainly Game Engines and Augmented Reality) to implement advanced computer graphics techniques.
- Take advantage of the computational power of current GPUs to develop advanced computer graphics algorithms in real time.

STUDY LOAD

<table>
<thead>
<tr>
<th>Type</th>
<th>Hours</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self study</td>
<td>90,0</td>
<td>60.00</td>
</tr>
<tr>
<td>Hours large group</td>
<td>18,0</td>
<td>12.00</td>
</tr>
<tr>
<td>Guided activities</td>
<td>12,0</td>
<td>8.00</td>
</tr>
<tr>
<td>Type</td>
<td>Hours</td>
<td>Percentage</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>Hours medium group</td>
<td>30,0</td>
<td>20.00</td>
</tr>
</tbody>
</table>

**Total learning time:** 150 h

**CONTENTS**

**Shader programming with GLSL**

**Description:**
1. Introduction to GLSL
2. Basic GLSL tools
3. Mathematical foundations for raycasting techniques

**Related activities:**
Delivery 1: Implementation of a procedural raycasting shader

**Full-or-part-time:** 50h
- Theory classes: 10h
- Practical classes: 10h
- Self study: 30h

---

**GPU programming with OpenGL**

**Description:**
1. Global view of the render pipeline
2. Normalized device coordinates (NDC)
3. Rendering of triangle meshes
4. Passing "global" variables to shaders
5. State handling (visibility tests, blending, etc)
6. Debug and optimization tools
7. Geometric transformations
8. Render to texture

**Full-or-part-time:** 50h
- Practical classes: 10h
- Guided activities: 10h
- Self study: 30h
**Advanced computer graphics techniques**

**Description:**
1. Bump, normal, and relief mapping
2. Full-screen effects
3. Screen Space Ambient Occlusion (SSAO)
4. Environment mapping
5. Physically based rendering
6. Water
7. Bloom

**Related activities:**
Final project: Implementation of a graphics engine

**Full-or-part-time:** 50h
- Theory classes: 10h
- Practical classes: 10h
- Self study: 30h

**GRADING SYSTEM**

Two practical exercises (PE) related to specific contents of the subject will be developed and evaluated during the course. As well, a final project (FP), where the student has to implement a graphics engine with some of the techniques presented at class, will be evaluated. Participation and learning attitude (PART) is also considered. This part will be evaluated according to the participation of the student at class (solving exercises, proposing solutions/alternatives), the interest shown in learning the different techniques. As a project subject, there is no reevaluation exam.

Final mark = 0.30 * PE1 + 0.30 * PE2 + 0.30 * FP + 0.1 * PART

**EXAMINATION RULES.**

Part of the activities will be developed at class with the assistance of the lecturer. Students should also work autonomously to finish the activities and projects proposed during the course.

Projects can be done in groups of 2 people and will be submitted via the Campus Virtual following the guidelines provided by the instructions document of each one (name of the files, etc.). Projects submitted after midnight of the specified date will be considered as NP. Any issues that do not allow the student to submit a project in time should have a reasonable cause and must be communicated with enough anticipation to the lecturer. The evaluation of the projects does not consist just on submitting the code, but also on oral presentations when required.

Projects have to be executed at CITM, so be sure that you work with the same version of the software as the one provided at the center and that your projects can be executed there without errors.

**BIBLIOGRAPHY**

**Basic:**

**Complementary:**
RESOURCES

Other resources:
ShaderToy: https://www.shadertoy.com
GLSL Sandbox: http://glsandbox.com
OpenGL: https://www.opengl.org
OpenGL loader: https://glad.dav1d.de
GLM library: https://github.com/g-truc/glm
GLFW library: https://www.glfw.org
ImGui library: https://github.com/ocornut/imgui
STB library: https://github.com/nothings/stb
Assimp library: https://www.assimp.org