270152 - VJ - Videogames

Coordinating unit: 270 - FIB - Barcelona School of Informatics
Teaching unit: 723 - CS - Department of Computer Science
Academic year: 2018
Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Teaching unit Optional)
ECTS credits: 6
Teaching languages: Catalan, Spanish

Teaching staff

Coordinator: - Antonio Chica Calaf (achica@cs.upc.edu)
Others: - Alejandro Ríos Jerez (arios@cs.upc.edu)
- Jesus Alonso Alonso (jalonso@cs.upc.edu)
- Marc Comino Trinidad (mcomino@cs.upc.edu)

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 the tools and 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, nubols) 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**

<table>
<thead>
<tr>
<th>Total learning time: 150h</th>
<th>Theory classes: 30h</th>
<th>20.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Practical classes: 0h</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td>Laboratory classes: 30h</td>
<td>20.00%</td>
</tr>
<tr>
<td></td>
<td>Guided activities: 6h</td>
<td>4.00%</td>
</tr>
<tr>
<td></td>
<td>Self study: 84h</td>
<td>56.00%</td>
</tr>
</tbody>
</table>
## Content

### Videogame development history

**Degree competences to which the content contributes:**

**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

**Degree competences to which the content contributes:**

**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

**Degree competences to which the content contributes:**

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

### 3D videogame programming

**Degree competences to which the content contributes:**

**Description:**

### Videogame design

**Degree competences to which the content contributes:**

**Description:**

### Particle systems

**Degree competences to which the content contributes:**

**Description:**
Generation, behavior, and extinction of particles and its use to achieve visual effects in a videogame.
### Artificial Intelligence for videogames

**Degree competences to which the content contributes:**

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

### Physics

**Degree competences to which the content contributes:**

**Description:**

### Additional systems

**Degree competences to which the content contributes:**

**Description:**
Scripting systems. Audio management. Network programming.
## Planning of activities

<table>
<thead>
<tr>
<th><strong>Introduction to game programming</strong></th>
<th><strong>Hours:</strong> 2h</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Basic videogame architecture</strong></th>
<th><strong>Hours:</strong> 4h</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td></td>
</tr>
<tr>
<td>Description of the basic architecture of a videogame. Game Loop: presentation and update. Definition of a game engine and its components.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>2D game programming</strong></th>
<th><strong>Hours:</strong> 8h</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td></td>
</tr>
<tr>
<td>Introduction to 2D game programming, introducing concepts such as sprites, tiling, scrolling, using multiple layers, parallax and isometric view.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>2D game programming tools</strong></th>
<th><strong>Hours:</strong> 8h</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td></td>
</tr>
<tr>
<td>Introduction to the tools to be used for the realization of a 2D game.</td>
<td></td>
</tr>
</tbody>
</table>
### 2D game development

**Hours:** 12h  
Theory classes: 0h  
Practical classes: 0h  
Laboratory classes: 0h  
Guided activities: 0h  
Self study: 12h

**Description:**  
Development of a 2D game individually.

### Introduction to 3D game programming

**Hours:** 8h  
Theory classes: 4h  
Practical classes: 0h  
Laboratory classes: 0h  
Guided activities: 0h  
Self study: 4h

**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.

### 3D game programming tools

**Hours:** 22h  
Theory classes: 0h  
Practical classes: 0h  
Laboratory classes: 22h  
Guided activities: 0h  
Self study: 0h

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

### 3D game development

**Hours:** 44h  
Theory classes: 0h  
Practical classes: 0h  
Laboratory classes: 0h  
Guided activities: 0h  
Self study: 44h

**Description:**  
Team-based development of the 3D game.
### Interior rendering

**Hours**: 4h  
- Theory classes: 2h  
- Practical classes: 0h  
- Laboratory classes: 0h  
- Guided activities: 0h  
- Self study: 2h

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

### Exterior rendering

**Hours**: 8h  
- Theory classes: 4h  
- Practical classes: 0h  
- Laboratory classes: 0h  
- Guided activities: 0h  
- Self study: 4h

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

### Character animation

**Hours**: 8h  
- Theory classes: 4h  
- Practical classes: 0h  
- Laboratory classes: 0h  
- Guided activities: 0h  
- Self study: 4h

**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.

### Particle systems

**Hours**: 4h  
- Theory classes: 2h  
- Practical classes: 0h  
- Laboratory classes: 0h  
- Guided activities: 0h  
- Self study: 2h

**Description**:  
Description of the operation of a particle system. Generation, behavior, extinction and visualization of a particle.
## Artificial intelligence for videogames

**Description:**
Basic routing algorithms. Finite state machines.

**Hours:** 4h
- Theory classes: 2h
- Practical classes: 0h
- Laboratory classes: 0h
- Guided activities: 0h
- Self study: 2h

## Videogame physics

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

**Hours:** 4h
- Theory classes: 2h
- Practical classes: 0h
- Laboratory classes: 0h
- Guided activities: 0h
- Self study: 2h

## Additional systems

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

**Hours:** 4h
- Theory classes: 2h
- Practical classes: 0h
- Laboratory classes: 0h
- Guided activities: 0h
- Self study: 2h

## 2D game presentation

**Description:**
Implementation of a simple 2D game.

**Specific objectives:**
3, 4

**Hours:** 2h
- Guided activities: 2h
- Self study: 0h

## Presentation of a 3D game

**Hours:** 2h
- Guided activities: 2h
- Self study: 0h
270152 - VJ - Videogames

Description:
Implementation of a small 3D game using an engine designed for this purpose. Presentation of the resulting game.

Specific objectives:
3, 5, 14, 15

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 |

Hours: 2h
Guided activities: 2h
Self study: 0h

Qualification system

Grading:
20% Individual 2D game project
50% 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:

Others resources:
Hyperlink
http://www.glprogramming.com/red/

http://www.lighthouse3d.com