

# Tecnologias e Sistemas de Informação para a Web

## SISTEMAS GRÁFICOS – 2018/2019 – Project

### Goals

The Graphics Systems project aims to create a 3D graphic application using WebGL / Three.js. Students are free to choose work themes. Some suggestions:

1. 3D games  
Car races / Fight tanks / Starship shooter / Robot / 3D Puzzle / ...
2. Graphic Simulators  
Flying or driving / Sports simulators / Visualization of molecules / ...

Some examples:

- <http://nouvellevague.ultranoir.com/>
- <https://web.chemdoodle.com/demos/molgrabber-3d/>
- <http://arodic.github.io/p/jellyfish/>
- <http://cycleblob.com/>
- <https://www.cubeslam.com/ncokau>

The groups should identify themselves and describe the chosen theme in the Wiki available in Moodle until the **22th of April**.

### Implementation

Students must implement this work as a group. The groups must be constituted up to a maximum of 3 elements (these should be from the same class). Students with TE status can develop the project individually.

The use of frameworks or other graphic libraries other than **Three.js** and **Physi.js** must be previously agreed with the teacher of the curricular unit.

The use of available resources (online and not only) should be used as a source of inspiration or help in solving small algorithmic issues. All assets/code not developed by the group must be properly referenced. The work will be **disqualified** if plagiarism is detected. Plagiarism may trigger a **disciplinary process**.

### Requirements

For this work it is **mandatory** to:

- Use Three.js
- Implement object / camera animation
- Use textures

For this work it is advised to:

- Implement advanced modeling mechanisms (illumination, camera/object manipulation);
- Implement multiple cameras / scenes;
- Implement picking and/or dragging mechanisms;
- Import static 3D models.

For this work valuations will be considered (for > 17 values) if at least two of the following are included:

- Particle systems;
- Menu and help system;
- Overlays (information about the scene, i.e. mini map, application/player information, etc.);
- Imported animated models.

For this project students should concentrate on three components: **visualization**, **interaction** and **animation**. Originality, aesthetics and technical quality of the code are, of course, important factors to consider in the evaluation.

Applications may be based on class exercises, but the work already done will NOT be considered for evaluation (new features need to be developed and additional techniques needed).

Project delivery consists of a simple report (see structure in the next point) and the necessary files to visualize the application (all libraries, images and models included).

## Report

The final report (max 10 pages in **PDF** format) should include the theme and group identification and the following sections:

- **Introduction** Description of the proposed work and its goals, even if not achieved;
- **Application** Presentation of all implemented functionalities and possible interactions with the application; presentation of sample screenshots; it is important to describe the difficulties encountered by the group and how they were (or not) surpassed;
- **Conclusion** Brief conclusion of how the work was carried out and any other particularities that the group deems worthy of reference;
- **Resources** Indication of resources - bibliography, web resources (webography), assets references... - used to support the project development;
- **Annexes** Optional section to place information that does not fit in the previous points.

The report can be written in Portuguese or English.

## Evaluation

The final evaluation of the project is based on:

- Oral defense (10%);
- Report (10%);
- In class follow-up (10%);
- Project (70%).

## Delivery and defense dates

Delivery: ZIP file with all the application and report until **June 16** via Moodle.

- Work submitted after the deadline will not be accepted

Defense: date and time to be defined (mandatory presence of all group members)

- Absent elements will not be evaluated