

Project Requirements

Interactive Graphics Course
Prof. Marco Schaerf
Dept. of Computer, Control and
Management Engineering (DIAG)
Sapienza University of Rome
marco.schaerf@uniroma1.it

Passing the exam

- Two ways to pass the exam
 1. Homeworks (deadlines during the course) + Project (deadlines after the end of course, latest in September)
 2. Oral Exam
- Oral examinations will take place at fixed dates (see InfoStud). Please register on InfoStud before coming.

Project requirements 1/2

- You choose the theme
- Can be done in groups of 1 to 4 person
- You can use «basic» WebGL or advanced libraries, such as ThreeJS (<http://threejs.org/>) or Babylon (<http://babylonjs.com/>) or others (in this case I must approve them)
- You can use models created with a modeler or found on-line. **YOU CANNOT IMPORT ANIMATIONS**

Project requirements 2/2

- The project MUST include:
 - Hierarchical models
 - At least one and more complex of the model used in homework2
 - Lights and Textures
 - At least one light, textures of different kinds (color, normal, specular, ...)
 - User interaction
 - Depends on your theme, as an example: turn on/off lights, change viewpoint, configure colors, change difficulty,
 - Animations
 - Most objects should be animated, in particular the hierarchical models should perform animations that exploit their structure. ANIMATIONS CANNOT BE IMPORTED, should be implemented by you in javascript (WebGL, ThreeJS or other approved library)

Library for smooth animations

- For smooth animations I suggest using tween.js
<https://github.com/tweenjs/tween.js/>
- See here the documentation:
https://github.com/tweenjs/tween.js/blob/master/docs/user_guide.md
- It includes Easing functions to accommodate for different interpolation functions

Physics Engines in JS

The following Physics engines are available in javascript and can be integrated with WebGL, ThreeJS and BabylonJS:

1. Cannon.js: <https://schteppe.github.io/cannon.js/> is a lightweight implementation that can be integrated with all libraries
2. Ammo.js: <https://github.com/kripken/ammo.js/> is a javascript port of Bullet. Can be integrated with all libraries
3. Physijs: <https://chandlerprall.github.io/Physijs/> is built on top of Ammo.js to make it simpler its integration with ThreeJS

Physics Engines in the project

- You are not required to use a physics engine in your project, but (depending on the subject of your project) it might help to make it more realistic.
- If you use it, choose the one that is more easily integrated in your project and that is fast enough. Use it only if it does not slow down too much your project

Project steps

1. Come up with an idea for a possible project
2. Activate your GitHub Classroom repository at his URL https://classroom.github.com/g/BQq_EJhc register all the team members and create there your project. This repository should contain ALL the source code (including the used libraries) plus the documentation
3. Work on the project, before the deadline please activate GitHub Pages in the main directory and check that your project is executable on GitHub. Put the link in the README file.
4. Register in Infostud in the correct session and send me email (before the deadline) that you completed the project and it can be evaluated

Project presentation

- The accompanying document should be both a technical presentation and a user manual and contain:
 - Description of the environment used (basic WebGL or other)
 - List of all the libraries, tools and models used in the project but not developed by the team
 - Description of all the technical aspects of the project
 - Description of the implemented interactions
 - The length is up to you, at least 5-10 pages

Deadlines

- The project **MUST** be completed by one of these 3 deadlines:
 - June 27th 11.59pm
 - July 25th 11.59pm
 - September 26th 11.59pm
- When a deadline comes all completed projects will be marked.
- Late submissions will **NOT** be allowed, the repositories will close on the last deadline