# *CG Lab SS2016: Create a Movie*

**April 2016**

**Instructors:** Clemens Birklbauer, Samuel Gratzl, David Schedl, Marc Streit

**Primary contact:** cg-lab@jku.at

**Groups of two students**

- Groups can be built irrespective of lab group assignment in KUSSS.
- If you don't find a partner, contact us as soon as possible and we will match you with a colleague.
- If one of the partners drops the course (after the first submission), let us know immediately so that we can adapt the scope of the project.

## Grading

- Dropping the course after the first submission (movie concept) will result in a negative grade.
- Total: 100 pts
- Total points are split between team members. Students have to state the distribution depending on their workload (e.g., 60% student A and 40% student B) in the final submission.
- 50 pts/student = grade 1, <43.75 = 2, <37.5 = 3, <31.25 = 4, <25 = 5
- **Grading criteria**
  - **50pt** Minimal aspects and basic movie parts (see checklist below)
  - **30pt** Special effects (as listed below)
  - **10pt** Algorithmic effects description:
    - Describe how the basic movie effects and the special effects can be seen.
    - Describe how the special effects work in theory and how they are implemented.
  - **10pt** Effect integration:
    The effects need to conceptually fit into the submitted movie screenplay.

## Plagiarism

- Each group is supposed to work alone. However, supporting each other is of course legitimate.
- Providing code to other groups is not allowed! It does not matter whether it is on purpose or not.
- Do not copy code from the Internet. If you do so, make a comment in the code and you don't get any points for it.
- All involved groups in an incident get 0 points!

## Minimal requirements

- Must be 3D.
- Must be a movie with animated characters/objects.
- Use time-based animation (frame rate independent).
- Movie must contain three independent scenes (like stages in a theatre play).
- All scenes must be part of the same world.
- Each scene must be visited in an animated camera flight.
- At least one clearly visible object needs to be animated at any time during the camera flight.
- In addition, it should be possible to freely control the camera using the mouse and keyboard to fly though the scene. In this case, the animation inside the individual scenes must be automatically triggered when the camera comes close (i.e., enters a certain radius).
- Must be based on the lab framework (which will be provided in time).
- Lab framework (everything in the libs folder) must remain unmodified. However, you are free to add new scene graph nodes.
- Modular code (create own function for each required feature if possible)
- Code documentation (comments)
- Algorithmic effects description (separate document)
- All aspects of the basic movie effects (see list below) as well as the special effects must be clearly visible during the animated camera flight.
- The animated camera flight must last exactly **30 seconds.**
- Executability
    - Runs on Pool PCs with Atom Live Server (as in the tutorials).
    - Must be self contained including all required resources (e.g., libs folder).
- Final interviews
    - Understanding will be checked in individual interviews at the end of the semester. Can include conceptual questions as well (e.g., Phong shader theory, transformation pipeline, how to render objects without the scene graph implementation).
    - All parts of the project must be fully understood, even if it was primarily contributed by only one of the group partners.

## What is not allowed

- Usage of 3D/game engine (e.g., Three.js).
- Implement animations in external modelling software (e.g., Blender).
- 1:1 copies from the lab.

## Basic movie effects

- **Scene graph elements**
  - Add at least one manually composed model that consists of multiple parts (e.g., roboter with arms and legs).
  - Separate animation of different parts and animation (movement) of the model as a whole.
  - Create one scene graph node that renders a complex 3D shape (5-25 vertices; not a cube, sphere, quad, or loaded model). Fully specify properties for this object, i.e., vertices, normals, texture coordinates.
- **Materials**
  - At least two clearly different materials with specular properties (just setting different vertex colors is not sufficient).
- **Texturing**
  - Apply texture to your self-created complex render node (see scene graph elements above).
- **Illumination**
  - Use multiple light sources (at least 1 moving).
  - Implement at least one spot-light by extending existing light node (*LightSGNode*) and Phong shader.
  - Phong shading should be applied to all objects in scene.
- **Transparency**
  - At least one object must be semi-transparent based on an alpha texture. During rendering, the alpha value of each fragment should be determined through a texture lookup.
- **Camera**
  - Use the keyboard to control forward (up-key) and backward movement (down-key) along the viewing direction and the mouse to control the heading (mouse-x) and pitch (mouse-y) of the camera relative to the ground (no roll!).
  - Additional animated camera flight.

## Special effects

- Difficulty: easy = 10pt, medium = 20pt
- Multiple effects must be combined up to a maximum of 30 points.
- Trivial solutions can result in decreased points.
- No points for effects in external libraries, however, they can be used to enhance the movie (e.g., sound effects).

### Selection of effects

- **Easy (10 pt)**
  - **Terrain from heightmap**

    Must be implemented using a vertex shader.
  - **Multi texturing**

    Mix multiple textures (e.g., grass, rock) with an alpha map in a shader.
  - **Postprocessing shader**

    Toon shader, bloom shader, glare effect, sobel operator, color space transformation, etc.
  - **Level of detail**

    Implement one render node in three different detail levels and decide which version to render by taking into account the distance of the object to the camera.
  - **Billboarding**

    Implement a billboarding node that makes a flat object face the camera orthogonally at all time.
- **Medium (20 pt)**
  - **Animated water surface**

    Implement a water surface that reflects a skybox and animates wave movements.
  - **Minimap**

    Show minimap in one of the corners of the screen (always visible) that contains an abstract version of the world (2D texture or 3D rendered) together with the path of the camera over the last 10 seconds. Moving objects don't have to be shown. Use separate viewport for minimap.
  - **Particle system** (e.g., rain, smoke, fire)
  - **Motion blur**

    For both: moving objects and camera motion.

- *Own suggestions are welcome, but need to be described in the movie concept submission. Approval will be given by mail.*

## Deliverables

- **Part 1: Announce group members and hand-in movie concept**

  **DUE ON 20.05.2016 23:59**

  - One email per group to cg-lab@jku.at
  - Subject: [movie-concept] <MATR#_STUD_1>/<MATR#_STUD_2> <MOVIE_NAME>
  - Body:
    - Movie name
    - Name and immatriculation number of both students
    - Overall movie idea and concept
    - Chosen special effect(s) (adding up to exactly 30 points!)

- **Part 2: Hand-in final package**

  **DUE ON 19.06.2016 23:59**

  - One email per group to cg-lab@jku.at
  - Subject:

    [final-movie]<MATR#_STUD_1>/<MATR#_STUD_2> <MOVIE_NAME>
  - Create ZIP file for full package:

    <MATR#_STUD_1>_<MATR#_STUD_2>.zip
  - Package needs to contain:
    - `code/` Documented code + required resources
    - `howto/` User manual

      (screenshots + description of movie)
    - `special_effect/` Algorithmic description of chosen special effect(s)
    - `team_workload.txt`

      Needs to contain workload for each team member in %. Must add up to 100%!

      Content of file:

      <MATR#_STUD_1> <X>%

      <MATR#_STUD_2> <Y>%
  - Don't attach large package files to the mail - send link to Dropbox/GoogleDrive/etc.

    (we will download the packages **after** the deadline has passed)

- **Part 3: Individual interviews** (alone, but members of same group sequentially)

  **24.06.2016, 27.-28.06.2016**

  - Code walkthrough + conceptual questions
  - Sanity checks
  - Doodle poll for coordinating time slots

**Late submissions are possible, yet they will be penalized.**

- 1 day: 10% penalty
- 2 days: 20% penalty
- 3 days: 40% penalty
- >3 days: 100% penalty
- Note that the penalty always applies to the total project points.

**Penalty for changing a special effect after the movie concept deadline:**

- If reported before the final submission: 50% penalty of the effect points
- Without announcement: 100% penalty of the effect points