

Project assignment

Note: Please visit this page on a regular basis for possible updates.

Wild west and canyons

This worksheet describes the project assignment for this course. You are asked to provide solutions for the modeling, rendering, animation and interaction of a western/canyon scene, with animated and/or interactive elements.

The idea is to create a scenery in this theme. Your scene should involve modeling the ground through procedural means in the shader, and should include an animated river, and any adjoining animation and rendering effects. Your scene can be populated with any number of buildings, creatures and plants. You can use off-the-shelf 3D models but only parcimoniously: prioritize your own model designs, and procedural models.

You may include animated character(s) or animal(s) but this is not mandatory, and left to your imagination. Your scene may involve interactive elements or play like a game. The goal of the project is to showcase your acquired knowledge in modeling, rendering, animation and demonstrate how you used these techniques to your classmates.

Warning

Manage your time

We are well aware of the number of projects during the semester.

Some recommendations:

- *Start early.* The project assignment is given in the first half of the semester, so don't wait until the last minute.
- Focus on some aspects and keep others simple. Obviously try do your best and enjoy, but don't spend too much time on this project at the expense of others!

1. Requirements

What is important is the content of your project in terms of modeling, rendering and animation. The following requirements should be fulfilled:

Modeling

- Model at least one element with hierarchical and procedurally generated geometry. This could typically be a plant with hierarchical branches, or the river path. By procedural, we mean that the geometry needs to be generated by your code and not simply loaded from a 3D model file. The generation could be on GPU or CPU but should not use any external dependency outside the ones already provided with the code base.
- Model at least one other part of the scene or terrain procedurally. Again, by procedural, we mean that the geometry needs to be generated by your own code, with no external dependency.

Rendering

- Use lights and materials (as seen in [Practical 4 - Local Illumination](#)).

- Render an atmospheric effect on GPU in your shader (clouds, procedural sky, etc).
- render the surroundings (sky, background scenery) with a true skybox. By true skybox, we mean that the skybox should always be perceived at infinity (no parallax) and the observer could never get out of it. This means neutralizing the translational component of whatever proxy primitives or geometry you will use to render the skybox.
- render a particle system with e.g. splatting or textured billboards (e.g. smoke), or alternatively render a water surface effect (reflection/refraction, etc)

Animation

- Animate one element with keyframes (to move it or change one of its properties)
- Create a fluid animation (water, clouds, smoke) with a technique of your choosing. This could be optionally coupled with the particle system or water surface effect in the last bullet of the rendering section.
- Control one element with the keyboard.

Optimizations

- Demonstrate in your project that you are careful in your resource usage
- Show at least one speed / efficiency optimization. Simply reusing objects several times in your scene is not sufficient: please justify improvement in terms of number of polygons or draw calls rendered per frame, built-in feature in your procedural modeling, or some management aspect of the level of detail in your scene.

Other effect of your choice

- Add at least one element that was not covered in the practicals, whatever it is!
- Animating an animal or human character using the skinning animation technique is one idea. Most human models possess animations when loaded from an FBX file.
- Another idea is to create an additional procedural animation on CPU and/or GPU (e.g. water surface, moving plants, smoke, flock of animals or birds...). By procedural, we mean that the effect should be generated in your shader and or Python code and directly affect the geometry in the shader as a function of time.
- Normal mapping, caustics, shadow mapping, PBR, are all rendering techniques that can be included. These techniques are more or less listed here in order of difficulty.

Note

Some elements or effects can be basic, but at least one part of your project must be new or an improvement of what was provided in the practicals. Originality of the technical proposition is encouraged and rewarded. Choose what is the most interesting to you!

2. Logistics

Grading Criteria

- Pertinence of proposed choices and solutions
- Quality and difficulty of implemented techniques
- Fun?

Danger

Projects are graded personal work. Projects will be run into automated plagiarism detection tools. Any projects with significant and intentional overlap may be subject to penalty, including the worst case penalty of a 0 grade.

Teams

Projects must be realized by groups of three students. A small number of two student groups may be tolerated if group size isn't a multiple of 3 (ask your instructor directly).

Project archive

An archive of your project must be uploaded on Teide (Ensimag students). It must contain:

- Your entire source code, which must be tested and runnable directly on Ensimag or UFR PCs.
- All the additional material necessary to run it, like meshes and textures, in place.
- A list of controls must print when your program is executed, so that we can run *and* play your project!
- *Please follow this convention* : your archive *must* extract to a single directory called “*Team xx - Name1 Name2 Name3*”, with xx your *two-digit team number*, and then the *last names of your team members*.

Warning

Only put the resources strictly necessary to run your project in your uploaded archive and not a plethora, memory intensive set of assets. Also be careful to not include temporaries in your archive (such as `__pycache__` folders, editor temporary files, etc). You are not obligated to use all the assets provided, you can pick whatever you like or use other ones you found.

Danger

Any project that does not follow the basic rules above for the archive may be penalized on the final grade.

3. Deadline

This date may be slightly adjusted according to your other projects deadlines.

Polytech: Firday April 5th, 2024

Ensimag & Mosig: Friday April 19th, 2024