

Winter Semester 2024/25

Assignment on Virtual Reality and Physically-Based-Simulation - Sheet 5

Due Date January 28, 2025

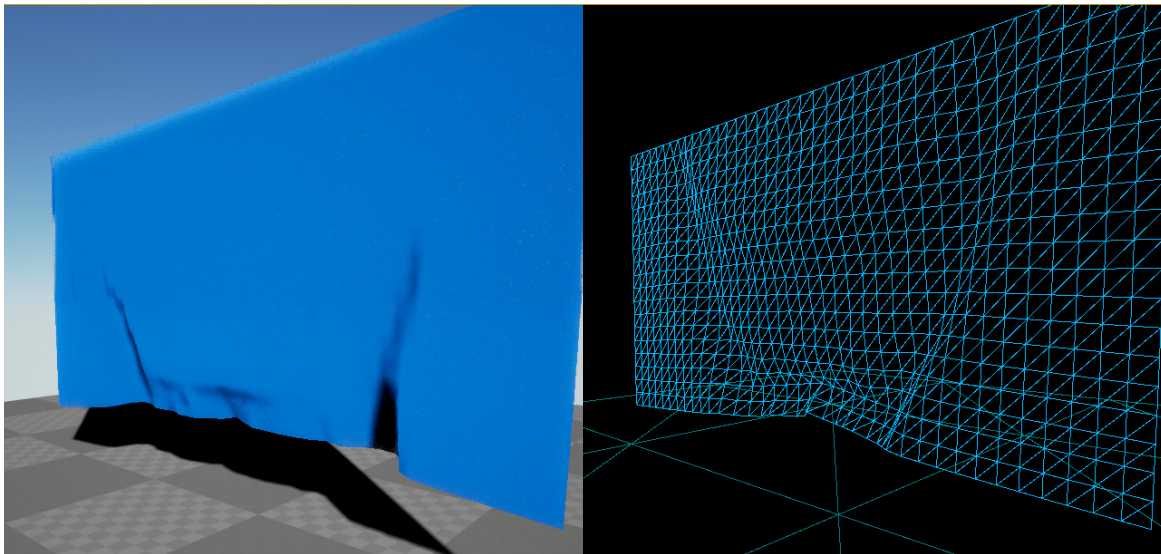


Figure 1: Final mass-spring system. Left: Rendering in lit mode. Right: Rendering in wire-frame mode.

Exercise 1 (Mass-Spring-Systems, 8 Credits)

The goal of this exercise is to implement a mass-spring system. On our website you can find an Unreal project with most parts already implemented.¹ When you recall the definition of a mass-spring system from the lecture slides, you will recognize the two main components in the `Spring.{h,cpp}` and `MassPoint.{h,cpp}` files. The `SpringMassActor.cpp` glues the system together. It initializes the mass points and springs, calls the update methods, and further creates a mesh to visualize our system (`ASpringMassActor::initSpringSystem`).

¹ http://cgvr.cs.uni-bremen.de/teaching/vr/uebungen/spring_mass.zip

In the provided level, you can press F to apply a force to the center of the mesh. The logic behind it is implemented in the `ASpringMassActor::Touch` function. By pressing the keys 0 and 9, you can switch the rendering between wire-frame and lit mode.

To generate the Visual Studio project files, right-click on the `.uproject` file and select *Generate Visual Studio project files* (If you don't see the option, it might be hidden under *Show more options*). Afterwards, open the `spring_mass.sln` file and start the unreal editor from Visual Studio. When you change the code, save the file and click on the compile button in the unreal editor. A notification in the bottom right corner informs you whether the code was reloaded successfully. This way you do not have to restart the editor for each change, and you can use the debugger from within Visual Studio (Figure 2). For MacOS users: The process of generating project files for XCode should be similar. Let me know if there are problems.

- a) Implement the force calculation for each spring and add it to the mass points (`Spring::Tick`). You can access the members of the connected mass points (`m_m1`, `m_m2`) directly, as `class Spring` is a friend of `class MassPoint`. (Certain slides from the lecture can be **very** helpful for this...)
- b) Add a gravitational force to each mass point in `MassPoint::updateGravity`.
- c) Implement a perturbed gravitation vector² instead of a constant one to each mass point.
- d) Change the integration method in `MassPoint::updateCurPos` to use the Verlet method.

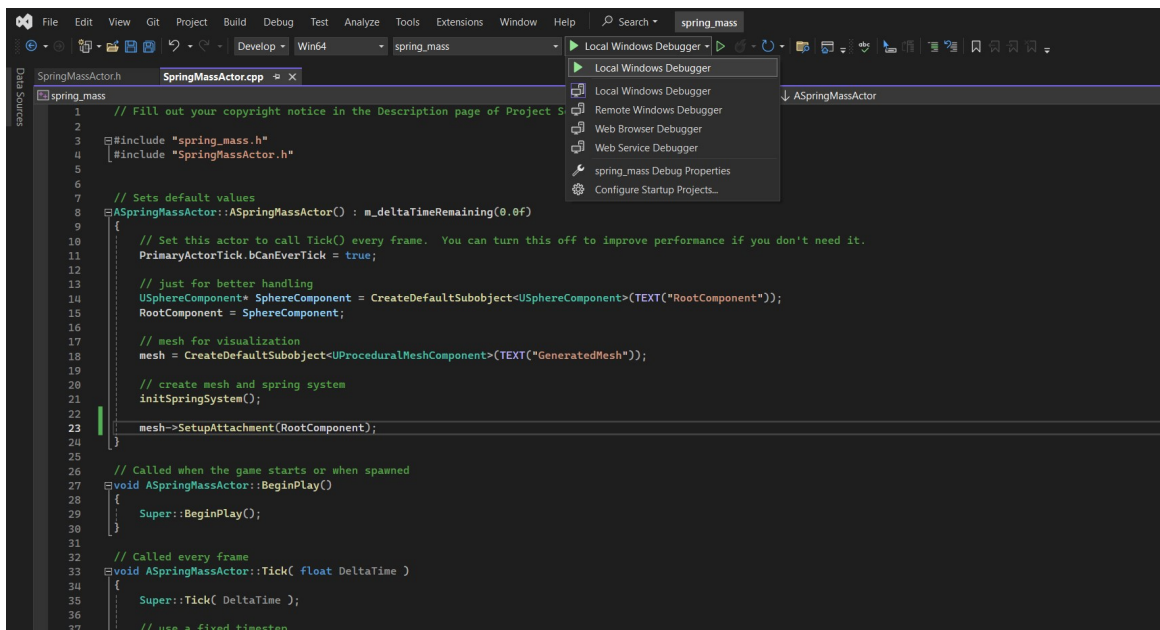


Figure 2: Cool people can hot reload changed code in the UE-Editor by starting the UE-Editor from Visual Studio.

Exercise 2 (Bonus: System stability, 2 Credits)

Explain why the system explodes for large damping factors.

² You can add a small, random vector offset to the gravitation vector.