Proposal

Prelude

The aim for the project is to simulate different fluids and visualize them in real-time. Simulation of various fluids can be done by e.g. modify the parameters of the Navier-Stokes equations and changing the rendering method.

Group 1

Niklas Andersson - nikan278@student.liu.se, Graphics Developer
Gabriel Baravdish - gabba873@student.liu.se, Math Implementation Developer
Joakim Deborg - joade361@student.liu.se, Core Developer
Kristofer Janukiewicz - krija286@student.liu.se, GUI Developer and Quality Assurance
Marcus Nygren - marny568@student.liu.se, Interaction Developer

Description of the physical system

The Navier-Stokes equations used for the fluid simulation are:

$$\frac{du}{dt} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + v\nabla^2\mathbf{u} + \mathbf{F}$$

$$\nabla \cdot \mathbf{u} = 0$$

The simplifications made to this model are that we consider the fluid to be incompressible and homogeneous. One additional property to be considered later is free surface modeling that will give a better result for water simulation. Information on the system and implementation can be found in Ref.1 and Ref.2

Simulation of the system

A pre-study will be done in Matlab to get a basic understanding for how the system works. The simulation will later be implemented in C++ with OpenGL and GLSL where the majority of calculations are performed on GPU. When simulating the fluids we will benefit heavily from the GPU being optimized for performing the same operations on different data in parallel.

Implementation of the animation

Since we aim to animate different kinds of fluids we will need to visualize different characteristics of each unique fluid. For example; during smoke simulation the temperature must be calculated at each point in the volume so that the appearance of hot smoke rising and cold smoke falling to the ground can be correctly modeled. In the case of simulating fire, it is required to keep track of how long it has been since the fuel was ignited. This provides the means to model how the fuel becomes exhausted. Things become a bit more complicated when dealing with water since we are no longer interested in visualizing the advection of density throughout the volume but rather the boundary between air and water. By first initializing each voxel with how far it is from the surface of water and then letting this property be carried throughout the simulation we know that the surface is where this value is exactly equal to zero.

Since the simulation will be calculated throughout a 3-dimensional volume of voxels, a volume rendering technique will be ideal for visualization purposes.The volume renderer will be created as a fragment-shader program in GLSL running on the GPU. We hope that a real-time rendering of the fluid simulation will allow the user to interact with the simulation by e.g. modifying the mathematical parameters or through direct physical interaction.Things become a bit more complicated when dealing with water since we are no longer interested in visualizing the advection of density throughout the volume but rather the boundary between air and water. By first initializing each voxel with how far it is from the surface of water and then letting this property be carried throughout the simulation we know that the surface is where this value is exactly equal to zero.

Documentation

All the documents created will be saved in Google Drive, whereas our code (including Tex code) is available on GitHub. We keep a detailed log to make sure we reach our deadlines in time and so that we do not work more or less than we have planned.

Plan of the work

Week 4: Decide on idea
Week 5: Proposal and research
- Deadline: 29/1, 9 AM: Submit proposal
Week 6: Implementation in MATLAB + C++ in parallel
Week 7: Implementation in MATLAB + C++ in parallel
- Deadline 13/2, 1-5 PM: Mid-term seminar
Week 8: Implementation in C++
Week 9: Implementation in C++
Week 10: Implementation in C++, report
Week 11: Report
- Deadline 14/3, 12 PM: Project deadline, programs and report
Week 12: Prepare for oral presentation
Week 13: Prepare for oral presentation
- Deadline: Oral presentation

References

- Ref.1: GPU Gems, Chapter 38, Nvidia, Mark J. Harris
- Ref.2: GPU Gems 3, Chapter 30, Nvidia, Keenan Crane, Ignacio Llamas, Sarah Tariq