Graphics and Simulation Research Journal

EMILJANO KURBIBA
Falmouth University
November 28, 2016

Introduction

In this journal, I will be exploring methods of simulation, water, fire & smoke with OpenGL and discussing how I can apply them to my projects. I will be explaining what they are and how I can use them in my projects.

Water Simulation

This section will be about different water simulations and how they are applied within openGL, obviously, there are many ways in implementing these so I will also discuss on how methods differ from each other and the challenges each one faces in development.

1 Real-Time Water Simulation

The process of rending water in real-time depends on how realistic the application needs to be according to the developer. Jagjeet Singh Dhaliwal wrote a report called REAL-TIME WATER SIMULATION [2] he breaks the complex simulation task down into three parts:

Surface Representation

This is the first thing a user notices when seeing a simulation, Jagjeet used 2D and 3D grids in order to fully represent a simulation however there are obvious differences between the two. 3D Grids give the developer more functionality and precise fluid movement but it inevitably more difficult to manage in the project. 2D Grids offer a more simplistic view of the simulation and easier functionality, so its functionality and capability is not a feasible as 3D but its simpler to maintain & represent in the project.

Height Field Generation

Janjeet used four methods of generating height-maps. Navier-Stokes Equation, this is the most realistic option but requires a lot of computation in order to achieve results thus is less feasible than other methods. 2D Wave Equation is the second method but this equation poses many problems when faced with sudden change within the simulation especially with real-time simulations. Coherent Noise using Perlin Noise, this is the method where when someone uses Perlin Noise to generate a noise texture that a developer is able to base a heightmap off. Fast Fourier Transform (FFT) this is where the simulation would generate the heightmap itself, it takes vales from a wave and generates a heightmap from it.

Reflection and Refraction

This part is done using pre-existing environment, methods of lighting and textures are used in order to create a reflection on the water itself.

SPH Based Shallow Water Simulation

I believe that Jagjeet problem was that his method had too much depth in the simulation as if it would have taken too much computational power to simulate that is why I have turned to a more efficient method according to a team in PhysX Research. Their method doesn't store surface heights using grid cells, they hide them with 2D SPH particles and generate the height according too the density of each particle location.[6]

2 Fire Simulation - Firestarter considerations

Unlike water fire is much more naturally chaotic. This proves difficult when trying to simulate, Marc de Kruijf's firestarter - A Real-Time Fire Simulator [4] he tries to achieve visualisation of fire and in doing so he made me realise the nature of quasi-random. Not only that but the process of fire is truly fascinating, his research goes as far to study shapes of a flame by calculating ratios of areas during the flame. This report did not give a perfect method in simulating fire but rather entertained me in the factors that developers should consider when attempting such a thing like atmosphere, motion of fire, cause of fire and types of fire you are trying to achieve. Truly fascinating that we can never fully achieve a perfectly natural flame in simulation.

3 Fire Simulation - Spring Mass Model

Murat Balci & Hassan Foroosh wrote a report on Real-time 3D Fire Simulation Using a Spring-Mass Model. [1] Visually represented the Spring Mass model can be pictured like a spring when pressure is applied in a direction it will repel in the opposite way, these guys are simulating fire using a 3D Visual representation of that model. The problem with this is that it does nothing for static stable fire however when adding wind force and atmosphere to the scenario that is where this model really shines because its only shows its potential with active fire with dynamics. Perhaps if this model was developed with considerations seen in Kruijf's Firestarter[4] it would have made for a very promising method.

4 Smoke Simulation

In Morgan McGuire's A Real-Time, Controllable Simulator for Plausible Smoke [5] report it explores the interaction between smoke and physical obstacles in a simulation. Challenges are presented when using three dimensional grids because the simulation is too slow and requires too much computational power. This essay dictates that the elements like turbulence, gravity, vortices, wind & drag should accommodate smoke and work with it in order for a better simulation.

In Tommy Hinks' Procedural Smoke Particle System with OpenGL 2.0[3] he explores the possibility of state preserving particle system that takes advantage of a GPU's power. He discusses that factors such as particle movement is achieved more realistically by adding 3D noise much like Jagjeet Singh Dhaliwal discussed in his water simulation model but this differs because it is applied to each particle. This sounds like a lot of computational power is required but Hinks claimed that the method can be entirely ran on the latest GPU's of which vague because he could mean production cards such as NVidia's Quadro series that range thousands of pounds or general consumer grade cards like NVidia's GTX series. This made me question the hardware limitations of exploring these fields & methods.

5 Conclusion

These simulations require so much research it's ridiculous! however it has become very clear to me that developers need to search for a balance between perfecting simulation and being able to run it without computational bottlenecks, this could be achieved by combining methods and/or applying methods by taking different approaches. It is very difficult to perfectly achieve phenomena's such as these in a computer simulation but Marc de Kruijf's Firestarter[4] truly taught me to consider other elements when trying to achieve these simulations.

References

1. M. Balci and H. Foroosh. Real-time 3d fire simulation using a spring-mass model. *2006 12th International Multi-Media Modelling Conference*.
2. Jagjeet S. Dhaliwal. Real-time water simulation. 2008.
3. Tommy Hinks. Procedural smoke particle system with opengl 2.0. Technical report, Linkoping, Sweeden.
4. Marc De Kruijf. Firestarter âĂŞ a real-time fire simulator. Technical report, Wisconsin.
5. Morgan Mcguire. A real-time, controllable simulator for plausible smoke. Technical report, Rhode Island, 2006.
6. Barbara Solenthaler, Peter Bucher, Nuttapong Chentanez, Matthias MÃijller, and Markus Gross. Workshop on virtual reality interaction and physical simulation vriphys (2011) sph based shallow water simulation. Technical report, ETH Zurich, 2011.