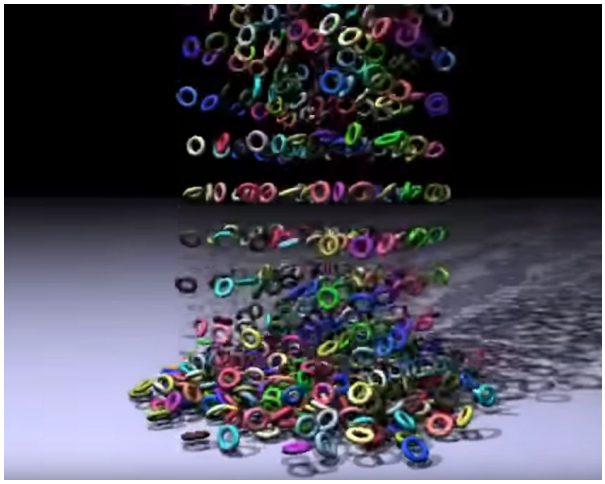
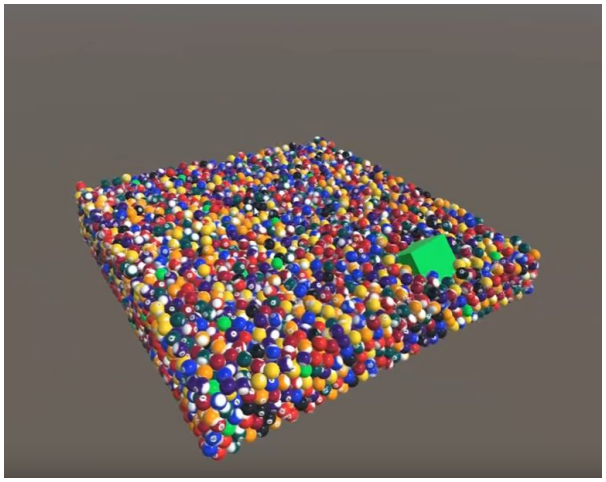

Overview

WebGL has taken off recently: it is easily accessible and runs with high performance. Unfortunately, its API (currently) only allows access to the graphics card through graphics-only functions, not making it easy to use the computational functionality of graphics cards. Real-time rigid body simulation, for example, would highly benefit from the parallel nature of GPU computation.

We propose a WebGL implementation of a rigid body simulation method which stores rigid body information such as linear velocity, rotational velocity, and forces in textures and uses shaders to perform physics calculations. In addition, this implementation represents rigid bodies as a set of particles, and computes collisions through interactions between these particles.

The technique detailed in the paper can also be generalized to particles with variable connectivity, allowing for substances like granular materials and fluids. These would be good stretch goals to pursue once we have a solid implementation of the core rigid body simulation in the browser. Since this project will be easily accessible through the browser, we will also be able to implement various interactive controls for the user to manipulate and interact with the scene.

Goals

- Implement a WebGL rigid body simulation
    - Rigid body collision detection and physics
    - Arbitrary mesh loading
    - Interactive controls
- Stretch goals
    - Granular materials
    - Fluids
    - Coupling simulations

References

GPU Gems 3 Chapter 29 http://http.developer.nvidia.com/GPUGems3/gpugems3_ch29.html