RESEARCH INTO THE SIMULATION OF SHOCK WAVES

Modelling A Shock Wave

Authors:
Liam Ireland, Michael Jones, Tomasz Mackow

Spheres
The first attempt we had at implementing a shock wave involved creating an expanding sphere that when it came into contact with moveable objects, it would exert a force on the object. However, early on we realised this implementation would not suffice as a sphere would not let the shock wave act in a realistic enough manner as it was not malleable enough. For example, it would have made diffraction and reflection difficult.

Particles
The most successful solution, and the one we have chose to pursue for the project, is the particle model. Particles offer us a way to expand the wave in a sphere-like manner and also give us the malleability needed to allow the shock wave to propagate around walls. Unity has objects called particle systems which we can also use to our advantage as they will allow us to emit particles from an origin in the way we desire. Each particle would represent a section of the wave front proportional to the amount of particles and applies an appropriate force depending on the number of particles and speed of the wave.

Meshes
Another solution which we considered was using meshes. This would have involved mapping out the shape of the wave front in a way that would have actively morphed as it hits objects and walls. While this solution seemed plausible it proved to be very difficult for us to implement as we didn't have an understanding of how to use and interact with the meshes offered to us in Unity. This was complicated further as we would have had to track changes in the size of the mesh in reflections and diffraction so that we could visually represent this using the mesh.

Smooth Particle Hydrodynamics
We were aware of smooth particle hydrodynamics as another candidate solution for our wave simulator however we chose not to implement this. Smooth particle hydrodynamics works by dividing a fluid, or in our case a wave front, and splitting it up into a set of individual components called particles. The particles are related in such a way that their properties, such as velocity or temperature, are obtained by summing the values of all the particles which are within a specified range. The influence of each particle in this range is weighted based on their distance from the selected particle as well as their density. A function known as a kernel function is used to make this more computationally efficient by excluding particles which are very far away thus offering little contribution to the value being calculated.