CM2303
Alex James | C1709318
Supervised by Bailin Deng
Moderated by Neetesh Saxena

Initial Plan: Realtime Liquid Simulation for 3D Environments

February 4th 2021

1 Project Description

Many strategies exist for simulating bodies of particles of varying states in two dimensions in real-time, but modern hardware is yet to make such methods extensible to three dimensions at a similarly fine resolution. While methods like hybrid grid-particle systems are available on the graphics industry's cutting edge, they are chiefly targeted at photorealistic fluid simulation for film visual effects rather than performative, interactive, real-time media. The field of real-time fluid simulation in three dimensions is not a novel one, and has been explored in tech demos over the past decades thoroughly, and yet there are so few commercial interactive products (i.e. video games) that take advantage of these developments; this in part due to the limitation that while in standalone demonstrations these methods are performative, creating complex simulations presented at acceptable frame rates, such performance is not present when the system is applied to a typical environment already pushing the available hardware to its limits.
I plan to explore several avenues of such simulation, with an objective to make a system that can create surface geometry for any fluid of any state that can then be applied to an arbitrary art style by providing information on the system's state to external programs (e.g. shaders). I will demonstrate the solution on an exemplary scene akin to what would be found in the games industry to illustrate the performance, reliability, and flexibility of the method. At least initially, the primary solutions explored will specifically involve hardware accelerated rendering using marching cubes ("meta surfaces") running on modern consumer graphics hardware, in the oft-seen commercial game engine *Unity3D*.

2 Project Aims and Objectives

1. **Fluid mechanics:** To understand the mathematics that drives Newtonian fluids.
    a. Create a bare-bones implementation of a fluid system and query using various resolutions, domain sizes, and solver iterations to statistically represent and understand the relationship between said parameters and the quality and performance of the simulation.
    b. Use these findings as a rough benchmark for the in-engine GPU solution, chiefly as a worst-case/lower boundary that the latter should easily exceed, in terms of performance.
2. **Particle simulation:** To be able to simulate a 3D body of particles that interact with one-another and their environment in a believable and performative way.
    a. Use the Navier-Stokes equations to solve a Newtonian fluid volume discretised into a finite set of particles that conserves mass, momentum, and is incompressible.
    b. Implement these methods in a parallelisable manner such that they can be executed by a GPU (via compute shaders), making use of flat-3D textures as a data structure to store the system's information.
    c. Investigate the practicality of separating these into a hybrid grid-particle method, treating the flow as a grid-based vector field and the concentration of individual particles as a series of points.
3. **Improve on the essential solution:** To extend on the basic fluid simulation to allow for a better ratio of apparent quality to performance.
    a. Investigate ways to streamline the implementation, such that aspects of the simulation are simplified or approximated, allowing for greater performance while losing true *material* accuracy but not apparent quality - stipulate the colloquial first law of computer graphics: "If it looks right, it is right."
    b. Parameterise these approximations such that in addition to being able to control the resolution and timestep etc. of the simulation, the degree of approximation can also be adjusted to fit the context (i.e. more approximation for more stylised, less realistic scenes).
    c. As well as accepting external forces into the system, it should also feed back out information on the forces acting at each point, to allow rigid bodies to move and be moved by the fluid. Enough abstracted information about these interactions should be present to allow a kinematic object (e.g. kinematic character controller commonly used in games) to respond to the system in whichever manner is desired.
4. **In-context practicality:** To evaluate the validity of transplanting the

implementation to a real world scenario.

a. Create several example scenes that mirror real-world examples of interactive media. Populate these scenes with both the fluid simulation and the typical approximations used in industry (subdivided planes animated via vertex/geometry shaders for large bodies, particle effects for smaller systems)

b. Compare both the performance and the aesthetics of the faux-fluid versions of the scenes and their simulated counterparts.

c. Procure feedback via contact with environment artists, graphics programmers, and other game developers to produce a consensus on the desirability of the solution.

3 Ethics

While the discrete mechanics of a fluid volume are objective, the aesthetics and perceived quality of their application are not; How detailed and resolute is enough? Should a stylized environment demand a less realistic, lower resolution fluid simulation? Does a fluid system modelled on reality provide enough detail by itself for an stylized art style, or would unrealistic additions be desired? In the event that answers to these questions are to be obtained from other developers and artists in the course of this project, Cardiff University's ethics policy must be considered and met - both the Research Integrity Online Training Programme and Ethical Approval Form must be completed and approved by the school's ethics department (*comsc-ethics@cardiff.ac.uk*), after which the collection of data from other people may be conducted.

4 Work Plan

(See Appendix fig. 1 for time allocation chart.)

4-1 Implementation

1. **Fluid Simulation Research:** Research into the Navier-Stokes equations, existing implementations of real-time fluid simulation, and GPU implementations. During this time, log the relevant theory in the final report as a background. Given the wealth of information widely available on the topics, this can likely be done in about a week.

2. **Compute Shader Research:** Research into compute shaders and processing on the GPU. A specific investigation into the concept of bringing non-graphical work onto the GPU to take advantage of its parallel computation capabilities, as well as the issues presented by this. The limitations of and changes that must be made to the processing of the fluid system are to be detailed in the final report also. Given that practical exercises will be carried out during this research, this could take around three weeks.

3. **Essential Implementation:** Implement a basic fluid simulation on the CPU without any optimisations. Once complete, run this system with a variety of parameters to create the benchmark against which the later implementations will be compared; collate this data in the final report. This could take as little as two weeks.

4. **GPU Implementation:** Move the simulation to the *Unity* engine and move the processing to the GPU (Compute shader); implement a means of visualising the simulation in 3D space (meta-balls/marching cubes) that is separable from the simulation itself and is extensible to allow for more complex and stylised variations on the presentation. The time frame for this is more flexible as it's core to the project, but I would estimate that this will occupy about a month.

5. **Streamlining & Parameterisation:** Implement optimisations and approximations to improve the performance relative to the apparent quality, parameterise these as well as expose the properties and state of the system. Parameterise the visualisation also. Given the more loosely defined structure and extent of this phase, this will take up a variable amount of time and can be actively revisited while later stages of the project are being worked on.

6. **Example Applications:** Create several environments that demonstrate the system and extend on it to fit the requirements and art styles present. This is akin to an end-user applying the solution to their own project, and will serve to illustrate how the system can be tweaked and built on to create bespoke results.

4-2 Final Report

1. **Introduction:** A cursory introduction containing the abstract and terminology of the project.

2. **Background:** Cover the background information on contemporary fluid simulation, a curt history of fluid simulation and its visualisation, as well as an overview of how the GPU has previously been used to handle computationally expensive simulations and tasks. This should be completed in a relatively small and predictable time frame.

3. **Essential Implementation & Results:** Detail the process of creating a simple fluid simulation on the CPU and record the frame-times for a variety of settings for this implementation. Again, this is a more straightforward task and should only take as long as the task of creating the basic implementation.

4. **Final Implementation:** By far the least predictable and likely lengthiest of the stages, this cannot be completed until the system has been implemented in the game engine and in compute shaders, which may not only be of a less foreseeable length but is also much more core to the project as a whole than other stages, and therefore is allocated a month for completion.

5. **Optimisations:** While core to the project's novelty, the optimisations to the simulation and visualisation of the system are not strictly defined, and so will occupy as much or as little time as they realistically can. The write-up of said optimisations is tied to their completion, and subsequent chapters of the report may be started while this is only partly complete.

6. **Performance Results:** A comparison of the final implementation's performance, with an awareness of the essential implementation's results, and a record of what impact the optimisations have on this. This can only be completed when all previous implementation stages have been finished.

7. **Example Applications:** A showcase of the example environments and how the system fits each one. This must proceed the implementations, parameterisation, and optimisations, but will likely occupy only a week for the final report.

8. **Hypothetical Extensions:** A short, week-long section briefly covering the potential extensions and changes that could be made to the system.

9. **Summary:** A coverage of the conclusions that can be drawn from the results and the process of researching and implementing the system, which can only realistically be done once the entirety of the preceding sections has been completed, but will take less than a week.

10. **Reflection:** Like the summary, this section mandates that previous sections are completed, and covers the knowledge ascertained from the project as a whole, occupying the final week of the project.

5 Appendix

Figure 1. Gantt Chart