

# Game Dev Project proposal

*by* Kallem Thomas

---

**Submission date:** 19-Oct-2025 11:24PM (UTC+0100)

**Submission ID:** 265703438

**File name:** 49116\_Kallem\_Thomas\_Game\_Dev\_Project\_proposal\_695044\_305449948.docx (28.96K)

**Word count:** 1781

**Character count:** 10522

# Fluid Dynamics Within Game Development

## Research Proposal

Name: Kallem Thomas  
Preferred supervisor(s): Nathan Morgan or Tim Bashford  
Word Count: 1,254

### Contents

|                                             |   |
|---------------------------------------------|---|
| 1. Introduction .....                       | 2 |
| 2. Background to the Proposed Research..... | 2 |
| 3. Research Problem statement .....         | 3 |
| 4. Research Aim & Objectives .....          | 4 |
| Aim:.....                                   | 4 |
| Objectives: .....                           | 4 |
| 5. Methodology.....                         | 4 |
| 5.1 Methods.....                            | 4 |
| 5.2 Materials .....                         | 4 |
| 6. Project Plan .....                       | 5 |
| 7. References .....                         | 5 |

## 1. Introduction

As I was playing various games, I found myself wondering what made certain ones feel so calming, beautiful, and enjoyable. Even when their graphics weren't particularly advanced. While searching for the answer, it began  over a lake in one of the games, and in that moment, I realized what it was: water. Lakes, oceans, rainfall. The presence of water has a natural way of bringing life and serenity to a virtual world. There's something inherently soothing about watching ripples spread across a lake, waves crash gently against the shore, or raindrops distort reflections on a puddle. These small details often hold more emotional impact than high-end textures or photorealistic lighting. Yet, despite water's beauty, I noticed that in most games, it doesn't truly behave like water should. The physics are often simplified, static, or unrealistic. For instance, in Minecraft, water flows for only eight blocks before coming to a stop. That realization became the foundation of my motivation: **to create a game where water feels alive**. Where it moves, reacts, and behaves realistically, whether it's contained in a glass or part of a vast natural landscape 

 al studies demonstrate that advanced simulation techniques can produce high visual realism and interactivity in water environments. For example, "Interactive fluid flow simulation in computer graphics using incompressible smoothed particle hydrodynamics" shows how ISPH models allow realistic free-surface flows, including fluid-structure interaction, with detailed visual effects like splashes. [1] 

## 2. Background to the Proposed Research

 r has always played a crucial role in how we perceive realism and atmosphere in digital worlds. In game development, it often becomes one of the most memorable visual and emotional elements, capable of transforming even simple graphics into something calming and believable. However, creating realistic water is one of the most technically challenging goals in computer graphics. Real-world water behaviour is governed by the Navier-Stokes equations, which describe how fluids move and interact. While these equations can produce highly accurate results in scientific simulations, they are far too computationally demanding to be used directly in real-time games [2].

Early breakthroughs in computer graphics came from simplifying these equations into manageable models. One of the most influential examples is Jos Stam's Stable Fluids method [2], which introduced a semi-Lagrangian, grid-based solver that could simulate fluids without becoming numerically unstable.

Because of these challenges, researchers began to look at particle-based methods. Smoothed Particle Hydrodynamics (SPH), originally developed for astrophysics, turned out to be an effective way to model water in computer graphics [4]. Instead of relying on a grid, SPH represents a fluid as a collection of particles that interact with each other. This makes it easier to simulate realistic motion, splashes, and other free-surface behaviours. Müller et al. [4] were among the first to bring SPH into interactive environments, showing that even complex liquid motion could be simulated in real time. Later, Solenthaler and Pajarola [3] improved the method with the Predictive-Corrective Incompressible SPH (PCISPH) approach, which enhanced stability and made simulations more efficient. Ihmsen et al. [1] built on this work with the Incompressible SPH (ISPH) model, which allowed even more accurate pressure control and realistic behaviour in free-surface flows.

As the field progressed, hybrid techniques started to appear. These combined the strengths of both particle and grid-based systems. Losasso et al. [7] and Chentanez and Müller [9] proposed methods that allowed fluids to interact more naturally with environments, while Kim et al. [10] introduced wavelet turbulence to simulate small surface details without requiring heavy computation. Such

methods began closing the gap between physically accurate simulations and what could run in a real-time engine.

Visual rendering techniques also advanced alongside these physical models. Instead of focusing purely on physics, some researchers explored how to make water look right. Bharathi and Ganesh [8] developed a screen-space approach to approximate reflections, refractions, and caustics, allowing realistic water surfaces at minimal cost. Yuksel et al. [11] worked on height-field simulations that could represent vast ocean scenes efficiently. These techniques demonstrated that visual realism could often be achieved through clever rendering rather than pure physics.

Modern game engines such as Unreal Engine and Unity now integrate some of these methods. Unreal Water Plugin [13], for instance, uses height-field displacement to create oceans and lakes, while NVIDIA's Flex [12] relies on GPU-accelerated particle physics to simulate fluids, cloth, and soft bodies. However, both still rely on simplified physics to maintain high frame rates on consumer hardware. This highlights a persistent gap between academic research—which focuses on accuracy—and game development, which prioritizes performance and playability.

Recent work has turned to GPU computing to bridge that gap. Macklin and Müller [6] introduced Position-Based Fluids (PBF), which reformulates fluid equations into constraints that can be solved efficiently on GPUs. This allowed tens of thousands of fluid particles to be simulated in real time, greatly improving visual and physical realism. Goswami and Günther [14] further explored hybrid GPU techniques for interactive use, showing that near-realistic results could be achieved even on mid-range systems.

Despite these advances, most games still treat water as a visual effect rather than a truly interactive element. In titles like Minecraft, water flows only a short distance before stopping, and in The Amazing Spider-Man, it acts as a non-interactive boundary. Bridson [5] notes that the challenge is not only computational but also perceptual: players expect water to behave realistically, but only up to a certain level of detail.



This creates a clear research gap: current methods either offer physical realism that's too costly for games, or lightweight systems that fail to behave convincingly. The focus of this project is therefore to explore a middle ground: developing a lightweight, GPU-optimized SPH-based water system inspired by ISPH [1] and visualized using efficient screen-space rendering [8] [13].

### 3. Research Problem statement

Real-time games often rely on simplified, non-physical water systems that lack realistic behaviour and interactivity. Current methods either prioritize visual fidelity over fluid accuracy or require computational resources beyond the scope of typical game engines. The problem this project addresses is the lack of an efficient, physically based water simulation method that balances realism, responsiveness, and performance for interactive applications.

## 4. Research Aim & Objectives



### Aim:

To develop and evaluate a real-time water simulation system using Smoothed Particle Hydrodynamics (SPH) techniques that achieves physically realistic motion and visual fidelity suitable for game development.

### Objectives:

1. To review and compare existing real-time fluid simulation methods [1]–[10].
2. To design a lightweight SPH-based water simulation prototype optimized for performance in Unity or Unreal Engine.
3. To integrate physically accurate behaviours such as viscosity, buoyancy, and surface tension [5], [6].
4. Enhance visual realism using shader-based rendering techniques [8], [13].
5. To evaluate the simulation's performance, visual quality, and user immersion through testing and feedback.

## 5. Methodology

### 5.1 Methods

A quantitative, experimental research method will be used, focusing on software development and evaluation. The project will employ an engineering research strategy, using iterative prototyping and performance testing to assess computational efficiency and visual realism.

The simulation will be implemented using Smoothed Particle Hydrodynamics (SPH) equations adapted for real-time computation [1], [3], [4], utilizing  U acceleration where possible [6], [12]. Parameters such as particle count, viscosity, and timestep will be varied to analyse their effects on realism and performance.

Data will be collected via frame rate benchmarking, memory usage profiling, and subjective visual evaluation (user feedback on realism). Quantitative metrics (e.g., FPS, computational time) will be analysed statistically to assess optimization efficiency [5].



### 5.2 Materials

1. **Software Tools:** Unity or Unreal Engine 5 [13], C#, HLSL or GLSL shaders, CUDA (optional) [12], Blender for asset creation, and MATLAB or Python for data analysis.
2. **Hardware:** Mid-to-high-end GPU (e.g., NVIDIA RTX 3060 or equivalent), 16GB+ RAM.
3. **Data:** Simulation parameters, benchmark results, and user evaluation responses.
4. **Ethics:** No personal or sensitive data will be collected; all user testing will ensure informed consent.

## 6. Project Plan



| Phase (Week) | Tasks                                                          | Duration (Weeks) |
|--------------|----------------------------------------------------------------|------------------|
| 1-2          | Literature review and reference collection                     | 2                |
| 3-5          | Design simulation architecture (SPH Model, rendering pipeline) | 3                |
| 6-8          | Prototype Implementation in Unreal                             | 3                |
| 9-10         | Performance optimization and gpu integration                   | 2                |
| 11-12        | User evaluation and data collection                            | 2                |
| 13           | Data analysis and refinement                                   | 1                |
| 14           | Write-up and presentation preparation                          | 1                |



## 7. References

- [1] M. Hassaballah, A. Abdelnaim, "Interactive fluid flow simulation in computer graphics using incompressible smoothed particle hydrodynamics," *Computer Animation and Virtual Worlds*, <https://onlinelibrary.wiley.com/doi/10.1002/cav.1916?utm>, Accessed: 10/10/2025
- [2] J. Stam, "Stable fluids," *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*, pp. 121–128, 1999.
- [3] B. Solenthaler and R. Pajarola, "Predictive-corrective incompressible SPH," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 1–6, 2009.
- [4] M. Müller, D. Charypar, and M. Gross, "Particle-based fluid simulation for interactive applications," *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 154–159, 2003.
- [5] R. Bridson, *Fluid Simulation for Computer Graphics*, 2nd ed., Boca Raton, FL, USA: CRC Press, 2015.
- [6] M. Macklin and M. Müller, "Position based fluids," *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 1–12, 2013.
- [7] F. Losasso, R. Fedkiw, and R. Bridson, "Two-way coupled SPH and grid-based fluids," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 4, pp. 797–804, 2008.
- [8] P. Bharathi and S. Ganesh, "Screen-space fluid rendering for interactive environments," *Computer Graphics & Applications Journal*, vol. 42, no. 3, pp. 56–67, 2022.
- [9] N. Chentanez and M. Müller, "Real-time simulation of large bodies of water with small-scale details," *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 197–206, 2010.
- [10] B. Kim, Y. Liu, I. Llamas, and J. Rossignac, "Wavelet turbulence for fluid simulation," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 1–6, 2008.
- [11] C. Yuksel, M. van de Panne, and T. Turk, "Efficient simulation of surface waves using height fields," *The Visual Computer*, vol. 29, no. 1, pp. 43–53, 2013.
- [12] NVIDIA, "NVIDIA Flex: Unified particle physics for real-time applications," NVIDIA Developer

Documentation, 2021. [Online]. Available: <https://developer.nvidia.com/flex>

[13] Epic Games, "Water system plugin overview," Unreal Engine Documentation, 2023. [Online]. Available: <https://docs.unrealengine.com>

[14] A. Goswami and T. Günther, "Hybrid particle-grid methods for real-time fluid simulation," Journal of Computer Animation and Virtual Worlds, vol. 31, no. 6, pp. e1950, 2020.

[15] K. Erleben, J. Sporring, and K. Henriksen, Physics-Based Animation, Natick, MA, USA: A K Peters, 2005.

# Game Dev Project proposal

---

## GRADEMARK REPORT

---

FINAL GRADE

GENERAL COMMENTS

63/100

### Introduction/Aims & Objectives 12/20

Make objectives SMART with measurable targets

### Background/Literature Review

16/20

Excellent literature engagement (15 references, mostly journals)

Clear research gap identified and articulated well

### Project Design 15/30

Appropriate methodology selection (SPH is suitable)

### Resources/Planning 10/15

Realistic project planning (14 weeks is achievable)  
Logical task progression in timeline

### Document Presentation 10/15

The project is feasible, interesting, and has clear practical application. The timeline is realistic and the literature foundation is strong.

Will the final artifact be in the shape of a demo scene or a tool for achieving the result?

---

PAGE 1

---

PAGE 2

---



#### Comment 1

overly anecdotal and informal for an academic proposal



#### Comment 2

This opening is too informal and anecdotal for an academic proposal. Whilst personal motivation is valuable, academic writing requires a more professional tone.



### Comment 3

Good early reference to literature showing you've engaged with research. However, this sentence feels disconnected from the paragraph before it. Consider integrating it more smoothly or moving it to the Background section.



### Comment 4

put the reference before the .



### Comment 5

Excellent opening to the background section. You clearly establish the technical challenge and reference fundamental equations. This is much more academic in tone than the introduction.

PAGE 3

---



### Comment 6

What constitutes "lightweight"? What performance targets? What level of visual fidelity? Define your "middle ground" quantitatively.

PAGE 4

---



### Comment 7

This was a bit weak, try to align with smart objectives.



### Comment 8

z



### Comment 9

zzz



### Comment 10

Will you be using test users?

PAGE 5

---



### Comment 11

This is certainly functional, but lacking any visual sugar



### Comment 12

zzzzz

