# COMP4906
# Honours Thesis Pre-Proposal
# Real-Time 3D Volumetric Fire Simulation For Games

Gabriel Racz

September 6, 2024

# 1 Motivation

In recent years, rapid improvements in consumer graphics hardware has allowed game developers to create more detailed worlds through new rendering and simulation techniques. AAA studios have begun leveraging GPU compute to simulate game systems such as volumetric light scattering [2], and interactive wind simulations [5]. One area that hasn't seen rapid adoption by industry is the simulation and rendering of interactive volumetric fire. Although interactive fire has seen an increase of representation in new titles such as Read Dead Redemption 2, The Legend of Zelda: Breath of the Wild, and Far Cry 5, these techniques largely rely on simplistic spreading interactions and either sprite-based particle rendering.

2018's God of War utilized a basic volumetric wind fluid simulation to allow to interact with vegetation, cloth, and hair [5]. I believe there is potential to push this idea further and incorporate a more detailed fluid simulation that handles fire, boundary conditions with game objects, and volumetric lighting.

# 2 Objective

The objective of this thesis is to research methods that when combined will result in a efficient GPU-based volumetric fire and fluid simulation for the use in games on current generation hardware. This will require working within memory and performance constraints such that it can be run in real-time, while still budgeting performance for other gameplay systems. To achieve pleasant visuals and high performance, a balance of physically based models and artistic parameters will be necessary.

As the field of fire simulation spans many decades, thorough literature review will be conducted in order to find areas for improvement and novelty. Firstly, I will review and implement some of the standard techniques used in past literature for real-time fluid dynamics such as the famous Stable Fluids [8], as well as other widely referenced works on the simulation and rendering of fire [6] [4]. I intend to develop a rendering and simulation engine using the Vulkan graphics API to facilitate GPGPU implementation of the various algorithms involved in the thesis. I opt for a custom engine to allow for complete control of the graphics hardware in order to achieve my goal of simulating and rendering the fluids at 60 fps or higher. The overarching constraint of the thesis will be developing a technique that can realistically be used in modern games, meaning the total frame budget for both the simulation and rendering passes should at minimum be below 16ms.

Below are some areas I have identified as potentially providing interesting extensions to existing methods.

- Adaptive/hierarchical grid simulation domain. In the majority of past research uniform and static voxel grids are used to simulate the fluid volume. Optimizing the cubic memory and runtime requirements of uniform grids could offer large performance benefits in terms of physics integration speed, level of detail over sparce game worlds, physical interactions with other game objects, and volumetric lighting. I will investigate "Multigrid integration" [7], dynamic octrees/bounding volume hierarchies, simulation LODs, and selectively budgeting higher resolution simulation/rendering to areas of higher detail.

- Simulation grid uspcaling utilizing noise and other non-physical techniques to fill in gaps between grid cells. This would offer a method to induce turbulence and other chaotic phenomena that are observed in real fire while saving simulation integration cost. Curl Noise [3], Wavelet Turbulence [1], or other procedural noise methods should be investigated.

- Rendering methods for volumetric lighting and interactions between flames, smoke, and game geometry. I will research techniques for rendering the flames themselves, including level set shells [6], ray marching/tracing, or other procedural texturing solutions. Investigate methods for representing volumetric light and shadow within the smoke volume using the existing fluid grid.

- Interaction with game geometry. Using game geometry to represent fuel and allowing for the dynamic spreading of flames through a scene. Handle boundary conditions between the fluids and game geometry. Allow for the player to influence the fluid volume through their actions.

## 3 Tentative Schedule

| **Period** | **Milestones** |
|---|---|
| September 9 - September 22 | Initial literature review. Research Stable Fluids [8] and other well established methods for real-time fluid simulation suitable for games. |
| September 23 - October 6 | Begin developing the Vulkan engine with the goal of reimplementing the techniques within [8] and [6] using a GPGPU pipeline. |
| October 7 - October 20 | Finish working prototype of 3D fluid simulation, measure performance and identify initial bottlenecks worth further research. Begin implementing raster-based pipeline for mesh geometry (and continue whenever time is available). |
| October 21 - November 3 | Add combustion to the fluid model (heat transfer, reactants, smoke). In parallel continue researching modern methods for rendering and simulating flames. Identify which physical models are necessary for realistic visuals. |
| November 4 - November 17 | Use gathered experience to inform areas of future study. Research methods for adaptive grids. Experiment with the addition of noise and/or upscaling. Submit rough draft of full proposal. |
| November 18 - December 1 | Use feedback from proposal rough draft to further direct experiments and literature review. Commit to thesis content. |
| December 2 - December 15 | Finalize and submit thesis proposal. Time permitting, continue more focused reading of papers and engine improvements. |

## References

[1] Fredrik Albers et al. "Physically-Based Animation of Fire for Android". In: (2020).

[2] Fabian Bauer. *Creating the Atmospheric World of Red Dead Redemption 2*. Aug. 2021. URL: `https://www.youtube.com/watch?v=9-HTvoBi0Iw&t=7100s`.

[3] Robert Bridson, Jim Houriham, and Marcus Nordenstam. "Curl-noise for procedural fluid flow". In: *ACM Transactions on Graphics (ToG)* 26.3 (2007), 46–es.

[4] Keenan Crane. *Chapter 30. real-time simulation and rendering of 3D fluids*. URL: `https://developer.nvidia.com/gpugems/gpugems3/part-v-physics-simulation/chapter-30-real-time-simulation-and-rendering-3d-fluids`.

[5] Sean Feeley. *Interactive Wind and Vegetation in "God of War"*. Jan. 2022. URL: `https://www.youtube.com/watch?v=MKX45_riWQA`.

[6] Duc Quang Nguyen, Ronald Fedkiw, and Henrik Wann Jensen. "Physically based modeling and animation of fire". In: *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 2002, pp. 721–728.

[7] Wouter Raateland et al. "DCGrid: An Adaptive Grid Structure for Memory-Constrained Fluid Simulation on the GPU". In: *Proc. ACM Comput. Graph. Interact. Tech.* 5.1 (May 2022). DOI: `10.1145/3522608`. URL: `https://doi.org/10.1145/3522608`.

[8] Jos Stam. "Stable fluids". In: *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 2023, pp. 779–786.