Project Proposal
Title: Position Based Dynamics Fluid Simulator
Name: Honglin Cao
Student ID: 20882680
User ID: h45cao

Final Project:

Purpose :

The purpose of this project is to develop a comprehensive and versatile position-based dynamics fluid simulator. The goal is to create a robust framework capable of accurately simulating fluid dynamics in real-time. By integrating Smoothed Particle Hydrodynamics (SPH) within a Position-Based Dynamics (PBD) framework and utilizing OpenGL for rendering, the simulator aims to achieve high fidelity in modeling fluid behavior. This project intends to push the boundaries of real-time fluid simulations, providing a powerful tool for applications in computer graphics, virtual reality, and scientific research. The resulting simulator will offer a blend of precision, performance, and flexibility, enabling detailed and interactive simulations of complex fluid systems.

Statement :

Paragraph: What it's about.
This project focuses on developing a Position-Based Dynamics (PBD) simulator to simulate fluid dynamics. The objective is to create a framework that can accurately model the interactions and behaviors of fluids in real-time, providing a powerful tool for applications in computer graphics, virtual reality, and scientific research.

Paragraph: What to do.
The project involves several key tasks: developing a flexible and efficient PBD framework, implementing SPH algorithms for fluid simulation, and modifying the rendering engine to use OpenGL for real-time simulation. Each of these tasks requires the design and implementation of specific data structures and algorithms to handle the complexities of fluid simulations and rendering.

Paragraph: Why it is interesting and challenging.
This project is both interesting and challenging due to the complexities involved in accurately simulating fluid dynamics and rendering them in real-time. Fluids have unique physical properties and behaviors, making their simulation non-trivial. Integrating SPH within a PBD framework and using OpenGL for rendering adds another layer of complexity, requiring advanced algorithms and optimization techniques to ensure stability, realism, and performance. The challenge lies in balancing these aspects to create a versatile and robust simulator.

Paragraph: What I will learn.
Through this project, I will gain a deep understanding of the principles and techniques behind position-based dynamics and smoothed particle hydrodynamics. I will learn how to implement and optimize algorithms for real-time fluid simulations and how to leverage OpenGL for efficient rendering. Additionally, I will develop skills in designing efficient data structures, solving constraint-based problems, and integrating various simulation components into a cohesive framework. This project will also enhance my problem-solving abilities and provide practical experience in tackling advanced topics in computer graphics and simulation.

Technical Outline :

Develop a Position-Based Dynamics (PBD) Framework

- Create a general-purpose PBD framework for simulating fluids.
- Support various constraints such as distance and volume preservation.
- Optimize the framework for performance and flexibility.

Important Data Structures and Algorithms:

- **Particle System Data Structures**
  - **Particles**: Store properties such as position, velocity, mass, and others.
  - **Grids/Spatial Hashing**: Facilitate efficient neighborhood search and interaction calculations.
  - **Constraints**: Represent physical constraints, such as distance and volume preservation.

Implement Smoothed Particle Hydrodynamics (SPH) for Fluid Simulation

- Develop SPH algorithms to simulate realistic fluid behavior.
- Implement kernel functions such as Poly6 and Spiky for SPH.
- Calculate density and pressure using SPH formulations.
- Solve the Navier-Stokes equations for particle-based fluids.
- Ensure stability and minimize numerical dissipation in the fluid simulation.

Important Data Structures and Algorithms:

- **SPH Algorithms**
  - **Kernel Functions**: Define the influence of neighboring particles (e.g., Poly6, Spiky).
  - **Density and Pressure Calculations**: Compute densities and pressures based on neighboring particles.
  - **Force Calculations**: Determine forces such as pressure and viscosity acting on particles.
  - **Time Integration**: Update particle positions and velocities using methods such as Leapfrog or Verlet integration.

Modify Rendering Engine to Use OpenGL

- Integrate OpenGL for real-time rendering of fluid simulations.
- Implement efficient data transfer between CPU and GPU.
- Develop shaders for visualizing fluid properties such as density and velocity.
- Optimize rendering pipeline for real-time performance.

Important Data Structures and Algorithms:

- **Rendering Algorithms**
  - **Shader Programs**: Implement vertex and fragment shaders for rendering particles.
  - **Data Transfer Mechanisms**: Optimize data transfer between CPU and GPU using techniques such as Vertex Buffer Objects (VBOs) and Framebuffer Objects (FBOs).
  - **Visualization Techniques**: Develop methods to visualize fluid properties like density and velocity fields.

References

[1] Müller, Matthias, et al. ”Position based dynamics.” *Journal of Visual Communication and Image Representation* 18.2 (2007): 109-118.

[2] Müller, Matthias, et al. ”Real time physics: class notes.” *ACM SIGGRAPH 2008 classes*. ACM, 2008.

[3] Macklin, Miles, et al. ”Unified particle physics for real-time applications.” *ACM Transactions on Graphics (TOG)* 33.4 (2014): 153.

[4] Müller, Matthias, David Charypar, and Markus Gross. ”Particle-based fluid simulation for interactive applications.” *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003.

[5] Ericson, Christer. *Real-time collision detection*. CRC Press, 2004.

[6] Macklin, Miles, and Matthias Müller. ”Position based fluids.” *ACM Transactions on Graphics (TOG)* 32.4 (2013): 104.

Objectives:

UWaterloo UserID: h45cao **Student ID:20882580**

1 Objectives

___ 1: Develop a flexible and efficient Position-Based Dynamics (PBD) framework.

- Create a general-purpose PBD framework for simulating fluids.
- Support various constraints such as distance and volume preservation.
- Optimize the framework for performance and flexibility.

___ 2: Implement Smoothed Particle Hydrodynamics (SPH) for fluid simulation.

- Develop SPH algorithms to simulate realistic fluid behavior.
- Implement kernel functions such as Poly6 and Spiky for SPH.
- Calculate density and pressure using SPH formulations.
- Solve the Navier-Stokes equations for particle-based fluids.
- Ensure stability and minimize numerical dissipation in the fluid simulation.

___ 3: Modify the rendering engine to use OpenGL.

- Integrate OpenGL for real-time rendering of fluid simulations.
- Implement efficient data transfer between CPU and GPU.
- Develop shaders for visualizing fluid properties such as density and velocity.
- Optimize the rendering pipeline for real-time performance.

___ 4: Optimize data structures for efficient simulation.

- Utilize grids or spatial hashing to facilitate efficient neighborhood search and interaction calculations.
- Design data structures to handle particles, constraints, and interactions efficiently.

___ 5: Ensure stability and performance optimization.

- Implement robust time integration methods such as Leapfrog or Verlet integration.
- Optimize the overall system to maintain stability and high performance in real-time simulations.