Graphics Final Project Write Up

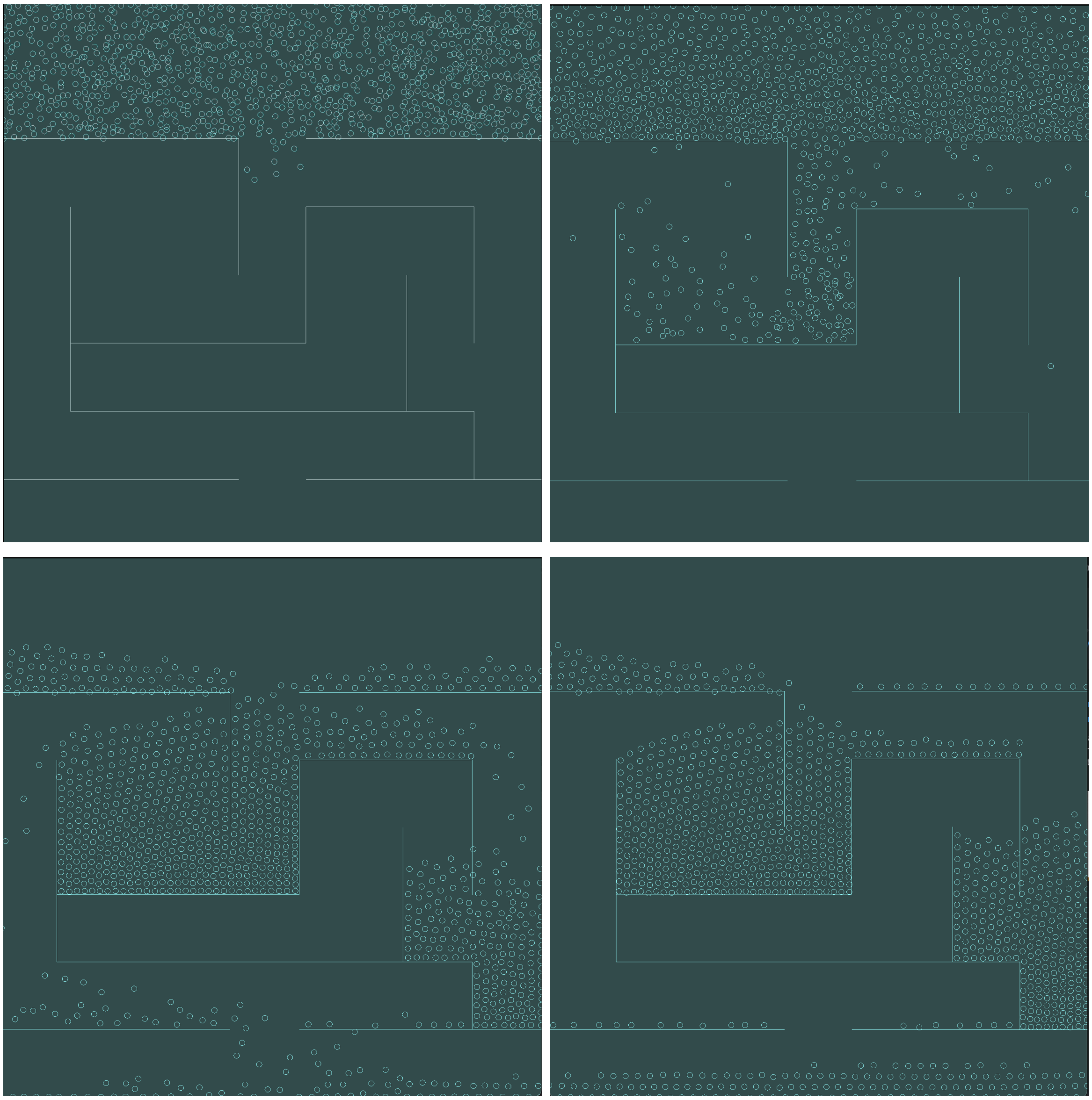
Team Members: Sahaj Bhakta (Sab6427), Vali Khan (vik84)

In our project, we aimed to develop a 2D fluid simulation using OpenGL with the ability for the fluid to solve a maze. Our primary objective was to dive into the complexities of fluid dynamics; however, we wanted to add something a little more innovative to our project. Inspired by the versatility of fluid mechanics, we devised a novel approach by integrating maze-solving elements into our simulation. Through this maze-solving component, we sought to showcase the adaptability of fluid dynamics in navigating complex environments, thereby highlighting its broader implications beyond theoretical understanding.

We split up the project into two main parts, one was the fluid dynamics of the individual particles interactions and the other was the maze solving component consisting of how the particles interacted with the walls of the maze. In the fluid dynamics aspect of our project, the behavior of individual particles was crucial in achieving a realistic simulation of fluid flow. We implemented a ParticleManager class to handle the management and interactions of these particles within a 2D space. Each particle possessed attributes such as position, velocity, and density, which were updated over time according to physical principles. Through methods like precomputeDensities and preasureForce, the particles dynamically responded to their surroundings, adjusting their velocities based on pressure gradients and densities. To accomplish these we took advantage of the Navier-Stokes equations to serve as the cornerstone for describing the motion of fluids. These partial differential equations govern the behavior of fluid flow by accounting for factors such as viscosity, pressure, and acceleration. We incorporated both the pressure and the acceleration, however, struggled with the viscosity so we were not able to fully apply it. Other than the Navier Stokes equations, another major part of our project was the smoothing Kernal function. This function plays a crucial role in modeling the interaction between particles within the fluid. This function is based on a smoothing kernel, a fundamental concept in computational fluid dynamics used to compute the influence of neighboring particles on a given particle.

In the maze-solving portion of our project, the focus shifted to how particles interacted with the maze walls. We implemented a Wall class to represent the walls of the maze, defining them by their starting and ending coordinates. When a particle approaches a wall, we calculate its distance from the wall and determine if it's within a certain radius. If so, we compute the normal vector of the wall and check if the particle is moving towards it. If the particle is moving towards the wall, we reflect its velocity to simulate a bounce effect, ensuring particles adhere to maze boundaries.

Below we have some sample images of our program running. You can see in the first image that all the particles start from the top, then as the simulation progresses the particles fall into the maze and eventually solve the maze. The last image shows all the particles settled with the simulation coming to an end.



References

- Breen, D, et al. Particle-Based Fluid Simulation for Interactive Applications. 2003. Accessed from: https://matthias-research.github.io/pages/publications/sca03.pdf
- Clavet, Simon, et al. *Particle-Based Viscoelastic Fluid Simulation*. Accessed from: http://www.ligum.umontreal.ca/Clavet-2005-PVFS/pvfs.pdf
- Koschier, D. Bender, J. Solenthaler B. Teschner M. Smoothed Particle Hydrodynamics
- Techniques for the Physics Based Simulation of Fluids and Solids Accessed from: https://sph-tutorial.physics-simulation.org/pdf/SPH_Tutorial.pdf