

# **Real-time Particle-based Snow Simulation with Vulkan**

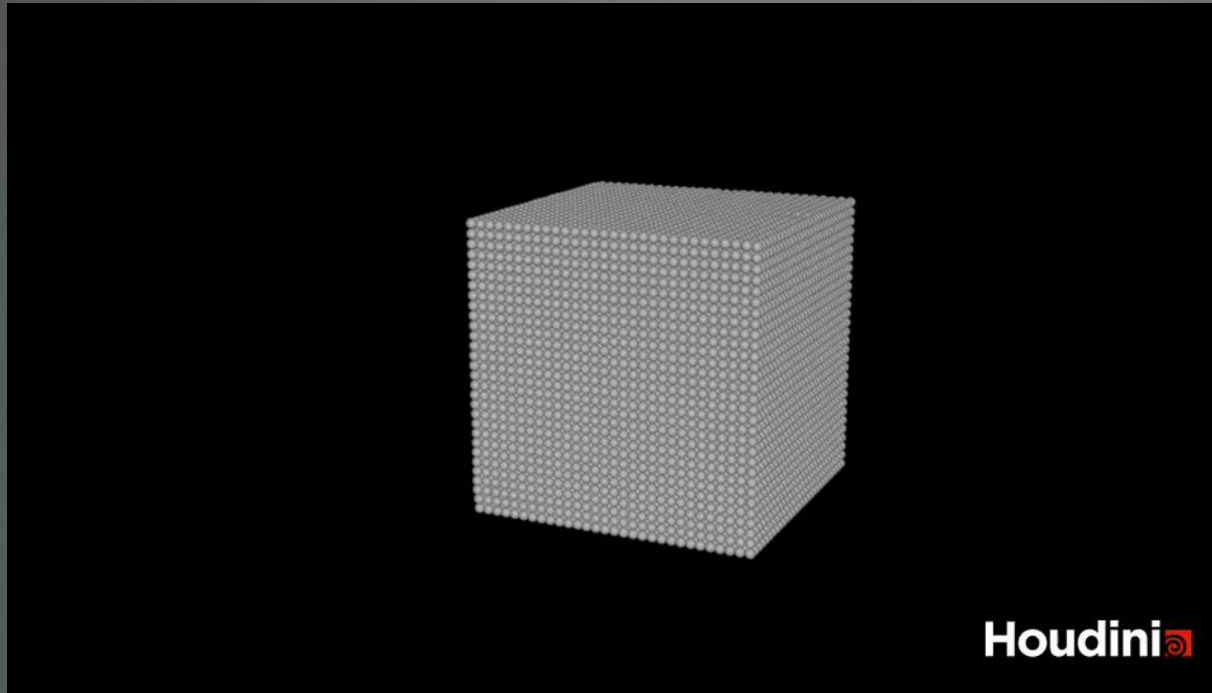
## Milestone 3

Qiaosen Chen, Haoyu Sui

# Progress

Last milestone:

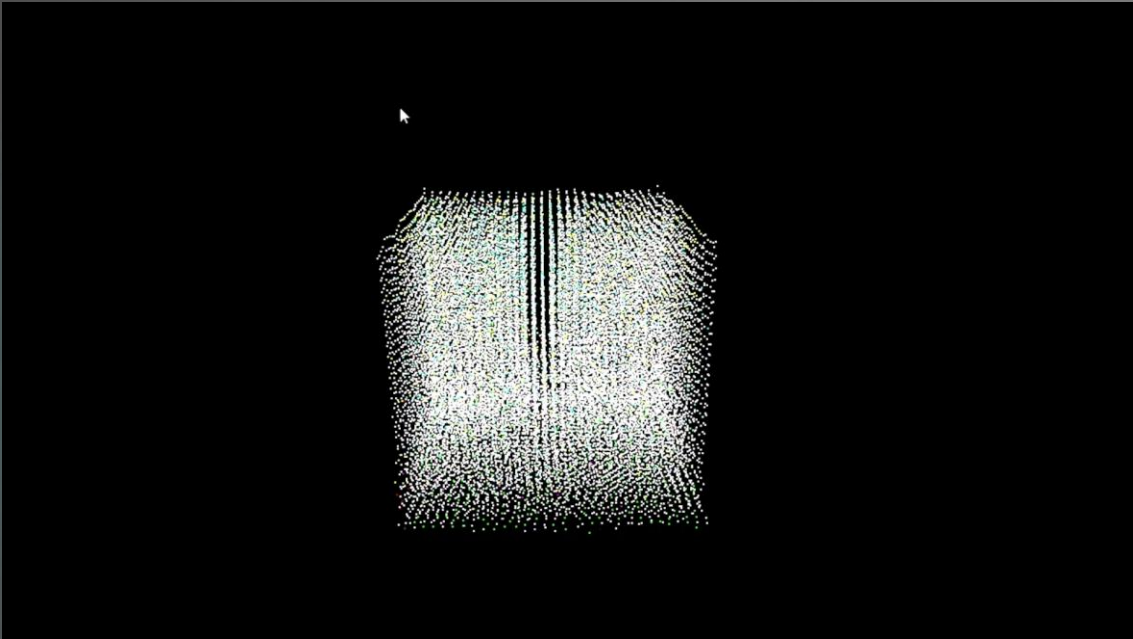
- Implemented the basic math and physics parts
- Achieved a off-line implementation worked on CPU



# Progress

- Complete the remaining parts of math and physics
- Accelerate the simulation process by using GPU computations (CUDA + OpenGL/Vulkan)
- Achieve the real-time simulation process

OpenGL

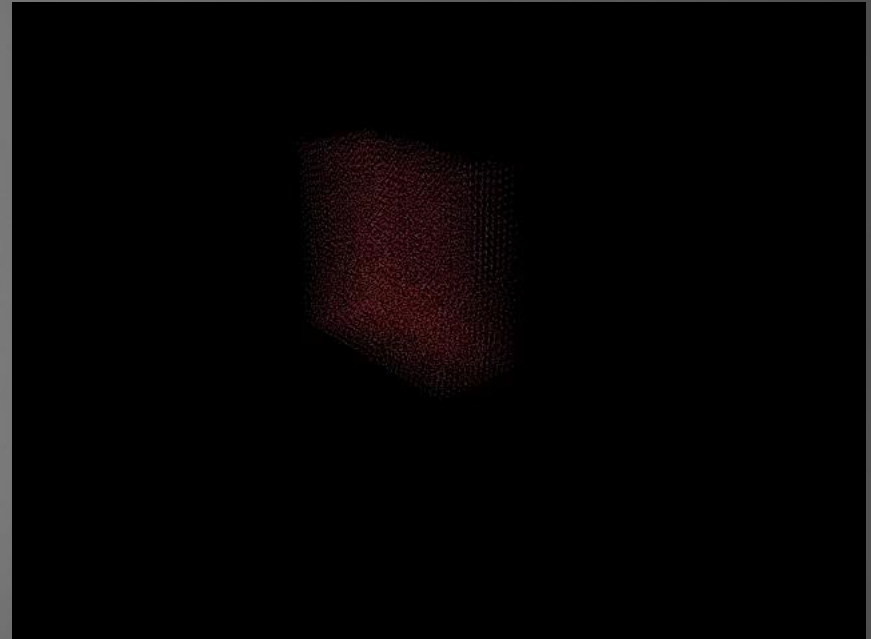


Particle number: 27000

Average compute time: 28.12ms per frame

(Average compute time on previous CPU implementation : 2140ms per frame )

Vulkan



Particle number: 27000

Average compute time: 45.66ms per frame

## Progress

- Complete the remaining parts of math and physics
- Accelerate the simulation process by using GPU computations
- Achieve the real-time simulation process

## Goals for next milestone

- Continue working on Vulkan compute pipeline
- Work on visualization part to get better effect
- Add gui to change key params
- Code optimization and trying to accelerate the algorithm further



# Live demo

# References

- Real-time particle-based snow simulation on the GPU  
<https://www.diva-portal.org/smash/get/diva2:1320769/FULLTEXT01.pdf>
- Real-time particle-based snow simulation on the GPU  
<https://www.diva-portal.org/smash/get/diva2:1118073/FULLTEXT02>
- A material point method for snow simulation  
<https://www.math.ucla.edu/~jteran/papers/SSCTS13.pdf>
- Nvidia: use GPU to simulate fluid  
<https://developer.nvidia.com/gpugems/gpugems/part-vi-beyond-triangles/chapter-38-fast-fluid-dynamics-simulation-gpu>
- Vulkan Tutorial  
<https://vulkan-tutorial.com/Introduction>
- tinyobjloader  
<https://github.com/tinyobjloader/tinyobjloader>