Interactive Visualization Term Project

SPH-simulation report

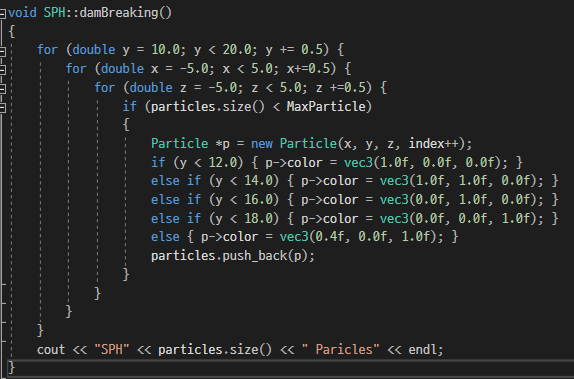
2016320189 이광렬

목차

1. [구현 목표](#구현목표)
2. [구현 과정](#구현과정)
   1. [기본 구현 사항](#기본구현사항)
      1. 3D SPH
      2. Lighting
      3. Background Texture
   2. [추가 구현 사항](#추가구현사항)
      1. Advanced Data Structure
3. [핵심 소스코드](#핵심소스코드)
4. [구현 결과 화면](#구현결과화면)
5. 구현 목표

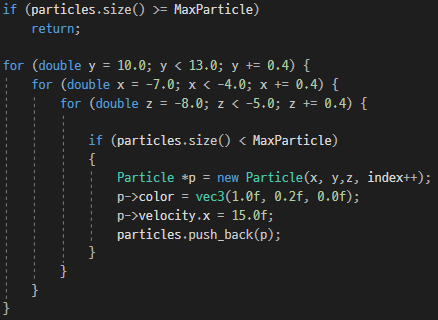
1) 3D 환경에서의 SPH 시뮬레이션을 통해 유체의 움직임을 구현

1. 구현 과정
   1. 기본 구현 사항
      1. 3D SPH
         1. Particle
            1. SPH.cpp - damBreacking 함수에서 시뮬레이션 초기의 파티클 위치를 고정



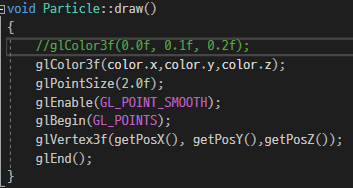
SPH.cpp – void SPH::damBreaking()

* + - * 1. damBreaking 함수와 마찬가지로 pouring 함수에서 사용할 파티클 위치 고정



SPH.cpp – void SPH::pouring()

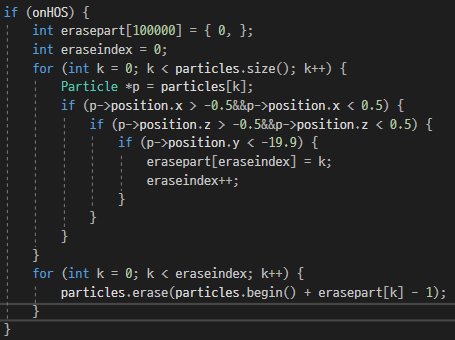
* + - * 1. Draw particles



Draw.cpp - void Particle::draw()

* + - * 1. Delete particles

Texture load 이후 특정 위치에 존재하는 particle을 제거

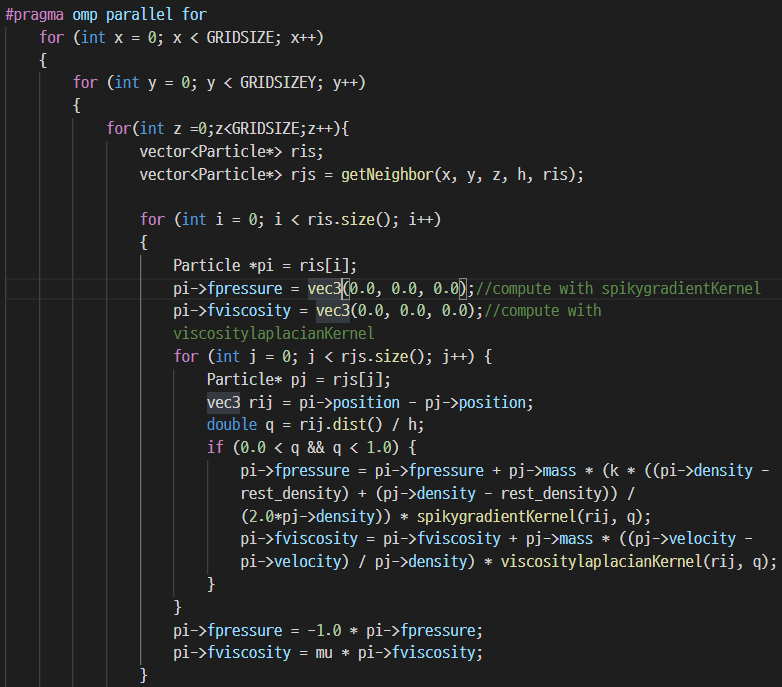


SPH.cpp - void SPH::update()

* + - 1. Compute Force, Density
         1. Compute Force

Skeleton code에서 2D->3D 변환

원활한 시뮬레이션을 위해 openmp 사용

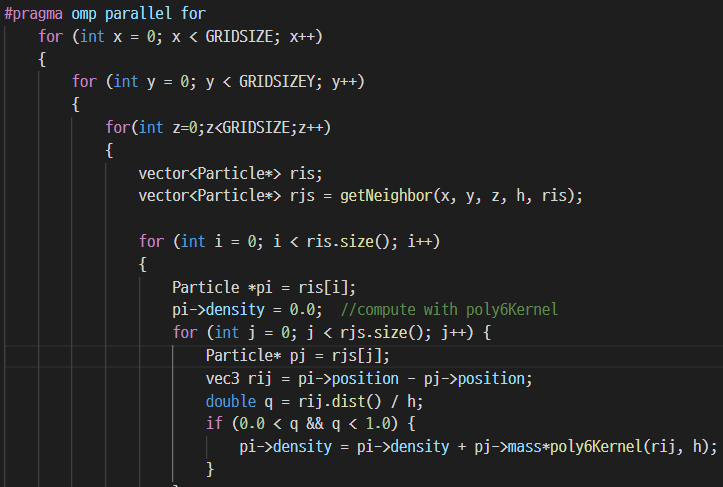


SPH.cpp - void SPH::computeForce()

* + - * 1. Compute Density

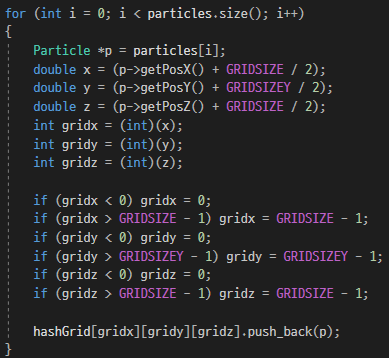
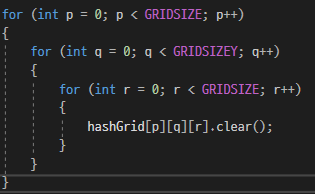
Skeleton code에서 2D->3D 변환

원활한 시뮬레이션을 위해 openmp 사용



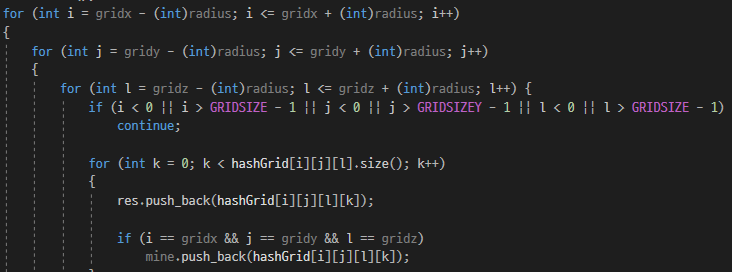
SPH.cpp - void SPH::computeDensity()

* + - 1. Hashgrid
         1. 3D grid 구현



SPH.cpp – void SPH::makeHashTable()

* + - * 1. 3D grid상에서 주변 파티클 확인



SPH.cpp - vector<Particle \*> SPH::getNeighbor()

* + 1. Lighting
       1. Ground
          1. Ground 선언

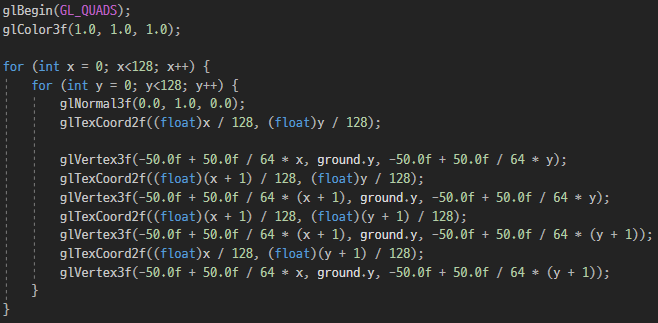
Y = -20.0 위치에 ground 선언



Simulator.cpp - void Simulator::Initialize()

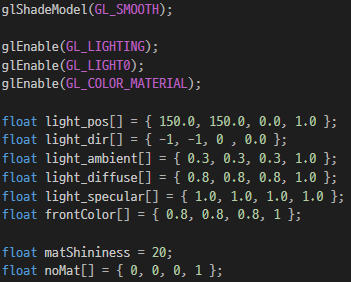
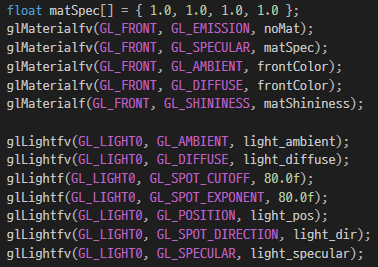
* + - * 1. Draw ground

(x,y,z) = (-50,-20,-50)~(50,-20,50)구간에 ground를 그려줌



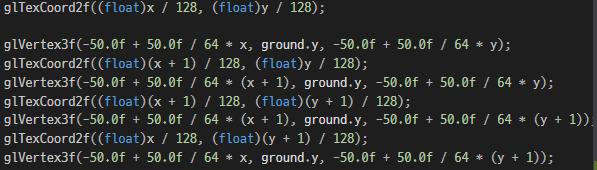
Simulator.cpp - void Simulator::DrawGround()

* + - 1. Lighting
         1. PA#2의 lighting code 사용

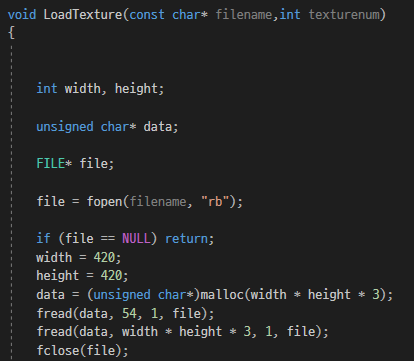
Simulator.cpp - void Simulator::Lighting()

* + 1. Background Texture
       1. Ground 위에 texture를 적용
          1. 전체 Ground -> (0,0)~(1,1) texture coordinate



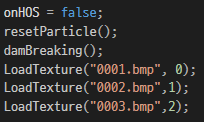
Simulator.cpp - void Simulator::DrawGround()

* + - * 1. Image load (PA#2에서 사용한 LoadTexture code 사용)



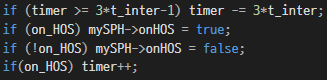
SPH.h - void LoadTexture()

* + - 1. 시뮬레이션 시간에 따라 texture 변경
         1. Texture에 사용할 Image들을 배열에 저장

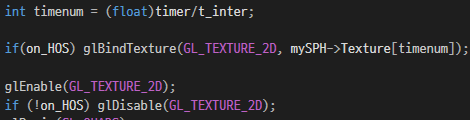


SPH.cpp - void SPH::init()

* + - * 1. 시간이 지남에 따라 다른 이미지를 불러와 texture 변경



Simulator.cpp - void Simulator::Update()



Simulator.cpp - void Simulator::DrawGround()

* 1. 추가 구현 사항
     1. Advanced Data Structure
        1. 3D SPH 구현 과정에서 skeleton code(2D) 변형하여(3D) 사용
           1. damBreaking(), pouring(), makeHashTable() 등등..

1. 핵심 소스코드

[2. 구현 과정](#구현과정)에 포함.

1. 구현 결과 화면

