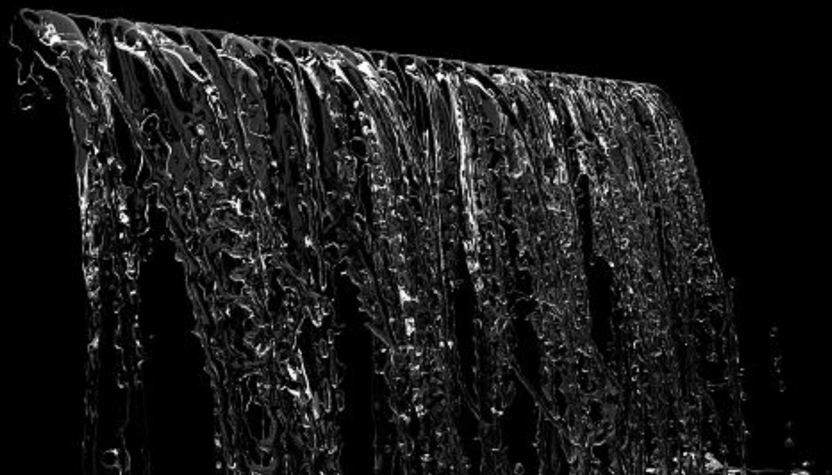
CS 334 Final Project

Water-like Particle-Based Fluid Simulation and Procedural Modeling of Mazes

Victor Fenton Aguilar

Overview

- Fluid Simulation
- SPH (smoothed particle hydrodynamics)
- Spatial Hashing Optimization
- Collision Detection
- Procedural Modeling of Maze



Smooth Particle Dynamics (SPH)

In: support length h, subdivision factor H and delta time Δt

function SPH$(h, H, \Delta t)$

$\textsc{Neighbours} \leftarrow \textsc{SearchNeighbors}(h, H)$
foreach $\mathcal{P}_i$ **in** $\textsc{Particles}$ **do**
  $\rho_i \leftarrow 0$; $\nabla C_i \leftarrow \mathbf{0}$; $\nabla^2 C_i \leftarrow 0$; $\mathbf{f}_i \leftarrow \mathbf{f}_i^{\text{ext}}$ `/* initialize */`
  foreach $\mathcal{P}_j$ **in** $\textsc{Neighbours}(\mathcal{P}_i)$ **do** `/* accumulate density */`
    $\rho_i \leftarrow \rho_i + m_j W^{\text{poly}}(\mathbf{r}_i - \mathbf{r}_j, h)$
  end
  $P_i \leftarrow k^{\text{gas}} \left(\left(\frac{\rho}{\rho_0} \right)^{\gamma} - 1 \right)$ `/* calculate pressure */`
  foreach $\mathcal{P}_j$ **in** $\textsc{Neighbours}(\mathcal{P}_i)$ **do** `/* accumulate forces */`
    $\mathbf{f}_i \leftarrow \mathbf{f}_i - V_i V_j \frac{P_i + P_j}{2} \nabla W^{\text{press}}(\mathbf{r}_i - \mathbf{r}_j, h)$ `/* (=` $\mathbf{f}_i^{\text{press}}$ `) */`
    $\mathbf{f}_i \leftarrow \mathbf{f}_i + V_i V_j \frac{\mu_i + \mu_j}{2} (v_j - v_i) \nabla^2 W^{\text{visco}}(\mathbf{r}_i - \mathbf{r}_j, h)$ `/* (=` $\mathbf{f}_i^{\text{visco}}$ `) */`
    $\nabla C_i \leftarrow \nabla C_i + V_j c_j^{\text{int}} \nabla W^{\text{poly}}(\mathbf{r}_i - \mathbf{r}_j, h)$ `/* (=` ∇C_i^{int} `) */`
    $\nabla^2 C_i \leftarrow \nabla^2 C_i + V_j c_j^{\text{int}} \nabla^2 W^{\text{poly}}(\mathbf{r}_i - \mathbf{r}_j, h)$ `/* (=` $\nabla^2 C_i^{\text{int}}$ `) */`
  end
  $\mathbf{f}_i \leftarrow \mathbf{f}_i - \sigma^{\text{int}} \nabla^2 C_i^{\text{int}} \frac{\nabla C_i^{\text{int}}}{|\nabla C_i^{\text{int}}|}$ `/* (=` $\mathbf{f}_i^{\text{int}}$ `) */`
end
foreach $\mathcal{P}_i$ **in** $\textsc{Particles}$ **do** `/* Leap-Frog */`
  $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t \frac{\mathbf{f}_i}{m_i}$
  $\mathbf{r}_i \leftarrow \mathbf{r}_i + \Delta t \mathbf{v}_i$
end
end

The Navier-Stokes equations are given by:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) = -\nabla p + \eta \nabla^2 \mathbf{u} + \rho \mathbf{g}$$

subject to the incompressibility constraint

$$\nabla \cdot \mathbf{u} = 0$$

$$\rho \frac{D\mathbf{u}}{Dt} = \sum \mathbf{F} = \mathbf{F}^{\text{pressure}} + \mathbf{F}^{\text{viscosity}} + \rho \mathbf{g}$$

$$\rho_i = \rho(\mathbf{r}_i) = \sum_j m_j \frac{\rho_j}{\rho_j} W(|\mathbf{r}_i - \mathbf{r}_j|, h) = \sum_j m_j W(|\mathbf{r}_i - \mathbf{r}_j|, h)$$

$$\mathbf{F}_i^{\text{pressure}} = -\nabla p(\mathbf{r}_i) = -\sum_j m_i m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}\right) \nabla W(|\mathbf{r}_i - \mathbf{r}_j|, h)$$

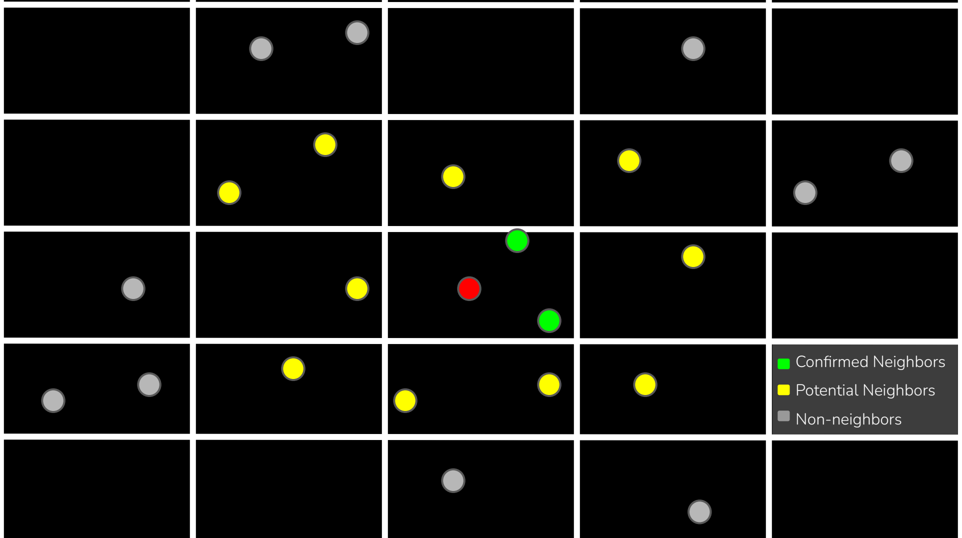
$$\mathbf{F}_i^{\text{viscosity}} = \eta \nabla^2 \mathbf{u}(\mathbf{r}_i) = \eta \sum_j m_j \frac{|\mathbf{u}_j - \mathbf{u}_i|}{\rho_j} \nabla^2 W(|\mathbf{r}_i - \mathbf{r}_j|, h)$$

SPATIAL HASHING

vs. Quadtrees?

```
int SHhash(Particle p)
{
    return static_cast<int>(p.pos.x / cellDim) + (static_cast<int>(p.pos.y / cellDim) * 1000);
}
```

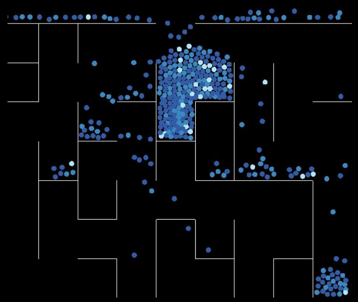
```
vector<Particle> findNeighbors(Particle p)
{
    vector<Particle> neighbors;

    int cell = SHhash(p);
    for (int i = -1; i <= 1; i++)
    {
        for (int j = -1; j <= 1; j++)
        {
            int key = cell + (i * 1000) + j;
            if (SHGrid.find(key) == SHGrid.end()) continue;
            for (auto& pn : SHGrid[key])
            {
                neighbors.push_back(pn);
            }
        }
    }
    return neighbors;
}
```

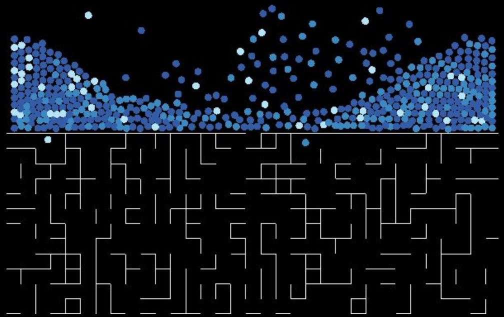


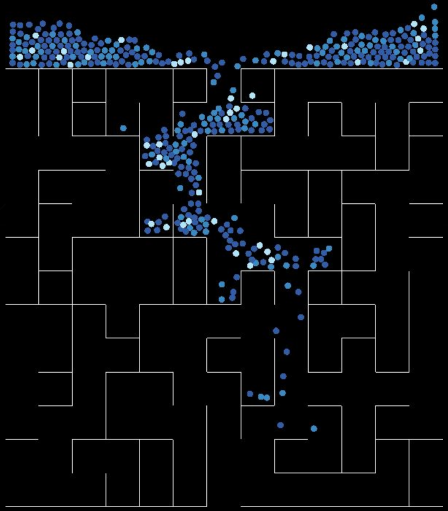
Procedural Modeling of Mazes

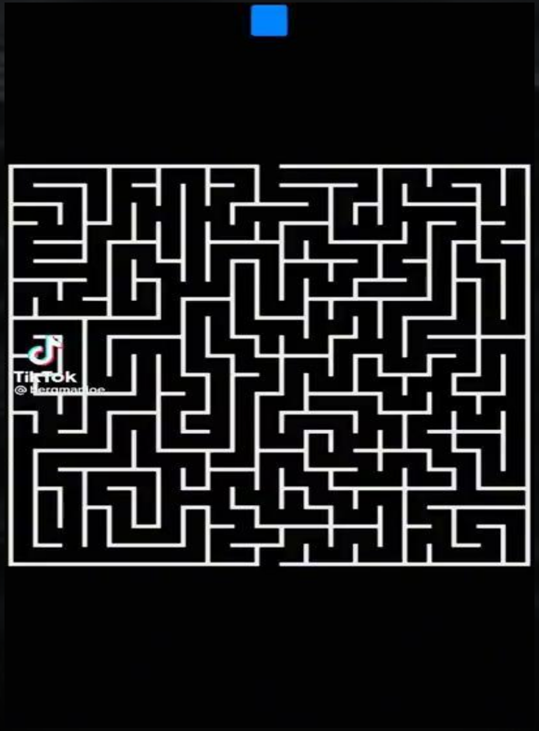
Approach:

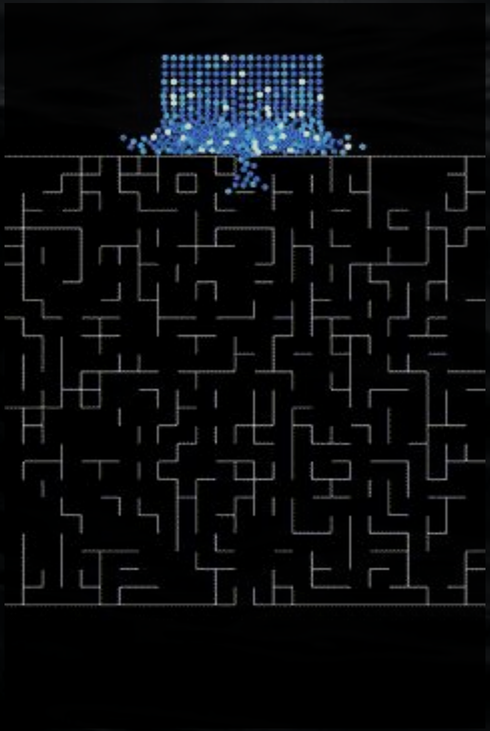
- User-Controlled size of grid.
- Fixed outer frame with carved entry/exit channels at the top and bottom.
- Inner maze walls generated in a loop.
- User provides a probability threshold to control randomization.
- Random number generated before each wall is created, if the number is less than threshold, the wall will not be drawn.
- Bias is added to horizontal threshold to encourage downflow.











Expected Demo

Achieved Demo

References

- https://cseweb.ucsd.edu/classes/sp19/cse291-d/Files/CSE291_09_ParticleBasedFluids.pdf
- https://www.cs.cornell.edu/courses/cs5643/2015sp/a1PositionBasedFluids/
- http://mmacklin.com/pbf_sig_preprint.pdf
- https://www.inf.ufrgs.br/cgi2007/cd_cgi/papers/harada.pdf
- https://nccastaff.bournemouth.ac.uk/jmacey/MastersProject/MSc16/13/thesis.pdf
- https://lucasschuermann.com/writing/particle-based-fluid-simulation
- https://lucasschuermann.com/writing/implementing-sph-in-2d
- https://users.csc.calpoly.edu/~zwood/teaching/csc572/final15/awang/index.html