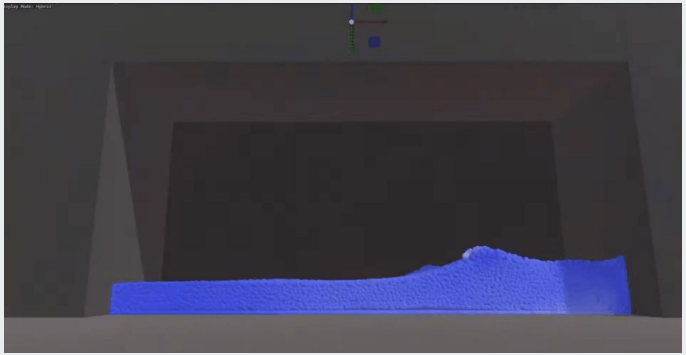
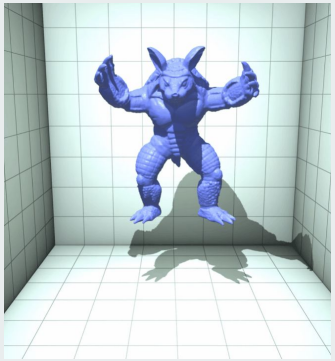
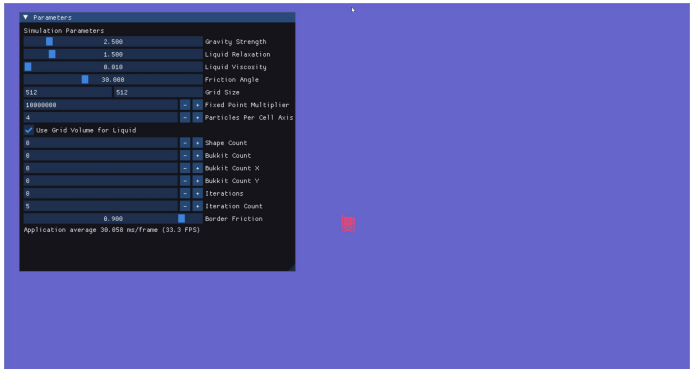
# BREAKPOINT Milestone 2


+

=


Daniel Gerhardt, Dineth Meegoda, Matt Schwartz, Zixiao (Steve) Wang

# DX12 Core Progress

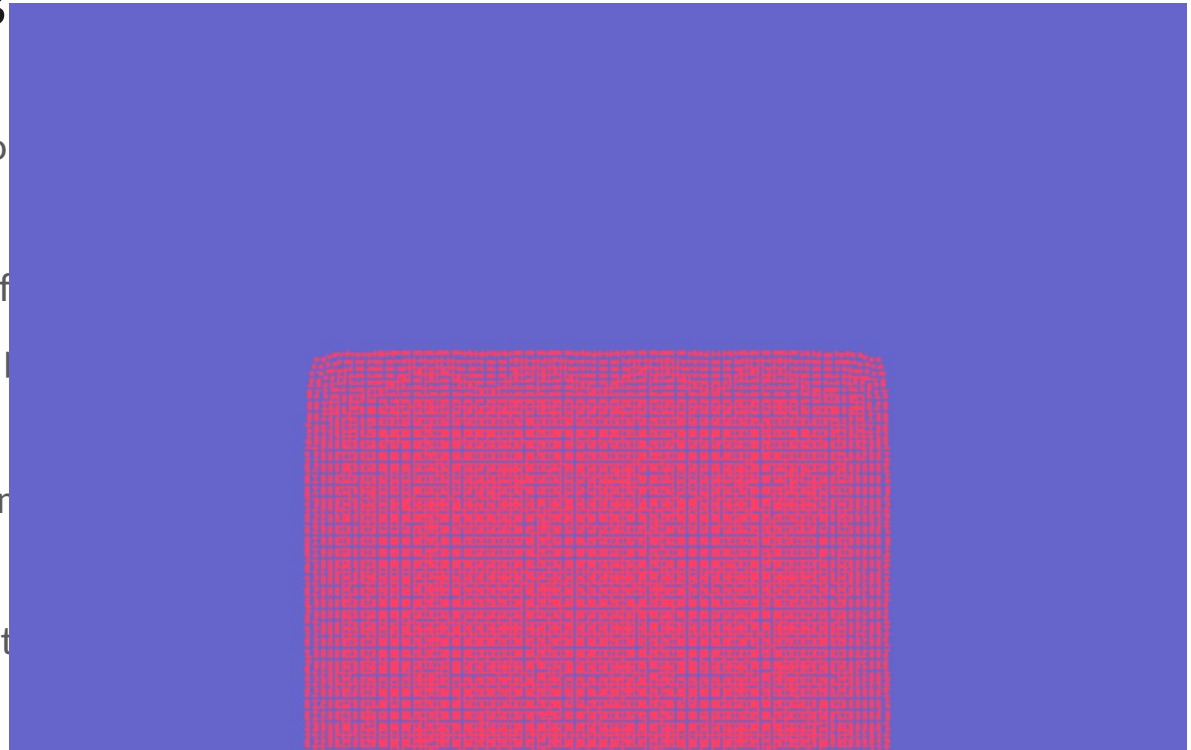



- ImGUI Integration
- Compute, GPU data reworks
- Scene rework with refactored render pipelines
- Most of Core in Milestone 1, Milestone 2 work focused elsewhere
  - PBMPM Implementation/Debugging
  - Fluid mesh shading
- PBMPM: 3d, shared memory optimization, integration with PBD

*"Give someone state and they'll have a bug one day, but teach them how to represent state in two separate locations that have to be kept in sync and they'll have bugs for a lifetime." -ryg*

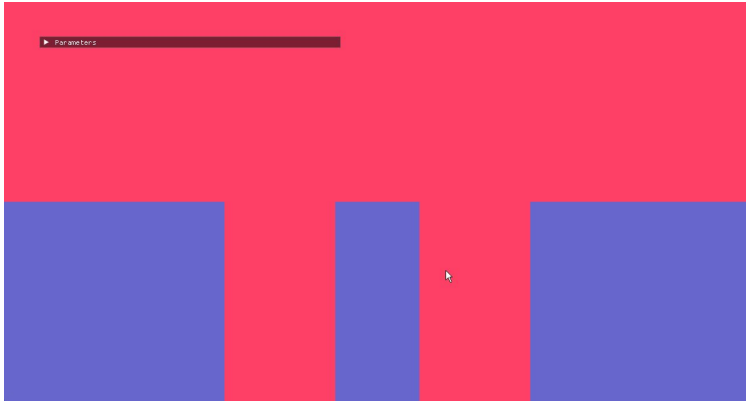
# PBMPM Progress

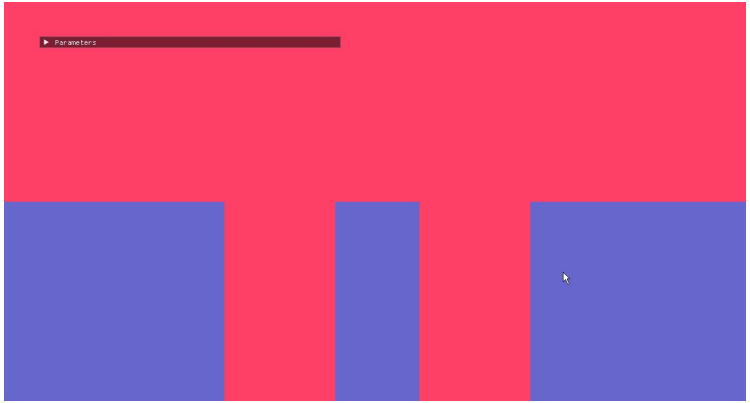
- 2D Paper implementatio
- Limitations:
  - Parameter Tuning f
  - d size scaled to
  - l:
  - PBMPM Impler
  - Particle emission
  - Dynamic forces wit
    layout adjustment



# Real Time Destruction Progress

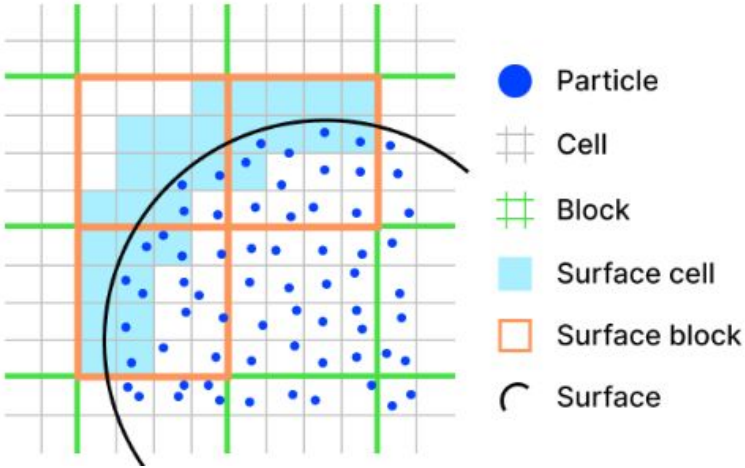
- Writing PBD 3D constraints
  - Gram-Schmidt orthonormal
  - Face to face connection constraints
- It can be break now with enough force but keep cubic voxel shape
- Limitation:
  - When force is not enough too break, the deformation of shape is not working.
- MS3 Goal: Try to fix the deformation and integrate with large scale voxels. Try to integrate into PBMPM



# Mesh Shading Pipeline Progress

- All 6 compute passes + marching cubes mesh shading are implemented
- Optimized the heck out of it!
  - Shared memory
  - Stream compaction
  - Reduction of modular arithmetic
  - And much, *much* more…
- TODO: fluid sim integration, benchmarking, possible mesh shading optimizations.



# Looking Ahead

1. Finish moving PBMPM to 3d
2. Test voxelization forces on voxelized mesh
3. Interacting PBMPM with PBD
4. Render PBMPM with fluid mesh shading
5. Render PBD with “normal” mesh shading