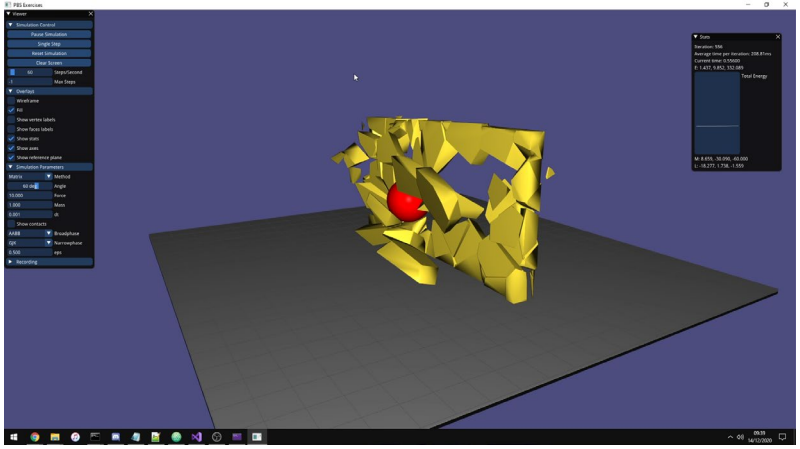
Physically-Based Simulation
Wall destruction

Group 8

Thierry Backes, Adrian Taubner, Stefan Jokić

Simulation Scenario

- Fractured wall destroyed by collision with rigid body which is thrown at wall
  - Rigid body simulator
  - Collision (Detection + Handling)
  - Fractures



Implementation

- Advance step:
  - Apply external forces (gravity)
  - Compute collisions and apply adequate forces (resting / impulse) where necessary
  - Integrate position and velocity via semi-implicit (symplectic) Euler
  - Update orientation using quaternions

Implementation : Collision detection

- Broad phase:
  - Axis-Aligned Bounding Boxes (AABB)
  - Check for possible collision if boxes overlap
- Narrow phase:
  - First Gilbert–Johnson–Keerthi (GJK) algorithm: Check if collision is present
  - Then Expanding Polytope Algorithm (EPA): Obtain information about collision (contact face, normal vector,…)

Implementation : Multiple resting contacts

- Implemented according to lecture notes "An Introduction to Physically Based Modeling" by D. Baraff
- First extracted resting contacts from all contacts by checking if relative velocity of bodies in contact is less than fixed threshold (eps).
- Then, using information about resting contacts, computed A matrix and b vector according to Baraff.

Implementation : Multiple resting contacts

- Next, we formulated an LCP problem using A and b as follows:

$$\min_{\boldsymbol{f}} \boldsymbol{f}^T(\mathbf{A}\boldsymbol{f}+\mathbf{b}) \qquad \text{subject to} \qquad \left\{ \begin{array}{c} \mathbf{A}\boldsymbol{f}+\mathbf{b} \geq \mathbf{0} \\ \boldsymbol{f} \geq \mathbf{0} \end{array} \right\}.$$

  where f is the unknown and corresponds to a vector of resting contact forces, one for each resting contact.

- We solved this problem by implementing Dantzig's LCP solver algorithm.

Implementation: Multiple contacts

- We then also included impulse-based contacts to A and adjusted b accordingly such that the system would now include all contacts.
- Our LCP solver would still solve the problem with no issues, however the computed impulse forces had "strange" values that would send the fractures of the wall flying around the scene.

Implementation: Multiple contacts

- Finally, had to resort to iterative handling of impulses again...
- Good news: We don't *really* need multiple contact handling (for impulses) for our simulation (e.g. it never occurs that a cube will fall flat on the ground)
- Besides, multiple resting contacts working fine (in most cases, at least)

Bugs...

- Encountered many issues regarding both GJK/EPA and the LCP solver
- Probably spent more time debugging than actually implementing anything. Still quite a few bugs left.
- For instance, our GJK implementation cannot find simultaneous vertex-face & edge-edge collision between two bodies.

Bugs...

- Also, our LCP solver will loop infinitely in certain cases.
- Unclear what exactly is causing it. Sometimes just the position of a cube on the ground will decide whether or not the algorithm gets stuck. (Numeric issues?)
- Fixed infinite loop problem for a few cases that we found out, but several cases still remaining ...

Fractured wall

- Generated fractured wall using blender (Plugin: Cell fracture)
- Fractures generated automatically and with random pattern.
- Problem: Fractures of wall will start falling down and colliding against each other.
- Fix: Implemented a small "hack" where gravity would just be turned off until ball was about to hit the wall.

Fractured wall

- Bonus targets: Dynamic fracturing and smoke effect not implemented.
- However, desired target still achieved. (Ignoring the bugs)

Demonstration

Q & A