

Simulation of paintball shot

A project by Kevin Wallimann, Andri Schmidt and Marc Maetz

December 18, 2013

1 Introduction

Inspiration

Expected challenges

2 Approach

3 Implementation

Low level

High level

Difficulties

Tricks

4 Demonstration

5 Conclusion

- Initial project idea:
Paintball shot on wall
- Final project:
Paintball shot on simple objects

Video

Expected challenges

- OpenGL
 - showing simple objects
- SPH solver
 - time to make it work as expected

Approach

- SPH solver
- Collision handling with no library but boundary boxes

Why?

- SPH solver
 - splashes and droplets
- Collision handling with no library but boundary boxes
 - late attacket
 - risk of heavy time investission

Low level

- C++ 11
 - eigen
- OpenGL
 - glsl
 - glm

High level

- 1 Shot with gravitation
- 2 Collision detection
- 3 On impact the SPH simulation starts

Difficulties

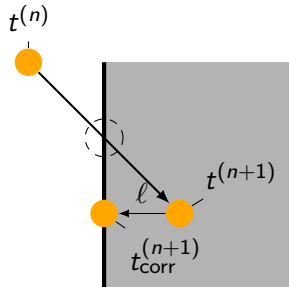
- (general code debugging)
- (parameter tuning)
- (coordinates: simulation space \leftrightarrow window space)
- many particles behaving like a ball
- particles pushed and trapped inside object
- sticky paint on object with given resolution

Tricks

- many particles behaving like a ball
 - particle-particle forces ignored before collision
- particles pushed and trapped inside object
 - projection further than on surface
- sticky paint on object
 - velocity dependent trace on object

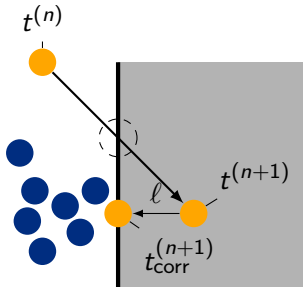
Particles pushed and trapped inside object

naive method



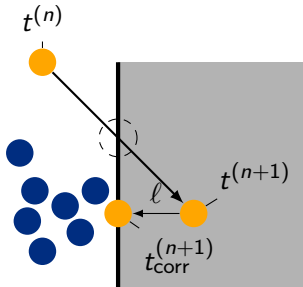
Particles pushed and trapped inside object

naive method

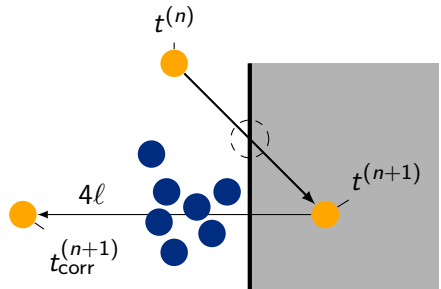


Particles pushed and trapped inside object

naive method



our method

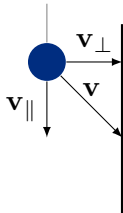


Sticky paint on object

Trace handling

Velocity split

SPH particle



Resulting trace

$$\mathbf{v}_\parallel = 0$$

$$|\mathbf{v}| \text{ big}$$



$$\mathbf{v}_\parallel \neq 0$$

$$|\mathbf{v}| \text{ big}$$



$$\mathbf{v}_\parallel = 0$$

$$|\mathbf{v}| \text{ small}$$



$$\mathbf{v}_\parallel \neq 0$$

$$|\mathbf{v}| \text{ small}$$



Demonstration

Performance

- Real time simulation for simple objects (500 vertices)
- For complex objects substantially slower

Performance

- Real time simulation for simple objects (500 vertices)
- For complex objects substantially slower

Limitations

- resolution and number of collision objects
 - bottleneck collision handling
- number of particles
 - bottleneck SPH simulation

What we learned

- OpenGL
- Blender

What we learned

- OpenGL
- Blender

What we would do different

- Start collision handling earlier