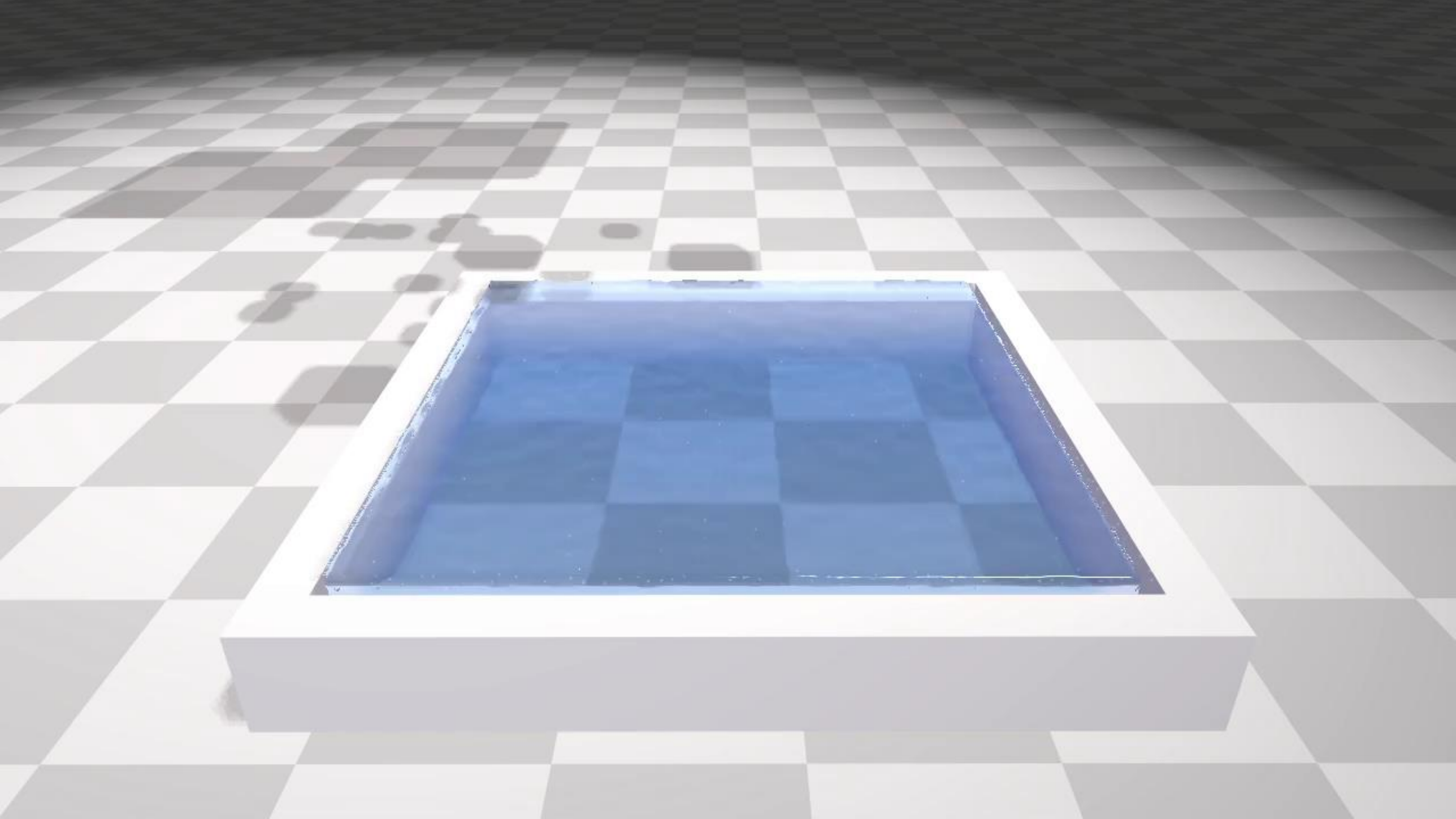
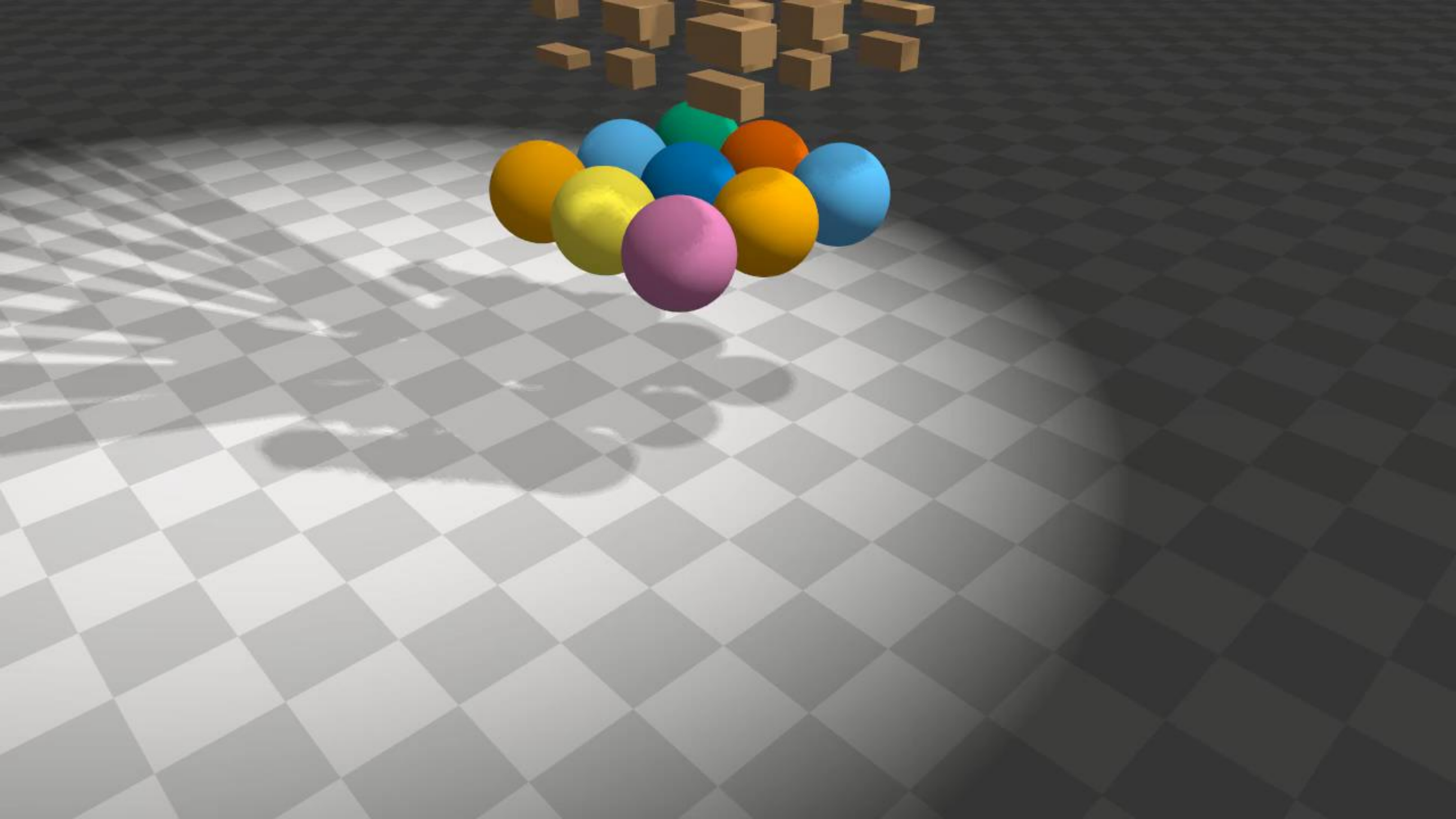
NVIDIA FleX助力打造全交互的游戏场景

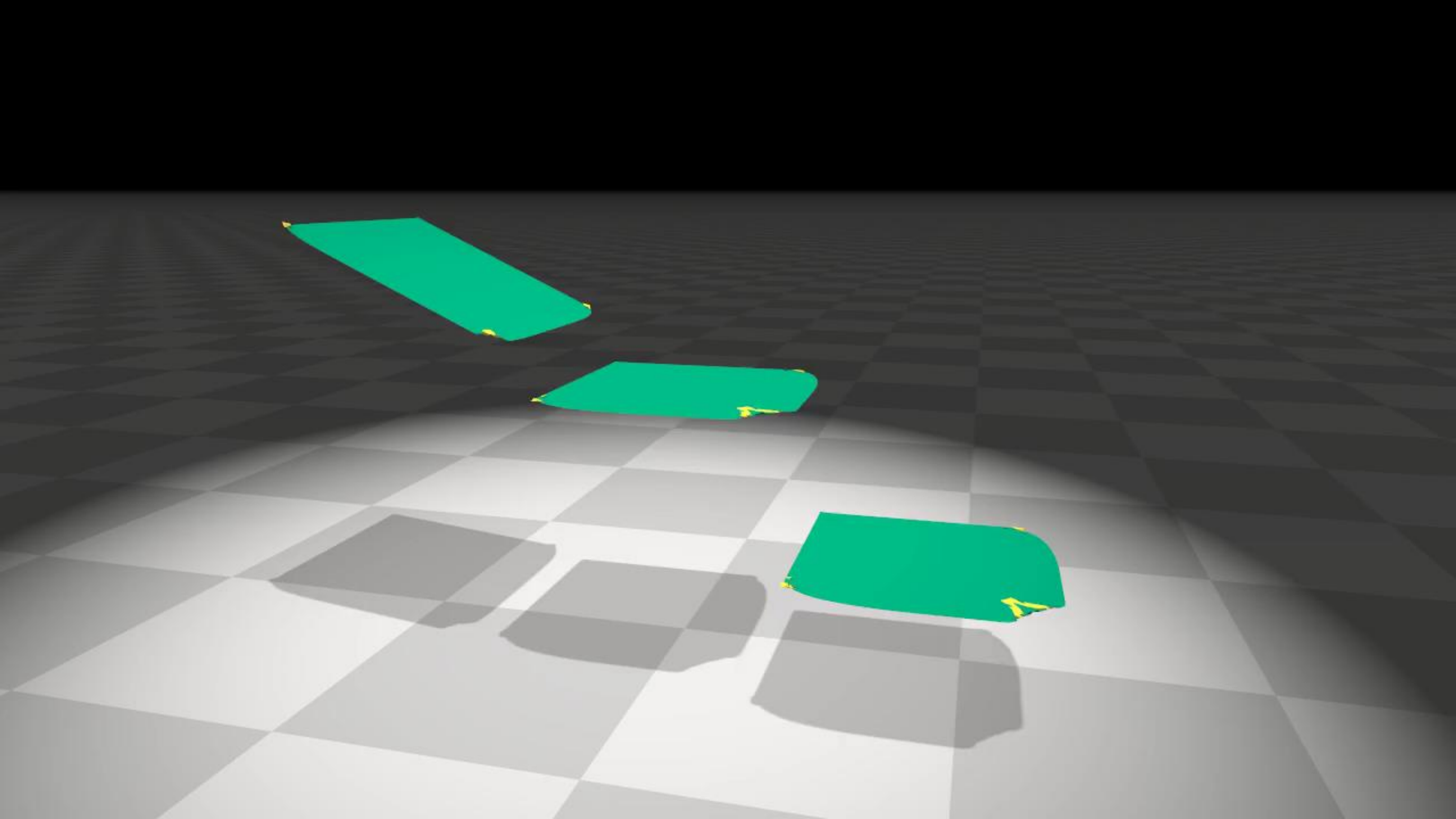
陈　泉

NVIDIA

什么是FleX?

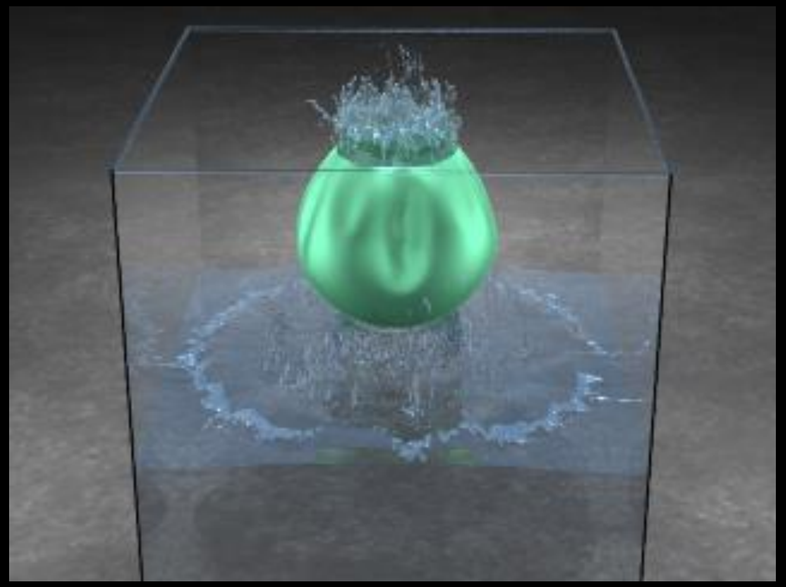




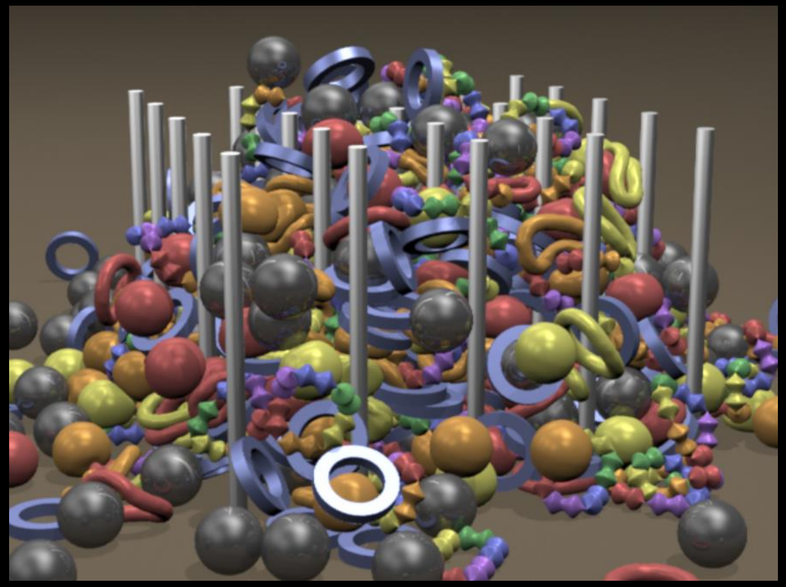


初衷

- 传统引擎每个特性一个解算器
- 带来冗余的工作
- 期望实现统一的解算器
- 期望实现所有特性之间的双向交互



[Robinson-Mosher et al. 2008]



[Shinar et al. 2008]

核心思想

每一个物体都是由某种约束连接起来的一组粒子

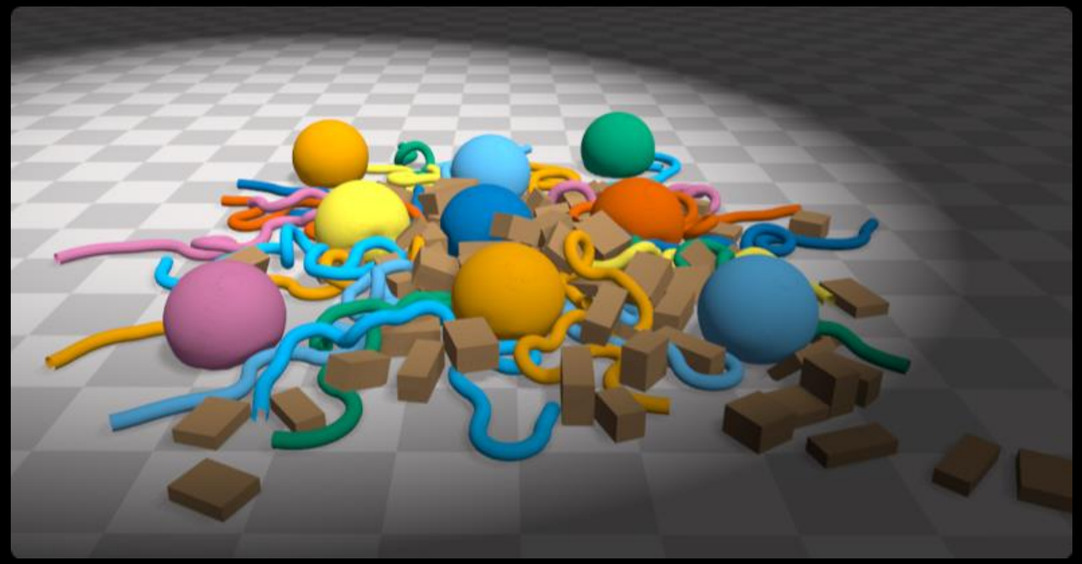
为何选择粒子

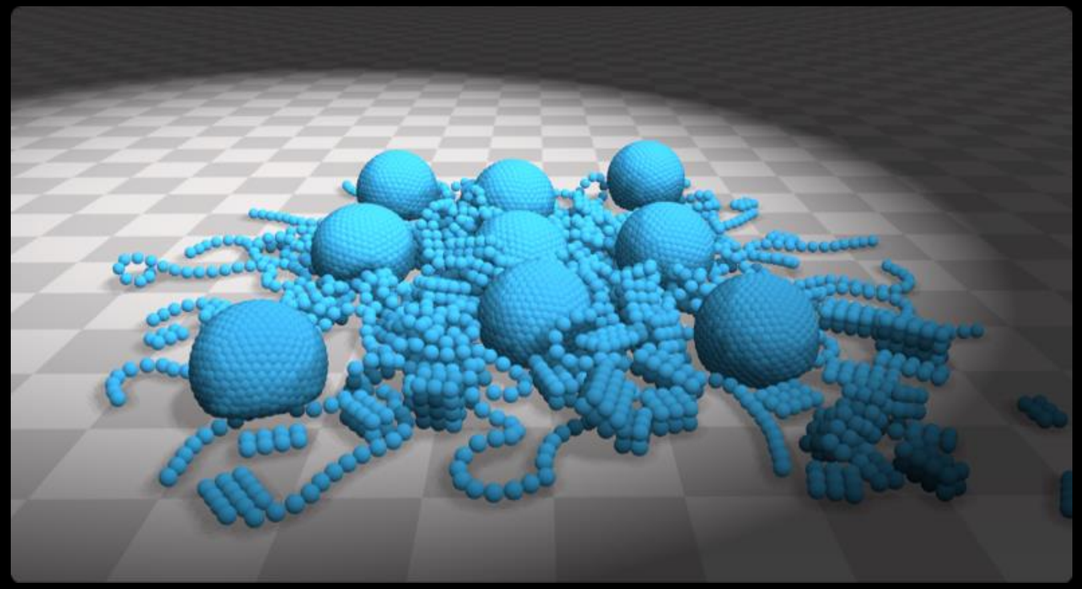
- 简化碰撞检测
- 各个特性物体间稳定的双向交互
  - 布料
  - 软体
  - 流体
  - 刚体
  - 气体（暂未发布）
- 适合在GPU加速

粒子

```
struct Particle
{
  float pos[3];
  float vel[3];
  float invMass;
  int phase;
};
```

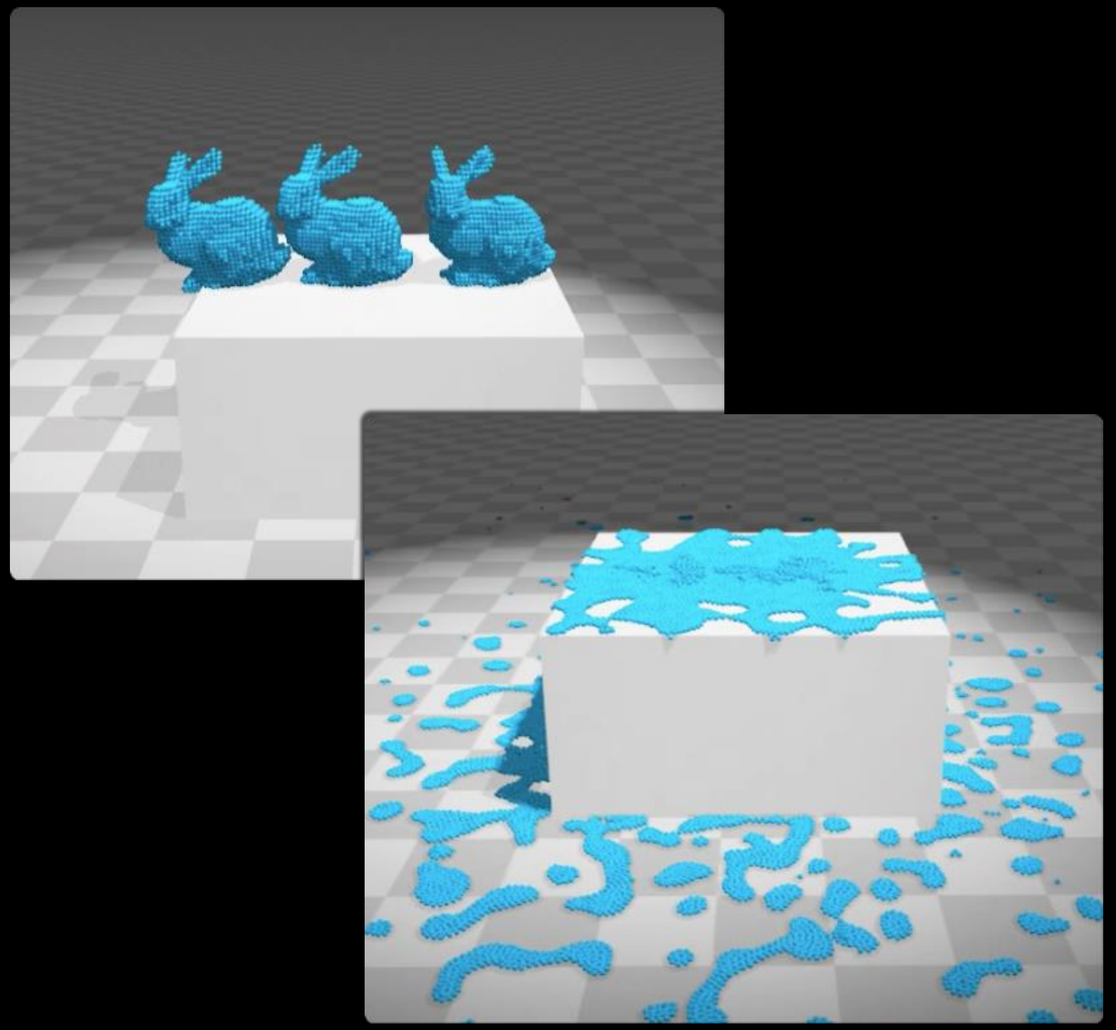
- Phase-ID 用于碰撞检测过滤
- 粒子不属于某一个特定的物体
- 粒子的碰撞半径相同





约束

- 约束类型：
  - 距离（布料）
  - 形状（刚体，塑料材质物体）
  - 密度（液体）
  - 体积（软体）
  - 碰撞（防止穿透，摩擦力计算）
- 组合各种约束，实现更多的效果
  - 融化，相变
  - 硬布料



Solver Loop

1. Apply forces (v = v + 1/m*f*dt)
2. Predict new positions (x* = x + v*dt)
3. Find neighbors, contacts
4. Pre-stabilization
5. For (k iterations)
   1. For each constraint group G, in parallel:

   deltaX = 0

   Solve constraints in G

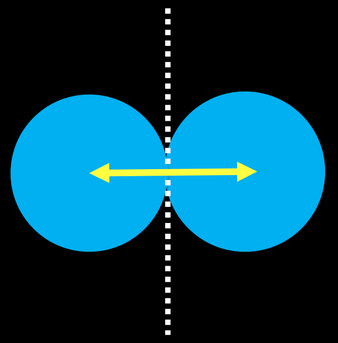
   x* += deltaX*(omega/n)

6. Update velocities (v = (x*-x)/dt)
7. Update positions (x = x*)

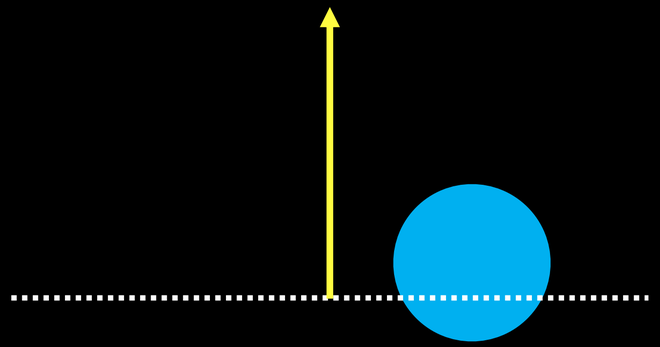
碰撞和摩擦力计算

碰撞检测

- 每一个动态物体都由粒子组成
- Kinematic 物体作为网格(Mesh)处理
- 两种类型的碰撞检测:
  - 粒子-粒子
  - 粒子-网格



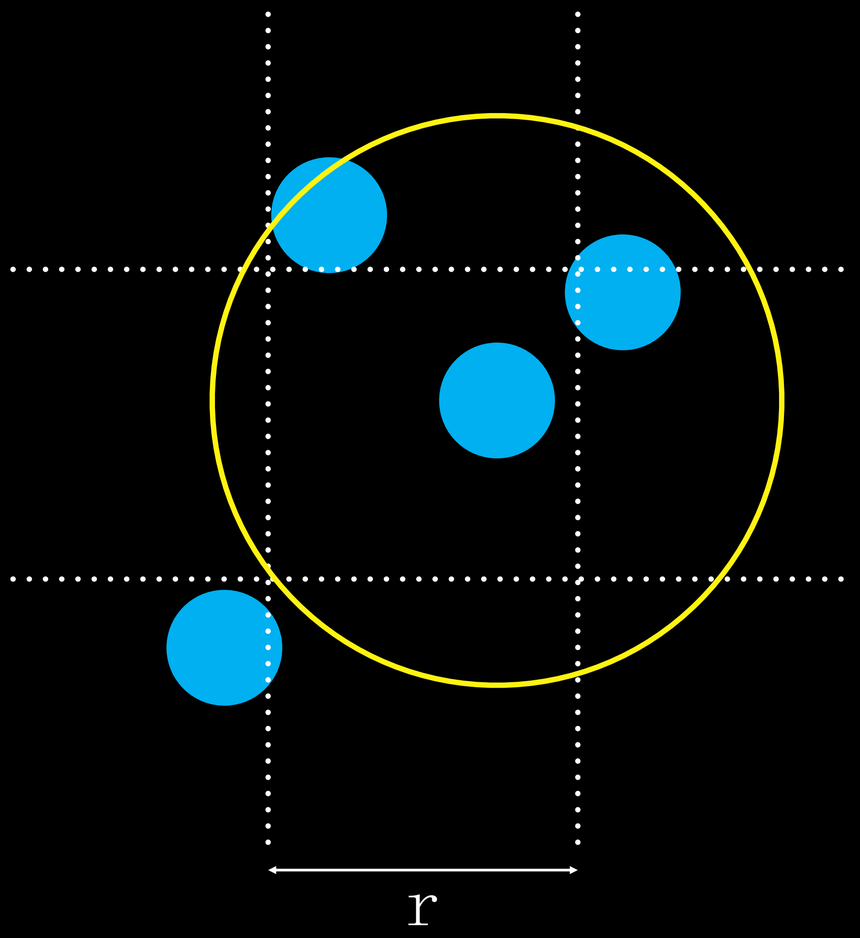
$$C_{contact} = |\mathbf{x}_i - \mathbf{x}_j| - 2r \geq 0$$



$$C_{contact} = \mathbf{n} \cdot \mathbf{x} - r \geq 0$$

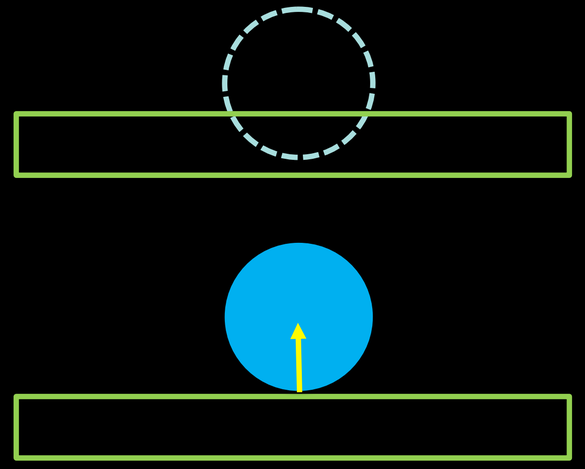
碰撞检测

- 粒子-粒子
  - 均匀划分空间网格
  - 设定寻找附近粒子的最大半径
  - 根据网格索引重新排序粒子，提升内存局部性

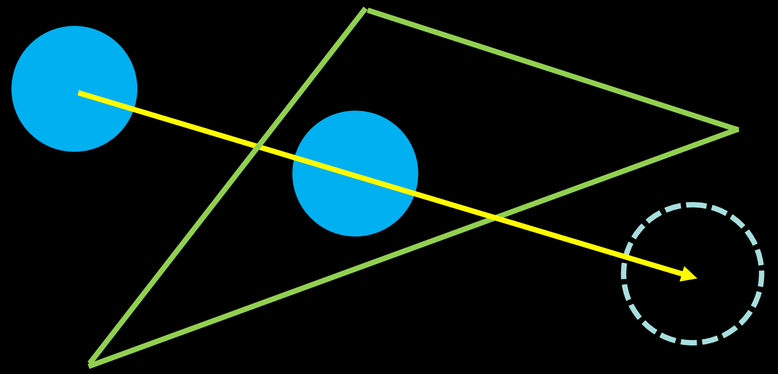


碰撞检测

- 粒子-网格
  - 支持的碰撞体类型
    - 平面
    - 球体，胶囊体
    - 凸包
    - 三角网格（CCD）
    - 有向距离场
  - 摩擦力（动态，静态）
  - 反弹系数



Convex Collision (MTD)
(projection)



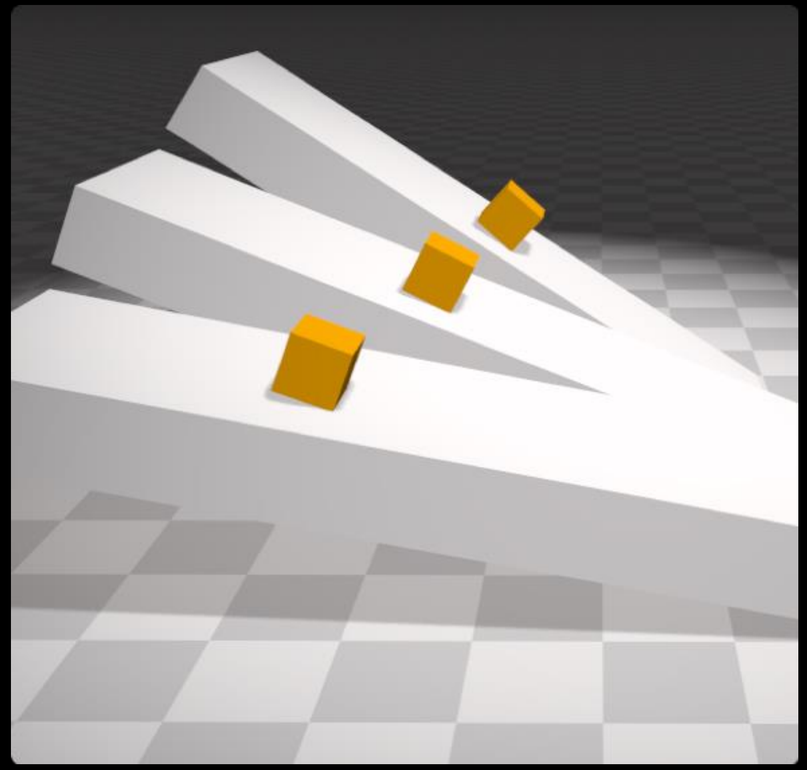
Triangle Collision (TOI)

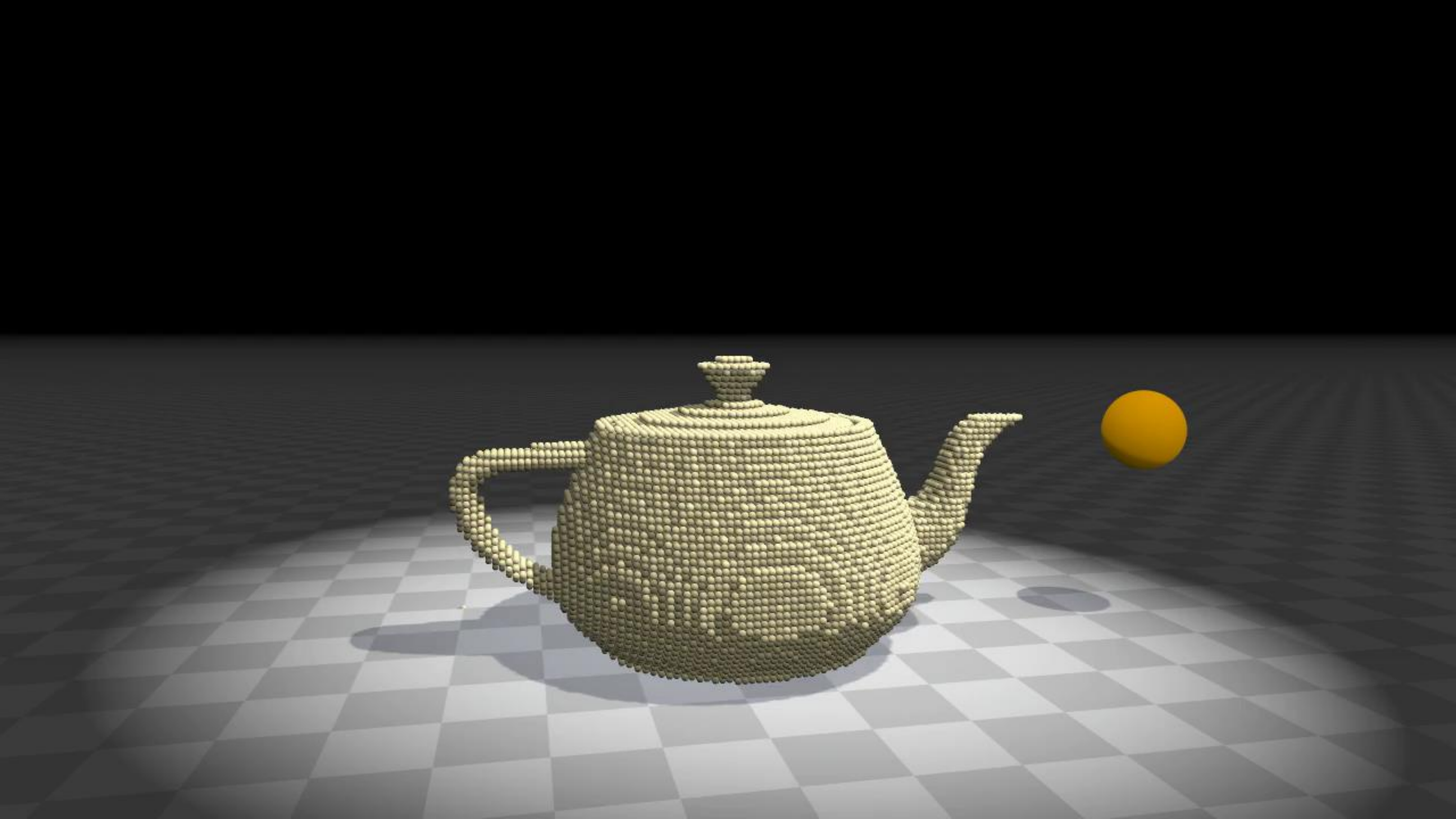
摩擦力

- PBD中通常使用速度的变化估算摩擦力
- 使用位置层面的摩擦力约束

$$C_{friction} = |(\mathbf{x} - \mathbf{x}_0)_{\perp}\mathbf{n}|$$

- 使用穿透深度限制lambda，估算Coulomb摩擦力

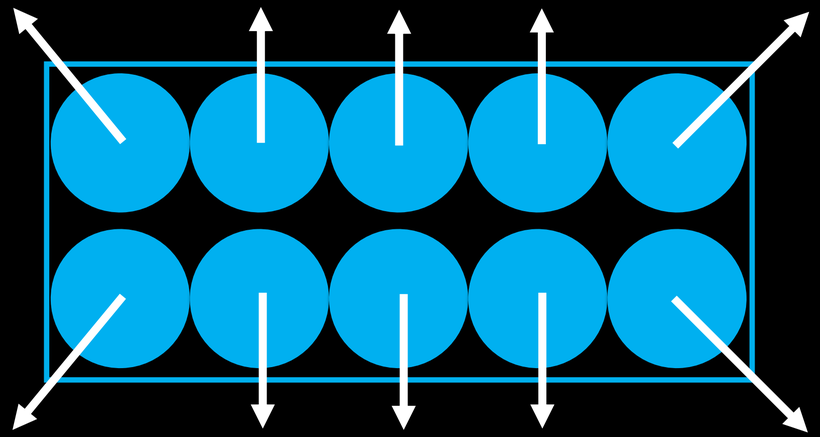


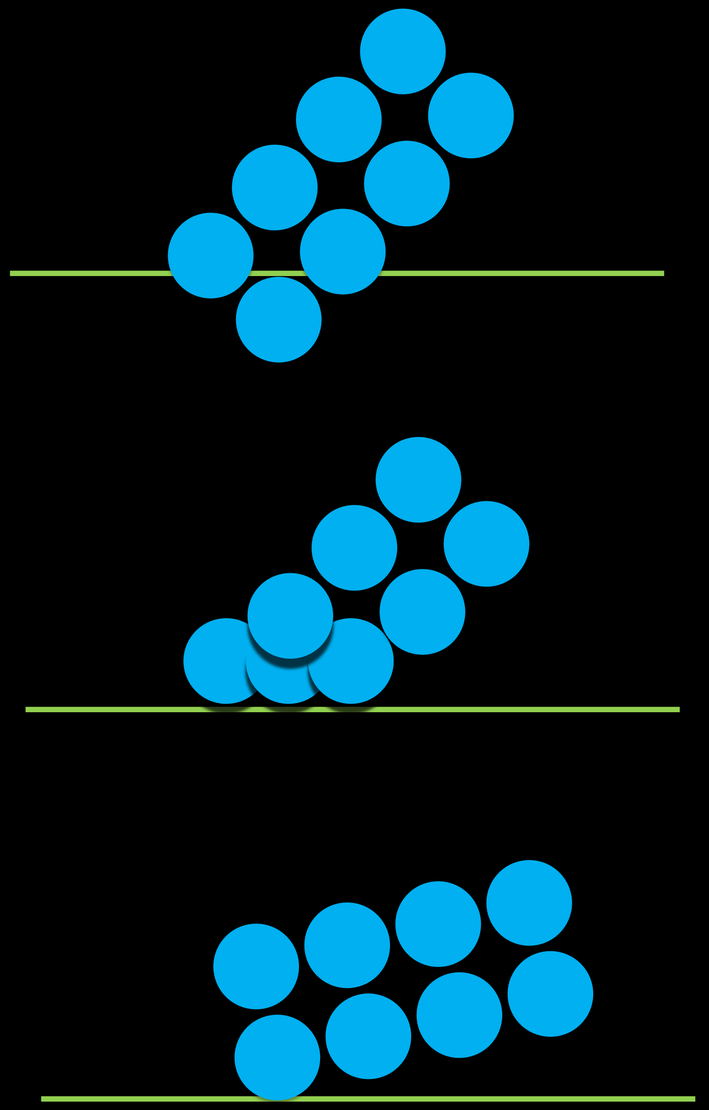


FleX支持的特性

刚体

- 将网格转换为SDF
- 粒子位置在形体内部
- 增加形状匹配的约束
- 在粒子上存储SDF距离和梯度信息





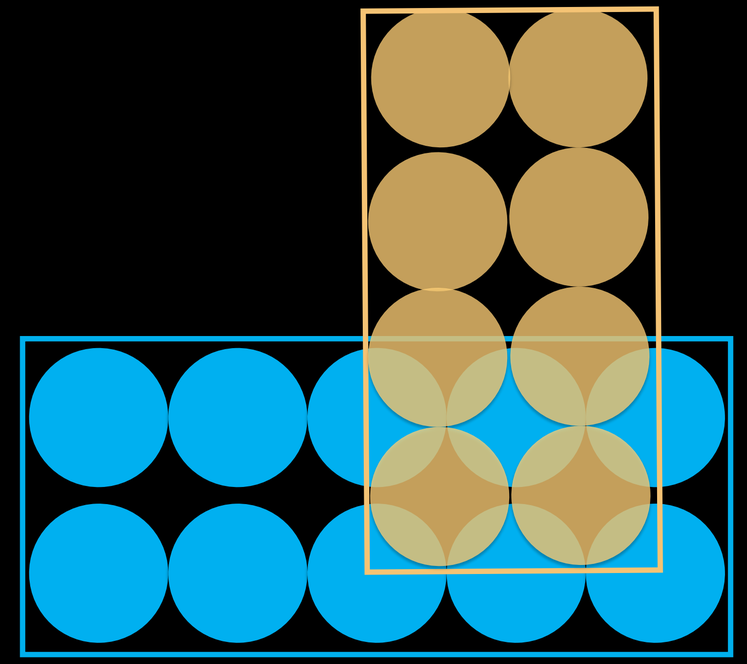
嵌入的状态

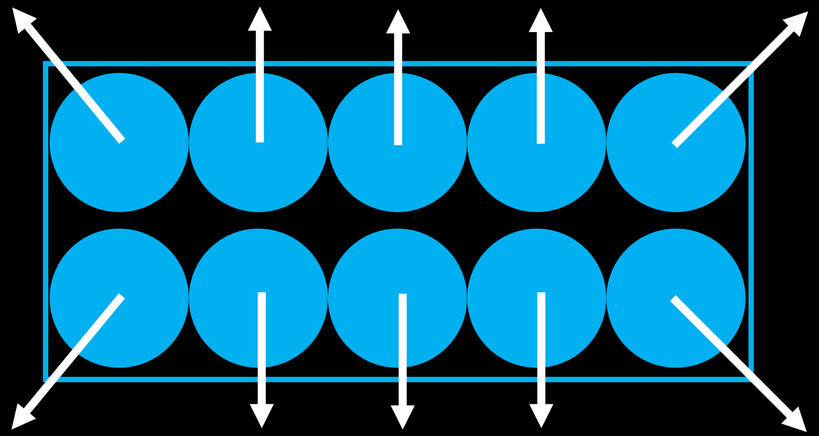
形变的状态（暂不考虑形状约束）

计算形状约束，恢复为刚体形状

刚体碰撞

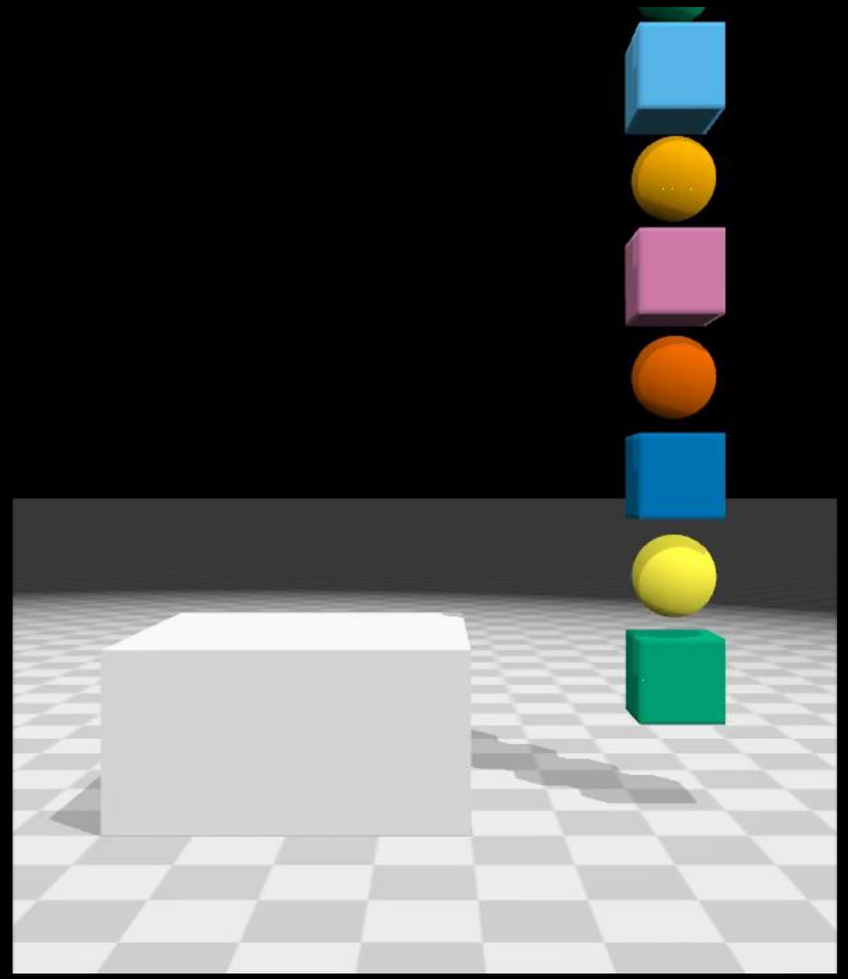
- 仅仅只用粒子间的碰撞不够
- 刚体可能在相互嵌入后卡住
- 刚体碰撞形状内部用粒子上的SDF数据
- 把形状表面的粒子处理为单面的 [Müller & Chentanez 11]





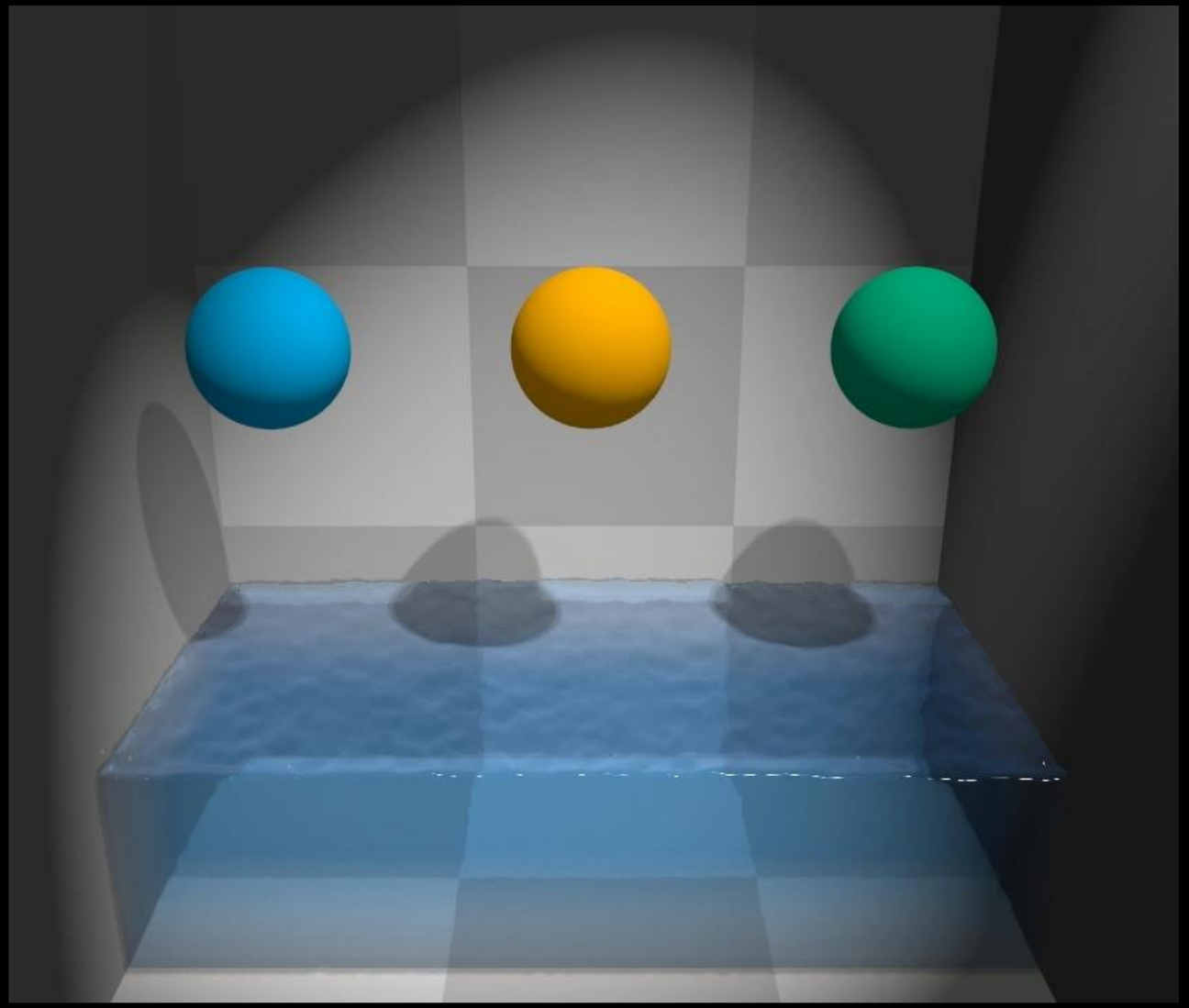
塑料材质形变

- 检测粒子形变是否超过设定的阈值
- 超过则改变粒子的rest configuration，允许形变
- 调整渲染网格（线性蒙皮）



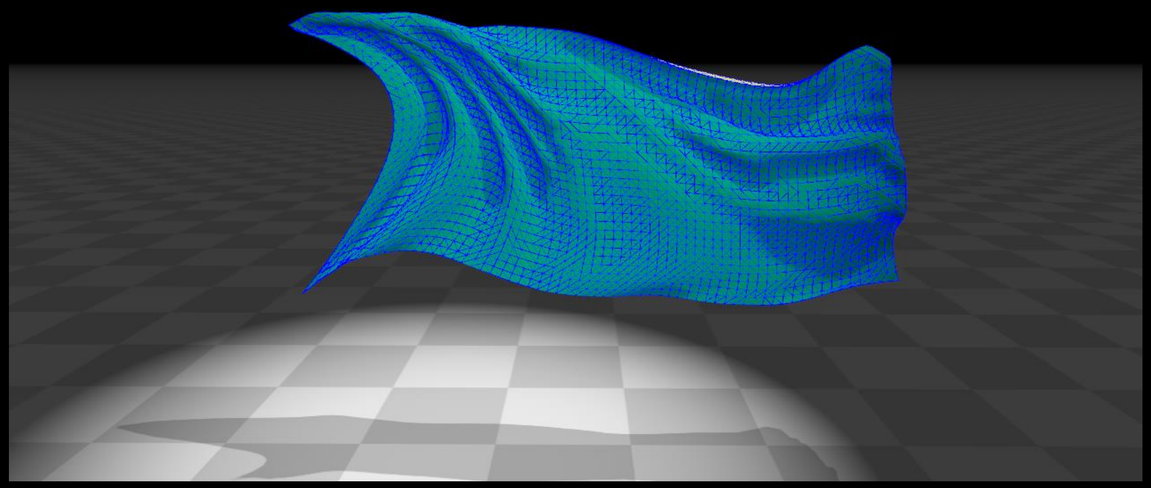
刚体和流体的双向交互

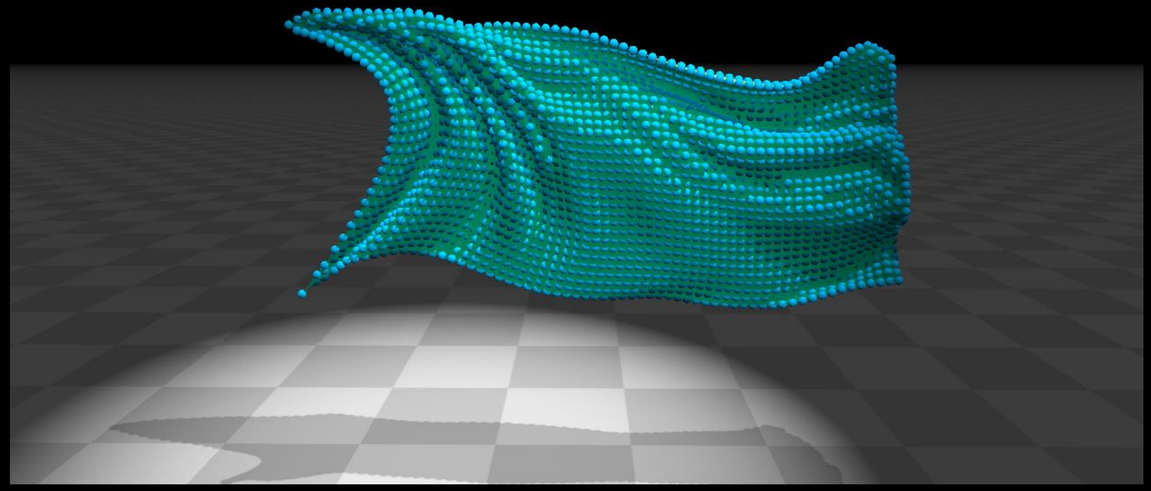
- 原生的支持
- 在估算流体密度时，考虑刚体和流体所有的粒子
- 将流体粒子处理为刚体粒子一样的固体粒子

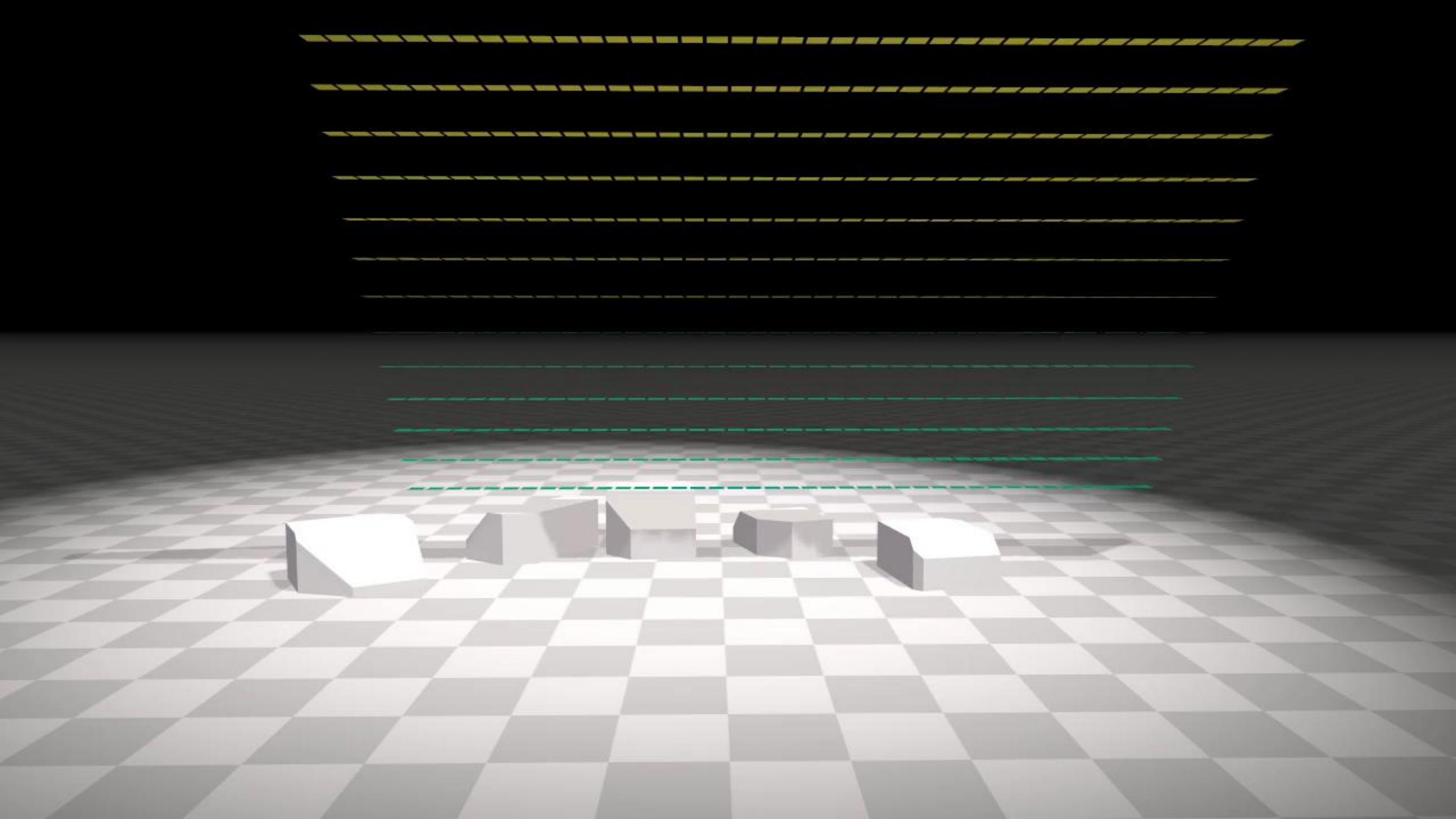


布料

- 距离和系绳约束图
- 增加/删除约束很容易，支持撕裂
- 原生支持布料自碰撞和布料间相互碰撞

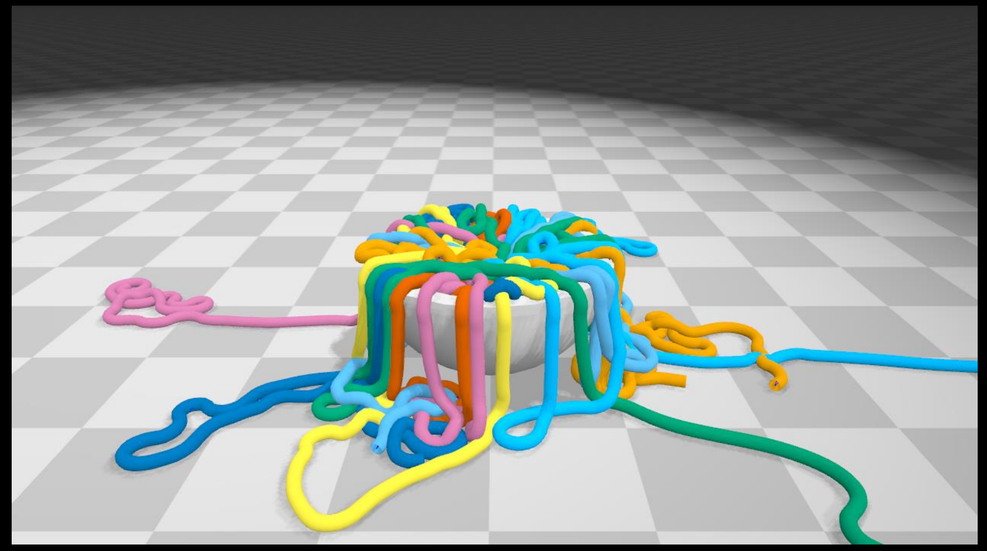


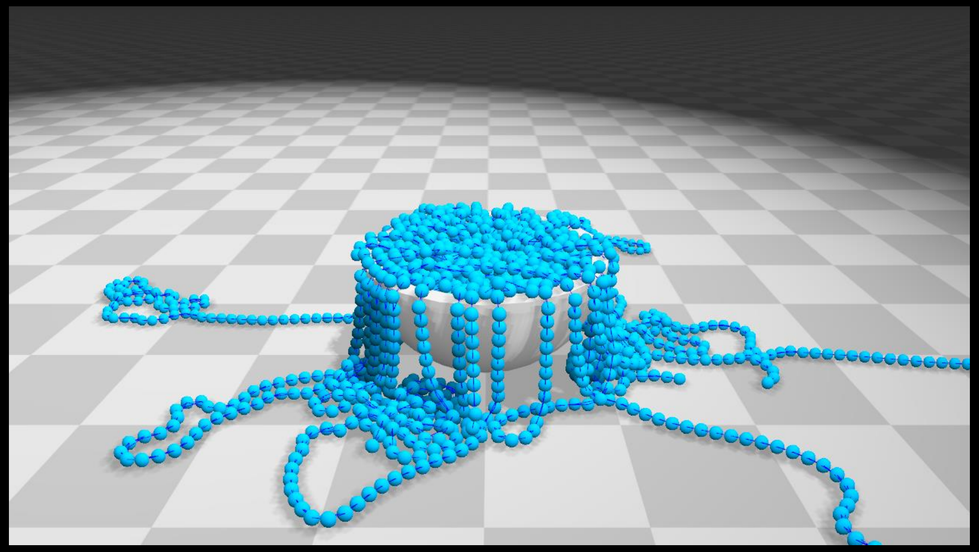


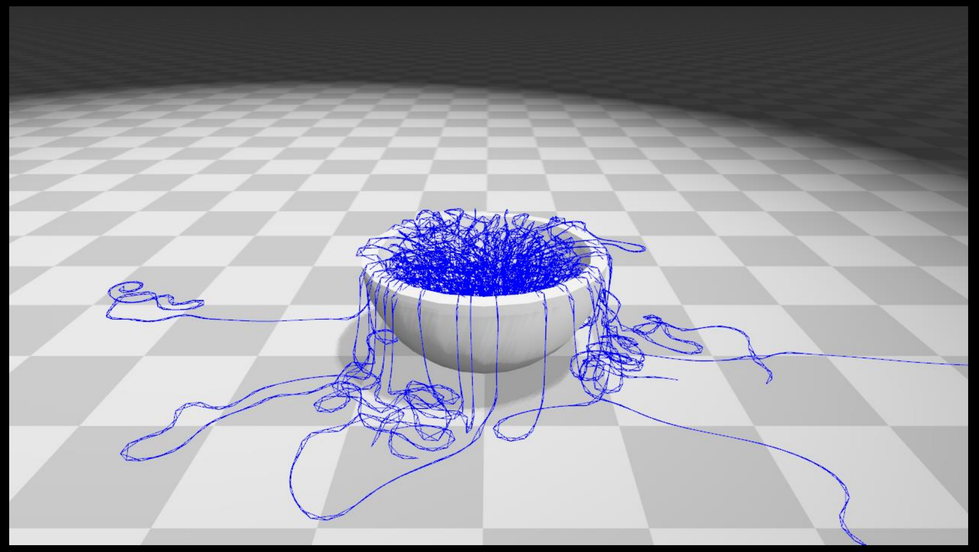


绳索

- 用距离和弯折约束构建
- 采用Catmull-Rom样条平滑粒子位置
- 适合于GPU曲面细分单元
- 不支持扭转约束（需要增加方向）

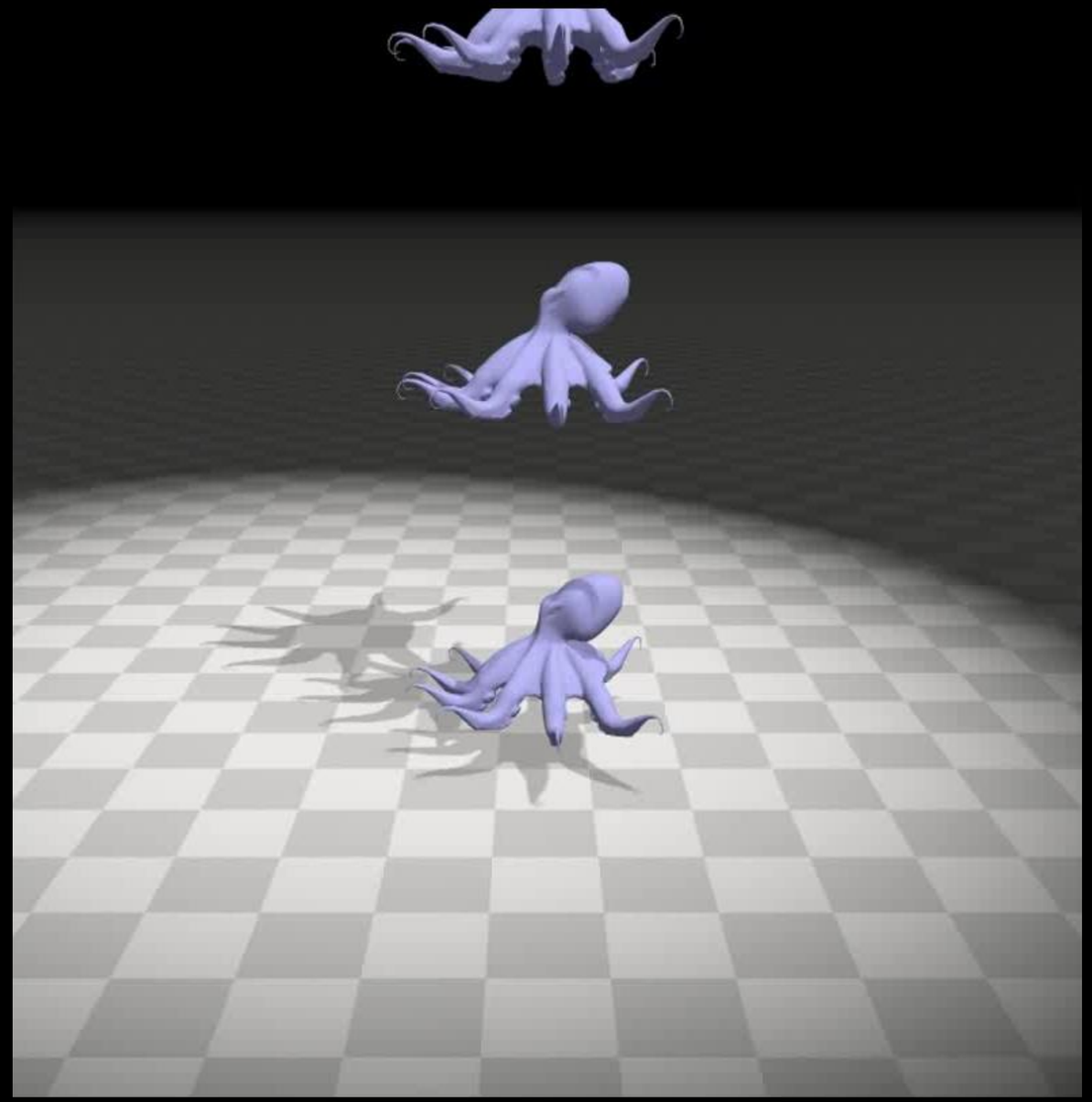




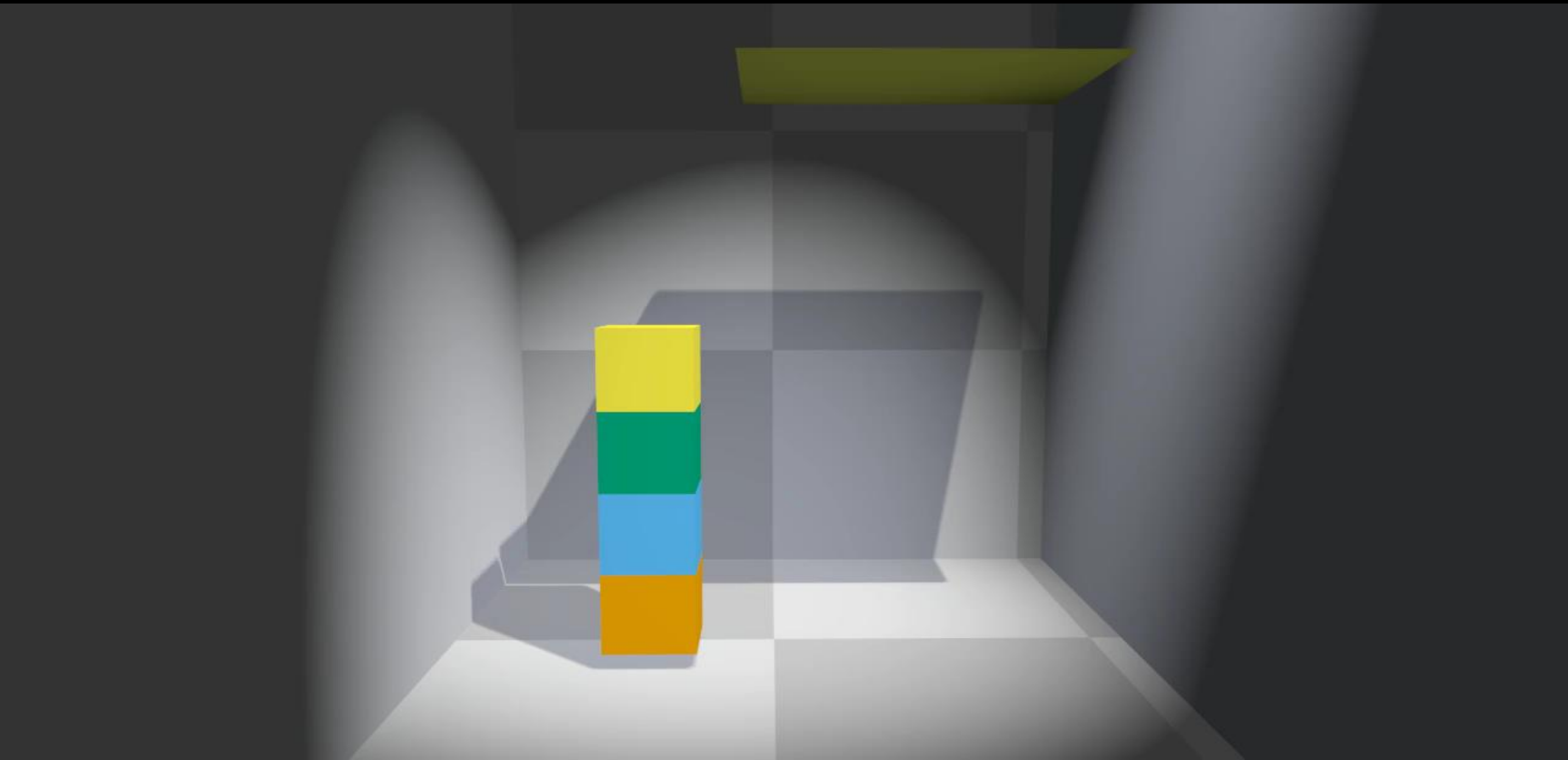


软体

- 四面体网格->质量弹簧系统
- 四面体体积约束
- 柔软物体的形状匹配



气体（未发布）



PhysX Vs. FleX

什么是PhysX

- PhysX致力于助力游戏开发者制作更好的游戏
  - PhysX 是特性完备的物理引擎
  - PhysX 已经成为游戏玩法和视觉特效的核心部分之一
  - PhysX 在所有主流游戏平台都有很好的支持
    - 主机
    - 智能手机
    - PCs，不管有无GPU加速

相同点

- 都是物理模拟引擎
  - 提升交互性，帮助游戏达到更好的效果
- 支持类似的特性
  - 刚体
  - 布料
  - 流体和粒子

不同点

- 平台
  - PhysX：全平台，从智能手机，主机，到PC，包括GPU加速
  - FleX：CUDA
- 解算器
  - PhysX：每个特性一个独立的解算器
  - FleX：统一的解算器
- 游戏逻辑
  - PhysX：对游戏逻辑是友好的
  - FleX：缺乏支持，还需完善（比如：粒子到游戏物体的映射，模拟事件回调）

不同点

- PhysX拥有更多游戏相关的附加功能
  - 人物控制器、关节、车辆控制器
  - 场景查询，比如射线检测
  - 序列化
- FleX可以更自然的处理特性之间的交互
- FleX需要与PhysX结合使用
  - 大规模地形建筑
  - 人物、其他动态物体之间的双向碰撞

FleX集成

FleX的集成

- FleX SDK由两部分组成
  - 核心实现库 （Core）
  - 辅助功能库 （Extensions）
- FleX Solver可以集成进任意编辑工具中
  - UE3/4
  - Max/Maya
  - Standalone

核心实现库

- C-style API
- 一个头文件，一个DLL，flex.h + flexRelease.dll
- 批量操作的API，例如：

```
FLEX_API void flexSetVelocities(FlexSolver* s, const float* v, int n, FlexMemory source);
FLEX_API void flexGetVelocities(FlexSolver* s, float* v, int n, FlexMemory target);

FLEX_API void flexSetPhases(FlexSolver* s, const int* phases, int n, FlexMemory source);
FLEX_API void flexGetPhases(FlexSolver* s, int* phases, int n, FlexMemory target);
```

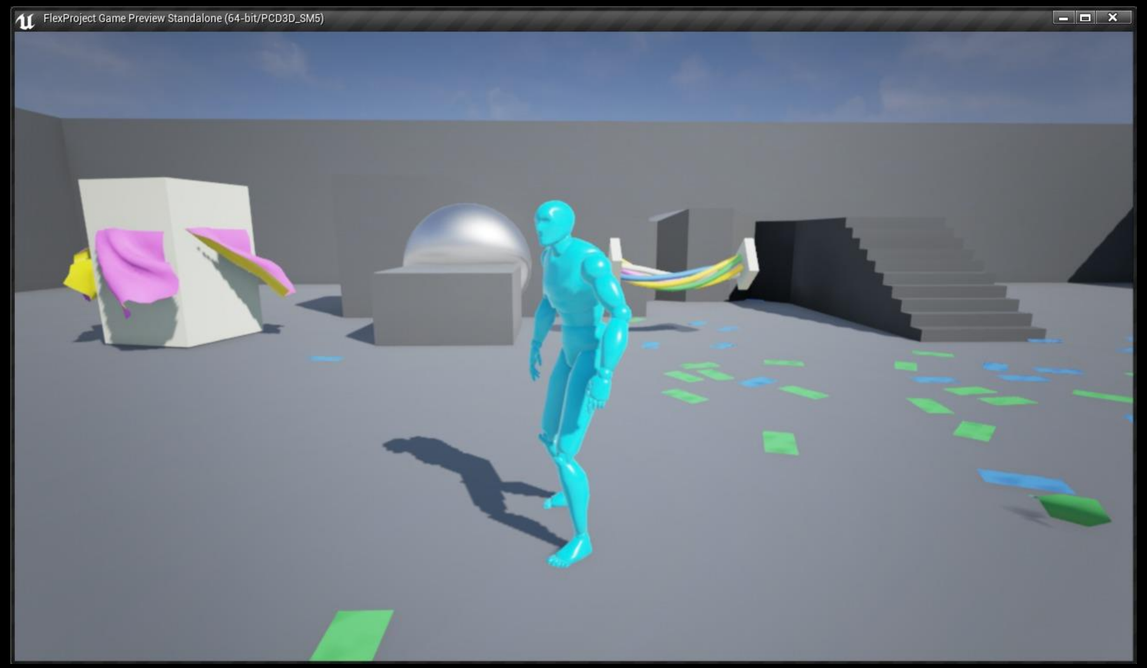
- CUDA实现
- 支持device->device拷贝的互操作

辅助功能库

- C-style API
- 一个头文件，一个DLL，flexExt.h + flexExtRelease.dll
- 提供的功能:
  - 增加和删除粒子
  - 体素化网格，将其转换成粒子表达
  - 创建布料的约束图
  - 创建四面体网格的质量弹簧系统
- 允许开发者按需定义生命周期管理
- 无CUDA代码，仅仅是核心API的封装和扩展

UE3和UE4的集成

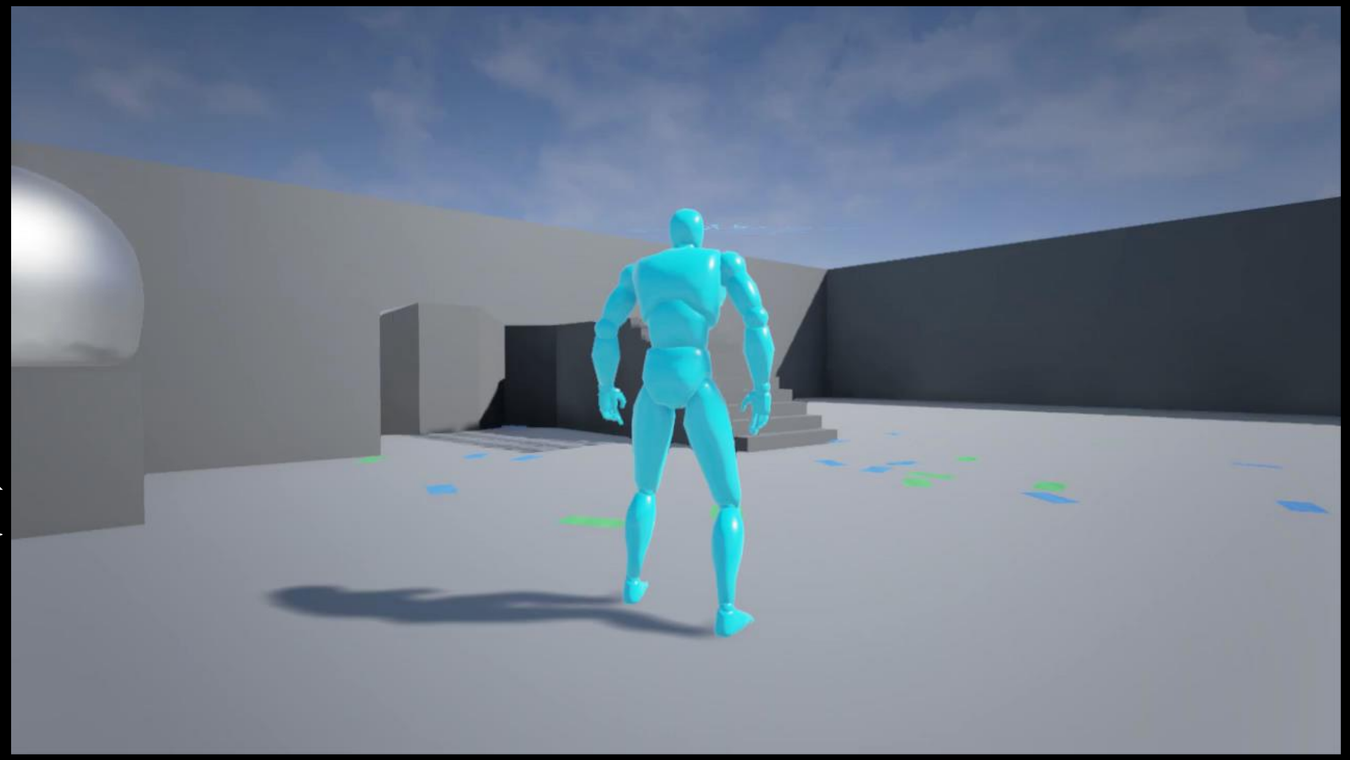
- UE3 and UE4 FleX已经集成完毕
- 蝙蝠侠和Killing Floor2已发布
- 支持下列Component:
  - Cloth, Rigids, Inflatables, Ropes, Fluids, Particles
- UE4集成代码已经发布到Github:



https://github.com/NvPhysX/UnrealEngine/tree/FleX

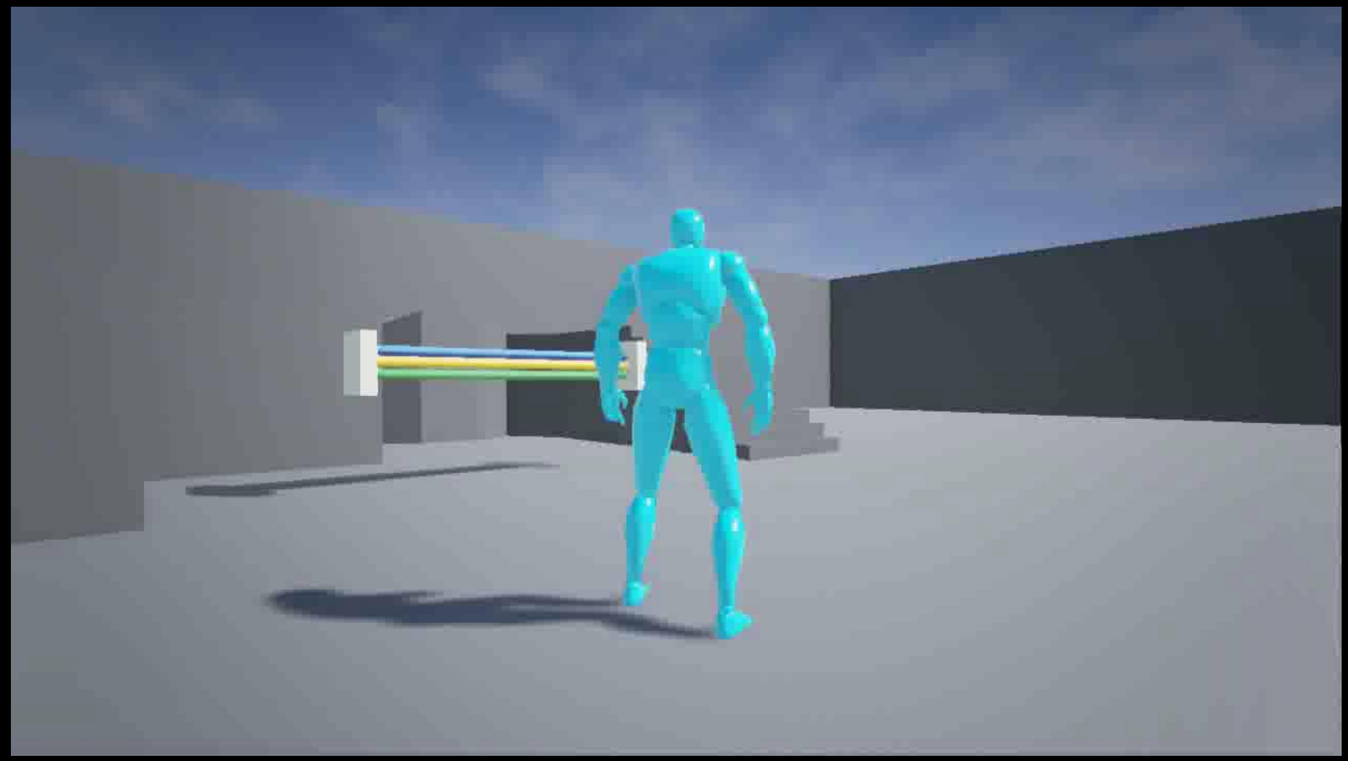
FleX Cloth

- 适合制作环境布料
- 通过CCD避免与其他FleX粒子穿透
- 可自动绑定到静态或动态物体
- 支持充气体（Inflatable）约束



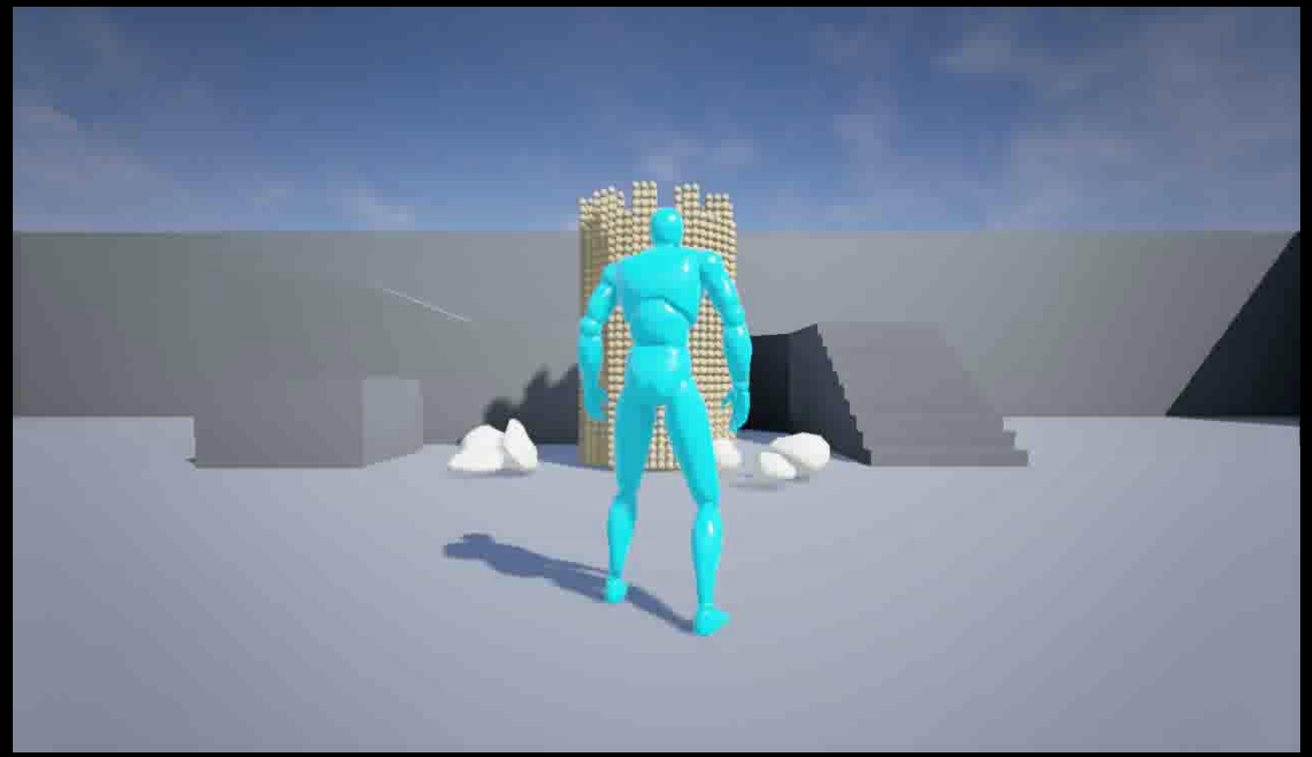
FleX Ropes

- 基于UE内建的UCableComponent
- 支持弯折、自碰撞和场景碰撞
- 未来会支持扭转



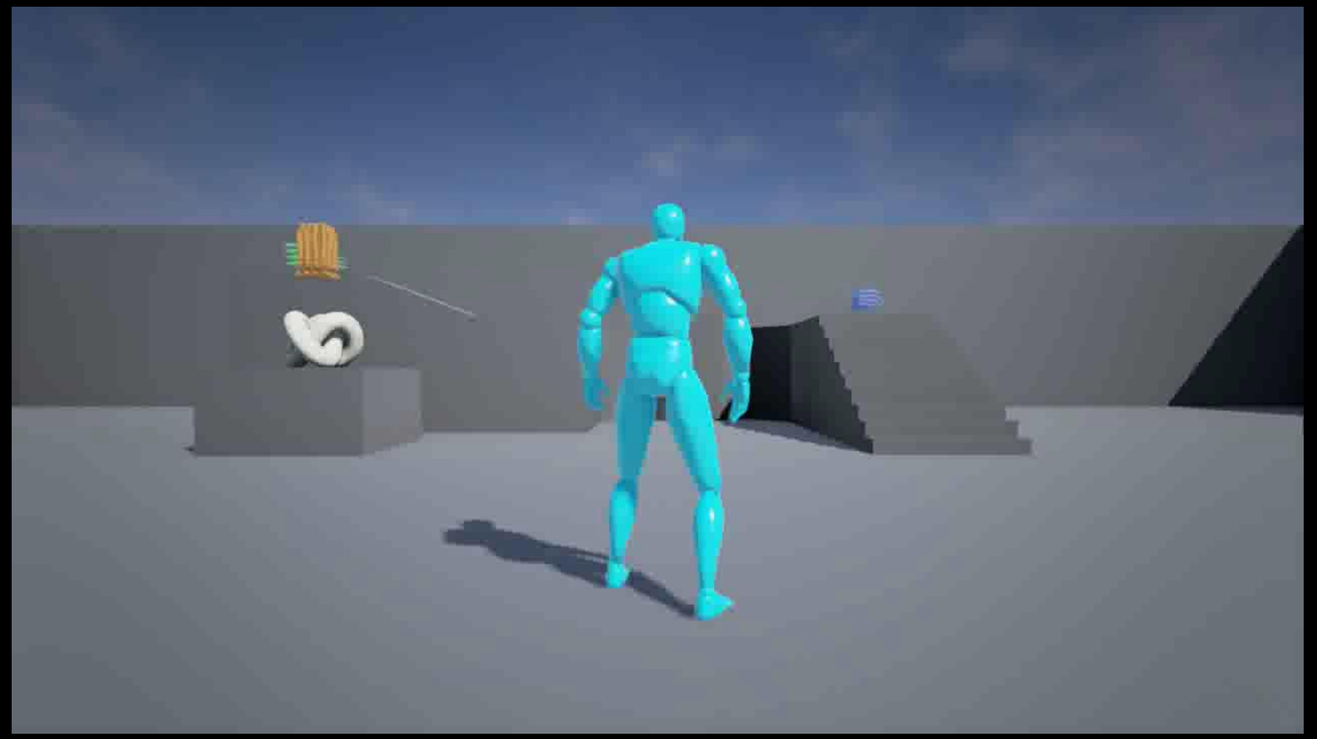
FleX Particles

- 集成在Cascade中
- 增加生成流体的模块
- 增加生成粒子组成的碰撞体的模块
- 增加生成充气体/布料的模块



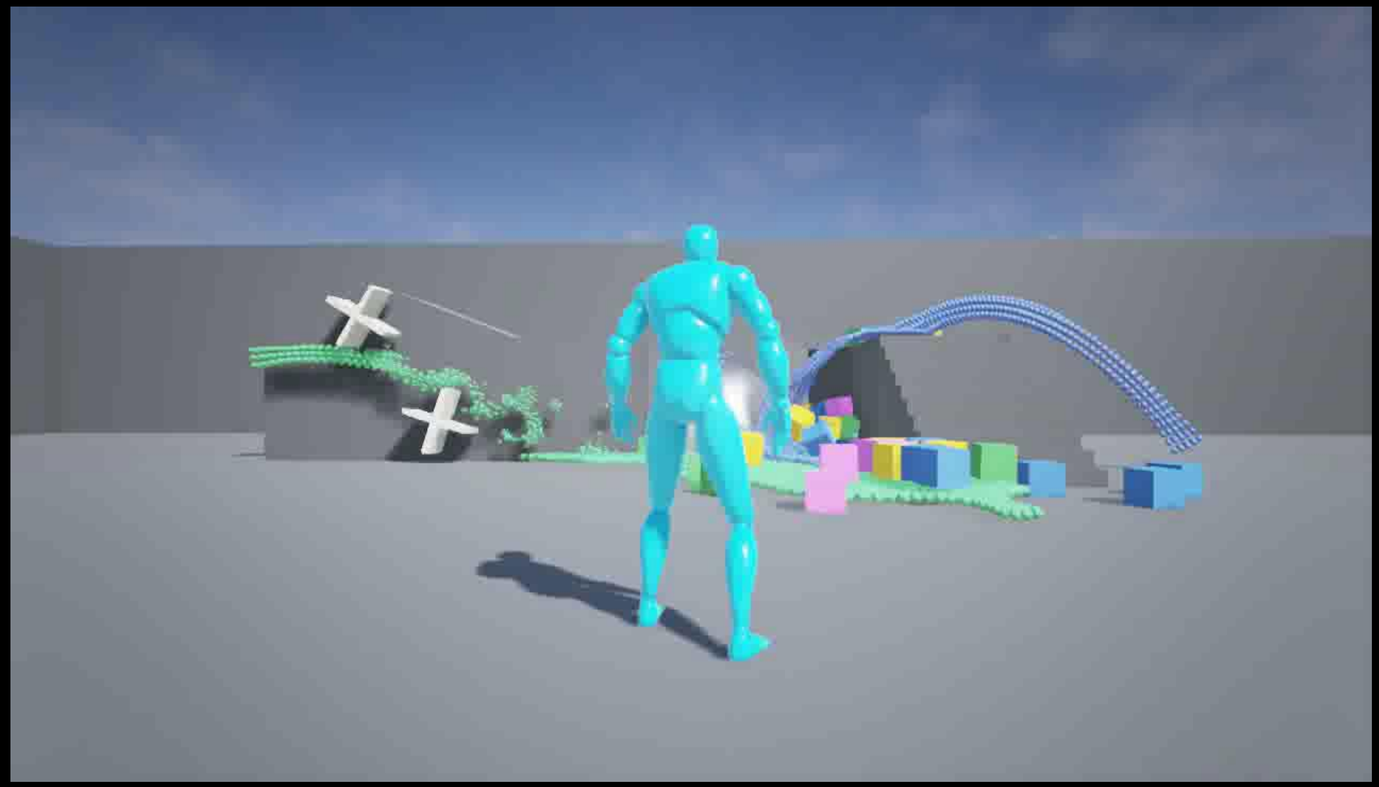
FleX Force Fields

- 集成在UE4 URadialForceComponent中
- 支持Blueprint脚本控制
- 通过FlexExtensions的接口将作用力传入FleX

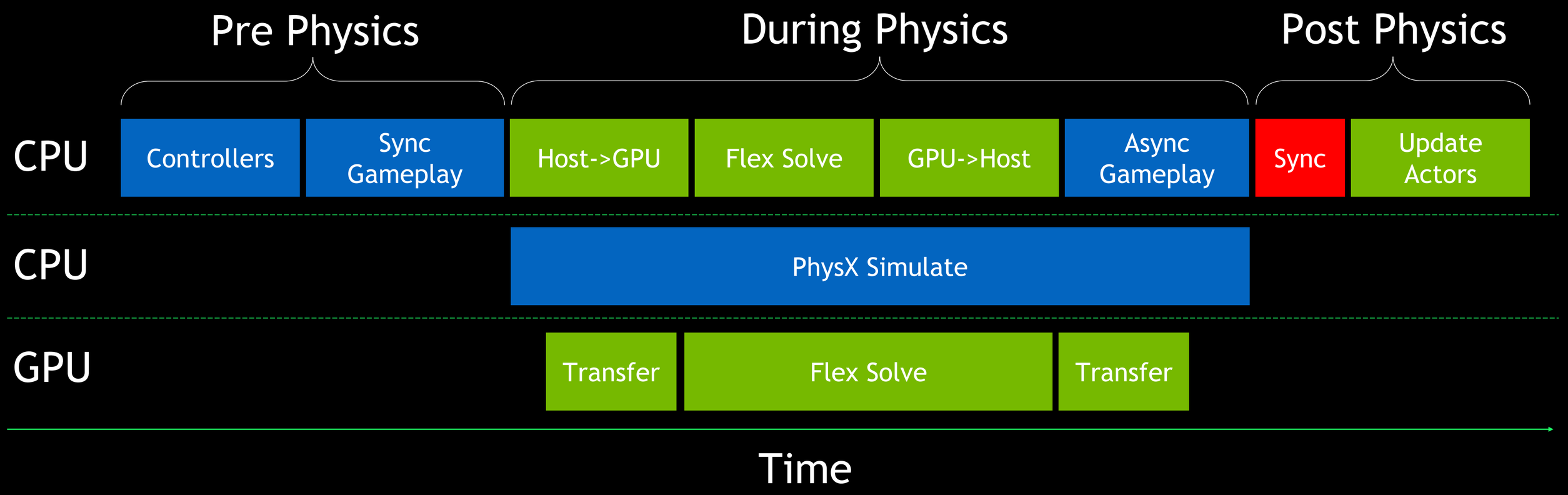


与PhysX的互动

- 已实现基本的FleX和PhysX间的双向交互
- FleX Actor的包围盒传入PhysX场景
- 对每个FleX Actor的包围盒做Overlap检测
- CCT可以与FleX物体碰撞



每帧执行流程


Pre Physics
During Physics
Post Physics
CPU
Controllers
Sync Gameplay
Host->GPU
Flex Solve
GPU->Host
Async Gameplay
Sync
Update Actors
CPU
PhysX Simulate
GPU
Transfer
Flex Solve
Transfer
Time

Game Demo:Killing Floor 2

谢 谢！

Q&A