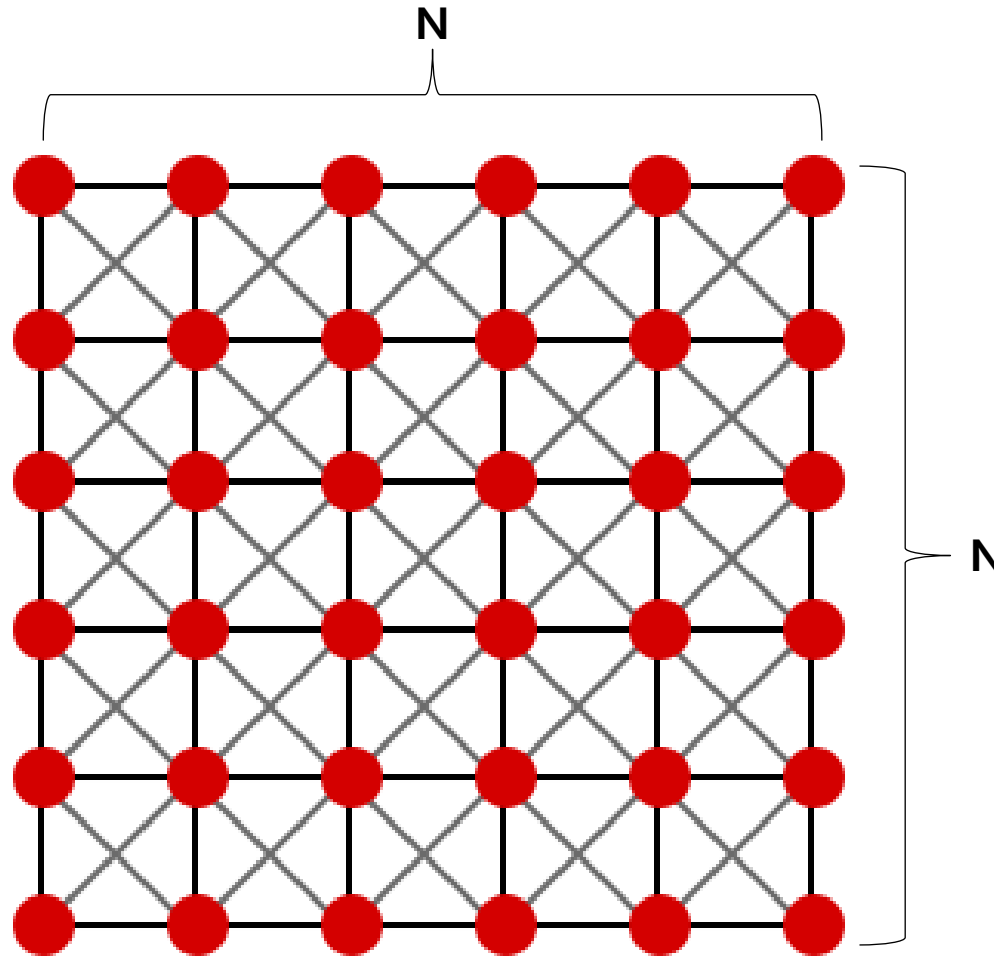


Position Based Dynamics / Compute Shader

OpenGL VI

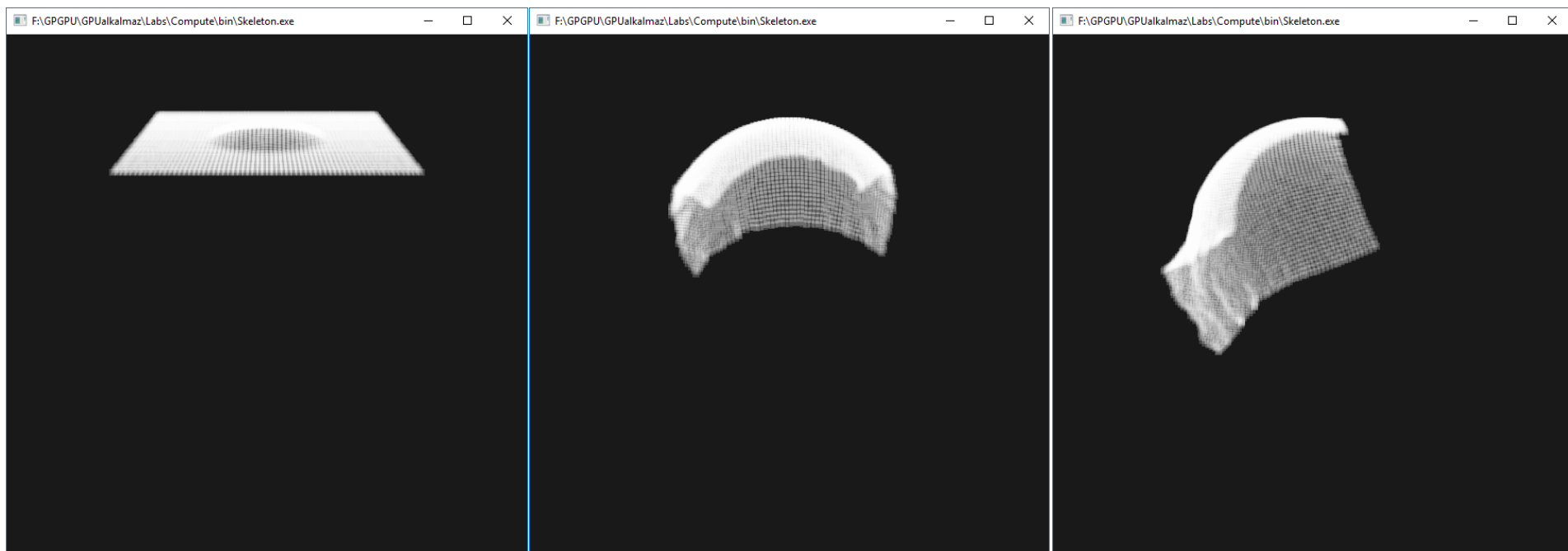
Feladat: Elasztikus anyag szimuláció



- Részecskék $N \times N$ -es hálója
- Összeköttetés:
 - Rugókkal (megpróbálják tartani az eredeti hosszukat)
 - vízszintes és függőleges szomszédok
 - átlós szomszédok
- Ez a háló egy vékony anyagot reprezentál (vastagság nélküli anyagdarab)

Feladat: Elasztikus anyag szimuláció

- Az anyag az XZ síkban van $Y = 0$ helyen
- Ráejtjük a $(0, -0.5, 0)$ középpontú, 0.5 sugarú virtuális gömbre (az anyag lassan lecsúszik)



Position Based Dynamics

Particle:

m_i	mass
\mathbf{x}_i	position
\mathbf{v}_i	velocity

$m=1$

Constraint:

Nálunk ez fix
(gömb, esetleg talaj)

n_j	cardinality
$C_j: \mathbb{R}^{3n_j} \rightarrow \mathbb{R}$	scalar constraint function
$\{i_1, \dots, i_{n_j}\}, i_k \in [1, \dots, N]$	set of indices
$k_j \in [0 \dots 1]$	stiffness parameter
<i>unilateral or bilateral</i>	type

Algorithm:

- (1) forall particles i
- (2) initialize $\mathbf{x}_i = \mathbf{x}_i^0, \mathbf{v}_i = \mathbf{v}_i^0, w_i = 1/m_i$
- (3) endfor
- (4) loop
- (5) forall particles i do $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t w_i \mathbf{f}_{ext}(\mathbf{x}_i)$
- (6) forall particles i do $\mathbf{p}_i \leftarrow \mathbf{x}_i + \Delta t \mathbf{v}_i$
- (7) ~~forall particles i do generateCollisionConstraints($\mathbf{x}_i, \mathbf{p}_i$)~~
- (8) loop solverIterations times
- (9) projectConstraints($C_1, \dots, C_{M+M_{coll}}, \mathbf{p}_1, \dots, \mathbf{p}_N$)
- (10) endloop
- (11) forall particles i
- (12) $\mathbf{v}_i \leftarrow (\mathbf{p}_i - \mathbf{x}_i) / \Delta t$
- (13) $\mathbf{x}_i \leftarrow \mathbf{p}_i$
- (14) endfor
- (15) endloop

gravitáció

Position Based Dynamics GPU

(1) forall particles i

(2) initialize $\mathbf{x}_i = \mathbf{x}_i^0, \mathbf{v}_i = \mathbf{v}_i^0, w_i = 1/m_i$

(3) endfor

(4) loop

(5) forall particles i do $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t w_i \mathbf{f}_{ext}(\mathbf{x}_i)$

(6) forall particles i do $\mathbf{p}_i \leftarrow \mathbf{x}_i + \Delta t \mathbf{v}_i$

~~(7) forall particles i do generateCollisionConstraints($\mathbf{x}_i, \mathbf{p}_i$)~~

(8) loop solverIterations times

(9) projectConstraints($C_1, \dots, C_{M+M_{coll}}, \mathbf{p}_1, \dots, \mathbf{p}_N$)

(10) endloop

(11) forall particles i

(12) $\mathbf{v}_i \leftarrow (\mathbf{p}_i - \mathbf{x}_i) / \Delta t$

(13) $\mathbf{x}_i \leftarrow \mathbf{p}_i$

(14) endfor

(15) endloop

Compute Shaders:

External force shader

Collision constraint shader

Distance constraint shader

(Bending constraint shader)

Final Update shader

(16) Render (vertex, fragment shader)

Skeleton projekt

- Bufferek létrehozva
 - GLuint positionBuffer; //x
 - GLuint positionBufferTmp; //p
 - GLuint velocityBuffer; //v
- Anyag megjelenítése kész
- Iteráció ciklusa üres
- TODO részeket kell kipótolni

External force compute shader

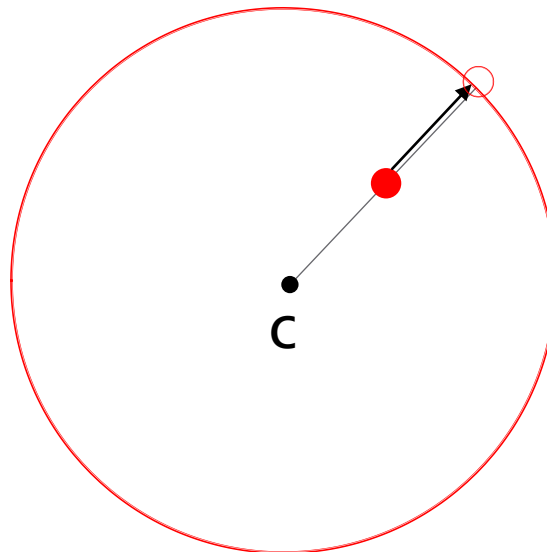
- Minden részecskére le kell futnia
- Minden részecskére
 - $f_i = (0, -0.98, 0)$ //gravity
 - $v_i += f_i * dt$
 - $p_i = x_i + v_i * dt$ //temporary forward step

Final Update Compute Shader

- Minden részecskére le kell futnia
- Minden részecskére
 - $v_i = (p_i - x_i) / dt$ //recalculate velocity from final position
 - $x_i = p_i$ //accept final position

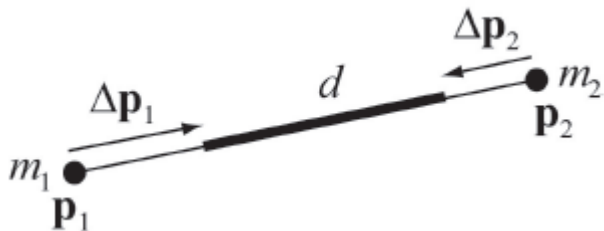
Collision Constraint Shader

- Gömb-részecske ütközés
- Ha a részecske behatolna a gömbbe, rakjuk vissza a gömb felületére
- Constraint weight: ne vigyük ki a felületig, a szükséges eltolás valahány %-át hajtsuk csak végre



Distance Constraint Shader

- Minden részecskére lefut
- Minden szomszédját megvizsgálja (fő irányok és átlók mentén is)
- Megpróbálja visszaállítani a kezdeti távolságot a szomszédjával
- Tehát vagy a szomszédja irányába indul, vagy tőle távolodik.
- Nem az egész távolságot korrigálja, csak a felét, mert a másik részecske is javítani fog
- ConstraintWeight: még annál is kevesebbet javítunk: a szükséges javítás valahány %-át (lehet külön súly az átlós irányokra)



$$dp_1 = (p_2 - p_1) / |p_2 - p_1| * (|p_2 - p_1| - d) * 0.5$$

(Bending Constraint Shader)

- Opcionális
- Egyszerűsített megoldást tudunk csak hirtelen berakni
- A részecskék megpróbálják megakadályozni, hogy meghajoljon az anyag az ő pontjukban
- Vízszintes, függőleges és átlós irányokban levetítjük a szomszédok által meghatározott egyenesre

