

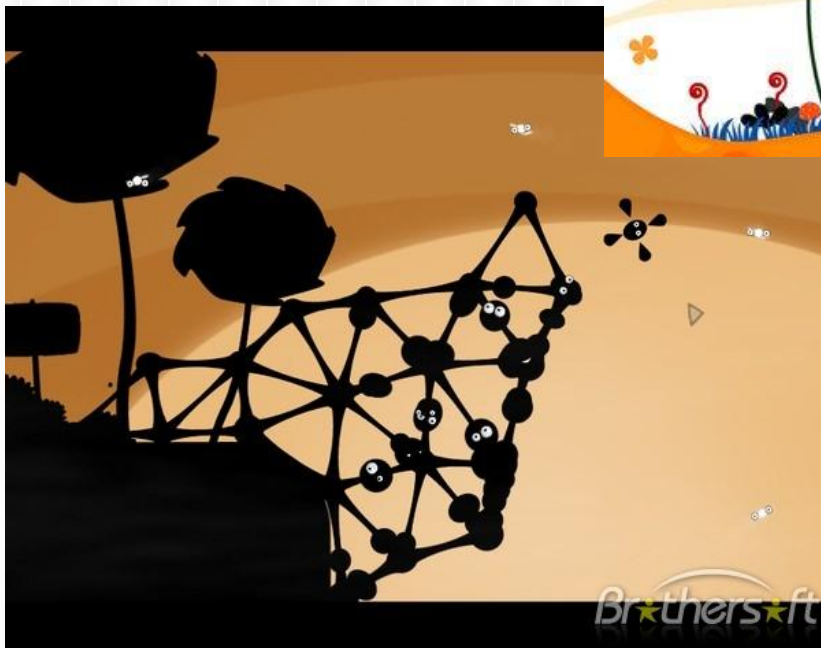
Proste symulacje

.....
Voxel Meeting #3

Maciej Matyka, 2012-12-13

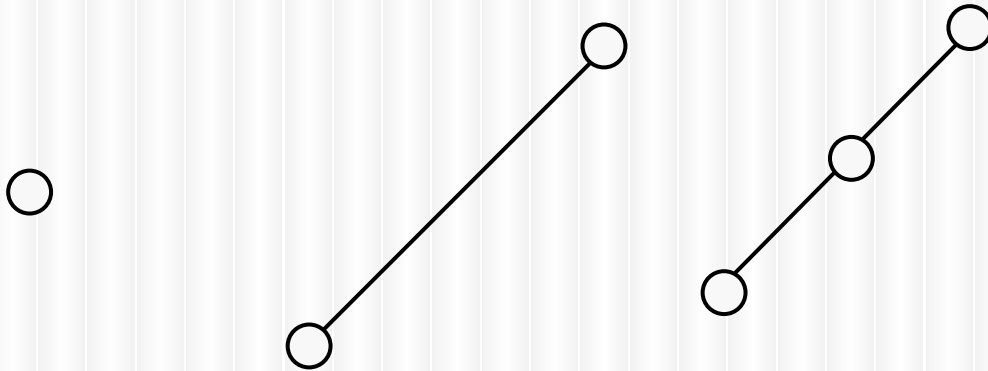
Fizyka czasem się przydaje

.....



O czym będzie ten wykład?

.....



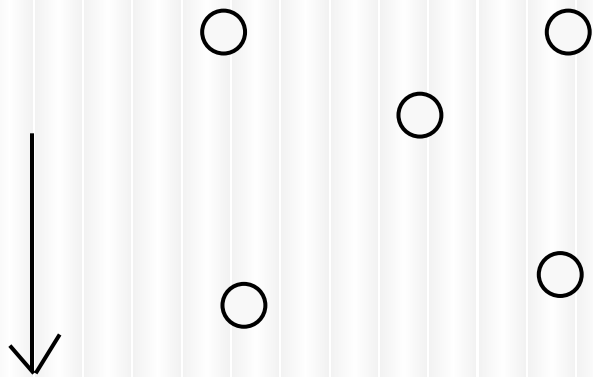
Jak zbudować układy mas połączonych sprężynkami?

Jak je opisać w programie?

Jak wyznaczyć siły, a następnie ruch?

Co ciekawego można z tego zbudować?

Punkty w polu grawitacyjnym



Punkt materialny

.....

Punkt materialny cechują:

- masa (kg)
- pozycja (m)
- prędkość (m/s)
- siła ($\text{kg} \cdot \text{m/s}^2$)
- ...

```
struct point
{
    float m;
    CVector3 r;
    CVector3 v;
    CVector3 f;
};
```

Prosty przepis

1. Inicjalizuj punkty
2. Policz siły
3. Przesuń punkty
4. Więzy, kolizje, korekty, odbicia..
5. Wróć do 2.

Prosty przepis

.....

1. Inicjalizuj punkty
2. Policz siły
3. Przesuń punkty
4. Więzy, kolizje..
5. Wróć do 2.

```
void initpoints(void)
{
    for(int i=0; i<N; i++)
    {
        float r1 = rand()/float(RAND_MAX);
        float r2 = rand()/float(RAND_MAX);
        float r3 = rand()/float(RAND_MAX);
        float r4 = rand()/float(RAND_MAX);

        points[i].v = CVector3( r1 , r2 , 0);
        points[i].r.x = r3;
        points[i].r.y = r4;
        points[i].m = 5;
    }
}
```

Prosty przepis

.....

1. Inicjalizuj punkty
- 2. Policz siły**
3. Przesuń punkty
4. Więzy, kolizje...
5. Wróć do 2.

```
CVector3 g = CVector3(0,-9.8,0);
```



```
// gravity  
for(int i=0; i<N; i++)  
{  
    points[i].f = g * points[i].m;  
}
```

Metoda Verleta

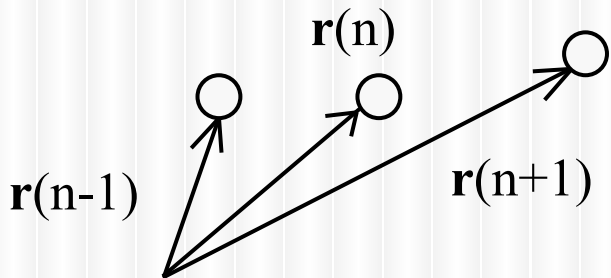
.....

Chcemy przesunąć punkt materialny w nową pozycję $\mathbf{r}(n+1)$

Znamy jego starą pozycję $\mathbf{r}(n-1)$, bieżącą $\mathbf{r}(n)$ oraz siły $\mathbf{f}(n)$, które na niego działają.

Najprostszym sposobem jest wykorzystanie jedno krokowej metody Verleta:

$$\mathbf{r}(n+1) = 2*\mathbf{r}(n) - \mathbf{r}(n-1) + h*h*\mathbf{f}(n) / m$$



Brakuje tylko początku

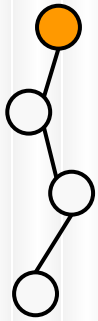
.....

Pierwszy krok w metodzie Verleta (np.) :

$$r(1) = r(0) + h * h * \mathbf{f}(0) / m$$

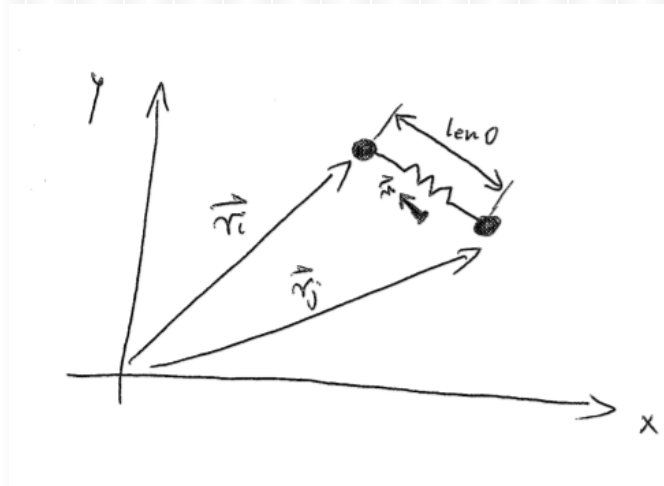
(przykład)

Punkt zawieszony na linie



Siła między punktami: sprężystość + tłumienie

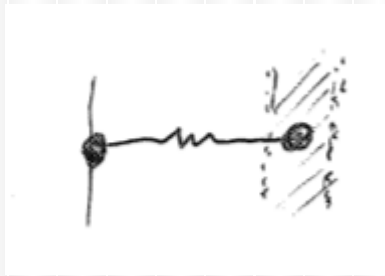
$$f(i,j) = -k_s * (|r_i - r_j| - len_0) * n + k_d * |v_2 - v_1| * n;$$



Więzy związane z odległością

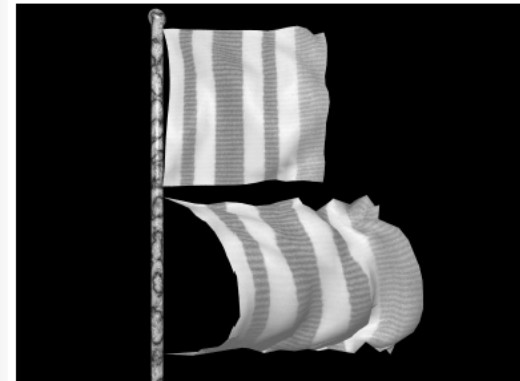
.....

Inverse dynamics constraints (Xavier Provot, INRIA):

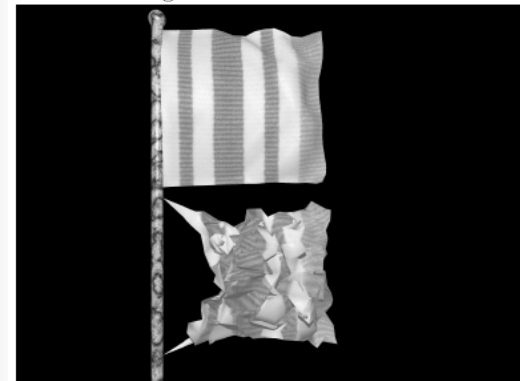


punkty odchylają się tylko o pewien procent od położenia równowagi.

Prosta implementacja - znaczący wzrost stabilności układu



(a) Elastic flag and our “semi-rigid” flag when stiffness is low.



(b) Increasing stiffness: elastic flag becomes chaotic and “semi-rigid” flag remains stable.

Figure 6: Comparison for the case of a flag.

Wracając do zawieszonej liny...

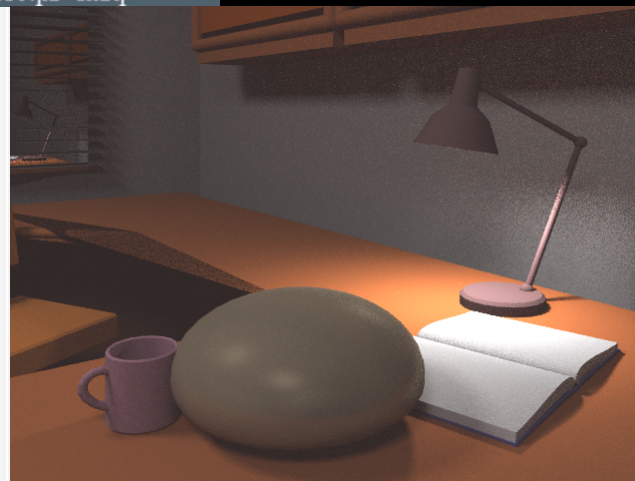
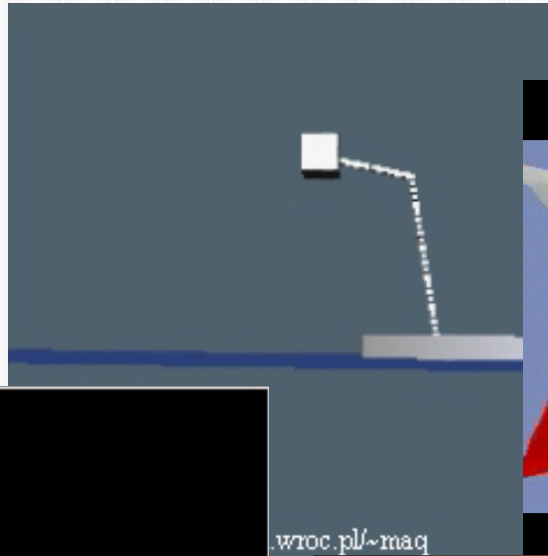


(program)

Przykłady zastosowań

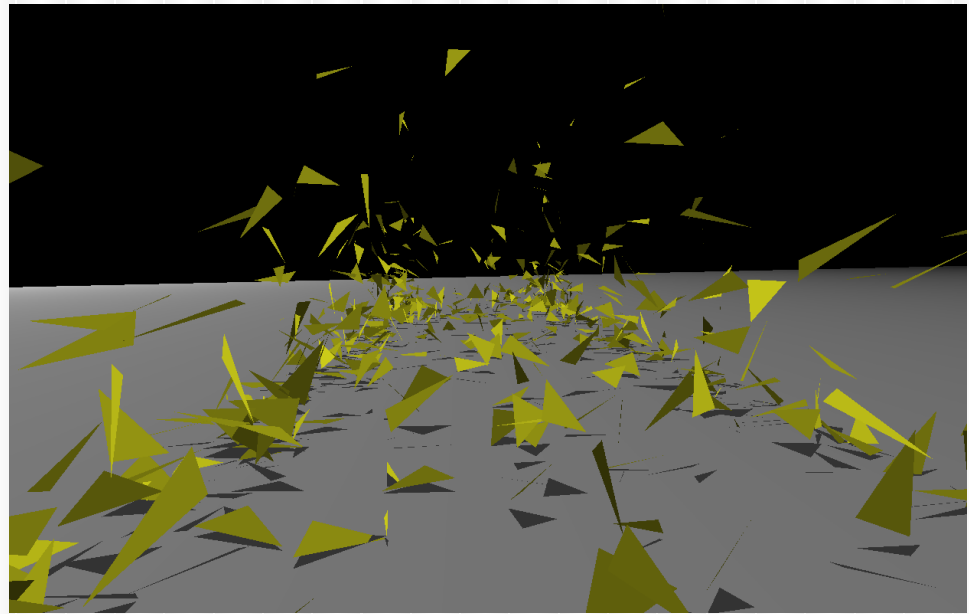
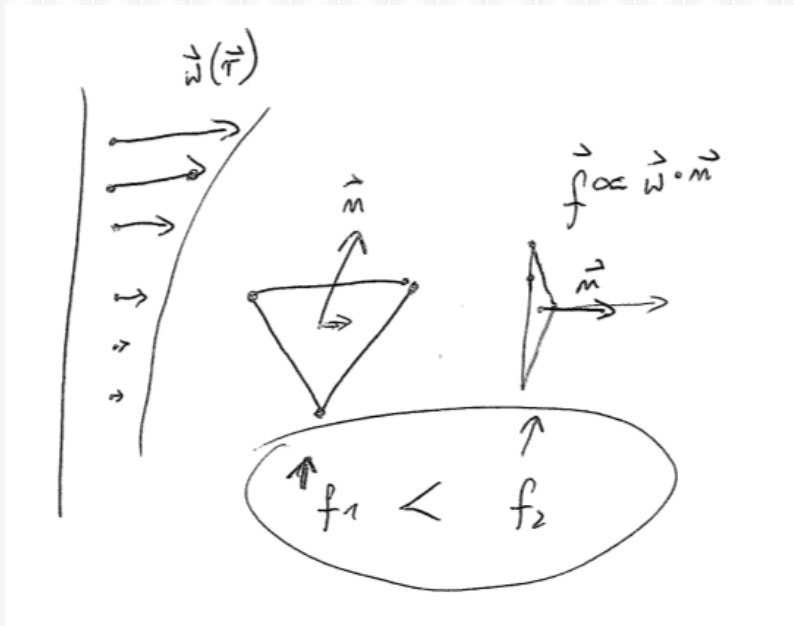
.....

- postacie
- tkaniny
- ciała miękkie



Uproszczona aerodynamika

Siła proporcjonalna do rzutu wektora przepływu na wektor normalny ścianki:



Ciekawa literatura

.....

- Advanced Character Physics (Thomas Jakobsen)

<http://www.pagines.ma1.upc.edu/~susin/files/AdvancedCharacterPhysics.pdf>

- Wejchert, J. and Haumann, Animation Aerodynamics, Computer Graphics (ACM)

Koniec



Dziękuję za uwagę

maciej.matyka@gmail.com

<http://panoramix.ift.uni.wroc.pl/~maq/>