



MATEMATIKA V ANIMOVANÉM FILMU

Milan Jaroš
ŠKOMAM 2020

MATEMATIKA V ANIMOVANÉM FILMU



- | Funkce a jejich grafy
- | Matematický obraz
- | Procedurální textura
- | Renderování

MATEMATIKA V ANIMOVANÉM FILMU



„Je důležité si uvědomit, že umělci a designéři vidí tvary a obrazy, kdežto počítače pracují s čísly a rovnicemi. K propojení těchto dvou světů se využívá exaktní věda, kterou nazýváme **matematika**.“

Tony DeRose
Disney · PIXAR

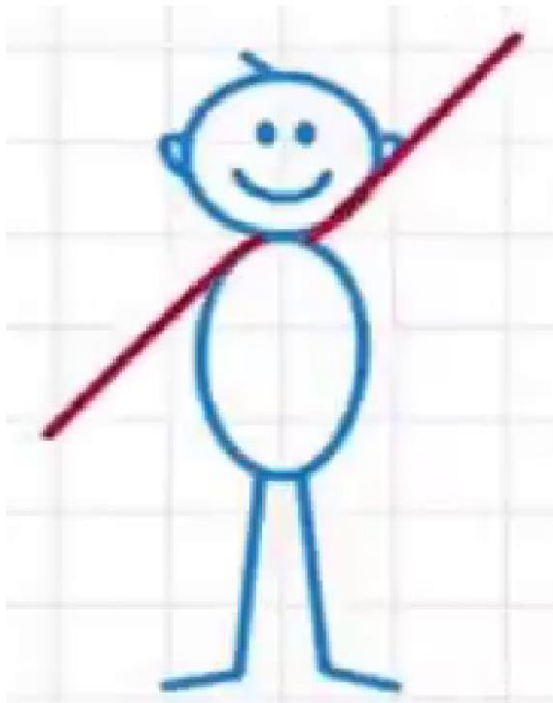
FUNKCE A JEJICH GRAFY



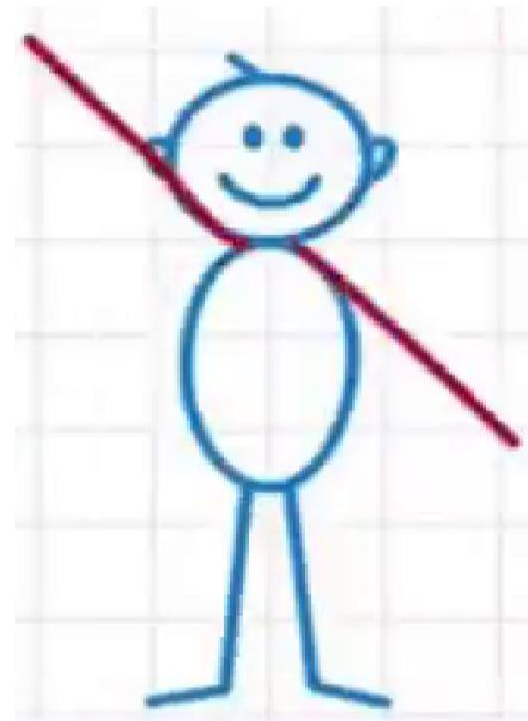
| Lineární funkce / Směrnice tvar rovnice přímky

$$y = kx + q$$

$$y = x$$



$$y = -x$$

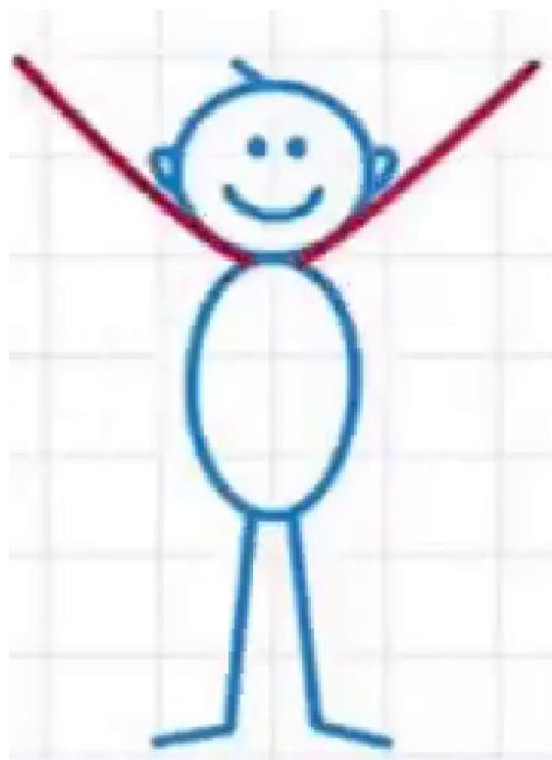


FUNKCE A JEJICH GRAFY

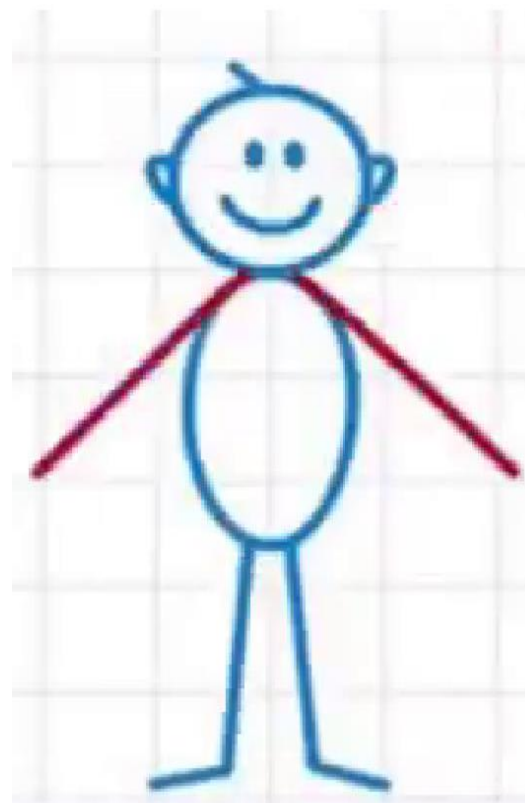


| Absolutní hodnota

$$y = |x|$$



$$y = -|x|$$

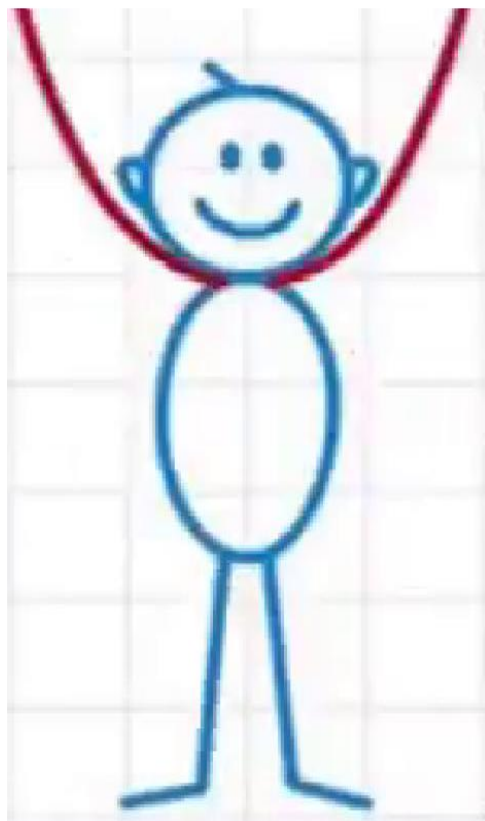


FUNKCE A JEJICH GRAFY



| Druhá a třetí mocnina

$$y = x^2$$



$$y = x^3$$

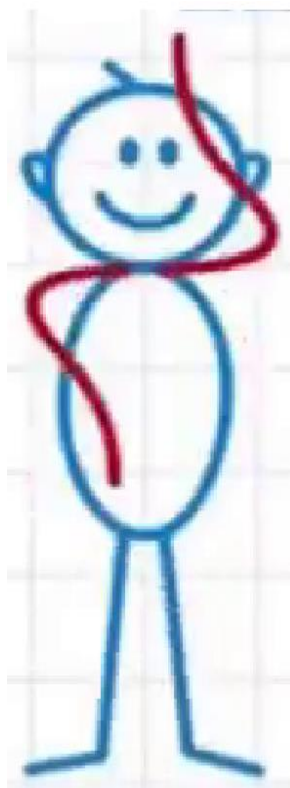


FUNKCE A JEJICH GRAFY

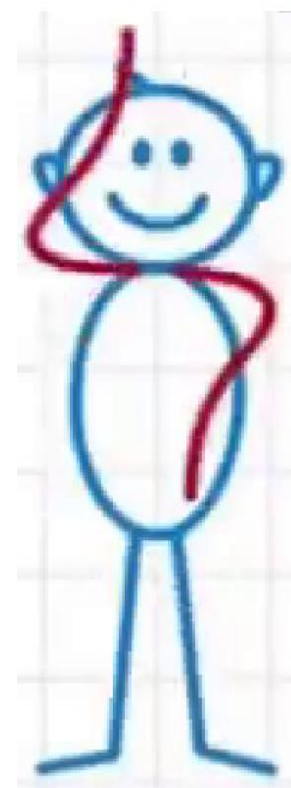


| Převrácená hodnota

$$y = \frac{1}{x}$$



$$y = -\frac{1}{x}$$

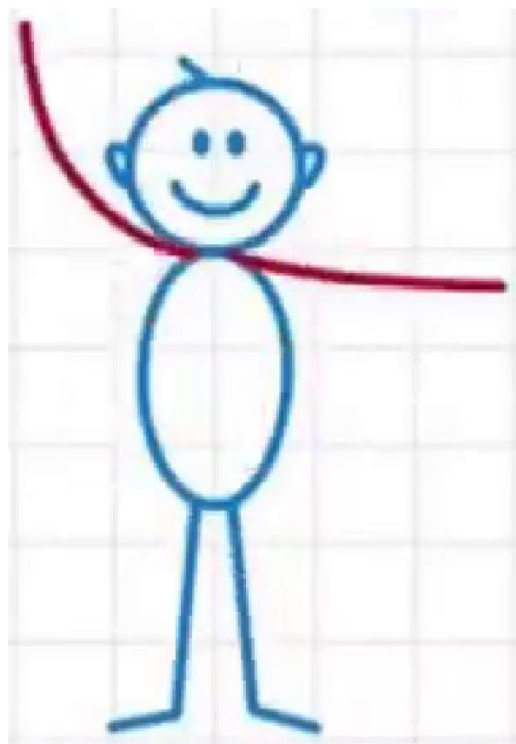


FUNKCE A JEJICH GRAFY

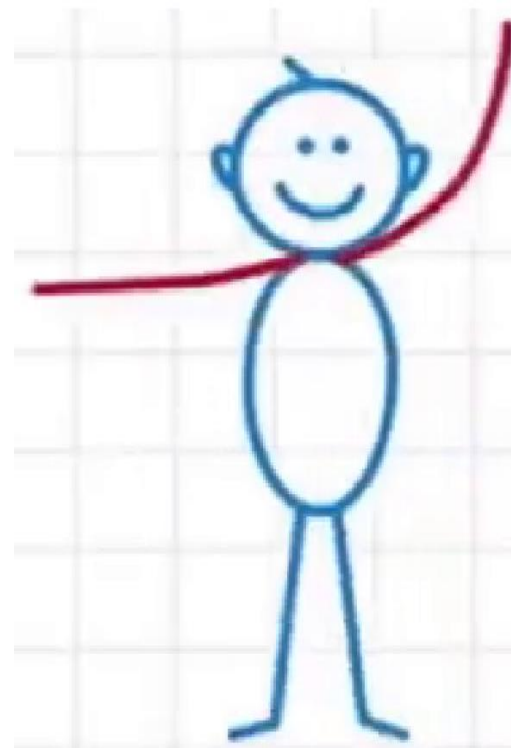


| Logaritmická a exponenciální funkce

$$y = \log_a x$$



$$y = a^x$$

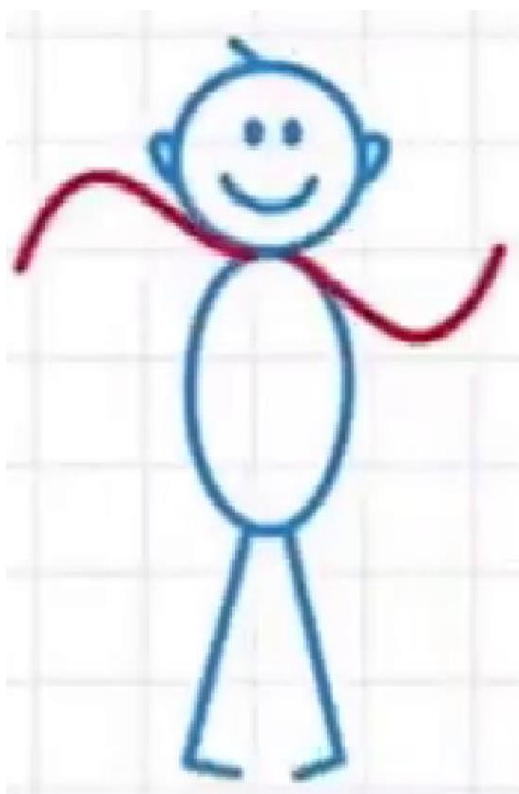


FUNKCE A JEJICH GRAFY

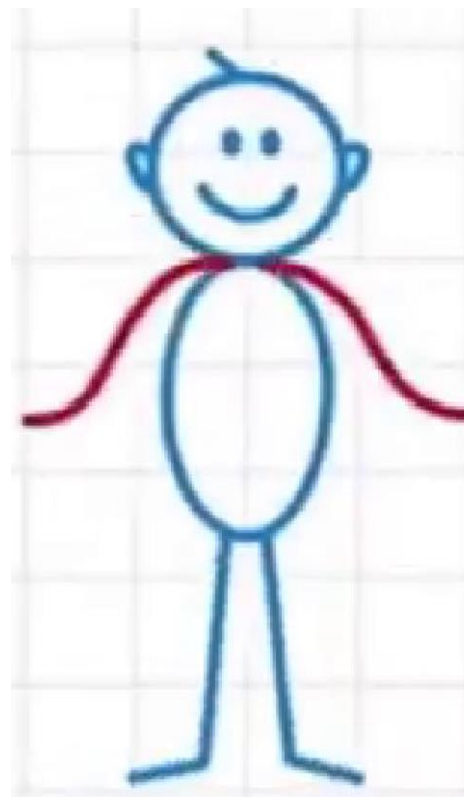


| Goniometrické funkce

$$y = \sin x$$



$$y = \cos x$$

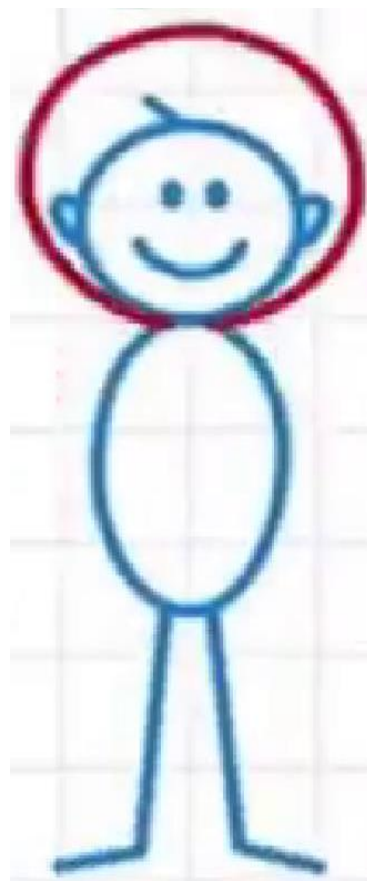


FUNKCE A JEJICH GRAFY

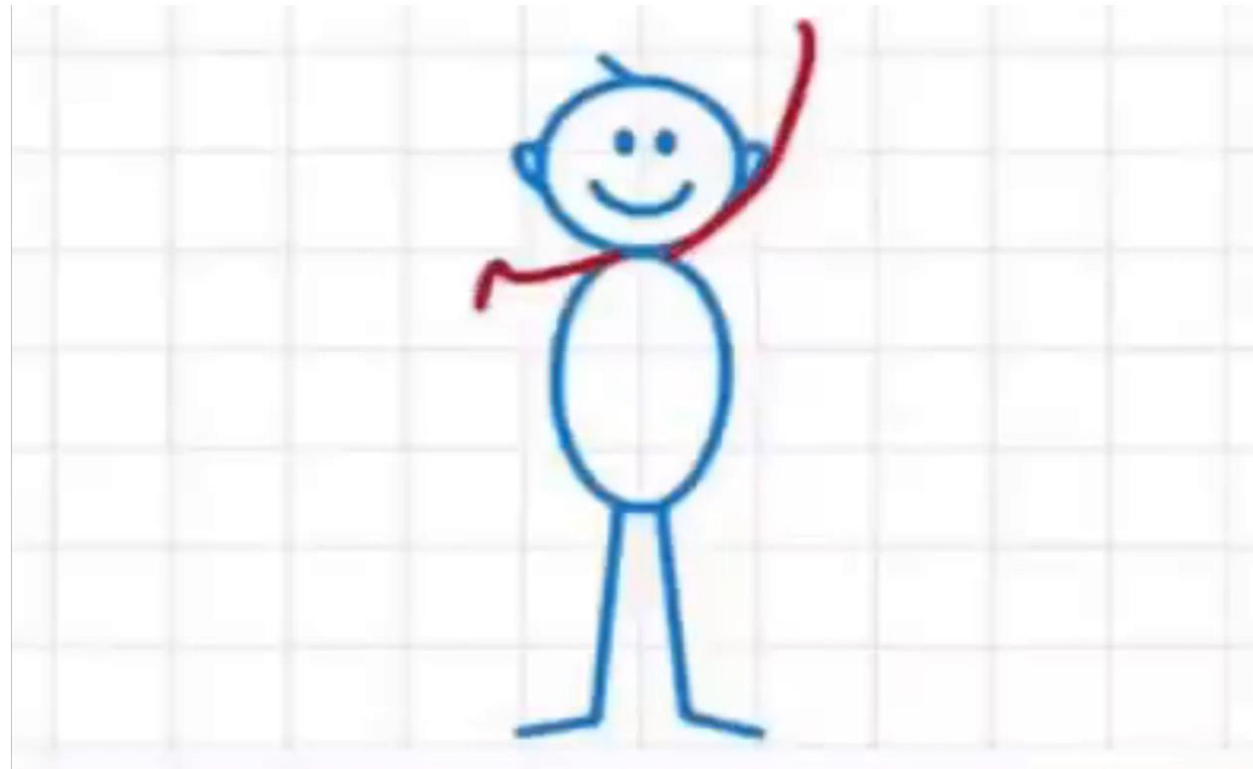


| Středová rovnice kružnice

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$



ANIMACE - MATH DANCE



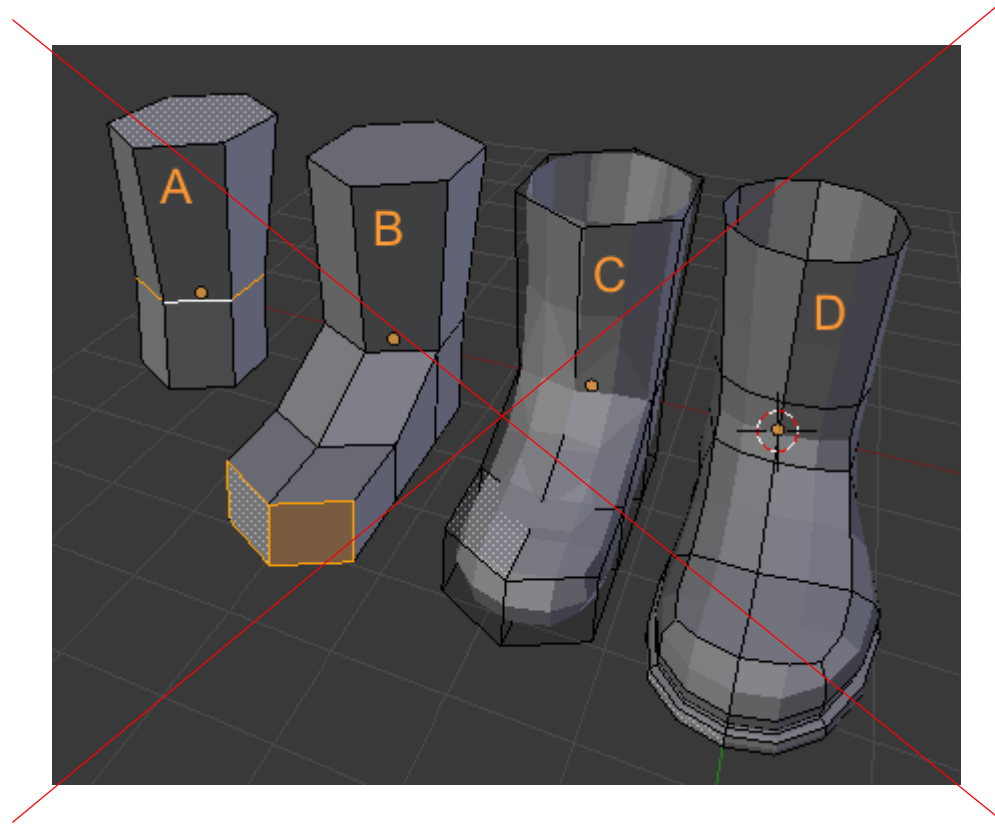
<https://www.youtube.com/watch?v=oeCc6A2WYfQ>

MATEMATIKA A SHADERY



| Matematický obraz

- | Žádná mesh, žádné modely, žádný renderer, žádné globální osvětlení.
- | Matematické výrazy definují tvary, umístění, barvu a světlo, stíny, pohyb.
- | „Batmanův vzorec“ pro dospělé (Inigo Quilez).



<https://s1135.photobucket.com/user/cabby24/media/Blender%20Pics%202012/shoe.png.html?src=pb>

MATEMATICKÝ OBRAZ



Inigo Quilez

<https://www.shadertoy.com/view/4tByz3>

MATEMATIKA A SHADERY



| Shader

- | = počítačový program, který je součástí vykreslovacího řetězce (renderingu)
- | Druhy: vertex, pixel, geometry shader, ...
- | Tvorba za pomoci: programovacích jazyků (GLSL pro OpenGL), grafického rozhraní (node editor v Blenderu)

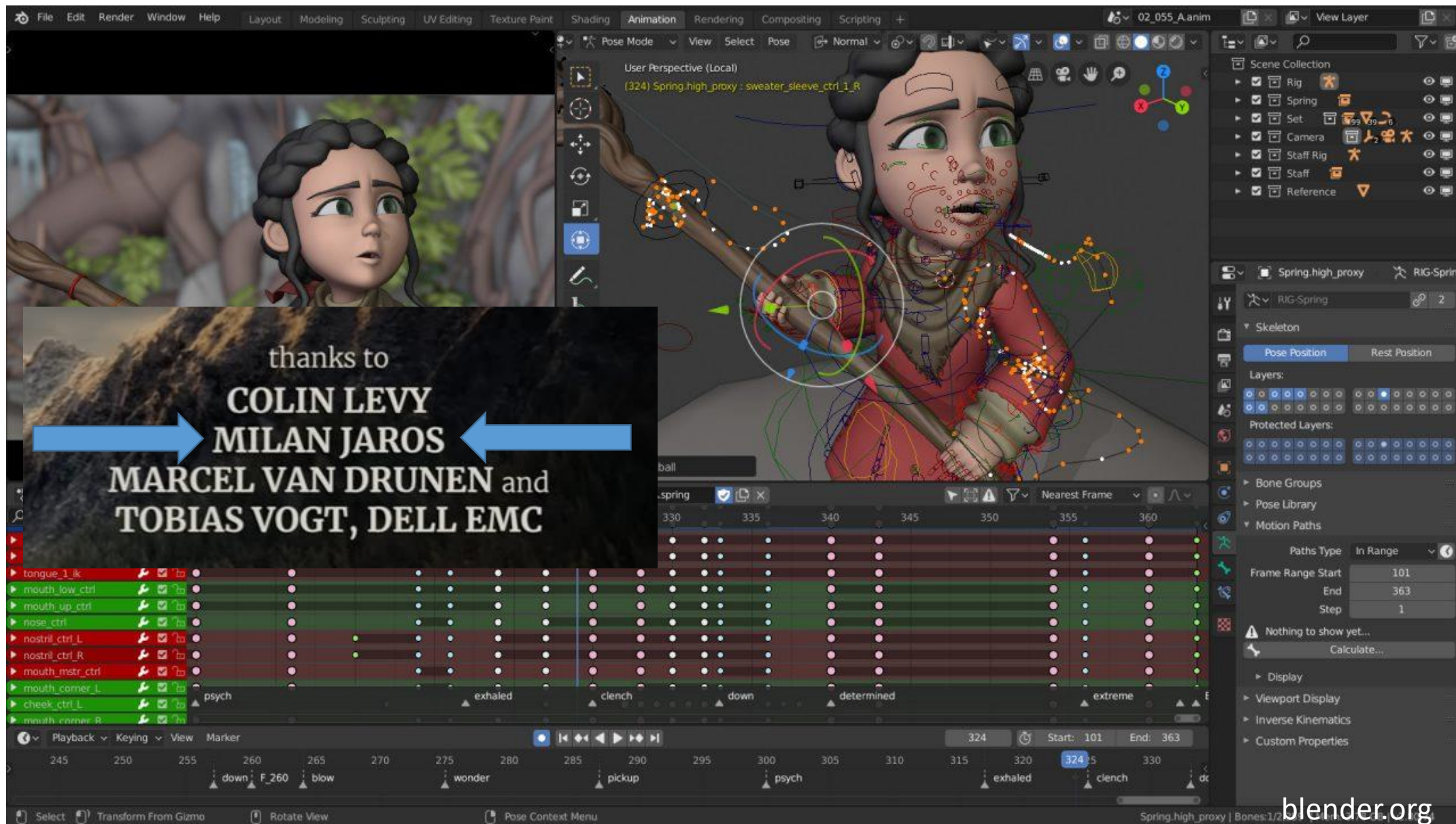
```
1 // Created by inigo quilez - iq/2017
2 // License Creative Commons Attribution-NonCommercial-ShareAlike 3.0
3
4 void mainImage( out vec4 fragColor, in vec2 fragCoord )
5 {
6     vec2 q = fragCoord / iResolution.xy;
7
8
9     // dof
10    const float focus = 2.35;
11
12    vec4 acc = vec4(0.0);
13    const int N = 12;
14    for( int j=-N; j<=N; j++ )
15    for( int i=-N; i<=N; i++ )
16    {
17        vec2 off = vec2(float(i),float(j));
18
19        vec4 tmp = texture( iChannel0, q + off/vec2(800.0,450.0) );
20
21        float depth = tmp.w;
22
23        vec3 color = tmp.xyz;
24
25        float coc = 0.05 + 12.0*abs(depth-focus)/depth;
26
27        if( dot(off,off) < (coc*coc) )
28        {
29            float w = 1.0/(coc*coc);
30            acc += vec4(color*w,w);
31        }
32    }
33
34    vec3 col = acc.xyz / acc.w;
35
36
37    // gamma
38    col = pow( col, vec3(0.4545) );
39
40    // color correct - it seems my laptop has a fucked up contrast/gamma setting, so I need
41    // to do this for the picture to look okey in all computers but mine...
```

Inigo Quilez

Compiled in 0.1 / 2.8 secs 567 / 14168 chars

<https://www.shadertoy.com/view/4tByz3>

CO JE BLENDER 2.8?



CO JE MATERIÁL?

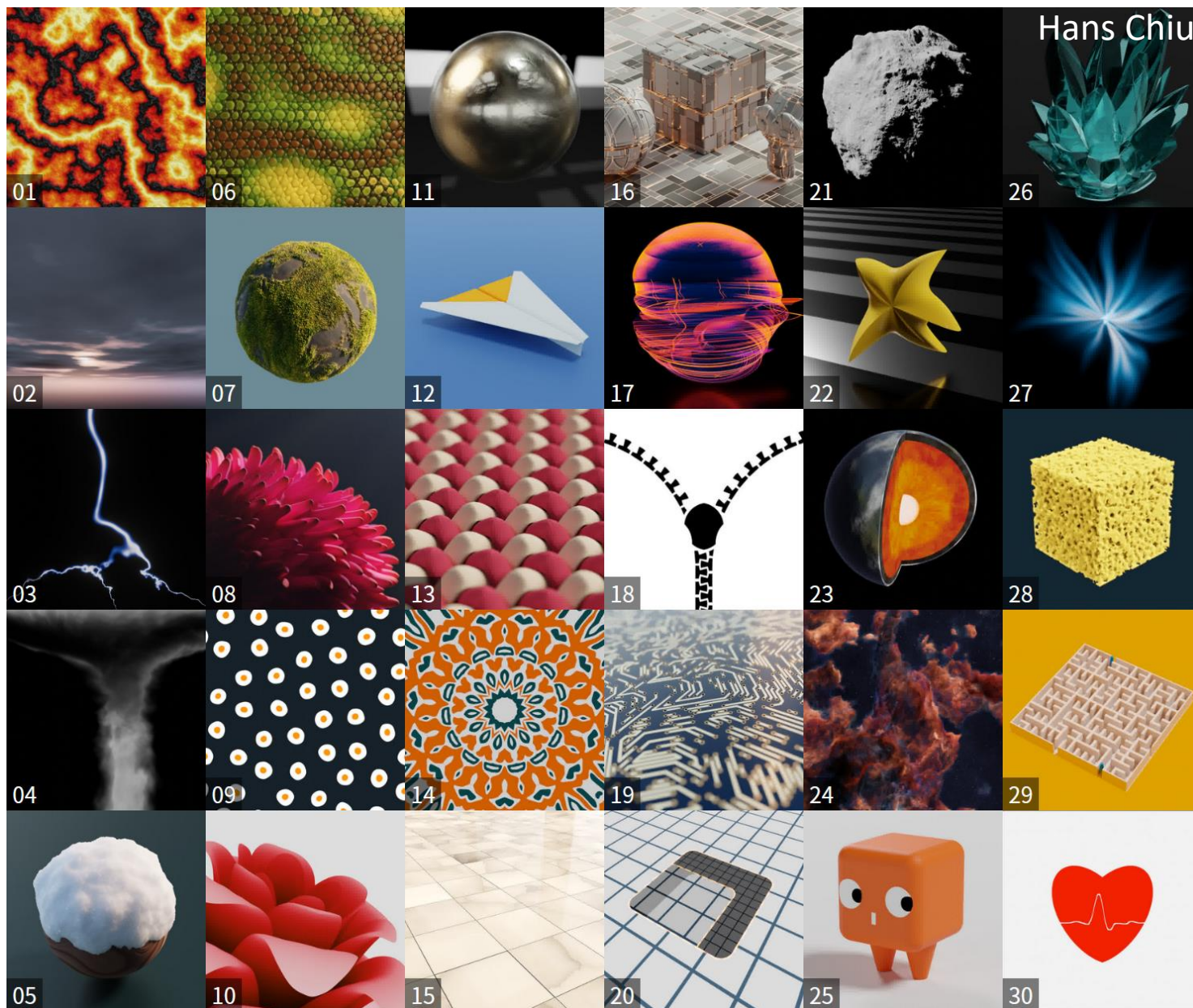


- | **materiál** - kolekce shaderů, které popisují povrch objektu
- | **shader** - definice za pomoci node editoru v Blenderu (SVM, GLSL)
- | **textura** - obrázek ze souboru (image), přidává detail do shaderu a nemá žádné fyzikální vlastnosti,
- | **procedurální textura** – definice za pomoci matematických vzorců/funkcí

materiál
(shader, node)
editor



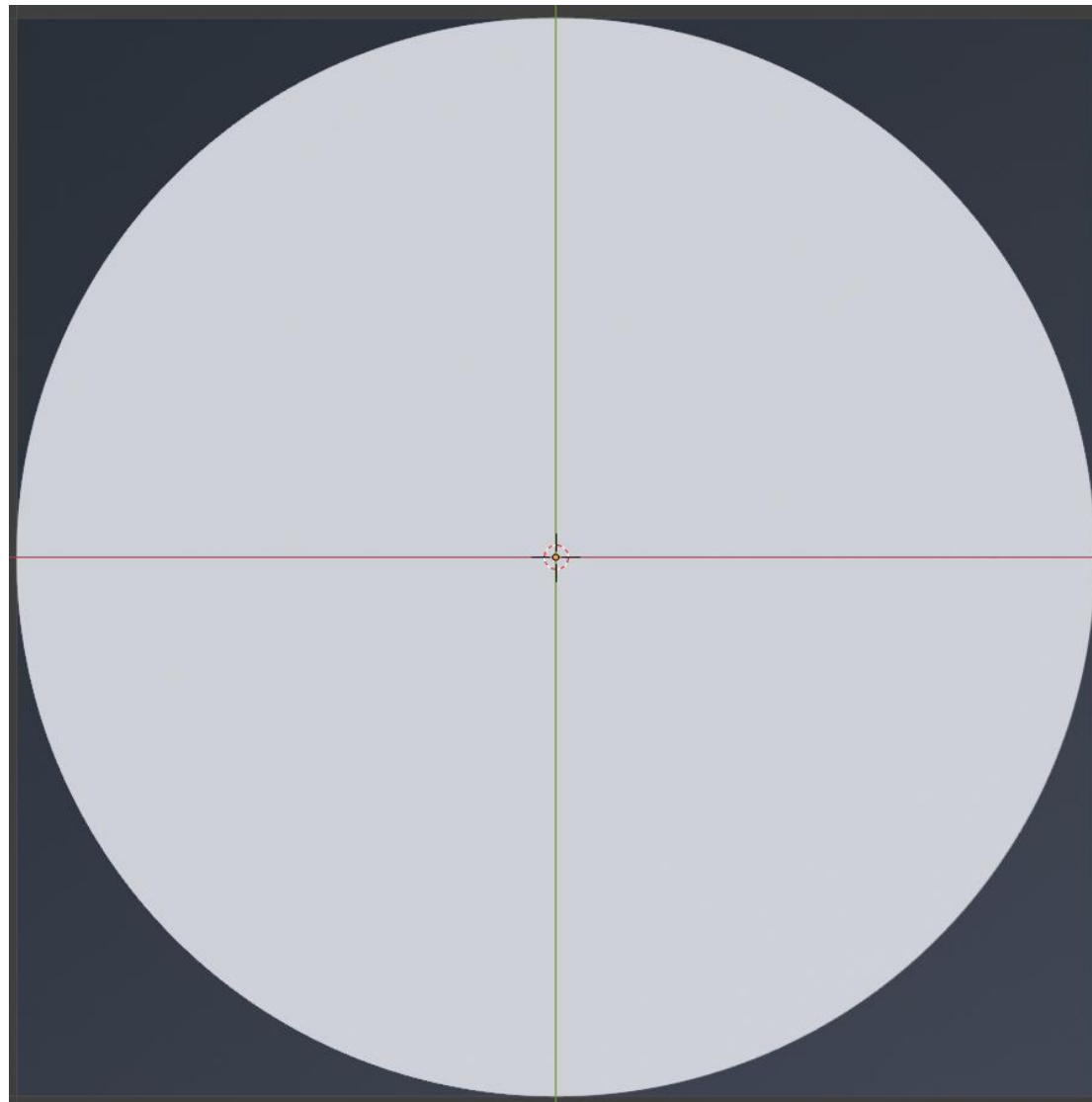
PŘÍKLADY PROCEDURÁLNÍCH TEXTUR



PROCEDURÁLNÍ TEXTURA - KRUŽNICE



$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2$$

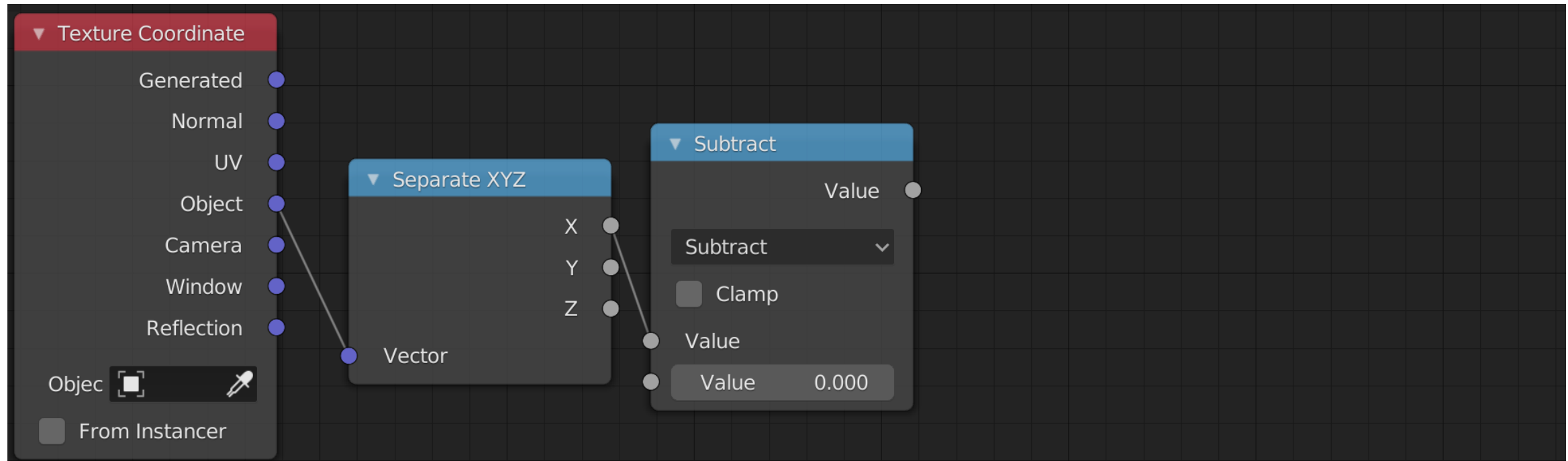


<https://b3d.interplanety.org/en/creating-procedural-textures-in-blender/>

PROCEDURÁLNÍ TEXTURA - KRUŽNICE



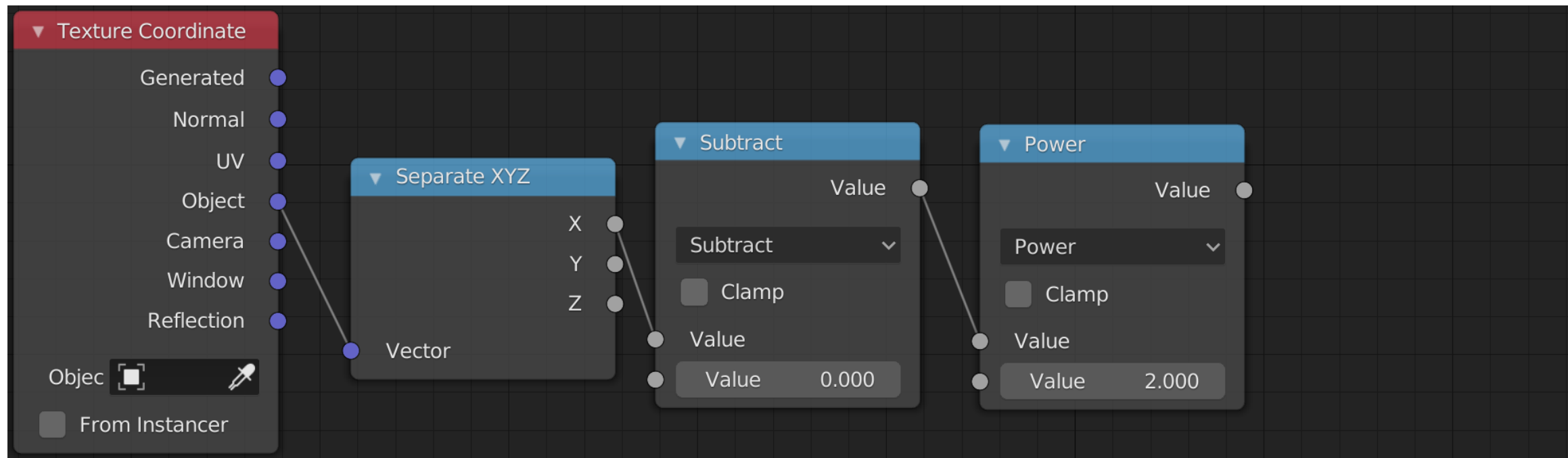
$$x - x_0$$



PROCEDURÁLNÍ TEXTURA - KRUŽNICE



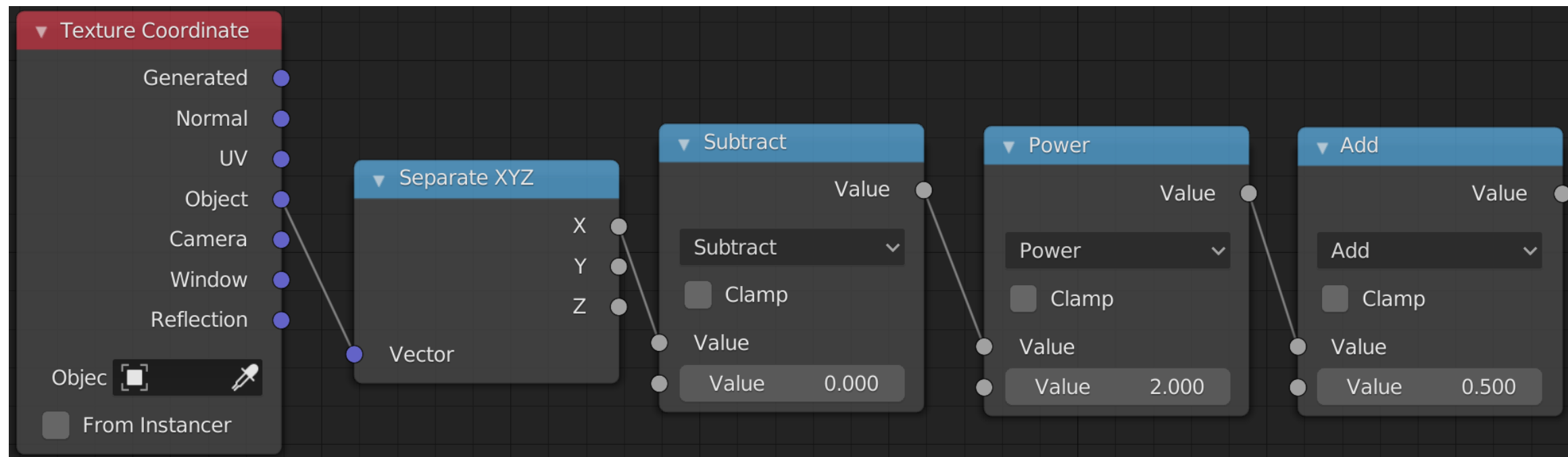
$$(x - x_0)^2$$



PROCEDURÁLNÍ TEXTURA - KRUŽNICE



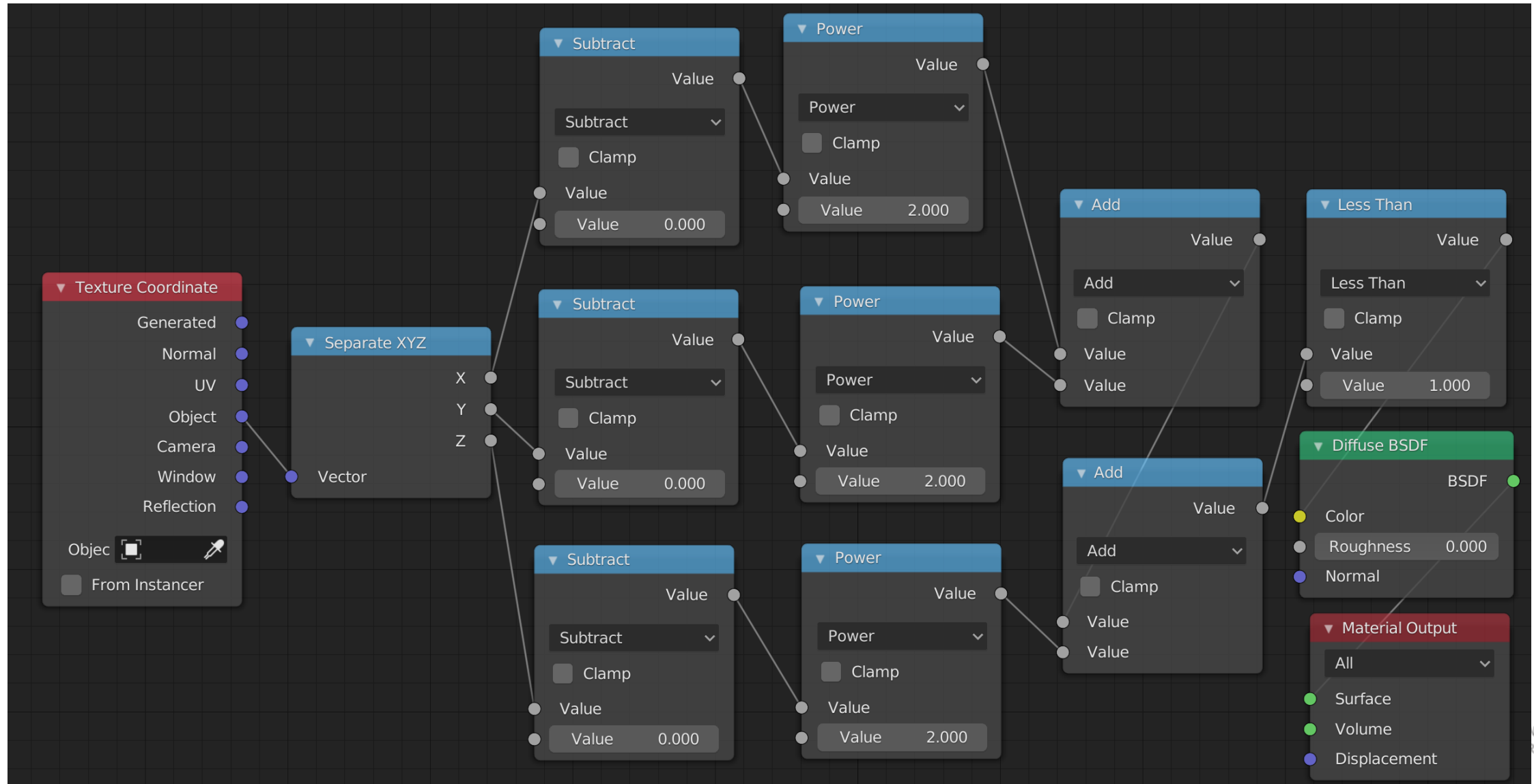
$$(x - x_0)^2 +$$



PROCEDURÁLNÍ TEXTURA - KRUŽNICE



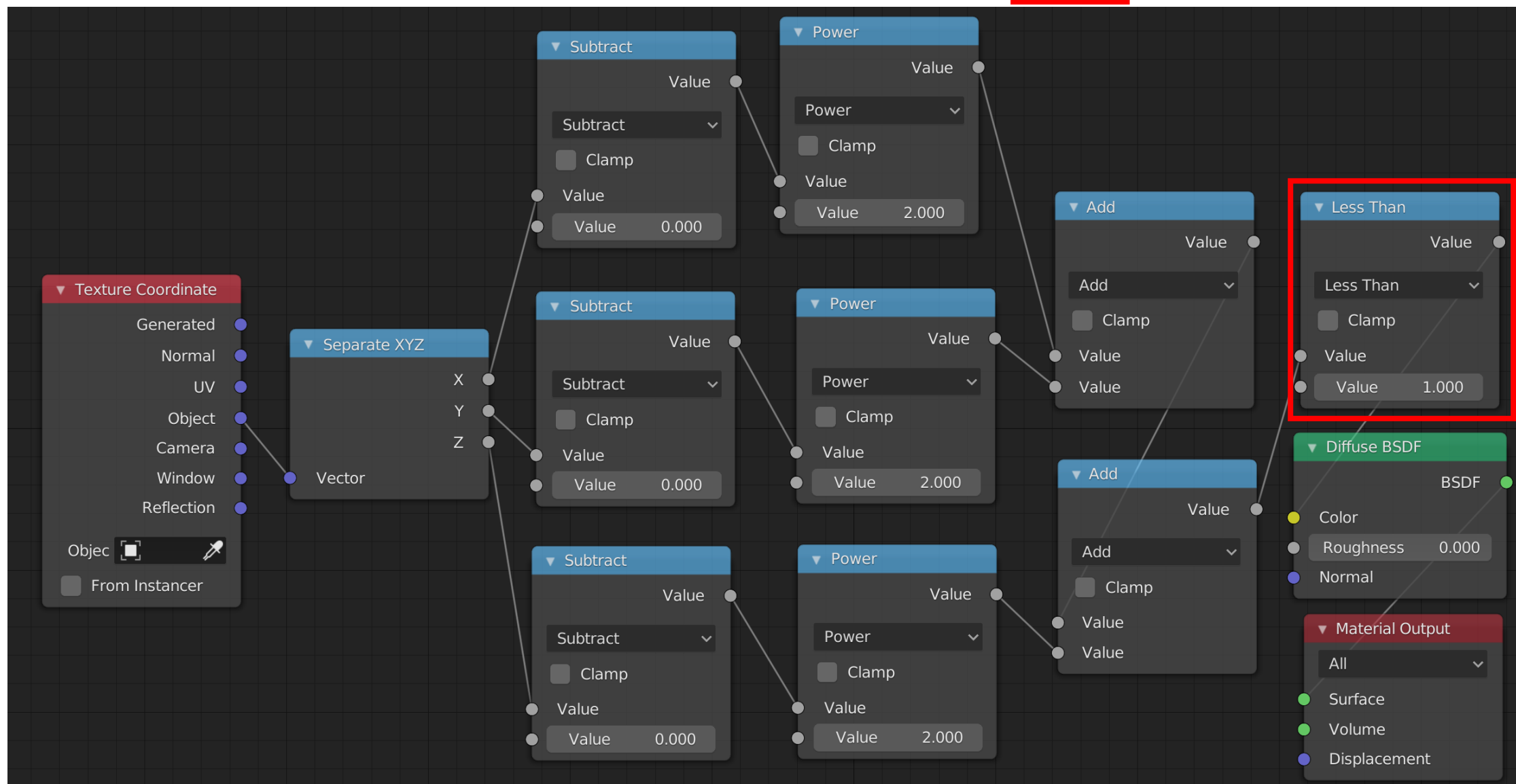
$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2$$



PROCEDURÁLNÍ TEXTURA - KRUŽNICE



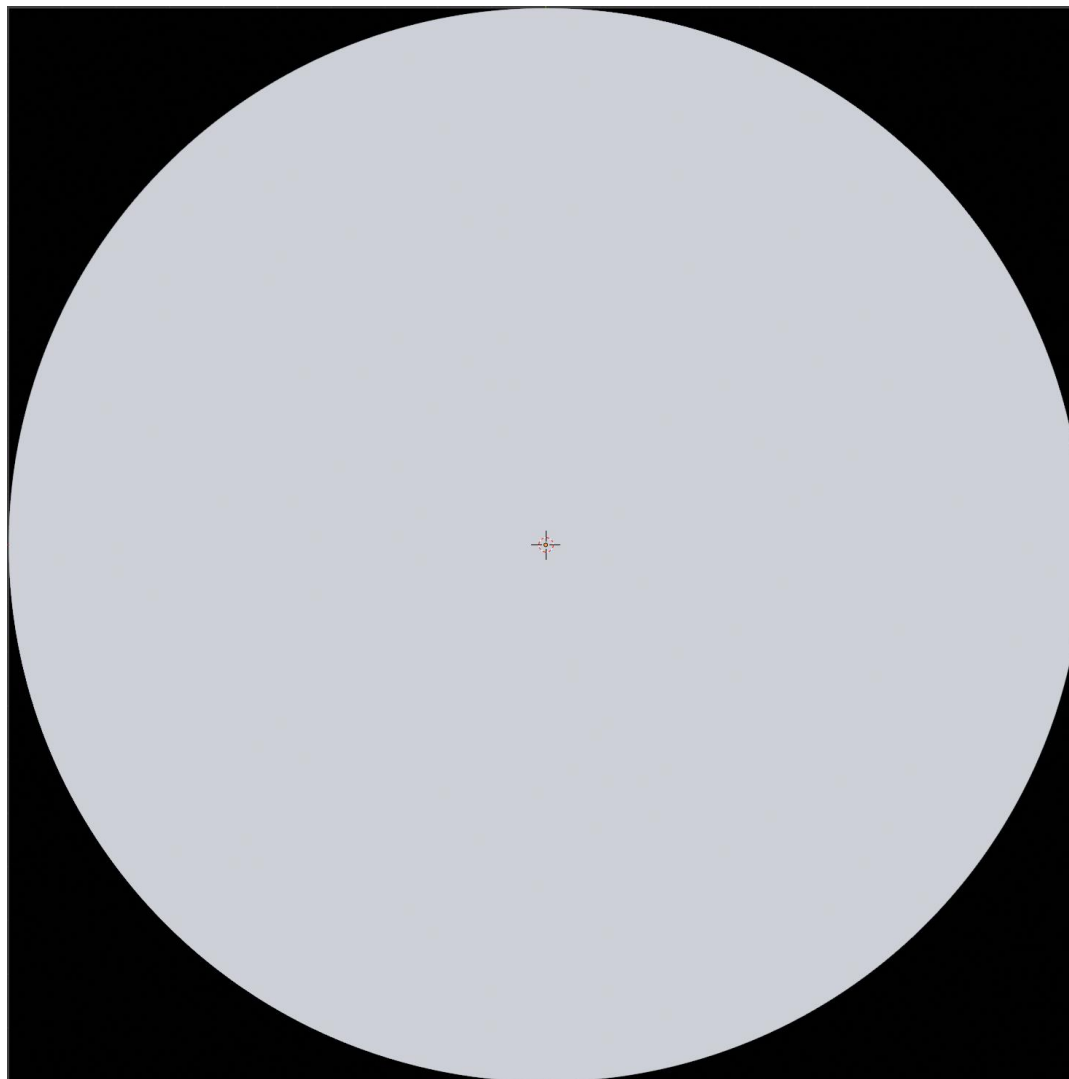
$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2$$



PROCEDURÁLNÍ TEXTURA - KRUŽNICE



$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2$$

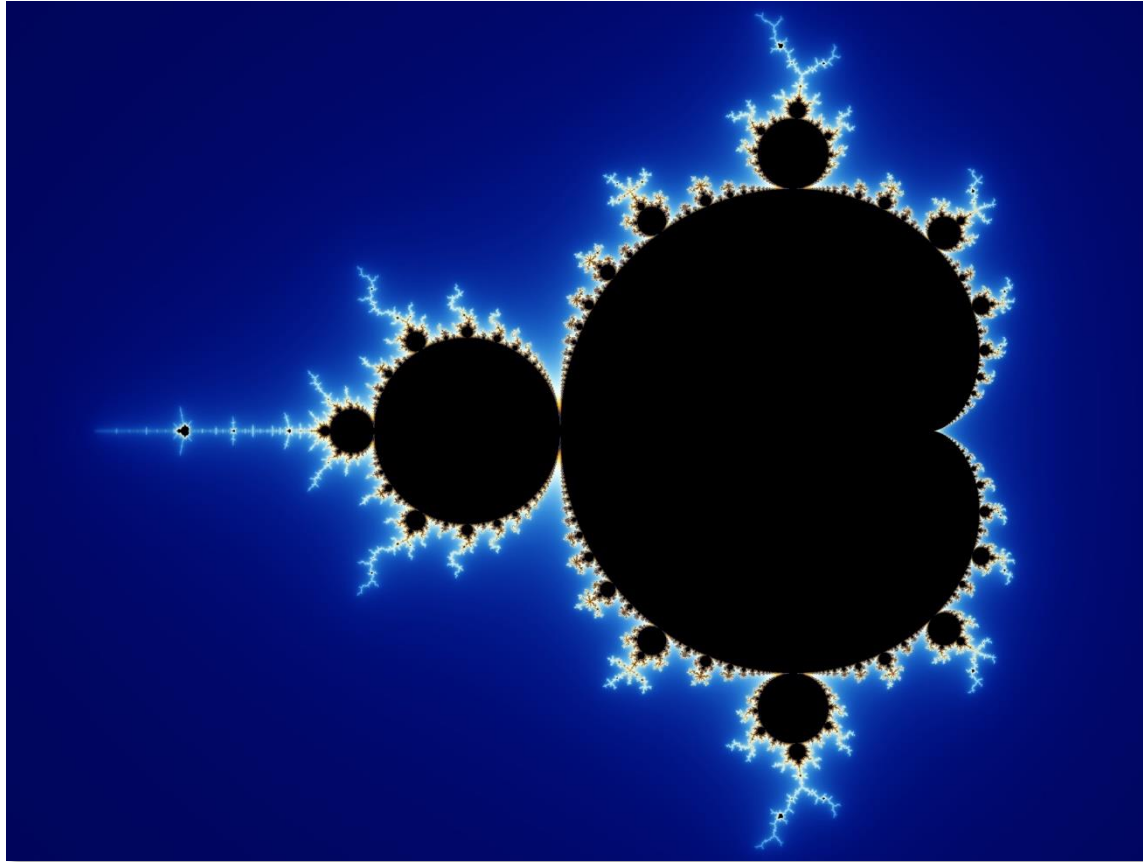


<https://b3d.interplanety.org/en/creating-procedural-textures-in-blender/>

MANDELBROTOVA MNOŽINA



= 2D fraktál, Benoit B. Mandelbrot (1979)



$$M = \{c | P_c^n(0) \neq \infty \forall n \rightarrow \infty; c, z \in \mathbb{C}\},$$

kde $P_c(z) = z^2 + c$

Příklad:

$$c = 1$$

$$\begin{aligned} P_1^0(0) &= 0 \\ P_1^1(0) &= 0^2 + 1 = 1 \\ P_1^2(1) &= 1^2 + 1 = 2 \\ P_1^3(2) &= 2^2 + 1 = 5 \\ P_1^4(5) &= 5^2 + 1 = 26 \\ &\vdots \end{aligned}$$

$$c = i$$

$$\begin{aligned} P_i^0(0) &= 0 \\ P_i^1(0) &= i \\ P_i^2(i) &= -1 + i \\ P_i^3(-1 + i) &= -i \\ P_i^4(-i) &= -1 + i \\ &\vdots \end{aligned}$$

PROCEDURÁLNÍ TEXTURA - MANDELBULB



| = 3D fraktál, Daniel White a Paul Nylander (2009)

| White a Nylanderův vzorec pro n-tou mocninu vektoru $\mathbf{v} = \langle x, y, z \rangle$ v $\mathbb{R}^3$

$$\mathbf{v}^n := r^n \langle \sin(n\theta) \cos(n\phi), \sin(n\theta) \sin(n\phi), \cos(n\theta) \rangle$$

$$\text{kde } r = \sqrt{x^2 + y^2 + z^2},$$

$$\phi = \arctan(y/x) = \arg(x + yi)$$

$$\theta = \arctan(\sqrt{x^2 + y^2}/z) = \arccos(z/r)$$

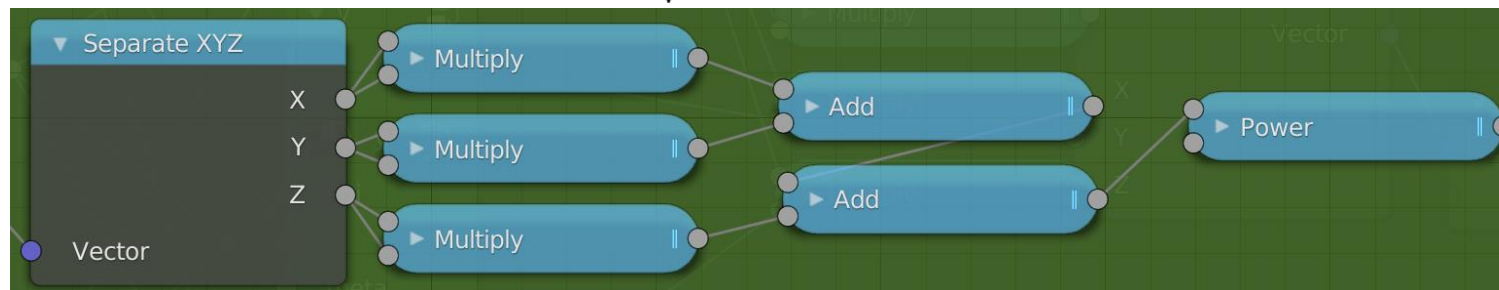
| Mandelbulbova množina

$$M = \{c \mid \mathbf{v} \mapsto \mathbf{v}^n + \mathbf{c} ; c \in \mathbb{R}^3\}$$

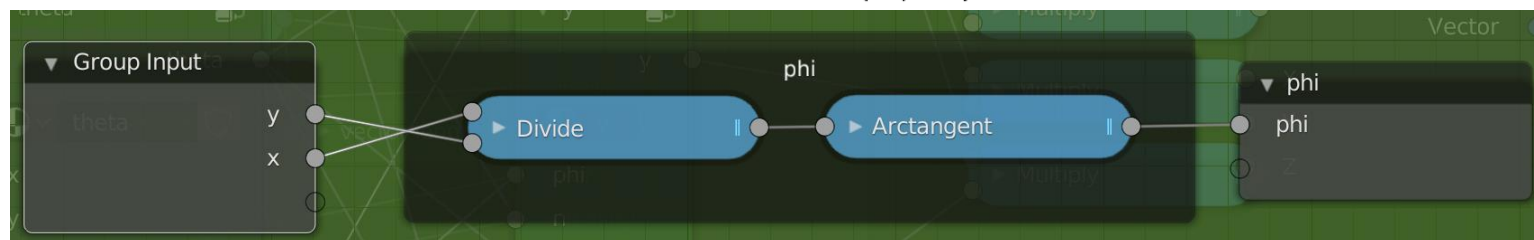
PROCEDURÁLNÍ TEXTURA - MANDELBULB



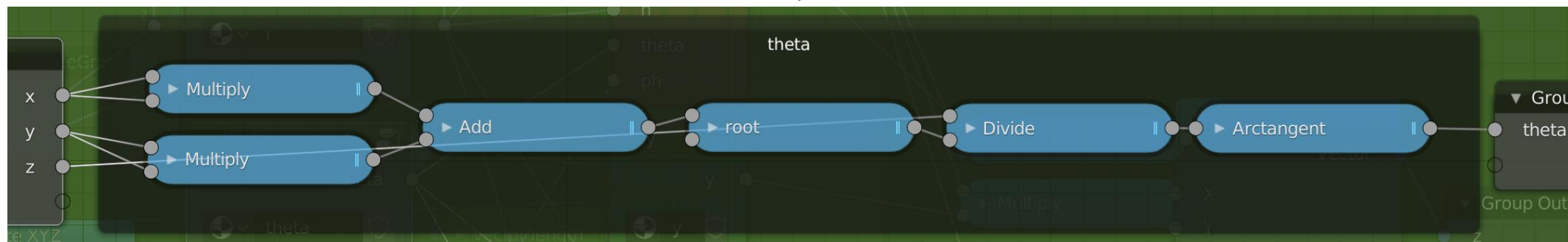
$$r = \sqrt{x^2 + y^2 + z^2}$$



$$\phi = \arctan(y/x)$$



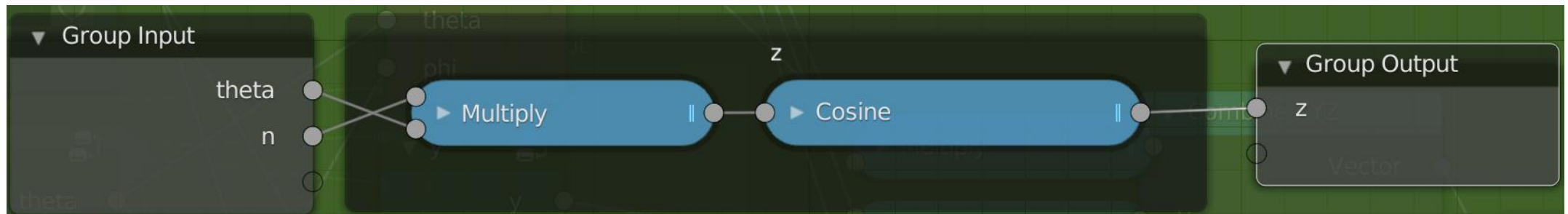
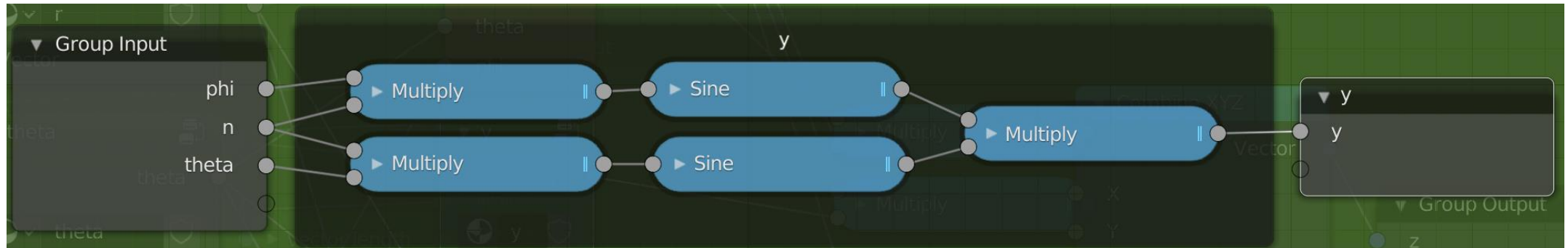
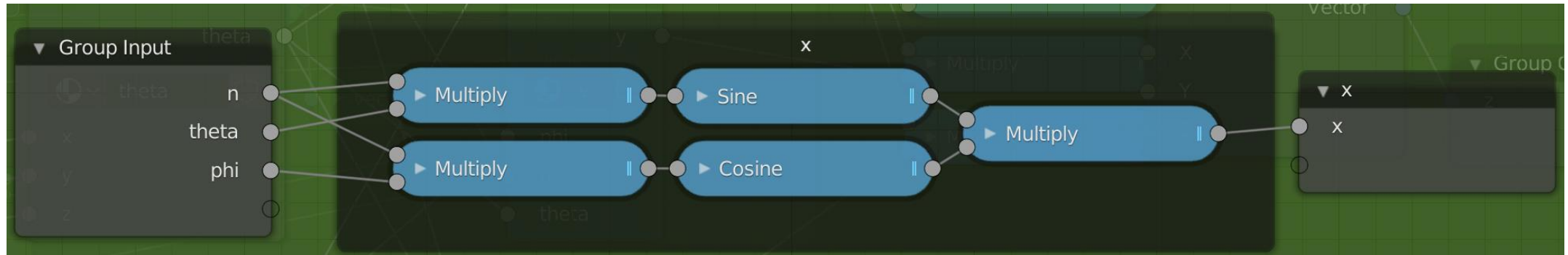
$$\theta = \arctan(\sqrt{x^2 + y^2} / z)$$



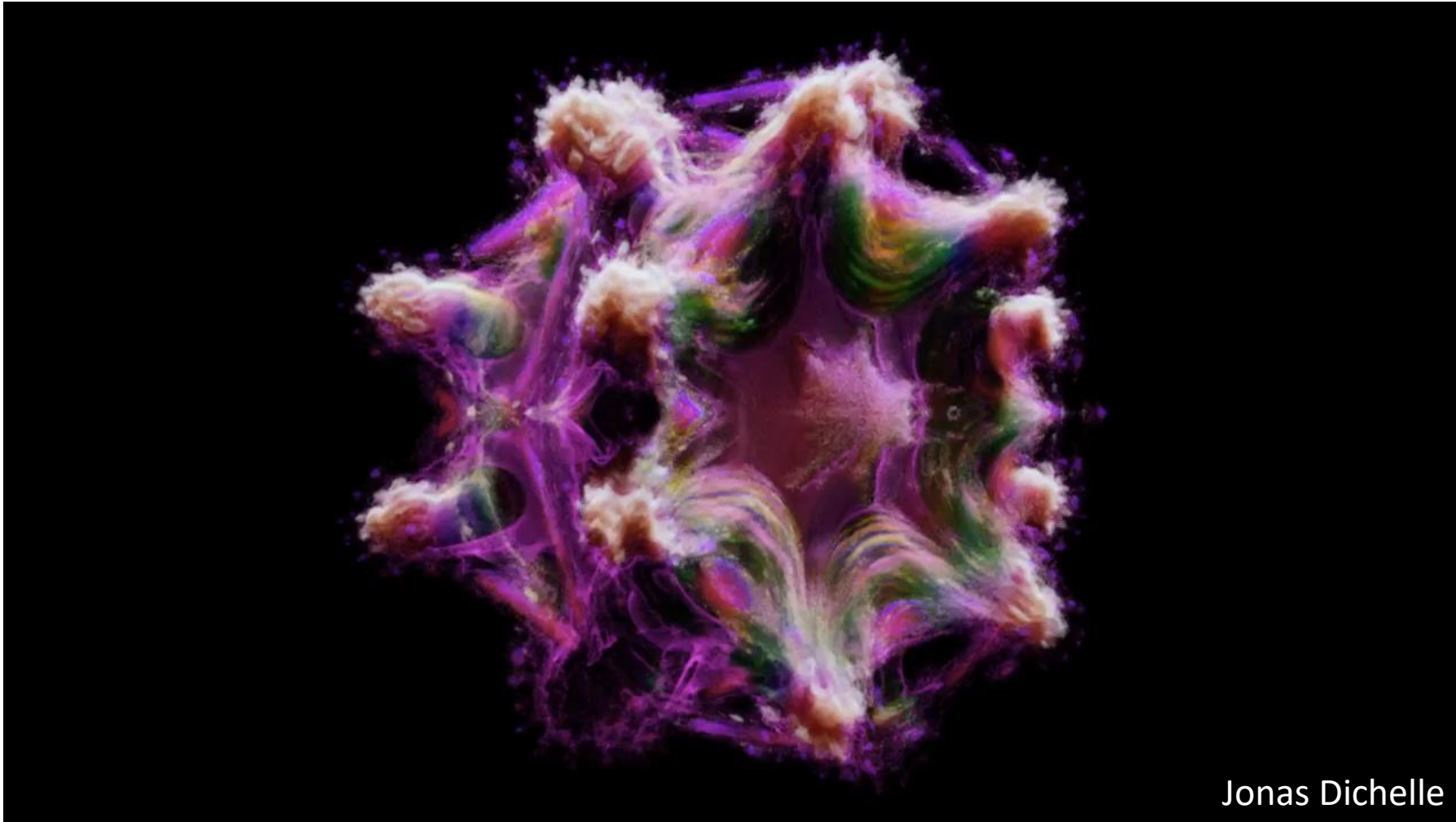
PROCEDURÁLNÍ TEXTURA - MANDELBULB



$$\mathbf{v}^n := r^n \langle \sin(n\theta) \cos(n\phi), \sin(n\theta) \sin(n\phi), \cos(n\theta) \rangle$$



PROCEDURÁLNÍ TEXTURA - MANDELBULB



Jonas Dichelle

https://www.youtube.com/watch?v=YtDG1_76_2k

RENDEROVÁNÍ



<https://www.youtube.com/watch?v=mN0zPOpADL4>



<https://www.youtube.com/watch?v=Y-rmzh0PI3c>



<https://www.youtube.com/watch?v=RJnKaAtBPhA>



<https://www.youtube.com/watch?v=WhWc3b3KhY>

RENDEROVÁNÍ STATE-OF-THE-ART



Point based global illumination (Christensen 2008)



Up
(2009, Pixar/Disney)



Kung Fu Panda 2
(2011, DreamWorks)

Path-tracing (Kajiya 1986)



Big Hero 6
(2014, Disney)



Trolls
(2016, DreamWorks)

- | Path-tracing renderery
- In-house Hyperion (Disney)
- In-house MoonRay V3 (DreamWorks)
- In-house Manuka (WetaDigital)
- Solid Angle Arnold renderer (Sony Pictures Imageworks, ILM, ...)
- **Blender Cycles**

SVĚTLO V ANIMOVANÝCH FILMECH

Špióni v převleku (2019)

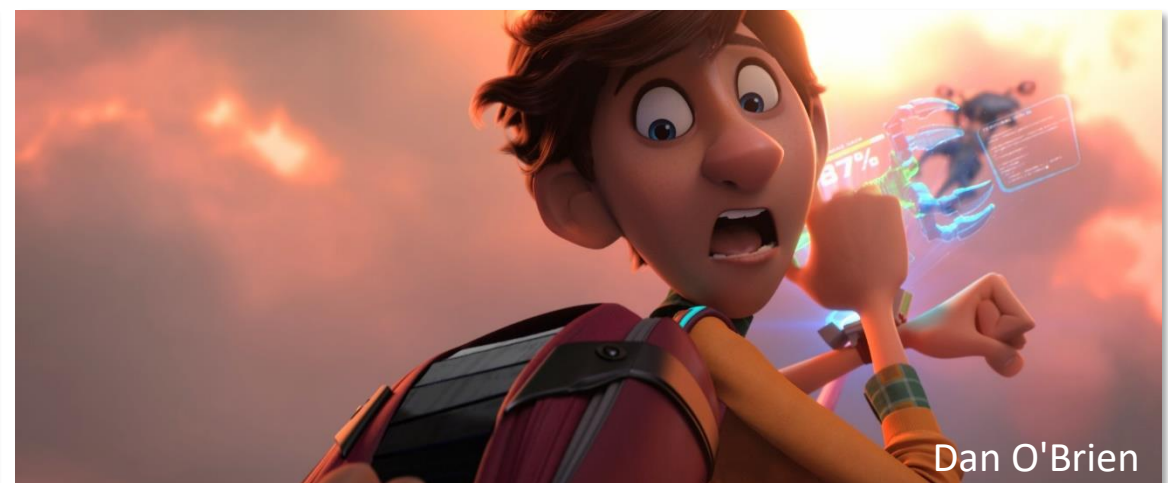
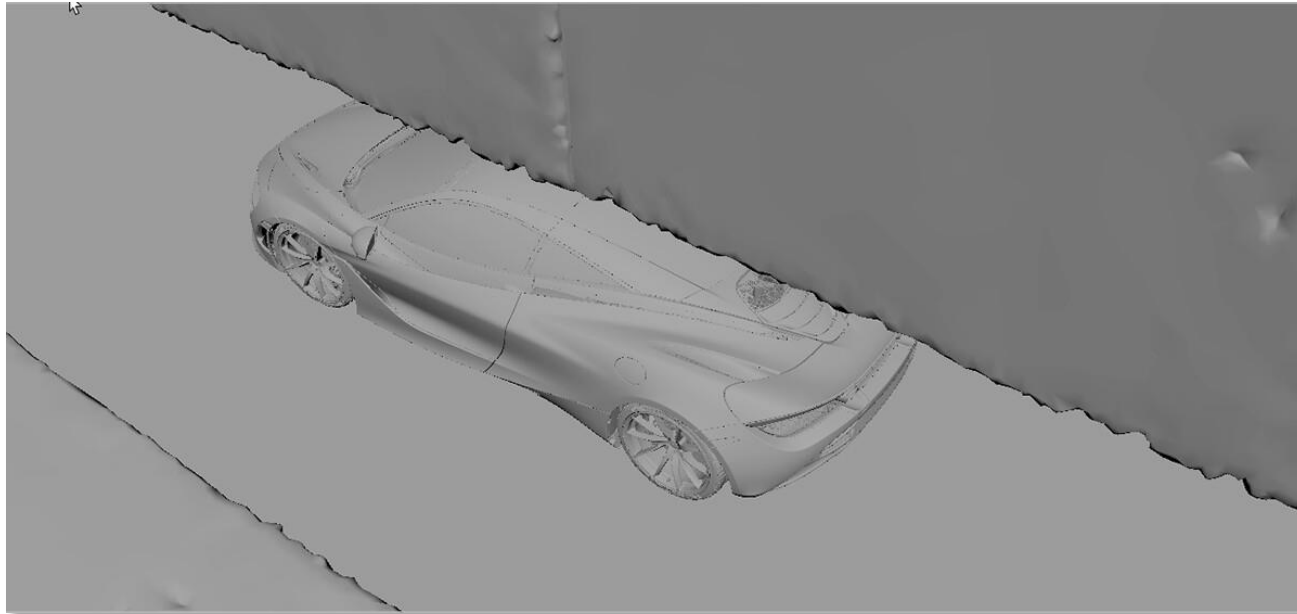


FOTO NEBO POČÍTAČOVÁ GRAFIKA?



Andrew Price

FOTO NEBO POČÍTAČOVÁ GRAFIKA?



CG



FOTO

FOTO NEBO POČÍTAČOVÁ GRAFIKA?



Andrew Price

FOTO NEBO POČÍTAČOVÁ GRAFIKA?

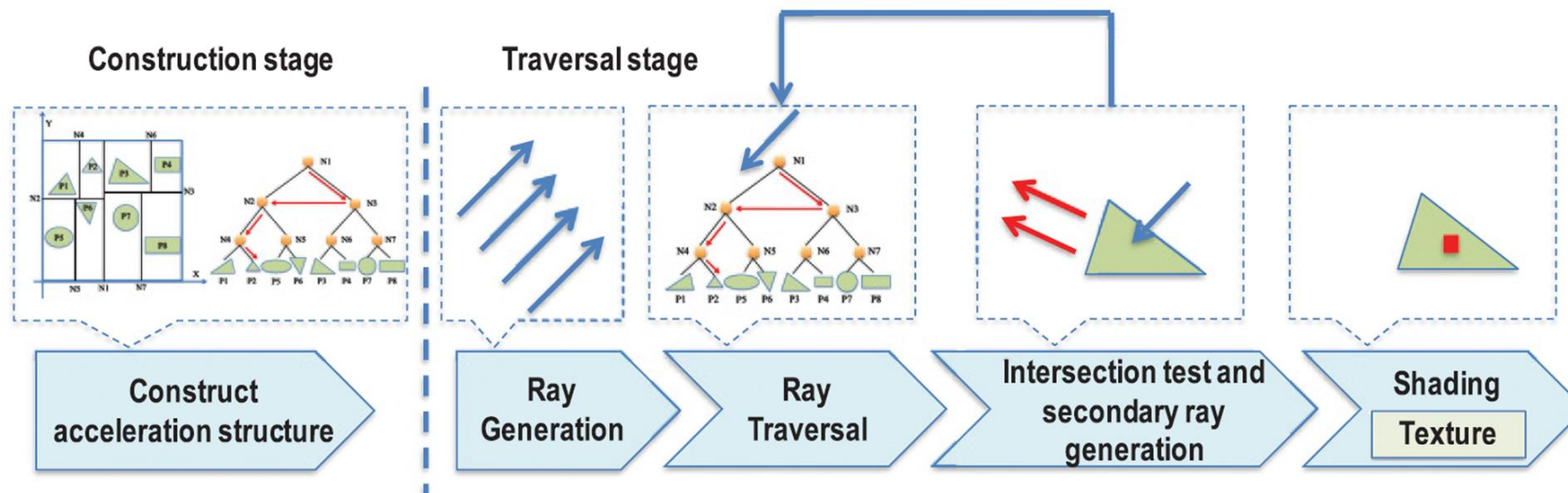
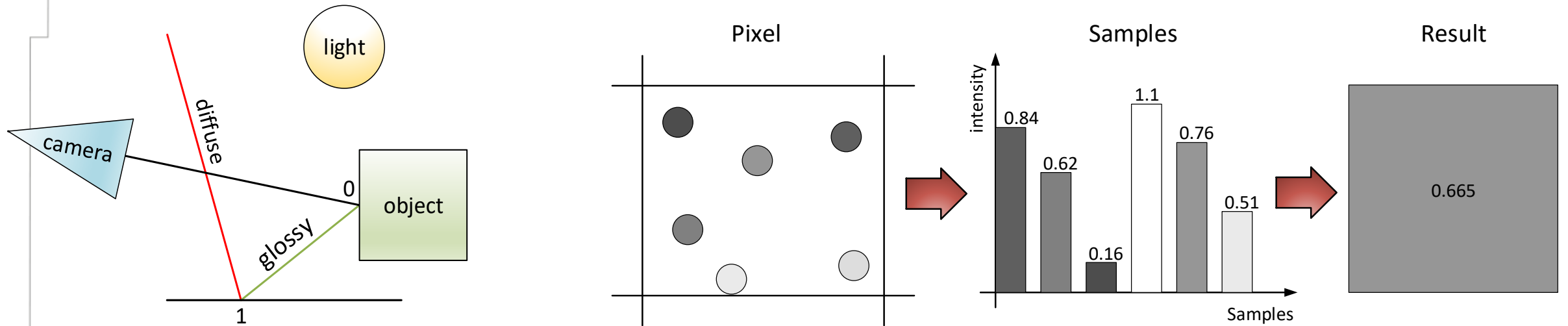


FOTO



CG

PATH-TRACING



Toward Real-Time Ray Tracing: A Survey on Hardware Acceleration and Microarchitecture Techniques, Yangdong Deng et al., ACM 2017

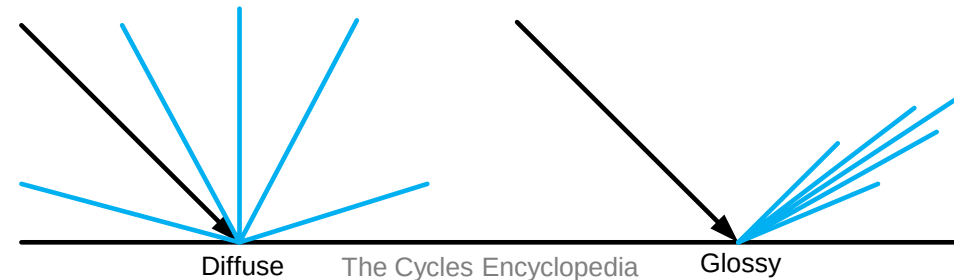
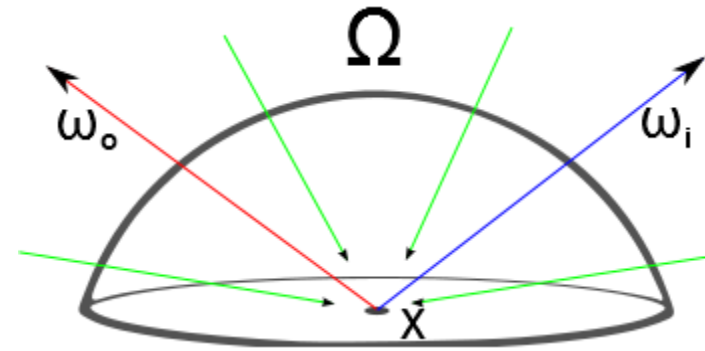
PATH-TRACING (KAJIYA 1986)



Zobrazovací rovnice a její řešení za pomoci Monte Carlo metody

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} L_i(x, \omega_i) f_r(x, \omega_i, \omega_o) (\omega_i \cdot n) d\omega_i$$

- | ω_o is direction of outgoing ray
- | ω_i is direction of incoming ray
- | L_o is spectral radiance emitted by the source from point x in direction ω_o
- | L_e is emitted spectral radiance from point x in direction ω_o
- | Ω is the unit hemisphere in direction of normal vector n with center in x , over which we integrate
- | L_i is spectral radiance coming inside to x in direction ω_i ,
- | $f_r(x, \omega_i, \omega_o)$ is distribution function of the image (BRDF) in point x from direction ω_i to direction ω_o ,
- | $\omega_i \cdot n$ is angle between ω_i and surface normal.



PATH-TRACING (KAJIYA 1986)



Zobrazovací rovnice:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} L_i(x, \omega_i) f_r(x, \omega_i, \omega_o) (\omega_i \cdot n) d\omega_i$$

Co je problematického na této rovnici?

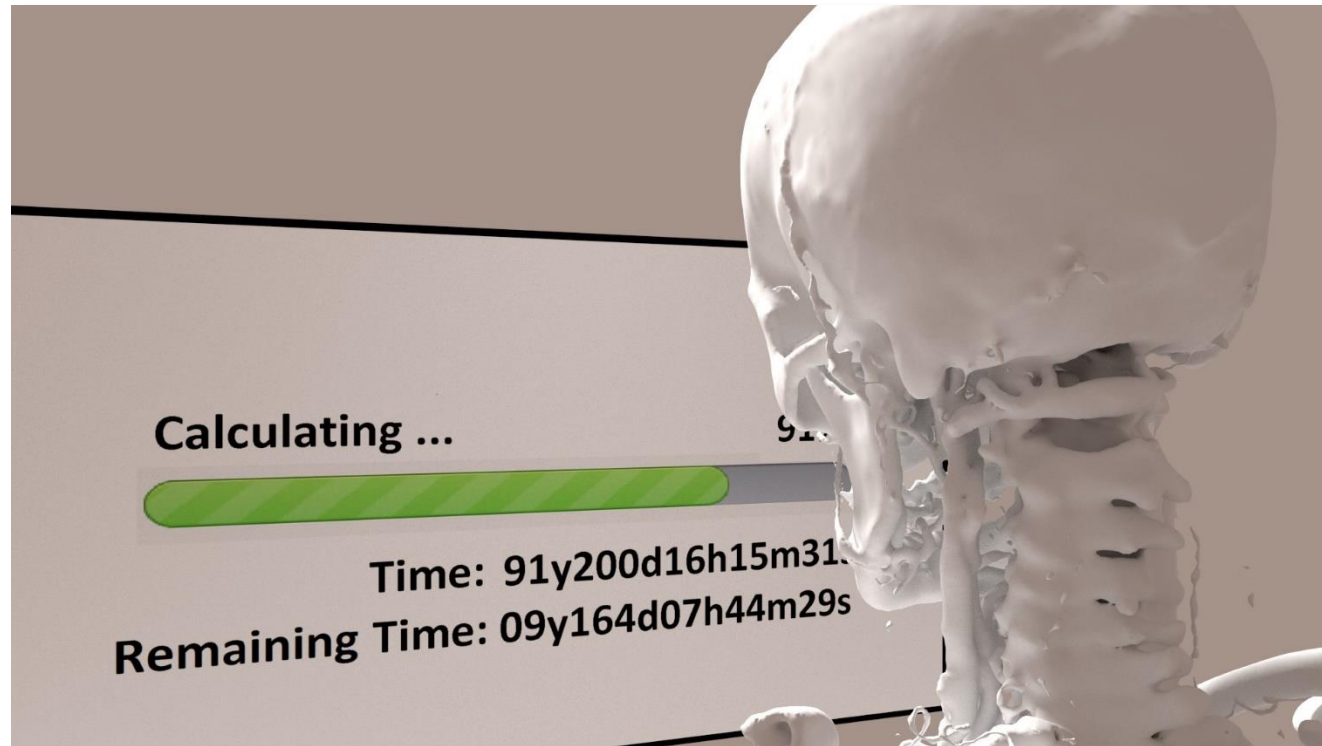
PATH-TRACING (KAJIYA 1986)



Zobrazovací rovnice:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} L_i(x, \omega_i) f_r(x, \omega_i, \omega_o) (\omega_i \cdot n) d\omega_i$$

**Redukce šumu v obraze vyžaduje vysoký počet samplů
a tím i dlouhý čas výpočtu.**



PATH-TRACING (KAJIYA 1986)



Zobrazovací rovnice:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} L_i(x, \omega_i) f_r(x, \omega_i, \omega_o) (\omega_i \cdot n) d\omega_i$$



1 sample



100 samplů

SHREKŮV ZÁKON



DREAMWORKS

| Mooreův zákon

Počet tranzistorů, které mohou být umístěny na integrovaný obvod se při zachování stejné ceny zhruba každých 18 měsíců zdvojnásobí.

SHREKŮV ZÁKON



DREAMWORKS

| Mooreův zákon

Počet tranzistorů, které mohou být umístěny na integrovaný obvod se při zachování stejné ceny zhruba každých 18 měsíců zdvojnásobí.

| Shrekův zákon

Počet renderovacích procesorových hodin potřebných k dokončení projektu se zdvojnásobí ve srovnání s množstvím času potřebného na předchozím projektu.

Shrek 1 (2001)	5 miliónů procesorových hodin
Shrek 2 (2004)	10 miliónů procesorových hodin
Shrek 3 (2007)	20 miliónů procesorových hodin
Shrek Forever 3D (2010)	50 miliónů procesorových hodin

SUPERPOČÍTANÍ (HPC)



- | Big Hero 6
199 miliónů jádro hodin
- | 1 osobní PC se 4 jádry
 $200\text{Mhod} / 4 \text{ cores} = 50\text{Mhod} = 2.1\text{Mdni} = \mathbf{5707 \text{ let}}$
- | Superpočítač s 1000 PC se 48 jádry
 $200\text{Mhod} / 48000 \text{ cores} = \mathbf{6 \text{ měsíců}}$



Big Hero 6
(2014, Disney)

HPC NA VŠB-TUO



| Anselm

- 209 nodes
- 2x Intel Xeon E5-2665, 2.4 GHz, 8 cores (Sandy Bridge)
- 64GB
- InfiniBand FDR56 / 7D Enhanced hypercube
- 180 nodes w/o accelerator
- 4 nodes with MIC accelerator (1x Intel Xeon Phi 5110P)
- 23 nodes with NVIDIA Kepler K20

| Salomon

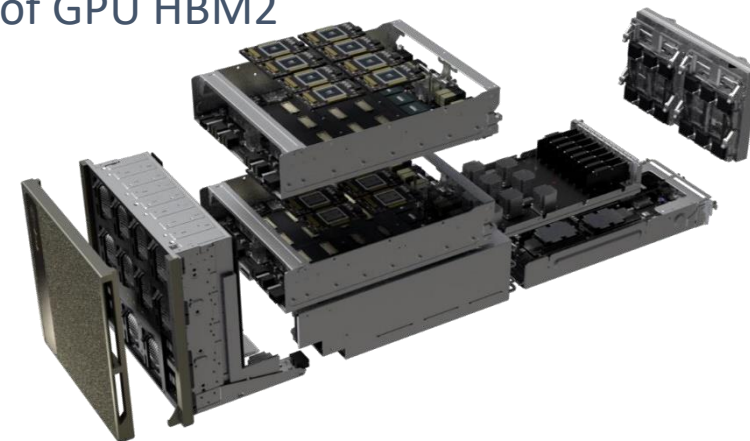
- 1008 nodes
- 2x Intel Xeon E5-2680v3, 2.5 GHz, 12 cores (Haswell) w. 128GB
- InfiniBand FDR56 / 7D Enhanced hypercube
- 576 nodes w/o accelerator
- 432 nodes with 2x Intel Xeon Phi 7120P, 61 cores, 16 GB RAM

| NVIDIA® DGX-2 SERVER

- 16 Tesla™ V100 32 GB GPUs
- 512 GB of GPU HBM2

| Barbora

- 192 nodes w/o an accelerator
- 8 nodes with 4x NVIDIA Tesla V100
- 2x Intel Xeon Cascade Lake, 2.6 GHz, 18 cores
- 192GB
- InfiniBand HDR 200Gb/s (100Gb/s per node)
- 30 GB/s LUSTRE storage





Milan Jaroš
milan.jaros@vsb.cz

IT4Innovations národní superpočítačové centrum
VŠB – Technická univerzita Ostrava
Studentská 6231/1B
708 00 Ostrava-Poruba
www.it4i.cz

VŠB TECHNICKÁ
UNIVERZITA
OSTRAVA

IT4INNOVATIONS
NÁRODNÍ SUPERPOČÍTAČOVÉ
CENTRUM