La matematica è arte

calcoli e numeri come pilastro
dello sviluppo dei videogiochi

Elisa Di Lorenzo

Co-Founder e Amministratore
Untold Games srl
elisa@untoldgames.com

Sol: 0 - 2 . 4
300,000 / 300,000
0
5,000 / 5,000
+120
-87
+33
2,000 / 6,000
0
5,000 / 5,000
0
Turn on the Oxygen Dispenser
use Suit cabinet
Turn off the Heater

?

Cosa usiamo principalmente?

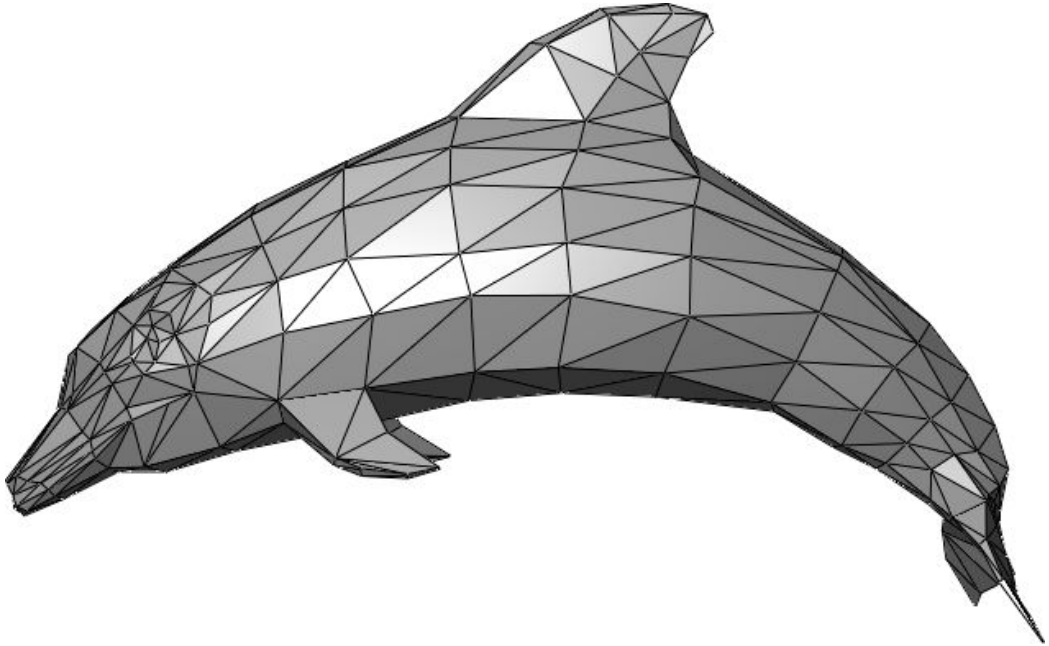
- Algebra
- Matematica discreta
- Trigonometria
- Analisi matematica
- Algebra lineare
- ...

- Vettori e versori, matrici
- Dot Product, Cross Product
- COS,SIN,TAN
- ...

Per fare cosa?

- ✘ Simulazione di fluidi
- ✘ Animazioni
- ✘ Rendering
- ✘ Gameplay scripting (camminare, saltare, costruire, sparare...)
- ✘ Analytics e data mining
- ✘ Programmazione di fisica (inerzia, ecc)
- ✘ Shader ed effetti speciali
- ✘ Path finding e IA
- ✘ Level design procedurale
- ✘ ...

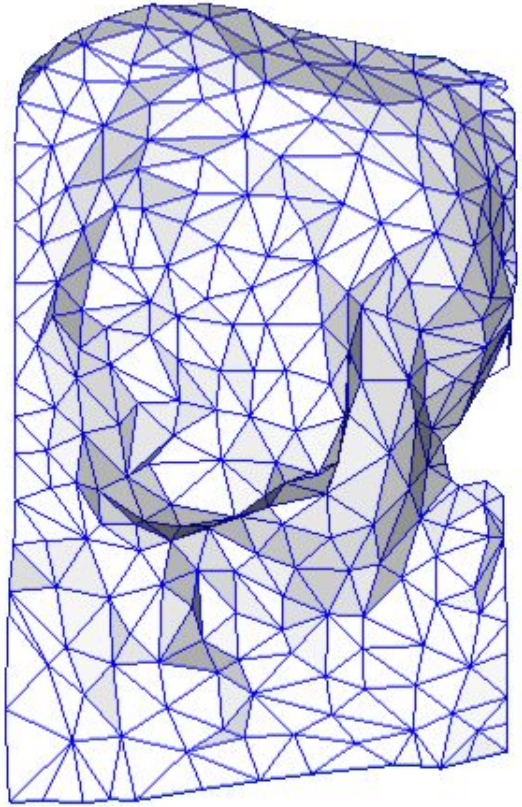
Shading



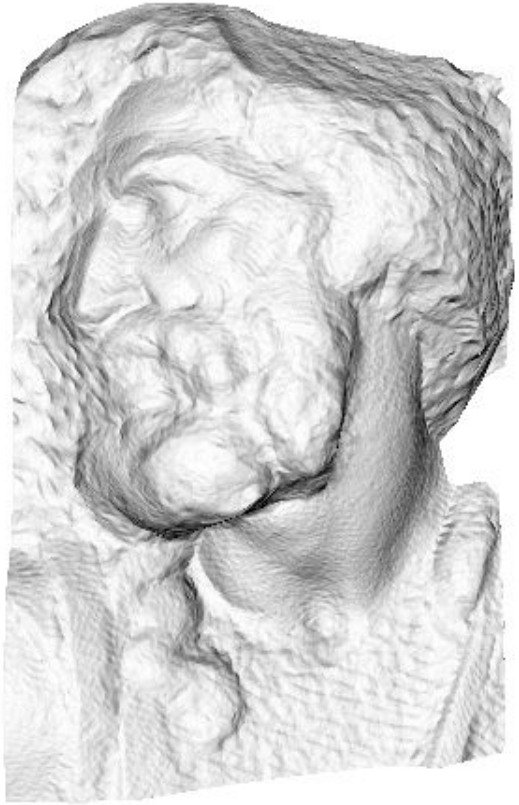
“

La maggior parte dei modelli di shading, in un modo o in un altro, sono progettati per catturare il processo di riflessione della luce

mesh originale
4M triangoli



mesh semplificata
500 triangoli

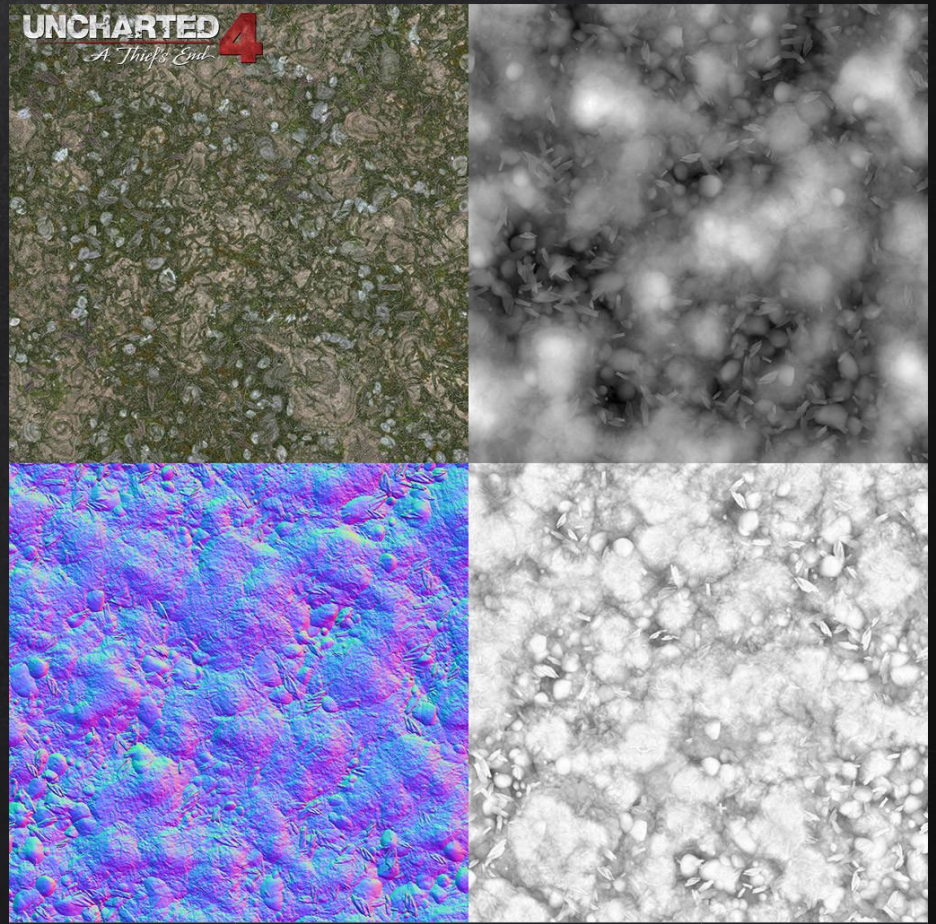


mesh semplificata
+ normal mapping
500 triangoli

Bradfolio.com BRADFORD SMITH

Mossy, Pebbly Terrain
Material Sample

NAUGHTY DOG



Bradfolio.com BRADFORD SMITH

Mossy, Pebbly Terrain
Albedo | Height | Normal | AO

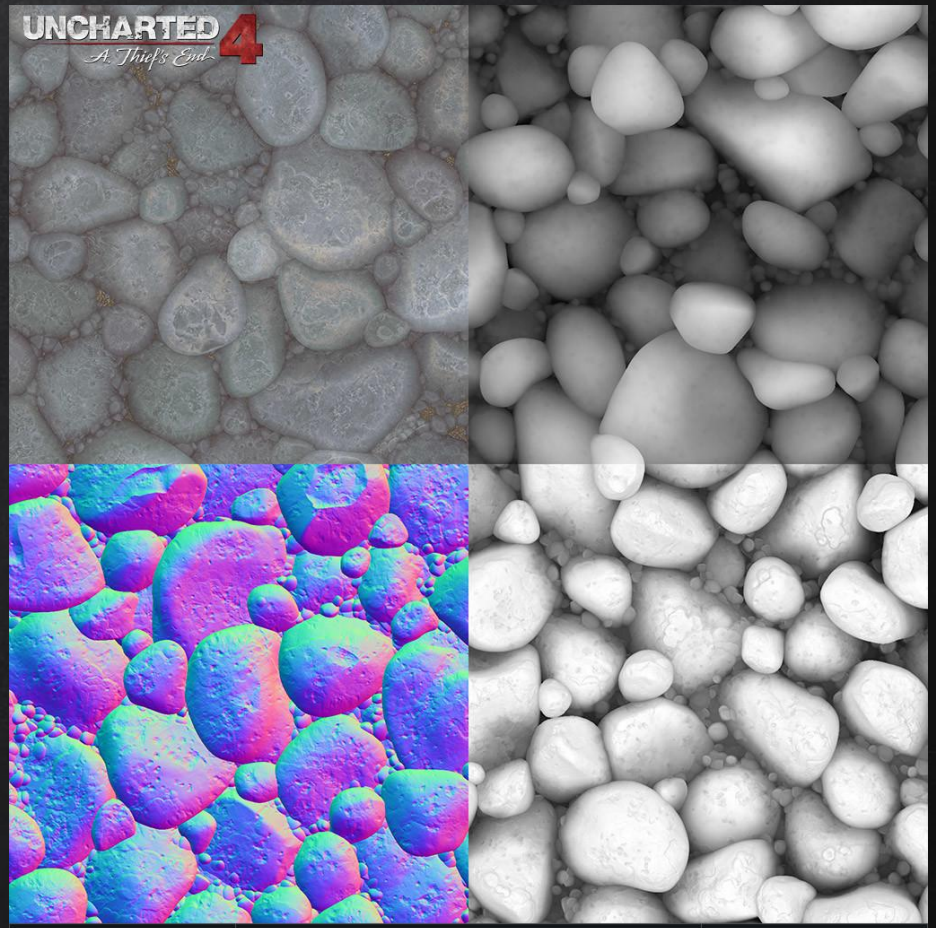
NAUGHTY DOG

Bradfolio.com BRADFORD SMITH

Underwater River Rock

Material Sample

NAUGHTY DOG



Bradfolio.com BRADFORD SMITH

Underwater River Rock

Albedo | Height | Normal | AO

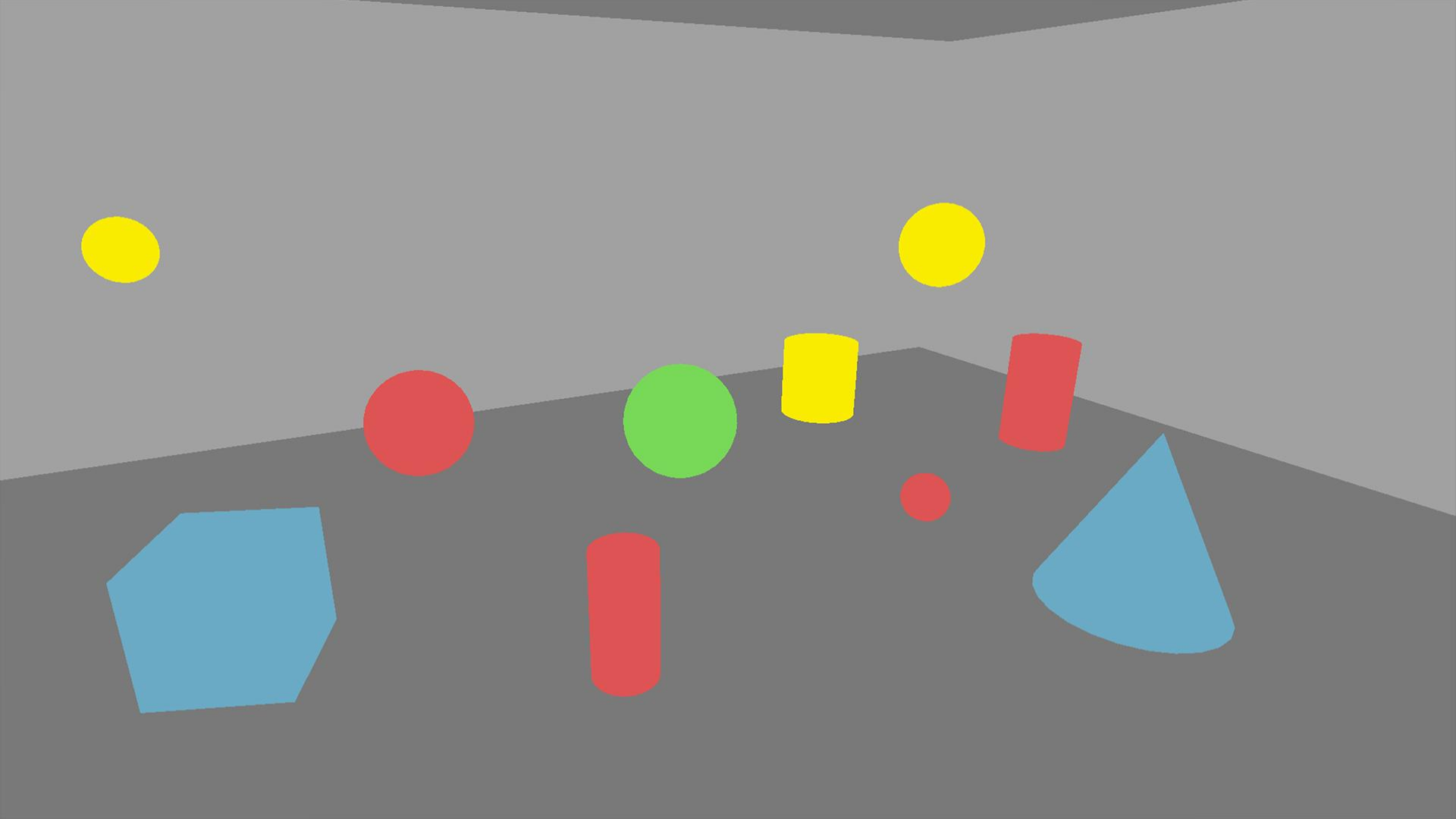
NAUGHTY DOG

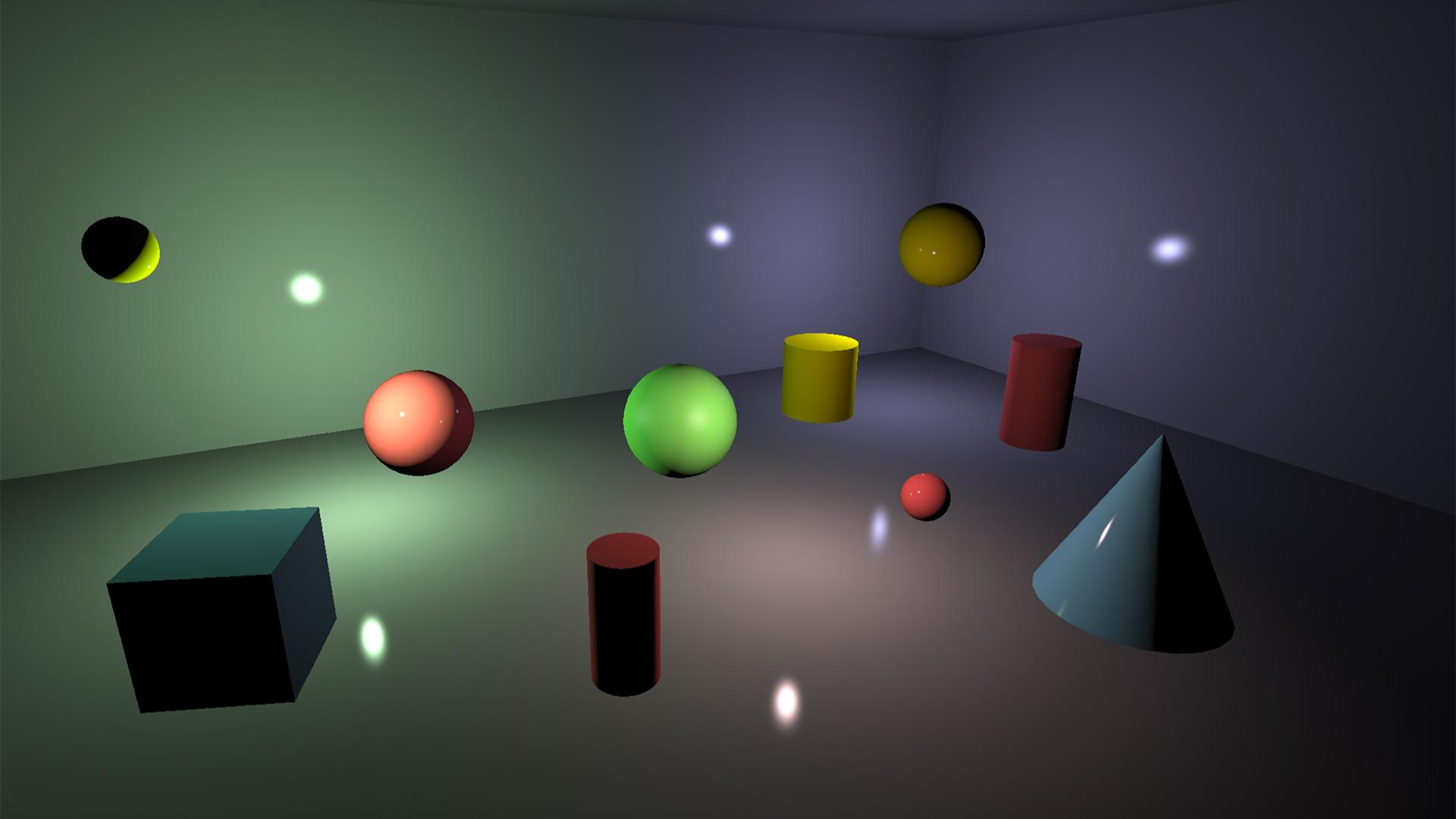
UNCHARTED
THE LOST LEGACY

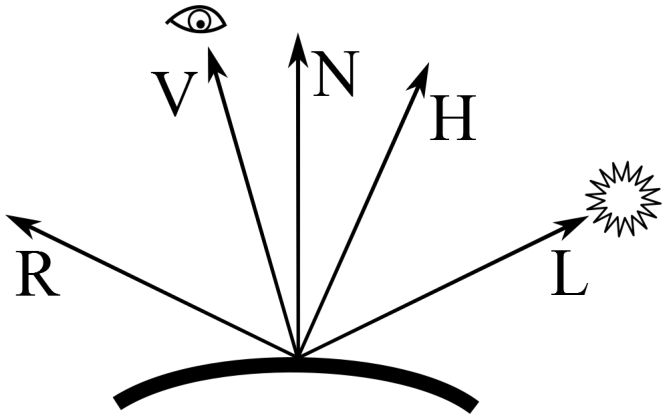
RICHARD PIPER - TEXTURE / SHADER ARTIST

NAUGHTY DOG

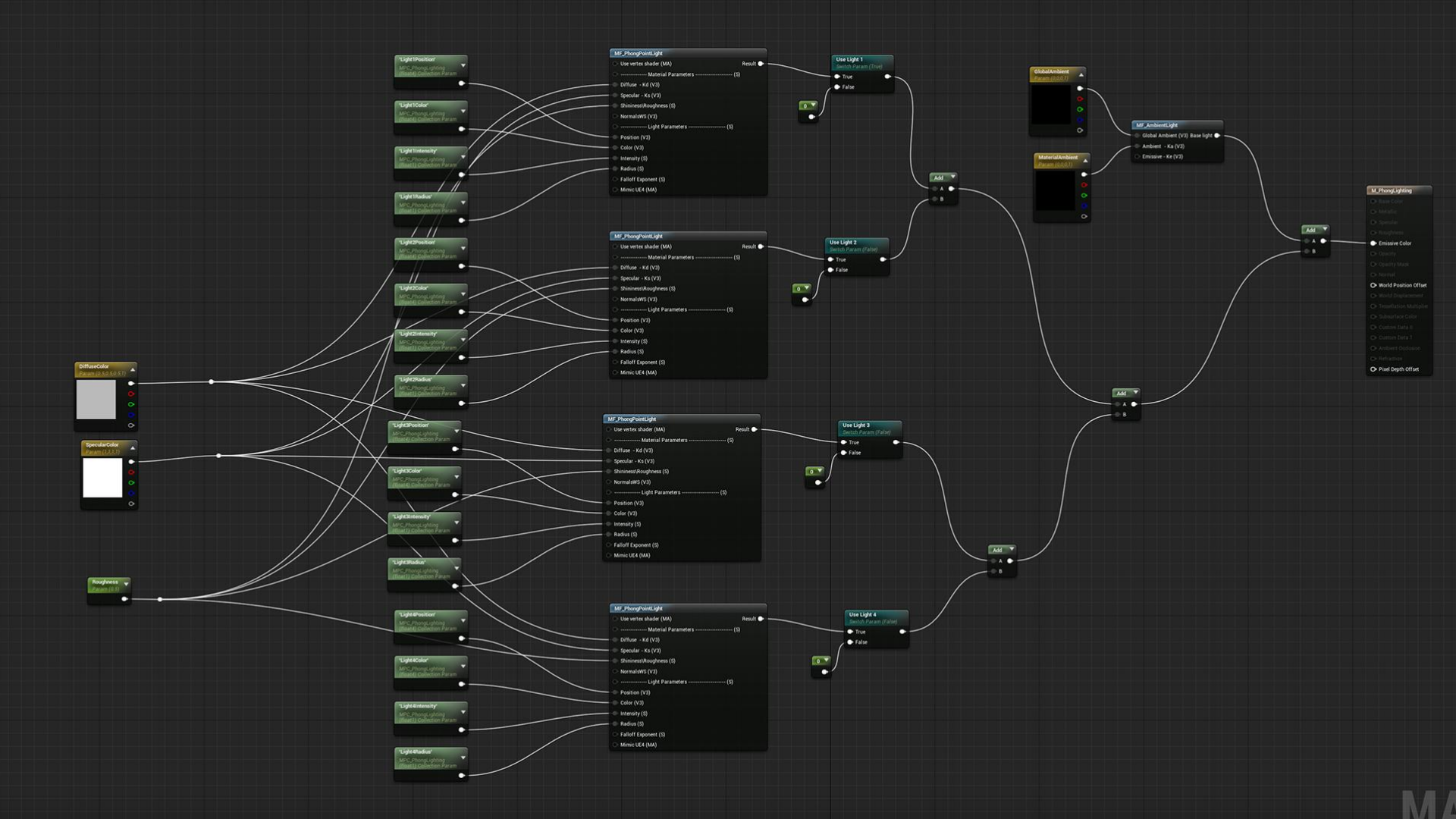
Phong shader

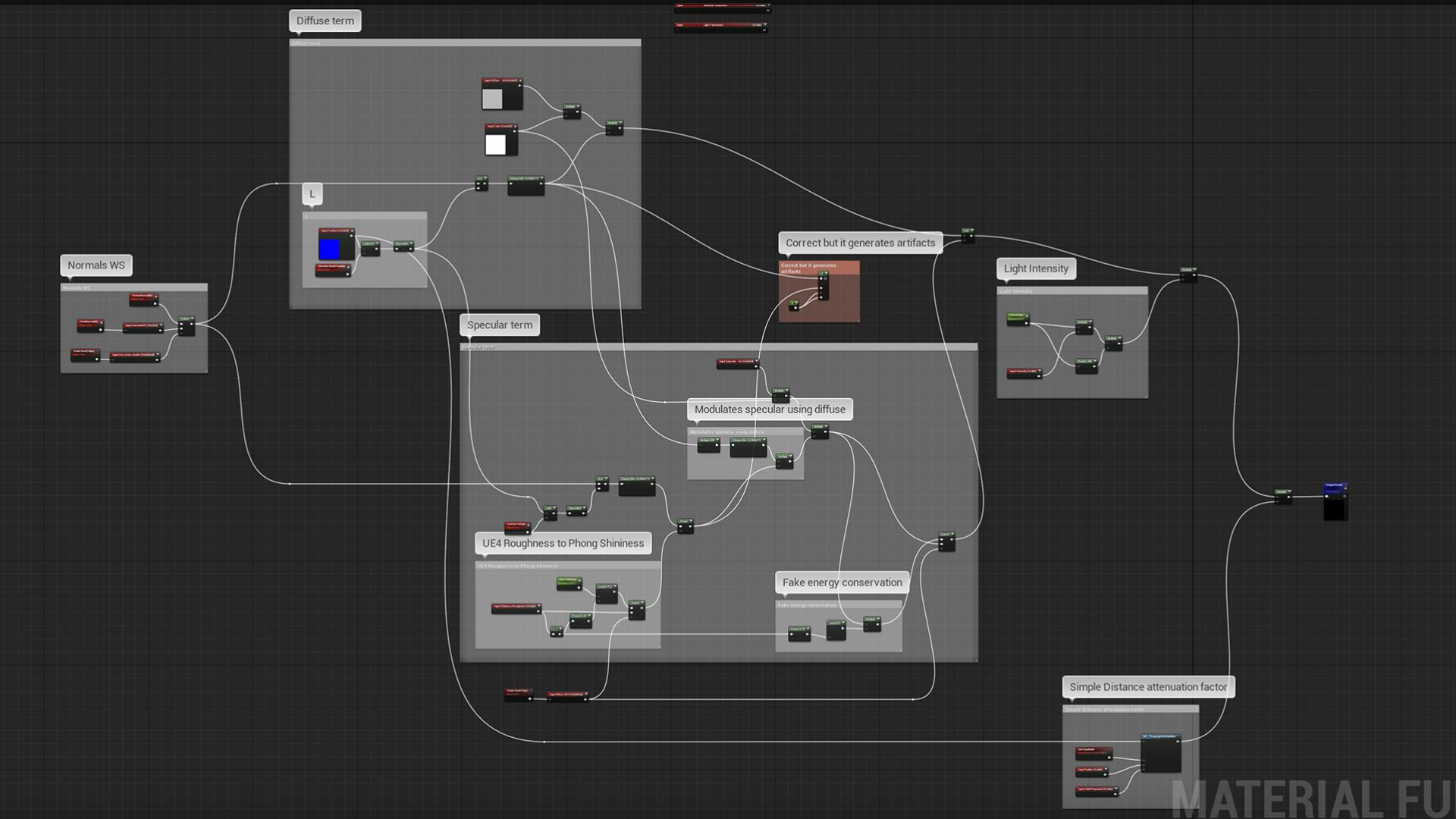


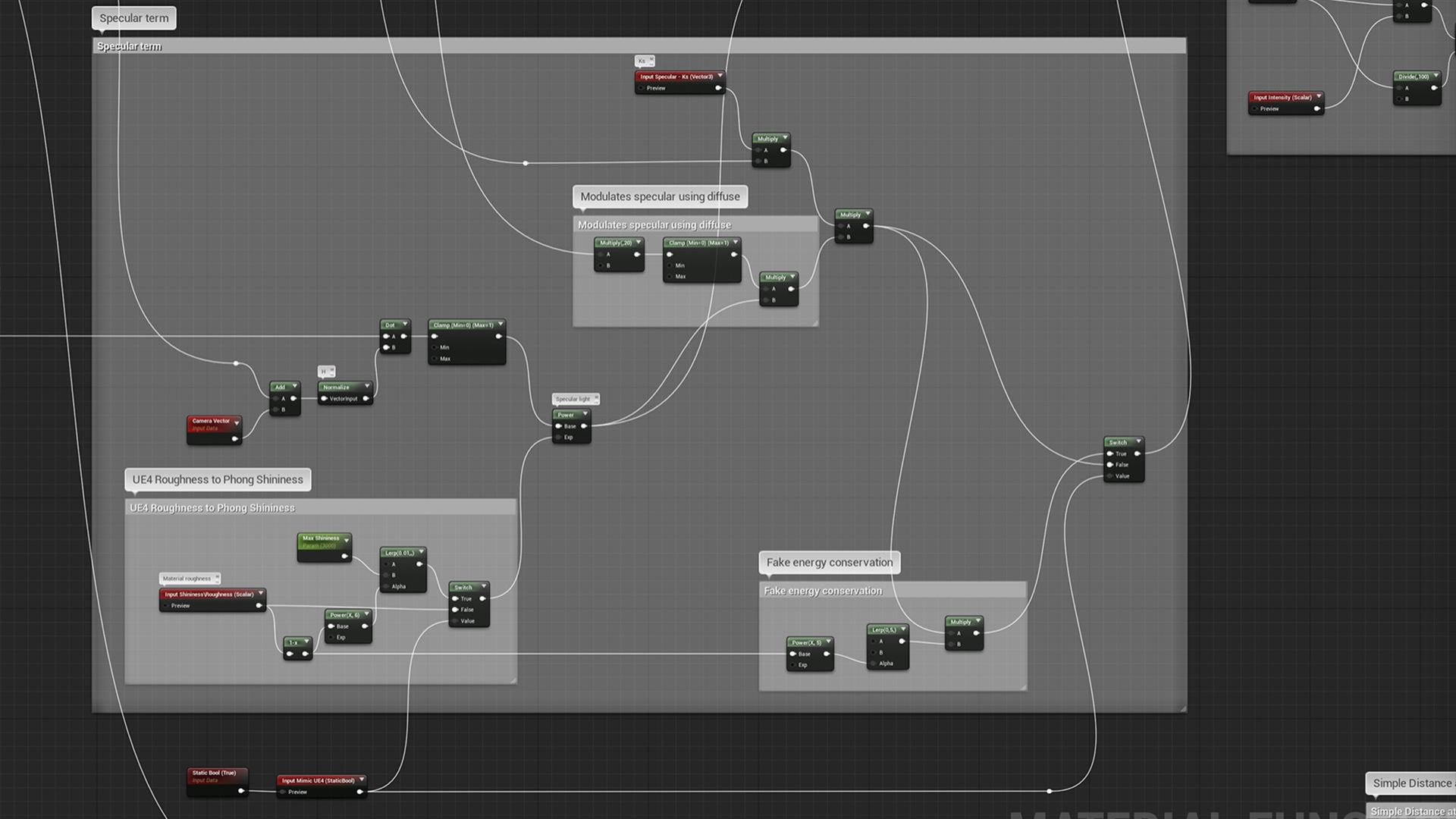




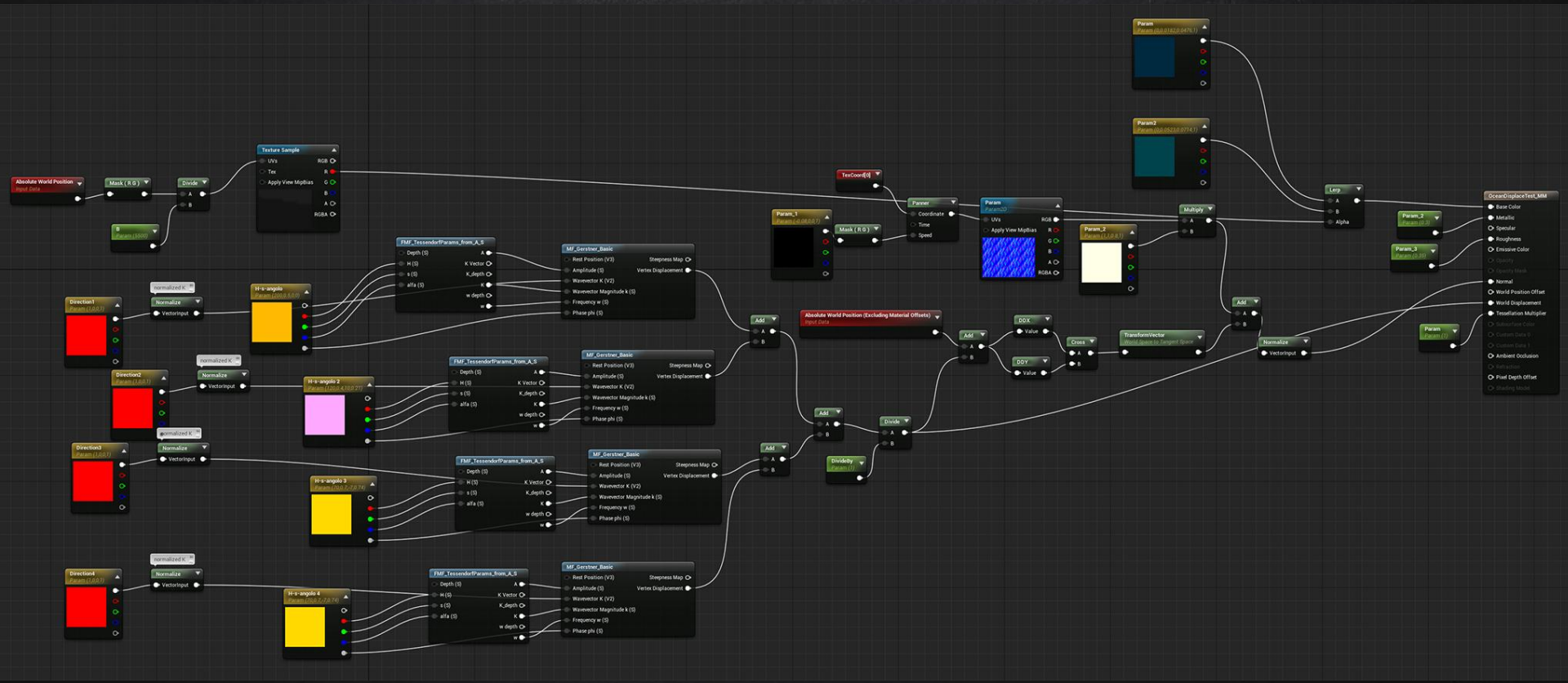
$$I_{\mathrm{p}} = k_{\mathrm{a}} i_{\mathrm{a}} + \sum_{m \in \text{lights}} \left(k_{\mathrm{d}} (\hat{L}_m \cdot \hat{N}) i_{m,\mathrm{d}} + k_{\mathrm{s}} (\hat{R}_m \cdot \hat{V})^{\alpha} i_{m,\mathrm{s}} \right).$$


DiffuseColor
Param (0.5,0.5,0.5,1)
SpecularColor
Param (1,1,1,1)
Roughness
Param (0.5)
'Light1Position'
'Light1Color'
'Light1Intensity'
'Light1Radius'
'Light2Position'
'Light2Color'
'Light2Intensity'
'Light2Radius'
'Light3Position'
'Light3Color'
'Light3Intensity'
'Light3Radius'
'Light4Position'
'Light4Color'
'Light4Intensity'
'Light4Radius'
MF_PhongPointLight
Use vertex shader (MA)
Result
Material Parameters (S)
Diffuse - Kd (V3)
Specular - Ks (V3)
Shininess\Roughness (S)
NormalsWS (V3)
Light Parameters (S)
Position (V3)
Color (V3)
Intensity (S)
Radius (S)
Falloff Exponent (S)
Mimic UE4 (MA)
Use Light 1
Switch Param (True)
Use Light 2
Use Light 3
Use Light 4
Switch Param (False)
True
False
0
Add
A
B
GlobalAmbient
Param (0,0,0,1)
MaterialAmbient
MF_AmbientLight
Global Ambient (V3) Base light
Ambient - Ka (V3)
Emissive - Ke (V3)
M_PhongLighting
Emissive Color
World Position Offset
Pixel Depth Offset


Diffuse term
L
Normals WS
Correct but it generates artifacts
Light Intensity
Specular term
Modulates specular using diffuse
UE4 Roughness to Phong Shininess
Fake energy conservation
Simple Distance attenuation factor


Specular term
Specular term
Ks
Input Specular - Ks (Vector3)
Preview
Multiply
A
B
Modulates specular using diffuse
Modulates specular using diffuse
Multiply(-20)
Clamp (Min=0) (Max=1)
Min
Max
Multiply
Dot
Clamp (Min=0) (Max=1)
H
Normalize
VectorInput
Add
Camera Vector
Input Data
Specular light
Power
Base
Exp
Switch
True
False
Value
UE4 Roughness to Phong Shininess
UE4 Roughness to Phong Shininess
Max Shininess
Lerp(0.01,1)
Material roughness
Input Shininess\Roughness (Scalar)
Power(X, 6)
1-x
Fake energy conservation
Fake energy conservation
Power(X, 6)
Lerp(0,5)
Multiply
Static Bool (True)
Input Data
Input Mimic UE4 (StaticBool)
Input Intensity (Scalar)
Divide(,100)
Simple Distance
Simple Distance a

Gerstner Wave Shader

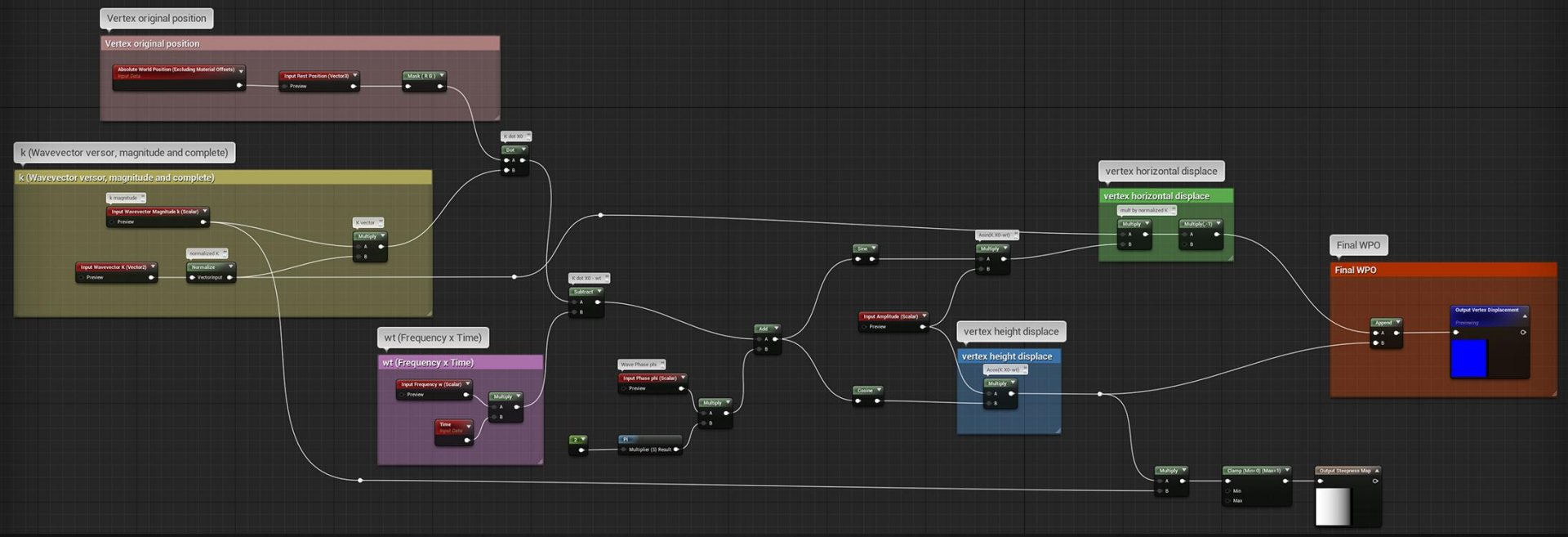

Absolute World Position
Input Data
Mask (R G)
Divide
B
Param (5000)
Texture Sample
UVs
Tex
Apply View MipBias
RGB
R
G
B
A
RGBA
Direction1
Param (1,0,0,1)
Direction2
Param (1,0,0,1)
Direction3
Param (1,0,0,1)
Direction4
Param (1,0,0,1)
normalized K
Normalize
VectorInput
H-s-angolo
Param (0.0,1,0,0)
H-s-angolo 2
Param (0.0,0.5,0.27)
H-s-angolo 3
Param (0.0,0.7,0.74)
H-s-angolo 4
Param (0.0,0.7,0.74)
FMF_TessendorfParams_from_A_S
Depth (S)
H (S)
s (S)
alfa (S)
A
K Vector
K_depth
K
w depth
w
MF_Gerstner_Basic
Rest Position (V3)
Amplitude (S)
Wavevector K (V2)
Wavevector Magnitude k (S)
Frequency w (S)
Phase phi (S)
Steepness Map
Vertex Displacement
Add
A
B
Divide
DivideBy
Param (1)
Param_1
Param (-0.08,0,0,1)
TexCoord[0]
Mask (R G)
Panner
Coordinate
Time
Speed
Param
Param20
UVs
Apply View MipBias
RGB
R
G
B
A
RGBA
Param_2
Param (1,1,0,0.1)
Absolute World Position (Excluding Material Offsets)
Input Data
DDX
Value
DDY
Value
Cross
TransformVector
World Space to Tangent Space
Multiply
Param
Param (0.0,0.1812,0.0476,1)
Param2
Param (0.0,0.0521,0.0714,1)
Lerp
Alpha
Normalize
Param_2
Param (0.7)
Param_3
Param (0.39)
Param
Param (1)
OceanDisplaceTest_MM
Base Color
Metallic
Specular
Roughness
Emissive Color
Opacity
Opacity Mask
Normal
World Position Offset
World Displacement
Tessellation Multiplier
Subsurface Color
Custom Data 0
Custom Data 1
Ambient Occlusion
Refraction
Pixel Depth Offset
Shading Model

Gerstner Wave (basic version):

it computes a single Gerstner wave influencing vertices using the basic equations
X = X0 - (K/k)Asin(K.X0 - wt + phi)
y = Acos(K.X0 - wt + phi)

Gerstner Wave (basic version):

it computes a single Gerstner wave influencing vertices using the basic equations
X = X0 - (K/k)Asin(K.X0 - wt + phi)
y = Acos(K.X0 - wt + phi)



RUINS
ARE
UPON
YOU
CAMP RULES
BOAT DOCK
THIS WAY

Grazie!

elisa@untoldgames.com