VEKTORIA-MANUAL
TEXTURIERUNG

Prof. Dr. Tobias Breiner

vektoria

www.vektoria-engine.com

Inhalt

Überblick über Knotenobjekte für Oberflächen

Knotenobjekte der Szenegraphen

1 2 3 4 5

Virtuelle Szene

Gruppenknoten
- Gruppenbehälter
- Transformationen
- Switchknoten (Level-of-Detail, Animation etc.)
- …

Blattknoten
- Lichter
- Geometrien
- Kameras
- Materialien, Texturen, Images
- Farben
- ...

Reale Szene

Gruppenknoten
- Gruppenbehälter
- Computer
- Kanäle
- Switchknoten (an, aus)
- …

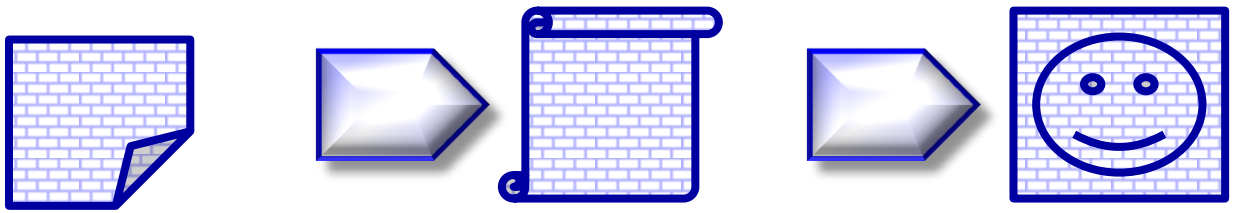
Blattknoten
- Eingabegeräte
- Sichtsysteme
- Sichtfenster
- Soundgeräte
- Basis-Render-API
- …

Kapitel 2

Überblick über Knotenobjekte für Oberflächen

Unterscheiden Sie bei Oberflächen!

1. Material
2. Textur
3. Bild

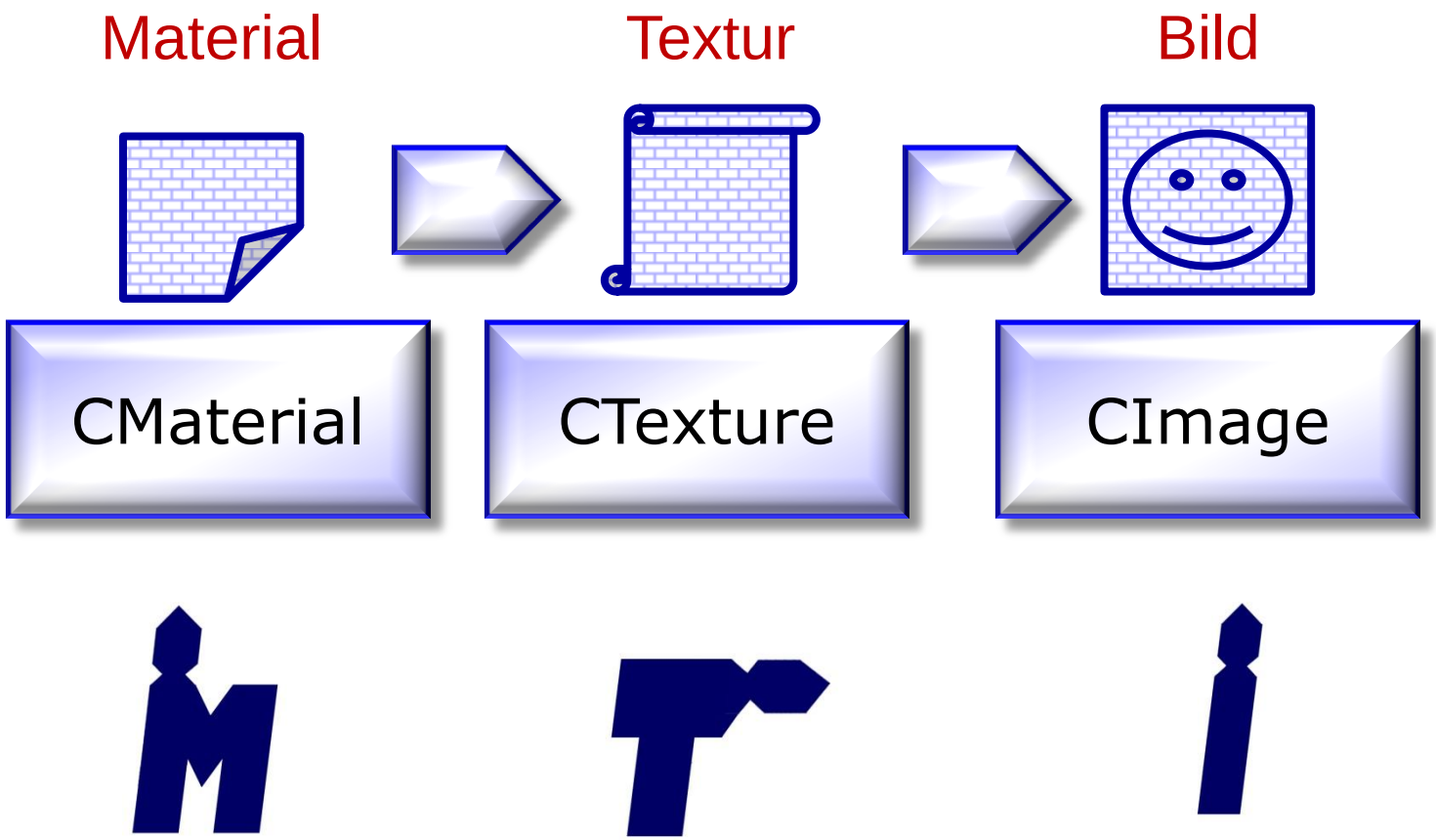


Ein Material wird auf eine Geometrie aufgebracht (ge-mappt). Es kann mehrere Texturen enthalten. Zusätz-lich hat es eventuell physikalische Parameter (Rollwiderstand, etc.)

Eine Textur hat optische Parameter und eventuell einen Verweis auf ein Bild. Es gibt verschiedene Arten von Texturen (Image, Glow, Bump, etc.)

Ein Bild enthält die eigentlichen 2D-Pixeldaten. Es hat einen Verweis auf eine JPG-, BMP-… oder TGA-Datei.

Oberflächenklassen in Vektoria



Kapitel 3

Images

Ein Bild enthält 2D-Pixeldaten.
Es hat einen Verweis auf eine Datei.
Es gibt verschiedene Bilddateiformate.

Je nach GameEngine/Szenegraf sind andere Formate ladbar, oft möglich sind z.B.:

- JPG
- BMP
- TGA
- PNG
- GIF

Auch die Bittiefe (8 bit, 24 bit, 32 bit) und die Kanalanzahl müssen beachtet werden!

CImage

Initialisierungs- und Finalisierungsmethode:

1.
2. void Init(char * acPath);
3.
4. void Fini();
5.

acPath gibt den kompletten Pfad der Textur an (inklusive Dateisuffix).
Es sind absolute und relative Pfade möglich.

CImage

Vektoria kann zurzeit folgende Dateieformate lesen:

- JPG, JPEG
- JPE,
- BMP & DIB
- TGA
- TIF, TIFF
- PNG
- GIF

Jeweils monochromatisch, trichromatisch (RGB) und trirochromatisch mit Alpha (RGBA)
Auch verschiedene Bittiefen (8 bit, 24 bit, 32 bit) sind möglich

Pixelbilder

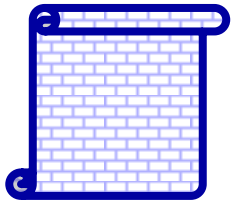
Images

Falls Sie nicht nur mit Vektoria arbeiten, sondern auch mit einem anderen Szenegrafen bzw. einer anderen Game Engine, müssen Sie bei der Auswahl der Bilddateien aufpassen:

- Viele Game Engines können keine komprimierten Bilddateiformate wie JPG, JPEG, PNG oder GIF verarbeiten.
- Die meisten tun sich bei der Verarbeitung von BMP schwer.
- Auch bei der Annahme der Bittiefe (2, 8, 16, 24, 32 bit) und der Kanalanzahl gibt es beträchtliche Unterschiede.
- Dateien mit Alphakanälen werden nicht von allen Game Engines/Szenegrafen verarbeitet.
- Einige akzeptieren nur Bilddateien, deren Kantenpixelanzahl einer Zweierpotenz entspricht (z.B. 64*64 Pixel)

Kapitel 4

Texturen allgemein



Synonyme in anderen Szenegrafen: Skins, Texsurfaces, Maps

Texturen werden gemäß folgender Kriterien kategorisiert:

- Zielort: Surface, Background …
- Herkunft: Prozedurale vs. Image Texturen
- Anzahl: Single-, Dual-, Multitextures
- Interpretation: Farbgebung (i.d.R.), Dot3-Bumpmap, Lightmap, Shadowmap, Vertex Displacement Map …
- Animation: statische Texturen vs. Videotexturen …
- Transparenz: Alpha-Texturen, Chromakeying-Texturen, opake Texturen
- Granularitätsstufen: Mipmapped (mit und ohne Blending)
- Aufbringungsweise: clamped, tiled, scaled, …
- Dimension: 1D, 2D, 3D, ….

CMaterial: Die Materialklasse in Vektoria

Folgende Texturarten sind in Vektoria möglich:

• Diffuse	Farb- und Helligkeitstextur für die beleuchtete Seite
• Glow	Farb- und Helligkeitstextur für die unbeleuchtete Seite
• Specular	Textur, welche die Stärke der Glanzlichter, Reflektionen und Refraktionen angibt
• Bump	Normalenmodifikationstextur für das Bumpmapping
• Height	Höhentextur für das Parallax-Occlution-Mapping
• Environment	Reflexions-Umgebungstextur
• Thickness	Dickentextur für das Subsurface-Scattering
• Sky	Spezielle Textur für Skydomes, Skycubes, etc.
• Sprite	Spezielle Textur für 2D-Sprites (Overlays, Writings Wribels, Billboards und Backgrounds)

CMaterial: Die Materialklasse in Vektoria

Die Textursetzmethoden der Klasse CMaterial:

1. void SetTextureDiffuse(CTexture *ptexture);
2. void SetTextureGlow(CTexture *ptexture);
3. void SetTextureSpecular(CTexture *ptexture);
4. void SetTextureBump(CTexture *ptexture);
5. void SetTextureHeight(CTexture *ptexture);
6. void SetTextureEnvironment(CTexture *ptexture);
7. void SetTextureThickness(CTexture *ptexture);
8. void SetTextureSky(CTexture *ptexture);
9. void SetTextureSprite(CTexture *ptexture);

ptexture ist ein Zeiger auf ein Texturobjekt

CMaterial: Schnelltexturierungsmethoden

Die Schnelltexturierungsmethoden:

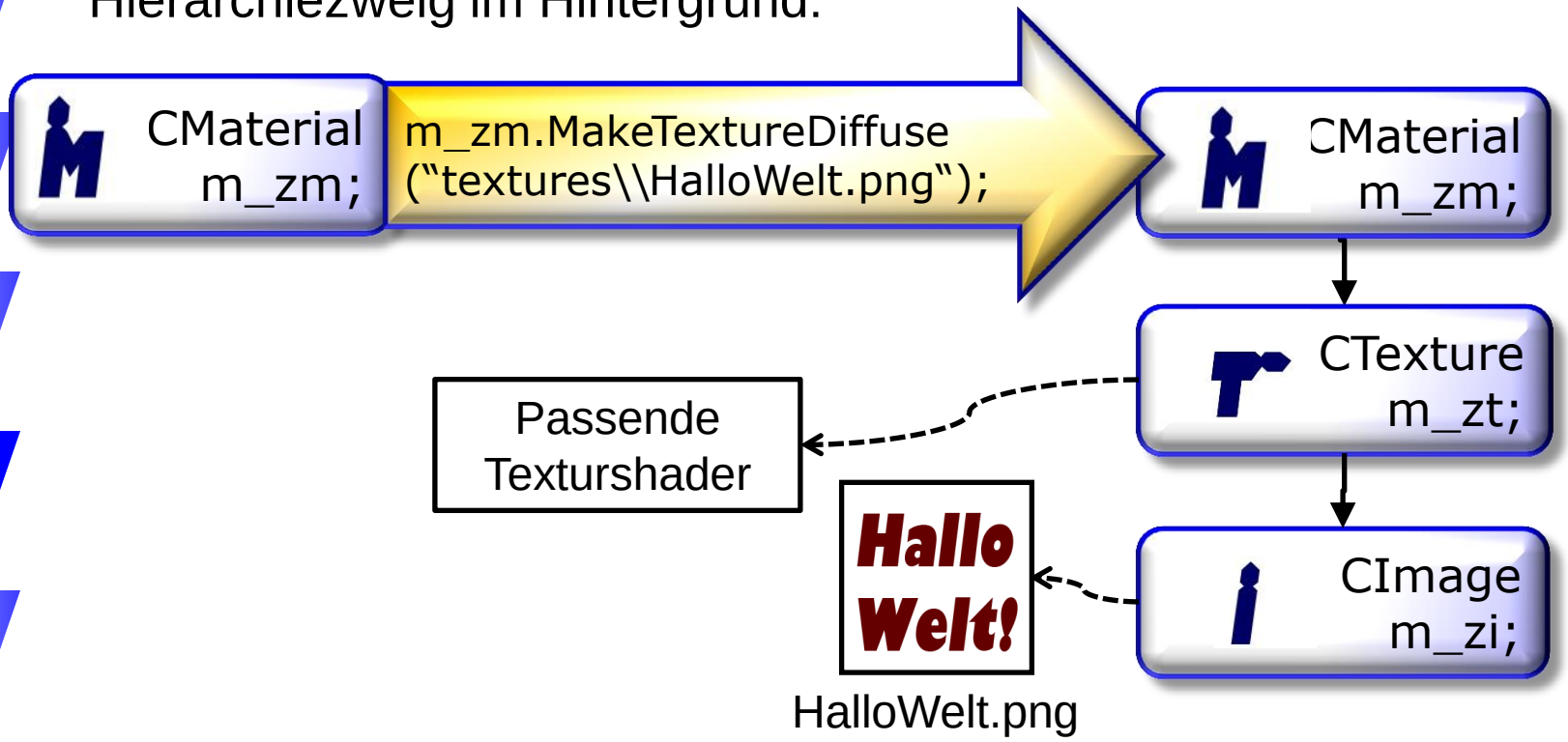
1. CImage * MakeTextureDiffuse (char * acPath);
2. CImage * MakeTextureGlow(char * acPath);
3. CImage * MakeTextureSpecular(char * acPath);
4. CImage * MakeTextureBump(char * acPath);
5. CImage * MakeTextureHeight(char * acPath);
6. CImage * MakeTextureEnvironment(char * acPath);
7. CImage * MakeTextureThickness(char * acPath);
8. CImage * MakeTextureSky(char * acPath);
9. CImage * MakeTextureSprite(char * acPath);

Es wird ein Zeiger auf das erzeugte Image-Objekt zurückgegeben.

acPath ist ein Pfad auf die Bilddatei. Es sind absolute und relative Pfade möglich. Achtung! Das Dateisuffix muss mit angegeben werden!

CMaterial: Die Materialklasse in Vektoria

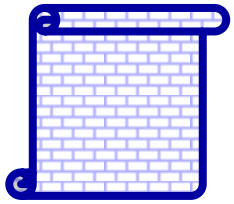
Die Schnelltexturierungsmethoden der Klasse CMaterial erzeugen mit einem einzigen Befehl einen ganzen Hierarchiezweig im Hintergrund:



Übung zu Materialien

1. Erzeugen Sie ein Material mit einer beliebigen Diffuse-Textur! Mappen Sie das Material auf eine Kugel!
2.
3. Freiwillige Zusatzaufgabe für die schnellen Nerds:
   - Erzeugen Sie zusätzlich noch eine Glow-Map!
4.
5.

Texturmappingmethodem

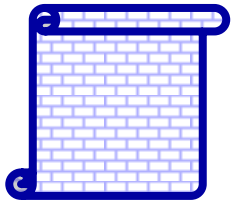


Texturabschneidung
(texture clamping)

Texturkachelung
(texture tiling)

Texturskalierung
(texture scaling)

Texturen mit Alphakanal



Zusätzlicher Alphakanal (zu Rot-, Grün- und Blaukanal)
gibt Transparenz an

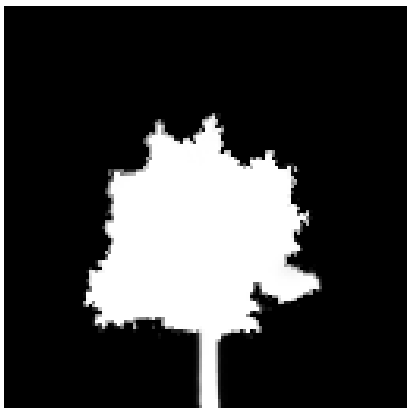
1

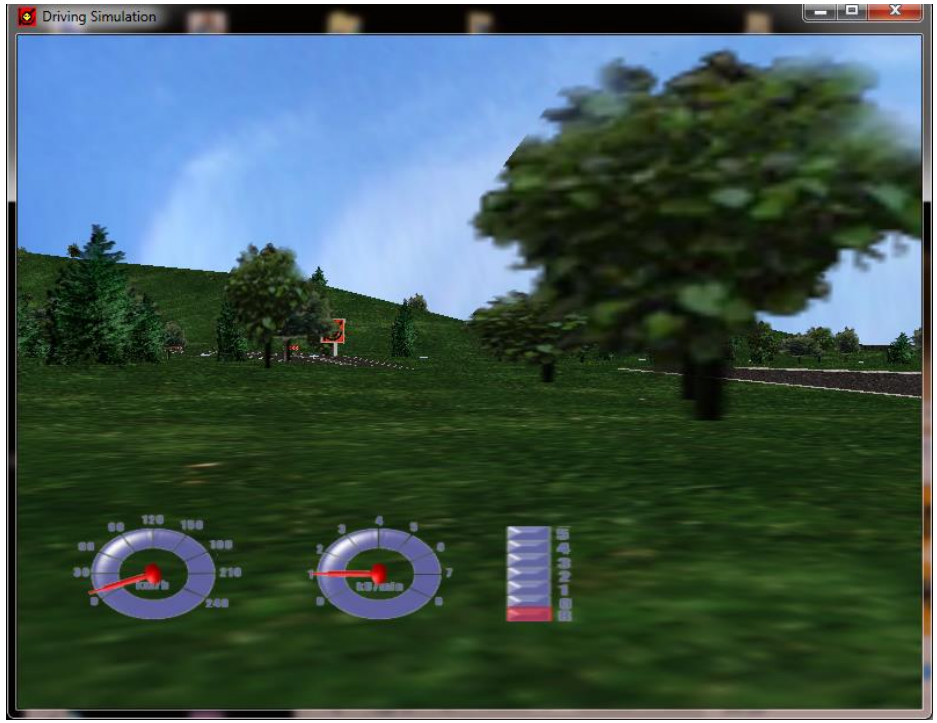
2

3

4

5





Material mit Alphakanaltextur

1. Um eine Alpha-Imagetextur anzuzeigen, reicht es nicht, dem Material mit Hilfe der Methode MakeTextureDiffuse den Pfad einer Textur mit Alphakanal zu übergeben!
2. Es muss zusätzlich der Alphakanal mit der Methode SetTransparencyOn eingeschaltet werden!
3.
4.
5.

Material mit Alphakanaltextur

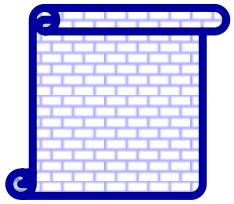
Soll ein Material mit einer gleichmäßigen Transparenz angezeigt werden, wird keine Aplhatextur benötigt,

Es kann der Alphawert direkt mit

SetTransparency(float frTransparency) angegeben werden. Der fraktionale Parameter sollte zwischen 0.0 und 1.0 liegen. Dabei gilt z.B.:

```
SetTransparency(0.0f); // opak (total undurchsichtig)
SetTransparency(0.1f); // wenig transparent (10%)
SetTransparency(0.3f); // wenig transparent (30%)
SetTransparency(0.5f); // halbtransparent (50%)
SetTransparency(0.8f); // sehr transparent (80%)
SetTransparency(1.0f); // total unsichtbar (sinnlos ;-))
```

Texturen mit Farbschlüssel



Chromatischer Schlüssel (hier: Magenta) kann ebenfalls Transparenz anzeigen

- Vorteil: Kein zusätzlicher Alphakanal notwendig
- Nachteil: Keine Semitransparenz möglich

Material mit Farbschlüssel

Um ein Material mit einer Farbschlüsseltextur zu erzeugen, muss einfach nur einmal beim Material die Methode

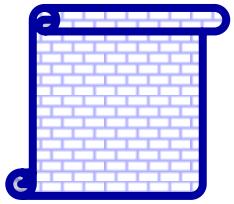
```
SetChromaKeyingOn()
```

aufgerufen werden

Das linke obere Pixel wird dann automatisch als Farbschlüssel (engl. chroma key) genommen.

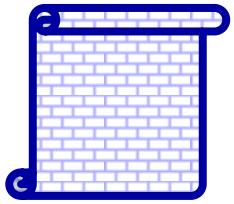
Achtung! Nur Pixel, deren RGB-Werte exakt dem Farbschlüssel entsprechen, werden transparent angezeigt.

Bumpmapping



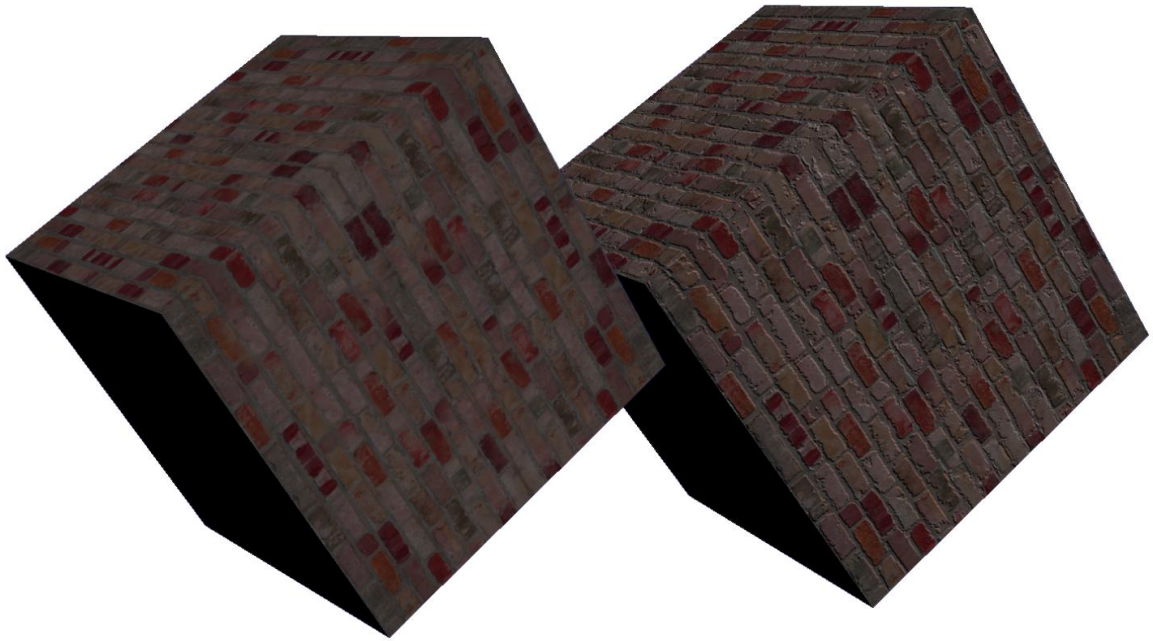
- Dot3-Bumpmapping erzeugt eine Oberflächenreliefanmutung durch Modifikation der Normalenvektoren, eine Szene wirkt dadurch sofort komplexer und realistischer, aber es bietet…
  - …keine räumliche Verschiebung bei Kameraanimationen.
  - …keine Selbstokklusion (Verdeckung hinterer Strukturen).
  - …keine Silhouettenbildung.
  - …keine Vesikelbildung (biologische Abspaltung).
- Abhilfe würde das Quaoaring-Bumpmapping liefern, das 2001 an der Universität Frankfurt entwickelt wurde,
  - es ist bisher allerdings in keiner einzigen Engine implementiert!
- Zumindest die ersten drei Mankos kann Displacement Mapping vermeiden.
  - Vektoria bietet jedoch kein Displacement Mapping an,
  - aber bald!

Dot3-Bumpmapping



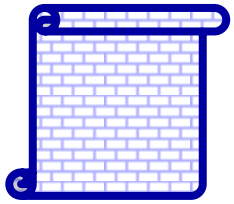
Dot3-Bumpmapping erzeugt durch Helligkeitsvariation den Anschein einer reliefartigen Oberflächenstruktur.

Damit sehen Objekte wesentlich plastischer und echter aus.



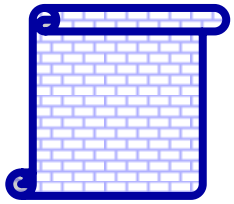
ohne Bumpmapping mit Bumpmapping

Dot3-Bumpmapping



ohne Bumpmapping | mit Bumpmapping

Bumpmapping

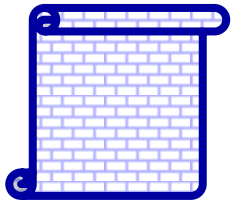


Beim DOT3-Bumpmapping werden die Normalenverktoren mit Hilfe des Vektorproduktes (engl. dot 3 product) „verbogen“. Dazu werde die RGB-Werte der Bumpmapping-Textur folgendermaßen umiterpretiert:

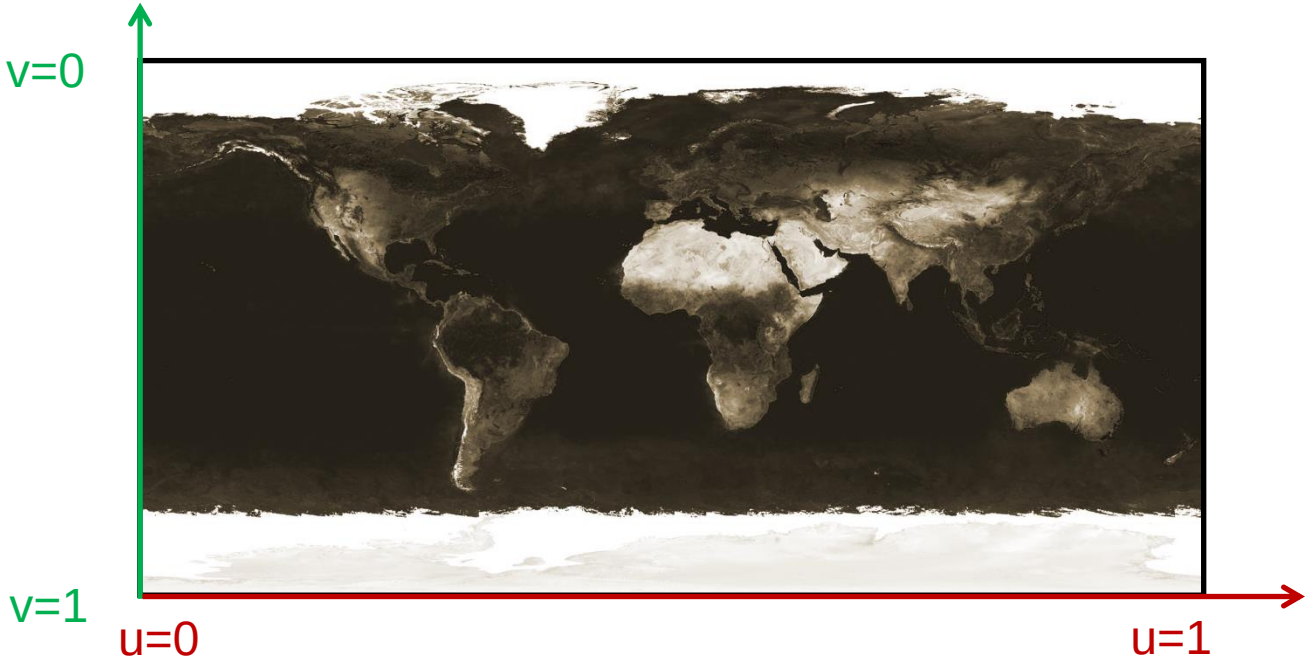
Rot:	Skalar in Richtung der Oberflächentangente
Grün:	Slalar in Richtung der Oberflächenbiitangente
Blau:	Skalar in Richtung der Oberflächennormale

Der Urspung liegt bei RGB= 128,128,128.
Da meist in Richtung der Oberflächennormalen verschoben wird, sehen DOT3-Bumpmaptexturen bläulich aus (RGB= 128,128,256).

Bumpmapping



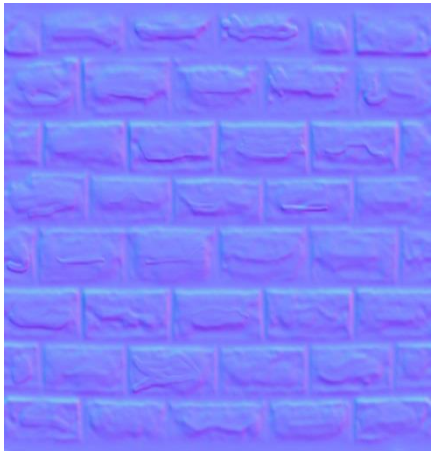
Tangente und Bitangente orientieren sich am UV-Mapping:
Die Tangente befindet sich stets in positiver U-Richtung,
die Bitangente stets in negativer V-Richtung.
Dies muss bei gespiegelten Texturen beachtet werden!



Erzeugung Bumpmap-Textur

Image-Textur

Highmap-Textur

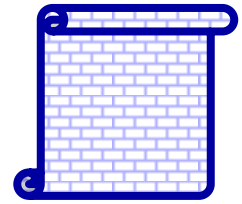


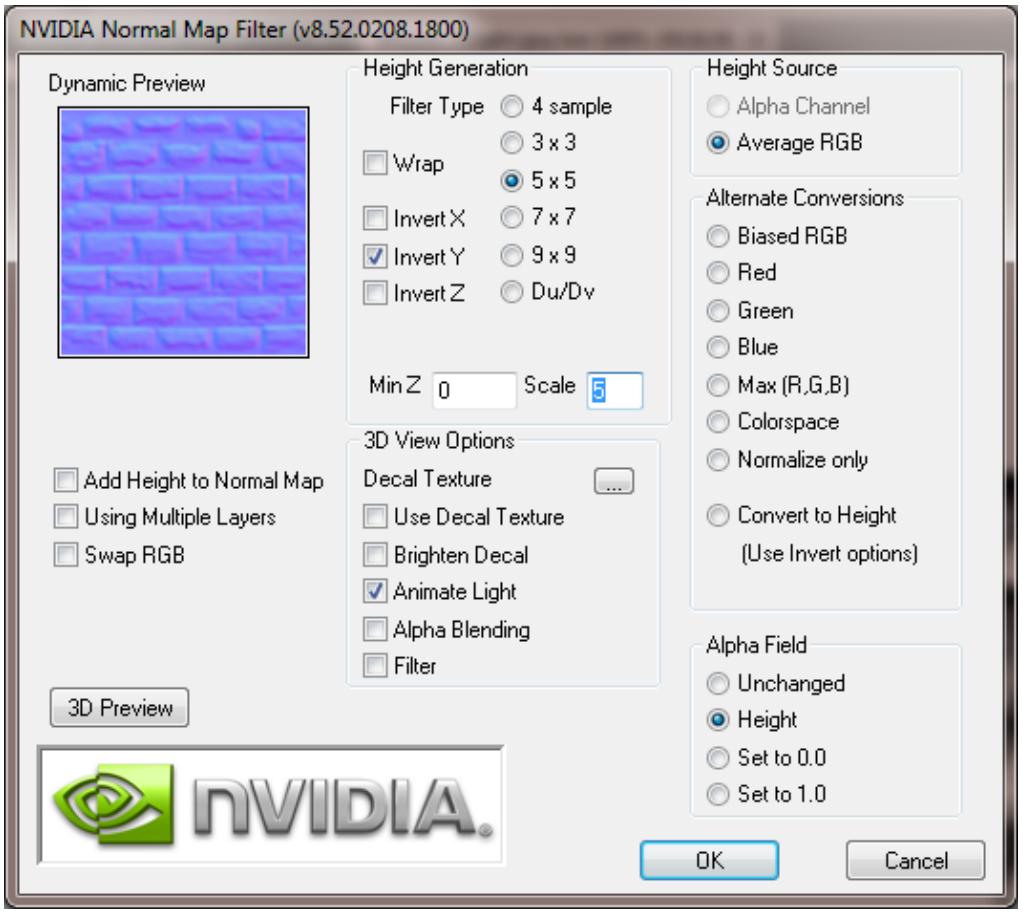
Bumpmap-Textur

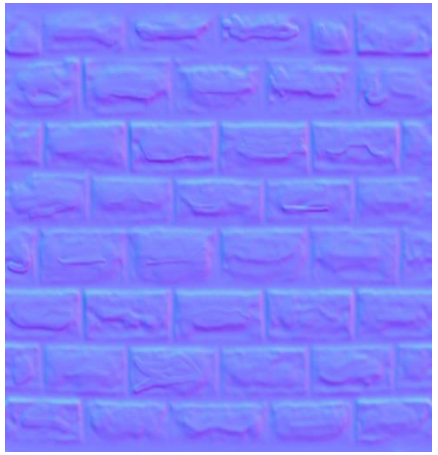
Pixelbearbeitungs-programm

Normalmap-Filter

Einstellungen Normal Map Filter

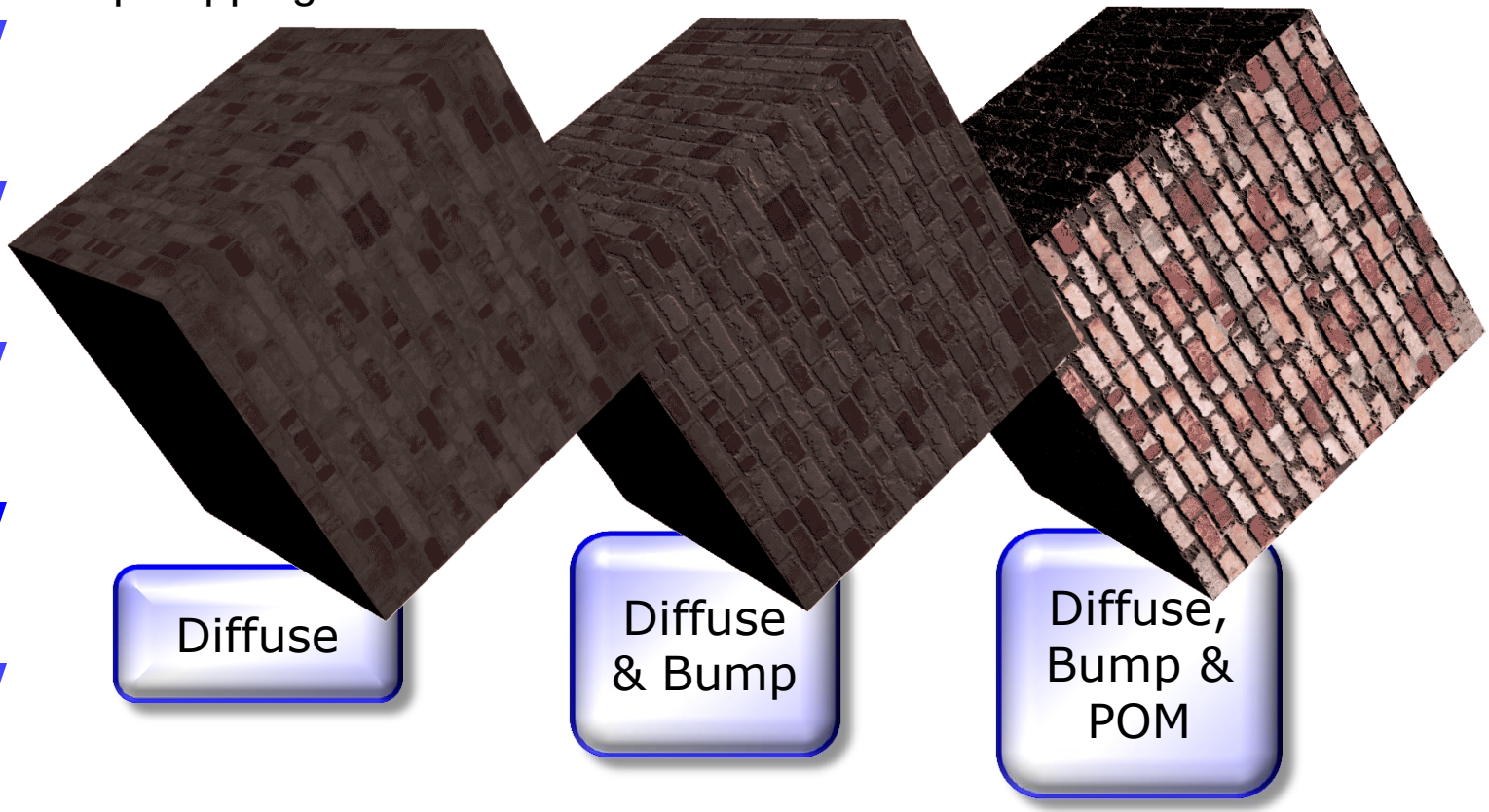




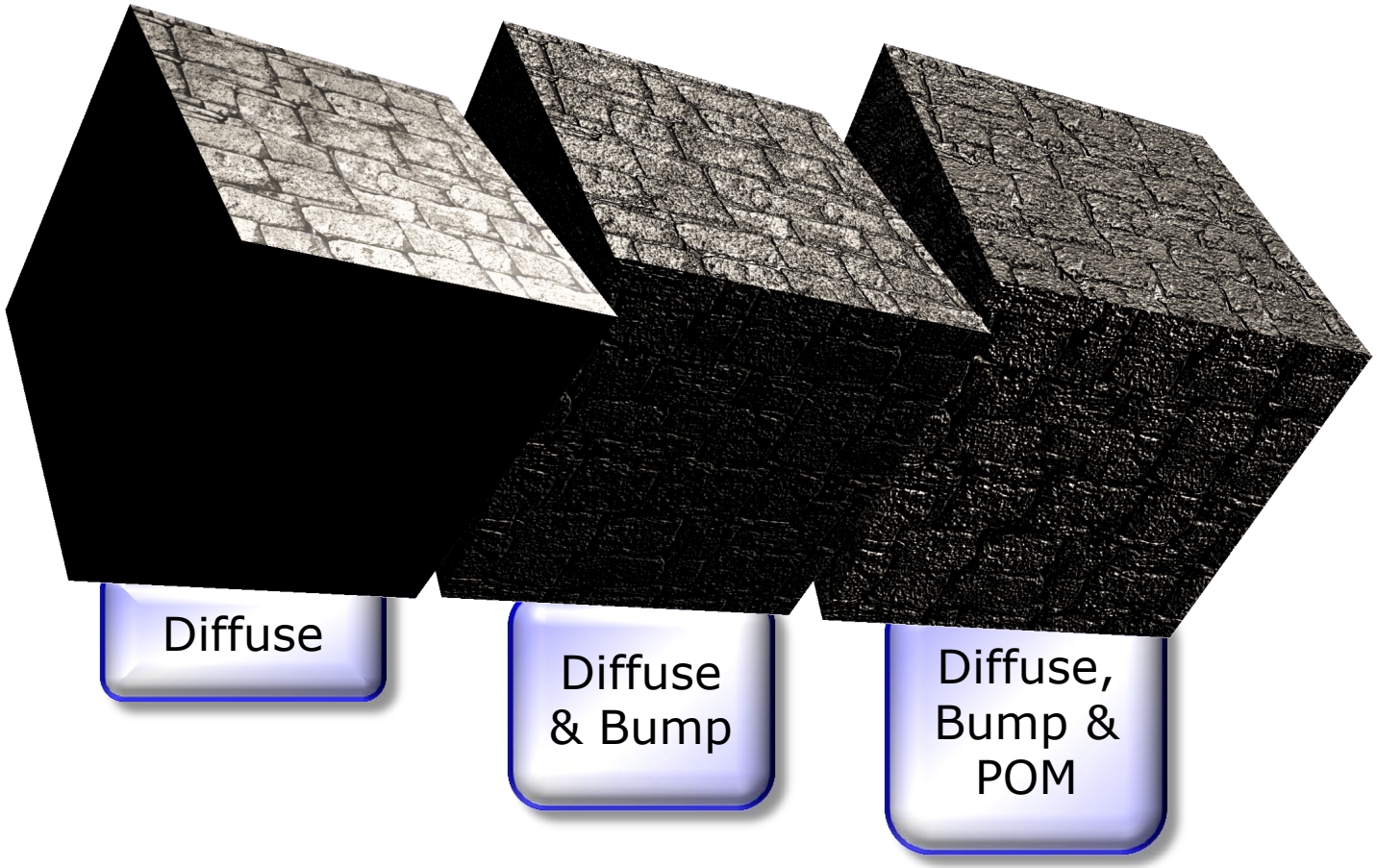


Parallax Occlution Mapping (POM)

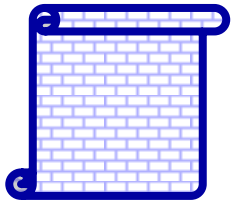
Parallax Occlution Mapping kann die Reliefstruktur des Bumpmappings noch erhöhen.



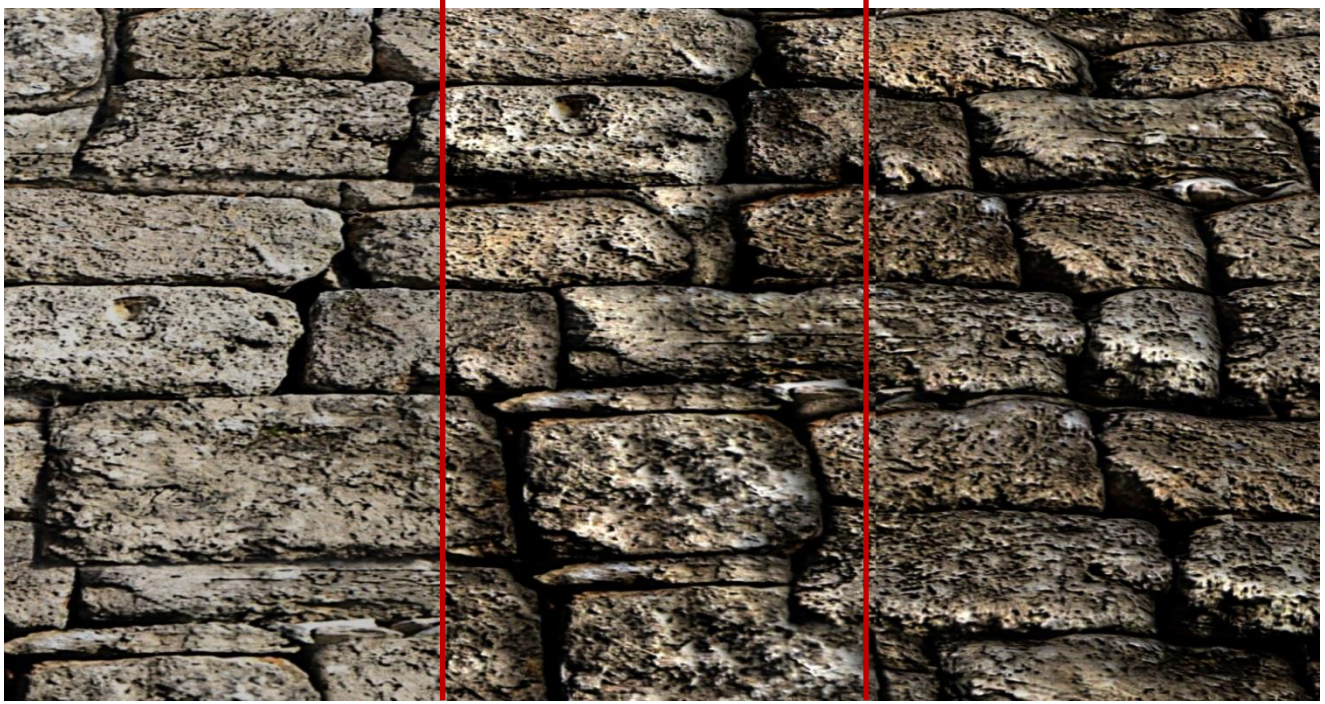
Parallax Occlution Mapping (POM)



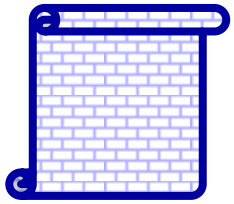
Parallax Occlution Mapping



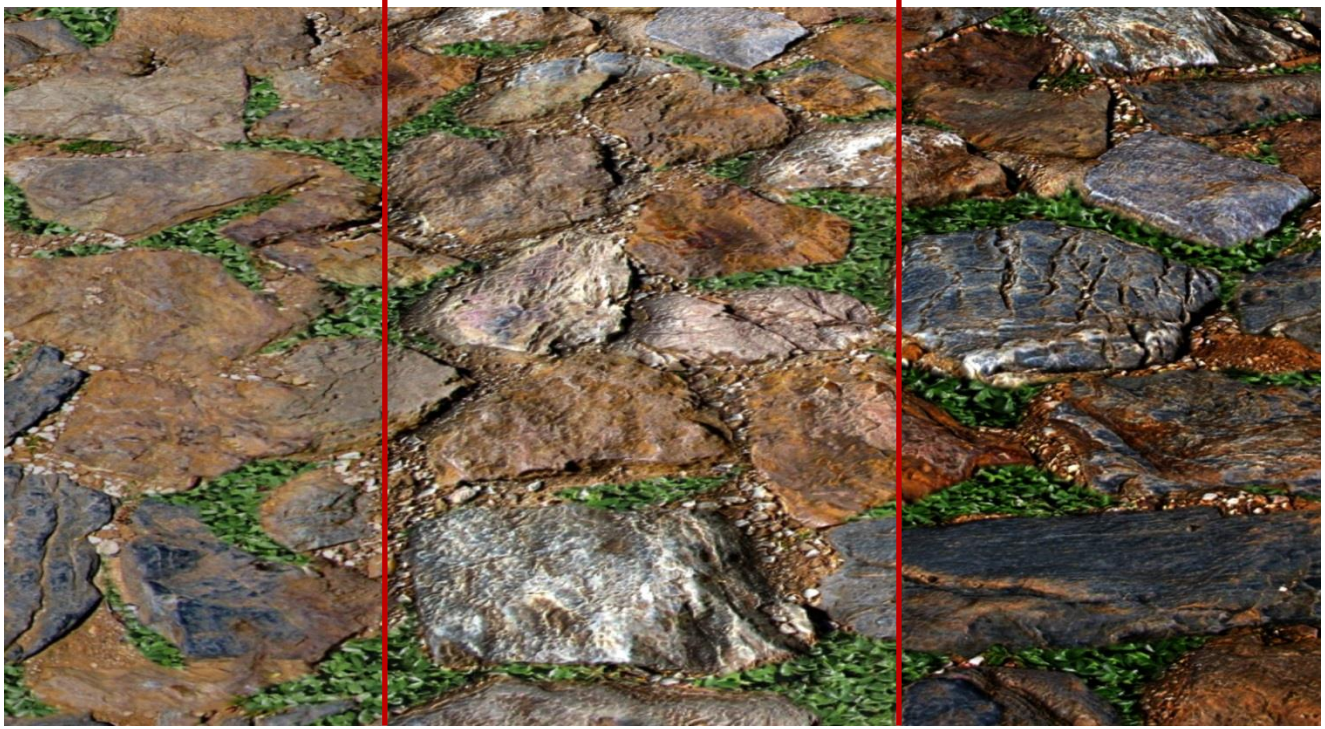
ohne Bumpmapping | mit Bumpmapping | mit Bumpmapping & POM



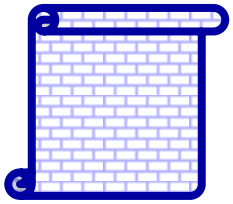
Parallax Occlution Mapping

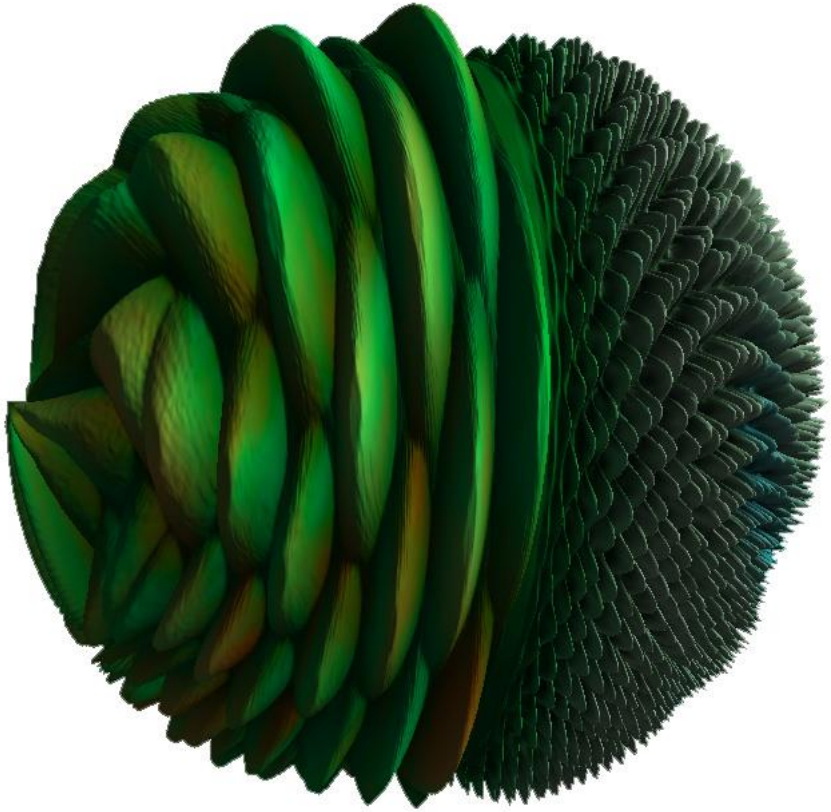


ohne Bumpmapping | mit Bumpmapping | mit Bumpmapping & POM

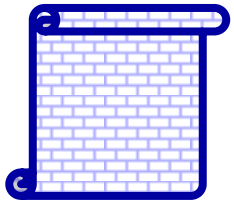


Quaoaring-Bumpmapping





Spekulare Textur



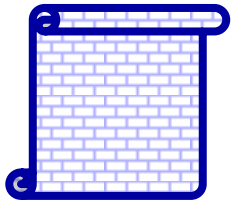
Die RGB-Werte der spekularen Textur geben an, welche Stellen der Oberfläche eine:

- Refraktion bzw. Lichtbrechung, (Rotkanal)
- Reflektion bzw. Lichtspiegelung und (Grünkanal)
- diffuse Spekularität bzw. Glanzlicht (Blaukanal)

aufweisen. Refraktion und Reflexion können nur in Verbindung mit einer sphärischen Umgebungstextur (spherical environmental map) dargestellt werden. Ist keine Umgebungstextur vorhanden, bleiben Rot- und Grünkanal unberücksichtigt.
Der Blaukanal zeigt an, wo matte (niedrige Blau-Werte) und glänzende (hohe Blau-Werte) Oberflächenstellen vorhanden sind.

Die Summe aller RGB-Kanalwerte sollte möglichst nicht 255 übersteigen!

Spekulare Texturwerte

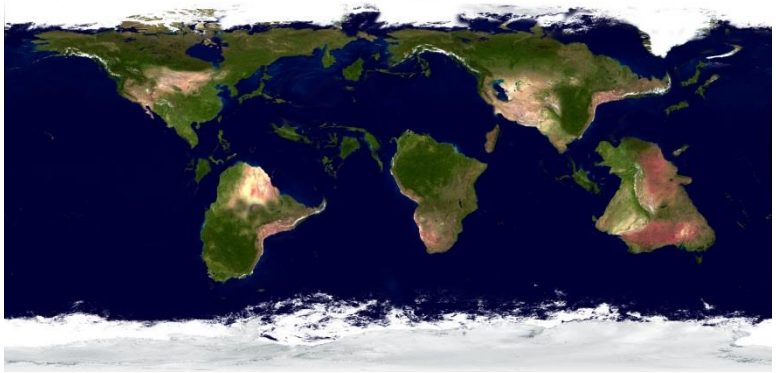


Damit ergeben sich folgende häufig gebrauchte Farbwert-Kombinationen für die spekulare Textur:

Schwarz	RGB(0,0,0)	matt
Nachtblau	RGB(0,0,63)	seidenmatt
Dunkelblau	RGB(0,0,127)	leicht glänzend
Blau	RGB(0,0,255)	stark glänzend
Rot	RGB(255,0,0)	refraktiv
Dunkelrot	RGB(127,0,0)	leicht refraktiv und matt
Grün	RGB(0,255,0)	spiegelnd
Dunkelgrün	RGB(0,127,0)	leicht spiegelnd
Olive	RGB(127,127,0)	halb spiegelnd, halb refraktiv

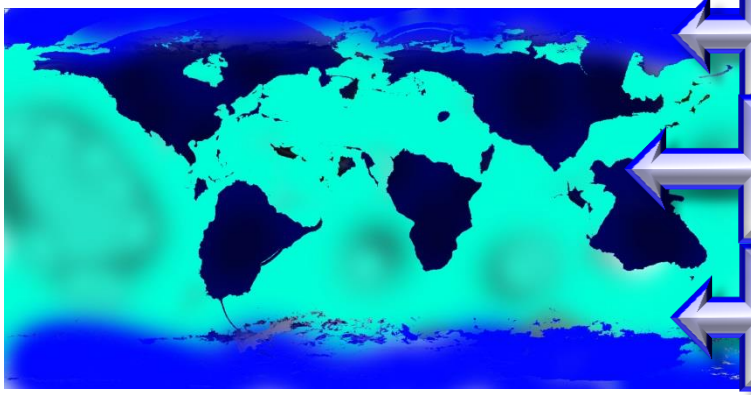
Spekulare Textur (Beispiel)

Diffuse Textur:



Ergebnis (zusammen mit anderen Texturen):

Passende Spekulare Textur:



- Die Eiskappen glänzen stark (Blau).
- Die Kontinente glänzen nur leicht (Dunkelblau).
- Die Ozeane glänzen und spiegeln (Cyan).

Spekulare Textur (Beispiel)

Spekulare Textur (Ergebnis)

Diffuse Textur:

Spekulare Textur:

Ergebnis mit Umgebungstextur:

Environmental

Mit Environmental-Maps kann man Spiegelungen faken.

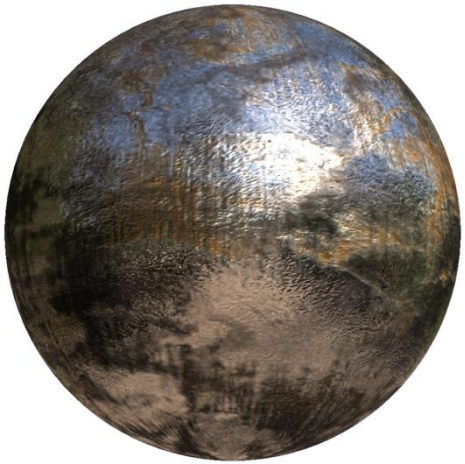
Damit lassen sich gut folgende reflektierende Materialiennachbilden, wie Glas, Aluminium, Chrom, Spiegel, Autolack, etc.

Die Environmental-Map muss dabei In Vektoria spärisch gemappt sein.

Sie werden vom Grün-Kanal der Specular-Map gesteuert.

Environmental & Bump

1
2
3
4
5

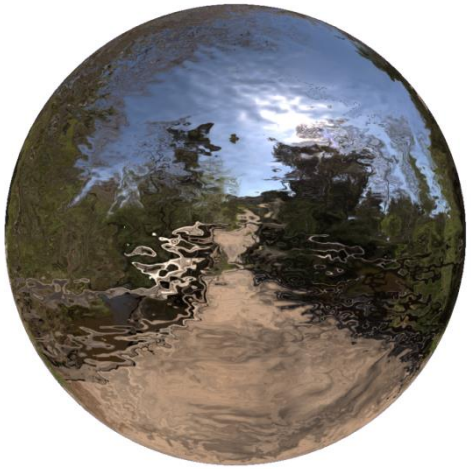


Rostige Metallkugel

Specular-, Environment-, Bump- und Diffuse-Mapping

Goldplatten-Kugel

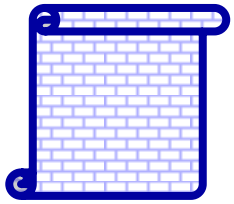
Specular- Environment- Bump- und Diffuse-Mapping



Quecksilber-tropfen

Specular- Environment- Refraction- und Bump-Mapping sowie SetAniOn

Animation von Texturen



- Videotexturen, „Animated GIF“-Texturen
- Veränderung der Ausgangsparameter bei Prozeduralen Texturen
- Texture Warping (Verändern der Texturkoordinaten)
- Texture Weighting (Verändern der Gewichte bei Multitextures über die Zeit)
- Generelle Parametermodifikation (z.B. Änderung des Höhenfaktors bei Bumpmaps oder des Transparenzwertes einer Textur über die Zeit)
- Ersetzen der Textur (bzw. Texturlink verändern)

CTexture: Die Texturklasse in Vektoria

Die Grundmethoden:

1 2 3 4 5

```
void Init(
CImage * pimage,
int eKind = 0);
```

Initialisiert den Computer. Der Parameter eKind kann zurzeit folgende Werte annehmen:
S_TEXTURE_DIFFUSE
S_TEXTURE_GLOW
S_TEXTURE_SPECULAR
S_TEXTURE_BUMP
S_TEXTURE_HEIGHT
S_TEXTURE_ENVIROMENTAL

```
void Fini();
```

Finalisiert die Texturklasse.

Kapitel 5

CMaterial: Die Materialklasse in Vektoria

Die Grundmethoden:

1. void Init();

   Initialisiert ein Material (optional, Konstruktor reicht aus)

2.

3.

4. void Fini();

   Finalisiert ein Material. (ebenfalls optional)

5.

Übung „Goldkugel“

1

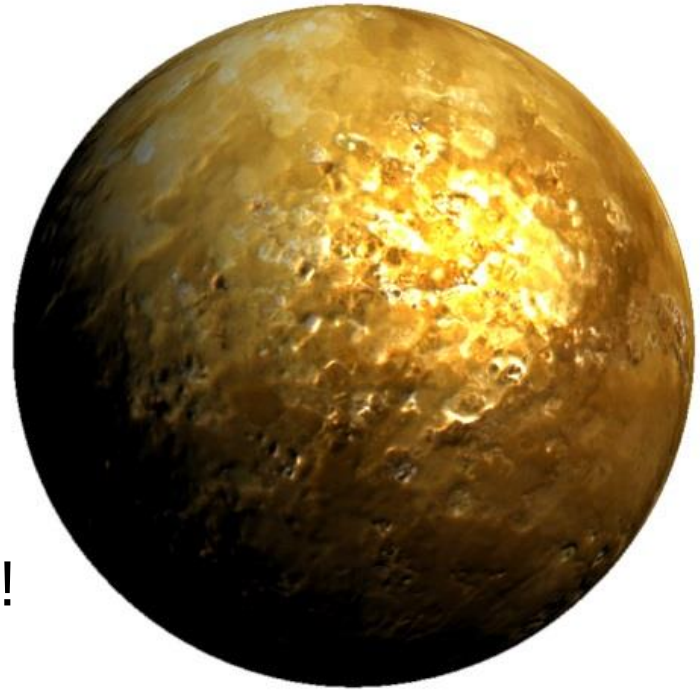
2

3

4

5

Erzeugen Sie mittels der Schnelltexturierungs-methoden eine möglichst realistische Goldkugel aus dem Märchen „Hans im Glück“!

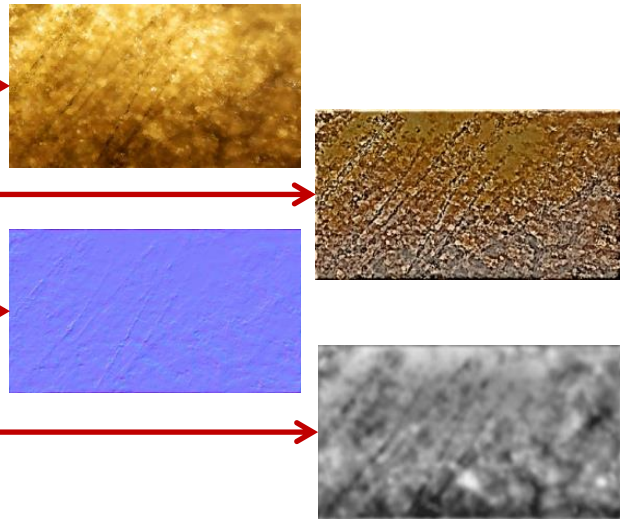


Freiwillige Übung für die schnellen Nerds:
Lassen Sie die Gold-kugel spontan aufblitzen!

Lösung „Goldkugel“ (1/2)

Nehmen Sie die „Hallo Kugel!“-Lösung aus dem vorherigen Foliensatz und erweitern sie diese um folgendes Material!

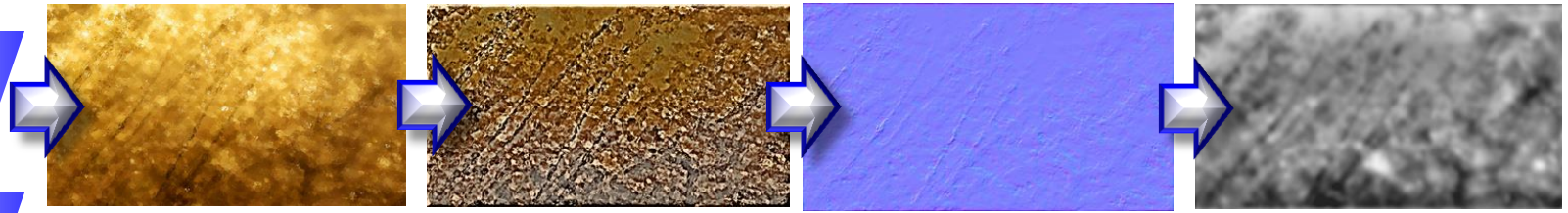
```
…
m_zm.MakeTextureDiffuse
     (“gold_diffuse.jpg“);
m_zm.MakeTextureSpecular
     (“gold_specular.jpg“);
m_zm.MakeTextureBump
     (“gold_bump.jpg“);
m_zm.MakeTexturePOM
     (“gold_pom.jpg“);
…
```

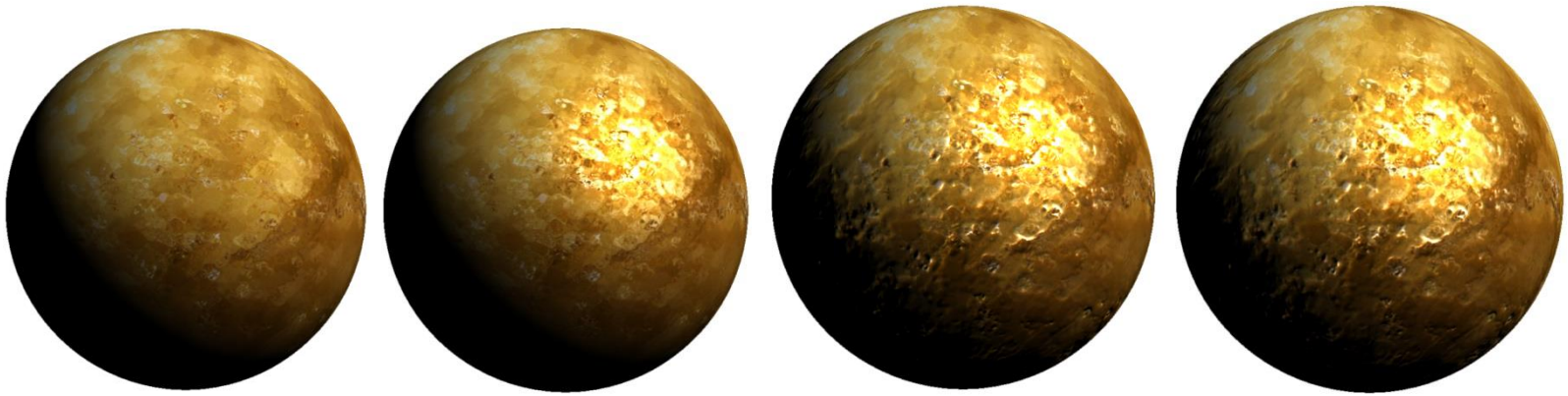


Lösung „Goldkugel“(2/2)

1 2 3 4 5

Diffuse Specular Bump POM





Übung zu Materialien

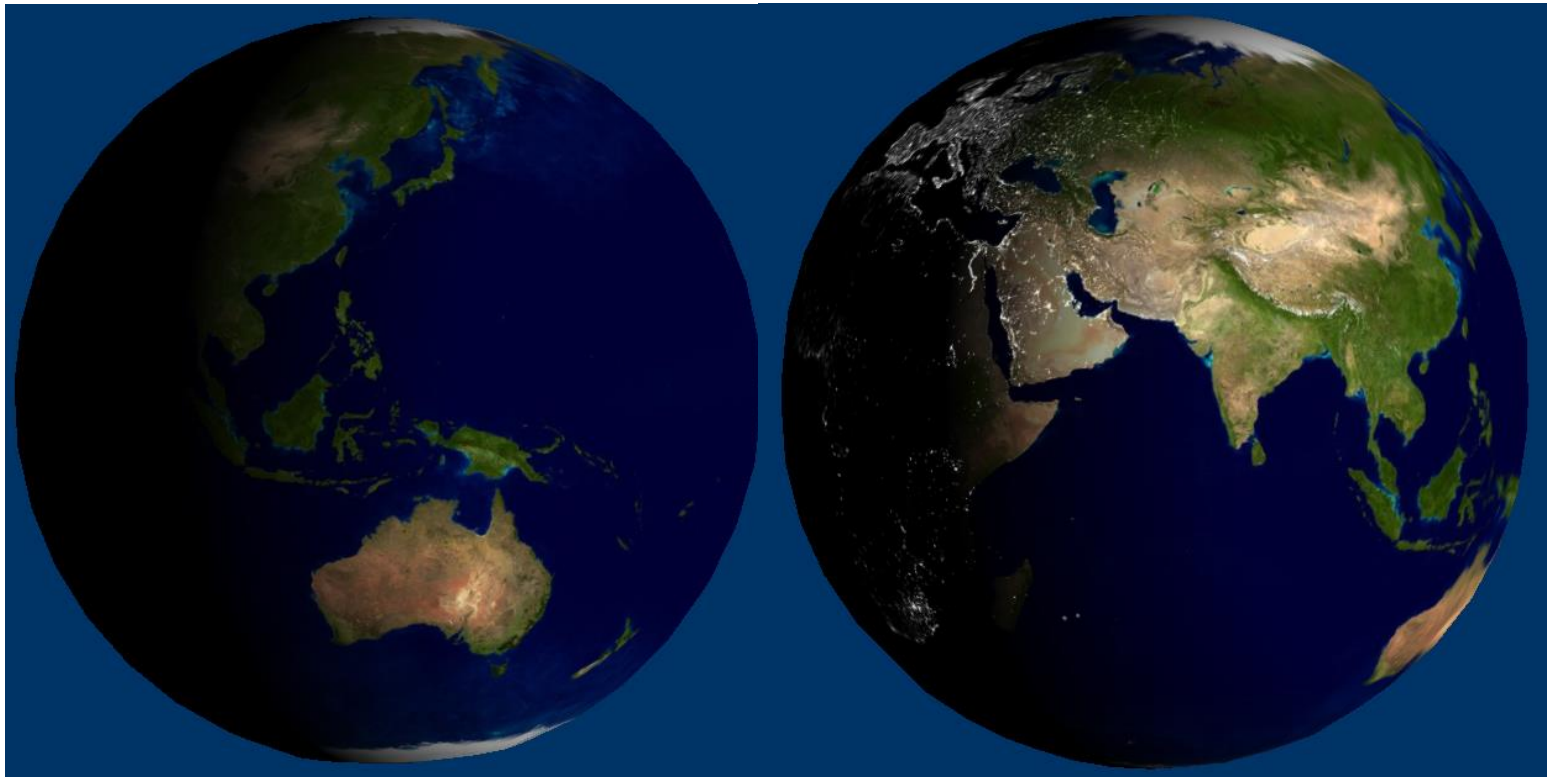
Erzeugen Sie mit Hilfe der Schnelltexturierungsmethoden eine Erde mit

- Diffuse Map
- Glow Map (Städte leuchten auf Nachtseite der Erde)
- Specular Map (Ozeane glänzen mehr als Festland)
- Bump Map (Oberflächenrelief durch Berge und Täler)

Freiwillige Zusatzaufgabe für die schnellen Nerds:

- Erzeugen Sie die Texturen selbst mit Adobe Photoshop, um einen erdähnlichen fiktiven Planeten zu erzeugen!

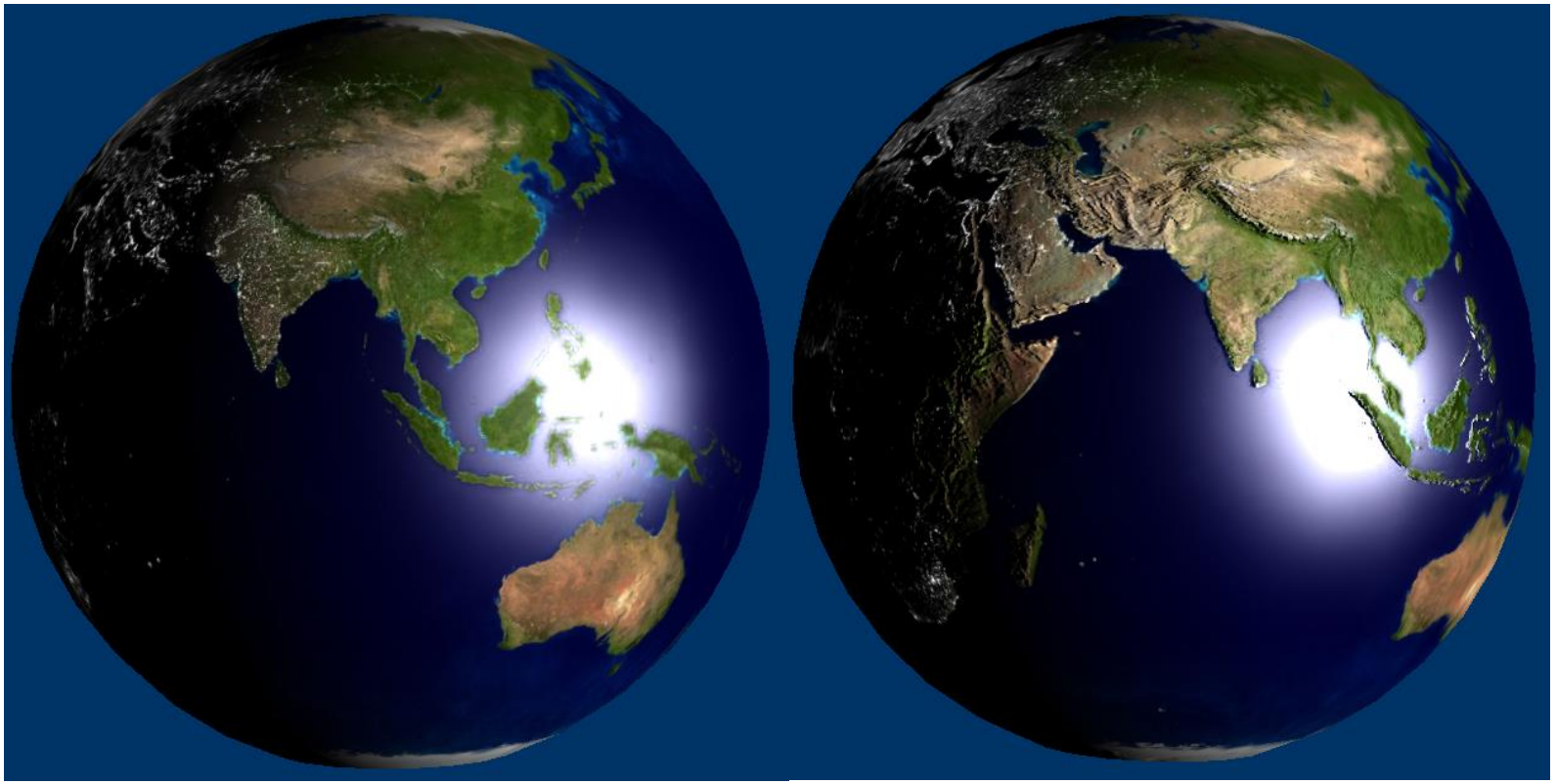
Lösungsausgabe



nur Diffuse

Diffuse & Glow

Lösungsausgabe



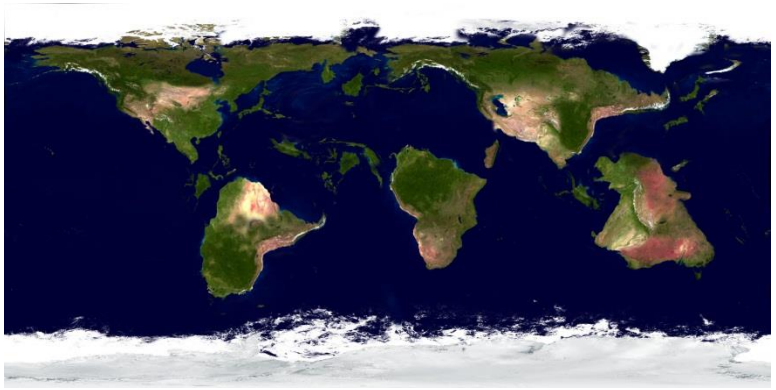
Diffuse, Glow & Specular

Diffuse, Glow, Specular & Bump

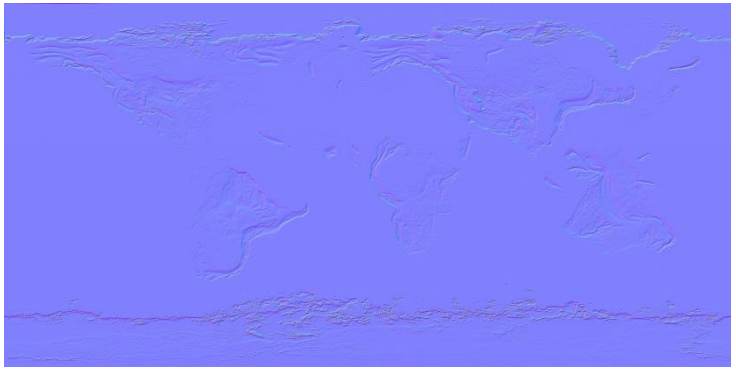
Lösungsausgabe

Nahaufnahme des Übergangs zwischen Glow und Image

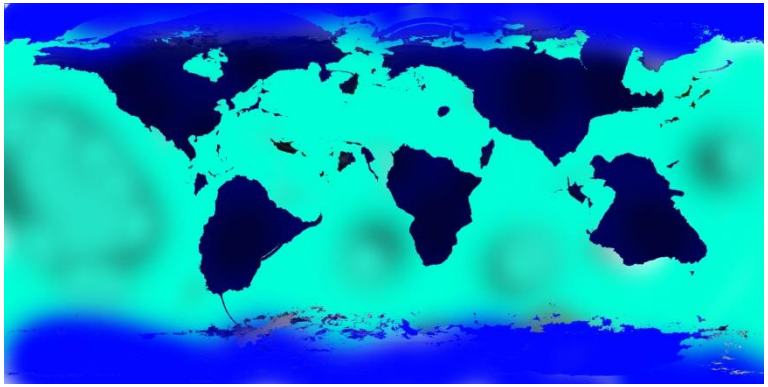
Nerdlösung Planet



Diffuse Textur



Bump-Textur



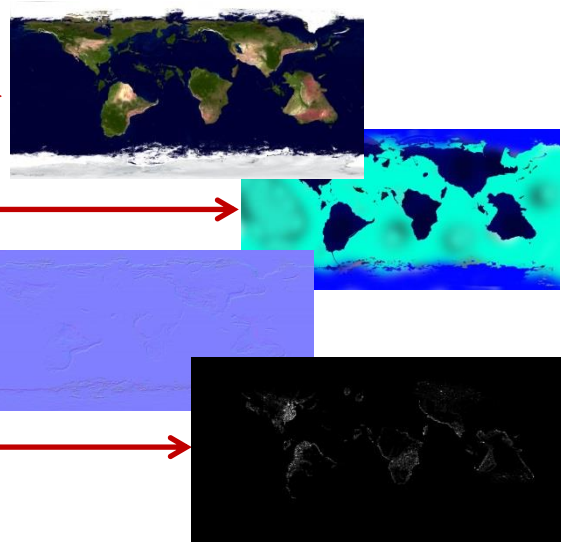
Spekulare Textur

Glow-Textur

Nerdlösung Planet

Nehmen Sie die „Hallo Kugel!“-Lösung aus dem vorherigen Foliensatz und erweitern sie diese um folgendes Material!

```
…
m_zm.MakeTextureDiffuse
     (“planet_diffuse.jpg“);
m_zm.MakeTextureSpecular
     (“planet_specular.jpg“);
m_zm.MakeTextureBump
     (“planet_bump.jpg“);
m_zm.MakeTextureGlow
     (“planet_glow.jpg“);
…
```



CMaterial: Die Materialklasse in Vektoria

Die Schaltmethoden der Klasse CMaterial:

1
2
3
4
5

```
void SetShadingOn();
void SetShadingOff();
```

Kann das gesamte Shading für Testzwecke an- und ausschalten.

CMaterial: Die Materialklasse in Vektoria

Die Glow-Modifikationsmethoden der Klasse CMaterial:

1. `void SetTextureGlowWhite();` // leuchtet im Schatten
2. `void SetTextureGlowBlack();` // dunkel im Schatten // (Default)
3. `void SetTextureGlowAsDiffuse();` // selbes Bild wie bei Diffuse
4. `void SetTextureGlowAsAmbient();` // leuchtet in der // Ambientfarbe
5. Verändert das Aussehen der lichtabgewandten Seite.

CMaterial: Die Materialklasse in Vektoria

Die Specular-Modifikationsmethoden der Klasse CMaterial:

1. `void SetTextureSpecularWhite();` // Helles Glanzlicht (Default)
2. `void SetTextureSpecularBlack();` // Kein Glanzlicht
3. `void SetTextureSpecularAsDiffuse();` // Glanzlicht verwendet
   // dasselbe Bild
   // wie diffuse Textur
4. Verändern das Aussehen des Glanzlichtes (Highlight)
5.

CMaterial: Die Materialklasse in Vektoria

Die Transparenzmethoden der Klasse CMaterial:

1. void SetTransparency(float frTransparancy);

   Setzt die Durchsichtigkeit, z.B. wenn
   frTransparency = 0.0 => Opak,
   frTransparency = 0.5 => Halbdurchsichtig
   frTransparency = 1.0 => voll durchsichtig

2. void SetTransparencyOn();
3. void SetTransparencyOff();

   Schaltet Transparenz dediziert an und aus (wichtig für Sortierung)

Übung zu Materialien

1

Erzeugen Sie drei Wolkenschichten mit Hilfe von Alpha-Texturen!

2

Freiwillige Zusatzaufgabe für die schnellen Nerds:

- Lassen Sie die Wolkenschichten gegeneinander verschieben!

3

4

5

Ausgabe Alphatexturierung

Ausgabe Alphatexturierung

CMaterial - Modifikationsmethoden

Weitere Modifikationsmethoden der Klasse CMaterial:

1. void SetColorAmbient(CColor color);
2. void SetBumpStrength(float fBumpStrength);
3. void SetSpecularSharpness(float fSpecularSharpness);
4. void SetDiffuseSharpness(float fDiffuseSharpness);

Verändern das Aussehen des Glanzlichtes (Highlight)

CMaterial - Modifikationsmethoden

Weitere Modifikationsmethoden der Klasse CMaterial:

1. void SetColorAmbient(CColor color);

   Setzt die ambiente Farbe (virtuelle Farbe des Hintergrunds)

2.

3. void SetBumpStrength(float fBumpStrength);

4.

5. Setzt die Stärke der Bump Map-Textur (Default = 1.0). Auch negative Werte sind möglich, Berge werden dann zu Tälern und umgekehrt.

CMaterial - Modifikationsmethoden

Weitere Modifikationsmethoden der Klasse CMaterial:

```
void SetSpecularSharpness(float
        fSpecularSharpness);
void SetDiffuseSharpness(float fDiffuseSharpness);
```

Setzt die Schärfe (Fokussierung und „Kleinheit“) des Glanzlichtes und der diffusen Schattierung.

CMaterial – Animierte Materialien

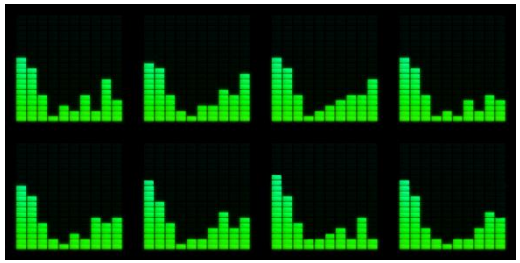
SetAni(ixPics, iyPics, fFps);

Erzeugt animierte Materialien,
dabei bedeuten:

- ixPics: Anzahl der Unterbilderspalten
- iyPics: Anzahl der Unterbilderzeilen
- fFps: Bildwiederholrate

ixPics = 4

iyPics = 2



CMaterial – Bot-Materialien

SetBot(ixPics, iyPics);
SetPic(ixPix, iyPic);

SetBot erzeugt ein steuerbares Material (= Bot-Material), dabei bedeuten:

- ixPics: Anzahl der Unterbilderspalten
- iyPics: Anzahl der Unterbilderzeilen

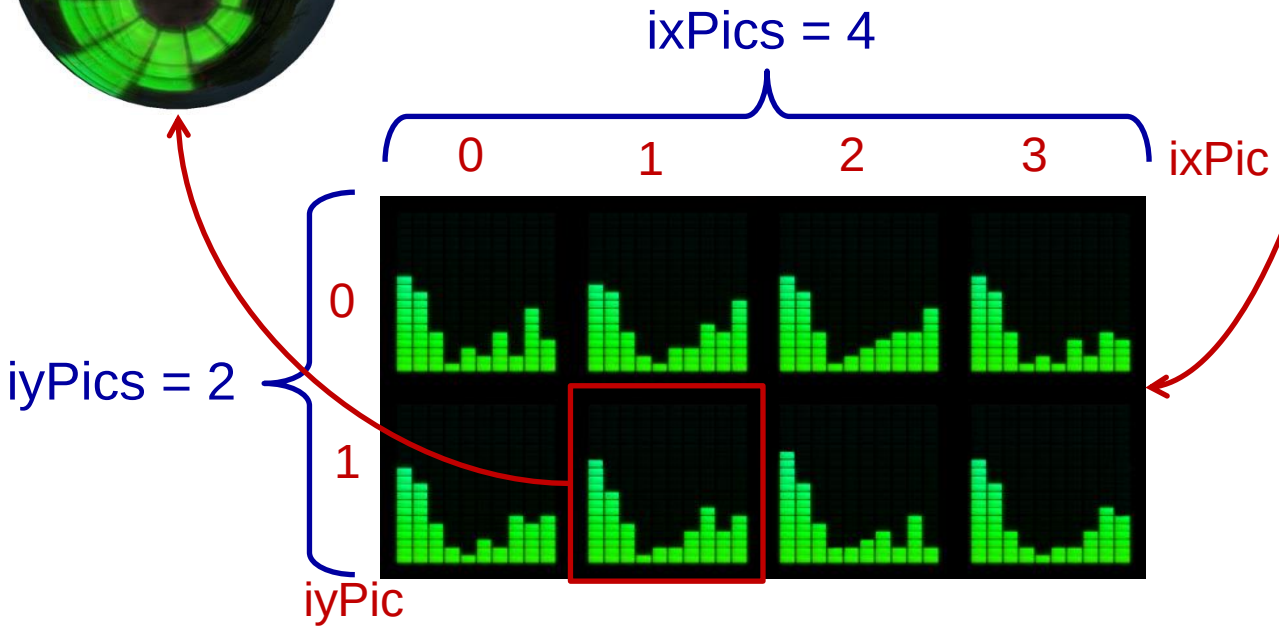
Mit SetPic wird dann das Unterbild eines Bot-Materials ausgewählt werden, dabei bedeuten:

- ixPic: Spaltennummer
- iyPic: Zeilennummer

Achtung, die Spalten- und Zeilennummern beginnen mit Null!

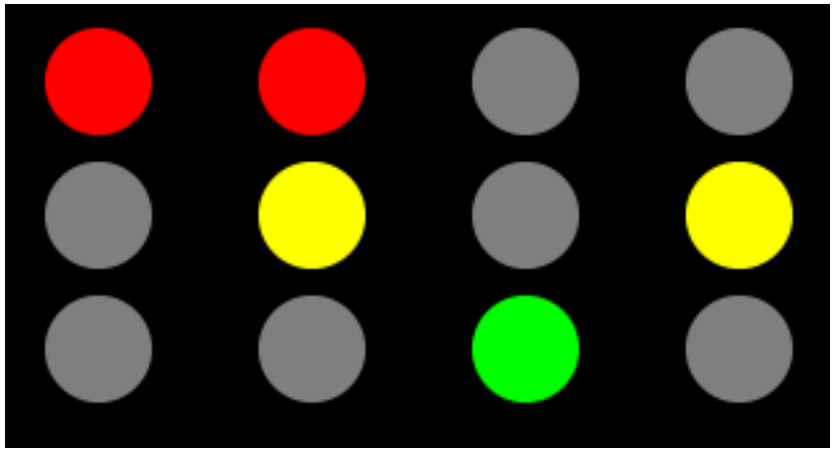
CMaterial – Bot-Materialien (Beispiel)

```
m_zm.MakeTextureDiffuse
    (“textures\\FrequencyMeter.jpg);
…
m_zm.SetBot(4, 2);
m_zm.SetPic(1, 1);
```



Übung zu Bot-Materialien

Erzeugen Sie mit Hilfe eines Bot-Materials und drei Overlays eine mitteleuropäische Ampelsteuerung!



1 2 3 4

Freiwillige Zusatzaufgabe für die schnellen Nerds:

- Erzeugen Sie länderspezifische Varianten!

GAME OVER

Danke für Ihr Interesse!