

# PBR - Texturen erstellen

Dieses deutsche Tutorial erklärt ausführlich die Erstellung von Texturen mit der neuen PBR Technik

Heute werden wir uns mit der Erstellung von PBR Texturen beschäftigen. Dieses Tutorial bezieht sich zwar auf [Bodentexturen](#), aber die Texturtechnik ist auch auf Meshes dieselbe. Wir werden uns mit denen jedoch in einem anderen Tutorial genauer auseinandersetzen.

**Schritt 0: Material** Wie in der Chemie kommen wir natürlich nicht ohne Werkzeuge aus. Folgende Programme werden für das Tutorial benötigt.

-> TrainzDev oder TRS19 (TRS19 ist im Moment closed beta unter NDA und zeigen von Screenshots aus dem Spiel nicht erwünscht. Dieses Tutorial wurde mit TrainzDev erstellt)

-> [Gimp](#) (Für das Tutorial wurde Version 2.10.2 verwendet)

-> [AwesomeBump](#)

## Schritt 1: Herstellen der Albedo

Die Albedo-Datei entspricht der alten Diffusemap. Oder wer weniger Wert auf Fachbegriffe aus der Welt des Objekt Designs legt: Die Albedo-Datei beinhaltet die grundsätzliche Textur für das Objekt, oder im Fall dieses Tutorials für eine Bodentextur



Sie bildet den Ausgangsstoff für unsere Arbeit und kann in der Bildbearbeitungssoftware unserer Wahl erstellt werden. Ich empfehle, die Albedo als .png Datei abzuspeichern, da Trainz dieses Format seit T:ANE unterstützt, welches deutlich kleiner als TGA ist.

Je nach dem, wie klein die Details auf der Albedo sind, empfiehlt es sich, diese entweder im Format 2048 x 2048 oder in 1024 x 1024 zu erstellen. Da man aber nachher immer noch verkleinern kann, rate ich dazu, während aller Arbeitsschritte erstmal 2048 x 2048 zu verwenden.

Um die Albedo in Trainz zu verwenden, braucht die texture.txt lediglich folgenden Inhalt

Code

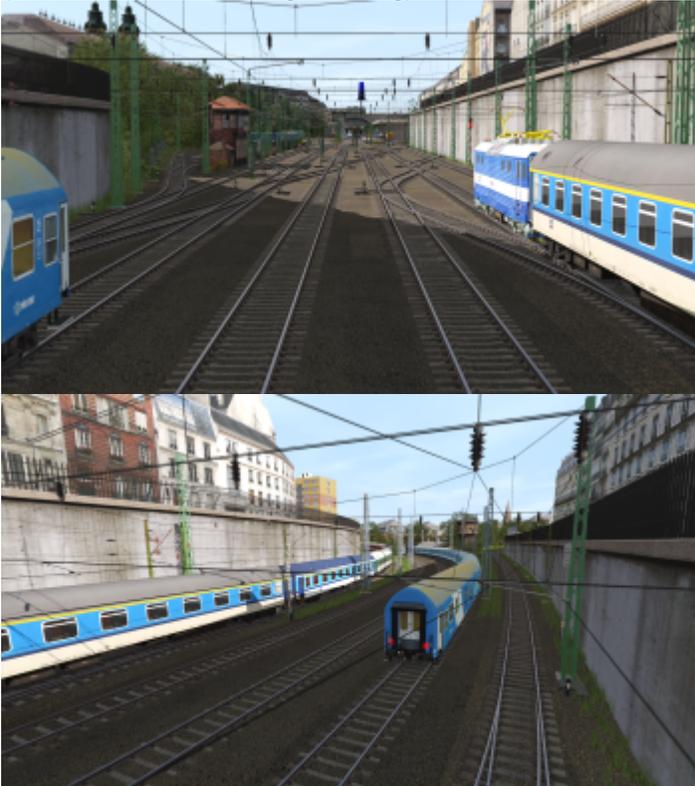
```
Primary=albedo.png  
Tile=st
```

## Schritt 2: Herstellen der Maps in Awesomebump

Nun kommt Awesomebump ins Spiel. Bevor wir jedoch überhaupt erst loslegen können, müssen wir erst einige Einstellungen treffen:

1. Die Albedo Datei per drag and drop auf die Diskette links oben ziehen. Sie sollte dann im Fenster erscheinen.
2. Unter Diffuse to Others "enable preview" aktivieren
3. unten auf Convert klicken und etwas warten
4. 3D Settings öffnen
5. DOF Deaktivieren
6. Bump mapping auswählen
7. Plane auswählen

(Bilder anklicken für Vergrößerung)



Nachdem diese Einstellungen getroffen sind, können wir noch einmal die Auswertung von AB im Mittelfenster betrachten, ob uns diese gefällt. Es gibt die Möglichkeit, noch viel Feintuning vorzunehmen, aber im Regelfall erkennt Awesomebump vieles bereits jetzt korrekt.

Nun möchten wir also die erstellten Maps exportieren

8. Normalmap

9. Heightmap

10. AO Map

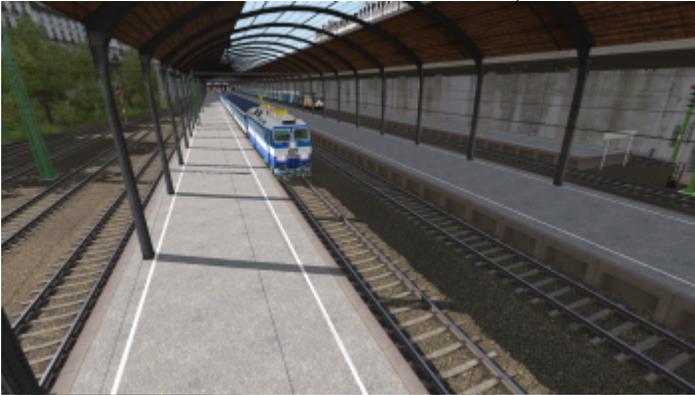
Diese speichern wir mit der zweiten rechten Diskette oben Links in einen Ordner. Die Normalmap und die Heightmap brauchen wir gleich wieder, die AO Map bewahren wir erstmal für später auf.

### **Schritt 3: Anpassen und Zusammenfügen von Height und Normalmap**

**Achtung! Für diesen Schritt ist die Benutzung von Gimp erforderlich!** Es hat sich gezeigt, dass Photoshop an dieser Stelle Fehler erzeugt, die dazu führen, dass die Heightmap nicht korrekt dargestellt wird!

Die Normalmap, die wir gerade erstellt haben, öffnen wir nun in Gimp.

Nach einem Rechtsklick auf das Bild in Gimp wählen wir Zerlegen und den Modus RGBA aus



Nach einem Klick auf OK erscheinen alle Kanäle rechts in der Ebenenauswahl. Für uns ist nur der unterste interessant, der Alphakanal.

Alle anderen können wir ausblenden, indem wir auf das Auge klicken. Die Alpha sollte komplett weiss sein.

Nun öffnen wir die Heightmap, die wir gerade erstellt haben in Gimp. Sie muss erstmal gedämpft werden.

Dazu nutzen wir das "Werte" Tool im Farbenmenü. Wir werden feststellen, dass dieses einen großen Berg anzeigt



Unten im Bereich der Zielwerte schieben wir die beiden Regler rechts und links auf die Mitte zu, sodass zwischen ihnen kein großer Abstand mehr herrscht. Dieser Abstand ist abhängig von der Größe der darzustellenden Objekte auf unserer Textur. Als Referenz: Für Gleisschotter sollte 120 als unterer und 140 als oberer Wert gewählt werden.

Wenn das Wertetool angewandt wurde, können wir durch erneutes öffnen des Wertefensters nochmal überprüfen, ob statt dem großen Berg nun nur noch eine kleine Wand in der Mitte angezeigt wird.

Ist dies der Fall, kann die Heightmap in Trainz verwendet werden. Sie kann einfach durch Kopieren und Einfügen in die Alphaebene der Normalmap, die wir vorher Zerlegt haben, eingefügt werden.

Nachdem dies getan ist, wieder ein Rechtsklick auf das Bild, Farben -> Komponenten, aber diesmal ist "Zusammensetzen" auszuwählen. Dabei darauf achten, dass der Farbmodus RGBA ausgewählt ist.

Nun ist die Normalmap auch schon fertig. Sie kann ebenfalls als png abgespeichert werden.

Damit sie in Trainz funktioniert, muss die texture.txt folgendermaßen aussehen

Code

```
Primary=normal.png  
Alpha=normal.png  
Tile=st  
NormalMapHint=normalmap
```

#### Schritt 4: Parameters Datei

So, nun fehlt uns nur noch die Parameters Datei. Dazu erstellen wir in Gimp erstmal eine neue Datei. Danach Ebene -> Transparenz -> Alphakanal Hinzufügen.

Nachdem dieser hinzugefügt ist, können wir das Bild wieder wie schon in Schritt 3 zerlegen.

Die Ebenen haben jetzt folgende Funktionen:

**Rot:** Selbstleuchten. Lässt sich für Texturen auf Objekten benutzen, die bei Dunkelheit selber leuchten sollen. Je heller eine Stelle, desto größer der Effekt. Für die meisten Anwendungsfälle muss die Rote Ebene komplett schwarz gelassen werden.

**Grün:** Matt/Glänzend. Dieser Effekt ist hauptsächlich für Metallerne und feuchte Oberflächen nützlich. Je dunkler eine Stelle, desto glänzender. Für die meisten Anwendungsfälle wird diese Ebene komplett weiss gelassen.

**Blau:** Ambient Occlusion. Hier wird die AO Map hineinkopiert, die wir zuvor mit Awesomebump erstellt haben.

**Alpha:** Metalleffekt. Gibt an, wie metallern eine Oberfläche ist. Ist die Textur Metallfrei, einfach komplett weiss lassen. (Schwarze Stellen markieren Metall)

Nachdem die AO Map eingefügt ist (und wir, wenn gewünscht, andere Einstellungen getroffen haben) das Bild wieder Zusammensetzen (siehe Schritt 3) und exportieren.

Für die texture.txt genügt folgender Inhalt

Code

```
Primary=parameters.png  
Tile=st
```

Eine passende Config kann zum Beispiel so aussehen:

Code

```
kuid
username
kind
trainz-build
texture
normal-texture
```

```
texture-variants
{
  0
  {

  }

  2
  {

  }

  3
  {

  }
}
```

param

param

param

```
season-selector
{
```

above-snow

```
  branch-true
  {

  }

  branch-false
  {

  }
}
```

```
category-class
category-era
category-region
description "Cracks brown ballast."
author
license
```

his texture file is in the folder "Cracks brown ballast."

```
thumbnails
{
  0
  {

  }
}
```

```
kuid-table
{
}
```

Alles anzeigen

Nun kann die Textur in Trainz importiert und dort begutachtet werden.

<https://cdn.discordapp.com/attachments/265529893343854593/452493971348914177/unknown.png>

## Schritt 5: Speicherplatzoptimierung

Nun werdet ihr sicher feststellen, dass alle Texturen zusammen ganz schön groß sind. Dies trifft nur auf [Bodentexturen](#) zu, bei Meshtexturen ist das ja vom Mapping abhängig, aber um kleine Texturen auf [Bodentexturen](#) auch in der korrekten Größe darzustellen, müssen sie stark geschrumpft und häufig nebeneinander auf der Textur angeordnet werden.

Dann, und nur dann, sehen diese auch sehr pixelig aus, wenn man eine 1024 x 1024 Albedo verwendet. Sind die Details größer ist eine Textur in der Größe völlig ausreichend, aber für Schotter z.B. empfiehlt sich das Maximum. Die dazugehörige Normal und Heightmap muss jedoch keine so große Auflösung besitzen, da sich die Details nicht wirklich in Geländeunterschieden äußern. Somit reicht eine 1024 x 1024 so gut wie immer aus. Wer weiter sparen will, wenn er in der Parameters Datei nur Gebrauch von der Ambient Occlusion gemacht hat und keine exakten Metall und Glanzwerte benötigt, kann diese sogar auf 512 x 512 reduzieren. Als Endergebnis sollte sich die Bodentextur irgendwo um die 10mb herum ansiedeln.

Viel Spaß mit den neuen Techniken. Die in diesem Tutorial erstellte Bodentextur ist unten angehängt und im Spiel unter "PBR Ballast 2" zu finden.

Greets, Mika