

# Grafik – how to create - für Rocks'n Diamonds

Mit diesem kleinen Tutorial möchte ich euch nun mal kurz an einem Beispiel beschreiben, wie ich meine Grafiken für das Spiel **'Rocks'n Diamonds' von Holger Schemel** produziert habe:

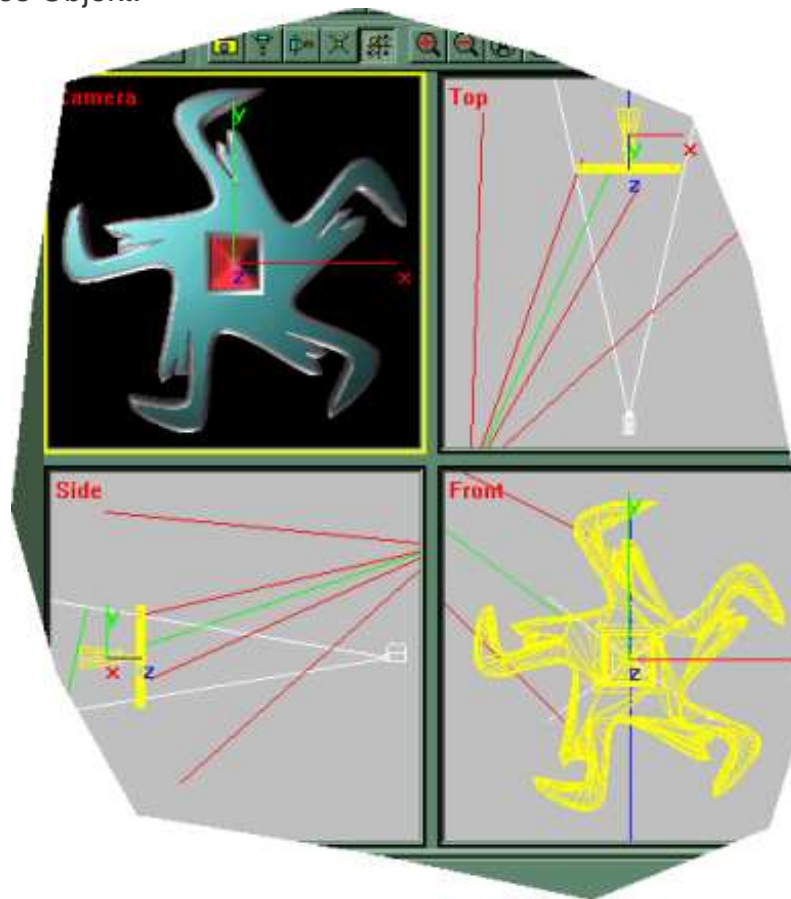
Zunächst meine Software-Voraussetzungen:

Ein einigermaßen gutes 3D Programm (ich benutze Simply 3D von Micrografx), eine Grafiksoftware (mein Favorit ,Paint Shop Pro') und ein einfache Texteditor.

Ich möchte nun mal kurz beschreiben, wie ich z.B. den Ersatz für das Dark Yam Yam (in meinem Grafikset als Destroyer bezeichnet) kreierte habe.

Am Anfang steht (wie überall) eine Idee. Ich wollte diesem Element die zerstörerischen Fähigkeiten grafisch zukommen lassen. Meine Idee war eine Säge zu schaffen.

Zunächst baute ich mir mit Simply 3D (in diesem Tutorial als S3D bezeichnet), ein entsprechendes Objekt:



Nun habe ich diesem Objekt eine Rotation zugewiesen und das ganze als .avi-Datei abgespeichert.

Nun folgt die Zerlegung der Animationsdatei in die einzelnen Bilder:



wichtig ist hierbei ein schwarzer Hintergrund.

Außerdem müssen die einzelnen Animations-Bilder 32x32 Pixel groß sein.

Dies kann man mit der resize-Funktion in dem entsprechenden Grafikprogramm erreichen.

Ich nutze hierfür PSP, da dort auch ein sog. Animation-Shop enthalten ist, der solche Funktionen sehr vereinfacht.

Nun werden die einzelnen 32x32 Pixel großen Grafiken der Reihe nach per Drag&Drop in eine beliebige Grafikdatei kopiert (siehe Bild oben).

Diese Grafikdatei wird nun als xyz.... pcx Grafikdatei abgespeichert. Wichtig ist das pcx-Format !

Nun sollte man sich die Anzahl der Animationsbilder (=Frames) notieren.

Dann geht es an die ,graphicsinfo.conf':

Am besten öffnet man die Datei aus dem Ordner ,classic graphics' oder aus dem Ordner Levels\Examples\example\_artwork\graphics.

Hier sucht man sich den Eintrag, der auf die Grafikdateien des dark\_yamyam verweist:

```
.....  
.....  
# black_orb.ypos: 9  
# black_orb.frames: 1  
# speed_pill: RocksElements.pcx  
# speed_pill.xpos: 14  
# speed_pill.ypos: 9  
# speed_pill.frames: 1  
# dark_yamyam: RocksElements.pcx  
# dark_yamyam.xpos: 8  
# dark_yamyam.ypos: 11  
# dark_yamyam.frames: 4  
# dark_yamyam.anim_mode: pingpong2  
# dynabomb.active: RocksElements.pcx  
# dynabomb.active.xpos: 12  
# dynabomb.active.ypos: 11  
# dynabomb.active.frames: 4  
# dynabomb.active.delay: 6  
.....  
.....
```

Nun müssen diese Einträge umgeschrieben werden. Zunächst sollte man das #-Zeichen entfernen, da dieses Zeichen dem Programm sagt, das diese Zeile nicht gelesen werden soll – ähnlich einem Rem-Befehl.

Zur Konfiguration:

(der in Klammern geschriebene Text dient nur zum besseren Verständnis und darf nicht mitgeschrieben werden !)

Als erstes muss die Grafik beschrieben werden:

**dark\_yamyam: xyz.pcx**

(hier den Namen der Bilddatei angeben)

**dark\_yamyam.xpos: 0**

(hier die Lage des 1. Frames der Bilderreihenfolge in x-Richtung [horizontal] angeben – dabei ist zu Beachten, das nicht bei 1 sondern bei 0 begonnen wird zu zählen – ebenso in der y-Richtung !)

**dark\_yamyam.ypos: 0**

(hier wird die Lage des 1. Frames in y-Richtung definiert )

**dark\_yamyam.frames: 14**

(hier wird die Anzahl der Einzelbilder definiert )

**dark\_yamyam.delay: 2**

(hier wird eine Pause zwischen den Bildern angegeben  
Angaben in 1/100 sec – hier gilt es zu probieren, damit die  
Grafik später auch flüssig erscheint.)

**dark\_yamyam.anim\_mode:** pingpong

Der anim\_mode beschreibt die Art der Darstellung: wird kein anim\_mode beschrieben, werden die Bilder einfach der Reihe nach abgespielt und nach dem letzten Bild folgt wieder das 1. Bild [ loop ]. Über das Beschreiben des anim\_mode's kann dies nun geändert werden. Dabei gibt es folgende Möglichkeiten:

**pingpong:** die Bilder werden immer hin und her abgespielt, das heißt wenn das letzte Bild erreicht ist läuft die Animation wieder Zurück zum 1. Bild.

**pingpong2:** wie pingpong, nur das das erste und letzte Bild 2mal angezeigt werden.

**linear:** die Grafik wird nur einmal abgespielt, das bedeutet wenn das letzte Bild erreicht ist bleibt die Animation stehen und das letzte Bild wird ständig gezeigt.

**reverse:** Die Animation wird rückwärts gespielt, 1. Bild ist somit das letzte Bild der Animation.

**loop:** Die Animation läuft in eine Endlosschleife, das selbe wie wenn anim\_mode nicht definiert wird.

**random:** Es werden aus der Animation zufällig Frames ausgewählt und angezeigt.

**none:** Es erfolgt keine Animation.

Zu Beachten ist auch noch, das die Befehle kombiniert werden können, dazu müssen sie durch Komma getrennt geschrieben werden; **Bsp. anim\_mode: linear,reverse**

**dark\_yamyam.name:** destroyer

(Hier kann dem neue Objekt ein Name gegeben werden, der dann im Editor erscheint.)

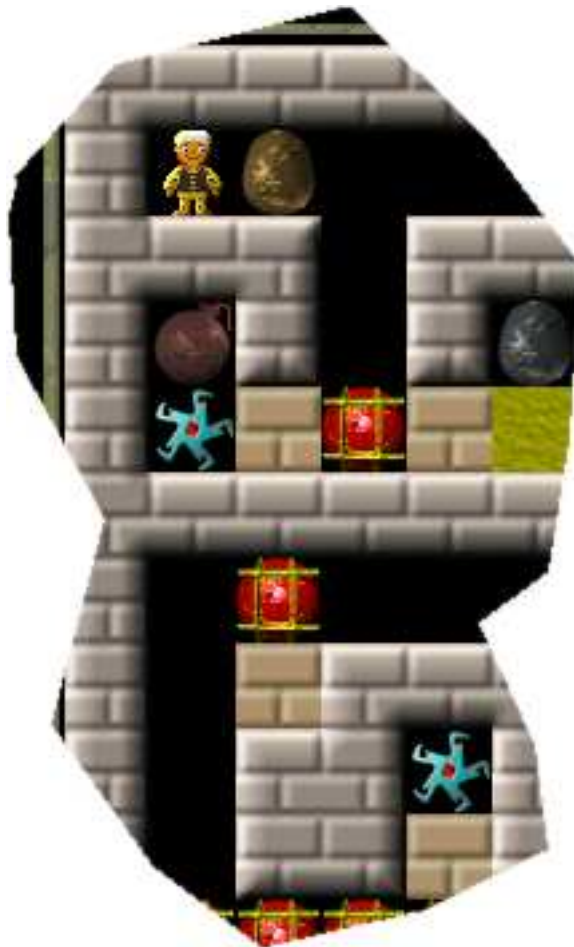
Der fertige Eintrag müsste dann in etwa so aussehen:

```
.....  
.....  
dark_yamyam:                Generic5.pcx  
dark_yamyam.xpos:           0  
dark_yamyam.ypos:           4  
dark_yamyam.frames:         32  
dark_yamyam.delay:          1  
dark_yamyam.name:           destroyer  
.....  
.....
```

Nun die fertige graphicsinfo.conf und die dazugehörige Grafik in den Ordner ...Levels\my\_level\graphics abspeichern.

Fertig.

Nun kann das ganze getestet werden und könnte dann so aussehen:



Viel Spaß und gute Ideen !

AB Graph