

Wildfire-Workshop

COLLABORATORS

	<i>TITLE :</i> Wildfire-Workshop		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 23, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Wildfire-Workshop	1
1.1	Wildfire Workshop	1
1.2	einleitung	1
1.3	workshop	2

Chapter 1

Wildfire-Workshop

1.1 Wildfire Workshop

```

#####
#                               #
#   Wildfire - an Animation-System for the Amiga                       #
#                               #
#   Copyright © 1995-1997 WK-Artworks                                   #
#   All rights reserved                                                #
#   Unauthorized Duplication Prohibited                                #
#-----#
#                               #
#   Amiga-Magazin CD Workshop                                          #
#                               #
#####

```

1. Einleitung

2. Workshop

3. Wildfire - Hauptprogramm

Documentation-Authors : Andreas Küssner, Andreas Maschke
 Last Changes : Friday 07-Mar-97
 Version : 2.82

1.2 einleitung

 Wildfire Workshop

(Der Workshop 1 ist in der Wildfire Dokumentation (in Englisch) beschrieben.)

Abspielen von Workshop 1 auf PAL

Abspielen von Workshop 1 auf PAL gezoomt

Willkommen zum Wildfire Workshop Nr. 2!

Auf der Amiga-Magazin-CD 5/6 97 sind alle benötigten Dateien enthalten, um diesen Workshop nachzuvollziehen.

Natürlich ist eine registrierte Version von Wildfire Voraussetzung, mit der Demoversion werden Sie schnell an die Grenzen Ihres Systemspeichers stoßen. Sie können jedoch auch mit der Demoversion einzelne Bilder berechnen lassen und somit die Komplexität und den Umfang des Programmes erkennen. Wenn Sie eine registrierte Version von Wildfire besitzen, genügen 6 MB Hauptspeicher und ca. 25 MB Festplattenspeicher, um den kompletten Workshop abzuarbeiten.

Natürlich sind die Ergebnisse schon auf der CD vorhanden, somit können auch unregistrierte Benutzer von Wildfire die fertigen Animationen betrachten.

Die Animationen auf der CD liegen alle im HALVE-Format vor, d.h. ca. in halber LoRes Auflösung. Sie können die Animationen natürlich auch in voller Größe berechnen, dazu müssen Sie später den Skalierungsfaktor in den Processoreinstellungen anstatt auf 0.5 auf 1 setzen.

Beim kreieren der 'Phoenix' Animationen und beim betrachten der vorberechneten Animationen wird auf das Verzeichnis 'WF:' zurückgegriffen. Mittels des Assign-scripts auf der CD können Sie das WF: Verzeichnis der CD zu Ihrem WF: Verzeichnis auf Festplatte - falls vorhanden - addieren.

Falls Sie Wildfire auf Harddisk installieren, kopiert das Installer-Script die Einzelbilder der Phoenix Animation (Directory wftut2:Phoenix/) und die fertigen Animationen (wftut2:halve/) NICHT auf die Harddisk, um Platz zu sparen.

1.3 workshop

Es wird vorausgesetzt, daß Sie Wildfire Ihren Wünschen gemäß installiert haben. (Entweder von CD gestartet oder auf HD installiert, in beiden Fällen das Assign-Script auf der CD gestartet)

Nach dem Starten sehen Sie die Benutzeroberfläche von Wildfire. Auf der Icon-Oberfläche befinden sich:

Trashcan - Löschen von erzeugten Icons (Drag & Drop)

Player - Abspielen von Animationen (Drag & Drop)

Viewer - Anzeigen von Bildern (Drag & Drop)

Editor - Editieren von Texten (Drag & Drop)

Die Einstellungen zu diesen Programmen können Sie in den Preferences vornehmen. Klicken Sie dazu auf den Button 'Preferences'

Hier können Sie folgende Einstellungen treffen:

Screen-Type: Custom/Public (eigener Bildschirm oder fremder)
 Screen-Mode
 Bildanzeige Mode
 Zeichensatz
 Zeichensatz des Icon-Managers
 Farben
 Screen, auf dem Animationen in Window-Mode abgespielt werden
 Zeit für einen Doppelklick
 Statistiken
 Expertmode (sollte Anfangs auf aus bleiben. An bedeutet keine Sicherheitsabfragen)
 BrutalBlit (Art des Blittens des Players)
 OpenDWA (DirektWindowAcces Fenster auf oder zu bei Programmstart)

Pfade:

diverse Directorys
 YAFA-Player ... Abspielprogramm für YAFA Anims
 ANIM-Player ... Abspielprogramm für IFF-Anims (wird für das Tutorial nicht benötigt) ←
 MPEG-Player ... Abspielprogramm für MPEG Anims (wird für das Tutorial nicht benötigt) ←
 Image-Processor ... z.B. AdPro (wird für das Tutorial nicht benötigt)
 Viewer ... Anzeigeprogramm für Bilder (wird für das Tutorial nicht benötigt) ←
)
 Editor ... Texteditor (wird für das Tutorial nicht benötigt)

Speichern Sie die gewählten Einstellungen ab.

Beginnen wir mit einer einfachen Animation.

1. tut_Phoenix

Eine YAFA Animation wird aus Einzelbildern generiert. Dazu öffnen wir das 'Converter'-Fenster. Als 'Input-Type' geben wir 'Pictures' an. Mittels 'Related Options' wählen wir die Bilder. Wir klicken auf 'Add Dir' und wählen die Directory der Phoenix-Bilder aus ('wftut2:Phoenix/') Nach kurzer Zeit werden 362 Bilder in der Liste angezeigt. Wir schließen die Picture-List.

Als Output Type wählen wir 'YAFA-Animation'. Mittels 'Related Options' nehmen wir folgende Einstellungen vor:

Image-Data - RAW
 Colormap - Locked (feste Palette)
 Die Palette des ersten Bildes der Animation wird benutzt.
 Alle Ausgangsbilder besitzen die selbe Palette, also kann

Colormap auf Locked gestellt werden.

Force Timecodes	- Off	(keine Timecodes)
Post-Compression	- None	(keine XPK-Komprimierung)
Delta-Compression	- WORD	(word-delta)
Uncompressed Frames	- Two	(zwei ungepackte Bilder)
UC-Mode	- Delta	(Delta Mode für die ersten Bilder)

Bitte lesen Sie die Dokumentation, um mehr über die Optionen zu erfahren.

Im 'Converter' stellen wir 'Preview' auf 'Screen', somit wird beim Konvertieren auf einem Screen die gerade generierte Animation dargestellt. Der 'Processor' bleibt ausgeschaltet.

Geben Sie als 'Output' Namen ein File auf Ihrer Festplatte an oder wählen Sie als 'Type' NIL:, um die Animationserstellung nur zu simulieren. Drücken Sie auf 'Convert'. Das Resultat liegt auf der CD schon vor.

Abspielen von tut_Phoenix auf PAL

Abspielen von tut_Phoenix auf AGA-Screen

Mittels 'Browse Session' können Sie die fertigen Einstellungen auswählen 'wf:sessions/tut_Phoenix.session' und mittels 'Load' einladen. Somit können Sie Ihre Einstellungen kontrollieren.

Einladen der Session

2. tut_PhoenixHALVE

Nehmen wir nun an, Sie möchten aus Speichergründen die Animation nur in halber Größe erzeugen. Nichts einfacher als das. Die meisten Einstellungen der Phoenix Animation bleiben erhalten, jedoch:

1. wird der Name geändert (tut_PhoenixHALVE.yafa)
2. in den Related Options der Output-Yafa-Animation stellen wir 'Colormap' auf 'unlocked' (keine feste Palette), da der Processor nun zu jedem Bild eine Palette berechnen wird.
3. weiterhin benutzen wir nun einen XPK-Packer für die Bilder, und zwar XPK-Fast
Dazu stellen wir in den 'Related Options' 'Post-Compression' auf 'XPK' und den 'Compressor' auf 'FAST' (Name des XPK-Kompressors) und den Pack-modus 'Mode' auf '100'

Weiterhin teilen wir dem Converter mit, daß er die Bilder im Chunky-Format speichern soll, denn im allgemeinen lassen sich Chunky-Bilder besser XPK-packen, als normale Amiga-RAW-Bilder.

Dazu stellen wir in den 'Related Options' 'Image-Data' auf 'Chunky'

Nun wird zwischen 'Input' und 'Output' der 'Processor' zwischengeschaltet. Die Einzelbilder werden nun zuerst an den Processor übergeben, von ihm umgerechnet und dann an den 'Output'-Kanal übergeben.

Der 'Processor' wird im Converter-Fenster auf 'Enabled' geschaltet.

Nun öffnen wir das Processor Fenster (vom Hauptfenster zu erreichen)
Folgende Einstellungen werden getroffen:

Mode: Color (Farbe)
Dither: Floyd (Dithering)
Smooth: On (Glättung)
Depth: 6 (Bit-Tiefe ... Sie bleibt Original. Um die Original Bit-Tiefe der Ausgangsbilder zu erhalten, gehen Sie in den 'Converter' in die 'Related Options' des 'Input' und Doppelklicken auf ein Bild. Nun werden die Daten des Bildes angezeigt, u.a. Depth=6)
CSkip: 0 (Ausgelassene Farben ... z.B. für Worbench-Bilder, hier 0)
Scale: 1 (Skalierungsfaktor, siehe spätere Beispiele)
3DQuant: 0 (3dQuantisierung, siehe spätere Beispiele)

Kommen wir zum eigentlichen Effekt. Klicken Sie im 'Processor' Fenster auf 'Add' und wählen Sie aus der Effekt-Liste 'Halve' (bei '2D-F/X') und doppelklicken Sie es. Im 'Processor' Fenster erscheint HALVE (in=STREAM, out=STREAM)

Doppelklicken Sie auch dies. Es öffnet sich das Einstellungs-Fenster dieses Effektes. Als Eingabe verwenden wir 'Stream', als Ausgabe auch 'Stream' (dies ist Vorgegeben)

Die Daten, die vom Converter-Input an den Prozessor geschickt werden, kommen im Stream an. Diese Daten (Einzelbilder) werden halbiert und wieder im Stream abgelegt. Der Converter-Output bekommt seine Daten wiederum aus dem Stream des Prozessors. Der Prozessor wurde somit zwischengeschaltet. Jeder Effekt des Prozessors ist als weitere Zwischenschaltung zu sehen. Wenn z.B. 2 mal Halve jeweils mit Ein- und Ausgabe Stream im Processor eingestellt wird, wird das Bild zweimal halbiert, bevor es an den Converter geschickt wird. Wir bleiben aber bei unserem einfachen Beispiel.

Außer dem 'Stream' Puffer gibt es noch die temporären Puffer 'Temp1' bis 'Temp8'. In Ihnen können Bilder zwischengespeichert werden. Dies wird bei diesem einfachen Beispiel aber nicht benötigt.

Hiermit sind alle nötigen Einstellungen getroffen. Nun kann im 'Converter' Fenster 'Convert' gedrückt werden.

Wiederum brauchen Sie zur Kontrolle nur 'wf:sessions/tut_PhoenixHALVE.session' im Converter einladen, um Ihre Einstellungen mit den Originaleinstellungen dieses Workshops zu vergleichen. Das zugehörige 'process'-File, in welchem die Processor-Einstellungen gespeichert sind, wird automatisch nachgeladen.

Abspielen von tut_PhoenixHALVE auf PAL

Abspielen von tut_PhoenixHALVE auf PAL (gezoomt)

Abspielen von tut_PhoenixHALVE auf AGA-Screen

Einladen der Session

2a. tut_Phoenix-a-HALVE

Beim Berechnen der Phoenix-Halve Animation hat der Processor bei jedem Bild eine neue, passende Palette berechnet. Da alle Ausgangsbilder die selben Farben besitzen, können wir die Animation auch mit fester Palette generieren. Hierzu schalten wir in den 'Related Options' des 'Converter' die 'Colormap' auf 'Locked'.

Weiterhin schalten wir 'Image-Data' 'Chunky' wieder auf 'RAW' um und den XPK- Packer (Compressor) auf 'Off', da wir diese Animation nun als Delta-Animation generieren wollen.

Bei fester Palette und sich nur wenig ändernden Bildern ist Delta meistens Effektiver als XPK-Packen der Einzelbilder.

Dazu wird 'Delta-Compression' auf 'Word-BR' geschaltet, dies bedeutet 'Word-Delta' mit 'ByteRun'

Außerdem:

Uncompressed Frames- One (nur das erste Bild ungepackt)
UC-Mode - Delta (Delta Mode auch für die zwei ersten Bilder)

Nun müssen wir dem Processor mitteilen, daß:

1. Nur eine Palette verwendet werden soll, und
2. welche Palette er zum Rendern der Animation benutzen soll.

Im 'Processor' Hauptfenster klicken wir auf 'Palette'.

Als 'File' wählen wir eines der Bilder unserer Animation aus (z.B. 'wftut2:Phoenix/Tut-pic.0001') und klicken auf Load. Nun wurde die Palette des Bildes geladen. Die Farb-Buttons des Palette-requesters werden sich nur verändern, falls Wildfire auf 256 Farben läuft. Dies ist nicht zu empfehlen. Nun teilen wir dem Processor mit, daß er diese Palette benutzen soll. Dazu schalten wir 'State' auf 'Enabled'. Beenden Sie nun den Palette-Requester.

Im Processor Fenster stellen wir 'Dither' auf aus, da wir bei dieser Animation kein Dithering benutzen wollen, damit Sie sich besser Delta-packen läßt.

Hiermit sind alle nötigen Einstellungen getroffen. Nun kann im 'Converter' Fenster 'Convert' gedrückt werden.

Wiederum brauchen Sie zur Kontrolle nur 'wf:sessions/tut_Phoenix-a-HALVE.session' im Converter einladen, um Ihre Einstellungen mit den Originaleinstellungen dieses Workshops zu vergleichen. Das zugehörige 'process'-File, in welchem die Processor-Einstellungen gespeichert sind, wird automatisch nachgeladen.

Abspielen von tut_Phoenix-a-HALVE auf PAL

Abspielen von tut_Phoenix-a-HALVE auf PAL (gezoomt)

Abspielen von tut_Phoenix-a-HALVE auf AGA-Screen

Einladen der Session

3. tut_waterHALVE

Nach diesen einfachen Animationen kommen wir nun zu speziellen Wildfire Anwendungen! Wenn Sie die folgenden Beispiele richtig erfaßt haben, werden Sie in der Lage sein, selbst komplexeste Animationen zu erzeugen. Wir werden die folgenden Animationen jeweils aus einem oder zwei Bildern erstellen, und Spezialeffekte auf diese Bilder anwenden. Die erste Animation soll 180 Bilder lang sein. Die Bilder werden nicht aus dem Input des Converters kommen, sondern im Processor selbst erzeugt. Das einzigste, was wir dem Converter mitteilen müssen, ist wieviele Bilder er erzeugen muß. Als eingabe verwenden wir 'BlackFrames', dies liefert schwarze Bilder. Als 'Related Options' wählen wir:
 'Frames' 180 (Anzahl der Bilder)
 'Width' 80, 'Height' 60 (Größe wird später durch Processor bestimmt, also kann jede beliebige Größe im Converter angegeben werden. Recht kleine Bilder benötigen nicht unnötig viel Speicher und sind deshalb zu empfehlen.

Der 'Preview' bleibt für die folgenden Animationen auf 'Screen' geschaltet. Der 'Processor' bleibt 'Enabled'.

Als Output-Name wird 'tut_waterHALVE.yafa' verwendet und die Optionen lauten wie folgt:

```
'Image-Data' 'Chunky'
'Colormap' 'Unlocked'
'Force Timecodes' 'off'
'Post-Compression' 'XPK'
'Compressor' 'FAST'
'Mode' '100'
'Delta-Compression' 'None'
```

Kommen wir nun zum Processor Fenster.

```
'Mode' 'Color'
'Depth' '6'
'Scale' '0.5' (Skalierungsfaktor 0.5, wenn Sie die Animationen in einer
Auflösung von 320x256 Pixeln rechnen wollen, setzen Sie den
Faktor auf 1. Ist er zwei, wird die Größe verdoppelt, bei
0.5 - wie wir es benutzen - wird das Bild halbiert. Keine
der Daten muß in den Scripts verändert werden, alle
Koordinaten können sich auf die Original-Größe der Bilder
beziehen.)
'Dither' 'Floyd'
'CSkip' '0'
'3DQuant' '0' (höchste 3D Genauigkeit, je größer 3DQuant, desto ungenauer
sind die 3D-Effekte. Es wird dann anstatt mit Punkten mit
Flächen gerechnet.)
'Smooth.' 'On'
```

Setzen Sie weiterhin den Status der 'Palette' aus aus, also:

```
'Palette' ---> 'State' 'Disabled', denn der Processor soll für jedes Bild
eine eigene Palette berechnen.
```

Kommen wir nun zur Bedeutung von:

'Folding' Scripts können 'weggefaltet' werden, wie aus Texteditoren bekannt. Ist zum Beispiel der Teil eines Scriptes in Ordnung und soll nicht mehr verändert werden, können die jeweiligen Zeilen von 'Visible' auf 'Hidden' geschaltet werden, und bei aktiviertem 'Folding' werden Sie dann

versteckt. Alle Processor-Scripte, die auf der CD vorliegen, folgen diesem Schema. Nur die Kommentarzeilen sind zu sehen. Wenn Sie die Scripte ansehen wollen, schalten Sie Folding auf 'Disabled'.

Wir wollen ein Bild einladen, es mit einem Wassereffekt Deformieren und mit der Balance Funktion ein wenig die Farben variieren.

Dazu drücken wir im Prozessor-Fenster auf 'Add' und wählen aus der 'Buffers'-Liste 'LoadImage' (mit einem Doppelklick) aus. Als nächstes 'Add' und aus der '3D-F/X'-Liste 'Water' und wiederum 'Add' (Color) 'Balancing'. In der Weiterführung des Workshops wird dies nichtmehr ausführlich erklärt, sondern (z.B. für diesen Effekt) nur noch folgendes geschrieben:

```
'Add' (Buffers) 'LoadImage'
'Add' (3D-F/X) 'Water'
'Add' (Color) 'Balancing'
```

Nun doppelklicken wir 'Loadimage' in der Processor-Liste. Es erscheint das Options-Fenster der 'Loadimage'-Funktion. Wir wollen das zu benutzende Bild EINMAL in einen temporären Speicher laden, und es dort behalten. Wir benutzen den Temporären Speicherplatz 1. Die anderen Effekte werden nur aus diesem Puffer laden, ihn jedoch nicht verändern. Somit reicht es aus, das Bild EINMALIG in diesen Puffer zu laden. Als 'Sequence' wählen wir 'wf:Trash/face1' und geben als 'Loop' 1 an und als 'Shift' 0. Dies bedeutet, das die Sequence 'face#?' bis zu Bild 1 geladen wird, und dann wieder von vorne beginnt. (Wenn Loop = 3 wäre, würde WF beim ersten Bild face1 laden, dann face2, dann face3 und wiederum face1). Shift gibt den Wert an, um den der Zähler verändert wird. Shift=2 bedeutet, das jedes zweite Bild geladen würde. Wir beschränken uns hierbei aber nur auf 1 einziges Bild. Mittels show können wir es uns anzeigen lassen. Wir beenden die 'LoadImage-Options' und geben in der 'Range' für diesen Effekt (Loadimage muß mit einem Balken dabei unterlegt - also selektiert - sein) 1 1 1 an. Die erste Zahl gibt das erste Anim-Bild an, bei dem der Effekt ausgeführt wird (also sofort bei Bild 1), die zweite Zahl gibt das letzte Bild an, bei dem der Effekt ausgeführt wird (also auch das erste Bild, der Effekt wird nur einmal ausgeführt, da das Bild nur einmal in den Speicher geladen wird) und die dritte Zahl gibt den Skip-Wert an, also wieviele Bilder der Animation weitergezählt werden. Ist der Wert 1, wird (im Rahmen der Range) auf jedes Bild der Anim der Effekt angewandt.

Für den 'Water' Effekt lassen wir die Voreinstellungen bestehen (1 9999 1), d.H. er wird auf jedes Bild der Anim angewandt. Ebenso für den 'BALANCING' Effekt.

Nun doppelklicken wir auf den 'Water' - Effekt. Weiterhin klicken wir auf 'Single Img.' Es sollten nun zwei neue Fenster offen sein,

1. 'Water-Options'
2. 'Single-Picture-Process'

Der 'Single-Picture-Process' simuliert ein Einzelbild der Animation, er kann zur Einzelbild-Anwendung benutzt werden. Wir wählen als 'Input' 'wf:Trash/face1' und klicken auf 'Apply Dimensions'. Dies hat zur Folge, daß die Bildgrößen (Breite, Höhe, Mittelpunkt etc.) in den selektierten Effekt (hier 'Water-Options') übernommen werden. Nun kann 'Single-Picture-Process' wieder geschlossen werden.

Kommen wir zu den 'Water-Options':

Als 'Input' das Bild, das im Temp1-Puffer liegt, also 'Temp1'
 Als Output der 'Stream'

Folgende Werte werden benutzt:

Zoom = 1 (wird später durch eine Variable ersetzt)
 CentreX = 160 (durch apply dimensions schon voreingestellt!) (X-Koordinate des Zentrums)
 CentreY = 128 (durch apply dimensions schon voreingestellt!) (Y-Koordinate des Zentrums)
 CamX = 0, CamY = 0, CamZ = -800 (Koordinaten der Kamera. Hierbei ist darauf zu achten, dass die Z-Koordinate immer außerhalb der Bilddimensionen liegen sollte, siehe WF-Docs)

Wir wollen das Bild um die X und Y-Achse rotieren, also:

Rotate XY, Alpha und Beta werden später durch Variablen ersetzt.

Wir schalten das Licht an und wählen 'Double' als Faces. (Flächennormalen auf beiden Seiten -> d.h. von beiden Seiten erscheint das Objekt hell)
 LightX = 50, LightY = 400, LightZ = -600 (Koordinaten der Lichtquelle)
 Intens. = 0.6 (Intensität)
 Ambient = 0.4 (Intensität des Umgebungs-Lichtes)
 Hardness = 0.96 (Härtegrad des Objektes)

Die Welle soll RADIAL entstehen, also wählen wir Axis XY
 Frame - derzeitiges Position der Welle - wird später ersetzt,
 Frames = 60 (ein voller Zyklus), Times = 8 (Anzahl der Ringe), Rings = 3 (so oft entstehen die Ringe), Amplitude - die Amplitude der Welle - wird später ersetzt, Wavel. = 40 - die Wellenlänge, OriginX=160, OriginY=128 das Wellenzentrum.
 MoveXY = On (Deformationsart wird ausgewählt -> On bedeutet sehr realitätsnahe Berechnung, jedoch können kleine Fehler auftreten)
 Damping = off (Die Wellen-Dämpfung wird ausgeschaltet.)

Nun können wir das 'Water-Options' Fenster schließen. Wir öffnen 'Balancing-Options' und wählen als 'Input' 'Stream', dies bedeutet das Bild, was vom Wasser-Effekt berechnet wird und als 'Output' auch 'Stream', dieses Bild wird also an den Processor übergeben.
 Red, Green und Blue (die Farben) werden später durch Variablen ersetzt. Wir schließen 'Balancing-Options'.

Wir öffnen 'Variables' und fügen folgende Variablen ein:

```
'Add' 'alpha'      - der Alpha-Winkel
'Add' 'beta'       - der Beta-Winkel
'Add' 'frame'      - der Zähler für den Wasser-Effekt
'Add' 'zoom'       - der Zoomwert
'Add' 'red'        - die Rote Farbkomponente
'Add' 'blue'       - die Blaue Farbkomponente
'Add' 'amplitude' - die Amplitude
```

Nun editieren wir die Variablen und bilden sogenannte Envelopes - d.h. zeitveränderliche Kurven. Dazu Doppelklicken wir jeweils auf die Variablen:

Alpha:

Wir stellen den 'Type' auf 'Lineat' - also Linearer Kurvenverlauf.
 Wir klicken mit der Mouse auf den ersten Punkt, der zu sehen ist. (es ist nur ein einziger zu sehen) und geben als X und Y jeweils 1 ein. Die

X-Koordinate ist immer die Bild-Nummer der Animation. D.h. beim berechnen von Bild 1 der Animation liefert diese Variable den Wert 1. Nun drücken wir auf 'Add Pnt' und klicken rechts neben den ersten Punkt. Wir selektieren den zweiten Punkt und geben als X 49 und als Y 76 ein. Dies bedeutet, bei Bild 49 gibt die Variable den Wert 76 aus. Zwischen Punkt 1 und zwei wird eine Linie gezogen, die den Werteverlauf der Variable kennzeichnet. Ebenso verfahren wir für die restlichen Punkte. Wir generieren noch:

Punkt 3: X=60, Y=30

Punkt 4: X=120, Y=30

Punkt 5: X=180, Y=360 (Wir erinnern uns, die Animation hat 180 Bilder, somit sollten wir für alle 180 Bilder einen Wert vergeben.)

Wir schließen das Envelope-Fenster und gehen genauso für alle anderen Variablen vor:

beta:

Typ: Linear

Punkt 1: X= 1 Y= 1

Punkt 2: X= 32 Y=-87

Punkt 3: X= 60 Y= 60

Punkt 4: X=120 Y= 60

Punkt 5: X=180 Y=360

frame:

Typ: Linear

Punkt 1: X= 0 Y= 1

Punkt 2: X= 61 Y= 1

Punkt 3: X=120 Y= 60

Dies ist der Zähler für die Wasserwelle. Von Bild 1 bis 60 bleiben wir beim Wasser-Bild 1, d.H. keine Welle ist zu sehen. von Bild 61 bis zum Bild 120 der Animation durchläuft die Wasserwelle einen kompletten Zyklus und danach (keine weiteren Punkte vorhanden!) wird der Wert der Variablen automatisch auf Null gesetzt.

zoom:

Typ: Linear

Punkt 1: X= 0 Y= 1

Punkt 2: X=120 Y= 1

Punkt 3: X=140 Y= 0,6

Punkt 4: X=155 Y= 1,4

Punkt 5: X=180 Y= 1

Der Zoomfaktor für den Wassereffekt. Bis zu Bild 120 der Animation ist der Faktor 1, danach wird er variiert.

red:

Typ: Linear

Punkt 1: X= 0 Y= 0

Punkt 2: X= 20 Y= 0

Punkt 3: X= 60 Y=-20
 Punkt 4: X=120 Y=-20
 Punkt 5: X=140 Y= 0

Die Veränderung des Rotwertes beim Balancing. Ein negativer Wert bedeutet, daß diese Komponente abgeschwächt wird.

blu:

Typ: Linear

Punkt 1: X= 0 Y= 0
 Punkt 2: X= 20 Y= 0
 Punkt 3: X= 60 Y= 40
 Punkt 4: X=120 Y= 40
 Punkt 5: X=180 Y= 0

Die Veränderung des Blauwertes beim Balancing. Ein positiver Wert bedeutet, daß diese Komponente verstärkt wird.

amplitude:

Typ: Linear

Punkt 1: X= 59 Y= 0
 Punkt 2: X= 60 Y= 3
 Punkt 3: X=120 Y= 3
 Punkt 4: X=121 Y= 0

Die Amplitude der Welle. Alle nicht belegten Werte (0 .. 58 und 121 .. 180) werden automatisch auf Null gesetzt.

Wir beenden das Variables Fenster und geben die Variablen in die Effekte ein.

`Balancing-Options`

`Red` ---> VAR-Knopf drücken und `red` auswählen

`Green` ---> `red` auswählen. (Für die Grüne Komponente wird dieselbe Farbveränderung wie für die Rotkomponente benutzt.)

`Blue` ---> `blue`

`Water-Options`

`Zoom` ---> `zoom`

`Alpha` ---> `alpha`

`Beta` ---> `beta`

`Frame` ---> `frame`

`Amplitude` ---> `amplitude`

Somit sind alle Variablen eingestellt und wir können die Animation konvertieren. (Im Converter auf `Convert` drücken) Da der Wasser-Effekt (ein hochkomplizierter 3D-Effekt) benutzt wird, kann die generierung der Animation einige Zeit (durchaus eine Stunde oder mehr, auf langsamen Rechnern) dauern.

Nun stellt sich die Frage, wie man auf die Einstellungen der Variablen

kommt, wenn man selber Animationen generieren möchte. Für wellen u.ä. Effekte sollte immer ein Zähler (frames!) linear hochgezählt werden, z.B. von 0 auf 60. Die meisten anderen Variablen sind Geschmackssache, so kann fast alles frei verändert werden. zum Ausprobieren eignet sich vorallem der 'Single Img.'-Mode. Dabei wird auf ein bestimmtes Bild das Processor-Script angewandt und man kann die Optionen verändern und sieht sofort (wenn der Rechner schnell genug ist!) die Veränderung. Es ist zu beachten, daß bei diesem Beispiel die Range des 'LOADIMAGE' Effektes auf '1 9999 1' zu setzen ist, da bei der Einzelbild-bearbeitung immer nur das gewählte Bild berechnet wird, und wenn man z.B. Bild 2 testen möchte, wird natürlich nichts im Temp-Puffer 1 sein, da der Temp-Puffer 1 nur bei Bild 1 der Animation gefüllt wird. Also geben Sie diese Range ein und klicken Sie auf 'Single Img.'.

Als 'MaxFrame' geben Sie '180' (Anzahl der Anim-Bilder) ein und mit dem Schieberegler können Sie den zu bearbeitenden Frame einstellen. Wählen Sie z.B. Bild 70 aus.

Wie schon früher erwähnt, werden die 3D -Effekte wesentlich schneller berechnet, wenn 3DQuant im Processor auf eine größere Zahl gesetzt wird, natürlich sind die Effekte dann ungenauer und diese Quantisierung sollte vor Berechnung der Animation wieder auf 0 gesetzt werden. Aber zur 'Vorschau' reicht meist ein Wert von 3 - 10 (je nach Rechner und Qualitätswünschen)

Im 'Single-Picture-Process' sollte dann die Breite 'Width' auf einen angemessenen Wert (z.b. zwischen 60 und 200) gesetzt werden. Dieser Wert (im Verhältnis gesetzt zum Input-Bild) bestimmt die Größe des zu berechnenden Beispielbildes. Ein nicht allzugroßer Wert ist zu empfehlen. Die Tiefe sollte auf die gewählte Tiefe des Wildfire-Screens gesetzt werden. 4 ist die Voreinstellung der Preferences. Nun wird 'Preview' auf 'Enabled' geschaltet - und nach kurzer Zeit kann man das Ergebnis erkennen. Nun kann man nach belieben die Einstellungen der Effekte ändern - nach jeder Änderung wird das Einzelbild neu berechnet - bis Preview wieder ausgeschaltet wird.

Wenn man im 'Single-Picture-Process' auf 'Save Image' klickt, wird das berechnete Einzelbild unter dem 'Output'-Namen im Format des gewählten 'Savers' (im Processor-Fenster zu erreichen) abgespeichert. Klickt man jedoch auf 'Save Rendered' werden beim Speichern auch noch die Einstellungen des Processors (Tiefe etc.) beachtet.

Wenn Sie diesen Effekt komplett nachvollzogen haben, sollten Sie in der Lage sein, eigene Effekte zu generieren. Wir empfehlen, alle Kommandos auszuprobieren und die Dokumentation zu lesen und sich die Beispielbilder zu den Effekten in der Dokumentation anzusehen. (Einfach HELP-Taste - am besten wenn Sie sich direkt im Prozessor Fenster befinden - drücken, um an die Dokumentation zu gelangen)

Abspielen von tut_waterHALVE auf PAL

Abspielen von tut_waterHALVE auf PAL (gezoomt)

Abspielen von tut_waterHALVE auf AGA-Screen

Einladen der Session

4. tut_wrapHALVE

Die Convertereinstellungen werden wie bei Beispiel 3 vorgenommen. Die Prozesseinstellungen bleiben auch gleich.

Wenn das Prozessor-Script nicht gelöscht ist (sich also Effekte im Prozessor befinden), drücken wir auf 'Clear' und beginnen von neuem.

Effekte hinzufügen:

```
'Add' 'LOADIMAGE'  
'Add' 'WRAP'
```

Variablen hinzufügen:

alpha:

Typ: Linear

```
Punkt 1: X= 0 Y= 0  
Punkt 2: X=180 Y=360
```

beta:

Typ: Linear

```
Punkt 1: X= 0 Y= 0  
Punkt 2: X=180 Y=-720
```

zoom:

Typ: Linear

```
Punkt 1: X= 0 Y= 1,2  
Punkt 2: X= 30 Y= 1,2 (kann natürlich etwas verändert werden, je nach  
Geschmack, z.B. auf 0.8 setzen etc.)
```

radius, lightX, lightY, hardness ... Lesen Sie bitte das Session-File ein und sehen Sie sich die Werte an. Dies sind wie gesagt nur Vorgabn, Sie können (sollten!) natürlich von Ihnen variiert werden. Lassen Sie Ihrer Phantasie freien Lauf.

Die Variablen werden nun in den Effekt eingefügt.

Effekte Editieren:

Edit 'LOADIMAGE'

```
'Output' Temp1  
'Sequence' 'wf:Trash/face1'  
'Loop' '1'  
'Shift' '0'
```

Edit 'Wrap'

```

`Input` Temp1
`Output` Stream
`Zoom` zoom
`CentreX` 160
`CentreY` 128
`CamX` 0
`CamY` 0
`CamZ` 0
`Rotate` XY
`Alpha` alpha
`Beta` beta
`Light` on
`Faces` Double
`LightX` lightX
`LightY` lightY
`LightZ` -600
`Intens.` 1
`Ambient` 0.24
`Hardness` hardness
`Axis` XY
`OriginX` 160
`OriginY` 128
`Method` Radius ... Bei diesem Effekt gibt es zwei Möglichkeiten, den Modus
                        Radius (der Radius des Effektes wird festgelegt) und
                        den Modus Amount (die Stärke des Effektes wird
                        festgelegt)
`Radius` radius

```

Dies sind alle Einstellungen. Wir empfehlen wiederum auch die Range des ersten Effektes (Loadimage) auf '1 9999 1' zu stellen und im 'Single Img.' Modus einzelne Werte für die verschiedenen Einstellungen auszuprobieren. Viel Erfolg!

Abspielen von tut_wrapHALVE auf PAL

Abspielen von tut_wrapHALVE auf PAL (gezoomt)

Abspielen von tut_wrapHALVE auf AGA-Screen

Einladen der Session

5. tut_logoHALVE

Nun erzeugen wir eine Animation, vor der ein Lightwave-Amiga-Logo rotiert wird!

Im Converter benutzen wir wiederum die selben Eingaben. Nur die Anzahl der Bilder wird in den 'Related Options' des 'Inputs' auf 60 gesetzt, da die Animation kürzer ist.

Bitte laden Sie die Session ein ...

Einladen der Session

... und begeben Sie sich auf den Processor-Bildschirm.

Schalten Sie 'Folding' auf 'Disabled'. Da Sie mittlerweile mit dem Hinzufügen und editieren der Effekte keine Probleme mehr haben sollten, werde ich mich auf die Erklärung der Einstellungen beschränken.

Lightwave-Objekt: (LWOB-Options)

Die Ausgabe erfolgt in den Stream. Das Objekt wird mittels 'flat shading' gerendert!

Zoom und Kamera-Einstellungen sollten je nach belieben eingestellt werden. Die Beta-Variable bewegt sich innerhalb der 60 Frames von 0 bis 360, also eine volle Umdrehung des Logos!

Lichteinstellungen etc. werden wieder je nach belieben getroffen. Vorallem mit den Intensitäten und der Härte kann experimentiert werden.

Der Text kann benutzt werden, um LW-Objekte zu katalogisieren! Dazu wird (wenn diese Funktion eingeschaltet ist) der Filename im Bild eingeblendet. Als File wird das auf der CD vorhandene 'wf:trash/amiga.lwob' verwendet. Die Breite und die Farben des Hintergrundes können natürlich variiert werden, sollten bei diesem Beispiel jedoch bei den Einstellungen bleiben. Die Textfarben und den Zeichensatz kann man natürlich nur einstellen, wenn man Text anschaltet!

Mittels dem THRESHOLD und dem CONVOLVE-Effekt wird das Alpha Bild erzeugt, welches für 'weiche Kanten' des Logos auf dem Hintergrundbild zuständig ist. Der komplette Effekt läßt sich auch ohne diese weichen Kanten erzeugen, das zugehörige Script enthält nicht die Kreierung dieser Bilder und verwendet beim letztendlichen composen keinen Alpha-Kanal. Dieses Script heißt tut_logo-a-HALVE.

Der Threshold-Effekt wandelt das gerenderte LW-Objekt in ein Schwarz-weiß Bild um und speichert es im Temp-Puffer 2. Der Convolve-Effekt nimmt diesen zweiten Puffer und 'smoothed' ihn (d.h. das Bild wird weicher an den Kanten) Das 'weiche' Bild wird wiederum im Puffer 2 abgespeichert.

Nun wird der Hintergrund mit dem Befehl CREATEIMAGE erzeugt. Als Bilddimension wird 320x256 gewählt, über die 4 Farbgadgets werden die Farben in den Ecken des Bildes gewählt, wobei ein Verlauf zwischen diesen Farben berechnet wird. (Achtung! Es werden nicht die wirklich ausgewählten Farben in den Gadgets gezeigt, sondern Bildschirmfarben! Bitte achten Sie daher auf die RGB-Werte!) Die Generierung des Hintergrundbildes geschieht nur einmal, also wird die Range dieses Effektes wieder auf '1 1 1' gesetzt. Das entstandene Bild wird in Puffer 3 gespeichert.

Der PIXELIZE Effekt greift auf den Puffer 3 zu und überschreibt ihn. Auch er wird nur einmal während der gesamten Animation aufgerufen, da mit ihm das Hintergrundbild nur ein wenig verändert wird. Das Hintergrundbild wird mit einem Raster überlagert. Das entstandene Bild wird wiederum in Temp 3 gespeichert. Während der kompletten Animation bleibt in Temp 3 dieses Bild stehen.

Der Wave3D-Effekt liest bei jedem Frame das Hintergrundbild und eine 3d-Welle wird darübergerechnet. Wiederum werden Variablen verwendet. Besonders bemerkenswert sind die Light-Variablen. (Siehe Verlauf im Envelope-Fenster der Variablen) Die Lichtquelle wird hin und her bewegt.

Dies sieht man in der Animation an der Veränderung des Lichteinfalls auf dem Wasser! Das erzeugte Hintergrundbild wird in Temp 1 gespeichert (Temp3 bleibt unberührt!)

Der Compose Effekt fügt nun den Hintergrund Temp1 (Wasser) und den Vordergrund (LW-Objekt) zusammen. Hierbei wird das generierte Alpha-Bild aus Temp 2 benutzt, damit die Kanten des Logos geglättet werden. Alle Bilder haben die gleiche Größe, also muß das Vordergrundbild nicht zentriert werden. Somit sind die Centre-Flags aus. Ein Einfarbiges Genlock wird benutzt. Es wird die Farbe 0,0,0 eingestellt. Dadurch erreicht man, daß man durch die Farbe 0,0,0 (schwarz!) des Vordergrundes hindurchsehen kann. Wir erinnern uns, daß das LW-Objekt auf einem schwarzen Hintergrund erzeugt wurde.

Abspielen von tut_logoHALVE auf PAL

Abspielen von tut_logoHALVE auf PAL (gezoomt)

Abspielen von tut_logoHALVE auf AGA-Screen

Einladen der Session

Abspielen von tut_logo-a-HALVE auf PAL

Abspielen von tut_logo-a-HALVE auf PAL (gezoomt)

Abspielen von tut_logo-a-HALVE auf AGA-Screen

Einladen der Session

6. tut_rotateHALVE

Nun erzeugen wir eine Animation, in deren Hintergrund sich eine Fläche bewegt und im Vordergrund ein Bild rotiert wird. Die Palette im Converter ann auf 'locked' gestellt werden. Jedoch braucht im Processor die Palette nicht eingeladen werden, da alle Bilder im Vooraus generiert werden und dann nur noch im Laufe der Animation die Bewegung der Bilder berechnet wird.

Einladen der Session

Generierung des Vordergrundes (EINMALIG!)

LOADIMAGE lädt wf:Trash/face1 in den Stream.

SCALE skaliert den Stream auf die Breite 192 Pixel und berechnet aus der Breitenangabe die Höhenangabe unter Berücksichtigung des Aspekts. Das Bild wird wieder im Stream gespeichert.

CREATEIMAGE erzeugt ein schwarzes Hintergrundbild der Größe 320x256 Pixel

COMPOSE legt die beiden Bilder übereinander, wobei der kleinere Vordergrund auf dem Hintergrund zentriert wird. Das Ergebnis wird in Temp2 gespeichert.

In diesem Puffer bleibt es die ganze Animation.

Generierung des Hintergrundes (EINMALIG!)

PLASMA erzeugt ein Bild in der Größe 320x256 mit einem Plasmaverlauf im Stream.

ANTIQUÉ wandelt das erzeugte Stream-Bild in Antique Farben um und speichert es erneut im Stream ab.

BALANCING verändert die Farben des Stream-Bildes und nimmt eine Gamma-Korrektur vor. Das Ergebnis wird im Temp3 Puffer gespeichert, in dem es für den Rest der Animation bleibt.

Somit ist das Vorder- und das Hintergrundbild erzeugt. Die weiteren Operationen werden nun bei jedem Bild der Anim ausgeführt.

ROLL bewegt das Hintergrundbild aus Temp3 um die Delta-Werte. Dieses Bild wird in Temp1 zwischengespeichert. 'Wrap' gibt an, ob die Bildteil, die aus dem sichtbaren Bereich verschoben wurden, auf der anderen Seite des Bildes wieder hereingeschoben werden. Dies ist hier nicht der Fall.

ROTATE rotiert das Bild um den angegebenen Amount. Wenn Sie den Radius eines Bildes bestimmen wollen, können Sie die halbe Länge der Diagonale ausrechnen. Dies ist exakt der Radius-Wert, der benötigt wird, um das komplett Bild zu rotieren. Wenn Sie kleinere Werte als Radius wählen, wird nur der kreisförmige Ausschnitt mit dem angegebenen Radius rotiert. Die Formel lautet: $\sqrt{(x/2)^2 + (y/2)^2}$, also $\sqrt{160*160 + 128*128}$ bei unserem Bild. Sie können exakt $\sqrt{160*160 + 128*128}$ in das Gadget eingeben und dann Enter drücken. Wildfire's Gadget können mathematische Ausdrücke verarbeiten. Sofort wird der Wert 205 erscheinen.

Das Ergebnis wird im Stream abgespeichert.

ROTATEBLUR lädt den Stream und rotiert das Bild weiter. Jedoch wird ein Blur-Effekt angewandt. Dabei bedeutet ein Wert von 3 dreimal um einen Pixel rotieren und dabei Blur anwenden, also vorsicht mit allzu großen Werten! Das Ergebnis wird im Stream abgelegt.

Nun wird mittels THRESHOLD und CONVOLVE aus dem Stream Bild ein weiches Alpha-Bild generiert und im Temp4 Puffer abgelegt.

Mittels des Compose-Effetes werden nun der Rotierte Vordergrund und der verschobene Hintergrund unter Verwendung des Alpha-Kanales überlagert.

Abspielen von tut_rotateHALVE auf PAL

Abspielen von tut_rotateHALVE auf PAL (gezoomt)

Abspielen von tut_rotateHALVE auf AGA-Screen

Einladen der Session

7. tut_fadeHALVE

Nun berechnen wir die letzte Animation dieses Tutorials. Sie besitzt wieder eine variable Palette. Laden Sie bitte die Converter-session ein:

Einladen der Session

Im Processor werden als erstes die zwei Gesichter Face1 und Face2 in die Puffer Temp4 und Temp5 geladen. Dort werden Sie nicht verändert, sondern

während der kompletten Animation gespeichert. Das Laden geschieht also nur einmalig.

Sequence 1: Bild 1 bis 30
Das erste Bild wird eingefadet.

Sequence 2: Bild 31 bis 60
Das zweite Bild wird in Graustufen umgewandelt und dieses Graustufenbild wird durch den Balance-Effekt von schwarz auf weiß geblendet. Dieses Graustufenbild wird beim Composen von Gesicht 1 und Gesicht 2 als Alpha-Kanal benutzt. Je Heller das Graustufenbild durch den Balance Effekt wird, desto stärker scheint das zweite Gesicht durch das erste, bis nur noch das zweite Gesicht zu sehen ist.

Sequence 3: Bild 61 bis 120
Ein Teil des zweiten Gesichtes (mit Radius 128 Pixel) wird getwirlet, d.h. verwirbelt.

Sequence 4: Bild 121 bis 150
Das zweite Gesicht wird in der Perspektive gedreht. Mittels des Compose Effektes wird das erste Gesicht dahintergelegt, und ist während des drehen des zweiten (vorderen) gesichtes im Hintergrund zu erkennen.

Sequence 5: Bild 151 bis 210
Das erste Gesicht (welches nun zu sehen ist) wird mit einer Welle überlagert.

Sequence 6: Bild 211 bis 240
Die Linien des vorderen Gesichtes werden zufällig verschoben und langsam nach rechts gedrängt. Im Hintergrund ist das andere Gesicht zu erkennen.

Sequence 7: Bild 241 bis 270
Mittels des 'Schwarzen Loch' Effektes wird das vordere Bild 'weggesaugt' und im Hintergrund ist das zweite Gesicht zu erkennen.

Sequence 8: Bild 271 bis 330
Unter Verwendung des LUT-Effektes werden die Farben des Gesichtes verändert und es damit langsam ausgefadet.

Abspielen von tut_fadeHALVE auf PAL

Abspielen von tut_fadeHALVE auf PAL (gezoomt)

Abspielen von tut_fadeHALVE auf AGA-Screen

Einladen der Session
