

# Malibu 1.4

---

Scala-Emulation für Hollywood

Andreas Falkenhahn



# Inhaltsverzeichnis

<b>1</b>	<b>Allgemeine Informationen</b> .....	<b>1</b>
1.1	Einführung .....	1
1.2	Urheberrecht und Haftungsausschluss .....	2
1.3	Voraussetzung .....	3
1.4	Installation .....	3
<b>2</b>	<b>Verwendung</b> .....	<b>5</b>
2.1	Scala-Skripte starten .....	5
2.2	Umgang mit Zuweisungen .....	5
2.3	Schriftartenpfade einrichten .....	5
2.4	Emulationseinstellungen .....	6
2.5	Emulationsinformationen .....	8
<b>3</b>	<b>Sonstiges</b> .....	<b>15</b>
3.1	F.A.Q. ....	15
3.2	Geschichte .....	16
3.3	Zukunft .....	16
3.4	Danksagungen .....	16



# 1 Allgemeine Informationen

## 1.1 Einführung

Malibu ist ein Plugin für Hollywood, das einen Importfilter für Scala-Präsentationen enthält, die mit der klassischen Scala-Software für Amiga-Computer erstellt wurden. Sobald Malibu installiert ist, kann Hollywood diese Scala-Präsentationen zeigen oder kompilieren.

Mit Malibu ist es nun möglich, diese Scala-Präsentationen ohne die eigentliche klassische Amiga-Hardware zu betreiben. Da die gesamte Emulation auf Hollywoods multimedia application layer läuft, können Sie Scala-Präsentationen auf jeder von Hollywood unterstützten Plattform ausführen. In vielen Fällen ist die Qualität der Bilder, die Malibu zeigt, viel besser als die ursprüngliche Scala-Qualität, da Scala oft viele Bilder mit sehr unterschiedlichen Farben zu einem 8-Bit-Display umwandeln musste, was zu einem massiven Verlust von Farbinformationen führte. Bei Malibu werden jedoch alle Grafiken in Echtfarben angezeigt.

Malibu emuliert nahezu den gesamten Scala-Befehlssatz und unterstützt alle Scala-Versionen von 1.0 bis Scala InfoChannel 500, der letzten Amiga-Version von Scala. Die Emulation erfolgt über den El Capitan Microkernel, der versucht, die bestmögliche Leistung des verwendeten Systems zu erreichen. Zum Beispiel wird die nächste Seite bereits berechnet, während die aktuelle Seite auf dem Display ist. Komplexe Übergangseffekte können ebenfalls vorberechnet und dann problemlos angezeigt werden.

Scala ist ein Programm, mit dem der Amiga zu einem echten Multimedia-Wunder wurde. Lange Zeit gab es keine Produkte, die einen Amiga mit Scala besiegen konnten. Daher wurde Scala in vielen verschiedenen Produktionsbereichen, von Filmen und Fernsehen bis hin zu großen Shows, eingesetzt. Jeder, der mit Scala gearbeitet hat, weiß, welche Stärken es hat. Aber diese Leistung ist nur möglich, weil Scala für die speziellen Chips des Amigas sehr hart codiert ist, was zu dem Problem führt, dass es nicht auf modernen Amiga-Systemen oder anderen Plattformen läuft. Viele Jahre lang beklagten sich alle möglichen Benutzer darüber, dass es kein Scala gab, das RTG-kompatibel war. Diese Zeit ist jetzt vorbei. Malibu belebt Scala auf eine komplett systemfreundliche Art und Weise.

Hier ein kurzer Überblick über die Funktionen von Malibu:

### Grafik:

- Beeindruckende Scala-Übergänge (z.B. Überlagern, Rollen, Fluss, Schieben, Diagonale)
- Unterstützung für die Scala-Textstile Umriss, Schatten, Unterstrichen, Zeichenabstand
- Scala-Formen: Ellipsen, Rechtecke, Linien
- Emulation von Hintergründen mit Raster-Typen
- Komplette Pinselunterstützung (inkl. Zuschneiden und Größenänderung)
- Animationen werden unterstützt
- Übergangseffekte für alle Scala-Objekte
- Unterstützung für Stile wie Abschrägung und Hintergrund
- Läuft vollständig im 24-Bit-Modus
- Farbschriften werden unterstützt
- Weiches scrolling für Scala Fly-Ons

- Emulation aller Grafikattribute
- Pinseltransparenz wird vollständig unterstützt
- Objekte können mit Übergangseffekten vom Bildschirm entfernt werden
- Alle Paletteneinstellungen werden respektiert und korrekt konvertiert

### **Sound:**

- Unterstützung für Soundmodule und Samples
- Audioausgabe über AHI inkl. volles Mischen
- Sound überblenden wird unterstützt
- Die Sound-Implementierung ist für die Verwendung auf Pegasos, Amithlon usw. umsetzbar.

### **Umgebung:**

- Alle Scala-Versionen bis InfoChannel 500 werden unterstützt
- Randeinstellungen und Tabulatoren werden respektiert
- Scala-Variablen können verwendet werden
- Schaltflächen und Ereignisse werden emuliert
- Fast vollständige Emulation des Scala-Befehlssatzes
- Scala-Präsentationen können zu ausführbaren Dateien kompiliert werden
- Die Emulation kann detailliert konfiguriert werden
- Völlig plattformunabhängige Implementierung
- Scala-Präsentationen können als Hollywood-Skripte gespeichert werden
- Absolut systemfreundlich

## **1.2 Urheberrecht und Haftungsausschluss**

Malibu ist © Copyright 2002-2018 bei Andreas Falkenhahn (im folgenden "der Autor" genannt). Alle Rechte vorbehalten.

Das Programm wird zur Verfügung gestellt "wie es ist" und der Autor kann für keinerlei Schäden, welcher Natur sie auch immer sein mögen, verantwortlich gemacht werden. Sie benutzen dieses Programm völlig auf eigene Gefahr und eigenes Risiko. Der Autor gibt keinerlei Garantien in Verbindung mit der Benutzung dieses Programmes, nicht einmal die Garantie der Funktionstüchtigkeit.

Das Programm kann frei weitergegeben werden solange die folgenden drei Bedingungen erfüllt sind:

1. Es dürfen keine Änderungen am Programm vorgenommen werden.
2. Das Programm darf nicht verkauft werden.
3. Wenn Sie das Programm auf einer Coverdisk veröffentlichen möchten, müssen Sie erst um Erlaubnis fragen.

Alle Warenzeichen sind Eigentum ihrer jeweiligen Firmen.

FÜR DIESES PROGRAMM GIBT ES KEINE GARANTIE, SOWEIT ES DIE ANZUWENDENDEN GESETZE ZULASSEN. SOFERN ANDERSWO NICHTS

GEGENTEILIGES GESCHRIEBEN STEHT STELLEN DER AUTOR UND/ODER DRITTE DAS PROGRAMM "SO WIE ES IST" ZUR VERFÜGUNG, OHNE IRGEND-EINE GARANTIE, WEDER DIREKT NOCH INDIREKT. DIES BEINHÄLTET, IST ABER NICHT DARAUf BESCHRÄNKT, VERKÄUFLICHKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN VERWENDUNGSZWECK. DAS VOLLSTÄNDIGE RISIKO DER QUALITÄT UND AUSFÜHRBARKEIT DES PROGRAMMS LIEGT BEIM ANWENDER. SOLLTE SICH DAS PROGRAMM ALS DEFEKT HERAUSSTELLEN, LIEGEN ALLE KOSTEN FÜR SERVICE, INSTANDSETZUNG ODER NACHBESSERUNG BEIM ANWENDER.

KEIN COPYRIGHT-INHABER ODER DRITTER, DER DAS PROGRAMM WIE OBEN ERLAUBT WEITERVERKAUFT, KANN FÜR SCHÄDEN IRGENDWELCHER ART HAFTBAR GEMACHT WERDEN (DIES BEINHÄLTET, IST ABER NICHT BESCHRÄNKT AUF, DATENVERLUST INFOLGE UNFÄHIGKEIT DES PROGRAMMS, MIT ANDEREN PROGRAMMEN ZUSAMMENZUARBEITEN), SELBST WENN EIN SOLCHER INHABER ODER DRITTER AUF DIE MÖGLICHKEIT EINES SOLCHEN SCHADENS HINGEWIESEN WURDE, AUSSER ES BESTEHT EINE SCHRIFTLICHE EINWILLIGUNG ODER WIRD VOM GESETZ VERLANGT.

### 1.3 Voraussetzung

- Hollywood 5.0 oder besser

### 1.4 Installation

Die Installation von Malibu ist unkompliziert und einfach: Kopieren Sie einfach die Datei `malibu.hwp` für die Plattform in Hollywoods Plugin-Verzeichnis. Auf allen Systemen außer auf AmigaOS und kompatiblen Systemen müssen Plugins in einem Verzeichnis mit dem Namen `Plugins` gespeichert werden, das sich im selben Verzeichnis wie das Hauptprogramm von Hollywood befindet. Auf AmigaOS und kompatiblen Systemen müssen Plugins stattdessen in `LIBS:Hollywood` installiert werden. Unter Mac OS X muss sich das Verzeichnis `Plugins` im Verzeichnis `Resources` des Programmpakets befinden, d.h. im Verzeichnis `HollywoodInterpreter.app/Contents/Resources`. Beachten Sie, dass `HollywoodInterpreter.app` im Programmpaket `Hollywood.app` selbst gespeichert ist, nämlich in `Hollywood.app/Contents/Resources`.

Unter Windows sollten Sie auch die Datei `Malibu.chm` in das Verzeichnis `Docs` Ihrer Hollywood-Installation kopieren. Dann können Sie die Online-Hilfe von der IDE erhalten.

Kopieren Sie unter Linux und Mac OS das Verzeichnis `Malibu`, welches sich im Verzeichnis `Docs` des Malibu-Distributionsarchivs befindet, in das Verzeichnis `Docs` Ihrer Hollywood-Installation. Beachten Sie, dass sich das Verzeichnis `Docs` unter Mac OS im Programmpaket `Hollywood.app` befindet, d.h. in `Hollywood.app/Contents/Resources/Docs`.





## 2 Verwendung

### 2.1 Scala-Skripte starten

Sobald das Malibu-Plugin korrekt installiert wurde, wird Hollywood automatisch Ihre Scala-Skripte anzeigen können. Daher können Sie Ihre Scala-Skripte jetzt genauso starten wie Ihre Hollywood-Skripte, z.B. mithilfe der Hollywood-GUI oder der CLI.

Weitere Informationen zum Starten von Skripten und zu möglichen Argumenten finden Sie in Ihrem Hollywood-Handbuch.

### 2.2 Umgang mit Zuweisungen

In der Regel verwenden Scala-Skripte Pfadnamen, die Zuweisungen enthalten, um auf externe Dateien zu verweisen. Auf Amiga-Systemen ist dies kein Problem, da Sie diese Zuordnungen einfach einrichten können und es funktioniert. Auf Nicht-Amiga-Systemen gibt es jedoch kein Konzept von Zuweisungen. Wenn also ein Scala-Skript versucht, Dateien über Amiga-Assigns zu öffnen (z.B. `SCALA:Scripts/Background.iff`), öffnet Malibu ein Dialogfenster. Sie wählen nun den Pfad auf Ihrer Festplatte aus, der dem Amiga-Assign entsprechen soll, damit Malibu alle Dateien finden kann. Sie können festlegen, dass diese Zuweisungen dauerhaft oder nur für die aktuelle Sitzung gespeichert werden.

Bei der Wahl der dauerhaften Zuweisung werden diese in der Datei `malibu.ini` gespeichert. Sie können Zuweisungen auch manuell in `malibu.ini` schreiben, indem Sie die folgende Syntax verwenden:

```
Assign 1="SCALA:E:/AmigaApps/Scala"
Assign 2="Obligement12:E:/ScalaMags/Obligement12"
```

Sie können so viele Zuordnungen hinzufügen, wie Sie möchten, aber sie müssen bei 1 beginnen und es darf keine Lücken in der Sequenz geben. Jede Zuweisungsdefinition besteht aus dem Zuweisungsnamen, der auf einem Doppelpunkt endet, gefolgt von dem Pfad, der für diese Zuweisung verwendet werden soll.

### 2.3 Schriftartenpfade einrichten

Auf AmigaOS und kompatiblen Systemen wird Malibu in `FONTS:` nach allen Schriften suchen, die vom Skript verwendet werden. Da dies auf Nicht-Amiga-Systemen nicht möglich ist, müssen Sie die Schriftart-Suchpfade manuell angeben. Dies geschieht durch Bearbeiten der Datei `malibu.ini`, die am gleichen Ort gespeichert werden muss, wie die ausführbare Hauptdatei `malibu.hwp`.

Wenn die Datei `malibu.ini` noch nicht da ist, dann erstellen Sie sie einfach. Es ist nur eine einfache Textdatei. Fügen Sie die folgende Zeile in `malibu.ini` ein, um Schriftart-Suchpfade zu Malibu hinzuzufügen:

```
Font search path 1="C:/Amiga files/Scala/Fonts"
```

Die obige Zeile fügt den Malibu-Schriftsuchpfaden den Pfad `"C:/Amiga-Dateien/Scala-Fonts"` hinzu. Dies bedeutet, dass Malibu diesen Pfad bei der Suche nach Schriftarten immer überprüft. Sie können so viele Schriftartensuchpfade hinzufügen, wie Sie möchten.

Sie müssen nur den Zähler neben dem Schlüsselnamen in der Datei `malibu.ini` erhöhen. Um zwei weitere Suchpfade hinzuzufügen, fügen Sie einfach diese zwei Zeilen hinzu:

```
Font search path 2="D:/Amiga files/Scala/Fonts"
Font search path 3="E:/Amiga files/Scala/Fonts"
```

## 2.4 Emulationseinstellungen

Auf AmigaOS und kompatiblen Systemen bietet Malibu ein Programm, mit dem Sie die Scala-Emulation feinabstimmen können. Sie können das Einstellungsprogramm über das Plugin-Menü in der Hollywood GUI oder über die Workbench starten. Das Einstellungsprogramm schreibt Ihre Einstellungen in die Datei `malibu.ini`, die im selben Verzeichnis wie die Hauptdatei `malibu.hwp` gespeichert ist. Sie können auch `malibu.ini` direkt ändern - siehe unten für Anweisungen.

Auf allen anderen Plattformen gibt es kein spezielles Einstellungsprogramm. Stattdessen müssen Sie `malibu.ini` direkt ändern. Die Datei `malibu.ini` muss im gleichen Verzeichnis wie die Hauptdatei `malibu.hwp` gespeichert sein. Wenn es noch nicht da ist, erstellen Sie sie einfach. Im Folgenden werden alle Optionen beschrieben, die im Einstellungsprogramm auf AmigaOS und Kompatible eingestellt werden können. Am Ende der Beschreibung der einzelnen Optionen finden Sie den Schlüssel, der in der Einstellungsdatei `malibu.ini` festgelegt werden muss, damit Sie bestimmte Optionen aktivieren oder deaktivieren können. Um beispielsweise die Skript-Schleife zu aktivieren, fügen Sie folgende Zeile zu `malibu.ini` hinzu:

```
Loop script=True
```

Folgende Dinge können konfiguriert werden:

### Emulationsoptionen:

- Loop script: Wenn Sie dieses Kästchen ankreuzen, wird jedes Scala-Skript, das Sie starten, beim Erreichen des Endes eine Schleife machen. Der Schlüssel für diese Einstellung ist **Loop script**. Der Standardwert ist **True**.
- Emulate pause commands at page end: Die meisten Scala-Skripte haben Pausenbefehle am Ende jeder Seite. Wenn Sie dieses Kontrollkästchen aktivieren, werden sie emuliert. Sofern Sie nicht einen sehr schnellen Amiga haben, müssen Sie dieses Kontrollkästchen nicht aktivieren, da Malibu automatisch mit dem Vorbereiten der nächsten Seite beginnt, wenn das Ende einer Seite erreicht ist. Und das dauert meistens länger als der ursprüngliche Pause-Befehl. Sie sollten dieses Kästchen nur aktivieren, wenn Sie einen sehr schnellen Amiga haben und Skripte zu schnell laufen. Der Schlüssel für diese Option ist **Emulate pause end**. Der Standardwert ist **True**.
- Emulate pause commands during page: Wenn Sie dieses Kontrollkästchen aktivieren, werden Pausenbefehle, die während einer Seite angezeigt werden, emuliert. Es wird empfohlen, dieses Kontrollkästchen zu aktivieren. Der Schlüssel für diese Option ist **Emulate pause page**. Der Standardwert ist **True**.
- Emulate pause accurately: Wenn diese Option aktiviert ist, wird Malibu versuchen, die Pausenbefehle von Scala so genau wie möglich zu emulieren. Die Aktivierung dieser Option kann zu Zeitproblemen führen, da die Übergangseffekte von Hollywood nicht genau so lang sind wie die von Scala. Deshalb ist es standardmäßig deaktiviert. Der Schlüssel für diese Option ist **Accurate pause**. Der Standardwert ist **False**.

### Animationsoptionen:

- Play animations: Wenn Sie dieses Kästchen markieren, werden Animationen abgespielt. Bitte beachten Sie, dass Sie auch "Play animations from disk" wählen sollten, da Animationen viel Speicherplatz benötigen, zumal sie alle in 24-bit konvertiert werden müssen. Z.B. würde eine 640x480 Große Animation in 16 Farben mit 100 Frames etwa 90 Megabyte benötigen, wenn sie in den Speicher geladen würde (!!!). Deshalb sollten Sie eigentlich auch "Play animations from disk" wählen. Der Schlüssel für diese Option ist **Play anims**. Standardwert ist True.
- Play animations from disk: Wenn Sie dieses Kästchen markieren, wird Malibu alle Animationen direkt von der Festplatte abspielen. Dies ist ein wenig langsamer, spart aber viel Speicher. Deshalb wird diese Option empfohlen und der Schlüssel ist **Play anims from disk**. Der Standardwert ist True.

### Übergangseffekt-Optionen:

- Picture transition speed: In dieser Combobox können Sie eine Geschwindigkeit festlegen, die für alle Bildübergänge verwendet wird. Der Schlüssel für diese Option ist **BGPic FX speed**. Ein Wert von 0 bedeutet langsam, 1 bedeutet normal und 2 bedeutet schnell. Der Standardwert ist 1.
- Default picture transition: Hollywood unterstützt nicht alle Scala-Übergangseffekte. Der Übergang, den Sie hier auswählen, wird angezeigt, wenn Malibu für den erforderlichen Scala-Übergang keinen gleichen Hollywood-Übergang finden kann. Die Standardeinstellung dafür ist 'random transition'. Der Schlüssel für diese Option ist **Default BGPic FX**. Es muss auf den ganzzahligen Wert eines Hollywood-Übergangseffekts gesetzt werden, der um eins erhöht wird. Um einen zufälligen Effekt zu verwenden, setzen Sie ihn auf 0, was auch der Standardwert ist.
- Object transition speed: In dieser Combobox können Sie eine Geschwindigkeit festlegen, die für alle Objektübergänge verwendet wird. Der Schlüssel für diese Option ist **Object FX speed**. Ein Wert von 0 bedeutet langsam, 1 bedeutet normal und 2 bedeutet schnell. Der Standardwert ist 1.
- Default object transition: Dies entspricht 'default picture transition' mit der Ausnahme, dass der Übergangseffekt, den Sie hier angeben, für Objektübergänge verwendet wird. Der Schlüssel für diese Option ist **Default object FX**. Es muss auf den ganzzahligen Wert eines Hollywood-Übergangseffekts gesetzt werden, der um eins erhöht wird. Um einen zufälligen Effekt zu verwenden, setzen Sie ihn auf 0, was auch der Standardwert ist.

### Exportoptionen:

- Save script to: Wenn Sie hier den Namen eines Dateinamens eingeben, speichert Malibu in dieser Datei automatisch eine Kopie des Scala-Skripts, das in ein Hollywood-Skript konvertiert wurde. Dadurch können Sie einige manuelle Änderungen am Skript vornehmen. Der Schlüssel für diese Option ist **Save script**. Standardmäßig wird eine leere Zeichenkette verwendet, was bedeutet, dass Malibu das aus dem eingegebenen Scala-Skript das generierte Hollywood-Skript nicht speichern sollte.

### Verschiedene Optionen:

- Enable quit with ESC: Wenn Sie dieses Kästchen markieren, können Sie jederzeit durch

Drücken der Escape-Taste beenden (nützlich im Vollbildmodus, wenn das Fenster kein Schließsymbol hat). Der Schlüssel für diese Option ist **Enable escape quit**. Der Standardwert ist **True**.

- Show busy pointer when working: Wenn Sie dieses Kästchen markieren, zeigt Malibu seinen Besetzt-Zustand an, indem er den Besetzt-Zeiger anzeigt. Auf langsameren Systemen kann es sinnvoll sein, dieses Kontrollkästchen zu aktivieren. Der Schlüssel für diese Option ist **Show busy pointer**. Standardwert ist **False**.
- Hide pointer all the time: Aktivieren Sie dieses Kontrollkästchen, wenn der Mauszeiger überhaupt nicht sichtbar sein soll. Nützlich für Präsentationen. Der Schlüssel für diese Option ist **Hide pointer**. Der Standardwert ist **False**.
- Allow relative paths: Scala verwendet immer absolute Pfaddefinitionen für externe Daten. Dies hat den Nachteil, dass Skripte nicht ohne Modifikationen auf anderen Systemen laufen, da jeder Dateiname absolut adressiert wird, z.B. `dh0:Apps/Scala Scripts/Demo.iff`. Wenn Sie dieses Kontrollkästchen aktivieren, sucht Malibu zuerst, ob die Datei im aktuellen Verzeichnis vorhanden ist, und lädt sie von dort. Andernfalls wird es den vollständigen Pfad versuchen. Der Schlüssel für diese Option ist **Try relative path**. Der Standardwert ist **True**.
- Substitute Intellifonts: Wenn diese Option ausgewählt ist, ersetzt Malibu alle Intellifonts durch eine Standardschriftart. Dies kann nützlich sein, wenn Sie Malibu auf einem System ausführen, das Compugraphic Intellifonts (wie Windows, Mac OS X oder Linux) nicht unterstützt. Der Schlüssel für diese Option ist **Replace intelli fonts**. Auf Nicht-Amiga-Systemen wird **True** und auf Amiga-Systemen **False** verwendet.
- Never fail because of missing fonts: Wenn Sie diese Option auswählen, ersetzt Malibu automatisch fehlende Schriftarten. Es wird nicht empfohlen, diese Option festzulegen, da Sie andernfalls nicht über fehlende Schriftarten informiert werden, sodass Sie diese Schriftarten nicht finden und installieren können. Diese Option ist also nur für temporäre Testzwecke nützlich. Der Schlüssel für diese Option ist **Never fail on fonts**. Der Standardwert ist **False**.
- Emulate font antialiasing: Wählen Sie diese Option, um das Schriftart-Antialiasing zu aktivieren. Beachten Sie, dass dies kein echtes vektorbasiertes Antialiasing ist, sondern nur eine Annäherung, die auf Unschärfe basiert, so dass es nicht so gut aussieht. Deshalb ist es standardmäßig deaktiviert. Der Schlüssel für diese Option ist **Emulate antialias**. Der Standardwert ist **False**.

## 2.5 Emulationsinformationen

Dieses Kapitel enthält einige Informationen zu den Scala-Befehlen, die von Malibu erkannt und emuliert werden.

### AACOLOR:

- nicht emuliert
- Anti-Aliasing wird von Hollywood noch nicht unterstützt

### ANIM:

- 100% emuliert

**ATTRIBUTES:**

- 3d: nicht emuliert
- antialias: nicht emuliert
- bevel: emuliert
- bold: emuliert
- center: emuliert
- edge: emuliert
- inactive: nicht emuliert
- italics: emuliert
- jam: emuliert
- kerning: nicht emuliert
- left: emuliert
- none: emuliert
- remap: emuliert
- right: emuliert
- underline: emuliert

**BLANK:**

- 100% emuliert

**BOX:**

- 100% emuliert

**BRUSH:**

- 100% emuliert einschließlich Schnitt, Maßstab und Transparenz
- alte Syntax (Scala 1.0) wird unterstützt

**BUTTON:**

- 100% emuliert

**COLOR:**

- 100% emuliert

**CONTINUE:**

- nicht emuliert
- ARexx wird von Hollywood nicht unterstützt

**CYCLE:**

- nicht emuliert
- Palettenzyklen sind in 24-Bit-Bildschirmmodi nicht möglich

**END:**

- 100% emuliert

**ELLIPSE:**

- 100% emuliert

**EVENT:**

- 100% emuliert

**EX:**

- nicht emuliert
- externe Module sind hardwareabhängig

**EXECUTE:**

- CLI-Ausführung emuliert
- ARexx und WB-Ausführung wird derzeit nicht unterstützt

**FKEYS:**

- 100% emuliert

**FONT:**

- 100% emuliert

**GETVAR:**

- nicht emuliert
- ARexx wird von Hollywood nicht unterstützt

**GOTO:**

- 100% emuliert

**GRID:**

- nicht emuliert

**IF:**

- 100% emuliert

**INTERACTIVE:**

- 100% emuliert

**JOYSTICK:**

- nicht emuliert

**LINE:**

- 100% emuliert

**MARGINS:**

- 100% emuliert

**MARK:**

- box: emuliert
- complement: emuliert
- fill: emuliert
- none: emuliert
- replace: emuliert

**MARKSOUND:**

- 100% emuliert

**MOUSE:**

- 100% emuliert

**NUMKEYS:**

- 100% emuliert

**PALETTE:**

- 12bit definitions: emuliert
- 24bit definitions: emuliert

**PAUSE:**

- 100% emuliert
- beide Syntaxversionen werden von PAUSE erkannt (der alte von Scala 1.0 bis MM200 und der neue von MM300 und höher)

**PICTURE:**

- 100% emuliert

**POINTER:**

- nicht emuliert
- Verbergen des Mauszeigers auf dem Workbench-Bildschirm kann den Benutzer verwirren

**QUIT:**

- nicht emuliert
- ARexx wird von Hollywood nicht unterstützt

**RESOLUTION:**

- 100% emuliert

**RETURN:**

- 100% emuliert

**SCRIPT:**

- 100% emuliert

**SELECT:**

- box: emuliert
- complement: emuliert
- fill: emuliert
- none: emuliert
- replace: emuliert

**SELECTSOUND:**

- 100% emuliert

**SET:**

- 100% emuliert

**SETVAR:**

- nicht emuliert
- ARexx wird von Hollywood nicht unterstützt

**SHOW:**

- nicht emuliert
- ARexx wird von Hollywood nicht unterstützt

**SOUND:**

- play: 100% emuliert
- stop: 100% emuliert
- volume: 100% emuliert
- wait: 100% emuliert
- delay: 100% emuliert
- fade: 100% emuliert
- fadein: 100% emuliert
- fadeout: nicht emuliert
- pan: nicht emuliert
- period: 100% emuliert

**STYLE:**

- transparency: emuliert
- shadowdir: emuliert
- shadowlen: emuliert
- 3dlen: nicht emuliert



- boldsize: nicht emuliert
- underpos: emuliert
- undersize: emuliert
- underair: emuliert
- italics: emuliert
- spacing: emuliert
- linespacing: emuliert
- antialiaslevel: emuliert
- remap: emuliert
- rastertype: emuliert
- linethickness: emuliert
- bevellen: emuliert

**TABS:**

- 100% emuliert

**TEXT:**

- 100% emuliert
- einschließlich Tabs, Variablen und Escape-Sequenzen
- alte Syntax (Scala 1.0) wird unterstützt

**TEXTOUT:**

- 100% emuliert

**TEXTWIPE:**

- emuliert
- wichtige Scala-Effekte verwenden ähnliche Hollywood-Effekte
- alle anderen Effekte verwenden zufällige Übergänge

**WIPE:**

- emuliert
- wichtige Scala-Effekte verwenden ähnliche Hollywood-Effekte
- alle anderen Effekte verwenden zufällige Übergänge
- alte Syntax (Scala 1.0) wird hier unterstützt

**WORDWRAP:**

- 100% emuliert



## 3 Sonstiges

### 3.1 F.A.Q.

Hier sind die Antworten auf einige häufig gestellte Fragen. Lesen Sie diese Informationen vor dem Senden von E-Mails.

**Q: Wieso bekomme ich für jedes Skript, das ich starte, keinen Speicher mehr?**

A: Wählen Sie in den Einstellungen "Play animations from disk". Wenn Sie immer noch nicht genügend Arbeitsspeicher zur Verfügung haben, gibt es keine andere Möglichkeit als Ihren Arbeitsspeicher zu erweitern.

**Q: Warum sind Animationen so langsam?**

A: Jedes Einzelbild muss vom planaren Format in True Color konvertiert werden. Dies dauert einige Zeit und daher kann die Wiedergabe langsam sein, aber es gibt keine andere Möglichkeit, dies zu tun. Versuchen Sie, eine Animation mit MultiView anzusehen; sie wird auch sehr langsam sein.

**Q: Warum dauert das Laden von Animationen so lange?**

A: Dies ist das gleiche Problem wie die vorherige Frage. Hollywood muss jedes Einzelbild umwandeln und wenn es viele Farben und eine hohe Auflösung verwendet, wird es einige Zeit dauern.

**Q: Malibu kann meine Animationen nicht laden.**

A: Überprüfen Sie, ob ein Datentyp für die Animation vorhanden ist. Manchmal verwendet Scala ein spezielles Animationsformat (ANIM8), das von animation.datatype nicht unterstützt wird und daher nicht von Malibu geladen werden kann.

**Q: Warum sieht Anti-Aliasing so schlecht aus?**

A: Scala bietet Anti-Aliasing für Bitmap- und Intellifonts mit einigen cleveren dedizierten Algorithmen. Hollywood unterstützt jedoch nur Antialiasing für TrueType-Schriftarten. So emuliert Malibu das Antialiasing von Scala mit Kantenunschärfe, aber das sieht natürlich nicht so gut aus wie Scalas dedizierte Algorithmen.

**Q: Warum wird die Scala-Zeitplanfunktion nicht emulieren?**

A: Dies wäre momentan zu schwierig, da Malibu viel mehr Zeit für die Vorbereitung jeder Seite benötigt, als Scala benötigt. Daher wäre eine Zeitplanunterstützung hier nicht sinnvoll. Aber es ist für PPC-Versionen geplant, in denen die Seitenvorbereitung schnell genug ist, um sie zu unterstützen.

**Q: Ist ARexx-Unterstützung geplant?**

A: Ja, ich arbeite daran. Hollywood muss jedoch zuerst Befehle dafür anbieten, bevor ich es implementieren kann.

**Q: Was ist mit Scala EX's?**

A: Dies wird nicht emuliert und wird in der Zukunft nicht emuliert, weil es fast unmöglich ist. Scala EX ist stark von externer Hardware abhängig und mir ist nicht klar, wie all das funktioniert, weil ich keine Hardware habe. Sie werden ohne dies leben müssen.

**Q: Was ist mit den coolen Kreis-Effekt von Scala?**

A: Dies ist auch fast unmöglich zu emulieren. Sie benötigen einen sehr schnellen Prozessor, um alle verschiedenen Zyklen in 24-Bit zu berechnen. Farbzyklen sind Tricks, die nur auf palettenbasierten Displays sehr schnell durchgeführt werden können. Im 24-Bit-Modus müssten Sie jedes Pixel ändern. 68k-Prozessoren sind weit davon entfernt, dies mit einer akzeptablen Geschwindigkeit zu tun.

## 3.2 Geschichte

Für ein vollständiges Änderungsprotokoll von Malibu beachten Sie bitte die auf englisch verfasste Datei `history.txt`.

## 3.3 Zukunft

- Unterstützung für den 3D-Stil hinzufügen
- Unterstützung für verknüpfte Übergänge hinzufügen
- eine bessere Unterstützung für die Schriftart-Anti-Alias hinzufügen
- 100% richtige Unterstützung für Animationen (Unterstützung von Objekten auf jedem Einzelbild!)
- mehr Scala-Übergänge implementieren (in Hollywood)
- Unterstützung für das Überprüfen von Ereignissen während den Pausenzuständen implementieren (ScalaVideo.script)
- Zeitplan-Unterstützung implementieren
- ARexx Unterstützung (vielleicht)
- Unterstützung für Sheared-Intelli-Schriftarten
- Verbesserung der Kompatibilität

Bitte schicken Sie mir eine mail, wenn Sie ein paar nette Ideen haben, die in Malibu umgesetzt werden sollen.

## 3.4 Danksagungen

Malibu wurde zwischen Dezember 2002 und Mai 2003 von Andreas Falkenhahn geschrieben. Es läuft als Plugin für den Multimedia Application Layer Hollywood und konvertiert Scala-Skripte on-the-fly in Hollywood-Skripte.

Malibu ist ein hochwertiges Softwareprodukt für Amiga-Computer von Airsoft Softwair.

Hollywood wurde auch von Andreas Falkenhahn geschrieben.

Danke an Jean-Yves Auger von Pixel Art.

Malibu wurde auf meinem Amiga 1200 entwickelt, der mit einem Phase5 Blizzard PPC 603e 200Mhz mit einer 68040 CPU, einer Phase5 BVision PPC Grafikkarte und 80 Megabyte RAM ausgestattet ist.

Ein besonderer Dank geht an Helmut Haake und Dominic Widmer für die Übersetzung des Handbuchs ins Deutsche. Fehler oder Verbesserungsvorschläge bzgl. des deutschen Handbuchs bitte an das Übersetzungsteam richten, welches unter [handbuch@gmx.ch](mailto:handbuch@gmx.ch) erreicht werden kann.

Wenn Sie mich kontaktieren möchten, können Sie entweder eine E-Mail an [andreas@airsoftsoftwair.de](mailto:andreas@airsoftsoftwair.de) senden oder benutzen Sie das Kontaktformular auf <http://www.hollywood-mal.com>.

