

CyberQT

Thore Böckelmann

COLLABORATORS

	<i>TITLE :</i> CyberQT		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Thore Böckelmann	June 24, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	CyberQT	1
1.1	main	1
1.2	background	2
1.3	requirements	2
1.4	install	3
1.5	encodings	3
1.6	shell	4
1.7	workbench	8
1.8	dither	9
1.9	operation	9
1.10	faq	10
1.11	contact	11
1.12	history	12
1.13	bugs	14
1.14	todo	15
1.15	xanim	15
1.16	argue	16
1.17	distribution	17
1.18	credits	17
1.19	disclaimer	17

Chapter 1

CyberQT

1.1 main

CyberQT 1.2

Copyright ©1996-1997 by Thore Böckelmann

CyberQT ist ein schneller QuickTime-Player für Rechner mit 68020 und AmigaOS 3.0 oder höher und einer Grafikkarte mit CyberGraphX oder AGA Chipsatz

Hintergrund
Vorraussetzungen
Installation

Unterstützte Kodierungen

Shell-Parameter
Workbench-Parameter

Wie steuert man das?

Häufig gestellte Fragen

Post an mich

Geschichte
Fehler
Was gibt's noch zu tun?

XAnim
Argue

Vertrieb
Rechtliches und Danksagungen
Disclaimer

CyberQT ist Freeware. Sie brauchen mir kein Geld oder irgendetwas anderes schicken (aber wenn Sie unbedingt wollen werde ich Sie natürlich nicht daran

hindern :). Ich wäre allerdings froh wenn Sie mir eine EMail schicken würden, wenn Ihnen CyberQT gefällt.

1.2 background

Der Hauptgrund für die Entwicklung von CyberQT war der, daß ich mit den bisher existierenden Programmen für QuickTime-Animationen nicht so ganz glücklich war. Das Programm QT von Marcus Comstedt läuft zwar wunderbar auf Grafikkarten unter CyberGraphX, hat aber den großen Nachteil, daß es grundsätzlich die komplette Animation vor dem Abspielen dekodiert. Dadurch wird zwar ein sehr genaues Timing beim Abspielen möglich, aber gleichzeitig wird die mögliche Größe der abspielbaren Animationen durch den freien Systemspeicher stark eingeschränkt. Außerdem dauert es selbst mit einem MC68060 sehr lange bis zum Beispiel eine Animation mit "nur" 10MB komplett dekodiert ist.

Also begann ich CyberQT zu schreiben. Die bisherige Entwicklungszeit (bis V1.0) betrug gerade mal zwei Wochen. Das liegt zum Einen daran, daß ich große Teile von CyberAVI (mein Programm zum Abspielen von AVI-Animationen) wiederverwenden konnte, zum Anderen daran, daß Apple eine wirklich sehr vorbildliche Beschreibung des QuickTime-Formates zur Verfügung stellt. Bei der Entwicklung von CyberAVI hätte ich mir etwas ähnliches von Microsoft gewünscht.

CyberQT wurde entwickelt mit:

- A4000 mit CyberStorm MK2 mit MC68060/50
- AmigaOS 3.1
- 2MB ChipRAM
- 16MB FastRAM
- FastLaneZ3 SCSI-Host (Plattenkapazität ca. 1.3GB)
- CyberVision64 mit 4MB Grafikspeicher
- Philips 17B 17" Monitor

1.3 requirements

minimale Hardwarevoraussetzungen:

- Amiga mit AmigaOS 3.0 (V39)
- MC68020
- Grafikkarte mit CyberGraphX (z.B. CyberVision64, Picasso II, Retina, etc) oder AGA Chipsatz
- 2MB FastRAM, 512K ChipRAM

Softwarevoraussetzungen:

- CyberGraphX 2.21 oder höher (wenn eine Grafikkarte benutzt wird)
- asyncio.library V39 (im Archiv)

empfohlene Hardwarevoraussetzungen:

- Amiga mit AmigaOS 3.1 (V40)
 - MC68040 mit mindestens 25MHz
 - Grafikkarte mit CyberGraphX 2.21 oder höher
-

Ich empfehle eine registrierte Version von CyberGraphX, weil die unregistrierte Version keine 24Bit-Bildschirme unterstützt.

1.4 install

Benutzen Sie einfach das beiliegende Installer-Script.

Wenn Sie genug Erfahrung mit dem Amiga haben können Sie CyberQT natürlich auch "von Hand" installieren. Kopieren Sie einfach die `asyncio.library` nach LIBS: und CyberQT wohin Sie wollen. Das ist alles.

1.5 encodings

Die von CyberQT benutzte Implementation der Dekodierung von QuickTimes basiert auf Mark Podlipecs Programm XAnim.

Ich werde mehr Dekodierer hinzufügen, soweit sie benötigt werden und ich Animationen zum Testen bekomme.

Bisher unterstützte Kodierungen:

Video:

Radius CinePak	(CVID)	8 Bit (nur Graustufen)
Radius CinePak	(CVID)	16/24/32 Bit
Kodak Photo CD	(KPCD)	16/24/32 Bit
Microsoft Video 1	(MSVC)	8/16 Bit
Apple Animation	(RLE)	1/4/8 Bit (Farbe und Graustufen)
Apple Animation	(RLE)	16/24/32 Bit
Apple Uncompressed	(RAW)	1/4/8 Bit (Farbe und Graustufen)
Apple Uncompressed	(RAW)	16/24/32 Bit
Apple Video	(RPZA)	16 Bit
Apple Graphics	(SMC)	8 Bit
Component Video	(YUV2)	16/24/32 Bit
Intel Raw	(YUV9)	16/24/32 Bit

Audio:

PCM	8/16 Bit Mono/Stereo
PCM0	8/16 Bit Mono/Stereo
TWOS	8/16 Bit Mono/Stereo
IMA ADPCM	16 Bit Mono/Stereo

Ich suche immer noch nach weiteren Kodierungen. Schreiben Sie mir, falls Sie Animationen mit einer der folgenden Kodierung haben:

Video:

- KPCD (Kodak PhotoCD)
- oder andere unbekannte Formate, außer "Intel Indeo"

Audio:

- MAC3
- MAC6
- μ -Law

Sie können mir Animationen mit einer der oben genannten Kodierung zuschicken. Packen Sie sie einfach in ein Archiv (z.B. LhA) und schicken Sie mir diese uuencoded zu einer meiner EMail-Adressen.

Mein größtest Problem ist und bleibt Intel Indeo. Intel verlangt \$5000 (in Worten: fünftausend US-Dollar) für eine Lizenz ihrer Sourcecodes. Es scheint also, als wäre es unmöglich Indeo jemals zu unterstützen :(

1.6 shell

Das Kommandozeilenmuster für CyberQT ist...

```
FILES/M,PUBSCREEN/K,SCREENMODEREQ=SMR/S,FORCE24/S,GRAY=GREY/S,AGA/S,HAM8/S,
DITHER/S,BUFFERSIZE=BUF/K/N,DELAY/K/N,LOOP/S,MAXFPS/S,NOSOUND/S,NOVIDEO/S,
SKIP/S,AUDIOPRELOAD=APL/S,MAGNIFY/K/N,STATS/S,QUIET/S,DEBUG/S
```

FILES Eine oder mehrere QuickTime-Animationen, die Sie sich ansehen wollen. Wenn Sie keine Dateinamen angeben wird ein Filerequester erscheinen, in dem Sie eine oder mehrere Animationen auswählen können. Joker, wie ?, #? oder *, sind hierbei erlaubt.

PUBSCREEN Diese Option leitet das Abspielen in ein Fenster auf einem Publicscreen um. Sowohl die Animation, als auch der Bildschirm müssen eine Farbtiefe von mindestens 15 Bit haben. Falls der angegebene Bildschirm nicht existiert oder falls die Farbtiefe des Bildschirms oder der Animation kleiner als 15 Bit ist wird CyberQT einen eigenen Bildschirm öffnen.

HINWEIS:

Diese Option funktioniert nur mit CyberGraphX!

SCREENMODEREQ Schaltet den Screenmode-Requester an. Sobald der Bildschirm zum Abspielen geöffnet werden soll, werden Sie nach einem Bildschirmmodus gefragt. Dieser Requester erscheint dann jedesmal, sobald CyberQT einen Bildschirm öffnen will.

FORCE24 Erzwingt das Abspielen von 16 Bit-Animation auf 24 Bit-Bildschirmen. Dadurch kann die Abspielgeschwindigkeit vielleicht etwas gesteigert werden.

HINWEIS:

Falls CyberQT auf einem System ohne CyberGraphX läuft ist diese Option völlig sinnlos und wird ignoriert. Diese Option

hat nur Auswirkungen auf 16 Bit-Animationen.
Bei 8 Bit-Animation wird sie ignoriert.
Bei 24 Bit-Animation ist sie (natürlich) sinnlos.

GRAY/GREY Diese Option setzt eine Graustufen-Platten anstatt der normalen Farben. Das Abspielen kann hierdurch etwas schneller werden, weil weitaus weniger Daten verschoben werden müssen.

AGA Schaltet AGA-Unterstützung ein. Diese Option braucht normalerweise nie angegeben werden, da CyberQT selbstständig erkennt, ob sie eine Grafikkarte oder AGA besitzen. Sie ist eigentlich nur dazu gedacht, um das Abspielen mit den AGA-Routinen zu erzwingen. Wenn diese Option angegeben wird oder nur der AGA Chipsatz vorhanden ist, werden alle Bilder einer Animation auf 256 Farben heruntergerechnet.

HINWEIS:

Versprechen Sie sich von der AGA-Unterstützung nicht zuviel. AGA ist SEHR langsam, verglichen mit modernen Grafikkarten unter CyberGraphX. Und da es immer noch einige Fehler in der graphics.library von OS3.x gibt, kann es sein, daß beim Abspielen Darstellungsfehler entstehen.

HAM8 Mit dieser Option werden Truecolor-Animationen (16 Bit Farbtiefe oder mehr) auf einem HAM8 Bildschirm dargestellt. Dadurch können die einzelnen Bilder besser aussehen, aber die Abspielgeschwindigkeit wird auf jeden Fall geringer.

HINWEISE:

Diese Option funktioniert nur mit AGA!
Diese Option schließt die GRAY Option aus!

DITHER Diese Option schaltet "ordered dithering" bei Truecolor-Animation ein, anstatt simple Farbreduktion. Dadurch können die einzelnen Bilder besser aussehen, aber die Abspielgeschwindigkeit wird auf jeden Fall geringer. Hier ist eine Liste aller Kodierungen, die ordered dithering unterstützen.

HINWEISE:

Diese Option funktioniert nur mit AGA!
Diese Option schließt die GRAY Option aus!

BUFFERSIZE Diese Option setzt die Puffergröße, die für das asynchrone Lesen verwendet wird. Die asynchio.library allokiert zwei Puffer der angegebenen Größe! Voreingestellter Wert ist 64K pro Puffer. Der angegebene Wert muß zwischen 1 und 4096 liegen.

HINWEIS:

Dieser Wert wird als Anzahl von Blöcken von 1024 Bytes interpretiert. Eine Wert von 50 bedeutet also 51200 Bytes

und NICHT 50 Bytes!!

DELAY	CyberQT wartet vor dem eigentlichen Abspielen ein paar Sekunden. Dies ist sehr hilfreich, falls Sie einen Monitor haben, der etwas langsam synchronisiert. So verpassen Sie nicht die ersten paar Bilder der Animation. Voreingestellter Wert ist zwei Sekunden. Der angegebene Wert muß zwischen 0 und 10 liegen.
LOOP	Schaltet zyklisches Abspielen der Animation an. Normalerweise wird am Ende der Animation das Programm beendet.
MAXFPS	Hiermit wird die Animation so schnell wie möglich abgespielt. Real wird nur eine Verzögerung von 0 Sekunden pro Bild gesetzt, dadurch wird die Animation so schnell abgespielt wie der Rechner kann.
NOSOUND	Schaltet Tonwiedergabe aus, es wird nur noch der Videoteil der Animation angezeigt.
NOVIDEO	Schaltet das Anzeigen der Videoframes ab. Bei QuickTimes mit Ton wird dann nur noch dieser abgespielt. Falls die Animation auch keinen Ton hat, wird nichts abgespielt.
SKIP	Diese Option erlaubt das Überspringen von einzelnen Bildern, falls die Abspielgeschwindigkeit zu gering ist.
AUDIOPRELOAD	Diese Option schaltet das Dekodieren der Audiosamples vor dem Abspielen an. Falls Sie Animationen mit nichtverschachtelten Audio- und Videodaten und genug freien Speicher haben können Sie durch Setzen dieser Option große Sprünge auf der Festplatte vermeiden, die zu ungleichmäßigem Abspielen führen können. HINWEIS: Diese Option ist sehr speicherintensiv! Falls Sie nicht genug Speicher frei haben ist es unmöglich eine zu große Animation mit dieser Option abzuspielen. Für die dekodierten Samples wird jeglicher freier Speicher benutzt, CyberQT ist also nicht auf eine entsprechende Menge freien ChipRAMs angewiesen.
MAGNIFY	Mit dieser Option können Sie einen Skalierungsfaktor für das Abspielfenster angeben. Mögliche Werte sind Zahlen zwischen -4 und +4, wobei negative Zahlen eine Verkleinerung bewirken und positive Zahlen eine Vergrößerung. Voreingestellter Wert ist 1, was keine Skalierung bedeutet. Sie sollten keine höheren Werte als +2 benutzen, weil CyberGraphX bei Faktoren von +3 oder +4 wirklich sehr langsam wird. Werte von 0 und -1 bewirken ebenfalls keine Skalierung.

HINWEIS:

Diese Option funktioniert nur mit CyberGraphX und auf öffentlichen Truecolorbildschirmen! Wenn die Animation auf einem eigenen Bildschirm abgespielt wird hat diese Option keinen Effekt.

STATS	Mit dieser Option werden nach dem Abspielen einige Informationen über die Abspielgeschwindigkeit ausgegeben (z.B. die Anzahl der dargestellten Bilder, erwartete und reale Geschwindigkeit, etc)
QUIET	Hiermit werden sämtliche Meldungen abgeschaltet. Diese Option hat allerdings keinen Einfluß auf Fehlermeldungen!!
DEBUG	Schaltet den Debug-Modus an. In diesem Modus werden etliche Ausgaben getätigt, die in etwa erklären sollen, was CyberQT gerade macht.

Achtung!! Folgende Kombinationen von Optionen können nicht verwendet werden und führen dazu, daß eine Option abgeschaltet wird:

- AGA und PUBSCREEN (AGA wird abgeschaltet)
- HAM8 und PUBSCREEN (HAM8 wird abgeschaltet)
- HAM8 und GRAY (GRAY wird abgeschaltet)
- DITHER und GRAY (GRAY wird abgeschaltet)
- NOVIDEO und NOSOUND (NOVIDEO wird abgeschaltet)

Beispiele:

CyberQT blabla.mov

Hiermit wird die Animation blabla.mov inklusive Ton (falls vorhanden und unterstützt) abgespielt.

CyberQT blabla.mov nosound maxfps

Hiermit wird die Animation blabla.mov ohne Ton und bei maximaler Geschwindigkeit abgespielt.

CyberQT blubb.mov screenmodereq buffersize=128

Hiermit wird vor dem Abspielen der Animation blubb.mov nach einen Bildschirmmodus gefragt. Beim Lesen wird ein Puffer von 128K Größe benutzt.

CyberQT brabbel.mov delay=5 loop stats

Vor dem Abspielen der Animation brabbel.mov wird 5 Sekunden gewartet. Durch

die LOOP-Option wird die Animation am Ende wieder von vorne abgespielt. Das Abspielen kann durch drücken einer Maustaste, ESC oder CTRL-C abgebrochen werden. Hinterher werden einige statistische Daten ausgegeben.

```
CyberQT veryslow.mov skip quiet gray
```

Hiermit wird die Animation veryslow.mov ohne jegliche Meldungen in der Shell angezeigt. Falls der Rechner zu langsam ist werden einige Bilder übersprungen. Statt der normalen Farbpalette werden Graustufen verwendet.

```
CyberQT verysmall.mov magnify 2 pubscreen ThisOne
```

Hiermit wird die Animation verysmall.mov auf dem Bildschirm "ThisOne" dargestellt und zusätzlich noch um den Faktor 2 skaliert, also doppelt so groß wie normal dargestellt.

1.7 workbench

Die folgenden ToolTypes können für einen Start von der Workbench aus benutzt werden. Die ToolTypes können sowohl in CyberQTs Piktogramm als auch in den Piktogrammen der Animationen abgegeben werden:

WINDOW Hiermit wird angegeben, wie das Ausgabefenster aussehen soll. Fehlt diese Option, so wird "NIL:" verwendet (keine Ausgabe). Diese Option gibt es nur als ToolType.

PUBSCREEN	\	
SCREENMODEREQ		
FORCE24		
GRAY or GREY		
AGA		
HAM8		
DITHER		
BUFFERSIZE		
DELAY		Diese Optionen sind mit den Shell-Optionen identisch,
LOOP		
MAXFPS		bitte lesen Sie die Erklärungen dort.
NOSOUND		
NOVIDEO		
SKIP		
AUDIOPRELOAD		
MAGNIFY		
STATS		
QUIET		
DEBUG	/	

HINEIS:

Die DEBUG-Option macht nur in Verbindung mit der WINDOW-Option Sinn. Wenn Sie also DEBUG angeben, aber nicht WINDOW, dann werden Sie nichts von den Debuginformationen sehen.

1.8 dither

Diese Kodierungen unterstützen ordered dithering unter AGA:

Radius CinePak	(CVID)	16/24/32 Bit
Kodak Photo CD	(KPCD)	16/24/32 Bit
Microsoft Video 1	(MSVC)	16 Bit
Apple Video	(RPZA)	16 Bit
Component Video	(YUV2)	16/24/32 Bit
Intel Raw	(YUV9)	16/24/32 Bit

Alle anderen Kodierungen haben KEINE Unterstützung!

1.9 operation

CyberQT versucht die Animation zeitlich so exakt wie möglich abzuspielen, das hängt aber hauptsächlich von Ihrer Hardware ab. Schnellere Prozessoren (z.B. MC68040 oder MC68060) und schnellere Grafikkarten ergeben eine wesentlich flüssigere Darstellung.

Zum Beenden von CyberQT oder zum Abbrechen der derzeitigen Animation haben Sie folgende Möglichkeiten:

- eine Maustaste oder ESC drücken
dadurch wird das Abspielen der derzeitigen Animation abgebrochen und stellt die einzige Möglichkeit dar, das Abspielen zu beenden, falls die Option LOOP angegeben wurde.
Ein Abbruch durch Drücken einer Maustaste funktioniert nur, falls CyberQT auf einem eigenen Bildschirm läuft. Wenn Sie CyberQT auf einem Publicscreen laufen lassen haben die Maustasten keine Wirkung!!
- Drücken von CTRL-C
hierdurch wird CyberQT sofort und komplett beendet
- Senden eines Break an den CyberQT-Task mit Tools wie Scout oder XOper
das Gleiche wie Drücken von CTRL-C

Während des Abspielens können sie mit den F-Tasten die Abspielgeschwindigkeit ändern:

- F1 - so schnell wie möglich, entspricht der Option MAXFPS
 - F2 - 1000% der normalen Geschwindigkeit
 - F3 - 500% der normalen Geschwindigkeit
 - F4 - 300% der normalen Geschwindigkeit
 - F5 - 200% der normalen Geschwindigkeit
 - F6 - normale Geschwindigkeit, wie durch die Animation vorgegeben
 - F7 - 75% der normalen Geschwindigkeit
 - F8 - 50% der normalen Geschwindigkeit
 - F9 - 20% der normalen Geschwindigkeit
 - F10 - 10% der normalen Geschwindigkeit
-

Die reale Geschwindigkeit kann von der erwarteten abweichen, aber das hängt von ihrem Rechner ab. Je schneller der Rechner, desto genauer ist die Abspielgeschwindigkeit.

1.10 faq

F: Warum wird "Intel Indeo" nicht unterstützt? Sehr viele meiner QuickTimes haben diese Kodierung!

A: Intel verlangt \$5000 (oder noch mehr) für ein "non-disclosure agreement" und den Sourcecode zu ihrer "Indeo technology". Ich bin nur ein Student mit stark begrenztem Einkommen. Ich denke, Sie können sich jetzt vorstellen warum...

F: Ich habe da eine Animation, deren Video- oder Audiokompression CyberQT nicht erkennt. Kann man das ändern?

A: Wenn es nicht gerade Intel Indeo (siehe oben) ist, dann schicken Sie mir doch bitte diese Animation zu. Ich werde dann versuchen in der nächsten Version von CyberQT diese Kodierung zu unterstützen. Noch nicht unterstützte Kodierungen (die aber durchaus leicht zu implementieren sind) können Sie hier nachsehen.

F: Warum hört sich der Ton manchmal so schlecht und rumpelig an?

A: Entweder ist ihr Rechner nicht schnell genug, um die Bilder "pünktlich" zu dekodieren und deswegen hängt der Ton dementsprechend hinterher, oder CyberQT hat noch einen Fehler. Die Tonwiedergabe ist immer noch nicht perfekt. Sie sollten die Option SKIP ausprobieren, um "verspätete" Bilder zu überspringen, falls der Rechner zu langsam ist.

Wenn der Ton viel zu tief und/oder zu langsam sein sollte (Frauenstimmen hören sich dann z.B. wie Männerstimmen an), dann hat die Animation wahrscheinlich eine Audiofrequenz von mehr als 28kHz. Diese 28kHz sind normalerweise die Obergrenze der Customchips, falls Sie die normalen PAL- oder NTSC-Modi benutzen. In Verbindung mit den DoubleScan-Modi, wie DblPAL und DblNTSC, sind aber Frequenzen bis 56kHz möglich. Rufen Sie CyberQT deshalb einfach probierhalber mit der SCREENMODEREQ-Option auf und wählen Sie einen DoubleScan-Modus. Falls sich der Ton nicht ändern sollte hat CyberQT einen Fehler :(

F: Ich habe eine Animation, bei der mir CyberQT immer sagt, daß der ".rsrc fork" fehlt. Kann ich die trotzdem irgendwie abspielen?

A: QuickTime-Animationen kommen ursprünglich vom Apple Macintosh, dessen Filesystem eine Datei in einen data- und einen resource-fork aufteilt. Beim Kopieren dieser Animationen vergißt man oft den entsprechenden Resourcezweig der Animation mitzukopieren. Ohne diesen Teil kann die Animation allerdings nicht abgespielt werden, da sämtliche wichtigen Daten (Kodierung, Auflösung,

etc) in diesem Resourcezweig enthalten sind.

Vielleicht haben Sie auch nur vergessen den Resourcezweig entsprechend umzubenennen. Er muß entweder genauso heißen wie die Animation und im Verzeichnis ".resource" liegen, oder im gleichen Verzeichnis wie die Animation liegen und die Endung ".rsrc" oder ".rs" haben.

Ein Beispiel: Animationsname blafasel.mov.

Für den Resourcezweig kommen jetzt also folgende Dateien in Frage:

- .resource/blafasel.mov
- blafasel.mov.rsrc
- blafasel.mov.rs

F: CyberQT tut gar nichts und verursacht nur Abstürze!

A: Bitte schreiben Sie mir. Es scheint, als ob Sie einen Fehler in CyberQT gefunden haben.

F: CyberQT weigert sich auf meinem alten A500 mit 68000/7 zu laufen, den ich damals vor einigen Jahren gekauft habe!

A: <seufz>!!

F: CyberQT weigert sich auf meinem Super-200MHz-Pentium mit Win95 zu laufen! Sollte ich einen schnelleren Prozessor und mehr RAM kaufen??

A: Idioten sterben NIE aus :(

1.11 contact

Meine Adressen:

"snail mail" (bzw "gelbe Post"):

Thore Böckelmann

Entgelhof 11

D-32278 Kirchlegern Tel: +49-5744-1309 und +49-5744-1323

Deutschland

Thore Böckelmann

Stephanusstraße 82

D-33098 Paderborn Tel: +49-5251-730837

Deutschland

Electronic mail:

tboeckel@uni-paderborn.de

tboeckel@guardian.fido.de (bevorzugt)

FidoNet: 2:2432/230.15

AmigaNet: 39:170/410.15

HINWEIS:

Ich bevorzuge guardian.fido.de und meine Fido-Adresse für "Smalltalk" und Fehlermeldungen. Falls Sie mir Animationen zum Testen zuschicken wollen, dann schicken Sie die BITTE an tboeckel@uni-paderborn.de, weil ich an der Uni nahezu unbeschränkten Platz für EMail habe.

Die jeweils neueste Version von CyberQT gibt es per Filerequest unter dem Namen "CyberQT.lha" in der Blind Guardian BBS. Filerequest ist möglich unter folgenden Telefonnummern:

Line 1:	38400 - 64000	Elink 310	+49-5742-920340	(2:2432/231.0)
Line 2:	2400 - 28800	Elsa TQV	+49-5742-920341	(2:2432/230.0)
Line 3:	2400 - 19200	ZyXEL EG+	+49-5742-920342	(2:2432/232.0)

1.12 history

Tut mir leid, ich bin wirklich zu faul das hier auch noch zu übersetzen. Und sooo wichtig ist die Geschichte nun auch wieder nicht.

V1.0:

27-Feb-97 - first release on Aminet

V1.1:

- 28-Feb-97
 - fixed a bug in RPZA decoder with AGA that produced wrong pixels/blocks of pixels
 - fixed "too slow video playback" when video timescale is not equal to anim timescale
 - fixed delayed reaction when changing playback speed
 - 4-Mar-97
 - CVID depth 32 is now supported, too
 - 5-Mar-97
 - added support for IMA4 audio compression. Unfortunately I was not able to test IMA4's stereo part because of no animation of that kind
 - improved handling of multi-track QTs
 - 6-Mar-97
 - implemented SKIP option to skip delayed frames and keep audio playback synchronous.
 - 8-Mar-97
 - fixed sound loop bug when playing more than one animation at a time. Reported by Thomas Börkel and many other people.
 - improved error handling in "out of memory" situations
 - 14-Mar-97
 - added support for KPCD video compression. Thanks to Lars Weigl for his NASA animation.
 - 15-Mar-97
 - fixed bug that made playback stuck because of wrong calculated time offsets
-

- 16-Mar-97
 - fixed bug with wrong interpreted elst chunks
 - small speed improvement for KPCD
- 17-Mar-97
 - hopefully fixed last problems with screenmode selection.
eg: 240x300 animations should be shown on a 640x480 screen
instead of a 320x240 screen.
 - fixed some skipping problems with "keyframe-only" QTs
- 19-Mar-97
 - fixed problems with TWOS 16bit stereo sound. SamplesPerChunk entry was misinterpreted as "bytes per chunk" instead of real "samples per chunk", so not enough data were read.

V1.2:

- 22-Mar-97
 - fixed some problems with oversized audio samples, although audio playback may still stumble a bit (happened with trailer to "Aliens 2")
 - added support for PCM0 audio encoding
 - 23-Mar-97
 - window on public screen is now resizeable on special request by Thomas Börkel :)
 - 28-Mar-97
 - fixed bad bug in RAW32 decoder
 - added some frame size checks to avoid crashes with corrupted frames
 - fixed some problems with resource fork recognition
 - fixed some problems with multi video tracks and different encodings
 - BIG speed improvement when converting YUV -> RGB in grayscale mode
 - 29-Mar-97
 - fiddled around with depth selection of animation with mixed truecolor and colormapped encodings. I hope it is correct now.
 - 31-Mar-97
 - did many changes on timing calculations
 - audio samples are now buffered in a queue instead of playing them directly. Thus playback should not stumble anymore if audio and video data are interleaved.
 - 1-Apr-97
 - Sorry, no jokes happened :)
 - 2-Apr-97
 - animations with more audio frames than video frames per second are now played correctly again
 - animations with with very small audio samples (eg. one sample per frame instead of one sample per 5 frames) are now more often played correctly. Playback may still listen bad because of too much overhead with these very small samples.
 - decoding of mono audio samples sped up by using just one buffer instead of two. Two buffers are now used for stereo samples only.
 - 3-Apr-97
 - added option AUDIOPRELOAD, which enables decoding of all audio samples before playback, to keep playback smooth with slow haddisks or non-interleaved animations.
-

- 4-Apr-97
 - fixed bug with PCM0 decoder
 - finally made animations with old style STSC chunk work
- 7-Apr-97
 - CyberQT now handles samples of ANY size. There is no limit of 128K per sample for audio.device anymore.
- 8-Apr-97
 - fixed bug when animations with and without sound were played within one call of CyberQT (eg selected via requester)
- 9-Apr-97
 - fixed bug with RLE1 decoder, RLE4 decoder was called accidentally
 - fixed bug with wrong grayscale palette for CVID animations
- 10-Apr-97
 - fixed bug with TWOS 16bit stereo audio that occurred when reading sample sizes
 - changed FPS rate in stats from integers to floats to be more precise
- 14-Apr-97
 - added option MAGNIFY to allow scaling of the display window on public by factors of 1..4
- 15-Apr-97
 - CVID decoder produced wrong pixel blocks in grayscale mode. This is fixed now.
- 20-Apr-97
 - CyberQT got a new icon. Much thanks to Bastian Kinne for painting it.
- 22-Apr-97
 - added option DITHER for ordered dithering with AGA
- 25-Apr-97
 - optimized checks for corrupted files. Truncated files are not always rejected, but recognized.
- 2-May-97
 - did some optimizations for RPZA dithering, looks better now
- 3-May-97
 - hopefully fixed the last elst-chunk problems

1.13 bugs

Bekannte Fehler:

- Je mehr Animationen Sie mit einem Programmaufruf abspielen, desto mehr Speicher wird verbraucht. Um das zu verhindern müssen Sie CyberQT lediglich mit weniger Animationen starten.
- Animationen mit mehr als 28kHz Audiofrequenz werden zu langsam abgespielt, weil die Audiohardware (normalerweise) solche Frequenzen nicht beherrscht. Falls Sie einen AGA-Rechner oder einen ECS-Rechner mit Grafikkarte haben sollten Sie die DoubleScan-Modi aktivieren (DblNTSC, DblPAL, etc). Damit sind dann Frequenzen bis 56kHz möglich.

Falls Sie noch irgendwelche weiteren Fehler finden sollten oder sonstige Vorschläge haben, dann schreiben Sie mir doch.

Falls CyberQT bei Ihnen Enforcer-Hits verursachen sollten, dann schreiben Sie mir.

Bitte fügen Sie (wenn möglich) einen Mitschnitt von CyberQTs

Debug-Informationen (Option DEBUG) hinzu, falls sich irgendwelche Komplikationen während des Abspielens ergeben sollten.

1.14 todo

Dringendere Dinge:

- bessere Tonwiedergabe (eine scheinbar unendliche Geschichte)
- Unterstützung von AHI für Tonwiedergabe. Hiermit wird der Zugriff auf unterschiedliche Audiohardware viel einfacher.

Noch zu tun:

- mehr Video- und Audio-Kodierungen unterstützen
- Fehler finden

1.15 xanim

Die von CyberQT unterstützten Kodierungen basieren alle auf dem Sourcecode von Mark Podlipecs Programm XAnim. XAnim unterstützt noch eine Zahl weiterer Kodierungen, sowie AVI-Animationen.

Die Hauptunterschiede zwischen CyberQT und XAnim bezüglich QuickTime-Animationen sind:

- XAnim ist "mächtiger" und universeller als CyberQT.
- CyberQT benötigt entweder eine zu CyberGraphX kompatible Grafikkarte oder den AGA Chipsatz.

Ein mit gzip komprimiertes Archiv von Mark Podlipecs XAnim kann von

<http://www.portal.com/~podlipec/home.html> "The XAnim Home Page"

bezogen werden.

```
/*
 * xanim.c
 *
 * Copyright (C) 1990,1991,1992,1993,1994,1995,1996,1997 by Mark Podlipec.
 * All rights reserved.
 *
 * This software may be freely copied, modified and redistributed without
 * fee for non-commercial purposes provided that this copyright notice is
 * preserved intact on all copies and modified copies.
 *
 * There is no warranty or other guarantee of fitness of this software.
 * It is provided solely "as is". The author(s) disclaim(s) all
 * responsibility and liability with respect to this software's usage
 * or its effect upon hardware or computer systems.
 */
```

*/

1.16 argue

Es gibt leider nur eine englische Anleitung zu Argue, und ich will Thorsten Stocksmeier nicht die ganze Übersetzungsarbeit abnehmen :)

It was around 1992 when Commodore released their new Amiga OS 2.0. With this, there were amazing changes for developers and users. All looked a bit more professional, and a lot of things were just easy and better to handle than in former times.

Earlier, developers had to write their own argument reading system. Often it was really unpractically and difficult to understand.

The guys at Commodore knew that and thought about a new standard for argument parsing to avoid confusion about all that. What they finally got was `ReadArgs()`, a system function that parses arguments automatically.

Developers now only had to write a template to specify, what arguments they would like to have. A template looks like this: `FILE/A,SWITCH/S...`

From now on, all the users could have a look at this template by adding a question mark to the program's name to execute.

But all in all, there was a problem. Folks still had to go "down" into a shell and type in all the arguments by hand.

So there are still a lot of people that write external interfaces for a specific tool. Some of them are even shareware!

This was really annoying as there was no tool that could manage ALL tools.

In early 1996 I developed a GUI layout system called NiceGUI. It was crap, but on this way I created the first version of Argue.

Argue's job was and is to read other tool's argument templates and prepare a nice user interface where the user can decide what he would like to have as arguments. Argue 0.3 was quite bad, but it was the first basis for further development.

Some months later I invented how to write MUI applications. It was very easy, and I implemented a new version of Argue with it. This was called Argue 0.6 and released to some BBSs here in Germany.

From then on Argue made giant steps towards user friendliness and efficiency. New features were added in masses, and now, at the time of Argue 1.0, there is a (near ;) complete interface creation system.

1.17 distribution

CyberQT ist Copyright ©1996-1997 by Thore Böckelmann.

CyberQT darf frei vertrieben werden, solange die folgenden Vereinbarungen eingehalten werden:

- alle Dateien müssen zusammen weitergegeben werden
- keine Datei darf verändert oder gepackt (z.B. mit Powerpacker) werden
- der einzige legale Verbreitungsweg ist das mit LhA gepackte Originalarchiv

1.18 credits

CyberQT wurde von Thore Böckelmann mit Hilfe von Amiga Oberon 3.10 und GCC 2.7.2 geschrieben.

Dank an Matthias Greim für seine "Peter Gabriel" CD-ROM mit CVID Animationen.

Dank an Peter Kunath für den Sourcecode seines "DrWho genie" für chunky->planar Konvertierung.

Dank an all die anderen Leute, die mir Animationen zum Testen und Weiterentwickeln von CyberQT geschickt haben.

Dank an Bastian Kinne (kinne@zfn.uni-bremen.de) für sein neues Programmicon.

Das Programm XAnim wurde von Mark Podlipec geschrieben. XAnim6 auf dem Amiga ist eine Portierung von Terje Pedersen. XAnim ist ©1990-1997 by Mark Podlipec.

Das Program QT wurde von Marcus Comstedt geschrieben. QT ist ©1996-1997 by Marcus Comstedt.

Die asyncio.library wurde von Martin Taillefer, Magnus Holmgren und Olaf Barthel geschrieben.

Argue ist ©1996 by Thorsten Stocksmeier

1.19 disclaimer

Hier war ich ebenfalls zu faul zum Übersetzen. Der englische Text ist so passend, daß man ihn auch eigentlich gar nicht übersetzen braucht.

No warranty, either express or implied, is made with respect to the fitness or merchantability of CyberQT.

Thore Böckelmann (referred to as "the author"), reserves the right to not develop any future versions of CyberQT.

The author will try to make a good faith attempt at correcting any problems if

any are discovered, but is in no way required, nor bound to correct them.

The author neither assumes nor accepts any responsibility for the use or misuse of these programs. He will also not be held liable for damages or any compensation beyond the original registration fee due to loss of profit or any other damages arising out of the use, or inability to use this program.

The author will not be liable for any damage arising from the failure of this program to perform as described, or any destruction of other programs or data residing on a system attempting to run the programs.

The user of this program uses it at his or her own risk.
