

# QL Today

## DEUTSCH

Jahrgang 5  
Ausgabe 1  
Mai/Juni  
2000

ISSN 1432-5446

Das Magazin über QL, QDOS,  
Sinclair Computer, SMSQ...

help Preview Font quit

Arial

Athletes

Atlas Overwate Normal

Aurora Bold

text for preview :  
AaBbCcDdEeFf

Font preview :

AaBbCcDdEeFf

fontutils|quit

pf b2pff

FontShow

FontPreview

do help ProWesS File Search quit Zzz

Search String QL

Extensions .txt Not

Directory WIN1 data

Tree	First Occurrence	Binary
Spaces	Word	Case Dependant
QLUSERS.TXT : line 1 :	QL Email Users Database Update - 24/12/97	
QLUSERS.TXT : line 65 :	Daglish, Neil <qltwirl@makai.vuw.ac.nz> Ne	
QLUSERS.TXT : line 298 :	Terkelsen, Jrgen <qlrjt@dk-online.dk> Denn	
IMSlist.txt : line 24 :	39 SMSQ A Colour IMS: SMSQ Updates (ATARI QL Cole	

wake do help PFlist quit Zzz

Printer \$N \$D printed with PFlist

tab distance 4

fontsize 10

Landscape

font Courier

device Two Columns

driver default driver

extensions Not

directory WIN1 data <- Tree All

EV/QBR.txt	delf_bas	IMSlist.txt	new_data_base_dba	RomDisq.ldp
Cardlist.txt	EindSO.198.txt	laptops_ldp	Opening_lib	SCART_ldp
cfg_bin	Eindhoven_prices_tab	mplate_ldp	postcodes.txt	style_aba
ChessClock.exe	Envelope_ldp	mf_scores	printer_dat	style.dbf
cologo_ldp	Envelope_ldp	monitors_T91	Q40Have.txt	style_def
colrep_ldp	gnugo.dat			
default_lay	IO PTR			

help ProW

Back Load Reload Er

Table : General ProWesS documentation

### ProWesS documentat

PROGS, Professional & Graphical Software  
Dr. Frans Hemerijckxlaan 13 /1  
2650 Edogem  
BELGIUM  
tel : +32 (0)31 457 84 88 fax : +32 (0)31 458 62 07 e-mail : joadrlm@club.innet.be  
www : http://www.club.innet.be/~year2827

wake do help Pwfile quit Zzz

action : Fileinfo link include subtree

sort method : alphabetic Commands

extensions : Not

directory : WIN1\_pws\_ <- Tree All

destination : RAM1\_ <- Exchange

file :

all : free on source 41MB on dest 1352kB

> doc	> mine	> pw
> ext	> pf	document1_parg
> fnt	> prg	

Mit  
ProWesS  
Cover  
Disk!

- Introduction
- What is ProWesS
- ProWesS manual
- Disclaimer & Copyright
- Present, Past and Future
- Installation

# Inhalt

- 2 Kleinanzeigen
- 2 Impressum
- 3 Editorial
- 4 Von Netz-Echos und dem Surfen mit schwarzen Kisten  
*J. Dent*  
*übersetzt von J. Merz*
- 5 Programmieren mit den C - Erweiterungen von EASYPTR (1Teil)  
*Frank Dibowski*
- 9 Zusammenfassung QL Today Volume 4, Issue 5 Jan./Febr. 2000  
*Wolfgang Uhlig*
- 11 Werbung Jochen Merz Software
- 12 Programmieren mit Hilfe des C68-Compilers und "kleiner" Hilfsprogramme!  
*Frank Dibowski*
- 15 QL-Bilder  
*Dietrich Buder*
- 16 Die Cover Disk - ProWesS
- 17 Neues Spiel: The Wall  
*Jochen Merz*
- RS QL 2000 Treffen

## Kleinanzeigen

Da QL Today eine der wichtigsten Quellen für QL-Neuigkeiten ist, werben auch die meisten QL-Händler hierin. Nun stellt sich natürlich die Frage, warum nur QL-Händler inserieren dürfen, warum nicht auch QL-User, die ihre eigenen Programme, Hardware oder Entwicklungen verkaufen möchten.

Anfangs konnte man selbstgeschriebene Software von Clubs vertreiben lassen, beispielsweise dem deutschen QL User Club. Aber warum soll man es denn nicht selbst vermarkten können? Es gibt hier keinen Unterschied zwischen privaten und kommerziellen Anzeigen, nur mit dem QL und Drumherum sollte es schon etwas zu tun haben.

Bis zu 50 Worte im englischen oder deutschen Teil kosten DM 5,- (oder 3 Internationale Antwortscheine), bis zu 100 Worte kosten DM 10,- (oder 6 Internationale Antwortscheine). Soll die Anzeige im deutschen und englischen Teil erscheinen, ist der Preis zu verdoppeln.

## QL Today DEUTSCH

ISSN 1432-5446

### Herausgeber:

Jochen Merz Software    Tel.    +49 203 502011  
Im stillen Winkel 12    Fax    +49 203 502012  
47169 Duisburg        Box1   +49 203 502013  
Deutschland            Box2   +49 203 502014  
Email: QLToday@j-m-s.de

**QL Today** erscheint alle zwei Monate, Erscheinungsdatum der ersten Ausgabe ist der 15. Mai. Das Abo beginnt mit der aktuellen Ausgabe zum Zeitpunkt der Bestellung. Preisinformationen und Antragsformulare sind bei Jochen Merz Software erhältlich.

Ihre Kommentare, Vorschläge und Artikel sind herzlich willkommen. SIE machen **QL Today** möglich. Wir verbessern das Magazin wo immer möglich, um Ihren Vorstellungen gerecht zu werden. Artikel sollten auf 3,5" Diskette (DD oder HD) eingeschickt werden. Das Format sollte ASCII, Quill oder Text87 (Druckertreiber angeben!) sein. Bilder sollten im \_SCR-Format geschickt werden, GIF und TIF ist auch möglich. BITTE senden Sie auch einen Ausdruck der Bilder. Wenn ein Bild an einer bestimmten Stelle plaziert werden soll, geben Sie es bitte auch an. Natürlich können auch alle Bilder/Artikel auf elektronischem Weg übersandt werden, also in die Box oder per E-Mail.

### Redaktionsschluß für Artikel und Werbung:

Ausgabe 1:	30. April
Ausgabe 2:	30. Juni
Ausgabe 3:	30. August
Ausgabe 4:	30. Oktober
Ausgabe 5:	30. Dezember
Ausgabe 6:	28. Februar

**QL Today** behält sich vor, eingeschicktes Material nicht zu veröffentlichen. **QL Today** ist unter keinen Umständen für die Richtigkeit der abgedruckten Artikel und Programmen haftbar, ebenso nicht für aus fehlerhaftem Material hervorgerufene Datenverluste, Unbenutzbarkeit oder ähnliche Probleme, die aus Artikeln in **QL Today** herrühren könnten. Die Meinung in diesem Magazin entspricht der des jeweiligen Autors und nicht notwendigerweise der des Herausgebers.

Dieses Magazin unterliegt dem Copyright und jegliches hierin veröffentlichte Material darf nicht ohne schriftliche Erlaubnis von **QL Today** reproduziert, übersetzt oder sonstwie verbreitet werden. Allen Copyrights und Trademarks wird hiermit Rechnung getragen.

Ja, liebe Leser, die Ankündigung vom letzten QL Today Editorial ist Realität geworden: Mein Q40 steht jetzt hier! Schnell habe ich ihn angeströmt und mit SERNET alle Dateien draufgespielt. Das hat auch bei 57600 Baud relativ lange gedauert (sind ja auch über 200MB an Daten). Währenddessen habe ich ein paar Tests gemacht (ist ja das erste Mal, daß ich einen Q40 zur beliebigen Verfügung habe) und mit den Farben herumgespielt. Toll!

Zur Geschwindigkeit: Für die meisten Anwendungen ist ja mein TT schnell genug (SMSQ ist glücklicherweise ein sehr effizientes Betriebssystem und die Anwendungen sind meistens auch entsprechend programmiert), doch beim Fax-Decodieren könnte etwas mehr Geschwindigkeit nicht schaden. Also schnell mal QFAX Fax-Viewer gestartet (QFV) und ein Fax geladen, von dem ich wußte, daß es recht komplex war und lange zum Decodieren braucht (Faxe sind in der Regel komprimiert, da sie nur 9600 oder, wenn man Glück hat, 14400 Baud zur Übertragung nutzen können. Ich muß sagen, ich bin begeistert! Ein echter, deutlich wahrnehmbarer Geschwindigkeitsprung!

An meinem BOOT brauchte ich auch kaum etwas ändern - andere Bildschirmauflösung und mehr Farben einstellen, und das war's im Prinzip - danach startete der Rechner mein (doch recht komplexes BOOT ganz durch). Wenn Interesse besteht, veröffentliche ich mein aktuelles BOOT in der nächsten Ausgabe - viel hat sich seit dem Druck in der ersten QL-Today Ausgabe verändert!

Ich will aber jetzt hier nicht weiter von Q40 schwärmen, es gibt ja schließlich noch andere Neuigkeiten.

Lange erwartet und doch endlich fertig und erhältlich: die Farbtreiber. Mit Q40 hat's ja angefangen - und als nächstes kam die QXL dran. Neues internes Interface mit besseren Möglichkeiten und schnellerer interner Übertragung sowie (wie Q40) 65536 verschiedene Farben! Details zum Upgrade-Preis und -Vor-

gehen gibt's auf der JMS-Seite. Tony Tebby wird vermutlich noch einen 256-Farb-Modus einbauen da er die Geschwindigkeit erhöhen wird - Sie haben dann die freie Wahl zwischen mehr Farben und mehr Geschwindigkeit.

Marcel hat die neuen Sources von Tony Tebby bereits in der Mache und es ist nur noch eine Frage der (seiner) Zeit, bis auch QPC2 mit viel mehr Farben erhältlich sein wird. Vielleicht gibt's ja noch eine Last-Minute Neuigkeit diesbezüglich kurz bevor das Heft in den Druck geht.

Ich fasse mich diesmal etwas kürzer - der Mai hat schon angefangen und in diesem Monat stehen sowohl Eindhoven als auch das USA-Treffen an, daher muß ich mich sputen.

Noch etwas Interessantes: Wie versprochen, sollte es ja in diesem Heft den Artikel über QL und Internet geben. Ich wollte ihn ja selber übersetzen, aber ich dachte mir: Laß es doch mal ein Programm versuchen! Nun ja, das Ergebnis war katastrophal, um nicht zu sagen, unbrauchbar. Und das, obschon ich "Computer" als Text-Typ angegeben habe, was die Übersetzung verbessern sollte (naja, ich habe nicht geschaut, was sonst herausgekommen wäre).

Die Nachbearbeitung, die notwendig war, hat vermutlich ebenso lange gedauert (oder länger) als es direkt selber zu übersetzen. Übersetzungen dieser Art sind mit Vorsicht zu genießen. Auch wer denkt, mit solchen Programmen einen "Überblick" zu bekommen weil er die Sprache nicht spricht, sollte vorsichtig sein - teilweise kommt totaler Unsinn oder das Gegenteil der Aussage heraus. Naja, wer's liest wird merken: Das ist nicht Jochen's Stil, aber einen Versuch war es wert.

Der versprochene EPROM-Artikel vom Frank Dibowski wird in der nächsten Ausgabe kommen, Frank hatte zuerst etwas anderes vorbereitet (in dieser Ausgabe).

Bis zum nächsten Mal verbleibe ich mit den besten Grüßen

*Jochen Merz*

# Von Netz-Echos und dem Surfen mit schwarzen Kisten

J. Dent - übersetzt von J. Merz

Im November 1999 hallte - unbemerkt vom Rest der Welt - ein Datagramm durch das Internet, das die Mitteilung "QL Forever" enthielt. Ein kleiner Schritt für ein Datagramm, aber einen Quantensprung für die Menschheit. Nun, vielleicht nicht unbedingt für die Ganze Menschheit, aber wenigstens für die Art von Mann (oder Frau was das anbelangt) der seinen kleinen schwarzen Kasten den Pentathlons (oder wie auch immer sie heutzutage genannt werden) vorzieht.

Ich beziehe mich hier auf die ersten Test-Übertragungen, die den TCP/IP Stapel benutzen, den ich für den QL und seine "Nachfolger" entwickelte.

Uns wird erzählt, daß im 21. Jahrhundert praktisch alles im Netz verbunden sein wird: Telefonieren, Fernsehen,

Zentral-Heizung, Herd, Kühlschrank, Waschmaschine oder was auch immer Sie wünschen, bzw.

was sich auf Ihren Wunsch hin drehen, auf- oder ab- oder einstellen soll... Es wäre traurig, wenn die Liste den QL nicht einschloße.

Der QL wird nicht ausgelassen. Die Prüfungen zeigen, daß Kommunikation über das Internet möglich ist. Das Protokoll, das für die Übertragung benutzt wurde, wird UDP genannt

(Abkürzung für Benutzer-Datagramm-Protokoll). Das Datagramm ist das ganze Paket von Daten, das durch das Internet übertragen wird. Es schließt die originalen Benutzer-Daten "QL Forever" in diesem Fall ein und enthält IP und UDP Kopf-Informationen.

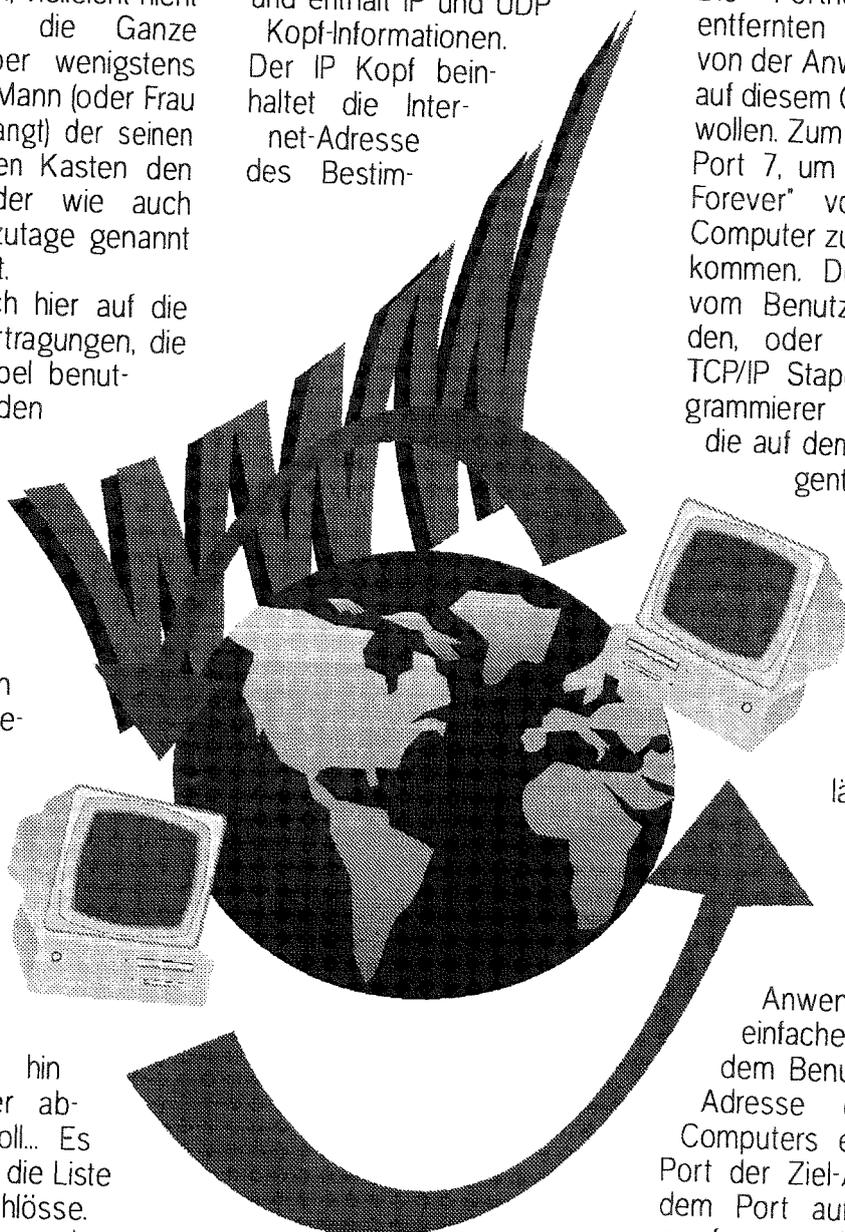
Der IP Kopf beinhaltet die Internet-Adresse des Bestimm-

werden. Die Adresse vom QL wird vom ISP zugeteilt (Internet-Service-Provider, z.B. AOL oder T-Online) und ist vielleicht jedes Mal eine andere, wenn Sie sich ins Internet einwählen. Der IP Kopf zeigt auch die Länge der Daten, die transportiert werden. Der UDP Kopf enthält die Nummer des Ports (logischer Anschluß) auf dem entfernten Computer und der Nummer des Ports auf dem QL. Die Portnummer auf dem entfernten Computer hängt von der Anwendung ab, die Sie auf diesem Computer benutzen wollen. Zum Beispiel nutzen Sie Port 7, um die Mitteilung "QL Forever" von dem anderen Computer zum QL zurückzubekommen. Der QL Port kann vom Benutzer zugeteilt werden, oder automatisch vom TCP/IP Stapel oder vom Programmierer der Anwendung, die auf dem QL läuft. Sein eigentlicher Wert ist

nicht wichtig so lange er einmalig ist, d.h. kein anderes Programm, das im Moment auf den QL läuft, benutzt diese Port-Nummer.

Für die oben erwähnte Prüfung ist die Anwendung ein einfaches Programm, das dem Benutzer erlaubt, die Adresse des entfernten Computers einzugeben, den Port der Ziel-Anwendung und dem Port auf dem der QL empfangen wird. Ein anderes Empfänger-Programm erwartet die gleiche Port-Nummer und wird jede UDP Mitteilung drucken, die vom Internet erhalten wird, die für diesen Port bestimmt ist. Eine Mitteilung,

mungs-Computers und vom Ziel-Rechner, in diesem Fall der QL. Die Adresse der Quelle muß vom Benutzer gewußt



wie "QL Forever" (was sonst) kann jetzt im ersten QL Programm eingegeben werden und - man mag es kaum glauben - es erscheint im Fenster des Empfänger-Programms darunter. Es ist zum örtlichen ISP geschickt worden, sein Computer hat die Adresse überprüft und hat es auf die richtige Art ins Internet geschickt und vielleicht wurde es von einem Netzwerk in ein anderes übertragen, bis es beim Bestimmungs-Computer ankommt. Die ECHO-Anwendung auf diesem Computer hat das Datagramm erhalten, kopiert "QL Forever" in ein neues Datagramm, tauscht den Bestimmungsort und die Quelle, tauscht die Portnummern und schickte es in die andere Richtung ab.

Dies ist die einfachste Art von Protokoll für Anwendung auf dem Internet. Es läuft jetzt auf dem QL, und es ist ein wich-

tiges Protokoll, aber das wichtigste Protokoll ist TCP. TCP steht für Transmission Control Protocol (Übertragungssteuerungs-Protokoll). Wenn Sie ein UDP Datagramm auf das Internet schicken, wissen Sie nicht, ob es bei der anderen Seite ankommt. Wenn Sie es dann über TCP schicken, wird TCP sicherstellen, daß es ankommt und es wiederholt die Sendung, wenn notwendig. Wenn alle TCP-Versuche die Daten zu übertragen scheitern, werden Sie gewarnt.

TCP wird für das Übertragen von Dateien, Webseiten und für alle Daten genutzt, bei denen Integrität wichtiger als Geschwindigkeit ist. UDP kann für Anwendungen benutzt werden, die sich mit der Frage von Integrität selber beschäftigen. Zum Beispiel in der ECHO-Anwendung bestätigt die Ankunft des Echos seine erfolgreiche Übertragung zurück beim Ursprung.

Anwendungen, die die Übertragung von Live-Musik oder Bildern verlangen, werden gut von UDP's Einfachheit und Geschwindigkeit bedient. Sie gewinnen keinen Vorteil von der Zuverlässigkeit von TCP. Ein wiederholter Block solcher Daten wird zu spät ankommen und wird nicht mehr zu verwenden sein.

Ich arbeite jetzt an TCP. Ungefähr 60% der Arbeit ist gemacht. Es gibt Anwenderprogramm, die auf UQLX (QL-Emulator für UNIX Computer) laufen, die den TCP/IP Stapel vom UNIX Betriebssystem, auf dem der Emulator läuft, benutzen. Diese schließen Lynx (ein Browser für's World Wide Web) und E-Mail-Programme ein.

Bereiten Sie sich deshalb darauf vor, ins 21. Jahrhundert zu springen und mit Ihrer schwarzen Kiste zum Surfer zu werden.

**Wir hoffen auf baldige Fort-**

---

## Programmieren mit den C - Erweiterungen von EASYPTR (1. Teil)

Frank Dibowski

Auf Grund des Artikels von Wolfgang Uhlig dachte ich mir, es wäre doch auch mal ganz praktisch zu sehen, wie einfach so ein Programm unter C erstellt werden kann. Als Basis nehme ich das Menü, das Wolfgang in seinem Artikel (QL-Today J.4/A.5) bearbeitet hat. Somit kann auch (hoffentlich) jeder meinen Erklärungen folgen, auch wenn er bisher noch nichts dergleichen in C gemacht hat.

Was brauche ich nun, um ein Programm mit Pointeroberfläche in C zu programmieren?

Einen C-Compiler, ich verwende nur den C68-Compiler, die Me-

nüs, die mit EasyMenu erstellt worden sind, die Sprites, die mit EasySprite erstellt wurden und dann natürlich die C-Library Routines (EasyPTR 3).

Zu dem C68-Compiler kann ich leider nicht mehr sagen als das er der Beste ist, der mir für den QL in die Finger geraten ist und irgendwie auch der Billigste. Was noch eine Untertreibung ist, da er umsonst ist!

Bei den Menüs und den Sprites muss man noch erwähnen, dass sie in einem anderen Format abgespeichert werden müssen als unter Basic. Beim Speichern eines Menüs oder Sprite hat man zum Beispiel noch die Wahl es in dem

Format "S-ROFF (\_rel)" abzuspeichern. Das ist für die Programmierung unter C sehr wichtig! Dadurch wird ermöglicht, dass man z.B. ein Sprite direkt mit Namen ansprechen kann, ohne das man genau weiß, wo es jetzt genau liegt. Das wird im Programm hoffentlich deutlich! Dem Menü habe ich den Namen Projekt gegeben und die zwei Sprites, die Wolfgang benutzt hat, habe ich Häkchen und Schatten genannt.

Wie sieht nun das dazu gehörige Programm aus?

Da ich in C die Möglichkeit habe "Modular" zu programmieren, werde ich das auch bei diesem Beispiel anwenden. Dadurch habe ich dann nicht nur ein Programm, sondern mehrere Kleine. Das ist in sofern ganz praktisch, weil man so kleinere Programme in

andere einfügen kann ohne sie neu zu schreiben. Insgesamt besteht das Programm nun aus sechs Dateien:

Projekt\_c  
 Teste\_PE\_c  
 Projekt\_mke  
 Projekt\_men\_rel  
 Schatten\_spr\_rel  
 Haeckchen\_spr\_rel

Zu den letzten drei Dateien brauche ich nichts weiter zu sagen, sie werden durch Easy-Menu bzw. EasySprite erzeugt! (Siehe auch QLToday J.4/A.5) Die Datei Projekt\_mke übersetzt die ersten zwei und verbindet alle Dateien miteinander. Dazu später mehr.

Die Datei Projekt\_c: (Listing 1)

In den ersten acht Zeilen wird dem System mitgeteilt, mit welchen Befehlen es arbeiten muss, das heißt welche Befehle es kann. Dabei sind die Include-Dateien "easy.h", "easywsa.h", "easywwd.h" und "easywdd.h" sehr wichtig, da mit ihnen das Ansprechen des Pointerenvironments überhaupt erst ermöglicht wird. Danach werden zwei Funktionen deklariert! Damit weiß der Compiler wieviele Funktionen im Programm sind und wieviele Parameter jede Funktion hat! In diesem Fall werden keine Parameter übergeben. Nun werden einige Variablen deklariert! Bei den Variablen, bei denen "extern" vor steht, geht der Compiler davon aus, das wir diese Variablen später noch in Form einer oder mehrere Dateien beifügen werden. (Was wir ja auch machen) Danach beginnt die Funktion "main". Sie bildet das eigentliche Hauptprogramm. Zuerst wird ein Fenster geöffnet und die Outline gesetzt. Danach wird getestet, ob überhaupt das Pointerenvironment existiert. Dazu wird die Funktion "test\_pe" aufgerufen. (Listing 2) (Sie ist in allen meinen Program-

## Listing 1: Projekt\_c

```

/* CFun include */
#include <stdio.h>
#include <qdos.h>
#include <easy.h>
#include <easywsa.h>
#include <easywwd.h>
#include <easywdd.h>

/* CFun Prototypen */
void main(void);
void setup_and_call_projekt (void);

/* CFun Globale_Variablen */
char _conname = NULL;
char _prog_name[] = "QLToday-Projekt"; /* Program name */
char _prog_vers[] = "0.00"; /* Program version */

long primary;
long wman, error;
short result[10];

extern long projekt; /* Das ist der Name des erstellten Menüs */
struct wwd *projekt_wwd;

extern long schatten;
extern long haekchen;

/* CFun main */
void main(void){
  primary = io_open ("con",0);
  eu_pwmak(&primary);
  if (test_pe(primary,&wman)){
    setup_and_call_projekt();
  }
}

/* CFun setup_and_call_projekt */
void setup_and_call_projekt (void){

struct qdosstring1 {
  short length;
  char strchr[12];
} text1={8,"mit Piep"};

struct qdosstring1 text2 = {9,"ohne Piep"};

short xsize=0,yssize=0;
short angekreuzt=2;
char status[];

  em_setup (&xsize,&ysize,&primary,&projekt,&statusadr,&resultadr,
&projekt_wwd);

/*  wm_stiob(0, 1,&text1,wman,projekt_wwd);
  wm_stiob(1, 1,&text2,wman,projekt_wwd); */

  em_wdraw(-1,-1,projekt_wwd);

  while(!(error=em_rptr( projekt_wwd,0 ))){
    switch (ret_type){
      case -2 : /* Ein Ereignis ist aufgetreten */
        break;

```

```

case -1 : /* Ein Menüpunkt ist angewählt */
switch (item_no){
case 0 : /* Taste: ESC T : ENDE */
if (angekreuzt==1) do_sound(15000,50,5,0,2000,15,0,0);
em_clear(projekt_wwd);
break;
case 1 : /* Taste: CF1 S : SCHLAFEN */
break;
case 2 : /* Taste: CF4 S : SCHIEBEN */
break;
case 3 : /* Taste: E T : Mpkt eins */
if (angekreuzt==1) beep(1000,10);
break;
case 4 : /* Taste: Z T : Mpkt zwei */
if (angekreuzt==1) beep(1000,10);
break;
case 5 : /* Taste: D T : Mpkt drei */
if (angekreuzt==1) beep(1000,10);
break;
case 6 : /* Taste: V T : Mpkt vier */
wm_stlob(0, 1,&text1,wman,projekt_wwd);
wm_stlob(1, 1,&text2,wman,projekt_wwd);
em_idraw(-3,projekt_wwd);
if (angekreuzt==1) beep(1000,10);
break;
case 7 : /* Taste: Keine T : */
if (angekreuzt!=1){ /* Menüitem ändern */
wm_stlob( 7,&haekchen,wman,projekt_wwd);
wm_stlob( 8,&schatten,wman,projekt_wwd);
status[ litem + item_no+1 ] = refresh ;
angekreuzt=1;
}
status[ litem + item_no ] = available + refresh ;
break;
case 8 : /* Taste: Keine T : */
if (angekreuzt!=0){ /* Menüitem ändern */
wm_stlob( 8,&haekchen,wman,projekt_wwd);
wm_stlob( 7,&schatten,wman,projekt_wwd);
status[ litem + item_no-1 ] = refresh ;
angekreuzt=0;
}
status[ litem + item_no ] = available + refresh ;
break;
};
em_idraw(-1,projekt_wwd);
break;
}
}
}

```

## Listing 2: Teste\_PE\_c

```

/**/
/*          CFun Teste Pointer Interface          */
/**/

#include <stdio.h>
#include <qdos.h>
#include <easy.h>
#include <easywsa.h>
#include <easywwd.h>
#include <easywdd.h>

/* CFun Prototypen */

long test_pe(long my_primary, long *my_wman);

/* Prüft ob das PE vorhanden ist */
/* gibt die Versionsnummer zurück wenn ja */

```

men enthalten und deshalb auch als eigene Datei beige-fügt!) Sollte es nicht existieren, wird die entsprechende Fehlermeldung ausgegeben, die ich in der Text-Variablen "error\_msg" gespeichert habe und das Programm wird beendet. Wenn (hoffentlich!) doch, wird die Funktion "setup\_and\_call\_projekt" aufgerufen. Durch den Befehl "em\_setup" wird das Menü in eine Form gebracht mit der man unter C arbeiten kann. Mit dem Befehl em\_wdraw wird das Menü nun gezeichnet. Die ersten zwei Parameter sind Positionsangaben. Mit "-1,-1" wird das Menü relativ zum Mauszeiger gezeichnet. Wenn man das Menü fest positionieren will, muss man die dementsprechenden Positionsangaben machen. Zum Beispiel "em\_wdraw(100,59,projekt\_wwd)" Nun folgt eine Schleife, in der die entsprechenden Aktionen abfragt werden, die im Menü mit Hilfe der Maus oder Tastatur ausgeführt werden. Der Befehl der für die Auswertung zuständig ist lautet "em\_rptr" und funktioniert ähnlich dem Befehl "MCALL" in Basic. Zuerst wird die Variable "ret\_type" abgefragt. Sie gibt an, ob ein Ereignis (-2), ein Menüpunkt (-1) oder ein Anwendungsfenster (0,1,2,... für das entsprechende Anwendungsfenster) angesprochen wurde. Ein Ereignis wäre zum Beispiel, das Fenster zu verschieben. Wird ein Menüpunkt angesprochen, wird über "item\_no" das entsprechende Menü-Item angesprochen. Im Basic wurden sie von -1 bis -9 angegeben. Hier werden sie nun von 0 bis 8 angegeben. Die Menüpunkte 1 und 2 sind nur proforma angegeben. Sie werden als Ereignis abgefragt und könnten hier weggelassen werden. Um nun, wie in dem Beispiel von Wolfgang, ein Häkchen zu setzen, muss das

mittels des Befehls "wm\_stlob" gemacht werden. Bei diesem Befehl gibt es aber leider noch ein kleines Problem. Das neue Objekt muss vom gleichen Typ sein, wie das vorherige. Ein Text-Objekt durch ein Sprite-Objekt zu ersetzen (oder umgekehrt) ist so leider nicht möglich. Darauf gehe ich aber in einem späteren Artikel näher ein. Mittels "status [..]" verändert man den Status eines Menüitems. Mit "refresh" wird nur dafür gesorgt, dass dieses Menüitem neu gezeichnet wird. Damit es auch neu gezeichnet wird, muss man noch den Befehl "em\_ldraw" anfügen. Beim Parameter "-1" wird nur das neu gezeichnet, was das Attribut "refresh" trägt. Mit dem Parameter "0" würde alles neu gezeichnet.

Um nun die Texte "mit Geräusch" und "ohne Geräusch" zu ändern, gibt man einfach zwischen "em\_setup" und "em\_wdraw" den Befehl

wm\_stlob(1,1,&text,wman, projekt\_wwd) ein! Vorher muss man noch einen QL-String definieren, was mittels

```
struct qdosstring {
    short length;
    char strchr[12];
}
text={8,"mit Piep"};
geschieht.
```

### Listing 3: Projekt\_mke

```
# Makefile for EasyC-Programs v1.00

# some short cuts

ASM = win2_C68_QMAC
ASMFLAGS = -NOLIST -NOSYM
CP = win2_C68_CP

size = 512

PRGN = ram1_Projekt_exe

# Utilities-Path

UPATH = win2_C68_Module_

# Source-Path

SPATH = win2_QLToday_
```

```
/* sonst eine QDOS Fehlermeldung */
/* Parameter : Hauptkanal */
/* Zeiger auf wman Vektor */
/* Aufruf: if(test_pe(primary,&wman)){ setup_and_call_...();} */

long test_pe (long my_primary, long *my_wman){
long my_error,my_vers;
char error_msg[250] = {"\n\n\n
Pointerenvironment not found! (PTR_GEN,WMAN)\n\n
Pointerenvironment nicht gefunden! (PTR_GEN, WMAN)\n\n\n\n\n\n\n\n\n\n
--> press any key,please <--\n\n
--> Bitte drücken Sie eine beliebige Taste <--"}
};
my_error = iop_pinf(-1,my_primary,&my_vers,my_wman);
if (my_error >= 0){
return(my_vers);
} else {
sd_clear(my_primary,-1);
io_sstrg(my_primary,-1,error_msg,230);
io_fbyte(my_primary,-1,error_msg);
return (0);
}
}
```

Oder man gibt diesen Befehl bei einem Menüpunkt ein. Dann muss man nur noch dafür sorgen, dass das Information-Window neu gezeichnet wird. Das geschieht mit dem Befehl "em\_ldraw". (Siehe Listing 1 ab "case 6") Was man bei der ganzen Sache nicht vergessen darf ist, dass hier immer bei Null angefangen wird zu zählen. Ausserdem sollte man tunlichst darauf achten, dass die Länge des vorherigen Textes nicht überschritten wird. Das könnte sonst zu komischen Effekten führen.

Nun noch etwas zu der Datei "Projekt\_mke". (Listing 3) Diese Datei erleichtert einem die Arbeit mit dem C68-Compiler sehr! In dieser Datei wird angegeben, wie das fertige Programm heißen soll, welche Dateien compiliert werden sollen, welche Dateien später zusammengesetzt werden sollen und .... Der Dateiname steht hinter der Variablen PRGN und kann jeder Zeit geändert werden. SPATH zeigt auf das Verzeichnis, in dem die Programm-sourcen liegen und sollte auf jedenfall angepasst werden. Ich kann ja nicht wissen, wo die Dateien abgelegt werden. Um nun das Programm zu übersetzen gibt man einfach in Basic

```
ex make;"-f win1_QLToday_
Projekt_mke"
ein (wenn es auf Floppy liegt natürlich WIN durch FLP ersetzen!). Der Rest geschieht dann von ganz alleine! (Hoffe ich!) Um nun das fertige Programm in ein anderes Verzeichnis zu kopieren, muss man nur
```

```
ex make;"copy -f win1_QLToday_
Projekt_mke"
eingeben. "copy" ist ein Makro, welches im Makefile "Projekt_mke" definiert ist. Solche Makros sind einfach zu
```

```

# Menuefinition

menus = ${SPATH}Projekt_men_rel \
        ${SPATH}Schatten_spr_rel \
        ${SPATH}Haekchen_spr_rel

# Die verwendeten Programmteile

objekte = ${SPATH}Teste_PE_o \
          ${SPATH}Projekt_o

# Die verwendeten Bibliotheken

biblio = -lPtrgen -lWMan -lEasymenC

mainprog : ${objekte}
            ${LD} -ms \
                -o${PRGN}\
                ${objekte}\
                ${menus}\
                ${biblio}\
                -bufp$(size)

clean :
        ${RM} -v -f ${SPATH}*_o ${SPATH}*_i ${SPATH}*_map

copy :
        ${CP} -v -i ${PRGN} ${SPATH}

_asm_o:
        ${ASM} $_asm ${ASMFLAGS} -BIN $_o

_asm_rel:
        ${ASM} $_asm ${ASMFLAGS} -BIN $_rel

```

erstellen und können sehr umfangreich sein, was einem die Arbeit schon sehr erleichtern kann.

So, für den ersten Teil hab' ich nun genug gesagt! Dass ich die Befehle nicht genauer erklärt habe hat zwei Gründe, für die ihr hoffentlich verständnis habt. Zum einen denke ich, dass das Listing so einfach gehalten ist, dass man ohne weiteres folgen kann und zum anderen möchte ich in der Hinsicht mal mit Albin Hessler sprechen, bevor ich hier eine komplette Anleitung zu seiner Erweiterung schreibe.

Solltet ihr noch Fragen haben oder irgendwelche Anregungen so könnt ihr mir jederzeit schreiben!

**Frank-Guido Dibowski**  
**Görreshof 17**  
**53347 Alfter**  
**E-Mail: uzsr37@uni-bonn.de**

---

## Zusammenfassung QL Today Volume 4, Issue 5 Jan./Febr. 2000

Wolfgang Uhlig

### Neuigkeiten

**Interessante Neuigkeiten von PROGS:** ProWess ist ab sofort für jedermann freigegeben, um allen QLern die Möglichkeit zu bieten, seine Fähigkeiten zu nutzen. Man kann es sich herunterladen von Joachim Auwers Webseite:

[www.triathlon98.com/Joachim/](http://www.triathlon98.com/Joachim/)

Auch für Q40 ist ProWess jetzt erhältlich.

**MINIVIEW** ist ein Textbetrachter für ASCII Texte von Dilwyn Jones. Man kann Texte laden, in ihnen scrollen und sie ausdrucken. Über Fileinfo aufzurufen. Es ist freeware und zu erhalten auf der Webseite:

[www.soft.net.uk/dj/index.html](http://www.soft.net.uk/dj/index.html)

Die Ausgabe 28.100 des uQLx Emulators ist schneller geworden, und läuft sehr stabil unter LINUX x86 2.0.35

ERGON Development hat Version 2.24 des Open World Grafik-Konvertier-Programms herausgebracht. Zusammen mit dem Spektrum-Emulator und Masterbasic 1.46 zu erhalten unter:  
[www.geocities.com/SiliconValley/Park/6533/davideeng.htm](http://www.geocities.com/SiliconValley/Park/6533/davideeng.htm)

### Q40 Neuigkeiten

**pqiv011** ist ein Bildbetrachter für den Q40. Er erfordert SMSQ/E 2.95. PCX, PNG, PIC und GIF Dateien können angeschaut werden.

Auch für Sounds kann man Dinge finden auf der Webseite: [www.q40.de](http://www.q40.de)

Auch bei **Jonathan Hudson** sind einige neue Sachen zu finden, wie die "sox" sound Konvertierung für QDOS, eine grafische Oberfläche für die QITools und Qlynx 3/3/00, das ist der Browser, der vollen Internetzugang gibt unter dem uQLx Emulator. Neue Adresse!

[www.hedwig.uklinux.net/index.html](http://www.hedwig.uklinux.net/index.html)

Bei **Dave Walker** gibt es die neuste Ausgabe der C68 Library für QDOS.

[www.itimpi.freeseerve.co.uk/](http://www.itimpi.freeseerve.co.uk/)

Von **Thierry Godefroy** gibts nun Archivers Control Panel 4.00 mit neuen Fähigkeiten: bzip, gzip, tar Kompression unterstützt, Teilen großer Dateien, ausgebreitetes Konfigurations-Menü.

Der Emulator **Qemulator 2.0** für Windows 9x/NT ist jetzt verfügbar. Er ist Shareware. Eine Ausprobierversion kann heruntergeladen werden:

[www.geocities.com/SiliconValley/heights/1296/](http://www.geocities.com/SiliconValley/heights/1296/)

**Just Words** hat jetzt eine Webseite:

[members.tripod.co.uk/geoffwicks/justwords.htm](http://members.tripod.co.uk/geoffwicks/justwords.htm)

**MIDIPlayer 2:** Al Boehm hat den MidiPlayer 2 fertig. Für alle, die einen Synthesizer besitzen und über ihren QL Musik hören wollen, holen ihn sich auf:

<http://boehm.home.hiwaay.net>

**Und nun zu den Artikeln im Heft:**

## **You and Your Programs - Just good Friends?**

**Du und deine Programme - einfach nur gute Freunde?**

### **Teil 7 - Handbücher - und warum sie keiner liest!**

Goeff Wicks beschreibt, welcher Aufwand es für einen Software-Autor ist, ein Handbuch dazu zu schreiben, wie schnell es veraltet, wenn man sein Programm auf dem Laufenden hält und wie wenig die Mühe eigentlich insofern belohnt wird, dass jemand das auch lesen würde. Seine Überzeugung ist, man möge dem Softwarebenutzer etwas an die Hand geben, was ihn interessiert, ihn neugierig macht.

Also:

1. Gutes Layout, übersichtlich, leicht zu lesen, attraktiv, die wichtigsten Dinge deutlich hervorgehoben.
2. a) Gute Basisinformation, also: was tut das Programm, wie muss ich es installieren;  
b) wie kann man schnell loslegen und  
c) schließlich noch alle Information, die man nicht unbedingt braucht, die aber interessant sind.

## **Der QL Stammbaum - Die 90er Jahre**

von **Dilwyn Jones**

Der zweite Teil der Geschichte des QL. Dilwyn beleuchtet die 90er Jahre. Sehr informativ und mit viel nostalgische Gefühle weckenden Informationen für jene, die dabei waren - DAMALS!

## **Die Funktion von Prozeduren und die Prozedur, Funktionen zu schreiben**

**Teil 2: Warum Funktionen? Das Resultat ist der Grund!**

von **Mark Knight**

Mark erklärt anhand einiger Beispiele den unterschiedlichen Gebrauch von Prozeduren und Funktionen. Letztere sind immer dann sinnvoll, wenn ein Ergebnis irgendeiner Art berechnet und zurückgegeben werden soll, z.B. zum Ausdrucken oder zum Übergeben in eine Variable, die dann weiter verwendet werden kann. Man kann damit verflossene Zeit berechnen, oder aber auch z.B. Klein- in Großbuchstaben umwandeln oder aber einfach Berechnungen ausführen, indem man alle nötigen Parameter an die Funktion übergibt. Das Ergebnis wird in jedem Fall mit einem RETURN zurückgegeben, anders als in einer Prozedur.

## **GEE Graphics! (On the QL?)**

**Part 15**

Herb Schaaf beschäftigt sich in diesem fünfzehnten Teil seiner Grafikpuzzeleien auf dem QL mit der dritten Dimension. Und zwar der dritten Dimension von Flächen und Linien. Nötig ist dafür das Programm FIT9c5\_bas, das auf der Cover Disk vom März-April 1999 war. Er gibt uns die mathematischen Grundlagen zur Berechnung der 3D Linien und Flächen und bezieht sich auf die Prozeduren und Funktionen, die in oben erwähntem Programm vorhanden sind.

## **Programmieren in Assembler - QLTDIs Teil 4**

von **Norman Dunbar**

Ein Artikel von 6 Seiten. Sicher sehr informativ für Leute die diesem Kurs von Norman Dunbar folgen. Für den weiteren Kommentar dazu siehe die vorhergehenden Ausgaben.

## **Achtung: Hotkeys - Verbrenn dir nicht die Finger, Teil 3**

von **David Denham**

In diesem Teil beschäftigt sich David mit den

# JOGHLEN MERZ SOFTWARE

Im stillen Winkel 12 D-47169 Duisburg  
Tel. 0203 502011 Fax 0203 502012  
<http://www.j-m-s.com/smsq/index.htm>

## SMSQ/E Version 2.98 Update

Ja, es existiert endlich! Die Farbtreiber funktionieren nun nicht nur auf Q40 sondern auch auf der QXL-Karte. 65536 Farben sind möglich! Das QXL-Interface wurde zudem verbessert, beschleunigt, der Tastaturtreiber wurde angepaßt, und die Geschwindigkeit der seriellen und Parallel-Schnittstelle wurde auch erhöht. So soll seriell jetzt bis zu 115kBaud möglich sein - je nach PC.

Die Versionen für andere Systeme wurden ebenfalls angepaßt. Die Hardware kann die "mehr Farben" zwar noch nicht, aber die Befehle sind implementiert. Neu ist, daß ein Bildschirmhintergrund (als Farbe, Muster oder Bild" angegeben werden kann, auch auf den ATARI- und GoldCard-Versionen.

### Preise für's Update/Upgrade

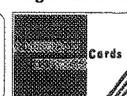
für ATARI ST, STE, TT	kostenlos
... mit Anleitungs-Ergänzung	DM 16,-
für GoldCard/SuperGoldCard	kostenlos
... mit Anleitungs-Ergänzung	DM 16,-
für QXL mit Anleitungs-Ergänzung	DM 79,90

#### **LIEFER- und ZAHLUNGSBEDINGUNGEN**

Versandkosten [Deutschland] DM 8,99 (wenn Rechnungsbetrag unter DM 50,- dann nur DM 5,99). Bei Rechnungsbeträgen über DM 500,- kostet es



DM 18,99 [Europa] DM 14,50 (wenn Rechnungsbetrag unter DM 50,- dann nur DM 9,50). Alle Preise inkl. 15% MwSt. Irrtum und Preisänderung vorbehalten. Verrechnungs-, Euroschecks und Kreditkarten werden akzeptiert. Bankeinzug möglich.



beiden Befehlen HOT\_KEY und HOT\_WAKE. HOT\_KEY tut im Prinzip nichts Anderes als ALTKEY aus dem Toolkit 2, wenn auch intern ein Unterschied besteht. Der Autor gibt eine Reihe von Beispielen, wozu man das gebrauchen kann. HOT\_WAKE ist dem HOT\_PICK Befehl ziemlich ähnlich. Jedoch werden Jobs nicht nur nach oben gezogen, sondern - falls das überhaupt zutrifft - Inhalte auf den neusten Stand gebracht. Ein Beispiel dafür ist das Dateien\_Menü aus QPAC2. Sollte ein Job gar nicht vorhanden sein, aber ein ausführbares Thing davon, wird eine neue Kopie dieses Things gestartet.

```
ERT HOT_THING('x', 'Dateien')
```

dagegen startet immer eine neue Kopie des Dateien-Menüs. Darüber hinaus kann man mit der HOT\_THING Funktion alle möglichen "Dinge" starten, u.a. auch die Funktion 'Schlafen'. Einfach definieren:

```
ERT HOT_THING('s', 'BUTTON_SLEEP')
```

und Sie können (beinahe) jeden Job schlafen legen indem Sie ALT&'s' drücken.

## Ein Dutzend Programme, ohne die man nicht leben sollte

von Darran Branagh

Der Autor stellt seine 12 Lieblings-, Wichtigst-, unverzichtbar seienden Programme vor. Verständlicherweise sind das alles Programme aus der Quanta-Bibliothek wie The Stripper, SCR2BMP, SCR2PCX, The Copier, Qformat, Sortback usw. Einige davon könnten auch für deutsche QLer interessant sein.

... und dies wird in dieser Ihnen vorliegenden Ausgabe vermisst:

## Programmieren mit den Basic-Erweiterungen von EASYPTR - Teil 3

Eine Serie von Wolfgang Uhlig

Hier hätte eigentlich der dritte Teil meiner Serie kommen sollen. Aus diversen privaten und gesundheitlichen Gründen bin ich nicht dazu gekommen. Bitte gedulden Sie sich bis zur nächsten Ausgabe.

---

# Programmieren mit Hilfe des C68-Compilers und "kleiner" Hilfsprogramme!

Frank Dibowski

Um ein Programm zu schreiben muss man sich leider auch überlegen, mit welcher Computersprache man dieses, manchmal doch hoffnungslose Unterfangen in Angriff nehmen will!

Bei mir war es zuerst BASIC, was irgendwie auch verständlich war. BASIC war (und ist immer noch) beim QL umsonst dabei und man sieht sofort das Ergebnis.

Leider war es nicht so schnell und ich bin auf PASCAL (Computer one) umgestiegen. Was aber auch nicht von langer Dauer war, da ich nicht herausgefunden habe, wie man mit PASCAL Programme für die Pointeroberfläche schreibt.

Also wieder zurück zu BASIC bis ich auf den C68-Compiler stieß. Er ist kostenlos und mittlerweile leicht zu erhalten, z.B. im Internet, in Jochens Box oder auf jedem QL-Treffen (auf denen ich war).

In diesem (und den folgenden) Artikel werde ich jetzt nicht darauf eingehen, wie er funktioniert oder wie er die selbst geschriebenen Sourcen verarbeitet - darüber mögen sich andere kluge Köpfe auslassen - ich werde versuchen zu erklären, wie man ihn installiert, welche Hilfsprogramme es für ihn gibt und wie man seinen selbstgeschriebenen Quelltext compiliert. Das eigentliche Ziel wird das Erstellen von Programmen für die Pointeroberfläche sein.

## Der C68-Compiler!

Wenn man sich den C68-Compiler aus dem Internet besorgt, erhält man sieben ZIP-Files die dann noch entpackt werden müssen. Wichtig sind vor allem die Files \*\*\*RUN1.ZIP, \*RUN2.ZIP und \*\*\*RUN3.ZIP. Sie enthalten den eigentlichen C-Compiler mit einer umfangreichen Bibliothek. (Das "\*\*\*" steht für die Versionsnummer, die sich in regelmässigen Abständen, wegen Verbesserungen ändert.) Nun muss man sich noch überlegen, auf was man sich den Compiler installiert. Ich selber arbeite mit einer QXL-Karte. Somit habe ich mir auch eine kleine Festplattenpartition (10Mb) angelegt, auf der ich den C68-Compiler installiert habe. Man kann ihn aber auch auf Diskette installieren. Man sollte dann aber zwei HD-Disketten einplanen, da er mit seinen Zusatzprogrammen schon auf 2Mb kommt. In Minimalform kann man ihn mit Sicherheit auch auf der 2Mb-RomDisQ installieren. Um nun den C68-Compiler auf der Festplatte (oder

Diskette) zu installieren, sollte man vorher ein paar Unterverzeichnisse anlegen. Ich habe zuerst einmal ein Verzeichnis mit dem Namen C68 mittels

```
make_dir win6_C68_
```

angelegt. Danach noch zwei

```
make_dir win6_C68_INCLUDE_
```

```
make_dir win6_C68_LIB_
```

die im Verzeichnis C68 liegen sollen und noch eins für die selbst geschriebenen Programme.

```
make_dir win6_CSource_
```

Nun habe ich die Zip-Files mittels unzip entpackt.

```
ex win1_prog_unzip_exe;"-x flp1_424FRUN1_ZIP -d win6_C68_"
```

```
ex win1_prog_unzip_exe;"-x flp1_424FRUN2_ZIP -d win6_C68_"
```

```
ex win1_prog_unzip_exe;"-x flp1_424FRUN3_ZIP -d win6_C68_"
```

Wer möchte kann so auch die Dokumentation und die Sourcen zu dem Compiler entpacken. Wenn das alles erledigt ist, dürfte der C68-Compiler vollständig installiert sein. Nun kann man gleich testen, ob er auch funktioniert. Dazu brauchen wir aber noch ein Testprogramm, besser gesagt, wir müssen versuchen, irgendein Sourcefile zu compilieren. Dazu werden wir jetzt mal "so eben" eins schreiben. Um das zu machen, brauchen wir zuerst einmal einen Editor. Kein Problem, beim C68-Compiler ist einer dabei. Dazu brauchen wir nur noch

```
exec win6_C68_qed
```

eingeben. Als Filename nehmen wir einfach "win6\_CSource\_Test.c". Bei Workspace bestätigen wir einfach mit ENTER. Jetzt können wir das eigentliche Programm schreiben.

```
#include <stdio.h>

void main(){

    printf ("Hallo alle miteinander!\n");

}
```

Danach einfach "F3" drücken und "sa" (für save) eingeben und mit ENTER bestätigen. Nun noch einmal "F3" drücken und "q" (für quit) eingeben. Jetzt setzen wir noch PROG\_USE und DATA\_USE auf "win6\_C68\_" und geben dann noch

```
exec cc;"win6_CSource_Test.c -o ram1_Test_exe"
```

ein. Auf dem Bildschirm sollte sich jetzt ein rot umrandetes Fenster öffnen, in dem folgendes steht:

```
cc v5.15 (Compiler driver) (c)1991-1998 D.J.Walker
WIN6_C68_cpp win6_CSource_Test.c Ram1_Test_i
WIN6_C68_c68 Test_i Test_s
"win6_CSource_Test.c", line 5: warning [5]: ignored return value from function
win6_C68_as68 Test_s win6_CSource_Test_o
win6_C68_ld -oram1_Test_exe win6_CSource_Test_o
Press a key to exit
```

Der letzten Aufforderung kommt man einfach nach und gibt in Basic nur noch

```
exec ram1_Test_exe
```

ein und hoffentlich sieht man dann das gewünschte Ergebnis.

Nun schreiben wir uns noch ein kleines Boot-Programm damit wir bestimmte Einstellungen nicht immer von Hand machen müssen und noch ein paar Erweiterungen, die uns (und dem Compiler) die Arbeit etwas erleichtern.

```
100 REMark Boot-Programm für den C68-Compiler
110 TK2_EXT
120 D1$="win6_C68_" : REMark für C68
130 D2$="win1_PIF_" : REMark - mit Vorsicht zu genießen
```

```

140 LRESPR D2$&"PTR_gen" : REMark Pointer Environment
150 LRESPR D2$&"WMAN" : REMark Window Manager
160 LRESPR D2$&"HOT_rext" : REMark HOTKEY System II
170 LRESPR D2$&"MENU_rext" : REMark Menü Erweiterung
180 LRESPR D2$&"THING_rext" : REMark Thing Erweiterung
190 LRESPR D2$&"QPAC2" : REMark QPAC2
200 LRESPR D1$&"PTH_rext" : REMark Path-Gerät
210 LRESPR D1$&"ENV_bin" : REMark C Environment Variablen
220 LRESPR D1$&"SIGEXT_rext" : REMark Signal Erweiterung
990 :
1000 OUTLN #0,512,228,0,28
1010 ERT HOT_LOAD ('E',D2$&"QD")
1020 ERT HOT_WAKE ('1','Dateien')
1030 ERT HOT_WAKE ('2','Exec')
1040 ERT HOT_WAKE ('3','Pick')
1050 ERT HOT_WAKE ('4','Hotkeys')
1060 ERT HOT_WAKE ('0','Ljob')
1070 ERT HOT_WAKE ('b','Sysdef')
2000 HOT_GO
2010 CLS #0
2020 WMON
2030 BT_SLEEP 'Exec'
2040 BT_SLEEP 'Pick'
2050 BT_SLEEP 'Hotkeys'
2060 BT_SLEEP 'Ljob'
2070 BT_SLEEP 'Sysdef'
2080 BT_EXEC Dateien
2400 :
2500 DATA_USE win6_C68_
2510 PROG_USE win6_C68_
2520 :
3000 SETENV "CFLAGS=-v -DQDOS" : REMark default CFLAGS
3010 SETENV "TEMP=RAM1_" : REMark temporary directory
3020 SETENV "TMP=RAM1_" : REMark (alternative for some programs)
3030 SETENV "C68=-warn=6" : REMark Set default warning level to 6
3040 SETENV "CC_PATH=WIN6_C68_" : REMark Directory for C68 compiler phases
3050 SETENV "INC_DIR=WIN6_C68_INCLUDE_" : REMark System Include files
3060 SETENV "LIB_DIR=WIN6_C68_LIB_" : REMark Libraries directory

```

Von BASIC-Zeile 140 bis 220 werden erst einmal einige Erweiterungen geladen, die zwar alle nicht unbedingt notwendig für das Arbeiten mit dem C68-Compiler sind - aber halt hilfreich. Die Zeilen 140 bis 200 kann man auch "noch" weg lassen wenn man möchte, aber da mein eigentliches Ziel die Programmierung der Pointer-Oberfläche in C ist, werden sie irgendwann benötigt. Wer sie am Anfang nicht mit eingibt, der kann dann auch die Zeilen 1000 - 2400 weg lassen, da sie ohne die Erweiterungen keine Bedeutung haben. Zeile 1000 ist nur aus versehen hinein geraten, da ich sie zum Arbeiten doch ab und an brauche. Sie erlaubt es mir in BASIC mit QPTR und QMenu einiges schnell auszutesten. Zeile 1010 lege ich einen Hotkey um QD (von Jochen Merz) zu laden. Wer ihn nicht besitzt oder nicht mag kann dort auch einen anderen Texteditor eingeben, z.B.

```
1010 ERT HOT_LOAD ('E',D1$"QED")
```

Die Zeilen 3000 bis 3060 habe ich nur aus dem mitgelieferten Boot kopiert. (Wie man unschwer erkennt). Sie werden vom Compiler genutzt, sind aber nicht unbedingt notwendig! Erläuterungen hierzu kommen in späteren Artikeln.

Über meine Zeilennummerierung sollte man sich nicht wundern. Das Bootprogramm wird mit Sicherheit noch erweitert und da ich nicht immer das ganze Listing angeben möchte, habe ich einige Lücken gelassen. Ich brauche dann später nur noch die neuen Zeilen anzugeben.

So das wars jetzt erst einmal! Der C68-Compiler ist jetzt (hoffentlich) installiert. Sollte ich hier etwas vergessen oder falsch erklärt haben, so laßt es mich wissen. Ich bin auch für sonstige Kritik, Anregung oder Fragen offen.

# QL-Bilder

Dietrich Buder

Das Heft QL Today vom März/April 2000 zeigt im Titelbild den Urvater aller QLs, einen englischen QL-CLASSIC, die geliebte schwarze Kiste. Von solchen Ur-QLs gibt es sicher noch einige, doch ohne jegliche Erweiterung, wie auf dem Titelbild, schlummern diese vermutlich unbenutzt in irgend einem Verließ. Doch mehr oder weniger hoch gerüstet sind einige dieser Kisten noch immer im täglichen Einsatz und auch bei QL-Treffen auf dem Tisch von Tony Firshman zu sehen.

Stand der QL-Technik ist derzeit vorrangig ein üblicher PC, verächtlich auch Windose gescholten, der mit dem QPC-Emulator oder der QXL-Karte läuft. Derartige QLs in einem PC-Gehäuse oder Laptop sehen zwar nicht völlig gleich

aus, sind aber selten ein Foto wert, außer wenn ihr Besitzer mit ins Bild kommt.

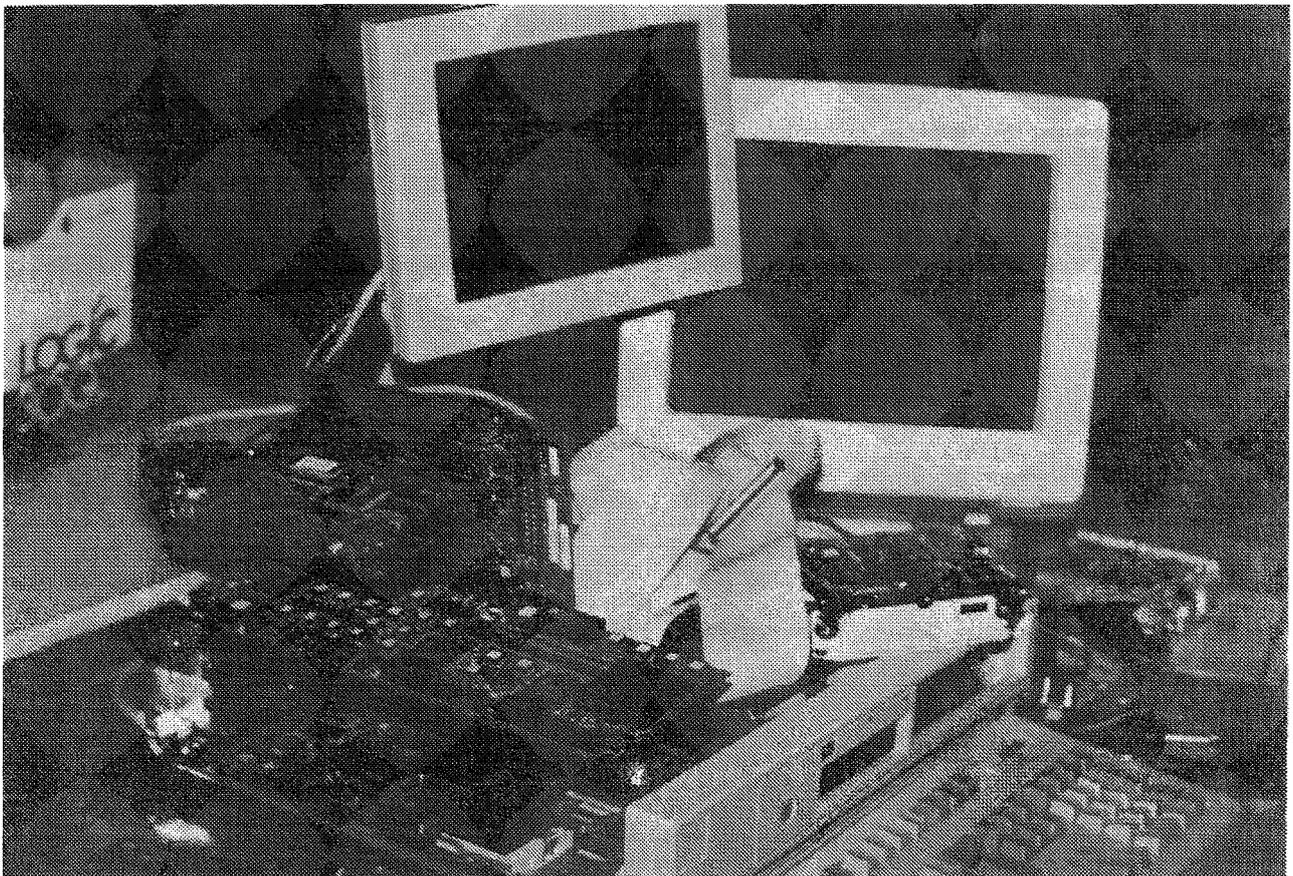
Das Titelbild der genannten Hefte veranlasste mich, zwei nicht alltägliche QLs als Ergänzung zu dem Titelbild vorzustellen.

Das erste Foto zeigt einen QL von Peter Blaha, den er 1998 zum QL-Treffen in Heidenreichstein (Österreich Waldviertel) benutzte. Leider kann ich zu der Technik dieses Geräts nichts weiter sagen. Mich beeindruckte aber der damals noch sehr teuren Flachbildschirm und dessen scharfe Bilder; dieses zählt auch heute noch nicht zum QL-Standard. Weiterhin fand ich das offene Gehäuse und die wackelige Anordnung mehrerer Karten sehenswert und betätigte deshalb den Auslöser der Kamera. Ähnliche Aufbauten hatte vor etlichen Jahren auch Jürgen Falkenberg bei diversen QL-Treffen, nur habe ich das leider nie fotografiert.

Das zweite Foto zeigt zwei meiner QL-CLASSICs und soll demonstrieren, dass trotz vieler Komponenten der verhasste Kabelsalat vermieden werden kann. Alle Kabel sind mit Kabel-Drillern (Fa. Conrad) auf passende Länge gebracht und teilweise mit Hellermann-Schellen unter der Tischkante festgeklemmt.

Mit meiner nachfolgenden Beschreibung möchte ich möglichst viele QLer ermuntern, ihre alten QLs nicht zu verbannen, sondern vielleicht doch noch etwas aufzurüsten und neben ihrem QPC/QXL-PC weiterhin sinnvoll zu nutzen.

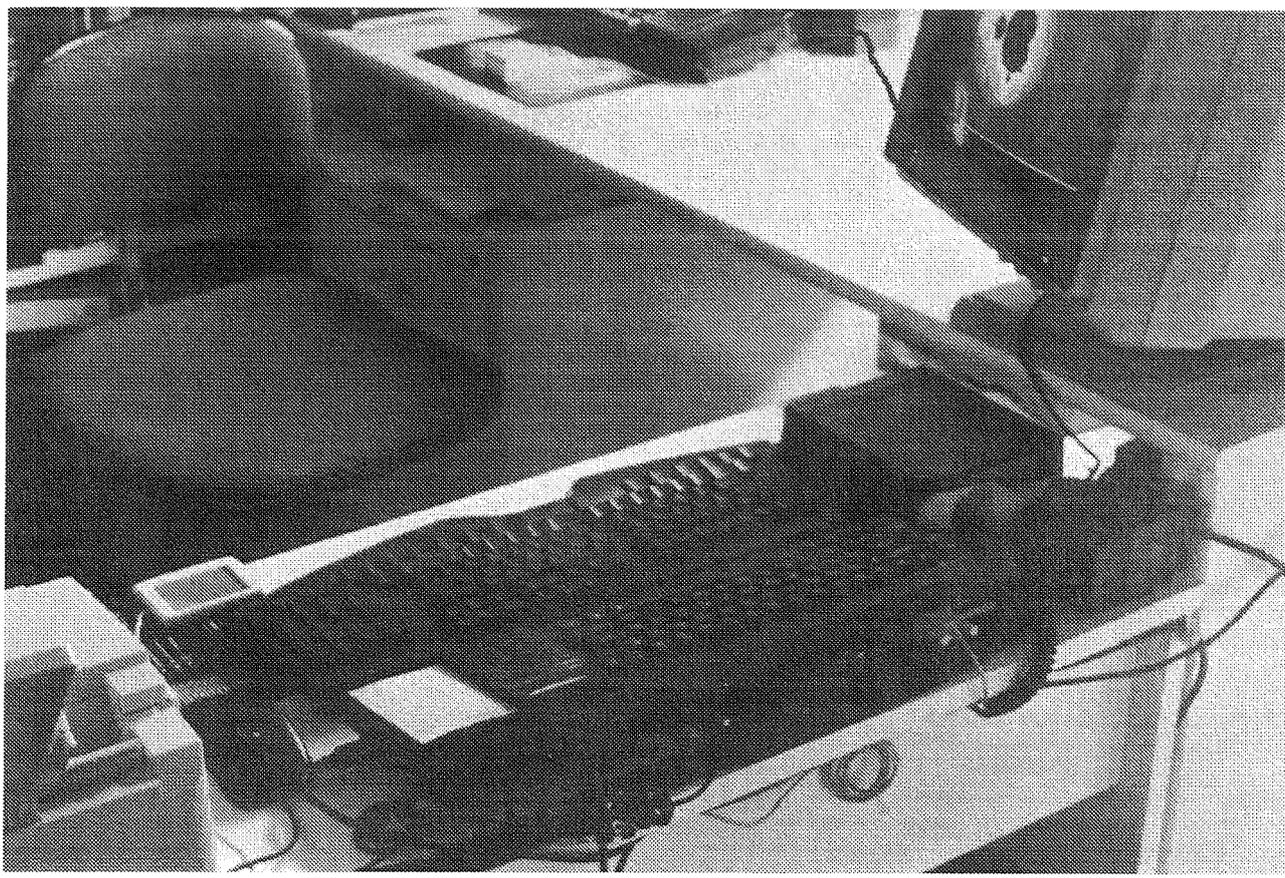
Beide fotografierten QLs haben SuperGoldCard, 8-MB-RomDisq, SerMaus sowie Schön-Tastatur und sind über das originale Netzkabel miteinander verbunden. Der vordere QL enthält einen Superhermes mit angeschlossener BTC-Tastatur und Touchpad am Mausport. Er arbeitet bewusst noch mit dem origi-



nalen MGG-ROM, weil der speziell für residente Prozeduren einige 'Klimmzüge' im BOOT erfordert. In der Diskettenstation befinden sich zwei ED-Laufwerke, mein 5 1/4 Laufwerk ist nicht mehr angeschlossen. Das Netzteil hat einen Schalter, so dass Komponenten bequem stromlos ausgetauscht werden können. Der hintere QL besitzt Hermes

V2.20 und einen speziellen Minerva-ROM V1.97 mit eingebrenntem Euro-Zeichen. Beide Geräte brauchen keine Festplatte, da auf eine 8-MB-Romdisk alle meine wichtigsten Arbeitsprogramme passen. Diese QLs arbeiten lautlos und eignen sich somit bestens zum stundenlangen Berechnen von Apfelmännchen. Weiterhin brauche ich sie unter QDOS zum

Kompilieren mit dem TURBO-Kompiler, was bisher auf PCs mit QPC/QLX-Karte unmöglich ist. Auch nehme ich diese QLs zum Vergleichen und Verbessern von Eigenprogrammen, zum Testen von irgendwelchen Komponenten und letztlich zur Prüfung meiner Programme auch unter QDOS. **Wer hat noch Bilder seines Systems? Immer her damit!**



## Die Cover Disk - ProWesS

Dieses Mal gibt es eine sehr interessante Cover Disk: ProWesS, der neue Window Manager mit skalierbaren Schriften, wurde von PROGS freigegeben und kann damit kostenlos verteilt werden.

ProWesS paßt nur auf eine HD-Diskette und kann nur vernünftig mit HD oder ED-Diskettenkapazitäten oder - noch besser - Festplatte betrieben werden. Wir wollen damit nicht die QLer mit "nur" DD-Laufwer-

ken verschrecken - die nächste Cover Disk wird wieder auf DDs kommen, doch dieses Mal würde es keinen Sinn machen, ProWesS auf zwei Disketten zu verteilen - die Nutzung wäre unmöglich.

Also nicht traurig sein und Geduld bis zur nächsten Cover Disk - es gab ja auch erst eine in der vergangenen Ausgabe.

Zur Installation von ProWesS Diskette in flp1\_ einlegen und booten, der Rest geht ziemlich

automatisch. Nach der Installation sind einige ProWesS-Anwendungen sofort nutzbar. Es gibt noch diverse andere Anwendungen, darunter auch eine Textverarbeitung namens "Paragraph" (eine Art Shareware, man kann sie kostenlos erhalten - z.B. aus der JMS Mailbox) - und, wenn es einem gefällt, den Autor durch Zahlung eines geringen Betrags zur Weiterentwicklung motivieren. Über Paragraph wurde schon in einer englischen Ausgabe von QL Today berichtet; wie wäre es mit einem deutschen Bericht?

# Neues Spiel: The Wall

Jochen Merz

Wer mich kennt weiß, daß ich gerne Spiele spiele. Daher hoffe ich, wird man mir verzeihen, daß ich über ein Produkt schreibe, das auch bei mir erhältlich ist.

Wer meine Spiele kennt (z.B. The Oracle oder Diamonds) weiß auch, daß ich selber

Faktor ist auch nicht zu verachten (und reich wird mit dem QL sowieso niemand!).

The Wall startet in verschiedenen Schwierigkeitsstufen - die erste Abbildung zeigt die einfachste. Man darf alle Teile im Feld anklicken; sind mindestens zwei gleiche Teile neben-

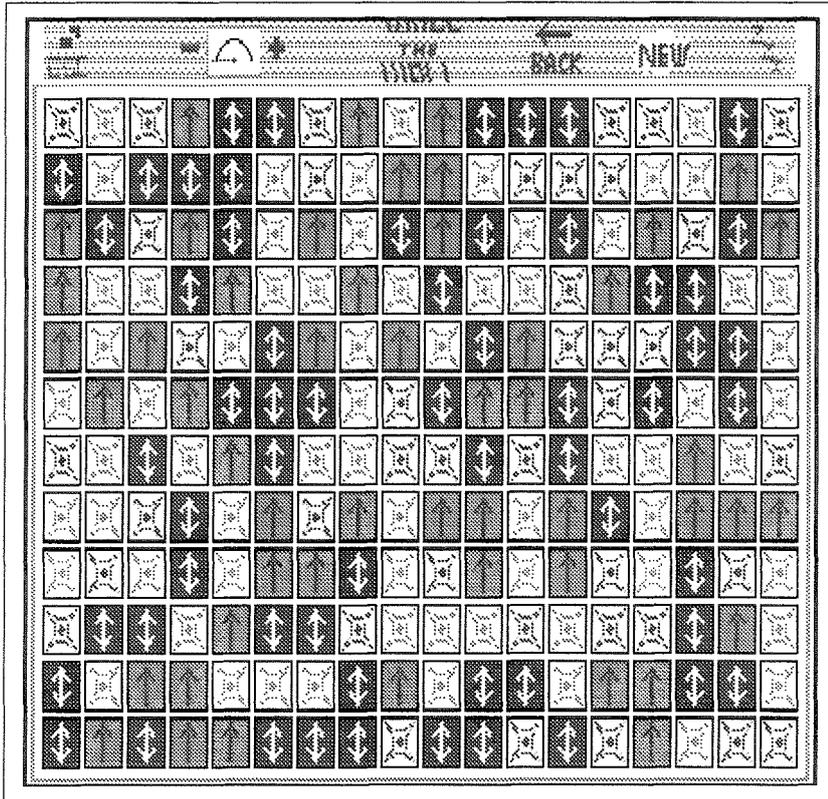
oder übereinander verschwinden sie. Zuerst rutscht alles von oben nach, dann von rechts nach links. Ziel ist - wie sollte es anders sein - das Feld leer zu bekommen. Ich hab's nach unzähligen Spielen (ach, ist das zeitraubend, aber man muß es ja testen) einmal geschafft - wie man im zweiten Bild unschwer sehen kann.

The Wall enthält definitiv einen Suchfaktor. Dies bestätigt mir auch Dilwyn und seine Freundin, die kaum mit dem englischen QL Today weiterkommen weil sie von der Testversion gefesselt sind (vielleicht erreicht es ja noch zur nächsten englischen Ausgabe einen Tester).

Geschafft haben sie's noch nicht - da bin ich "Erster" - allerdings nur im einfachen Level.

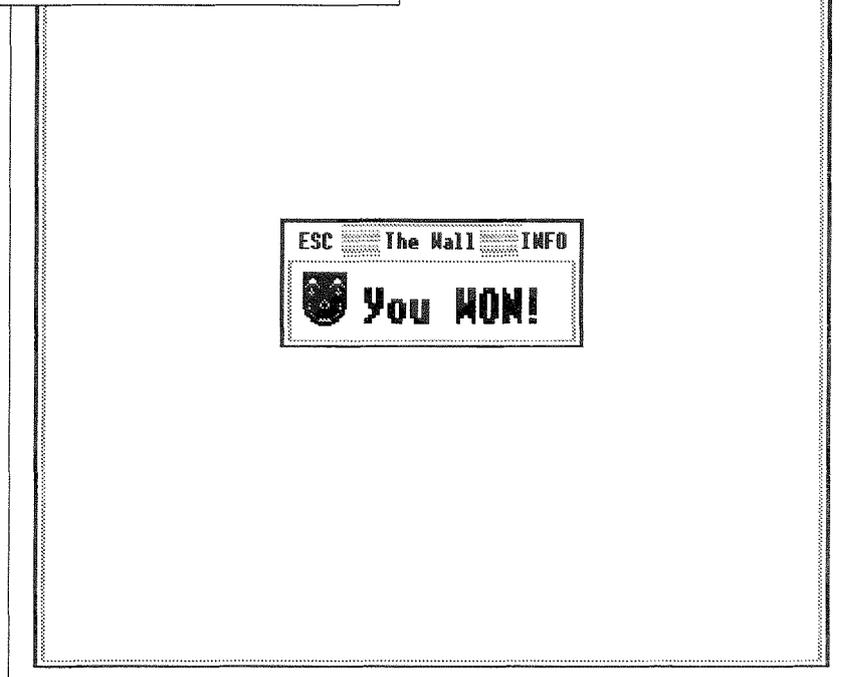
Wer meinen "Testbericht" als nicht unabhängig erachtet kann's gerne selber mal auf einem QL-Treffen probieren - ansonsten: The Wall kostet nicht die Welt und macht Spaß!

Glücklicherweise ist hiermit die deutsche Ausgabe fast fertig, also auf zum nächsten Spiel!



gerne Denkspiele programmiere. Obwohl - ich muß gestehen, daß Lonely Joker mein Lieblingsspiel ist und ich ganz bestimmt mit keinem anderen Spiel so lange und so viele Spiele gespielt habe.

The Wall ist nicht von mir, sondern von Wolfgang Lenerz. Man kennt Wolfgang ja als Utility-Programmierer (z.B. FiFi, Wined oder Basic Linker) - doch macht das Programmieren von Spielen eigentlich mehr Spaß als das Programmieren von Anwendungen. Bringt zwar nicht viel ein, aber der Spaß-



# **Nächstes Eindhoven-Treffen am Samstag, den 26. August!!!**

---

## **QL 2000**

Das große QL-2000 Treffen findet im Horizon Centre, Sundridge Close, Cosham statt.

Samstag, der 14. Oktober, ist der Eröffnungstag mit den normalen QL-Aktivitäten (Händler, Bring 'n' Buy, Quanta Tisch usw.). Wir starten um 10:00 Uhr und beenden das Treffen an diesem Tag um 16:00.

Sonntag, den 15. Oktober, geht es dann um 10:00 Uhr wieder mit einem "Open Forum" mit dem Namen "The Way Forward" (Der Weg Vorwärts) weiter. Wir hoffen, daß wirklich jeder kommt, denn es ist sehr wichtig für das zukünftige Überleben der QL Gemeinde. Wie es bislang (schon sehr positiv) aussieht wird eine größere Anzahl Besuchern außerhalb Englands erwartet, einschließlich unserer "Freunde" aus den USA.

Die nächstgelegene Übernachtungsmöglichkeit ist das Travel Inn am North Harbour (Nord-Hafen) mit einem Preis von £40,95 pro Raum (Raucher oder Nichtraucher-Räume sind möglich). Frühstück kostet £6 für ein typisches vollständiges English Breakfast bzw. £4 für ein "Continental Breakfast". Wer also gerne an diesem außergewöhnlichen Treffen teilnehmen möchte sollte möglichst vor Ende Juni buchen und sich seinen Raum reservieren. Telefonnummer: 0044 23 9232 1122.

Das Travel Inn befindet sich direkt neben einem "Beefeater" und uns wurde gesagt, daß man Abendessen besser einige Wochen im Voraus reservieren läßt.

Roy Brereton von Quanta, der das Treffen organisiert, schreibt: Wenn mir genügend Leute frühzeitig mitteilen, daß sie gerne an einem Abendessen teilnehmen möchten, dann kann ich Tische im Voraus für Freitag, den 13. (!!) und/oder Samstag, den 14. Oktober reservieren. Bitte sagt Roy frühzeitig Bescheid! Roy's Email-Adresse ist [roy.brereton@tesco.net](mailto:roy.brereton@tesco.net)

Das war's momentan an Informationen, mehr Infos gibt es wenn sie uns vorliegen.

Zu den anderen Treffen liegen uns nach wie vor keine Bestätigungen vor - sehr sicher ist jedoch, daß sie Ende September/Anfang Oktober stattfinden. Wir halten Sie auf dem Laufenden.