

# Optimieren und Compilieren

von Christian Schmitz

**Serie Realbasic, Folge 5** In der letzten Folge haben wir den Code unseres Programms bereits fertig gestellt. Damit daraus auch ein auslieferungsfähiges Produkt wird, wollen wir aber noch ein wenig an der Optik feilen

## EINSTIEG

Unsere Serie erklärt Schritt für Schritt, wie man in Realbasic ein Brettspiel à la „Mensch ärgere Dich nicht“ erstellt. In dieser letzten Folge zeigen wir, wie wir unser Programm zur Veröffentlichung fertig stellen und für Mac-OS X verschönern.

✦ **UNTER MAC-OS X** kommt vor allem dem Programm-Icon eine wichtige Bedeutung zu, denn die Icons sind unter Mac-OS X deutlich größer und fallen im Finder gut auf. Unter Mac-OS 9 und Mac-OS X bekommt ein Programm jedoch nur dann ein eigenes Icon, wenn es mit einem Creator-Code versehen ist. Über diesen Code ordnet der Finder die Icons den Programmen zu. Ein Creator-Code besteht aus vier beliebigen Zeichen, wobei alle Codes, die ausschließlich aus Großbuchstaben bestehen, bereits für Apple reserviert sind. FNDR ist beispielsweise der Creator-Code für den Finder, BOBO für Apple Works. Andere Beispiele sind nwSP für Mac Soup, VtPC für Virtual PC und MSWD für Microsoft Word.

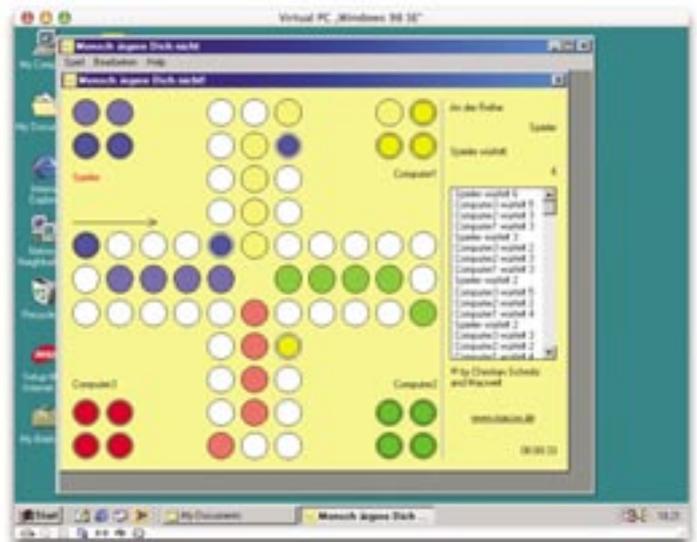
Den Creator-Code darf man sich jedoch nicht beliebig ausdenken. Alle Codes müssen bei Apple registriert sein, um Doppler zu vermeiden. Unter der Adresse <http://developer.apple.com/dev/cftype/information.html> kann man seinen Creator-Code prüfen und registrieren lassen. Dieser Code ist, wenn man ihn denn bekommt, einmalig auf der Welt. Für unser Spiel „Mensch ärgere Dich nicht“ haben wir bereits den Creator-Code „MäDn“ registriert, der bislang noch ungenutzt war. Wir tragen ihn im Fenster „Projekteinstellungen“ im Bearbeiten-Menü von Realbasic ein.

## Icons organisieren

Es gibt mehrere Möglichkeiten an ein hübsches und aussagekräftiges Icon zu kommen. Apple hat dafür extra ein Team von Grafikern

### Serie: Realbasic für Profis

1 Interface	Heft 12/2001
2 Computerspieler	Heft 01/2002
3 Mensch spielt mit	Heft 02/2002
4 Multiplayer	Heft 04/2002
5 <b>Compilieren und Release</b>	<b>Heft 05/2002</b>



eingestellt und auch Sie können sich ein Icon bei einem Grafiker malen lassen. Das kann allerdings schon mal 100 Euro oder mehr kosten. Nicht jeder kann sich das leisten.

Für unser Programm machen wir es uns etwas einfacher. Wir fertigen ein Icon aus einem verkleinerten Screenshot unseres Programms an. Dazu lassen wir das Spiel ein paar Minuten lang automatisch laufen (vier Computer spielen gegeneinander) und wenn es interessant aussieht, schieben wir die Maus beiseite und lösen mit den Tasten Befehl-Umschalt-3 einen Screenshot aus.

Diesen Screenshot öffnen wir zum Beispiel mit dem Programm Grafikkonverter. Nun ziehen wir einen quadratischen Bereich über der Spielfläche auf und achten dabei darauf, dass Breite und Höhe des Ausschnittes, die rechts oben bei den Mauskoordinaten angezeigt werden, gleich groß sind. Anschließend wählen wir im Menü Bearbeiten den Punkt „Auswahl freistellen“ (Befehl-Y).

Diesen quadratischen Ausschnitt müssen wir nun noch auf 128 mal 128 Pixel herunterrechnen. Dazu wählen wir im Menü Bild unter

**Fremdgänger** Auch unter Windows macht unser Spiel eine gute Figur. Hier sieht man Virtual PC unter Mac-OS X mit Windows 98.

## Tipp | Hilfsprogramme im Internet

Alle in dieser Folge erwähnten Hilfsprogramme und Utilities finden Sie im Internet. Zumeist handelt es sich um kostenlose Software oder Shareware, für die man ein paar Euro oder Dollar Sharewaregebühr zahlen muss.

Programm	Vertrieb	Preis	Internet
Grafikkonverter	Lemkesoft, Shareware	25 €	www.lemkesoft.de
Res-Edit	Apple	kostenlos	ftp.apple.com/Apple_Support_Area/Apple_Software_Updates/English-North_American/Macintosh/Utilities/ResEdit_2.1.3.sea.bin
Iconographer	Mscape, Shareware	15 US-Dollar	www.mscape.com/products/iconographer.html
Icon Paste Mac-OS 9	Realsoftware	kostenlos	ftp.realsoftware.com/REALbasic/release/IconPaste.sit
Icon Paste Mac-OS X	Realsoftware	kostenlos	ftp.realsoftware.com/REALbasic/release/IconPasteX.sit
Plug-in-Sammlung	Monkeybread Software	kostenlos	http://monkeybreadsoftware.de/realbasic

Bildgröße die Option „Skalieren...“ aus (Befehl-Control-Y). Wir geben für die Breite 128 Pixel ein und aktivieren die drei Häkchen für Proportional, Bild Skalieren und Fehlerkorrektur. Das nun skalierte Bild speichern wir im Format PICT.

### Icon zusammensetzen

Um das Icon in eine Mac-Resource zu wandeln gibt es verschiedene Möglichkeiten. Eine ist das Programm Res-Edit von Apple (siehe Kasten „Hilfsprogramme im Internet“)

Wir legen in Res-Edit eine neue Datei an. Dort ergänzen wir eine Resource vom Typ „ic18“ mit der ID 128 und fügen in dem Icon-Editor Icons für die verschiedenen Auflösungen über die Zwischenablage ein. Diese Resource-Datei speichern wir nun.

Die zweite Möglichkeit ist entgegenkommender, kostet aber Geld. Gemeint ist das Programm Iconographer. Es ist für viele Entwickler das Icon-Malprogramm der Wahl, denn es erstellt auch Icons für Mac-OS X. Für reine Mac-OS Classic-Programme reicht Res-Edit aus.

In Iconographer legen wir ein neues Icon an. Zuerst sehen wir uns im Menü „Icon“ das Infofenster an. Die ID sollte 128 sein, den Namen wählen wir beliebig und stellen „Mac OS Universal“ als Format ein. Unter der Karteikarte „Member“ können wir noch detailliert festlegen, welche Icon-Typen die Datei enthalten

soll. Da der Finder fehlende Icons bei Bedarf selbst berechnet, könnte man beispielsweise die Huge-Icons weglassen, denn diese sind nur interessant, wenn die Icons beim Skalieren, zum Beispiel im Dock unter Mac-OS X, nicht schön aussehen.

Wir kopieren nun unser Icon per Zwischenablage von Grafikkonverter in Iconographer in das Icon „Thumbnail 32bit Icon“. Die Maske füllen wir mit Schwarz, denn nur die Icon-Pixel, die in der Maske schwarz sind, werden später auch vom Finder angezeigt. Mit der 8-Bit-Maske kann man auch wunderschöne Schatten erzeugen wie man es an den Realbasic-Standard-Icons sieht.

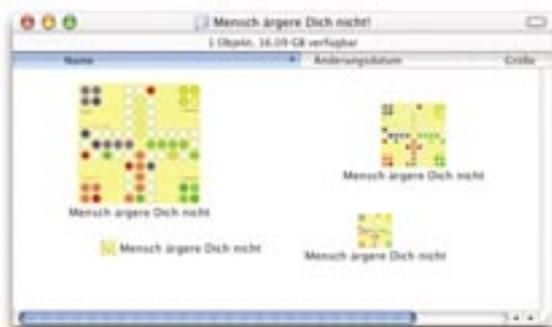
Mit dem Menübefehl „Add Member...2“ im Icon-Menü erzeugen wir automatisch aus dem großen Icon die anderen Icon-Typen. Von dieser Funktion machen wir Gebrauch, sie spart uns eine Menge Arbeit. Trotzdem sehen wir uns jedes so generierte Icon noch einmal kritisch an. Die ein oder andere Retusche ist meist notwendig.

### Einbauen ins Projekt

Der Dialog „Compiler-Einstellungen“ im Ablage-Menü von Realbasic ist bei Icons nicht besonders ausgereift. Ein großes Mac-OS X Icon können wir hier leider nicht einfügen. Wir müssen sie als Mac-Resource ins Projekt hineinziehen und für das Windows-Binary zusätzlich noch ein kleines und mittleres Icon in diesen Dialog einfügen.

Iconographer speichert eine einzige Datei, die all unsere Ressourcen beinhaltet. Realbasic akzeptiert nur Resource-Dateien mit dem Type „rsrc“ und dem Creator „RSED“. Anders gesagt: Realbasic akzeptiert nur Resource-Dateien von Res-Edit, obwohl es Res-Edit für Mac-OS X gar nicht gibt. Damit das klappt, müssen wir den Type und den Creator der Datei ändern und Realbasic vorspiegeln, dass sie von Res-Edit stammt. Erfreulicherweise kann man dies mit Res-Edit selbst erledigen. ▶

Viele bunte Icons Diese Montage zeigt die Icons für unser Spiel in vier verschiedenen Skalierstufen. Mac-OS X zeigt das 128 Pixel große Icon sehr detailliert.





Die modifizierte Resource-Datei ziehen wir nun per Drag-and-drop ins Realbasic-Projektfenster, wo sie als Alias verbleibt.

### Im Build Dialog

Werfen wir einen Blick auf den Screenshot (oben) „Build-Settings“. Er zeigt den fertigen Dialog mit allen Einstellungen, wie er in Realbasic 4 unter Mac-OS X aussieht. Wenn wir jetzt das Programm compilieren, wird Realbasic die Warnung bringen, dass Ressourcen überschrieben werden. Wir ignorieren das, denn das ist genau das, was wir wollen!

Wir können unser Programm für Mac-OS Classic und für Mac-OS Carbon erzeugen. Classic-Programme laufen ab System 7 bis Mac-OS 9 und passen sich automatisch dem jeweiligen Erscheinungsbild des Mac-OS an.

Mac-OS-Carbon-Programme sind in erster Linie für Mac-OS X gedacht. Sie laufen mit installierter Carbon Lib jedoch auch unter Mac-OS 8.6 bis Mac-OS 9.2.2.

Carbon als Schicht zwischen dem Programm und der jeweiligen Mac-OS Version benötigt aber unter Mac-OS Classic (nicht bei Mac OS X!) etwa 3% mehr Rechenzeit für Funktionsaufrufe, so dass man vorzugsweise sein Programm für beide Umgebungen anbieten sollte. Da unser Spiel nicht sehr zeitkritisch arbeitet, würde uns auch eine Carbon-Version für den Classic-Betrieb genügen.

Die Speicherzuteilung für die Classic-Umgebung muss man selber grob schätzen oder raten. Es gibt wenig konkrete Hilfe den Speicherbedarf eines Realbasic-Programms herauszufinden. Einem Carbon-Programm sollte man mindestens 2 MB zuteilen. Je nach Anzahl der Grafiken, Sounds und Filmen im Programm erhöht sich der Speicherbedarf, so dass schnell auch mal 10 MB zusammen kommen. Für dieses Spiel in der Variante ohne viele Bilder (wir haben keine Bilder als Ressourcen eingebunden und erzeugen auch keine

zur Laufzeit) sollten 2 MB reichen. Wir schreiben also 2048 KB in die zwei Felder für die Speicherzuteilung.

### Einstellungen für Windows

Besitzer von Realbasic Pro können das Programm auch für Windows compilieren. Der Dateiname muss dabei mit „.exe“ enden. Um laute im Programmnamen können sich wegen der unterschiedlichen internen Kodierung beim Kopieren zu Windows hin verändern. Der Einfachheit halber nennen wir das Spiel schlicht „Mensch.exe“.

Unter Windows gibt es zwei mögliche Programmtypen. MDI steht für „Multiple Document Interface“ und SDI für „Single Document Interface“. Das Windows-Programm Notepad ist ein Beispiel für ein SDI-Pro-

**Häusle baue** Im Build Dialog tragen wir die Dateinamen und Programminformationen für die compilierten Versionen ein.

gramm, denn es kann immer nur ein Dokument in einem Fenster öffnen. Einige Word und Excel Versionen können aber auch mehrere Dokumente in unterschiedlichen Fenstern öffnen. Bei einem solchen MDI-Programm umschließt ein großes Fenster mit der Menüleiste alle Dokumentenfenster.

In unserem Spiel kann man mehrere Spielbretter öffnen. Wenn wir uns jedoch für SDI entscheiden, müssen wir dies unterbinden, denn nur das erste Spielfenster bekommt eine Menüleiste. Schließt man diese, hat man gar keine Menüleiste mehr.

Wir aktivieren also das Häkchen bei „Multiple Document Interface“ und bekommen eine Menüleiste in einem übergeordneten Fenster. Vergrößert man dieses Parent-Fenster, hängt die Menüleiste wie beim Mac am oberen Bildschirmrand.

### Wir wollen uns verewigen

In Deutschland gilt das Urheberrecht, wonach unsere Arbeit als Programmierer automatisch gesetzlich geschützt ist. Wir dürfen selbst entscheiden, ob und wie wir unser Programm veröffentlichten möchten. Richtig verkaufen können wir es aber nicht, sondern nur lizenzieren. Wir räumen also zum Beispiel einer Firma Vertriebs- und Nutzungsrechte ein oder vergeben einfach, à la Open Source, eine Lizenz zum uneingeschränkten Weiterverwenden des Programms an alle.

Die meisten Hobbyprogrammierer werden die Programme, mit der Erlaubnis sie kostenlos privat einzusetzen, einfach weitergeben. Wer das Programm dann vertreiben oder aus-

### Tip | Icons für Mac-OS und Windows einbinden

Wir wollen unser Programm auch für Windows compilieren. Ein Icon für Windows bekommen wir über den Realbasic-Dialg „Compiler-Einstellungen“. Wenn wir ein Icon dort einfügen, müssen wir in einem Rutsch alle sechs möglichen Icon-Typen über die Zwischenablage hineinclippen. Realsoftware hat für diesen Zweck ein kleines Programm namens Icon Paste entwickelt, das dies deutlich einfacher macht. In Icon Paste fügen wir alle Icon-Typen einzeln ein. Dazu öffnen wir in Res-Edit unser fertiges Mac-OS-Programm und kopieren die Ressourcen über die Zwischenablage Stück für Stück in Icon Paste. Wir benötigen:

Resource icl8, ID 128 (Icon in 32x32 Pixel und 256 Farben)

Resource icl4, ID 128 (Icon in 32x32 Pixel und 16 Farben)

Resource ICN#, ID 128 (Icon in 32x32 Pixel in S/W mit der Maske)

Resource ics8, ID 128 (Icon in 16x16 Pixel und 256 Farben)

Resource ics4, ID 128 (Icon in 16x16 Pixel und 16 Farben)

Resource ics#, ID 128 (Icon in 16x16 Pixel in S/W mit der Maske)

In Icon Paste wählen wir anschließend im Menü Edit den Befehl „Copy Icons“ aus und fügen den Inhalt der Zwischenablage in Realbasic im Dialg „Compiler-Einstellungen“ ein. Dazu klicken wir einmal auf das kleine Icon links unten, um es auszuwählen und fügen die Icons per Befehl-V ein. Wenn alles geklappt hat, sieht man im compilierten Windows Programm nach dem Starten links oben im Fenster das kleine Icon.

bauen will, muss sich beim Programmierer melden. Damit jeder weiß, wo er sich melden soll, sollte man sich im Programm verewigen und zwar so vorausschauend, dass man auch nach fünf Jahren noch Kontakt bekommt und auch dann, wenn alle Readme-Dateien verloren gegangen sein sollten. Was unserem Spiel also noch fehlt, ist Dekoration und vor allem eine Dialogbox mit Informationen über den, der das Programm entwickelt hat.

## Die Aboutbox

Wir erstellen ein neues Fenster namens „AboutWindow“ und geben ihm den Titel „Über Mensch ärgere Dich nicht“. Wir möchten gerne in diesem Fenster links oben das aktuelle Programm-Icon als Logo anzeigen. Dafür gibt es mehrere Möglichkeiten: Wir können das Icon noch einmal als Bilddatei speichern, in das Projekt legen und in einem Canvas als Backdrop auswählen.

Alternativ dazu kann man unter Mac-OS mit der Zeile

```
me.backdrop = app.resourceFork.getIcon(128)
```

das Icon mit der ID 128 aus der Resourcefork des Programms selber laden. Das klappt auch in unserem Spiel, hat jedoch zwei Nachteile: Erstens bekommt man nur ein Icon mit maximal 32 Pixel Breite zu Stande und zweitens funktioniert das unter Windows nicht, denn dort gibt es generell keine Resourcefork. Besorgt man sich eine spezielle Plug-in-Sammlung für Realbasic von Monkeybread Software kann man über die Zeile

```
app.applicationFile.DrawWidelIcon  
g,20,14,128
```

im Paint-Event des Fensters das Programm-Icon in voller Größe zeichnen. Nur unter Windows sollte man statt der 128er Größe besser die 32er Icons nehmen, da es in diesem System keine so großen Icons gibt.

## Internet-Links im Fenster

Wir nehmen einen Statictext, schreiben eine Internet-Adresse wie zum Beispiel „macwelt.de“ hinein, setzen die Farbe auf Blau und die Schrift auf Times 14 mit dem Attribut Unterstrichen – schon sieht das aus wie ein Internet-Link im Browser. Ab Realbasic 4.0 hat ein Statictext-Object auch MouseDown- und MouseUp-Events. Benutzt man eine ältere Realbasic-Version, so muss man erst ein Canvas darüber legen um an diese Events zu kommen. Egal ob Canvas oder Statictext, wir schreiben in den MouseDown-Event des Links:

```
me.textColor=Rgb(255,0,255) // I!a text  
return true  
// Ja, wir verarbeiten diesen Event
```



Sechs auf einen Streich Icon Paste erleichtert die Zusammenstellung der richtigen Icon-Resources für das Programm-Icon unter klassischem Mac-OS.



Alle Infos So sollte ein sinnvoller About-Dialog aussehen: mit Logo und Kontaktmöglichkeit zum Programmierer.

Beim MouseUp-Event tippen wir analog dazu den folgenden Realbasic-Code ein:

```
me.textColor=Rgb(0,0,255)  
// Wieder auf blau  
showURL "http://www.macwelt.de"  
// Webseite im Browser öffnen
```

Der Befehl „showURL“ akzeptiert je nach Betriebssystem unterschiedlich tolerant viele verschiedene URLs. Wir sollten hier also besonders auf die Vollständigkeit der URL achten. „macwelt.de“ oder „info@macsw.de“ reicht nicht aus. Bei der URL müssen wir das Protokoll mit angeben, also „http://www.macwelt.de“ oder „mailto:info@macsw.de“.

## Ein Menüeintrag im Apple Menü

Ein Doppelklick auf das Menü im Projektfenster bringt uns in den Realbasic-Menüeditor. Dort klicken wir einmal auf das Apfelmenü und geben im noch leeren Menüeintrag „Über Mensch ärgere Dich nicht...“ ein. Realbasic weist dem Menüeintrag automatisch einen langen Namen zu, aber wir nennen ihn einfach „AppleAbout“. In unserer Application Subklasse („App“) ergänzen wir einen Menühändler für „AppleAbout“ und rufen dort den Dialog mit folgenden Zeilen auf:

```
aboutWindow.show  
return true
```

Unter Windows gibt es kein Apfelmenü, sondern nur ein Startmenü zum Beenden. Realbasic schiebt die Menüeinträge in diesem Fall an das Ende des letzten Menüs ganz rechts. Bei uns ist das also das Menü „Bearbeiten“. Wir verbessern das, indem wir ein Menü namens „Help“ anlegen. Beim Mac-OS werden alle Einträge in diesem Menü in das jeweilige Hilfe-Menü geschoben, so dass aus „Help“ unter einem deutschen Mac-OS automatisch „Hilfe“ wird. Unter Windows klappt das leider

nicht immer, was man aber mit einer Konstante ändern kann, die dann unter Mac-OS den Wert „Help“ und unter Windows den Wert „Hilfe“ bekommt. Als Text schreiben wir dann in den Menüeintrag ein Doppelkreuz „#“ gefolgt vom Konstantenname, also zum Beispiel „#Hilfemenutitle“.

Damit dieses Hilfemenü aber nicht so allein da steht, bekommt es als weitere Einträge noch „Homepage“ und „Email an den Autor“. Diese Einträge verknüpfen wir dann über die Application-Subklasse und rufen sie wieder mit ShowURL auf. In Realbasic 4 gibt es da einen kleinen Bug, denn die Einträge im Hilfemenü sind in der Entwicklungsumgebung niemals aktiviert, im fertig compilierten Programm klappt jedoch alles wie es soll. Als letztes bleibt uns nur noch ein paar Freunde zu suchen, die das Programm noch einmal auf Herz und Nieren prüfen und dann, wenn wirklich die Version 1.0 final daraus wird, das Programm auf eine Homepage zu legen und es in der Öffentlichkeit bekannt zu machen.

## Fazit

In dieser Serie haben wir die Entwicklung eines Brettspiels in Realbasic von Anfang bis Ende verfolgt und viele neue Programmier-Techniken gelernt. Bei entsprechender Resonanz sind wir gerne bereit in einer der nächsten Ausgaben der *Macwelt* mit einem neuen Projekt zu starten. Schreiben Sie uns! *cm* ✕

## ONLINE...

Sie finden die Projektdateien und die fertig compilierten Programme im Internet unter [www.macwelt.de/\\_magazin](http://www.macwelt.de/_magazin) oder auf der aktuellen Abo-CD. Eine 30-Tage-Demo-Version von Realbasic gibt es unter [www.application-systems.de/realbasic](http://www.application-systems.de/realbasic)