

Highscore-Saver

Ihr kennt sicher die Situation aus verschiedenen Spielesessions - Alleine oder mit Freunden in einer gemütlichen Runde.

Eurer Lieblingspiel besitzt zwar eine Highscore-Liste in der man sich nach erfolgreichem Spiel eintragen kann, man freut sich auch besonders ganz oben auf dem Topplatz zu stehen. Aber nach dem Beenden des Spiels geht der Inhalt der Liste verloren weil sie nicht gespeichert werden kann. Beim nächsten Spieleabend gibt es dann Streitereien wer denn den Highscore hält!

Aus welchem Grund gibt es so viele Spiele die über eine Highscore-Liste verfügen aber keine Speicherung vorsehen? Das hat sicher mit der Problematik "Speichermedium" zu tun, die meisten Spiele gab es in einer Version auf Diskette und Kassette, eventuell sogar auch auf Modul - was die Programmierung natürlich beeinflusst hätte.

Abhilfe kommt in illegaler Form, denn viele bekannte Crackergruppen haben daher neben Trainer und anderen Verbesserungen auch diese sogenannten "Highscore-Saver", im Angebot.

Die Liste wird einfach, nach Eingabe eines Highscores, auf Diskette gesichert und kann, falls gewünscht beim nächsten Start des Spiels wieder geladen werden. Wie Cracker dies nachträglich in ein Spiel einbauen wollen wir uns heute einmal anschauen.

Es gibt sicherlich einige Möglichkeiten dies zu bewerkstelligen. Es müssen allerdings auch einige Parameter bekannt sein.

Wichtige Parameter

- 1) Bereich und Form in dem die Highscoreliste (im Speicher) liegt
- 2) Startadresse des Spiels
- 3) Freier Speicherplatz für unsere zusätzlichen Codes
- 4) Codefragment
"Highscoreeingabe abgeschlossen"

Um jetzt einen einfachen Highscore-Saver zu programmieren brauchen wir zumindest Sourcecode für das Speichern und Laden von Speicherbereichen. Das Laden kann auch ohne Speicherbereich erfolgen da immer an die gespeicherte Adresse geladen wird.

LOAD.SRC

```

    *= $1000 (Startadresse)
INST = $23
SETLFS = $FFBA
SETNAM = $FFBD
LOA = $FFD5
    SEI
    LDA #$00
    STA $d020
    LDA #$01
    LDX #$08
    LDY #$00
    JSR SETLFS
    LDA #FLNM2-FLNM1
    LDX #<FLNM1
    LDY #>FLNM1
    JSR SETNAM
    LDA #$00
    LDX #$18 / $1800 bis
    STA INST
    STX INST+1
    LDA #INST
    LDX #$FF / $18FF
    LDY #$18
    JSR LOA
    LDA $37
    STA $01
    RTS
```

```
FLNM1 .TEXT "Highscore"
```

```
FLNM2 = *
```

SAVE.SRC

```

    *= 1040 (Startadresse)
INST = $23
SETLFS = $FFBA
SETNAM = $FFBD
SAV = $FFD8
```

```

    LDA #$36
    STA $01
    LDA #$01
    LDX #$08
    LDY #$FF
    JSR SETLFS
    LDA #FLNM2-FLNM1
    LDX #<FLNM1
    LDY #>FLNM1
    JSR SETNAM
    LDA #$00
    LDX #$18
    STA INST
    STX INST+1
    LDA #INST
    LDX #$FF
    LDY #$18
    JSR SAV
    LDA $37
    STA $01
    RTS
```

```
FLNM1 .TEXT "@:Highscore"
```

```
FLNM2 = *
```

Beide Codebeispiele liegen im TASS-Format vor (Turbo Assembler).

Zusätzlich sind eventuell noch Verschiebe-Routinen nötig, Routinen um einen Hilfetext anzuzeigen oder die Highscoreliste zu de- und encodieren.

Die wahrscheinlich bekannteste Methode und die wohl auch am meisten genutzte ist die, direkt nach dem Crackintro die letzte Highscore-Liste zu laden und jedesmal nach Eingabe eines Highscores diese Liste mit den neuen Daten zu überschreiben. Hier muss man sich in den Code "Highscoreeingabe abgeschlossen" einklinken und auf die Speicherroutine verzweigen. Hier wird dann immer nach erfolgreichen Highscore gesichert (wenn sich die Diskette im Laufwerk befindet). Es gibt aber auch Gründe das man nur zu bestimmten Zeiten abspeichern möchte, z.B. nur wenn man selber einen Highscore erzielt hat und nicht der gegnerische Spieler. Das kann man durch ein Highscore-Saver-Menu, das durch einen bestimmten Tastendruck bzw. eine Tastendruckkombination (z.B. Shift-F1, sollte vom Spiel nicht benötigt werden) angezeigt wird und in dem man dann seine Highscore speichern oder auch wieder laden kann. Hier muss man dann natürlich diese Tastenabfrage zusätzlich in den Spielecode hinzufügen. Für ganz hartgesottene könnte man auch verschiedene Speicherstände mit unterschiedlichen Namen abspeichern / laden. Hier hat man dann aktiv die Möglichkeit das speichern des Highscore's zu beeinflussen. Der Programmieraufwand steigt aber deutlich an. Da der Speicherplatz aber sehr oft knapp bemessen ist sollte man sich auf das nötigste beschränken. Schauen wir uns mal ein Beispiel mit einem nachträglich installiertem Highscore-Saver an. In der "Neuzeit" machen das eigentlich alle große Gruppen, Anfang/Mitte der Achtziger war das aber eher den absoluten Könnern vorbehalten.

Meine Wahl fiel auf Krakout, ein Spiel was ich gerne mochte und immer noch mag. Die normale Version von "The Commodore Boys" 196 Blocks und New Krakout 4 (mit 100 neuen Level) von "Luke Skywalker" vom 12.07.1987 mit 134 Blocks. Bei genauerem Hinsehen ähneln die Highscore-Dateinamen doch schon gewaltig ("High-Scores/Sky " und "High-Scores/TCB "). Die Save-Routine liegt bei beiden genau an der gleichen Stelle und auch die Routine ist 100%ig die gleiche. Bei der Loadroutine gibts dann im Sourcecode und auch in der Unterbringung Unterschiede. Sicher ist das hier wohl jemand vom anderen abgekupfert hat oder gibt es noch weitere Versionen?

Der Highscore bei Krakout liegt übrigens von \$F800-\$F9E0.

Die Einsprungsadresse liegt bei \$8045 bzw. \$3BD0.

Bei beiden Varianten wird direkt nach dem Intro und Entpacken ohne Rückfragen auf die Loadroutine verzweigt. Falls man einen Highscore erreicht wird auch sofort nach dem Beenden der Eingabe die Highscoreliste mit der neuen überschrieben. Load Highscore (Luke Skywalker)

```

$3EC0 SEI
    LDA#$00
    STA$D020 (schwarzer Border beim
laden ?)
    LDA#$01
    LDX#$08
    LDY#$00
    JSR$FFBA
    LDA#$16 (Länge Filename)
    LDX#$E0
    LDY#$3F (Adresse Filename, $3FE0)
    JSR$FFBD
    LDA#$00
    LDX#$00
    LDY#$F8 (Ladeadresse $F800)
    JSR$FFD5
    JMP$8045 (Startadresse Krakout)

```

speichern liegt bei Luke Skywalkers Version bei

```

$3F00 LDA#$12
    LDY#$3F
    LDX#$DE ($3FDE, Filename mit
vorangestellten "@:")
    JSR$FFBD
    LDA#$01
    LDX#$08
    TAY
    JSR$FFBA
    JSR$FFC0
    LDX#$01
    JSR$FFC9
    LDA#$00
    JSR$FFD2
    STAA$AE
    LDA#$F8
    JSR$FFD2
    STAA$AF
    LDY#$00
    LDA$AE
    CMP#$E0
    BNE$3F36
    LDA$AF
    CMP#$F9
    BEQ$3F50
    NOP
    ...

```

und so weiter, und so fort - vergleicht auch mit dem TASS Code-Beispielen von oben.

Den Rest könnt ihr euch ja selber erarbeiten. Beide Versionen liegen dem Magazin als Diskette bei.

Abschließend kann man sagen das es keine Hexerei ist, aber man muss den Code doch sehr genau nachverfolgen und auch lesen können. Die programmiertätigkeit nimmt eher einen untergeordneten Punkt ein und da der Platz für den zusätzlichen Code in den meisten Fällen kaum reicht, ist hier der "Monitor-Coder" im Vorteil. Er kann seine Codehappen auf den zu Verfügung stehenden Platz verteilen....