

2018 BASIC 10-Liner PUR-80

„2nd Stroop“ by Christian „Irgendwer“ Krüger

Vorwort:

John Ridley Stroop war ein US-amerikanischer Psychologe nach dem der so genannte „Stroop-Effekt“, benannt ist. Dieser „ist ein experimentalpsychologisches Phänomen, das bei mentalen Verarbeitungskonflikten auftritt. Er zeigt, dass trainierte Handlungen nahezu automatisch ablaufen, während ungewohnte Handlungen eine größere Aufmerksamkeit benötigen.“

(<https://de.wikipedia.org/wiki/Stroop-Effekt>)

Das klassische Experiment arbeitet dabei mit Farbwörtern, die nicht der Druckfarbe des Wortes entsprechen. Wer meinen Beitrag zum Zehnzeilerwettbewerb aus dem Jahr 2014 kennt, weiß, dass genau dieses Prinzip die Grundlage für mein Spiel „Colfusion“ ist. Neben diesem Test gibt es jedoch noch weitere: Der „Physische oder numerischer Zahlenstroop-Test“ bildet dabei die diesjährige Basis meines Beitrags.

Das Spiel:

Ziel des Spiels ist es, innerhalb von zwei Minuten möglichst viele Zahlen nach ihrem Wert auszuwählen. Zwei einstellige Zahlen werden in einer Runde angezeigt, der Spieler hat nun die Aufgabe die numerisch größere Zahl zu bestimmen.

Befindet sie sich links, ist die „1“-Taste zu benutzen, die rechte Zahl kann mit einem Druck auf die „2“ selektiert werden. Der oben genannte Verarbeitungskonflikt entsteht durch unterschiedliche Schriftgrößen, die nicht im Zusammenhang mit dem Wert der Zahl stehen. Für eine richtige Wahl gibt es einen Punkt, wird jedoch die falsche Zahl ausgesucht werden 5 Punkte abgezogen. Anschließend beginnt eine neue Runde, bis die zwei Minuten vorüber sind. Der Schluss des Spiels wird mir der „Systemglocke“ signalisiert und der Höchstpunktestand angezeigt.

Wie auch schon bei „Colfusion“ habe ich versucht ein Spiel zu erstellen, bei dem keine offensichtlichen Mängel oder Verbesserungen auf Grund der Programmgröße ersichtlich sind.

Das Programm:

Dieses Spiel in 10 Zeilen mit diesmal nur 80 Zeichen pro Zeile umzusetzen war eine besondere Herausforderung. Trotzdem ist es mir gelungen eine Tonausgabe einzubauen sowie die schon von „Colfusion“ bekannte Unterstützung für PAL- und NTSC-Systeme (Anpassung der Zeitbasis) umzusetzen.

Um die Vorgabe der „PUR-80“-Kategorie zu erfüllen, hat die Programmaufteilung etwas gelitten: Programmzeilen sind nicht immer gleich lang, Ausgleich musste an Stellen erfolgen, der nicht immer sofort logisch nachzuvollziehen ist. Ich habe mir daher die Mühe gemacht ein Flussdiagramm zum Ablauf zu erstellen.

Das beigefügte „ATR“ enthält das Programm als „AUTORUN.BAS“ und muss mit angeschaltetem Basic gebootet werden. Das Spiel startet automatisch.

(In einigen Bereichen verlässt sich das Programm auf die initialen Werte des Systems, so wird z.B. das Prioritätsregister nicht extra gesetzt. Im Zusammenspiel mit einem „Loader/Menu“ der diese Werte verändert, kann ein problemloser Betrieb nicht garantiert werden!)

Nachfolgend der Quelltext als Beleg, dass die Kriterien der Kategorie „PUR-80“ erfüllt sind, sowie eine etwas schönere Formatierung des gleichen Programms mit Erklärung der - durch die Wettbewerbsbedingungen sehr kurz geratenen – Variablennamen.

Ich wünsche viel Spaß beim Spielen
Christian Krüger

Der Quelltext (max. 80-Zeichen pro Zeile mit Abkürzungen), in den Atari eingetippt:

```
11111111112222222222333333333344444444445555555555666666666677777777778
1234567890123456789012345678901234567890123456789012345678901234567890
0GR.0:FOR I=1 TO 6:REA.A,V:POK.A,V:N.I:REA.H,C:B=127:D=C+1:GOS.9:T=764:POS.15,1
1?"2nd Stroop":V=(PEEK(H+20)=1):TL=224-V*112:TH=0+V*5:HI=-999:X=255:B=6:POS.8,3
2?"Which value is greater?":POS.7,4:"(2 min.-enter '1' or '2')":D=33330:Z=57473
3P=0:POS.18,14:"[Key]":POK.T,X:F.I=0 TO 1:I=(PEEK(T)<>X):N.I:POK.20,TL:POK.19,TH
4POS.17,19:"Pts.":P;"":F=INT(RND(0)*10):S=F+INT(RND(0)*8)+1:S=S-10*INT(S/10)
5L=INT(RND(0)*2):POK.H,106+36*L:POS.24-9*L,9:F:C=Z+S*8:GOS.9:POK.T,X:S0.0,0,0,0
6IF PEEK(19)>28 THEN HI=(P>HI)*P+(P<HI)*HI:POS.18,20:"{253}Hi:":HI;"":G.3:D.752,1
7K=PEEK(T):IF K=X THEN G.6:D.54279,128,53277,2,559,42,33280,0,704,154,53248,33280
8R=((F>S)AND(K=(30+L)))OR((F<S)AND(K=(31-L))):P=P+R+(R-1)*5:S0.0,200-P,10,10:G.4
9POS.15+9*L,9:"":POS.18,14:"":F.I=0 TO B:POK.D+I,PEEK(C+I):N.I:RET.:REM CK
```

Die besser lesbare Version:

Initialisation

```
0 GR.0:FOR I=1 TO 6:REA.A,V:POK.A,V:N.I:REA.H,C:B=127:D=C+1:GOS.9:T=764:POS.15,1
1 ?"2nd Stroop":V=(PEEK(H+20)=1):TL=224-V*112:TH=0+V*5:HI=-999:X=255:B=6:POS.8,3
2 ?"Which value is greater?":POS.7,4:"(2 min.-enter '1' or '2')":D=33330:Z=57473
```

Game loop

```
3 P=0:POS.18,14:"[Key]":POK.T,X:F.I=0 TO 1:I=(PEEK(T)<>X):N.I:POK.20,TL:POK.19,TH
```

Round loop: New random numbers

```
4 POS.17,19:"Pts.":P;"":F=INT(RND(0)*10):S=F+INT(RND(0)*8)+1:S=S-10*INT(S/10)
5 L=INT(RND(0)*2):POK.H,106+36*L:POS.24-9*L,9:F:C=Z+S*8:GOS.9:POK.T,X:S0.0,0,0,0
```

Round loop: Time out?

```
6 IF PEEK(19)>28 THEN HI=(P>HI)*P+(P<HI)*HI:POS.18,20:"{253}Hi:":HI;"":G.3:D.752,1
```

Round loop: Key press?

```
7 K=PEEK(T):IF K=X THEN G.6:D.54279,128,53277,2,559,42,33280,0,704,154,53248,33280
```

Evaluate key press

```
8 R=((F>S)AND(K=(30+L)))OR((F<S)AND(K=(31-L))):P=P+R+(R-1)*5:S0.0,200-P,10,10:G.4
```

Subroutine "memcpy/move" (together with some screen clearing as here was room)

```
9 POS.15+9*L,9:"":POS.18,14:"":F.I=0 TO B:POK.D+I,PEEK(C+I):N.I:RET.:REM CK
```

A = ADDRESS
V = VALUE
I = ITERATOR

B = BLOCK SIZE
C = SOURCE ADDRESS
D = DESTINATION ADDRESS

F = FIRST VALUE
S = SECOND VALUE
L = DISPLAY SECOND VALUE LARGER
R = RIGHT (CORRECT)
P = POINTS
HI = HIGH SCORE

K = KEYBOARD VALUE
T = KEYBOARD SYS ADDRESS
X = 255

V = VIDEO SYSTEM (NTSC=0,PAL=1)
Z = ADDRESS SYSTEM FONT: CHAR ZERO
H = HPOSPO (53248)
TL = SYSTEM TIMER LOW
TH = SYSTEM TIMER HIGH

DATA:

CURSOR OFF; PMBASE,128; GRCTL,2; SDMCTL,42; PO,0; PCOLRO LIGHTBLUE(154),
HPOSPO Address, Player 0 Address