

```

25181 IF SP=1 THEN FOR I=1 TO AS:N1$(I)=N$(
(I):K1(I)=SC(I):NEXT <228>
25205 QH=HI(SG) <223>
25210 FOR NN=1 TO AS:IF K1(NN)>HI(SG) THEN
HI(SG)=K1(NN):HN$(SG)=N1$(NN) <254>
25220 NEXT <082>
25250 PRINT" {HOME,7DOWN}";:FOR I=1 TO 5 <188>
25260 PRINT" {DOWN,6RIGHT}"HN$(I);HI(I):NEX
T <061>
25300 IF QH=HI(SG) THEN 25400 <203>
25310 OPEN 1,8,1,"@:HI-1" <009>
25320 I=1 <121>
25330 PRINT#1,HN$(I) <111>
25340 PRINT#1,HI(I) <119>
25350 I=I+1:IF I<6 THEN 25330 <173>
25360 CLOSE 1 <225>
25400 FOR J=1 TO 9 <108>
25410 FOR I=1 TO 9 <110>
25420 IF K2(I)<K2(I+1) THEN V1=K2(I):K2(I)=
K2(I+1):K2(I+1)=V1:GOSUB 25450 <072>
25430 NEXT:NEXT <031>
25440 GOTO 25460 <091>
25450 V1$=N2$(I):N2$(I)=N2$(I+1):N2$(I+1)=
V1$:RETURN <005>
25460 PRINT" {HOME,7DOWN}";:FOR I=1 TO 5 <144>
25470 PRINT" {DOWN,25RIGHT}"N2$(I);K2(I):NE
XT <177>
25500 IF SP<2 THEN 30000 <219>
26050 OPEN 1,8,1,"@:HI-2" <116>
26060 I=1 <099>
26070 PRINT#1,N2$(I) <018>
26075 PRINT#1,K2(I) <246>
26080 I=I+1:IF I<6 THEN 26070 <078>
26085 CLOSE 1:SP=0:TT=0:CC=0:FOR KO=1 TO 1
0:T(KO)=0:SC(KO)=0:T1(KO)=0:S1(KO)=0
:NEXT <009>
30000 : <002>
30005 FOR I=1 TO 48:GOSUB 40500:IF PEEK(56
320)<112 THEN POKE CO+18,0:GOTO 6300 <094>
30010 NEXT:POKE 54276,16:POKE 54283,128:PO
KE 54290,32:POKE 54296,0:SYS 37233:G
OTO 6006 <033>
40000 POKE CO,207:POKE CO+1,34:POKE CO+5,9
:POKE CO+6,0 <044>
40010 IF JJ=1 THEN POKE CO,157:POKE CO+1,6
9:POKE CO+5,12:POKE CO+6,240 <049>
40020 POKE CO+24,15:POKE CO+4,33 <033>
40030 FOR D=1 TO 1000:NEXT <026>
40040 POKE CO+4,32:RETURN <120>
40500 SYS 49850:FOR DD=1 TO 267:NEXT:RETUR
N <162>

```

© 64'er

Listing 4. »Tonti« (Schluß)

Billard

Dieses Programm simuliert die Bewegungen von bis zu sieben Billardkugeln in Realzeit physikalisch richtig und im Rahmen der möglichen Bildschirmauflösung sehr genau.

Der C64 verfügt mit den Sprites über hervorragende Möglichkeiten, grafische Figuren besonders durch Maschinenbefehle sehr schnell auf dem Bildschirm zu bewegen. Es stellte sich aber heraus, daß die CPU auch nach Ausschöpfung aller programmtechnischen Möglichkeiten mit Bewegung und Stoßauswertung so ausgelastet ist, daß eine realistische Effet-Behandlung in Realzeit nicht mehr möglich ist (für Interessierte, die auf einem anderen Rechner ähnliches realisieren möchten, sei darauf hingewiesen, daß aber noch genügend Zeitreserven für eine Programmlösung auch ohne Sprites vorhanden sind!).

Trotz dieser Einschränkung läßt sich aber ein Billardspiel gut simulieren.

Vor dem eigentlichen Programm jedoch noch einige prinzi-

pielle Bemerkungen zum wichtigsten Teilproblem, der Stoßberechnung.

Da sich alle Kugeln zu jeder Zeit, also insbesondere auch, wenn ein Stoß auszuwerten ist, gleichmäßig bewegen sollen, wird durch diese Rechnung entscheidend die Schnelligkeit der Kugelbewegungen bestimmt. Wie so oft liegt hier die Schwierigkeit weniger in der programmtechnischen Ausführung als in der rechnergerechten Formulierung des Problems.

Ein kleiner Ausflug in die Physik

Eine Geschwindigkeit ist bekanntlich ein Vektor, das heißt, sie wird bestimmt durch einen Betrag und durch eine Richtung. Ein Vektor läßt sich grafisch gut als Pfeil darstellen, dessen Länge hier für die Größe der Geschwindigkeit steht, während die Spitze beziehungsweise der Winkel mit der Horizontalen die Richtung angibt (siehe Bild 1).

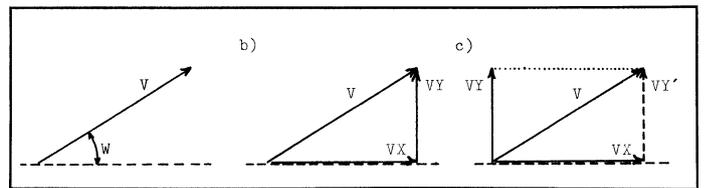


Bild 1. Vektordarstellung durch Winkel und Komponenten

Für grafische Betrachtungen dürfen wir in diesem Zusammenhang Vektoren beliebig verschieben, nicht aber drehen.

Wie in Bild 1 b) gezeigt, kann eine Geschwindigkeit V in ihre Komponenten in x-Richtung (VX) und in y-Richtung (VY) zerlegt werden und ist durch diese auch eindeutig bestimmt.

Bild 1 c) verdeutlicht, wie sich aus den Komponenten VX und VY wieder die Geschwindigkeit V konstruieren läßt. Dazu braucht lediglich VY bis zur Spitze von VX verschoben zu werden, und die Verbindung des Fußpunktes von VX mit der Spitze des verschobenen VY (VY') stellt V dar. Ein solches Aufspalten und Zusammenfügen von Vektorkomponenten bezüglich zweier Richtungen ist immer möglich, wenn die Richtungen aufeinander senkrecht stehen. Bei der Stoßbetrachtung werden uns später zwei andere Richtungen interessieren, wo die Zerlegung dann nach dem gleichen, hier in Bild 1 beschriebenen Prinzip, durchgeführt wird.

Die Aufspaltung in x- und y-Komponenten paßt sehr gut zur Art der Spritebewegung, so daß es naheliegt, alle vorkommenden Geschwindigkeiten so zu zerlegen und VX und VY getrennt zu behandeln. Damit wird bei der Bewegung das Rechnen mit Winkeln vermieden.

Zur Stoßberechnung selbst bieten sich zunächst die Energie- und Impuls-Erhaltungssätze an (Mancher wird sich jetzt wohl an den Physikunterricht erinnern). Diese allein liefern aber noch keine eindeutige Lösung, es ist noch eine zusätzliche Annahme erforderlich.

In Bild 2 sind zwei sich berührende Kugeln dargestellt, die

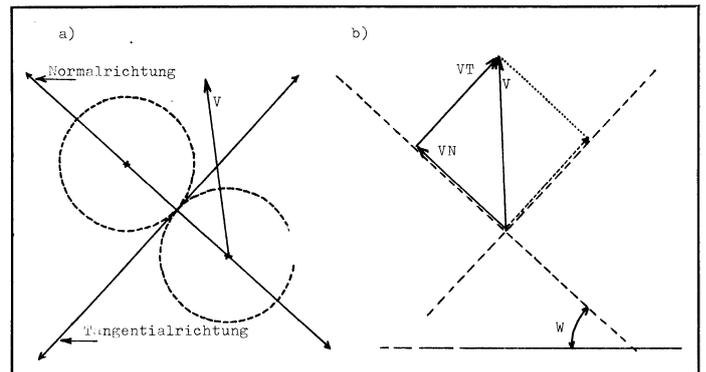


Bild 2. Stoß zweier Kugeln

gleiche Größe und gleiches Gewicht haben sollen. Außerdem sind zwei ausgezeichnete Richtungen angedeutet, die senkrecht aufeinander stehen: die sogenannte Tangential- und die Normalrichtung. Denken wir uns nun die Kugelgeschwindigkeiten ähnlich wie oben in einen Anteil von Tangential- (VT) und einen Anteil in Normal- (VN) Richtung zerlegt, dann gilt folgende Regel:

Bei einem Stoß bleibt für jede Kugel deren Geschwindigkeitsanteil in Tangentialrichtung nach Betrag und Richtung erhalten, während jede Kugel die Normalkomponente der anderen Kugel übernimmt.

Mit dieser Annahme lassen sich nun die Kugelgeschwindigkeiten nach einem Stoß durch Zerlegen der vorherigen Geschwindigkeiten in VT und VN und entsprechendes Zusammenfügen dieser Komponenten zeichnerisch einwandfrei bestimmen.

Für das Programm sind aber mathematische Gleichungen erforderlich. Dazu müssen die Tangential- und Normalgeschwindigkeiten jede für sich noch einmal in einen x-Anteil (VXT, VXN) und einen y-Anteil (VYT, VYN) zerlegt werden:

$$\begin{aligned} VXT &= (VX * SIN(W) - VY * COS(W)) * SIN(W) \\ VXN &= (VY * SIN(W) + VX * COS(W)) * COS(W) \\ VYT &= -(VX * SIN(W) - VY * COS(W)) * COS(W) \\ VYN &= (VY * SIN(W) + VX * COS(W)) * SIN(W) \end{aligned}$$

Der Winkel W ist der Winkel zwischen Normalenrichtung und der Horizontalen (siehe Bild 2); er läßt sich aus den Koordinatendifferenzen der Kugelpositionen bestimmen.

Dann ergeben sich die resultierenden Kugelgeschwindigkeiten nach dem Stoß beziehungsweise deren x- und y-Komponenten:

$$\left. \begin{aligned} VX1 &= vxn2 + vxt1 \\ VY1 &= vyn2 + vyt1 \end{aligned} \right\} \text{für Kugel 1}$$

$$\left. \begin{aligned} VX2 &= vxn1 + vxt2 \\ VY2 &= vyn1 + vyt2 \end{aligned} \right\} \text{für Kugel 2}$$

Für die sich stoßenden Kugeln 1 und 2 sind die Geschwindigkeiten vor dem Stoß klein, nach dem Stoß groß geschrieben.

Mit einiger Knobelei habe ich die Gleichungen auf diese Form gebracht, die den Vorteil hat, daß alle vorkommenden Summanden, Klammerausdrücke und Zwischenergebnisse nicht größer werden können als die Ausgangsgröße V.

Werden für V dann nur Werte zwischen 0 und 127 zugelassen, so können die Rechnungen mit positiven und negativen Ein-Byte-Zahlen durchgeführt werden, was, wie Sie aus dem laufenden Assemblerkurs wissen, besonders einfach und schnell geht.

Geschwindigkeit durch Tabellen

Bei diesen schönen Ein-Byte-Rechnungen stören aber noch die vielen »*SIN« und »*COS«, die viel zuviel Rechenzeit beanspruchen. Es bleibt nur die Möglichkeit, mit Tabellen zu arbeiten, wobei aber nicht alle benötigten Werte gespeichert werden müssen. Zunächst ergeben sich in jedem Viertelkreis immer die gleichen SIN- und COS-Werte, nur mit unterschiedlichem Vorzeichen. Außerdem ist der COS symmetrisch zum SIN:

COS(W) = SIN(W+90 Grad). Ich unterteile also den Viertelkreis in 23 Winkelteile und lege die zugehörigen SIN-Werte im Fließkommaformat in einer Tabelle ab.

Nun stört nur noch die Multiplikation. Da wie oben beschrieben nur Faktoren von 0 bis 127 vorkommen, ist auch hier eine Tabelle noch vertretbar. So werden nach Start des Programmes zu jedem SIN-Wert 128 Ein-Byte-Zahlen in eine Multiplikationstabelle geschrieben: (Byte 0 bis 127) * SIN.

Das sind jetzt zwar schon eine ganze Menge Bytes, dafür wird das Programm aber auch kürzer und sehr schnell.

Durch geeignetes Setzen von Zeigern können alle Werte »Byte * SIN« und »Byte * COS« nun ohne weitere Rechnung direkt abgerufen werden. Es bleibt nur noch die schnelle Ein-Byte-Addition und -Subtraktion. Die gewählte Winkel- und Geschwindigkeitsauflösung ist im Rahmen der Bildschirmauflösung voll ausreichend.

Da hier Winkel durch das Verhältnis der Geschwindigkeiten VX und VY dargestellt werden, ist klar, daß die Genauigkeit bei sehr kleinen Geschwindigkeiten geringer wird. Bei einer Gesamtgeschwindigkeit von beispielsweise 2 kann der Winkel eben nicht mehr so genau dargestellt werden wie mit höheren Werten, die viel mehr Kombinationen VY/VX zulassen.

Aus diesem Grund dürfen die Kugeln auch nicht rollen bis ihre Geschwindigkeit Null geworden ist, sondern die Bewegung wird schon etwas früher abgebrochen.

Am Schluß dieser Betrachtungen noch zwei Beispiele, an denen Sie bei Billardprogrammen einmal die Richtigkeit der Stoßberechnung überprüfen können:

Die Verhältnisse sind besonders einfach, wenn eine bewegte Kugel eine ruhende stößt. Erfolgt der Stoß zentral, so muß die stoßende Kugel anschließend stillstehen, während die andere Kugel die Geschwindigkeit voll aufgenommen hat. Bei nichtzentralen Stößen müssen die Richtungen, in die sich die zwei Kugeln nach dem Stoß bewegen, einen rechten Winkel einschließen.

Tips zum Eintippen

Es besteht aus zwei Teilen, dem eigentlichen Billardprogramm, das in Maschinencode geschrieben ist und am Basic-Anfang bei Adresse 2048 beginnt, und einem Basic-Teil im Speicher ab 4973, in dem die Spielregeln definiert werden.

Die ersten Bytes des Maschinenprogrammes entsprechen einer Basic-Zeile: »1 SYS 2061«. Beim Start des Programmes findet der Basic-Interpreter diesen SYS-Befehl und startet das Maschinen-Programm bei 2061. Dort wird der Basic-Anfang auf 4973 hochgesetzt, die oben beschriebene Multiplikationstabelle erstellt und zum Basic-Teil gesprungen, von dem aus dann die weiteren Maschinen-Routinen aufgerufen werden.

Auf diese Weise ist das Maschinen-Programm so eingebunden, daß Sie das gesamte Programm später wie ein gewöhnliches Basic-Programm ohne Zeigerumstellungen handhaben können.

Da der MSE die gleiche Methode benutzt, also auch ab 2048 im Speicher liegt, wird die Eingabe des Programmes etwas umständlich.

Verfahren Sie daher am besten folgendermaßen:

1. Mit dem MSE (siehe Seite 8) den Maschinenteil ab \$C000 eingeben und unter »BILL(MP)« abspeichern.
2. Mit dem Checksummer (Seite 6) das Basic-Programm eingeben und unter »BILL(BASIC)« abspeichern.
3. Beide Programmteile binden:
 - a) »BILL(MP)« absolut laden (»LOAD "BILL(MP)",8,1« oder »...,1,1«)
 - b) Basic-Anfang auf 4973 setzen mit Eingabe im Direktmodus »POKE 43,109: POKE 44,19: POKE 4972,0: NEW«

- c) Maschinen-Programm im Direktmodus übertragen
»FOR I=49152 TO 52076: POKE I-47104, PEEK(I): NEXT«
- d) Basic-Programm laden mit »LOAD "BILL(BASIC)",8« (beziehungsweise »...,1« für Datasette)
- e) Basic-Anfang zurückstellen und Gesamtprogramm speichern mit »POKE 43,1: POKE 44,8: SAVE "BILLARD",8« (beziehungsweise »...,1«)

Im Programm »BILLARD« sind nun beide Teile enthalten. Es läßt sich wie ein übliches Basic-Programm nach Einschalten des C 64 laden und starten.

Die Bedienung des Programms erfolgt bei der Spielwahl über die Tastatur; Schlagwinkel und Schlagstärke werden über Joystick an Port 2 eingegeben. Die Spielregeln sollen hier nicht näher erklärt werden; Sie werden sie während des Spiels schnell herausfinden.

Änderungen erwünscht!

Betrachten Sie den Basic-Teil von »BILLARD« lediglich als Vorschlag und Anregung zu einem eigenen Programm.

Das Maschinenprogramm ist so modular und variabel aufgebaut, daß Sie leicht ein Billardspiel nach eigenem Geschmack und mit eigenen Regeln realisieren können (zum Beispiel ein Lochbillard mit fünf Kugeln unter Berücksichtigung von Stößen).

Zur Erkennung eines Stoßes oder einer Bandenberührung werden die entsprechenden Sprite-Register **nicht** benutzt, wodurch Sie auch in der grafischen Gestaltung völlig frei sind.

Die drei Tongeneratoren werden vom Programm vor jedem Stoß auf die benötigten Werte eingestellt. Sie müssen nur von Basic aus das Register 54296 für die Lautstärke einstellen (=15).

- Generator 1: Bandenberührung
 - Generator 2: Stoßgeräusch
 - Generator 3: Ton, wenn eine Kugel in ein Loch fällt
- Alle drei Generatoren werden mit der Dreieck-Wellenform »16/17« geschaltet.

1. Veränderbare Programmparameter

Der Bewegungsablauf kann durch drei Parameter beeinflusst werden.

- a) **Byte 3687:** Dämpfung (hier = 10)
Ein größerer Wert bedeutet hier schwächere Dämpfung; die Kugeln rollen dann bei gleicher Schlagstärke weiter. Ein kleinerer Wert bewirkt stärkere Dämpfung.
- b) **Byte 4200:** Ende der Kugelbewegung (hier = 2)
Wie schon oben bei der Stoßauswertung beschrieben, ist es sinnvoll, die Kugelbewegung abubrechen, bevor die Geschwindigkeit auf 0 abgesunken ist. Ein zu kleiner Wert an dieser Stelle hat vor allem bei schwacher Dämpfung ein Abknicken der Bewegungsrichtung kurz vor Stillstand zur Folge.
- c) **Byte 4248:** Verzögerung des Bewegungsablaufes (hier = 10)

Eine automatische Geschwindigkeitsanpassung abhängig von der Anzahl der rollenden Kugeln ist an anderer Stelle eingebaut; sie sorgt für gleichmäßige Bewegung, gleichgültig, ob sich gerade nur eine oder sieben Kugeln bewegen.

An dieser Stelle bewirkt ein größerer Wert zusätzliche Verzögerung. Sie können sich beispielsweise mit »255« einen Stoß in Zeitlupe ansehen. Während mit der Dämpfung die zurückzulegende Strecke bestimmt wird, läßt sich hier die Geschwindigkeit der Bewegung festlegen.

Experimentieren Sie ruhig etwas mit diesen Parametern. Nach einem entsprechenden POKE sehen Sie die Wirkung gleich beim nächsten Stoß.

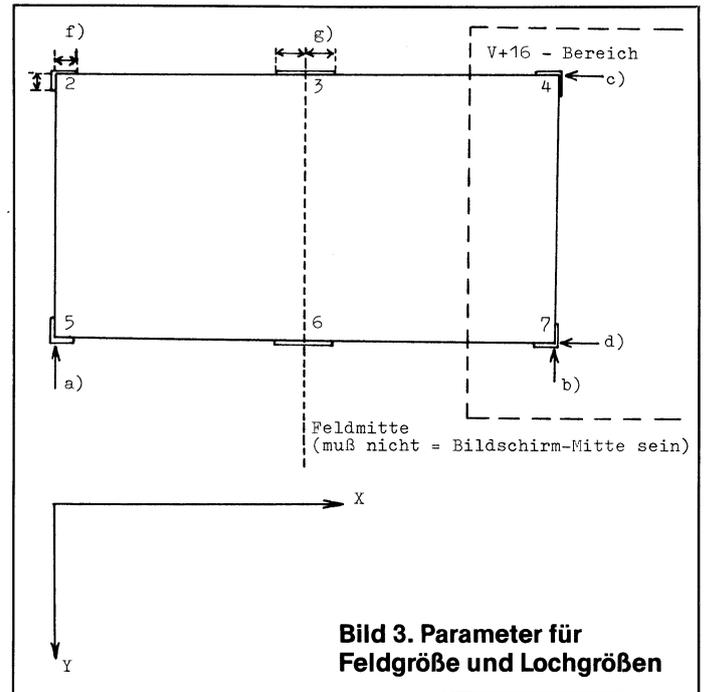


Bild 3. Parameter für Feldgröße und Lochgrößen

Wenn Sie nach Änderungen hier das Programm wieder neu abspeichern, werden die neuen Werte mitgespeichert. Sie können selbstverständlich aber auch während des Programmlaufes die Parameter mehrfach ändern.

2. Parameter für Feld- und Lochgrößen

Sie sind in Bild 3 mit den hier verwendeten Bezeichnungen dargestellt.

- a) **Byte 3204:** Linker Feldrand (hier = 20)
- b) **Byte 3205:** Rechter Feldrand (hier = 56)
- c) **Byte 3206:** Oberer Feldrand (hier = 73)
- d) **Byte 3207:** Unterer Feldrand (hier = 73)
- e) **Byte 3208:** (hier = 3) Dieser Wert legt fest, um wieviel Grafikpunkte sich der Cursor (zur Einstellung der Stoßrichtung) über den Feldrand hinaus bewegen lassen soll. Dieser Cursor — hier ein Zielkreuz (Sprite 0) — ist übrigens in Sprite-Block 39 definiert und kann durch Eingabe entsprechender Daten in 2496 bis 2559 geändert werden.
- f) **Byte 3209:** Größe der Ecklöcher (hier = 6)
- g) **Byte 3210:** Größe der mittleren Löcher (hier = 10)

Es gilt hier die Einschränkung, daß der rechte Feldrand sowie die rechten Ecklöcher im Bildschirmbereich des V+16-Registers (53264) liegen müssen, während die anderen Werte nicht in diesem Bereich liegen dürfen (für den rechten Feldrand ist unter b natürlich das Low-Byte einzugeben).

Als kleine Hilfe: Die Kugeln bleiben voll sichtbar im Bereich X zwischen 12 und 64, Y zwischen 41 und 229.

Auch diese Parameter stehen im Programm und werden bei erneutem Abspeichern mit übernommen. Allerdings sehen Sie Änderungen hier nicht gleich beim Programmlauf, da diese Werte erst beim Programmstart abgerufen werden. Ein Unterprogramm errechnet daraus die im Programm benötigten Daten und schreibt sie an die richtige Stelle. Wollen Sie diese Parameter während des Programmes ändern, rufen Sie nach den entsprechenden POKes das Maschinen-Programm »SE« (siehe später) auf.

Wenn Sie das geänderte Programm abspeichern, vergessen Sie nicht, vorher den Basic-Anfang zurückzustellen mit:

»POKE 43,1: POKE 44,8«! Andernfalls wird möglicherweise nur das Basic-Programm gespeichert.

Wie das Zielkreuz (Sprite 0, Block 39) stehen auch die Sprite-Daten der Kugeln im Maschinenprogramm in den Sprite-Blöcken 40 bis 47, zwischen denen zur Simulation einer Rollbewegung hin und her geschaltet wird.

Jeder Kugel ist entsprechend ihrer Sprite-Nummer eine Zahl zwischen 1 und 7 zugeordnet. Ebenso sind die Löcher, wie in Bild 3 gezeigt, durchnummeriert von 2 bis 7.

Nach jedem Stoß stehen Ihnen folgende Informationen zur Auswertung zur Verfügung:

a) Stoßregister \$FD (= Basicvariable SR = 253):

Hier sind die Kugeln angegeben, die mindestens einmal mit der Stoßkugel kollidiert sind, und zwar im Format »2 hoch Kugelnummer 1 bis 7«. Dieses Format entspricht dem, das Ihnen beispielsweise vom Sprite-Sprite Berührungsregister her bekannt ist.

b) Lochregister \$FE (= Basicvariable LR = 254):

Hier finden Sie im gleichen Format wie beim Stoßregister die Kugeln, die in ein Loch gefallen sind.

c) Zur weiteren Auswertung ist für jede Kugel ein zusätzliches Lochregister vorhanden: 52123 + Kugelnummer 1 bis 7. Dieses enthält »2 hoch Lochnummer« des Loches, in das die entsprechende Kugel gefallen ist. Liegt sie noch auf dem Feld, steht hier 0.

Alle Register a) bis c) werden vor jedem Stoß automatisch auf 0 zurückgesetzt.

Zum besseren Verständnis folgen einige Beispiele:

Zu a): IF PEEK(SR) = 12 THEN »Stoßkugel kollidierte mit den Kugeln 2 und 3« ($2^2 + 2^3 = 12$)

Zu b): IF PEEK(LR) AND 8 THEN »Kugel 3 gefallen«
IF PEEK(LR) AND 84 THEN »Kugel mit gerader Nummer, also Kugel 2, 4 oder 6 gefallen«

Zu c): IF PEEK(52123+K) = 64 THEN »Kugel K in Loch 6«
IF PEEK(52123+K) AND 28 THEN »Kugel K in einem der oberen Löcher, also Loch 2, 3 oder 4«

Die Maschinensprache-Routinen

Es folgt nun eine Kurzbeschreibung der Maschinenroutinen: Hinter SYS steht die im Basic-Programm benutzte Variable, die zugehörigen Adressen sind später aufgeführt.

SYS SE: Setzt die Feld- und Lochparameter direkt in das Programm. Die Routine braucht nur aufgerufen zu werden, wenn diese Parameter während des Programmlaufes verändert worden sind.

SYS IN, Anzahl der Kugeln 1 bis 7 + 0 oder + 128: Diese Routine muß einmal vor Beginn eines neuen Spieles aufgerufen werden. Sie definiert für ein Spiel die Anzahl der Kugeln, wobei dann beispielsweise mit einer Kugelzahl von 3 die Kugeln 1, 2 und 3 im Programm berücksichtigt werden. Zusätzlich wird durch Bit 7 festgelegt, ob Lochbillard (Bit 7 = 1, also + 128) oder Bandenbillard (Bit 7 = 0) gespielt werden soll.

SYS AL, Kugelnummer, X-Koordinate (2 Byte 0 bis 511), Y-Koordinate: Hierdurch wird eine Kugel auf das Feld gelegt, wobei die Register V+21 und V+16 automatisch entsprechend eingestellt werden. Wird eine Kugel so gesetzt, daß sie eine andere berühren würde, wird sie so weit wie nötig zur Feldmitte hin verschoben. Für regelgerechte Bewegungen muß die Kugel selbstverständlich in das Spielfeld gesetzt werden.

SYS VL, Kugelnummer: Legt eine Kugel vor dem Loch wieder auf, wenn sie gefallen ist, andernfalls hat der Aufruf keine Wirkung. Im Fall einer Berührung mit anderen Kugeln wird wie bei »AL« verschoben. Sie können diese Routine beispielsweise benutzen, wenn eine Kugel in ein verbotenes Loch gefallen ist.

SYS KS, Nummer der Stoßkugel: Dieser Aufruf muß vor

jedem Stoß erfolgen. Es wird hier die Stoßkugel festgelegt und außerdem das Kreuz auf diese Stoßkugel gesetzt.

SYS JS: Nach diesem Aufruf kann mit Hilfe des Joysticks an Port 2 das Zielkreuz bewegt werden. Nach »FIRE« erfolgt Rücksprung in Basic, wo nun die Geschwindigkeits-Eingabe folgen sollte.

SYS GS, Anfangsgeschwindigkeit 0 bis 127: Für die Anfangsgeschwindigkeit der Stoßkugel sind hier Werte zwischen 0 und 127 erlaubt. Nach Auswertung der Geschwindigkeits-Eingabe beginnt die Bewegung der Kugeln. Rücksprung in Basic erfolgt erst, wenn alle Kugeln wieder in Ruhe sind, vorher ist ein Aussprung nur mit »STOP / RESTORE« möglich.

SYS N1, Anfangsgeschwindigkeit: Wählen Sie statt »GS« diesen Einsprungspunkt, so werden vor der Bewegung alle Sprite-Positionen und die Nummer der Stoßkugel in einem Speicherbereich 1 abgelegt. Da damit alle wichtigen Parameter vor einem Stoß erhalten bleiben, wird eine Stoßwiederholung ermöglicht.

SYS SW: Wählen Sie diesen Einsprungspunkt an Stelle von »JS«, und haben Sie den vorherigen Stoß mit »N1« durchgeführt, werden alle Sprites auf die dort gespeicherten Werte zurückgesetzt, die ja den Zustand vor dem letzten Stoß wiedergeben.

Außerdem werden die jetzigen Positionen in einem anderen Speicherbereich 2 abgelegt. Das Programm fährt fort bei »JS« mit der Joystickabfrage.

Wählen Sie dann für die Geschwindigkeitseingabe bei der Schlagwiederholung den Einsprungspunkt »GS«, haben Sie nach diesem wiederholten Stoß die Wahl zwischen drei Möglichkeiten:

Einmal können Sie den vorherigen Stoß mit »SW« und »GS« noch einmal wiederholen. Sie können aber auch mit »JS« und »N1« vom jetzigen Zustand aus weiterspielen. Für die dritte Möglichkeit, die ich in meinem Programm gewählt habe, fehlt noch eine Routine:

SYS V2: Hierdurch werden die Kugeln auf die in Speicherbereich 2 abgelegten Werte gesetzt, die ja den Zustand nach dem ursprünglichen Stoß wiedergeben. Fährt man danach mit »JS« und »N1« fort, ist die Schlagwiederholung sozusagen »außer Konkurrenz« durchgeführt worden, da ja vom Zustand nach dem ursprünglichen Stoß aus weitergespielt wird.

SYS ZS, 2 hoch Kugelnummer (= 2 bis 254); Geschwindigkeit, Winkel: Diese letzte Routine benutze ich, um einen zufälligen Anfangszustand vor Beginn eines Spieles zu erzeugen. Es lassen sich eine bis sieben Kugeln gleichzeitig mit der gleichen Anfangsgeschwindigkeit zwischen 0 und 127 anstoßen. Die Bewegungsrichtung wird durch einen Winkel in Altgrad (ein rechter Winkel entsprechend 90 Grad) eingestellt. Auf diese Weise läßt sich auch ein berechneter Stoß mit der Stoßkugel durchführen.

Der Umgang mit diesen Routinen erscheint nach dem ersten Durchlesen vielleicht etwas kompliziert. Sie werden aber sicher nach Lektüre des Demonstrationsprogrammes (Listing 1) und schlimmstenfalls nach einigem Probieren feststellen, wie einfach sich das beschriebene Maschinenprogramm anwenden läßt. (Bernhard Tertelmann/ev)

Die wichtigsten Programmdateien:

Adresse des Maschinen-Programms: 2048 bis 4972
Multiplikationstabelle und verschiedene Arbeitsspeicher: 49152 bis 52245

Adressen der Routinen: SE = 2158, IN = 3211, AL = 2372,
VL = 2327, KS = 2444, SW = 3255, JS = 3294, N1 = 3566,
GS = 3588, V2 = 3232, ZS = 4828

Benutzte Zeropage-Adresse: \$FB bis \$FE und \$57 bis \$64

Außerdem werden einige Betriebssystem-Routinen aufgerufen wie beispielsweise »FAC = KONST * FAC«

Adresse des Basic-Programms: ab 4973

```

1 REM ***** <138>
2 REM * BILLARD ('85) * <210>
3 REM * ----- * <103>
4 REM * B. TERTELMANN * <132>
5 REM * OSTENDORF 36 * <186>
6 REM * 4435 HORSTMAR * <030>
7 REM ***** <144>
8 REM <070>
10 PRINT "{CLR}":V=53248:POKE V+32,2:POKE V
+33,9:T=54272:JJ=56320 <143>
15 SR=253:LR=254:L=52123:EB=21 <119>
20 SE=2158:IN=3211:AL=2372:VL=2327:KS=2444 <179>
25 SW=3255:V2=3232:JS=3294:N1=3566:GS=3588
:ZS=4828 <191>
30 G(0)=229:G(1)=225:G(2)=106:G(3)=32 <236>
35 POKE V+39,0:POKE V+40,15:POKE V+43,10:P
OKE V+45,10:POKE T+24,15 <105>
40 X1$="{LIG.RED}0123456789{RED}";X2$="{
RVSON,RED}{LIG.RED,10SPACE,RED}" <162>
45 Y$="{RVSON}{RVOFF}{RVSON}{15SPACE}{
RVOFF}{RVSON}{15SPACE}{RVOFF}{RVSON
N}" <055>
50 Z$="{38SPACE}" <179>
55 POKE V+21,0:PRINT "{CLR,RED,RVSON,2SPACE
}"Z$"{2SPACE}"Z$"{2SPACE}"Z$"{BLACK,2SP
ACE}"Z$; <195>
60 FOR I=0 TO 19:PRINT "SPC(38) ";NEXT <065>
65 PRINT "Z$ {HOME,4DOWN}":POKE 56295,0:P
OKE 2023,160 <188>
70 PRINT SPC(14)"RS.SS(4SPACE)S"; <245>
75 PRINT SPC(30)"RWS--RSRWR"; <163>
80 PRINT SPC(30)"EXZZZZEZZE"; <016>
85 PRINT SPC(30)"YYYYYYYYYY" <044>
90 PRINT TAB(3){DOWN}TASTE'1':BANDENBIL
L.(1SPIELER)" <029>
95 PRINT TAB(3){DOWN}TASTE'2':BANDENBIL
L.(2SPIELER)" <130>
100 PRINT TAB(3){DOWN}TASTE'3':LOCHBILL
.{LIG.RED}00{BLACK,SPACE}(1SPIELER) <113>
105 PRINT TAB(3){DOWN}TASTE'4':LOCHBILL
.{LIG.RED}0{GREEN}0{BLACK,SPACE}(2SPI
ELER)" <104>
110 PRINT TAB(3){DOWN}TASTE'5':LOCHBILL
.{LIG.RED}0{GREEN}0{BLACK,SPACE}(1SPI
ELER)" <156>
115 PRINT TAB(3){DOWN}TASTE'6':LOCHBILL
.{LIG.RED}0{GREEN}0{BLACK,SPACE}(2SPI
ELER)" <226>
120 PRINT {2DOWN,2RIGHT,RVSON}SCHLAGWIEDER
H.'W'{3SPACE}SPIELWAHL'HOME'(HOME)"; <121>
125 GET A$:IF A$<"1"OR A$>"6"THEN 125 <242>
130 PRINT {5DOWN}:FOR I=0 TO 15:PRINT {RI
GHT}"Z$ {RIGHT}";NEXT <217>
135 ON VAL(A$)GOTO 365,310,285,235,200 <150>
139 REM * SPIEL 6 * <173>
140 GOSUB 425:GOSUB 445 <240>
145 PRINT {RVSON,RED,4SPACE}"SPC(11)"YYYYY
YYYYY"SPC(11){4SPACE,UP}";E(0)=255:E
(1)=255:GOSUB 470 <143>
150 GOSUB 485 <150>
155 GOSUB 560:GOSUB 620:R=PEEK(LR):IF R<3
THEN 150 <235>
160 F=1:FOR J=2 TO 6:IF PEEK(L+J)AND 168 T
HEN SYS VL,J:F=0 <167>
165 J=J+1:IF PEEK(L+J)AND 84 THEN SYS VL,J
:F=0 <076>
170 NEXT:E=PEEK(V+21):IF E(I)=255 THEN IF
E<252 THEN J=-((E AND 84)=84):GOSUB 50
0 <019>
175 IF(E AND E(I))=0 THEN 195 <180>
180 IF(E AND E(1-I))=0 THEN I=1-I:GOTO 195 <213>
185 IF NOT E AND 2 OR F=0 OR (R AND E(1-I))
THEN 150 <241>
190 GOTO 155 <102>
195 GOSUB 525:ON F GOTO 145,55 <062>
199 REM * SPIEL 5 * <231>
200 GOSUB 425:Z=0:B=999:PRINT {LIG.RED}SPI
EL{RED}"SPC(9)X1$SPC(28)X2$ <035>
205 GOSUB 460 <013>
210 GOSUB 560:S=S+1:PRINT SPC(5){RED,RVSO
N,3SPACE,RVOFF,BLACK,3LEFT}"S {UP}":GO
SUB 620:IF PEEK(LR)<3 THEN 210 <092>
215 FOR J=2 TO 6:IF PEEK(L+J)AND 168 THEN
SYS VL,J <050>
220 J=J+1:IF PEEK(L+J)AND 84 THEN SYS VL,J <114>
225 NEXT:IF PEEK(V+21)>3 THEN 210 <048>
230 GOSUB 510:ON F GOTO 205,55 <149>

```

```

234 REM * SPIEL 4 * <008>
235 GOSUB 415:POKE V+42,5:POKE V+44,5:POKE
V+46,5:GOSUB 445 <214>
240 PRINT {RVSON,RED,4SPACE}"SPC(11)"YYYYY
YYYYY"SPC(11){4SPACE,UP}";E(0)=0:E(1
)=0:GOSUB 470 <036>
245 GOSUB 485 <245>
250 GOSUB 560:GOSUB 620:R=PEEK(LR):IF R<3
THEN 245 <112>
255 IF E(I)=0 THEN GOSUB 495 <052>
260 E=PEEK(V+21):IF(E AND E(I))=0 THEN 280 <229>
265 IF(E AND E(1-I))=0 THEN I=1-I:GOTO 280 <162>
270 IF NOT E AND 2 OR (R AND E(1-I)) THEN 24
5 <045>
275 GOTO 250 <037>
280 GOSUB 525:ON F GOTO 240,55 <073>
284 REM * SPIEL 3 * <058>
285 GOSUB 415:POKE V+42,10:POKE V+44,10:PO
KE V+46,10 <210>
290 Z=0:B=999:PRINT {LIG.RED}SPIEL{RED}"SP
C(9)X1$SPC(28)X2$ <223>
295 GOSUB 460 <105>
300 GOSUB 560:S=S+1:PRINT SPC(5){RED,RVSO
N,3SPACE,RVOFF,BLACK,3LEFT}"S {UP}":GO
SUB 620 <120>
302 IF PEEK(LR)=0 OR PEEK(V+21)>3 THEN 300 <134>
305 GOSUB 510:ON F GOTO 295,55 <042>
309 REM * SPIEL 2 * <081>
310 POKE V+41,12:POKE V+42,10:PRINT {RI
GHT,2DOWN,RVSON}"Z$ {HOME}";GOSUB 450 <135>
315 PRINT {LIG.RED,RVSON}SP.{3SPACE,RED,9S
PACE}YYYYYYYYYYY{9SPACE,LIG.RED}SP.{3SP
ACE,UP}";L$="{RVSON}"Z$ <255>
320 PRINT TAB(11){RED,RVSON,2SPACE}"TAB(2
7)" {2SPACE,UP}" <119>
325 SYS IN,3:G=0:S(0)=0:S(1)=0:POKE V+21,0
:POKE V+16,0 <176>
330 SYS AL,1,23,219:SYS AL,2,36,215:SYS AL
,3,26,206 <246>
335 SYS ZS,14,80+RND(TI)*15,23+RND(1)*9 <008>
340 GOSUB 485 <086>
345 SK=I+1:GOSUB 565:GOSUB 620:IF PEEK(SR)
<9 THEN 340 <004>
350 S(I)=S(I)+1:PRINT {RVOFF,BLACK}"TAB(11
)MID$(STR$(S(0)),2)TAB(27)MID$(STR$(S(
1)),2)" {UP}" <045>
355 POKE T+18,16:POKE T+18,17:IF S(I)<EB T
HEN 345 <229>
360 Z(I)=Z(I)+1:PRINT {LIG.RED,RVSON}"TAB(
3)Z(0)TAB(37)Z(1):GOSUB 530:ON F GOTO
320,55 <014>
364 REM * SPIEL 1 * <134>
365 POKE V+41,12:POKE V+42,10:PRINT {RI
GHT,2DOWN,RVSON}"Z$ {HOME}"; <026>
370 Z=0:B=999:PRINT {LIG.RED}SPIEL{RED}"SP
C(9)X1$SPC(28)X2$:L$="{RVSON}"Z$ <025>
375 PRINT {BLACK}PKT.{RED,RVSON,3SPACE}"SP
C(8)"YYYYYYYYYYY{6SPACE,BLACK,RVOFF}BES
T.(HOME)"; <255>
380 P=0:S=0:Z=Z+1:PRINT {RVSON,LIG.RED}"SP
C(5)Z {DOWN}" <115>
385 SYS IN,3:SK=1:G=0:POKE V+21,0:POKE V+1
6,0 <186>
390 SYS AL,1,171,190:SYS AL,2,260,147:SYS
AL,3,88,147 <109>
395 GOSUB 565:S=S+1:PRINT TAB(27){BLACK}"
MID$(STR$(S),2)" {HOME,DOWN}":GOSUB 620 <020>
400 IF PEEK(SR)<12 THEN 395 <171>
405 P=P+1:POKE T+18,16:PRINT SPC(4)P;:POKE
T+18,17:IF P<EB THEN 395 <145>
410 GOSUB 510:ON F GOTO 375,55 <135>
414 REM FELD FUER SP. 3 UND 4 <201>
415 PRINT {RVSON}SPC(38)"F{RVOFF}SPC(3
8)" {RVSON}Y$ {HOME}";:POKE 2023,254 <158>
420 PRINT {3DOWN}SPC(15)"SPC(15)"
SPC(15)"F{RVOFF}SPC(38)";GOTO 440 <001>
424 REM FELD FUER SP. 5 UND 6 <092>
425 POKE V+42,5:POKE V+44,5:POKE V+46,5 <090>
430 PRINT {RVSON}SPC(38)"F{RVOFF}{GREEN
}SPC(17)" {LIG.RED}00"SPC(17)" {GREEN}
00{BLACK,RVSON}Y$ {HOME}";:POKE 2023,
254 <175>

```

Listing 1. Das Basic-Listing zu »Billard«. Bitte nur mit dem Checksummer 64V3 in diesem Heft eingeben.

```

435 PRINT " (3DOWN)TTF"SPC(15) "TTF"SPC(15) "
TTF(RVOFF)R(LIG.RED)H"SPC(17) " (GREEN)J
H"SPC(17) " (LIG.RED)H(BLACK)"; <053>
440 POKE V+41,10:PRINT " (RVSON)R"SPC(38) "V
(HOME)";:L$=Y$:RETURN <014>
444 REM VORBEREITUNG FUER SP. 2,4 UND 6 <157>
445 K$(0)="ROT":K$(1)="GRN":K(0)=84:K(1)=1
68 <222>
450 P(0)=1033:P(1)=1054:Z(0)=0:Z(1)=0:I=1 <050>
455 PRINT " (RVSON,GREY 3)SPIELER 1 (SPACE,RE
D,4SPACE)"X1$" (4SPACE,GREY 3,SPACE)SPI
ELER 2"SPC(14)X2$:RETURN <103>
459 REM VORBEREITUNG FUER SP. 3 UND 5 <142>
460 PRINT " (BLACK)STOSS (RVSON,RED,3SPACE)"S
PC(7) "YYYYYYYYYYY"SPC(5) " (BLACK,RVOFF)B
EST.: (HOME)"; <099>
465 S=0:Z=Z+1:PRINT " (LIG.RED,RVSON)"SPC(5)
Z" {DOWN}" <135>
469 REM KUGEL-GRUNDPOS. FUER LOCHBILL. <192>
470 SYS IN,135:SK=1:G=0:POKE V+21,0:POKE V
+16,0 <045>
475 SYS AL,1,260,147:SYS AL,2,111,147:SYS
AL,3,100,140:SYS AL,4,89,133 <109>
480 SYS AL,5,100,154:SYS AL,6,89,161:SYS A
L,7,89,147:RETURN <052>
484 REM SPIELERWECHSEL <210>
485 POKE T+15,222:POKE T+19,3:POKE T+18,32
:POKE P(I),160 <232>
490 I=1-I:POKE P(I),209:POKE T+18,33:RETUR
N <015>
494 REM ZUORDNUNG KUGELFARBE ZU SPIELER <237>
495 J=1:IF R AND 84 THEN J=0 <070>
500 F$(I)=K$(J):F$(1-I)=K$(1-J):E(I)=K(J):
E(1-I)=K(1-J) <098>
505 PRINT " (BLACK,RVOFF)"F$(0)SPC(34)F$(1) "
{UP}";:RETURN <081>
509 REM SPIELLENDE FUER SP. 1, 3 UND 5 <166>
510 POKE T+19,5:FOR J=9 TO 30:POKE T+15,J↑
1↑1:POKE T+18,32:POKE T+18,33:NEXT <073>
515 IF S<B THEN B=S <180>
520 PRINT TAB(36) " (RED,RVSON,3SPACE,RVOFF,
BLACK,3LEFT)"B:GOTO 540 <195>
524 REM SPIELLENDE FUER SP. 2, 4 UND 6 <225>
525 Z(I)=Z(I)+1:PRINT SPC(12) " (BLACK,RVOFF
)"MID$(STR$(Z(0)),2)TAB(27)MID$(STR$(Z

```

```

(1)),2) <020>
530 POKE P(1-I),160:POKE T+19,5:FOR J=9 TO
30:F=81-(F=81)*128:POKE P(I),F <040>
535 POKE T+15,J↑1↑1:POKE T+18,32:POKE T+18
,33:NEXT <119>
540 POKE 214,25:PRINT " (UP,3RIGHT,RVSON,BLA
CK)SPIELWAHL: 'HOME' (5SPACE)WEITER: 'FIR
E' {UP}" <093>
545 IF NOT PEEK(JJ)AND 16 THEN PRINT L$" {H
OME,DOWN}":F=1:RETURN <242>
550 GET A$:IF A$<>" (HOME)"THEN 545 <189>
555 F=2:RETURN <234>
559 REM STOSSKUGEL AUF GRUNDPOSITION <070>
560 IF(PEEK(V+21)AND 2)=0 THEN SYS AL,1,26
0,147 <206>
565 SYS KS,SK <064>
570 IF(PEEK(JJ)AND 31)<31 THEN 595 <124>
575 GET A$:IF A$=" (HOME)"THEN 10 <001>
580 IF A$<>"W"OR G=0 THEN 570 <151>
584 REM SCHLAGWIEDERHOLUNG <030>
585 POKE V+32,8:SYS SW:GOSUB 600:SYS GS,G <001>
590 FOR J=0 TO 2000:NEXT:GOSUB 625:SYS V2:
POKE V+32,2:GOTO 570 <108>
594 REM JOYSTICK UND GESCHW.-EINGABE <255>
595 SYS JS <088>
600 G=1:FOR F=1079 TO 1088:FOR J=0 TO 3:IF
PEEK(JJ)AND 16 THEN RETURN <225>
605 POKE F,G(J):G=G+2↑1:NEXT:NEXT:FOR J=0
TO 100:IF PEEK(JJ)AND 16 THEN RETURN <001>
610 NEXT:GOSUB 625:GOTO 600 <174>
620 SYS N1,G:REM STOSS UND KUGELBEWEGUNG <233>
624 REM GESCHW.-SKALA LOESCHEN <124>
625 FOR J=1088 TO 1079 STEP-1:POKE J,160:N
EXT:RETURN <098>
630 REM <184>
631 REM * DIE VARIABLE 'EB' IN ZEILE 15
* LEGT DAS SPIEL-ENDE <148>
632 REM * FUER BANDENBILLARD FEST. <052>
635 REM <189>
636 REM-ZEILEN BRAUCHEN NICHT MIT IN'S
PROGRAMM UEBERNOMMEN ZU WERDEN ! <198>

```

0 64'er

Listing 1. Das Basic-Listing zu »Billard« (Schluß)

PROGRAMM : BIL	C000 CB6F	C118 : F1 B7 BD 69 0C 2C 15 D0 99	C248 : 00 00 00 00 00 00 00 00 49
C000 : 00 0B 08 01 00 9E 32 30 C6	C120 : D0 F4 48 0D 15 D0 BD 15 56	C250 : 00 00 00 00 00 00 00 00 51	C258 : 00 00 00 00 00 00 F0 00 03 E6
C008 : 36 31 00 00 00 A0 6D 84 9B	C128 : D0 8A A8 0A AA B9 9B CB 27	C260 : 9C 00 06 66 00 07 FE 00 7F	C268 : 0F FF 00 0F FF 00 0F FF 95
C010 : 2B 88 84 7A A9 13 85 2C 91	C130 : 29 1C D0 03 A9 D6 2C A9 51	C270 : 00 0F FF 00 07 FE 00 07 6E	C278 : FE 00 03 FC 00 00 F0 00 9B
C018 : 85 7B A0 00 9B 91 7A A9 D6	C138 : 50 9D 01 D0 68 2D 10 D0 83	C280 : 00 00 00 00 00 00 00 00 81	C288 : 00 00 00 00 00 00 00 00 89
C020 : C0 85 FC A0 00 84 FB 98 3B	C140 : D0 45 F0 39 20 F1 B7 8A 9C	C290 : 00 00 00 00 00 00 00 00 91	C298 : 00 00 00 00 00 F0 00 03 26
C028 : 91 FB C8 10 FA B4 FB A9 02	C148 : 4B 20 FD AE 20 EB 87 68 07	C300 : FC 00 07 CE 00 07 F6 00 4C	C308 : 0F FB 00 0F FB 00 0F FF 93
C030 : 00 85 FD F0 03 18 69 05 31	C150 : A8 0A AA B9 69 0C 48 0D 11	C310 : 00 00 00 00 00 00 00 00 01	C318 : 00 0F FF 00 07 FE 00 07 AE
C038 : 85 FE A0 0C 20 A2 BB 20 2C	C158 : 15 D0 8D 15 D0 A5 65 9D E6	C320 : 00 00 00 00 00 00 00 00 09	C328 : 00 00 00 00 00 00 00 00 0B
C040 : CA BB A4 FD 20 A2 B3 20 F7	C160 : 01 D0 A5 14 9D 00 D0 A5 1E	C330 : 00 00 00 00 00 00 00 00 0C	C338 : 00 00 00 00 00 00 00 00 0D
C048 : 52 E0 20 9B BC A5 65 A0 56	C168 : 15 F0 09 68 0D 10 D0 8D F4	C340 : 00 00 00 00 00 00 00 00 0E	C348 : 00 00 00 00 00 00 00 00 0F
C050 : 00 91 FB E6 FB D0 02 E6 11	C170 : 10 D0 D0 13 68 49 FF 2D AA	C350 : 00 00 00 00 00 00 00 00 10	C358 : 00 00 00 00 00 00 00 00 11
C058 : FC A5 FB 29 7F 85 FD D0 09	C178 : 10 D0 8D 10 D0 8D 00 D0 F2	C360 : 00 00 00 00 00 00 00 00 12	C368 : 00 00 00 00 00 00 00 00 13
C060 : E1 A5 FE C9 64 D0 CE A9 68	C180 : C9 A6 B0 03 A9 34 2C A9 69	C370 : 00 00 00 00 00 00 00 00 14	C378 : 00 00 00 00 00 00 00 00 15
C068 : 00 91 FB C8 10 FB AD 84 E9	C188 : 4A 4C B0 10 20 F1 B7 86 A4	C380 : 00 00 00 00 00 00 00 00 16	C388 : 00 00 00 00 00 00 00 00 17
C070 : 0C AA 8D E1 0E 38 ED 88 DC	C190 : FC AC 10 D0 98 3D 69 0C 32	C390 : 00 00 00 00 00 00 00 00 18	C398 : 00 00 00 00 00 00 00 00 19
C078 : 0C 8D 12 0D 8A 38 6D 89 A4	C198 : F0 05 98 09 01 D0 03 98 26	C400 : 00 00 00 00 00 00 00 00 1A	C408 : 00 00 00 00 00 00 00 00 1B
C080 : 0C 8D CF 0F 8D 0B 10 8A AF	C200 : 29 FE 8D 10 D0 8A AB 0A C6	C410 : 00 00 00 00 00 00 00 00 1C	C418 : 00 00 00 00 00 00 00 00 1D
C088 : 18 6D 85 0C 38 6A 8D 81 4A	C208 : AA BD 00 D0 8D 00 D0 BD E3	C420 : 00 00 00 00 00 00 00 00 1E	C428 : 00 00 00 00 00 00 00 00 1F
C090 : 09 AA 38 6D 8A 0C 8D 07 99	C210 : 01 D0 8D 01 D0 AD 15 D0 0D	C430 : 00 00 00 00 00 00 00 00 20	C438 : 00 00 00 00 00 00 00 00 21
C098 : 0F 8D 13 10 8A 38 ED 8A 6C	C218 : 09 01 8D 15 D0 60 00 00 58	C440 : 00 00 00 00 00 00 00 00 22	C448 : 00 00 00 00 00 00 00 00 23
C0A0 : 0C 8D D3 0F 8D 0F 10 AD 37	C220 : C0 00 00 00 00 00 00 00 C1	C450 : 00 00 00 00 00 00 00 00 24	C458 : 00 00 00 00 00 00 00 00 25
C0A8 : 85 0C AA 8D 21 0F 18 6D 55	C228 : 00 00 00 00 00 00 00 00 09	C460 : 00 00 00 00 00 00 00 00 26	C468 : 00 00 00 00 00 00 00 00 27
C0B0 : 88 0C 8D 2C 0D 8A 38 ED 09	C230 : 00 90 00 00 60 00 00 60 F7	C470 : 00 00 00 00 00 00 00 00 28	C478 : 00 00 00 00 00 00 00 00 29
C0B8 : 89 0C 8D 91 0F 8D 02 10 62	C238 : 00 00 90 00 01 08 00 00 65	C480 : 00 00 00 00 00 00 00 00 2A	C488 : 00 00 00 00 00 00 00 00 2B
C0C0 : AD 86 0C AA 8D 7E 0F 38 82	C240 : 00 00 00 00 00 00 00 00 F9	C490 : 00 00 00 00 00 00 00 00 2C	C498 : 00 00 00 00 00 00 00 00 2D
C0C8 : ED 88 0C 8D EE 0C 8A 38 98	C248 : 00 00 00 00 00 00 00 00 01	C500 : 00 00 00 00 00 00 00 00 2E	C508 : 00 00 00 00 00 00 00 00 2F
C0D0 : 6D 89 0C 8D 38 09 8D F2 9F	C250 : 00 00 00 00 00 00 00 00 09	C510 : 00 00 00 00 00 00 00 00 30	C518 : 00 00 00 00 00 00 00 00 31
C0D8 : 0E 8D 2F 0F AD 87 0C AA F7	C258 : 00 00 00 00 00 00 00 00 03 A6	C520 : 00 00 00 00 00 00 00 00 32	C528 : 00 00 00 00 00 00 00 00 33
C0E0 : 8D EF 0F 18 6D 88 0C 8D 92	C260 : FC 00 07 3E 00 06 FE 00 D2	C530 : 00 00 00 00 00 00 00 00 34	C538 : 00 00 00 00 00 00 00 00 35
C0E8 : FD 0C 8A 38 ED 89 0C AB 42	C268 : 0D FF 00 0D FF 00 0F FF 13	C540 : 00 00 00 00 00 00 00 00 36	C548 : 00 00 00 00 00 00 00 00 37
C0F0 : 88 8C 35 09 8D EB 0E 8D 88	C270 : 00 0F FF 00 07 FE 00 07 2E	C550 : 00 00 00 00 00 00 00 00 38	C558 : 00 00 00 00 00 00 00 00 39
C0F8 : 2B 0F A9 27 8D F8 07 A0 F8	C278 : FE 00 03 FC 00 00 F0 00 5B	C560 : 00 00 00 00 00 00 00 00 3A	C568 : 00 00 00 00 00 00 00 00 3B
C100 : 07 A9 05 99 DA CB A9 00 03	C280 : 00 00 00 00 00 00 00 00 41	C570 : 00 00 00 00 00 00 00 00 3C	C578 : 00 00 00 00 00 00 00 00 3D
C108 : 99 B0 CB 99 B7 CB A9 28 F0		C580 : 00 00 00 00 00 00 00 00 3E	C588 : 00 00 00 00 00 00 00 00 3F
C110 : 99 F8 07 88 D0 EB 60 20 26		C590 : 00 00 00 00 00 00 00 00 40	C598 : 00 00 00 00 00 00 00 00 41

Listing 2. Der Maschinenspracheteil von »Billard«, einzugeben mit dem MSE (Seite 8). Beachten Sie bei der Eingabe die Programmbeschreibung.

C328 : 0F FF 00 0F FF 00 0F FB 4D
 C330 : 00 0F FB 00 07 F6 00 07 ED
 C338 : CE 00 03 FC 00 00 F0 00 2B
 C340 : 00 00 00 00 00 00 00 41
 C348 : 00 00 00 00 00 00 00 49
 C350 : 00 00 00 00 00 00 00 51
 C358 : 00 00 00 00 00 F0 00 03 E6
 C360 : FC 00 07 FE 00 07 FE 00 32
 C368 : 0F FF 00 0F FF 00 0F FF 95
 C370 : 00 0F FF 00 07 FE 00 06 6C
 C378 : 66 00 03 9C 00 00 F0 00 F7
 C380 : 00 00 00 00 00 00 00 81
 C388 : 00 00 00 00 00 00 00 89
 C390 : 00 00 00 00 00 00 00 91
 C398 : 00 00 00 00 00 F0 00 03 26
 C3A0 : FC 00 07 FE 00 07 FE 00 72
 C3A8 : 0F FF 00 0F FF 00 0F FF CD
 C3B0 : 00 0D FF 00 06 FE 00 07 9D
 C3B8 : 3E 00 03 FC 00 00 F0 00 1B
 C3C0 : 00 00 00 00 00 00 00 C1
 C3C8 : 00 00 00 00 00 00 00 C9
 C3D0 : 00 00 00 00 00 00 00 D1
 C3D8 : 00 00 00 00 00 F0 00 03 66
 C3E0 : FC 00 07 FE 00 05 FE 00 A2
 C3E8 : 0D FF 00 0B FF 00 0B FF 82
 C3F0 : 00 0D FF 00 05 FE 00 07 CD
 C3F8 : FE 00 03 FC 00 00 F0 00 1B
 C400 : 80 7E F2 DB 70 80 7B DE AE
 C408 : E5 FF 80 76 FA C3 04 80 BB
 C410 : 70 96 4D 42 80 69 0D C7 7E
 C418 : 08 80 64 F9 2E 2D 80 5B BE
 C420 : 84 A8 08 80 55 01 3F 82 6A
 C428 : 80 4A 12 FC A6 80 40 AB 87
 C430 : DD C3 80 35 04 F3 34 80 62
 C438 : 28 93 C2 0B 80 1D 2B 36 46
 C440 : 49 80 0E 00 05 03 80 03 CA
 C448 : B5 FE 6B 7F 64 F9 2E 2E 72
 C450 : 7F 53 DD FA C3 7F 2E F8 32
 C458 : F2 60 7F 06 B7 53 19 7E 9E
 C460 : 37 2D BE 8B 7D 39 6A CE 38
 C468 : 23 01 02 04 08 10 20 40 0F
 C470 : 80 0C 0C 0C 0C 0C 0B E2
 C478 : 0A 09 08 07 06 50 23 0F 77
 C480 : 07 02 01 01 14 38 49 DD CD
 C488 : 03 06 0A 20 F1 B7 86 FB 04
 C490 : 8A 29 07 8D 65 0E 8D 99 53
 C498 : 0E 8D CC 10 A8 4C 01 09 A5
 C4A0 : AD 9B CB 8D 15 D0 A0 11 3C
 C4A8 : B9 89 CB 99 FF CF 88 D0 8E
 C4B0 : F7 AD 89 CB 85 FC 60 AD 77
 C4B8 : 15 D0 8D 9B CB AD 15 CC 24
 C4C0 : 8D 15 D0 A0 11 B9 FF CF 9E
 C4C8 : 99 89 CB 89 03 CC 99 FF 4D
 C4D0 : CF 88 D0 F1 A5 FC 8D 89 E1
 C4D8 : CB AD 03 CC 85 FC A9 14 E3
 C4E0 : 85 64 AC 00 CD 98 29 01 FC
 C4E8 : D0 0A AD 01 D0 C9 46 F0 9F
 C4F0 : 03 CE 01 D0 98 29 02 D0 31
 C4F8 : 0A AD 01 D0 C9 E0 F0 03 A1
 C500 : EE 01 D0 98 29 04 D0 15 D6
 C508 : AD 10 D0 4A AD 00 D0 80 BA
 C510 : 04 C9 11 F0 08 D0 03 CE 0C
 C518 : 10 D0 CE 00 D0 98 29 08 CA
 C520 : D0 15 AD 10 D0 4A 90 07 9C
 C528 : AD 00 D0 C9 3B F0 08 EE 78
 C530 : 00 D0 D0 03 EE 10 D0 98 11
 C538 : 29 10 D0 06 A7 28 29 0F 49 2C
 C540 : 0F D0 06 A9 14 85 64 D0 0F
 C548 : 0A A5 64 D0 04 A2 04 D0 5F
 C550 : 04 C6 64 A2 14 A0 FF 88 7C
 C558 : D0 FD CA D0 F8 20 E1 FF 0C
 C560 : D0 80 60 A9 01 2D 10 D0 19
 C568 : AC A0 D0 20 91 B3 20 0C 9C
 C570 : BC A4 FC 98 0A AA B9 69 80
 C578 : 0C 2D 10 D0 F0 02 A9 01 01
 C580 : BC 00 D0 20 91 B3 20 53 52
 C588 : B8 A6 FC A9 01 24 66 10 F3
 C590 : 02 A9 00 9D BE CB 20 CA 7B
 C598 : BB AC 01 D0 20 A2 B3 20 2A
 C5A0 : 0C BC AC FC 0A AA BC 01 FE
 C5A8 : D0 20 A2 B3 20 53 B8 A6 74
 C5B0 : FC A9 01 24 66 10 02 A9 88
 C5B8 : 00 9D C5 CB A5 61 D0 07 2B
 C5C0 : A9 49 A0 E3 20 A2 B8 A0 FA
 C5C8 : 00 A9 57 20 0F B8 46 66 2B
 C5D0 : A2 84 A0 CB 20 D4 BB 20 2E
 C5D8 : CA B8 20 52 E0 A9 BC A0 62
 C5E0 : B9 20 67 B8 20 71 BF A2 6C
 C5E8 : FE A0 CB 4C D4 BB AD 15 BF
 C5F0 : D0 8D 15 CC A0 11 B9 FF DF
 C5F8 : CF 99 03 CC 88 D0 F7 A5 28

C600 : FC 8D 03 CC 20 FD AE 20 0A
 C608 : 8A AD 20 0C BC A9 FE A0 49
 C610 : CB 20 05 BB 20 CA BB 20 2C
 C618 : 9B BC A5 65 A6 FC B0 B7 60
 C620 : CB A9 84 A0 CB 20 A2 BB B5
 C628 : 20 52 E0 20 9B BC A5 65 AE
 C630 : A6 FC 9D B0 CB A9 00 8D F7
 C638 : 06 D4 8D 0C D4 8D 0D D4 25
 C640 : 8D 14 D4 A9 10 8D 05 D4 6D
 C648 : A9 07 8D 13 D4 A9 04 8D 00
 C650 : 01 D4 A9 32 8D 0B D4 A9 2C
 C658 : 1E 8D 0F D4 AD 17 D4 29 D4
 C660 : F8 8D 17 D4 A0 07 A9 0A 7C
 C668 : 8D 54 0F 8D 2E 10 99 CC F8
 C670 : CB 99 D3 CB A9 00 99 A2 BC
 C678 : CB 99 A9 CB 99 9B CB 88 AA
 C680 : D0 E4 85 FD 85 FE AD 15 14
 C688 : D0 29 FE 8D 15 D0 85 63 13
 C690 : A2 10 8E 0B D4 EE 0B D4 DA
 C698 : A0 07 B9 69 0C 25 63 D0 70
 C6A0 : 03 4C 9C 10 9B 0A AA BD F2
 C6A8 : 00 D0 85 57 AD 00 85 06
 C6B0 : 59 8D 01 D0 85 58 A9 00 04
 C6B8 : 85 62 B9 A2 CB 18 79 B0 F6
 C6C0 : CB 30 06 9A A2 CB 4C 5E CF
 C6C8 : 0F 29 7F 99 A2 CB A9 01 B0
 C6D0 : 85 62 B9 BE CB D0 40 B9 84
 C6D8 : 69 0C 24 59 D0 27 A5 57 07
 C6E0 : C9 14 D0 2D 24 FB 10 10 10
 C6E8 : A5 58 C9 D7 90 03 4C A0 BA
 C6F0 : 0F C9 50 B0 03 4C 98 0F 21
 C6F8 : 20 A5 10 A9 01 99 BE CB 94
 C700 : FE 00 D0 D0 47 A5 57 D0 ED
 C708 : 08 B9 69 0C 45 59 8D 10 3E
 C710 : D0 DE 00 D0 4C 4C 0F B9 40
 C718 : 69 0C 24 59 F0 21 A5 57 19
 C720 : C9 38 D0 1B 24 FB 10 0A 13
 C728 : A5 58 C9 D7 B0 79 C9 50 05
 C730 : 90 62 20 A5 10 A9 00 99 30
 C738 : BE CB DE 00 D0 0D FE 59
 C740 : 00 D0 D0 08 B9 69 0C 05 FF
 C748 : 59 8D 10 D0 98 AA DE CC 7A
 C750 : CB D0 08 A9 0A 9D CC CB 13
 C758 : DE B0 CB 98 0A AA B9 A9 C4
 C760 : CB 18 79 B7 CB 30 06 99 16
 C768 : A9 CB 4C 35 10 29 7F 99 2C
 C770 : A9 CB A9 01 85 62 B9 C5 67
 C778 : CB D0 71 A5 58 C9 49 D0 57
 C780 : 66 24 FB 10 55 B9 69 0C DA
 C788 : 2C 10 D0 F0 3E B0 D0 D0 82
 C790 : C9 32 90 46 A9 10 D0 11 E0
 C798 : A9 04 D0 0D A9 08 D0 09 49
 C7A0 : A9 20 D0 05 A9 40 2C A9 CF
 C7A8 : 80 99 9B CB A9 10 8D 12 CB
 C7B0 : D4 EE 12 D4 A9 00 9D 01 2E
 C7B8 : D0 B9 69 0C AA 4D 15 D0 4C
 C7C0 : 8D 15 D0 8A 05 FE 85 FE BA
 C7C8 : 4C 70 10 8D 00 D0 C9 18 EC
 C7D0 : 90 C6 C9 9C 90 04 C9 B1 7D
 C7D8 : 90 C2 20 A5 10 FE 01 D0 25
 C7E0 : A9 01 99 C5 CB D0 3F DE 27
 C7E8 : 01 D0 D0 3A A5 58 C9 DD C0
 C7F0 : D0 31 24 FB 10 20 B9 69 9D
 C7F8 : 0C 2C 10 D0 F0 09 B0 00 87
 C800 : D0 C9 32 80 A2 90 0F BD BE
 C808 : 00 D0 C9 1B 90 92 C9 9C 44
 C810 : 90 04 C9 B1 90 8E 20 A5 94
 C818 : 10 DE 01 D0 A9 00 99 C5 7E
 C820 : CB F0 03 FE 01 D0 98 AA 52
 C828 : DE D3 CB D0 08 A9 0A 9D 2E
 C830 : D3 CB DE B7 CB A5 62 F0 ED
 C838 : 25 98 AA DE DA CB D0 18 AF
 C840 : A9 05 9D DA CB BE F8 07 D3
 C848 : B9 BE CB F0 02 BE EB CA 12
 C850 : 8A 29 07 09 28 99 F8 07 93
 C858 : 98 0A AA 20 A2 10 BE 69 DD
 C860 : 0C B9 80 CB 19 B7 CB C9 03
 C868 : 02 90 05 BA 05 63 D0 0F 12
 C870 : A9 00 99 B0 CB 99 B7 CB 96
 C878 : B9 69 0C 49 FF 25 63 85 D4
 C880 : 63 D0 01 60 A2 FF 18 90 43
 C888 : 01 E8 0A D0 FA 90 01 E8 A4
 C890 : BD 70 0C AA CA D0 FD A2 D5
 C898 : 0A CA D0 FD 88 F0 03 4C B0
 C8A0 : 9A 0E 4C 98 0E A9 10 8D F1
 C8A8 : 04 D4 EE 04 D4 60 A9 63 10
 C8B0 : 84 64 8D 29 11 B9 69 0C 8B
 C8B8 : 2D 10 D0 F0 02 A9 01 85 BC
 C8C0 : 5C BD 00 D0 85 5A BD 01 39
 C8C8 : D0 85 5B A0 07 C4 64 F0 50
 C8D0 : 5A B9 69 0C 2D 15 D0 F0 83

C8D8 : 52 98 0A AA BD 00 D0 38 DE
 C8E0 : E5 5A 85 5E A9 00 2A 45 ED
 C8E8 : 5C AA B9 69 0C 2D 10 D0 41
 C8F0 : F0 04 8A 49 01 AA 8A 4A D2
 C8F8 : A5 5E B0 06 49 FF 69 01 F5
 C900 : F0 29 C9 0C B0 25 85 5E 80
 C908 : 86 60 98 0A AA A9 01 85 2D
 C910 : 61 8D 01 D0 38 E5 5B 80 2C
 C918 : 06 C6 61 49 FF 69 01 85 5D
 C920 : 5F A6 5E DD 71 0C B0 03 66
 C928 : 4C 63 11 88 F0 03 4C CD 6F
 C930 : 10 A4 64 60 A4 64 98 0A 9B
 C938 : AA FE 00 D0 D0 22 B9 69 53
 C940 : 0C 0D 10 D0 8D 10 D0 4C 26
 C948 : 85 10 A4 64 98 0A AA BD 8B
 C950 : 00 D0 D0 09 B9 69 0C 4D BF
 C958 : 10 D0 8D 10 D0 DE 00 D0 DB
 C960 : 4C 85 10 A2 10 8E 0B D4 2B
 C968 : EE 0B D4 49 0F 18 65 5E 3E
 C970 : 4A AA A9 00 6A 85 5A 85 C1
 C978 : 5C 8A 69 BE 85 5B A5 5E D2
 C980 : 49 0F 18 65 5F 4A 18 69 F7
 C988 : BE 85 5D A5 64 0A AA A5 A1
 C990 : 57 9D 00 D0 A5 59 8D 10 48
 C998 : D0 A5 58 9D 01 D0 98 AA 53
 C9A0 : 20 2B 12 8A A6 64 85 64 98
 C9A8 : 20 2B 12 A4 64 BD E1 CB CA
 C9B0 : 18 79 EB CB 38 10 05 49 E3
 C9B8 : FF 18 69 01 9D B0 CB A9 20
 C9C0 : 00 2A 9D BE CB 89 E1 CB BE
 C9C8 : 18 70 EB CB 38 10 05 49 FD
 C9D0 : FF 18 69 01 99 B0 CB A9 F7
 C9D8 : 00 2A 99 BE CB BD EF CB 2D
 C9E0 : 18 79 F6 CB 38 10 05 49 97
 C9E8 : FF 18 69 01 9D B7 CB A9 88
 C9F0 : 00 2A 9D C5 CB 89 EF CB 07
 C9F8 : 18 70 F6 CB 38 10 05 49 B1
 CA00 : FF 18 69 01 99 B7 CB A9 60
 CA08 : 00 2A 99 C5 CB E4 FC D0 B6
 CA10 : 05 B9 69 0C D0 07 C4 FC 20
 CA18 : D0 07 BD 69 0C 05 FD 85 F4
 CA20 : FD B9 69 0C 05 63 85 63 1E
 CA28 : 8A A8 60 BC B0 CB B1 5A 9B
 CA30 : 85 5E BD BE CB 45 61 F0 7A
 CA38 : 07 A9 00 38 E5 5E 85 5E 3F
 CA40 : BC B7 CB B1 5C 85 5F BD EC
 CA48 : C5 CB 45 60 4A A0 01 A5 49
 CA50 : 5F 80 04 49 FF 69 01 18 B1
 CA58 : 65 5E 10 06 49 FF 18 69 79
 CA60 : 01 88 84 5F 85 5E AB BD 1C
 CA68 : BE CB D0 07 38 FD B0 CB EF
 CA70 : 9D B0 CB BD C5 CB D0 07 1C
 CA78 : 38 FD B7 CB 9D B7 CB B1 40
 CA80 : 5A A8 A5 5F 45 61 F0 0C BF
 CA88 : 98 49 FF 18 69 01 9D E1 A1
 CA90 : CB 98 10 09 98 9D E1 CB 62
 CA98 : 49 FF 18 69 01 18 7D B0 3C
 CAA0 : CB 9D EB CB A4 5E B1 5C AA
 CAA8 : A8 A5 5F 45 60 D0 0C 98 91
 CAB0 : 49 FF 18 69 01 9D EF CB 80
 CAB8 : 98 10 09 98 9D EF CB 49 C9
 CAC0 : FF 18 69 01 18 7D B7 CB 2A
 CAC8 : 9D F6 CB A9 7F 9D A2 CB 0F
 CAD0 : 9D A9 CB A9 01 9D CC CB 32
 CAD8 : 9D D3 CB 60 20 FF 08 20 C0
 CAE0 : F1 B7 BE 80 CB 20 FD AE 74
 CAE8 : 20 BA AD A2 84 A0 CB 20 CA
 CAF0 : D4 88 20 FD AE 20 8A AD DB
 CAF8 : A9 67 A0 13 20 28 BA A2 53
 CB00 : FE A0 CB 20 D4 B8 20 64 BA
 CB08 : E2 A9 84 A0 CB 20 28 BA CB
 CB10 : A9 01 24 66 10 03 4A 46 DF
 CB18 : 66 8D 81 CB 20 9B BC A5 3C
 CB20 : 65 8D 82 CB A9 FE A0 CB 12
 CB28 : 20 A2 B8 20 6B E2 A9 84 0A
 CB30 : A0 CB 20 28 BA A9 00 06 CB
 CB38 : 66 85 66 2A 8D 83 CB 20 A4
 CB40 : 9B BC A2 07 18 0E 80 CB 4E
 CB48 : 90 17 AD B2 CB 9D B0 CB 23
 CB50 : A5 65 9D B7 CB AD 81 CB CE
 CB58 : 9D BE CB AD 83 CB 9D C5 95
 CB60 : CB CA D0 E0 4C 35 0E 7B 7E
 CB68 : 0E FA 35 12 00 FF 00 00 83

Listing 2. Der Maschinenspracheteil von »Billard« (Schluß).