

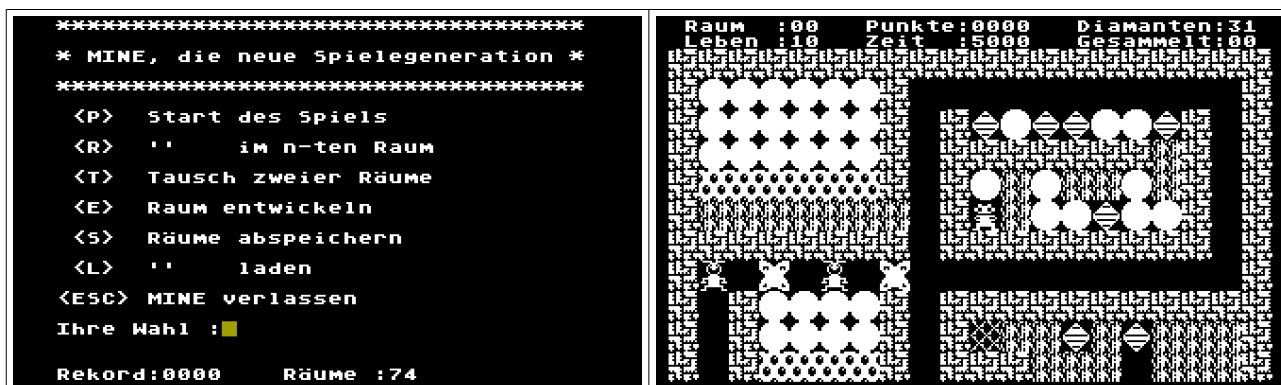
## "MINE" - ein Boulderdash<sup>1</sup>-Clone für den Ju+Te-6k

Das Spiel und die zugehörigen Raumdaten wurden durch "PC-Opa" von einer Kassette als Dateien gerettet. "Laden" und "Sichern" der Räume erfolgt normalerweise per Kassette aus dem Programm heraus. Für die Benutzung mit einem Emulator sind die Dateien entsprechend angepasst und benannt.

Programmierer: "STS" lt. Schlussbild (Thomas Singer?)

### Dateien (Bezeichnungen selbst gewählt):

- DEMO.BIN Demonstration des Editors und der Elemente (als "Spiel")
- MINE74.BIN das Spiel + 74 Räume<sup>2</sup>
- MINE.BIN das Spiel ohne Räume
- DAT06.BIN Daten für 6 Räume
- DAT74.BIN Daten für 74 Räume



### Bedienung am JTCEMU:

- Spiel laden nach %A000
- Raumdaten laden nach %BB52 (nicht bei DEMO und MINE74)
- Nach Start auf %A000 werden in der Fußzeile die beste Punktzahl (in der aktuellen Sitzung) und die aktuelle Anzahl der geladenen Räume angezeigt.
- Nach 'P' bzw. 'R' (und dann Raumnummer auswählen) gelangt man zur Ansicht. Mit den Kursortasten lässt sich ggf. der Bildschirm scrollen und so der nicht mehr auf den Schirm passende Teil betrachten.
- Die Leertaste oder <ENTER> startet das Spiel.
- <S> pausiert das Spiel, jede beliebige Taste (außer Kursortasten) setzt fort.
- Verliert man ein Leben, so wird der aktuelle Raum nach ca. 5 Sekunden (oder der Taste <ESC>) neu betreten, solange noch Leben verfügbar sind.
- <ESC> bricht ansonsten das Spiel im aktuellen Raum ab bzw. beendet das Menü.
- Ein Editor ermöglicht die Bearbeitung von Räumen. Dabei kann sowohl ein auszuwählender bestehender Raum geändert als auch ein neuer gestaltet werden (letzte Nummer in der Auswahl-Liste).

### Editor:

1

Die Nennung der Wortmarke "BOULDER DASH" (2017, BBG ENTERTAINMENT GMBH) dient nur zur Verdeutlichung des Spielprinzips.

2

Modifizierte Kompakt-Variante für EPROM oder JTCEMU:

Es gibt eine einzige Komplet-Datei, die das Spiel und auch die 74 Räume enthält. Funktionen zum Laden und Sichern von Räumen sind nicht vorhanden. Der Raum-Editor ist funktional; veränderte Räume bleiben nur bis zum Verlassen des Spiels/nächsten Neustart vorhanden.

Es gibt 16 verschiedene Elemente, die mit einer Buchstabentaste gesetzt werden können:

A	= Luft (bzw. vorhandenes Element löschen)	I	= horizontal bewegliches Alien
B	= Wand	J	= vertikal bewegliches Alien
C	= Sprungfeder	K	= hor. + vert. bewegliches Alien
D	= Stein	P	= Säure
E	= Diamant	M	= Spieler
F	= Sieb	L	= horizontaler Flatter
G	= Ausgang	N	= vertikaler Flatter
H	= Sand	O	= hor. + vert. Flatter

<+>, <->      Änderung der Diamanten-Sollzahl  
 <ENTER>        übernimmt den aktuellen Stand.  
 <ESC>            bricht das Bearbeiten ab (Original bleibt erhalten).

### **Demoversion:**

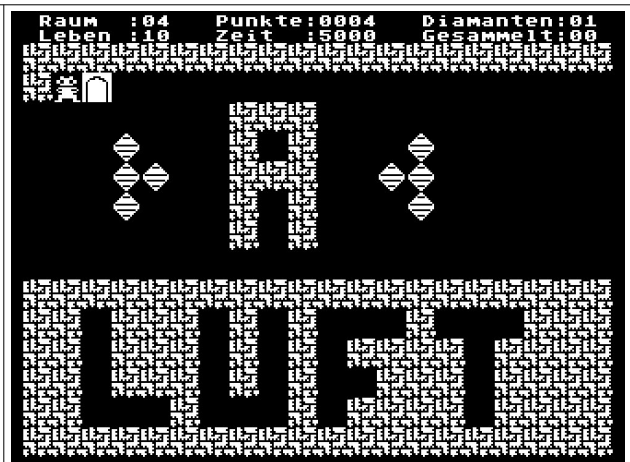
Eine der Dateien enthielt zur Demonstration des Elementverhaltens und Beschreibung der Editorkommandos 24 entsprechende "Räume".

Hier kann das Verhalten der Elemente studiert werden (und was bei einer Berührung passiert...).

Es ist jeweils nur 1 Diamant einzusammeln, dann geht es zum nächsten Raum.

Beispiel: Taste <A> "LUFT"

Ein Leersymbol wird gesetzt (z.B. zum Entfernen eines gesetzten anderen Elements).

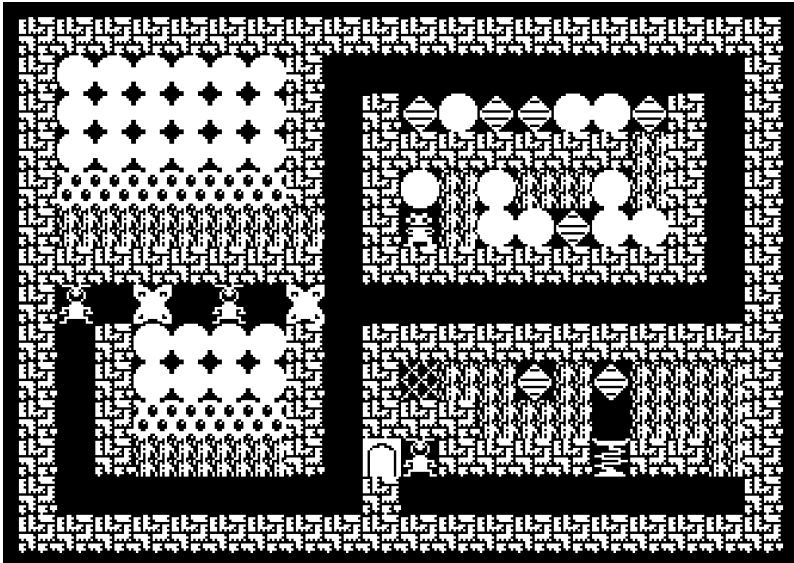


## Interna

### Raumdaten:

- liegen ab %BB52 im Speicher
- bei Programmstart wird ermittelt, wie viele Räume verfügbar sind
- 1 Raum = 142 Bytes lang
- Aufbau:
  - Kennung Raumbeginn: %FE
  - 140 Bytes für 20 \* 14 Bildschirm-Zeichen (1 Byte = 2 Zeichen):  
Inhalte 0...F entsprechend der o.a. Elementreihenfolge
  - letztes Byte = Anzahl zu sammelnde Diamanten für Freischaltung Ausgang
- am Dateiende muss nach letztem Raum ein %7F angehängt sein (Endeerkennung)

### Beispiel: Raum01

	<pre>11111111111111111111 13333331000000000001 13333331014344334101 13333331011111117101 15555551013737737101 1777777701C733433101 11111111011111111101 1A0E0A0E000000000001 10133331011111111111 1013333101F774747771 10155551011177707771 10177771069111121171 10000000010000000001 11111111111111111111</pre>
--	--

### Probleme:

1. Wird nach Programmstart und dann mit <R> sofort der letzte verfügbare Raum gewählt, so erscheint "Müll" auf dem Bildschirm. Erst wenn ein anderer Raum zumindest betrachtet wurde, ist auch der letzte ok. Wieso?
2. ...