

C64 BoulderDash I raw cave data and info

In this document:

- Cave size info
 - Object codes
 - Format of the raw cave data
 - Raw cave data
-

Cave size info

- Cave size
 - Logical size: 40 x 22
 - Playing area: 38 x 20
 - Visible area: 19.5 x 11.5 (I think)
- Intermission size
 - Logical size: 22 x 12
 - Playing area: 20 x 10
 - Visible area: the whole thing

The cave size is 40 (horizontally) x 22 (vertically) including the bounding steel wall; that is, a playing area of 38 x 20. This appears to be based on the C64's actual screen size: the C64 has 25 lines of 40 columns. My guess is that caves are only 22 lines high rather than 25 lines high because when playing the game, one line is given over to the status bar at the top of the screen, and one line is lost because of the way that the border of the screen is made larger in order to facilitate smooth scrolling by the C64's VIC chip. Anyway, when the game is in progress, the screen is magnified by 2 in each direction: one object occupies a 2 x 2 space on the C64's screen. This means that at any time, the player can see a (I think) 19.5 x 11 portion of the cave (19.5 objects horizontally takes 39 character spaces, one character space is lost to horizontal scrolling; 11 objects vertically takes 22 character spaces, plus 1 character space for the status bar, plus one lost to vertical scrolling). During an intermission, the border is shrunk back to its normal size, permitting viewing of the entire cave.

Object codes

BoulderDash uses a 6-bit code to represent which object is at each cave position as the game runs. Some of the available 64 codes aren't used in BoulderDash I, however later versions added some more objects. Here are the codes used in BoulderDash I.

\$00 Space
\$01 Dirt
\$02 Brick wall
\$03 Magic wall
\$04 Pre outbox (ie before it starts flashing)
\$05 Flashing outbox
\$06 *
\$07 Steel wall
\$08 Firefly position 1
\$09 Firefly position 2
\$0A Firefly position 3
\$0B Firefly position 4
\$0C Firefly position 1, scanned this frame
\$0D Firefly position 2, scanned this frame
\$0E Firefly position 3, scanned this frame
\$0F Firefly position 4, scanned this frame
\$10 Boulder, stationary
\$11 Boulder, stationary, scanned this frame
\$12 Boulder, falling
\$13 Boulder, falling, scanned this frame
\$14 Diamond, stationary
\$15 Diamond, stationary, scanned this frame
\$16 Diamond, falling
\$17 Diamond, falling, scanned this frame
\$18
\$19
\$1A
\$1B Explode to Space stage 0
\$1C Explode to Space stage 1
\$1D Explode to Space stage 2
\$1E Explode to Space stage 3
\$1F Explode to Space stage 4
\$20 Explode to Diamond stage 0
\$21 Explode to Diamond stage 1
\$22 Explode to Diamond stage 2
\$23 Explode to Diamond stage 3
\$24 Explode to Diamond stage 4
\$25 Pre Rockford stage 1 (ie when inbox morphs into Rockford)
\$26 Pre Rockford stage 2
\$27 Pre Rockford stage 3
\$28 Pre Rockford stage 4
\$29
\$2A
\$2B
\$2C
\$2D
\$2E
\$2F
\$30 Butterfly position 1
\$31 Butterfly position 2

\$32 Butterfly position 3
\$33 Butterfly position 4
\$34 Butterfly position 1, scanned this frame
\$35 Butterfly position 2, scanned this frame
\$36 Butterfly position 3, scanned this frame
\$37 Butterfly position 4, scanned this frame
\$38 Rockford
\$39 Rockford, scanned this frame
\$3A Amoeba
\$3B Amoeba, scanned this frame
\$3C
\$3D
\$3E *
\$3F *

Notes on the object types

- The pre-outbox (object \$04) looks and acts exactly like a steel wall (object \$07) until Rockford has collected enough diamonds, at which point it morphs into a flashing outbox (object \$05).
- The fireflies and butterflies have four positions (can face in four directions); see the section on fireflies and butterflies for more info.
- The "scanned this frame" bit is to prevent a single object being scanned more than once per frame. For example, as a scan of the cave takes place, a boulder might be found. The boulder might fall down one space. When the scan continues, it will reach that same boulder again in its new position, so the boulder has to be marked as already scanned, to prevent it being moved more than one once per frame.
- Boulders and Diamonds have two states: stationary and falling. A falling boulder can kill Rockford whereas a stationary boulder can't.
- When something explodes, it can explode either into space (eg a firefly, Rockford) or into diamonds (butterfly). See the section on Explosions for an explanation for why the "zereth" stage exists.
- There are four pre-Rockford stages: when the inbox turns into Rockford, it takes four frames for Rockford to "be born".
- For some reason, I have character code \$06 marked for slime, \$3E for expanding wall, and \$3F for expanding wall "scanned this frame". I can't remember if these are the actual codes used in later versions of BoulderDash.

Format of the raw cave data

Jeff Bevis and Peter Broadribb have written a C program which will decode the C64's BoulderDash cave data into human readable instructions and ASCII diagrams. Things too look at are:

- `decodecaves.c` - the program that does the conversion

- cavedata.h - an include file that contains the raw cave data
- the raw cave data web page (further down this page)
- the decoded cave data web pages

Offset	Description
\$00	Cave number
\$01	Magic wall milling time, also max amoeba time at 3% growth.
\$02	Initial diamond value
\$03	Extra diamond value
\$04	Initial randomiser seed value for difficulty level 1
\$05	Initial randomiser seed value for difficulty level 2
\$06	Initial randomiser seed value for difficulty level 3
\$07	Initial randomiser seed value for difficulty level 4
\$08	Initial randomiser seed value for difficulty level 5
\$09	Diamonds needed on difficulty level 1
\$0A	Diamonds needed on difficulty level 2
\$0B	Diamonds needed on difficulty level 3
\$0C	Diamonds needed on difficulty level 4
\$0D	Diamonds needed on difficulty level 5
\$0E	Cave time for difficulty level 1
\$0F	Cave time for difficulty level 2
\$10	Cave time for difficulty level 3
\$11	Cave time for difficulty level 4
\$12	Cave time for difficulty level 5
\$13	Background colour 1
\$14	Background colour 2
\$15	Foreground colour
\$16	unused
\$17	unused
\$18	Random object number 1
\$19	Random object number 2
\$1A	Random object number 3
\$1B	Random object number 4
\$1C	Probability of object 1
\$1D	Probability of object 2
\$1E	Probability of object 3
\$1F	Probability of object 4
\$20 onwards:	cave data, delimited with \$FF.

The cave data (beginning at offset \$20) is broken into variable sized records where the first byte contains the object to store (low 6 bits only) and the kind of structure to create (upper 2 bits). Depending on the structure to create, there are 2, 4 or 5 bytes of parameters that follow.

The object codes in the low six bits are discussed elsewhere. Since the cave data can specify any of the 64 object types, this means, for example, that flies can start facing any direction.

The four structure codes are:

- 00 = store a single object
 - param byte 1: column
 - param byte 2: row
- 01 = draw a line of that object
 - param byte 1: begin column

- param byte 2: begin row
- param byte 3: length
- param byte 4: direction code (0=up, 1=up/right, 2=right, 3=down/right, 4=down, 5=down/left, 6=left, 7=up/left)
- 10 = draw a rectangle of that object, filled with a second object
 - param byte 1: top left column
 - param byte 2: top left row
 - param byte 3: length
 - param byte 4: height
 - param byte 5: object to fill with
- 11 = draw a rectangle of that object, don't modify the insides.
 - param byte 1: top left column
 - param byte 2: top left row
 - param byte 3: length
 - param byte 4: height

Decoding "randomly placed" objects

All caves have the opportunity to place up to four objects in "random" positions. They aren't really random, in the sense that the randomiser is set to a given seed value, and probabilities are checked against a given probability code; this way, objects will be "randomly" placed in the same positions each time you try the cave. Method: For each cave position (20 lines x 40 columns), pick a "random" number (0-255). If the random number is less than any of the four probability codes given at offsets \$1C to \$1F in the cave data, then put the object code given at the corresponding offset from \$18 to \$1B at the current position. (If the random number is less than more than one of the given probabilities, preference is given to the later object codes - in order of offsets \$1B, \$1A, \$19, \$18). Note that after this routine has finished, the bounding steel wall is placed over the extremities of the cave (effectively DrawRect(0,2,40,22)).

Raw Cave Data

- Cave A B C D Intermission 1
- Cave E F G H Intermission 2
- Cave I J K L Intermission 3
- Cave M N O P Intermission 4

Cave 1 (Cave A)

See also the decoded version.

```
01 14 0A 0F 0A 0B 0C 0D 0E 0C 0C 0C 0C 0C 96 6E
46 28 1E 08 0B 09 D4 20 00 10 14 00 3C 32 09 00
42 01 09 1E 02 42 09 10 1E 02 25 03 04 04 26 12
FF
```

Cave 2 (Cave B)

See also the decoded version.

```
02 14 14 32 03 00 01 57 58 0A 0C 09 0D 0A 96 6E
46 46 46 0A 04 09 00 00 00 10 14 08 3C 32 09 02
42 01 08 26 02 42 01 0F 26 02 42 08 03 14 04 42
10 03 14 04 42 18 03 14 04 42 20 03 14 04 40 01
05 26 02 40 01 0B 26 02 40 01 12 26 02 40 14 03
14 04 25 12 15 04 12 16 FF
```

Cave 3 (Cave C)

See also the decoded version.

```
03 00 0F 00 00 32 36 34 37 18 17 18 17 15 96 64
5A 50 46 09 08 09 04 00 02 10 14 00 64 32 09 00
25 03 04 04 27 14 FF
```

Cave 4 (Cave D)

See also the decoded version.

```
04 14 05 14 00 6E 70 73 77 24 24 24 24 24 78 64
50 3C 32 04 08 09 00 00 10 00 00 00 14 00 00 00
25 01 03 04 26 16 81 08 0A 04 04 00 30 0A 0B 81
10 0A 04 04 00 30 12 0B 81 18 0A 04 04 00 30 1A
0B 81 20 0A 04 04 00 30 22 0B FF
```

Cave 5 (Cave E)

See also the decoded version.

```
05 14 32 5A 00 00 00 00 04 05 06 07 08 96 78
5A 3C 1E 09 0A 09 00 00 00 00 00 00 00 00 00
25 01 03 04 27 16 80 08 0A 03 03 00 80 10 0A 03
03 00 80 18 0A 03 03 00 80 20 0A 03 03 00 14 09
0C 08 0A 0A 14 11 0C 08 12 0A 14 19 0C 08 1A 0A
14 21 0C 08 22 0A 80 08 10 03 03 00 80 10 10 03
03 00 80 18 10 03 03 00 80 20 10 03 03 00 14 09
12 08 0A 10 14 11 12 08 12 10 14 19 12 08 1A 10
14 21 12 08 22 10 FF
```

Cave 6 (Cave F)

See also the decoded version.

```
06 14 28 3C 00 14 15 16 17 04 06 07 08 08 96 78
64 5A 50 0E 0A 09 00 00 10 00 00 00 32 00 00 00
82 01 03 0A 04 00 82 01 06 0A 04 00 82 01 09 0A
04 00 82 01 0C 0A 04 00 41 0A 03 0D 04 14 03 05
08 04 05 14 03 08 08 04 08 14 03 0B 08 04 0B 14
03 0E 08 04 0E 82 1D 03 0A 04 00 82 1D 06 0A 04
00 82 1D 09 0A 04 00 82 1D 0C 0A 04 00 41 1D 03
0D 04 14 24 05 08 23 05 14 24 08 08 23 08 14 24
0B 08 23 0B 14 24 0E 08 23 0E 25 03 14 04 26 14
FF
```

Cave 7 (Cave G)

See also the decoded version.

```
07 4B 0A 14 02 07 08 0A 09 0F 14 19 19 19 78 78
78 78 78 09 0A 0D 00 00 00 10 08 00 64 28 02 00
42 01 07 0C 02 42 1C 05 0B 02 7A 13 15 02 02 14
04 06 14 04 0E 14 04 16 14 22 04 14 22 0C 14 22
16 25 14 03 04 27 07 FF
```

Cave 8 (Cave H)

See also the decoded version.

```
08 14 0A 14 01 03 04 05 06 0A 0F 14 14 14 78 6E
64 5A 50 02 0E 09 00 00 00 10 08 00 5A 32 02 00
14 04 06 14 22 04 14 22 0C 04 00 05 25 14 03 42
01 07 0C 02 42 01 0F 0C 02 42 1C 05 0B 02 42 1C
0D 0B 02 43 0E 11 08 02 14 0C 10 00 0E 12 14 13
12 41 0E 0F 08 02 FF
```

Cave 9 (Cave I)

See also the decoded version.

```
09 14 05 0A 64 89 8C FB 33 4B 4B 50 55 5A 96 96
82 82 78 08 04 09 00 00 10 14 00 00 F0 78 00 00
82 05 0A 0D 0D 00 01 0C 0A 82 19 0A 0D 0D 00 01
1F 0A 42 11 12 09 02 40 11 13 09 02 25 07 0C 04
08 0C FF
```

Cave 10 (Cave J)

See also the decoded version.

```
0A 14 19 3C 00 00 00 00 00 0C 0C 0C 0C 0C 96 82
78 6E 64 06 08 09 00 00 00 00 00 00 00 00 00 00
25 0D 03 04 27 16 54 05 04 11 03 54 15 04 11 05
80 05 0B 11 03 08 C2 01 04 15 11 00 0D 04 C2 07
06 0D 0D 00 0D 06 C2 09 08 09 09 00 0D 08 C2 0B
0A 05 05 00 0D 0A 82 03 06 03 0F 08 00 04 06 54
04 10 04 04 FF
```

Cave 11 (Cave K)

See also the decoded version.

```
0B 14 32 00 00 04 66 97 64 06 06 06 06 06 78 78
96 96 F0 0B 08 09 00 00 00 10 08 00 64 50 02 00
42 0A 03 09 04 42 14 03 09 04 42 1E 03 09 04 42
09 16 09 00 42 0C 0F 11 02 42 05 0B 09 02 42 0F
0B 09 02 42 19 0B 09 02 42 1C 13 0B 01 14 04 03
14 0E 03 14 18 03 14 22 03 14 04 16 14 23 15 25
14 14 04 26 11 FF
```

Cave 12 (Cave L)

See also the decoded version.

```
0C 14 14 00 00 3C 02 3B 66 13 13 0E 10 15 B4 AA
A0 A0 A0 0C 0A 09 00 00 00 10 14 00 3C 32 09 00
42 0A 05 12 04 42 0E 05 12 04 42 12 05 12 04 42
16 05 12 04 42 02 06 0B 02 42 02 0A 0B 02 42 02
0E 0F 02 42 02 12 0B 02 81 1E 04 04 04 00 08 20
05 81 1E 09 04 04 00 08 20 0A 81 1E 0E 04 04 00
08 20 0F 25 03 14 04 27 16 FF
```

Cave 13 (Cave M)

See also the decoded version.

```
0D 8C 05 08 00 01 02 03 04 32 37 3C 46 50 A0 9B
96 91 8C 06 08 0D 00 00 10 00 00 00 28 00 00 00
25 12 03 04 0A 03 3A 14 03 42 05 12 1E 02 70 05
13 1E 02 50 05 14 1E 02 C1 05 15 1E 02 FF
```

Cave 14 (Cave N)

See also the decoded version.

```
0E 14 0A 14 00 00 00 00 1E 23 28 2A 2D 96 91
8C 87 82 0C 08 09 00 00 10 00 00 00 00 00 00
81 0A 0A 0D 0D 00 70 0B 0B 0C 03 C1 0C 0A 03 0D
C1 10 0A 03 0D C1 14 0A 03 0D 50 16 08 0C 02 48
16 07 0C 02 C1 17 06 03 04 C1 1B 06 03 04 C1 1F
06 03 04 25 03 03 04 27 14 FF
```

Cave 15 (Cave O)

See also the decoded version.

```
0F 08 0A 14 01 1D 1E 1F 20 0F 14 14 19 1E 78 78
78 78 8C 08 0E 09 00 00 00 10 08 00 64 50 02 00
42 02 04 0A 03 42 0F 0D 0A 01 41 0C 0E 03 02 43
0C 0F 03 02 04 14 16 25 14 03 FF
```

Cave 16 (Cave P)

See also the decoded version.

```
10 14 0A 14 01 78 81 7E 7B 0C 0F 0F 0F 0C 96 96
96 96 96 09 0A 09 00 00 10 00 00 00 32 00 00 00
25 01 03 04 27 04 81 08 13 04 04 00 08 0A 14 C2
07 0A 06 08 43 07 0A 06 02 81 10 13 04 04 00 08
12 14 C2 0F 0A 06 08 43 0F 0A 06 02 81 18 13 04
04 00 08 1A 14 81 20 13 04 04 00 08 22 14 FF
```

Cave 17 (Intermission 1)

See also the decoded version.

```
11 14 1E 00 0A 0B 0C 0D 0E 06 06 06 06 06 0A 0A
0A 0A 0A 0E 02 09 00 00 00 14 00 00 FF 09 00 00
87 00 02 28 16 07 87 00 02 14 0C 00 32 0A 0C 10
0A 04 01 0A 05 25 03 05 04 12 0C FF
```


Cave 18 (Intermission 2)

See also the decoded version.

```
12 14 0A 00 0A 0B 0C 0D 0E 10 10 10 10 10 0F 0F
0F 0F 0F 06 0F 09 00 00 00 00 00 00 00 00 00 00
87 00 02 28 16 07 87 00 02 14 0C 01 50 01 03 09
03 48 02 03 08 03 54 01 05 08 03 50 01 06 07 03
50 12 03 09 05 54 12 05 08 05 50 12 06 07 05 25
01 04 04 12 04 FF
```

Cave 19 (Intermission 3)

See also the decoded version.

```
04 0A 00 0A 0B 0C 0D 0E 0E 0E 0E 0E 0E 14 14 14
14 14 06 08 09 00 00 00 00 00 00 00 00 00 00 87
00 02 28 16 07 87 00 02 14 0C 00 54 01 0C 12 02
88 0F 09 04 04 08 25 08 03 04 12 07 FF
```

Cave 20 (Intermission 4)

See also the decoded version.

```
14 03 1E 00 00 00 00 00 00 06 06 06 06 06 14 14
14 14 14 06 08 09 00 00 00 00 00 00 00 00 00 00
87 00 02 28 16 07 87 00 02 14 0C 01 D0 0B 03 03
02 80 0B 07 03 06 00 43 0B 06 03 02 43 0B 0A 03
02 50 08 07 03 03 25 03 03 04 09 0A FF
```