

Solstice level data documentation

Solstice level data and some misc data documentation by JP32.

PM @ J^P on romhacking.net if you have any questions.

This is made with open-office 4 in case this doesn't show up properly.

One super important thing is [tetraspace's map](#), super handy AND it has all room ID's for all rooms (its hard to see since its in dark gray against black background, but its there).

- Room data always starts with Room attributes, 3 bytes for rooms palette to be specif
- If the room doesn't sprites, then its just straight out skipped(goes from room attributes to graphics).
- If the room doesn't have any kind of blocks, their amount bits are just set "00" and then "00" again(three 00 in row)
- There are few Unknown bits, I have no idea what they do other than fucking the whole room up or doing nothing.
- I have no idea how the game layouts all rooms in top-down perspective(the pause menu map)
- Things like list of all sprites are listed in separate text files
- All rooms are back-to-back on the rom(one exception for the ending cut-scene room)
- Room used in the opening cut-scene doesn't use normal room data AFAIK.
- Ending cut-scene is normal room but some data(palette and sprites) are overwritten.
- Very first room starts at hex-address **18398**

Room data order is:

- Room attributes 5 bytes
- Sprite(s) 5 bytes +5 for each additional one (seems like three is max?)
- Graphics 13 bytes
- Door(s) and exit(s) 1 byte, +2 for each exit in the room
- Normal block amount 1 byte
- Normal block(s) 4 bytes **or** 16 bytes each
- Special block amount 1 byte
- Special block(s) 16 bytes each

In detail, byte by byte:

Room attributes:

- 1 Room palette 1
- 2 Room palette 2
- 3 Room palette 2
- 4 Room size
- 5 Unknown

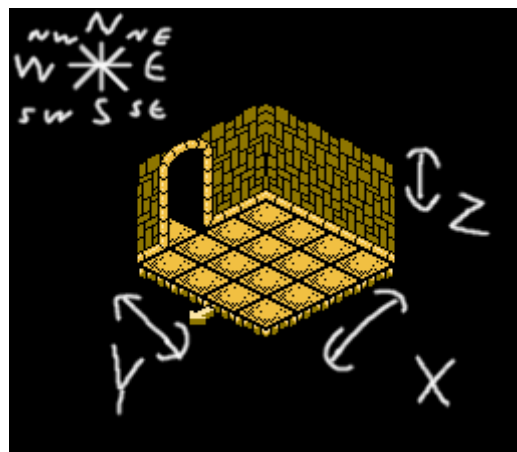
**Unknown bit 5 seems to just fuck whole room up?*

**Room sizes are listed further into this doc.*

Sprites:

- 1 Sprite type
 - 2 Sprites X and Y position
 - 3 Sprites Direction and Z position
 - 4 Sprites palette 1
 - 5 Sprites palette 2
- +5 more for each additional sprite**

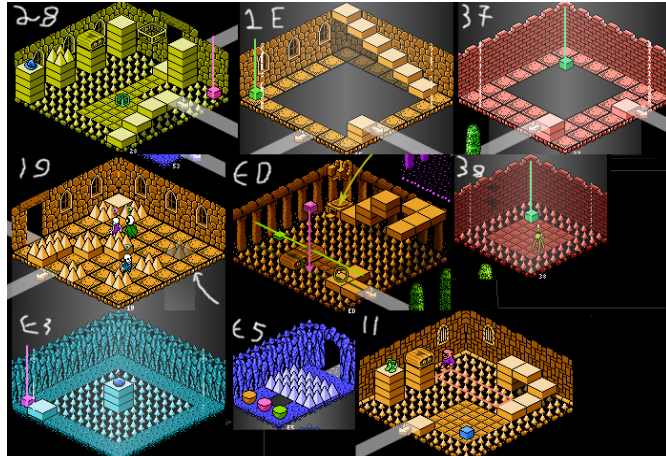
**SW to NE or NW to SE if its odd or even(0 or 1)*



**If room doesn't have sprites then this is skipped*
**List of Sprite types is in separate text file because its long.*

Walls, Floors, their graphics and types:

- 1 NW wall Pattern
- 2 NE wall Pattern
- 3 NW wall Graphics
- 4 NE wall Graphics
- 5 Floor Type
- 6 Floor Graphics
- 7 Unknown
- 8 Unknown
- 9 Unknown
- 10 Unknown
- 11 Unknown
- 12 Unknown
- 13 Unknown



**Rooms 28, 1E, 37, 19, ED, 38, E3, E5 and 11 seems to use different format since they have two floor graphics and types, they use bits 7-13 but I couldn't figure it out, probably hard-coded to some degree or maybe Im just overthinking it, in any case I dont know how it works.*

**list of wall/floor graphics and types are listed further in this doc.*

Exits:

- 1 unknown
- 2 exit Z and X or Y position
- 3 exit destination
- +2 more for each exit the room has

**Unknown bit 1 seems to just fuck whole room up?*

Normal blocks:

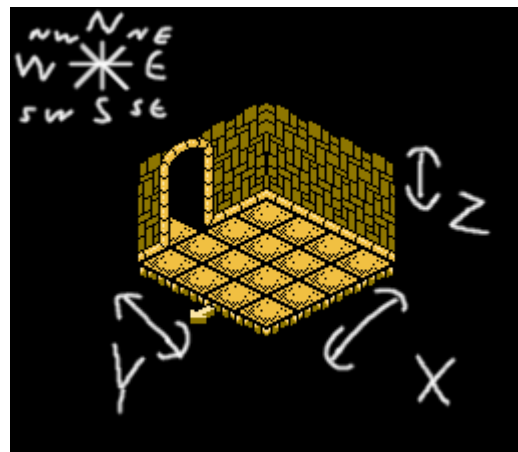
- 1 How many normal blocks to load

If its normal block with no manual graphics fixes:

- 1 block graphics
- 2 block XY position (SW/NE, NW/SE)
- 3 block Z position (UP/DOWN)
- 4 unknown, its 80 if it has no manual graphics fixes?

+4 for each additional block (if it ends in 80)

**list of block graphics are listed further into this doc.*



If its normal block with manual graphics fixes:

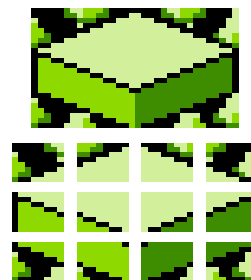
- 1 block graphics
- 2 block XY position (SW/NE, NW/SE)
- 3 block Z position (UP/DOWN)
- 4 unknown, its 00 if it has manual graphics fixes?

- 1 blocks top graphics 1 (8x8) (left side)
- 2 blocks top graphics 2 (8x8) (middle-left)
- 3 blocks top graphics 3 (8x8) (middle-right)
- 4 blocks top graphics 4 (8x8) (right side)

- 1 blocks middle graphics 1 (8x8) (left side)
- 2 blocks middle graphics 2 (8x8) (middle-left)
- 3 blocks middle graphics 3 (8x8) (middle-right)
- 4 blocks middle graphics 4 (8x8) (right side)

- 1 blocks bottom graphics 1 (8x8) (left side)
- 2 blocks bottom graphics 2 (8x8) (middle-left)
- 3 blocks bottom graphics 3 (8x8) (middle-right)
- 4 blocks bottom graphics 4 (8x8) (right side)

+16 for each block that has manual graphics fixes in them



Special blocks:

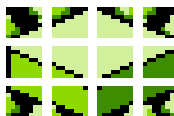
- 1 How many special blocks to load
- 1 Block graphics
- 2 Block XY position (SW/NE, NW/SE)
- 3 Block Z position (UP/DOWN)
- 4 Special block type

- 1 blocks top graphics 1 (8x8) (left side)
- 2 blocks top graphics 2 (8x8)
- 3 blocks top graphics 3 (8x8)
- 4 blocks top graphics 4 (8x8) (right side)

- 1 blocks middle graphics 1 (8x8) (etc etc..)
- 2 blocks middle graphics 2 (8x8)
- 3 blocks middle graphics 3 (8x8)
- 4 blocks middle graphics 4 (8x8)



- 1 blocks bottom graphics 1 (8x8)
- 2 blocks bottom graphics 2 (8x8)
- 3 blocks bottom graphics 3 (8x8)
- 4 blocks bottom graphics 4 (8x8)

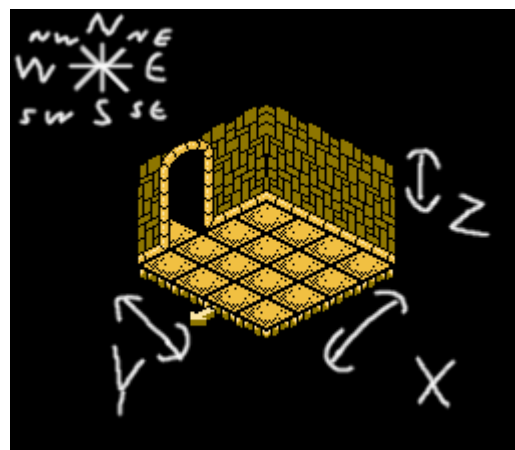


+16 for each special block

**list of special block types and graphics are listed further into this doc.*

End of level data

**(sometimes theres extra 00 at end, sometimes theres not, I dont know why).*



So how do you find specif room that I didn't list?

Its easy.

1. Open up FCEUX and Solstice rom
2. From **debug** drop-down menu open: **PPUview**, **Code/data logger** and **Hex editor**
3. From **hex editor**, on **view** drop-down menu select **rom file**
4. Now on the **emulator** go to the room before the room you're looking for
5. On **code data logger**, press **reset log**, then **start**
6. Now on **emulator** move to the room you want(from the previous room)
7. now press **pause** on **code data logger**
8. Switch to **hex editor**, press **control+F**, type in top-left palette from **PPU view**(hover cursor over the color to see colors HEX number), type it to the search box ignoring the transparency/first color, it should be something like **"1C 0C 2C"**
(Or alternatively you can just manually scroll down the hex editor, its just about as quick tbh)
9. Since there are multiple rooms that shares same palette(they all are individual, even if they share same palette!), keep pressing **find next**(or **F3**) until you find area thats highlighted in **blue**.
10. It should look something like this:
..B0 B1 B2 B3 B4 1C 0C 2C 00 01 08 C0 03 17 27 54 54 07 07 05 05 FF FF FF FF FF FF 19 F4 00 03 FE
02 F2 08 0C 11 00 80 0B 11 03 80 0C 15 00 80 0B 15 03 80 0C 51 00 80 0B 51 03 80 0C 55 00 80 0B 55 03
00 AD D5 D6 C8 81 D7 D8 84 AD AE C7 C8 00 1C 0C 2C 00 02...
11. Since room data always starts with palette..
12. Write down the rooms offset and what room number it is for convince

To find out what the room ID is, just look at the tetraspace map or look at ram address 15.

This is outdated since I've listed all rooms in the seperate text file.. I included this just in case if theres any stray rooms outside of the usual bank. Also its worth to note Japanese rom has some differences, mostly in the last few rooms. European rom is identical(room-wise) to US rom outside of one potion having fucked up graphics.

Room sizes

Hex	X Y
00	7x7 (largest)
01	7x6
02	6x7
03	7x5
04	5x7
05	7x4
06	4x7
07	7x3
08	3x7
09	6x6
0A	6x5
0B	5x6
0C	6x4
0D	4x6
0E	6x3
0F	3x6
10	5x5
11	5x4
12	4x5
13	5x3
14	3x5
15	4x4
16	4x3
17	3x4
18	3x3 (smallest)

**Anything beyond 18 is invalid/buggy.*

Graphics and types for floors, walls and blocks

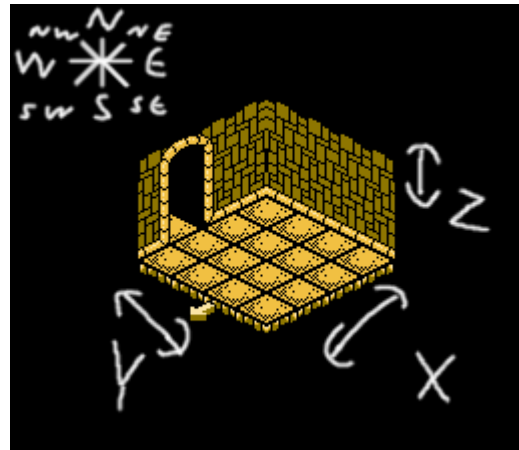
Floor graphics:s

- 00 Basic tiles
- 01 Small spikes
- 02 Dithered small tiles
- 03 dot-pattern
- 04 dot-pattern but different outline
- 05 rounded-edge tiles
- 06 Cave dot pattern
- 07 Dithered small&large tiles
- 08 small-tiles
- 09 small spikes but different outline
- 0A hole/nothing
- 0B another dot-pattern with different outline

Floor types:

- 00 normal
- 0A hole
- 81 small spikes

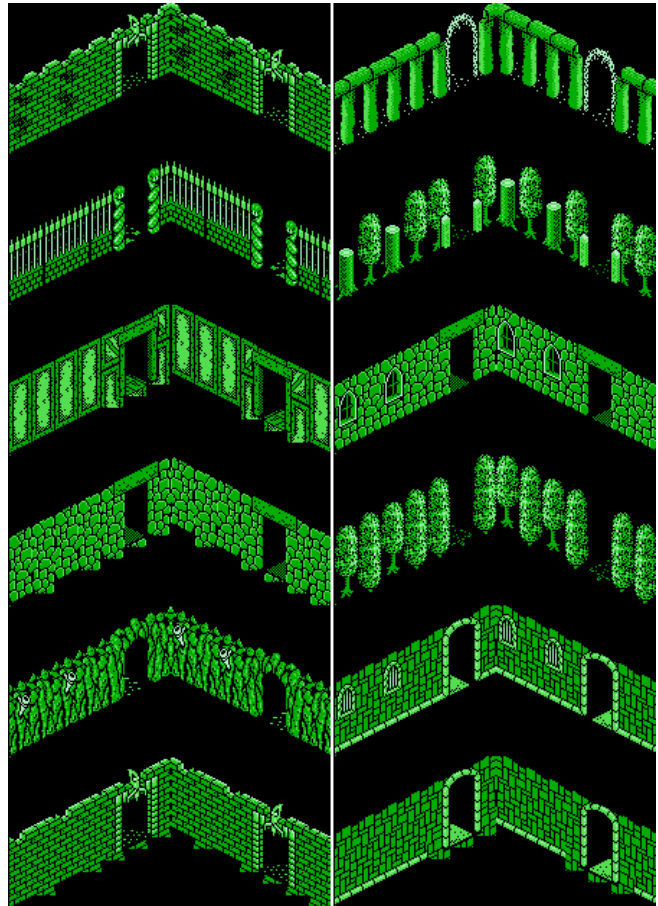
**needs double-checking, but these values should work for rooms with just one floor-type.*



Wall Graphics:

- 00 Main castle 1 + windows (starting areas)
- 01 Main castle 1 no windows
- 02 caves of doomed souls + torch
- 03 Wood-like looking wall + window (gardens of pain)
- 04 Castle-walls with odd shading (tower of tears)
- 05 Same as above but normal
- 06 Blood-gardens
- 07 Main castle 2 + windows (later areas)
- 08 stonehedge gardens
- 09 stonehedge gardens but taller
- 0a stonehedge gardens but middle?
- 0b stonehedge gardens but lower?
- 0c gardens of pain cut trees
- 0d gardens of pain normal trees
- 0e gardens of pain normal trees but taller
- 0f caves of doomed souls no torches and tall
- 10 same as above?
- 11 same as above again?
- 12 Mains castle 2 no windows (later areas)
- 13 same as above?
- 14 blood gardens again?
- 15 Blood gardens but tall
- 16 Gardens of pain wood but tall

*Note: tall = if theres door on NW or NE wall
(Z axis on door higher than 0) then tall variation is used*



Block Graphics:

- 00 Basic square block
- 01 Oreo circle
- 02 marshmallow block
- 03 escalator block (NW/SW)
- 04 escalator block (SW/NE)
- 05 outline/transparent block
- 06 Square oreo block
- 07 square fired-egg
- 08 duplicate of 07
- 09 upside-down square fired-egg
- 0a spikes
- 0b chimney
- 0c chimneys base
- 0d cactus's(?) base
- 0e cactus's(?) tip
- 0f top of teleporter
- 10 base of the teleporter
- 11 nothing/transparent

Special block attributes:

- 00 escalator that pushes to SE
- 01 escalator that pushes to SW
- 02 escalator that pushes to NW
- 03 escalator that pushes to NE
- 04 block that disappears when you jump in top of it
- 05 invisible block that appears when you jump in top of it
- 06 large spikes
- 07 teleporter (the bottom part)

