

# JSWED v2.3.7

John Elliott

April 7, 2015

## **Abstract**

JSWED is an editor for games based on the *Jet Set Willy* and *Manic Miner* engines. This document describes how to use it.

## Contents

<b>1</b>	<b>Changes since previous versions</b>	<b>7</b>
1.1	Since 2.3.5, the last unstable version . . . . .	7
1.2	Since 2.2.9, the last stable version . . . . .	7
<b>2</b>	<b>Legal Bits</b>	<b>7</b>
<b>3</b>	<b>Installing JSWED</b>	<b>8</b>
3.1	Installing under Windows . . . . .	8
3.2	Installing under UNIX . . . . .	8
3.3	Installing under MacOS X . . . . .	8
3.4	Installing under BeOS . . . . .	8
<b>4</b>	<b>Running JSWED</b>	<b>8</b>
4.1	Load file . . . . .	9
4.1.1	+3DOS file support . . . . .	9
4.2	Save file . . . . .	9
4.3	Edit game . . . . .	9
4.4	Configure . . . . .	9
4.5	About JSWED . . . . .	9
4.6	Leave program . . . . .	10
<b>5</b>	<b>The user interface</b>	<b>10</b>
<b>6</b>	<b>Terminology</b>	<b>10</b>
<b>I</b>	<b>Jet Set Willy (JSW48 and JSW128)</b>	<b>11</b>
<b>7</b>	<b>Editing games</b>	<b>11</b>
7.1	Game . . . . .	11
7.1.1	Upgrading to 128k / Converting to JSW64 . . . . .	11
7.1.2	Adjacent ropes . . . . .	12
7.1.3	Black Willy . . . . .	12
7.1.4	Fall any height . . . . .	12
7.1.5	No auto pause . . . . .	12
7.1.6	Mono . . . . .	12
7.1.7	Upside-down . . . . .	12
7.1.8	Items wobble . . . . .	12
7.1.9	Items jiggle . . . . .	12
7.1.10	Fix bug in 'fast JSW' . . . . .	12
7.1.11	Softtricks death . . . . .	13
7.1.12	Attract mode . . . . .	13
7.1.13	Display room number . . . . .	13
7.1.14	Safe restart places (48k games only) . . . . .	13
7.1.15	Fix victory bug (Henry's Hoard only) . . . . .	13
7.2	Message . . . . .	13
7.3	Memory . . . . .	13
7.4	Sprites . . . . .	13
7.4.1	Importing / Exporting . . . . .	14

7.5	The Room Editor . . . . .	14
7.5.1	Moving between rooms . . . . .	15
7.5.2	Editing exits . . . . .	15
7.5.3	Clipboard . . . . .	15
7.5.4	Import and Export . . . . .	16
7.5.5	Room options . . . . .	16
7.5.6	Guardians . . . . .	16
7.5.7	Conveyor options . . . . .	16
7.5.8	Guardian / guardian start frame . . . . .	16
7.5.9	Cell editors . . . . .	17
7.6	Teleport editor . . . . .	17
7.7	Start position editor . . . . .	17
7.8	Font editor . . . . .	17
7.8.1	Importing / Exporting . . . . .	17
7.9	Screen editor . . . . .	18
7.9.1	The Zoom window . . . . .	18
7.9.2	Editing pixels . . . . .	18
7.9.3	Editing attributes . . . . .	18
7.9.4	Grids . . . . .	18
7.9.5	Next / Previous . . . . .	18
7.9.6	Undo . . . . .	19
7.9.7	Import / Export . . . . .	19
7.10	The Title Screen Editor . . . . .	19
7.11	The Music editor . . . . .	20
7.11.1	Using the original tune player . . . . .	20
7.11.2	Using the SoundTracker player . . . . .	20
7.12	The Hex editor . . . . .	21
<b>8</b>	<b>The Guardian editor</b>	<b>22</b>
8.1	On Guardian Classes . . . . .	22
8.2	Editor overview . . . . .	22
8.3	Guardian types . . . . .	22
8.4	Editing conventional guardians . . . . .	23
8.5	Rope editing . . . . .	23
8.6	Arrow editing . . . . .	23
8.7	Hex editing . . . . .	23
8.8	Geoff Mode bounds . . . . .	23
<b>9</b>	<b>Good Advice</b>	<b>23</b>
<b>II</b>	<b>Manic Miner</b>	<b>25</b>
<b>10</b>	<b>Editing Games</b>	<b>25</b>
10.1	Game . . . . .	25
10.2	Message . . . . .	25
10.3	Sprites . . . . .	25
10.4	The Room Editor . . . . .	26
10.4.1	Moving between rooms . . . . .	26
10.4.2	Guardians . . . . .	26

10.4.3	Editing cells . . . . .	26
10.4.4	Room options . . . . .	26
10.4.5	Conveyor options . . . . .	26
10.4.6	Item options . . . . .	27
10.5	Guardians page . . . . .	27
10.6	Portal editor . . . . .	27
10.7	Screen editor . . . . .	28
<b>11</b>	<b>The Guardian editor</b>	<b>28</b>
11.1	No Guardian Classes . . . . .	28
11.2	Editor overview . . . . .	28
11.3	Guardian types . . . . .	28
<b>III</b>	<b>JSW64 games</b>	<b>28</b>
<b>12</b>	<b>JSW64 Overview</b>	<b>29</b>
12.1	Variants . . . . .	29
<b>13</b>	<b>Memory Map</b>	<b>29</b>
<b>14</b>	<b>Rooms</b>	<b>29</b>
14.1	Cells . . . . .	29
14.1.1	Extra cell types . . . . .	30
14.2	Room properties . . . . .	30
14.3	Room-specific graphics . . . . .	31
14.4	Conveyor animation . . . . .	31
14.5	Final Barrier . . . . .	31
<b>15</b>	<b>Guardians</b>	<b>31</b>
15.1	Guardian editor . . . . .	32
15.1.1	Skylabs . . . . .	32
15.1.2	Angry Eugenies . . . . .	32
15.1.3	Triggers . . . . .	32
15.1.4	Switches . . . . .	33
15.1.5	Vanishing Walls . . . . .	33
15.1.6	Stoppers . . . . .	33
15.2	Guardian table . . . . .	33
<b>16</b>	<b>Teleport editor</b>	<b>34</b>
<b>17</b>	<b>Start position / Portal editor</b>	<b>34</b>
17.1	Portal special effect . . . . .	35
<b>18</b>	<b>JSW64 Good advice</b>	<b>35</b>
<b>IV</b>	<b>Editing Jet Set Willy 2</b>	<b>35</b>

<b>19 Editing Games</b>	<b>35</b>
19.1 Game	36
19.2 Message	36
19.3 Sprites	36
19.4 Cells	36
19.5 The Room Editor	37
19.5.1 Guardians	37
19.5.2 Editing cells	37
19.5.3 Room options	37
19.6 Room Specials Editor	39
19.6.1 Editing teleporters	39
19.7 The Title Screen Editor	39
<b>20 The Guardian editor</b>	<b>40</b>
20.1 Guardian behaviour notes	40
20.2 Editor overview	40
<b>A JSW64 for machine-code programmers</b>	<b>41</b>
A.1 JSW64 Room Formats	41
A.1.1 Bytes #00-#B5 for formats V and X	41
A.1.2 Bytes #00-#B5 for formats W, Y and Z	42
A.1.3 Bytes #00-#B5 for format [	42
A.1.4 Bytes #B5-#FF (all formats)	42
A.1.5 Bytes #100 onwards: Format V	44
A.1.6 Bytes #100 onwards: Format W	44
A.1.7 Bytes #100 onwards: Format X	44
A.1.8 Bytes #100 onwards: Format Y	44
A.1.9 Bytes #100 onwards: Format Z	44
A.1.10 Bytes #100 onwards: Format [	44
A.2 Cell Types Table (formats W, Y, Z)	44
A.3 Cell Types Table (format [)	45
A.4 JSW64 Guardians	45
A.4.1 Horizontal and Diagonal Guardians	47
A.4.2 Ropes	47
A.4.3 Arrows	47
A.5 Patch Vectors	47
A.5.1 Patch Vector Jumpblock	48
A.5.2 Music Jumpblock (original music player)	48
A.5.3 Music Jumpblock (SoundTracker)	49
A.6 Guardian Patch Vectors	49
A.7 Memory Map	50
A.7.1 Banks 1,3,4,6	51
A.7.2 Bank 7 memory map	51
<b>B Format of exported room definitions</b>	<b>51</b>
B.1 Overall format	51
B.2 <room> node	52
B.2.1 <layout>	52
B.2.2 <exits>	52
B.2.3 <items>	53

B.2.4	<guardians> . . . . .	53
B.2.5	<portal> . . . . .	54
B.2.6	<hints> . . . . .	54
B.3	<sprite> node . . . . .	54
<b>C</b>	<b>Assembly file format</b>	<b>54</b>

# 1 Changes since previous versions

## 1.1 Since 2.3.5, the last unstable version

1. File chooser rewritten in the Windows version to allow going 'up' from the root of a drive to a list of drives.
2. Validation in the file chooser improved, hopefully reducing the risk of crashes.

## 1.2 Since 2.2.9, the last stable version

1. The window size has been increased to 800x600. Many of the screens have been redesigned to take advantage of the greater space this offers.
2. JSWED can edit Mihai Novitchi's 128k version of Manic Miner.
3. JSWED can edit games based on the Jet Set Willy 2 engine (currently only snapshots are supported, not TAP or +3DOS).
4. There is a specialised title screen editor for JSW and JSW2, supporting the way that these games build up the "impossible triangle" graphic.
5. Rooms in JSW48/128/64, Manic Miner and JSW2 games can be exported and imported.
6. Fonts can now be exported and imported.
7. The navigation in the room pages now works in step; if you go to a room on (say) the "Room" screen, and then switch to the "Tele" screen, you'll still be in the same room.
8. It is possible to adjust the speed of animations in the room editors, using the keypad +/- keys or F11/F12.
9. 'Trap' cells can be set up in JSW64 games.
10. Ian Collier's music player can be replaced with Soundtracker, if so desired. 2.3.6 fixes at least three bugs in the tune import code for this screen.
11. "Home directory" and "Root directory" shortcut buttons have been added to the file chooser.
12. The Manic Miner sprite editor can now import and export sprites.
13. Various buffer overflows have been found and corrected.

# 2 Legal Bits

This version of JSWED is released under the GNU General Public License(sic). If you are just using JSWED, this won't affect you; but if you intend to distribute copies, then you must distribute all the source code as well as the programs. See the file COPYING (in the Windows version, COPYING.TXT) for details.

The Z80 patches (room\_\*.hex, rtime\*.bin) which are loaded as part of a JSW64 game are not covered by the GPL (since this would make it impossible to link them with the JSW engine). They are released under the zlib/libpng licence.

The Soundtracker patches include the SoundTracker player by BZYK (Piotr Baczekiewicz); to the best of my knowledge and belief, this is public domain. They also include two tunes (versions of 'In The Hall Of The Mountain King' and 'If I Was A Rich Man') by Gasman; if you distribute one or both of these with your game, please credit him appropriately.

## **3 Installing JSWED**

### **3.1 Installing under Windows**

This release of JSWED should run on anything Win32ish, from Windows 95 up to Windows 7.

If you are troubled by frequent crashes, this is probably caused by bugs in JSWED. However, it may also be an idea to stop JSWED using DirectX, to see if this improves stability. To disable DirectX, add the following line to your AUTOEXEC.BAT file:

```
SET SDL_VIDEODRIVER=windib
```

and reboot.

### **3.2 Installing under UNIX**

Provided you have the SDL, SDL\_image and libxml libraries on your system, you should just be able to get away with

```
./configure
make
su root
make install
```

### **3.3 Installing under MacOS X**

Building under OS X is just the same as any other UNIX. Make sure the prerequisites are installed, open a terminal, change to the directory where you unpacked JSWED, and:

```
./configure
make
sudo make install
```

### **3.4 Installing under BeOS**

This should behave the same way as UNIX, though you may have to use `-prefix=` options in the `'./configure'` line.

## **4 Running JSWED**

In Windows, double-click JSWED.EXE or start it from the Start menu. In UNIX, type `jswed` at a command prompt, or set up a shortcut to it in your preferred desktop environment.

When JSWED is started, the main window will appear. There will be six menu options:



## 4.1 Load file

Brings up a file selector from which you can choose the file to load. JSWED supports +3DOS, TAP, SNA and Z80 formats. Note that 128k .SNA files can be saved and loaded by JSWED, but may not be compatible with emulators. If you intend to use +3DOS files you should read the notes below (section 4.1.1).

To change to a different drive (Windows only), tab to the filename field at the bottom of the screen, clear out what's in there (Press CTRL+Y) and type its drive name, followed by a colon. Then press RETURN.

Once a file is successfully loaded the top line of the screen shows the loaded file and its type (48k, 128k, Softricks, Henry's Hoard, etc.).

### 4.1.1 +3DOS file support

As far as 48k games go, this is quite easy. The file format JSWED uses is a single file, CODE 32768,32768. The fun comes with 128k games, because they are stored in multiple +3DOS files.

When loading or saving, JSWED must be given the filename of the BASIC loader (JetSet.128). The other files that make up JSW128 are assumed to be in the same directory as the loader, and have the standard +3DOS filenames for JSW files (rtime.js2, rooms.js6 etc.) Be especially careful when saving, because files with these names will be overwritten without asking.

The loader program (JetSet.128) saved in +3DOS mode is a 'universal' loader that ought to work on cassette or disc.

## 4.2 Save file

Brings up the same sort of file selector, so you can save the game. The type of file used will be the same as the one used in loading, unless you changed it while editing.

## 4.3 Edit game

This is the heart of JSWED and is described fully in section 7.

## 4.4 Configure

This is used to set up the Spectrum emulator used for testing, and the Autosave feature. There are three fields to enter:

1. The Spectrum emulator filename; you should enter the full path to the emulator. The emulator in question must be able to load .Z80 files.
2. The Autosave filename. This should be the name of a .Z80 snapshot, which will be saved at the specified interval.
3. The Autosave interval, in seconds. Enter 0 to disable Autosave completely.

## 4.5 About JSWED

Displays the current version, and supported game types.

## 4.6 Leave program

Will leave JSWED instantly. Unsaved work will be lost. Closing the JSWED window also has this effect.

## 5 The user interface

The JSWED user interface is based on the Spectrum 128 editor, plus features from more recent GUIs. On most screens, the current control will be highlighted by a blue dotted border, and you can use TAB to move from one to the next. Text controls contain a blue/white flashing cursor, which can be moved by the cursor keys (the control will scroll if there's more text than can be seen).

In the menus, use the cursor keys and ENTER to choose an option; PageUp/PageDown allow scrolling menus to move a page at a time; Shift+Home/Shift+End to move to the top or bottom. Press ESCAPE to cancel the menu.

## 6 Terminology

In a Manic Miner / Jet Set Willy game, you may expect to see the following things in a room:

**Willy.** Our hero.

**Guardians.** Most of these are 16x16 shapes which move about and kill Willy on contact; ropes and arrows are also guardians, as are switches in JSW64 games.

**Cells.** Cells are the 8x8 building blocks used to construct the shape of the room. Floors, walls, conveyors, ramps, those annoying little bushes that kill Willy when he touches them - all of these are composed of one or more cells. Switches in Manic Miner and JSW2 are also cells.

**Items.** These are the glowing things that Willy has to collect. Like cells, they're an 8x8 graphic.

**Portals.** A 16x16 shape; unlike the guardians, it doesn't move or animate. If Willy enters it while it's flashing, he is transported to another level.

**Oddities.** In JSW games, Maria looks and behaves like a guardian, but she isn't. The same goes for Eugene in Manic Miner games.

There are some additional concepts which you don't normally see in a room, but are important for editing games:

**Guardian class:** In JSW48 and JSW128 games, the guardians you see in each room are instances of particular guardian classes. Changing the class affects all instances.

**Cell class:** The cells, as mentioned above, are the building blocks of the room. A room has a limited number of cell classes; for example, the original JSW48 has six. Look at any room in the original JSW, and you will see no more than six combinations of colour and pattern. Changing the cell class will affect the colour and pattern of all cells of that class.

## Part I

# Jet Set Willy (JSW48 and JSW128)

## 7 Editing games

The game editor is composed of a series of pages, with a list in the top left-hand corner. To choose a page, click on its title or use ALT+UP / ALT+DOWN to choose between them.

Each page has a context menu, which can be accessed by right-clicking the screen or pressing the “menu” key (next to the bottom right-hand CTRL key).

At any time while editing a game, you can press F2 to save your file, or F5 to run it in a Spectrum emulator.

The screens are:

### 7.1 Game

This page allows you to change the file format used for saving (note that 128k games saved in SNA format can be reloaded in JSWED, but the 128k SNA format may not be compatible with emulators). The page also contains a list of patches which can be applied to the game engine. Each patch is accompanied by a tickbox which is either ticked (the patch is applied), empty (the patch can be applied) or shaded (the patch cannot be applied).

#### 7.1.1 Upgrading to 128k / Converting to JSW64

At the bottom of the Game screen is a button. In 48k games it is marked “Upgrade to 128k”; in 128k games, it reads “Convert to JSW64”. If you click it, it will offer a choice of formats to convert your game to:

Upgrade to	Rooms	Guardians / room	Cell classes / room	Free space / room
JSW128	256	8	4	0
JSW64 V	128	13	8	64 bytes
JSW64 W	128	8	13*	0
JSW64 X	64	13**	8	576 bytes
JSW64 Y	64	8**	13*	512 bytes
JSW64 Z	64	8**	13***	256 bytes
JSW64 [	64	4**	16	512 bytes

\* There are 13 cell classes defined per room, and three ‘global’ ones which are the same in all rooms.

\*\* The guardian table can be moved into the free space, giving you up to 32 guardians per room.

\*\*\* JSW64 Z allows any attribute to be used in a room; so a cell can belong to one of 13 cell classes or behave like one of 240 differently-coloured ‘water’ blocks.

Click it to upgrade a 48k game to 128k. If you click it and then confirm your request, you will see all the upgrades being applied and then be returned to the main menu. Geoff Mode games and Henry’s Hoard games cannot be upgraded to 128k. Games

patched in other ways (for example, with Andrew Broad's custom sprite code) may upgrade cleanly to JSW128, but will not upgrade to JSW64.

*Warning: If you convert a JSW128 game to JSW64, only the first 128 (or 64, depending on the format you chose) rooms will be converted. The others will be lost.*

The patches that can be applied are:

### **7.1.2 Adjacent ropes**

Normally, if you want multiple ropes in a room, you have to put blank guardians between them. If you turn this patch on, it becomes possible to have all eight guardian slots occupied by ropes. Note that there are (old) and (new) versions of this patch; The (old) one was supplied in previous JSWED versions; the (new) one is the one you should use in new games.

### **7.1.3 Black Willy**

Make Willy black. You will need to set backgrounds to light colours.

### **7.1.4 Fall any height**

Stop Willy dying if he falls too far.

### **7.1.5 No auto pause**

Normally JSW pauses if left by itself for too long. To disable it, apply this patch.

### **7.1.6 Mono**

The entire game area will be drawn in white on black, like the Dragon 32 version of JSW.

### **7.1.7 Upside-down**

Break the player's brain by turning the entire game engine upside-down.

### **7.1.8 Items wobble**

Collectable items wobble from side to side. Designed for use in conjunction with the 'Monochrome' patch, since items no longer colour-cycle if the screen is monochrome.

### **7.1.9 Items jiggle**

The same sort of thing as the previous patch, except that the items move up and down rather than right and left.

### **7.1.10 Fix bug in 'fast JSW'**

This should be used in conjunction with Mark Woodmass's "marginally faster" Jet Set Willy engine. It moves the rope coordinates table from #8300 to #9F00, so that the ropes are no longer all over the place. If you use it on a normal JSW engine, it will do no harm but you lose the use of the sprite page at #9F00.

#### **7.1.11 Softricks death**

For some reason games written with the Softricks editor show the title screen briefly when Willy dies, and the “death” sound effect is different. This patch switches the effect on and off.

#### **7.1.12 Attract mode**

After the title screen music has played, all the screens in the game will be displayed in turn.

#### **7.1.13 Display room number**

Displays the current room number in the bottom right-hand corner of the screen. You will have to set the colours of this bit of screen to something other than black on black, or the numbers will be invisible.

#### **7.1.14 Safe restart places (48k games only)**

Switches to Ian Collier’s “No infinite death” mode, in which every room definition contains a safe start position. If Willy dies, he will reappear in this position.

Note that if you turn this effect on, on a game which previously didn’t have it, it is *essential* to set up a valid restart position in every room, and to set the initial room again. Otherwise the game will crash on startup or when Willy loses a life.

#### **7.1.15 Fix victory bug (Henry’s Hoard only)**

The earlier (1985) version of Henry’s Hoard doesn’t show the victory screen when you win the game. Put a tick in this box to fix the bug.

### **7.2 Message**

This page contains messages printed by the game - the title screen message, the “Game Over” message, and the “objects collected” message. You can edit them as you like; the program will not allow you to enter more text than will fit.

If you are editing Henry’s Hoard, then the messages will all be in capitals. The game engine can only print in capitals, so don’t try entering lower-case messages.

### **7.3 Memory**

This page is only present on JSW128 and JSW64 versions. It shows the memory being used for various game aspects such as sprites and guardian tables. Click on a cell (or tab to it and press SPACE) to change its meaning; note that to allocate a guardian table you will need four consecutive memory pages (eight for upgraded Softricks games).

### **7.4 Sprites**

The sprite editor screen is divided into two parts. The bottom half contains a list of sprites; select these using the left and right cursor keys (use PageUp/PageDown to move faster) and click or press ENTER on the chosen sprite. You can also scroll the list with the mouse, by clicking on the arrows in the bottom corners.

Once you have chosen a sprite, switch to the editor using TAB and use the four cursor keys to move the cursor, SPACE to toggle the cell colour (or use the mouse).

Press CTRL+Z or CTRL+U to undo changes to the current sprite. Press it again to undo the undo.

#### **7.4.1 Importing / Exporting**

Right-click on the bitmap editor and you will be given the option to perform bitmap operations (reflect, rotate, invert and so on), or to export or import.

When exporting, you are asked to specify the first and last sprite to be exported, and the file format to use (+3DOS or PNG). The +3DOS format saves as a chunk of Spectrum memory; PNG as an image with all the sprites side by side.

When importing, JSWED will detect the file format in use (+3DOS, PNG, BMP or JPG). It will then ask which is the first sprite to be replaced; the last sprite to replace is then calculated automatically. If you click 'Preview' at this point, you will be able to see the sprites in the file being imported.

Imported images should be 16 pixels high, with the sprites side by side in black on white.

### **7.5 The Room Editor**

The room editor is controlled a bit differently from the other screens. Instead of using TAB to move between controls and SPACE to select, the cursor keys are used to move the blue/white cursor in the room display. The keyboard is used to choose what to do under that cursor. The full list of keys is:

Key	Meaning
Cursor keys	Move room cursor
CTRL+Cursor keys	Go to another room, following the JSW exits
< >	Go to room numbered 1 higher or 1 lower
J	Go to a room chosen from a menu
X	Set up the room's exits (7.5.2)
B	Clipboard (7.5.3)
O	Room options (7.5.5)
G	Guardians. (7.5.6)
ALT+C	Conveyor options (7.5.7)
1-8	Choose guardian to position using the cursor
ALT+1-8	Change guardian start frame (7.5.8)
A/W/E/F	Draw Air/Water/Earth/Fire cell under the cursor when SPACE is pressed
R	Ramp editor
C	Conveyor editor
I	Toggle item under the cursor when SPACE is pressed (Use SHIFT+SPACE to place multiple items in the same place)
CTRL+P	Go to the previous room you were looking at (if any).
CTRL+J	Go to another room by entering its number.
CTRL+L	Refresh this room, as if you had gone to another and come back.
^Z or ^U	Undo changes to current room.
F11 or keypad +	Increase animation speed
F12 or keypad -	Decrease animation speed

Animation speed ranges from 1 (full speed) to 16 ( $\frac{1}{16}$  speed). After 16 comes 0, in which guardians and conveyors are not animated at all.

### 7.5.1 Moving between rooms

The panel in the bottom left-hand corner contains controls used to move between rooms. When you go to another room the current room is saved.

The four arrows (corresponding to CTRL+cursor keys) move you to neighbouring rooms in the JSW map. The < > signs use the room number sequence, and "Enter no." (press J) lets you choose the new room from a menu. If you know the number of the room you want to go to, you can press Ctrl+J and type the number.

### 7.5.2 Editing exits

When you ask to edit exits, a menu appears showing the current exit assignments. Press ESC to leave them as they are, or choose the exit to set. If you chose an exit, a list of rooms then appears for you to choose the new destination.

### 7.5.3 Clipboard

The clipboard allows you the usual copy/cut/paste options, plus the ability to clear the room to any of the four cell classes and to mirror the structure of the room horizontally or vertically. Note that mirroring does not alter ramps, conveyors or exit assignments.

#### 7.5.4 Import and Export

The Export option allows you to save the layout of one or more rooms as an XML file. When you select it, you will be asked to choose the range of rooms to export - click on 'From' and 'To' to select the appropriate rooms. Once you have selected the correct range, click OK and enter the file name to save.

Importing is the reverse of exporting - choose a file to load, and then choose the first room to replace. The rooms stored in the file will then replace that room and as many subsequent ones as necessary.

Import and Export are new functions and it's as well to be careful with them. You may not always get the expected results. In particular, if you export from one game engine (say, JSW64) and then import into a different game engine (say, Manic Miner) JSWED will do its best, but the result may require manual tweaking.

#### 7.5.5 Room options

The room options menu gives you two options:

- General: Allows you to set the border colour, room title and (in 128k games) Willy's sprite.
- Cells: Shows the bitmaps and colours for the seven cell classes, and allows you to edit them. Right-click a bitmap for reflect/rotate/move options, as in the sprite editor. Click the "<" or ">" to choose a different cell class to edit.

#### 7.5.6 Guardians

If you don't have a guardian currently selected, you are given the option of adding another one (assuming, that is, that you don't have eight in the room already). If you do have a guardian selected, you are also able to delete it from the room or to edit it (section 8).

#### 7.5.7 Conveyor options

Use the conveyor options menu to set four special conveyor types:

- Ramp left / Ramp right: Turn the ramp into a conveyor ramp (escalator), with no separate conveyor.
- Sticky / off: The conveyor is stationary.

#### 7.5.8 Guardian / guardian start frame

To choose a guardian to work with, click its number in the second menu from the right. The guardian's eight frames will appear at the bottom of the screen, and an outline of its path will be visible at the top. To position a guardian, use the cursor or mouse.

To change the guardian's start frame, press ALT+numbers, or click on the animations within the guardian list.



### 7.5.9 Cell editors

Rooms in JSW are composed of the four elements known to Greek philosophy - Air, Earth, Water and Fire. They can also have at most one ramp and one conveyor.

Edit the layout of the room by pressing A,E,W,F,R or C, moving the cursor to where you want, and pressing SPACE or clicking the mouse. You can drag the mouse to draw shapes.

The ramp and conveyor editors need two presses of SPACE (or two clicks), one at each end. For a ramp, start at the bottom; as you press left/right, the ramp will grow. For a conveyor, start at the “from” end and move the cursor towards the “to” end. If you press SPACE twice without moving in between, the ramp/conveyor will be deleted; to create a ramp/conveyor one character long, grow it to two characters and then shrink it.

Items are superimposed on the other cell types. Select the item type by pressing “I”. To add an item, click where one is not; to remove it, click where one is.

### 7.6 Teleport editor

The teleport overlay is supported for JSW48, JSW128 and JSW64 games. By default teleportation is not supported; to get teleporter support you must load the overlay. In JSW48 you then have to sacrifice a room to the teleporter code; in JSW128/JSW64 it is allocated from the memory map.

Once the teleporter overlay is loaded, you can click on the room to create a teleporter (which will appear as an orange “T”). You will then be asked to choose its destination.

If you click on an existing teleporter, you can delete it or change its destination.

### 7.7 Start position editor

This editor allows you to position three special sprites - Willy’s initial position, Maria and the toilet. Click the sprites (or press W / M / T) and then position them in the room you want. Note that in Henry’s Hoard you can only position Willy/Henry. In a Geoff Mode game you can’t have Maria and the toilet in the same room; in a normal game, you can.

If you have the “No infinite death” patch turned on, there will be a fourth special sprite - the safe restart position (a green version of Willy). This must be set to a valid value for each room in the game; if Willy dies in a room where this position has not been set, the game will crash!

### 7.8 Font editor

The font editor works exactly like the sprite editor - click on a letter, and edit its bitmap.

#### 7.8.1 Importing / Exporting

Right-click on the font editor and you will be given the option to perform bitmap operations (reflect, rotate, invert and so on), or to export or import.

When exporting, you are asked to specify the first and last character to be exported, and the file format to use (+3DOS, PNG or PSF). The +3DOS format saves as a chunk of Spectrum memory; PNG as an image with all the characters side by side; PSF in the PC Screen Font format (version 2).

When importing, JSWED will detect the file format in use (+3DOS, PSF, PNG, BMP or JPG). It will then ask which is the first character to be replaced; the last character to replace is then calculated automatically. If you click 'Preview' at this point, you will be able to see the characters in the file being imported.

Imported images should be 8 pixels high, with the characters side by side in black on white.

## **7.9 Screen editor**

The screen editor works like the room editor - use cursor keys or the mouse to move the cursor within the image, and press SPACE or click the mouse to draw. Depending on which screen you are editing, you may be able to edit just the pixels, just the attributes, or both. Press the "M" key or click the "Mode" icon to switch between pixel and attribute modes.

### **7.9.1 The Zoom window**

To the left of the main screen window is a zoom window. This shows a scaled-up version of the character cell the cursor is currently in. Clicks in this window have no effect; it's for viewing only.

### **7.9.2 Editing pixels**

Click on a pixel to toggle it between ink and paper; you can drag the mouse to set several pixels to the same colour. The keyboard control may be more useful here because moving the mouse by one pixel at a time is quite fiddly.

### **7.9.3 Editing attributes**

The current attribute that will be used is set using I(nk), P(aper), B(right), F(lash). Once you have chosen the attribute combination you want, click on a character square to set its colours.

### **7.9.4 Grids**

Press "G" or click on the "Grid" icon to switch between the two available grids. They are the Pixel grid (attributes show as normal, but a pattern of squares appears instead of the pixels) and the Attribute grid (pixels show as normal, but the attributes are a fixed pattern).

### **7.9.5 Next / Previous**

Press "<" / ">" or click the same symbols onscreen to move from one picture to the next. The following pictures exist in the different games:

Game	Screen no.	Screen name	Comments
JSW 48k	1	Bottom third	256x64, attributes only
JSW 128k	1	Bottom third	256x64, attributes only
	2	Title screen	256x128, bitmap and attributes
	3	Title screen animation	attributes only
	4	Title screen animation	attributes only
Henry's Hoard	1	Title screen 2 of 3	Only present in 1985 version
	2	Victory screen	256x64, bitmap only
JSW64 games	1	Bottom third	256x64, attributes only
	2	Air-remaining bar	256x8, attributes only
	3	Title screen	256x128, bitmap and attributes
	4	Title screen animation	attributes only
	5	Title screen animation	attributes only
	6	'Final Barrier' screen 1	256x64, bitmap and attributes
	7	'Final Barrier' screen 2	256x64, bitmap and attributes
Manic Miner	1	Bottom third	256x64, attributes only
	2	Title screen, top third	256x64, bitmap and attributes
	3	Title screen, middle third	256x64, bitmap and attributes

### 7.9.6 Undo

Press Control-Z or Control-U to undo everything you have done to a picture since the last “<” or “>”. Press Control-Z again to undo the undo.

### 7.9.7 Import / Export

Right-click on the screen being edited and you are given the option to import or export the current image. Images can be exported as PNG or Spectrum SCREEN\$; they can be imported as SCREEN\$, PNG, BMP or JPG.

Trying to import an image for a screen where you can only edit the attributes (for example, the “bottom third” graphic) will load only the imported screen’s attributes.

## 7.10 The Title Screen Editor

The title screen editor works rather like the screen editor in attribute mode. There’s no grid, and no option to move to another screen; you just get the colour chooser. The difference is that when you draw in particular colours, you will see sections of triangle appearing on the screen.

All colour combinations except ten will draw as “/” slopes. Of the ten combinations, black on black is drawn as solid black. The other nine are chosen by right-clicking and selecting “Colours”. You will see a screen with three sections, allowing you to choose nine colours:

**Colours for blank cells:** The four colours which will be drawn with no triangle segments at all. In the original JSW, these are flashing red/magenta (for the title wording) and solid blue, green and cyan.

**Colours for \ sloping cells:** The four colours which will force a triangle to be drawn which slopes in the opposite direction. In the original JSW, these are green on cyan, black on blue, blue on cyan and cyan on black.

**Substitute colour:** The first “sloping cell” colour is replaced with this colour. This is to get round the tricky “<” bit of the triangle, in which sections have to be the same colour yet slope different ways. In the original JSW, this is cyan on green - ie, the exact inverse of the first slope colour.

It is possible to edit the 'slope' graphics by right-clicking and selecting “Tiles”. Images can be imported and exported as usual.

## 7.11 The Music editor

### 7.11.1 Using the original tune player

The music editor is present only in 128k games (JSW128 and JSW64). It allows the three tunes to be played, exported or imported. Under the description of each tune are three buttons:

**Test** runs the game in an emulator, just as pressing F5 does; except that the title screen tune is changed to be the tune you want to test.

**Export** saves the selected tune. Two formats are supported: +3DOS (this just saves the tune as a block of memory) and Assembly (a text format, described fully in Appendix C).

**Import** loads a new tune over the currently selected one. Both formats are supported. If the new tune won't fit in the space available, the import will fail. JSWED will automatically fix-up the file as it loads, so you can load a tune into a slot other than the one it was saved from.

### 7.11.2 Using the SoundTracker player

If you click the button at the bottom of the screen, the original music player is replaced by the one from SoundTracker. The following aspects of the screen change:

- The Export and Import options now support three formats – +3DOS, TAP and RAW.
- By default, there is no memory available for the cheat-mode tune, and 504 bytes for the in-game tune. If you want to load two tunes and/or a longer tune, you will need to allocate one or more areas of memory for them using the Memory page.
- The title screen tune is still limited to 2774 bytes; this can't be changed.
- When loading a tune, JSWED will try to find the most suitable place for it to fit. Note that loading an in-game tune will overwrite a cheat-mode tune, if there is nowhere else it will go.
- To prepare a song for JSWED: Load it into the Soundtracker compiler, select 'compile song', and save the result back to a new .tap file, without going through the 'Merge play routine' option.

## 7.12 The Hex editor

The Hex editor is pretty much the same in all games, though the keys to access the 128k memory banks only apply in 128k games.

*Be warned that it's very easy to damage your game beyond repair using the Hex editor. Only use it if you are sure you know what you are doing!*

When using the Hex editor, remember the following:

1. Any outstanding changes you make are shown in magenta. Until you click "Write" (or press Control-W), they will not be applied to the game.
2. You can click "Undo" (or press Control-Z) to remove all outstanding changes.
3. Switching to another screen and back also removes all outstanding changes.

The Hex editor displays three columns of data. The left column gives addresses; the middle column gives the hex values of the bytes at those addresses; and the right column gives ASCII equivalents (if any). Values that can't be edited (ie, those bytes which would be occupied by the Spectrum ROM) are shown in blue. The bottom bar shows the address that the cursor is currently on.

The following keys can be used:

**Cursors:** Move left/right/up/down.

**Page Up / Page Down:** Move up/down a screen (34 lines) at a time.

**Home / End:** Move to the start/end of the current line.

**Shift+Home / Shift+End:** Move to addresses #0000 and #FFFF respectively.

**Tab:** Switches between the 'hex' and 'text' columns.

**Control-A:** Enter address. This moves the cursor to the specified address.

**Control-F:** Find bytes. Enter a sequence of Hex bytes to find. The search will start at the byte after the current one.

**Control-G:** Find again. Repeats the last search, starting at the byte after the current one. You can also use F3 to get the same effect.

**Control-W:** Write data. All outstanding changes (shown in magenta) are applied to the game.

**Control-Z:** Undo. All outstanding changes are reversed.

**Control-0 to Control-7:** Select memory bank at #C000. For 128k games only. Note that banks 5 and 2 are always present at #4000 and #8000 respectively.

**Other keys:** If the cursor is in the 'hex' column, 0-9 and A-F allow the Hex values to be modified. If it's in the 'text' column, then any key producing an ASCII value will insert that value at the current location. This includes ALT+P for Copyright.

## 8 The Guardian editor

### 8.1 On Guardian Classes

In Jet Set Willy games (other than JSW64 and JSW2), guardians in a room are drawn from a table of GUARDIAN CLASSES, with up to 127 entries (Softricks games have 255 entries in their guardian table; 128k games can have more than one guardian table). A particular reference to a guardian in a room is referred to as a GUARDIAN INSTANCE.

The guardian editor operates on the classes in this table. If you change a guardian class that's present in more than one room, you'll find that every guardian instance using that class will change its behaviour. For example, in the original JSW, changing the flag in "On top of the house" will also affect the moon-faced guardian in "Under the Drive".

### 8.2 Editor overview

When a guardian class is being edited, the top half of the screen shows a preview of the result, while the bottom half contains controls. The controls shown vary for different guardian types, but include:

Control	Types	Description
Type	All	Change guardian type
Sprite	All except arrow & rope	Change sprite page
Y	Horizontal, diagonal	Vertical position
Y	Vertical	Initial vertical position
right of Y	Vertical, diagonal	Vertical step
L/R?	All except vertical	Initial direction - left or right
Colour & bright	All except arrow & rope	Guardian colour
Animation	All except arrow & rope	Animation type
Bounds	All except arrow & rope	Left/right or up/down limits of travel
Fast	Vertical	Uses fast animation
Fast	Horizontal, diagonal (JSW128)	Moves at double speed
Slow	Horizontal, diagonal (JSW128)	Moves at half speed
Stopped	Rope (JSW128)	If ticked, the rope does not move.
Rope pos	Rope	Initial position of rope.
Rope len.	Rope	Length of the rope
Rope swing	Rope	How far the rope swings out
Bitmap	Arrow	Top / bottom of the arrow.
Start pos	Arrow	Initial position of the arrow.
!	All	Edit the guardian as hex (section 8.7)
Wrap	Geoff Mode	Is the guardian one-way?
H/D?	Geoff Mode	Horizontal or diagonal?

### 8.3 Guardian types

In JSW48, there are four guardian types - horizontal, vertical, arrow and rope. In JSW128 there are ten; the four mentioned earlier, two diagonal types, and flashing

versions of all except the arrow and the rope. In general JSWED tries to preserve guardian data when changing from one type to another, but this is not always possible; so changing from a vertical guardian to a rope and back won't give you the guardian you had before.

In Geoff Mode, there are four guardian types (horizontal/diagonal, vertical, arrow and rope). Use the "H/D" setting to switch between horizontal and diagonal.

## 8.4 Editing conventional guardians

There are no real pitfalls with editing conventional guardians. If for some reason the preview of the guardian's bounding rectangle gets out of step with the values you are entering, it is possible to force a recalculation. To do this, click [!] for the hex editor, and then click [OK] to dismiss it - don't change any of the fields!

## 8.5 Rope editing

The initial position of the rope is positive if the rope starts in the right-hand half of the screen, negative if the rope starts in the left-hand half, and zero if the rope starts dead centre. It should be less than the rope "swing" value.

The rope length can be 1-32; 32 is the usual value.

Unless you are using the (new) "adjacent ropes" patch, ropes must be stored last in a room, *or* be followed by an arrow, or by an instance of a blank guardian such as Guardian 0. This is because a bug (or feature?) in the game engine causes a rope to write bytes into the following guardian's data.

## 8.6 Arrow editing

Arrows have 256 possible positions. Numbers 0-31 are onscreen; the others are off-screen. This is set up by the "start pos" field. The small bitmap editor above it is used to set the pattern used on the top and bottom of the arrow.

## 8.7 Hex editing

Press the [!] button to edit a guardian as hexadecimal bytes. It is possible to create guardians that crash JSW by doing this, so take care.

## 8.8 Geoff Mode bounds

In Geoff Mode, the guardian bounds are, respectively, an "initial frame count" and a "maximum frame count", rather than "left limit" and "right limit".

# 9 Good Advice

The following notes are based on the documentation of Paul Rhodes's JSW editor, with additional suggestions by Andrew Broad. Note that if you have the 'Black Willy' patch applied, any reference to 'white' below should be read as 'black'.

1. Don't put anything (except items) in the path of a guardian. Guardian collisions are pixel based; you have to be sure that none of the guardian's pixels touches

any pixels already on the screen. If a guardian has white INK, it will collect any items it touches.

2. Don't have arrows passing through anything with white INK.
3. If you have white INK in air cells, any items in the room on background cells will be automatically collected on entry to the room (see swimming pool).
4. Be careful if you have two cell classes with the same attributes. In particular, don't make other cells the same colour as Fire, and don't make Fire the same as Air but with white ink.
5. Having the same attributes for ramp and conveyor creates an escalator (see Chapel). Other combinations that may be useful are giving Water or Earth the same attributes as the ramp or conveyor.
6. Avoid having Fire cells at the top of the screen above a gap at the bottom (see right-hand half of Under the Roof).
7. Do not put anything in the path of a rope (try it and see!), and do not allow an arrow to pass through a rope (especially not a white rope). Note that putting the arrow *after* the rope in the room's guardian instance list is OK - but putting it *before* the rope results in a collision.
8. Never allow guardians to collide. Their paths can overlap if you make sure that only one guardian enters the overlapping space at any one time.
9. Items on top of other cell types (floors, walls, etc.) cannot be collected by Willy. They can be collected by being touched with white arrows or white guardians. (In Manic Miner and JSW64 games, there's an exception: if the item is on top of crumbly floor, then once the floor has crumbled away, the item can be collected).
10. It is possible for a JSW game to contain infinite-death scenarios, where Willy loses all his lives and you can do nothing about it. For example, jumping off the end of the upper platform in The Front Door does this. Try to stop these occurring unless the player deliberately does something suicidal (as in the above example).
11. The use of non-obvious features of the game engine (known by some as 'Quirky Features' and by others as 'bugs') is a controversial topic. There are those who will be disappointed if your game doesn't contain any, and there are also those whose reaction to their presence is to delete the game with a comment of "I was supposed to know I should do *that*?". Various solutions have been proposed, including the construction of 'easy' and 'hard' versions of games, and the use of these features as bonuses for advanced players. Either way, it's best if you are familiar with the various quirks of the game engine, because otherwise you may create unintentional bugs in your games.
12. Playtest every room thoroughly, especially all possible exits from the room. If you make even a trivial change to a room, never forget to playtest it again.
13. Do a complete playtest to make sure the final revision of the game is fully completable (to the point where Willy sticks his head down the toilet) before you release it.



14. Save your work regularly by pressing F2.
15. Hard-drives fail, floppy disks get corrupted, CDs become unreadable, buildings burn down and files uploaded to the Internet get removed, but if you regularly save your files in three completely different places, then the probability of losing your game is very slim.

## Part II

# Manic Miner

## 10 Editing Games

The game editor behaves in the same way as the Jet Set Willy one, but with fewer pages.

If you're editing Mihai Novitchi's 128k version of Manic Miner, note that the game stores its rooms (and screens) in a compressed format. Whenever you save or playtest, JSWED will generate the compressed data (you'll see a progress bar); if the result won't fit, then JSWED will tell you but your changes *will not be saved*. Such are the consequences of a compressed room format.

### 10.1 Game

This page acts like the JSW 'Game' page, but there is only one patch - "Specials by room". This allows each room to control whether it has vertical guardians, skylabs, solar power, Kong or Eugene (instead of these options being hard-coded).

### 10.2 Message

This page contains messages printed by the game - the title screen message, the "Game Over" message, and the "score" message. You can edit them as you like; the program will not allow you to enter more text than will fit.

### 10.3 Sprites

The sprite editor screen is divided into two parts. The bottom half contains a list of sprites; select these using the left and right cursor keys (use PageUp/PageDown to move faster) and click or press ENTER on the chosen sprite. You can also scroll the list with the mouse, by clicking on the arrows in the bottom corners.

Once you have chosen a sprite, switch to the editor using TAB and use the four cursor keys to move the cursor, SPACE to toggle the cell colour (or use the mouse). The context menu for this screen allows you to reflect, rotate, move, invert and clear the bitmap.

Press CTRL+Z or CTRL+U to undo changes to the current sprite. Press it again to undo the undo.

Note that this only covers Willy and guardian sprites, not portals, Eugene or other special graphics.

## 10.4 The Room Editor

The room editor is similar to the JSW version, but the following functions are different:

### 10.4.1 Moving between rooms

Since Manic Miner has no exits at the edges of rooms, there are no arrows to move between rooms. Likewise, there is no option to set up the exits.

### 10.4.2 Guardians

The guardians are set up on a separate screen - see section 10.5.

### 10.4.3 Editing cells

Manic Miner supports more cell classes than Jet Set Willy's 4-element system. It allows:

- Air
- Water
- Water 2: If the room has a Kong beast, this is used for switches. Otherwise it's another type of Water cell.
- Crumbly - disintegrates when Willy walks on it.
- Earth
- Convey - animated conveyor. You can only have one animated conveyor in a room.
- Conv 2 - non-animated conveyor. You can have as many cells of this type as you like.
- Fire
- Fire 2
- ? - user-defined attribute. Not normally useful.

### 10.4.4 Room options

Selecting "Room options" takes you straight to the room property screen. You can set the border colour and the amount of air remaining.

- General: Allows you to set the border colour, room title and air remaining.
- Shapes: Shows the bitmaps and colours for the eight cell types, and allows you to edit them. Right-click a bitmap for reflect/rotate/move options, as in the sprite editor.

### 10.4.5 Conveyor options

Use the conveyor options menu to set the conveyor to be Off or Sticky. It also allows you to move the conveyor animation up and down.

#### **10.4.6 Item options**

Use this screen to design the item graphic, and to set the colours of the up to five items in the room. The item graphic will be the same for all of them.

#### **10.5 Guardians page**

The guardians page is used to position guardians within a room. Select the guardian you want from the list at the bottom of the screen, and then you can set its start position using the mouse (or cursor keys plus space).

Right-click to add a new guardian, or to edit or delete the currently selected one.

#### **10.6 Portal editor**

This editor allows you to edit Willy's start position in each room, and the portal which takes you to the next room. If the "specials by room" patch has been applied, then you can also control whether the room has the various special features.

Keypresses you can use here are:

- B** Set whether portal is bright.
- D** Set Willy's starting direction (facing left / facing right).
- E** Toggle whether Eugene operates in this room.
- F** Set whether portal flashes. If the portal is flashing Willy can go to the next level whether or not he has collected all the items in the room.
- G** Edit the portal graphic and the Eugene graphic.
- I** Set the ink colour for the portal.
- K** Toggle whether there is a Kong Beast in this room.
- N** Set Willy's start frame.
- P** Set the paper colour for the portal.
- S** Toggle whether there is solar power in this room.
- V** Toggle whether there are vertical guardians in this room.
- W** Set Willy's start position.
- X** Set the position of the portal.
- Y** Toggle whether there are skylabs in this room (skylabs aren't quite the same as vertical guardians).

## 10.7 Screen editor

The screen editor has three screens to edit. The first one only allows attributes to be changed; the other two can have both bitmaps and attributes changed.

Game	Screen no.	Screen name	Comments
Manic Miner	1	Bottom third	256x64, attributes only
	2	Title screen, top	256x64, bitmap and attributes
	3	Title screen, middle	256x64, bitmap and attributes

## 11 The Guardian editor

### 11.1 No Guardian Classes

In Manic Miner, guardians are set up by room; changing a guardian in one room will not affect that guardian in any other room.

Note that the guardian editor does not allow you to set a guardian's start position - you do that through the guardian page from which the editor was launched.

### 11.2 Editor overview

When a guardian is being edited, the top half of the screen shows a preview of the result, while the bottom half contains controls. The controls shown vary for different guardian types, but include:

Control	Types	Description
Start frame	All	Initial frame
Speed	Vertical, Skylab	Initial speed and direction
L/R?	Horizontal	Initial direction - left or right
U/D?	Vertical, Skylab	Initial direction - up or down
Flash	Vertical, Skylab	Does guardian flash?
Slow	Horizontal	Does guardian move at half speed?
Colours & bright	All	Guardian colour
Bounds	All	Left/right or up/down limits of travel
!	All	Edit the guardian as hex (section 8.7)

### 11.3 Guardian types

Manic Miner allows up to four horizontal guardians, and up to four vertical guardians/skylabs. You can't change this ratio and have (eg) five horizontal and three vertical. It's not a good idea to put vertical guardians (or skylabs) in rooms 0, 1 or 2, because they overwrite the "Swordfish", "Barrel" and "Foot" sprites.

## Part III

# JSW64 games

## 12 JSW64 Overview

JSW64 is a redesign of the Jet Set Willy engine intended to remove a lot of the limits on what you can have in rooms. For example, in standard JSW games, a room can have at most one ramp and one conveyor. JSW64 makes “ramp” and “conveyor” into standard cell classes, so you can have as many of either as you like (though, as in Manic Miner, only one conveyor will be animated). To accommodate the extra data, the JSW64 room definition is larger than the original JSW one, and there are fewer rooms (128 or 64, compared to the 256 in JSW128).

Another difference is that there isn’t just one JSW64 engine - there are six. They all have similar feature sets, but make different tradeoffs in the way the rooms are stored. One variant can’t be changed to another after the fact (except by exporting everything from one variant and importing back into another), so you have to get it right at the time you first upgrade a 48k game to JSW64.

Most of the pages in the JSW64 editor are the same as their JSW48 / JSW128 counterparts. This part of the documentation describes the ones that are different.

### 12.1 Variants

The six variants of JSW64 are:

Variant	Rooms	Guardians	Cell classes per room	Global cell types	Cell types set	Free
V	128	13	8	0	by room	64
W	128	8	13	3	globally	0
X	64	13	8	0	by room	576
Y	64	8	13	3	globally	512
Z	64	8	13**	3	globally	256
L	64	4	16	0	globally	512

The “free” figure is the amount of unused space in the room definition. It is possible to put guardian definitions, patch vector code or sprites in this space.

\*\* There are 13 full cell classes, but any of the other 240 colour combinations can be used; they behave like differently-coloured water.

## 13 Memory Map

JSW64 does not have guardian tables. Otherwise this screen is identical to the JSW128 one.

## 14 Rooms

As with the Manic Miner editor, the guardians have been moved to a page of their own. Other differences are:

### 14.1 Cells

In Manic Miner, JSW48 and JSW128, the meaning of cell classes is fixed - the first cell is air, the second is water, and so on. In JSW64, this is controlled by the game

itself. In the V and X variants, the cell types are set up by room; so one room could have three types of 'fire' cell and no conveyors, while another could have one fire, and two conveyors (one going each way). In the other four variants, the cell type is set up globally; your choice of cell behaviour affects all rooms.

Because of this definable behaviour, the room editor doesn't use shortcuts like A=Air, W=Water, E=Earth and so on. It uses 0,1,2,3 and so on to choose the first,second,third... pattern. Ramps and conveyors are just normal cell types.

The "Cells" screen for JSW64 looks like the one for other games, but also has a "Type" control which can be used to set the behaviour of each cell. Remember that in variants other than V and X, changing a cell type affects every room in the game.

#### 14.1.1 Extra cell types

JSW64 has the following cell types:

**Air/Water/Earth/Fire** behave like their JSW48 / JSW128 counterparts.

**Ramps and conveyors** are now normal cells rather than one-offs. You can therefore have as many of either as you want in a room.

**Crumbling blocks** behave as in Manic Miner.

**Trampolines** are an addition from the Dragon 32 version of JSW. When Willy stands on one, he is forced to jump.

**Traps** are based on Andrew Broad's "harmless fire" patch. If any part of Willy is above a trap, he falls. Traps should be used with caution; letting Willy jump up onto a trap from below can have unexpected consequences.

The editor also lets you position three other "cell" types:

**Item** lets you position items, exactly as in JSW48 / JSW128.

**Solar** is used to set the start point of the Solar Power beam (if the room has solar power). The start position of the beam is indicated by an orange sun. Note that the editor will only let you put the sun in the top half of the room.

**Custom** (Variant Z only): Place any combination of foreground and background attributes. These will be drawn with the pattern of cell 1 (cell 0 is usually used for air and therefore is unlikely to have a pattern). In the game, custom cells will behave as Water.

### 14.2 Room properties

The "Room properties" screen has a few more options than the JSW48 / JSW128 one. These are:

**Willy** Set Willy's colour.

**Solar** Set the colours for the Solar Power beam. If you don't want a room to have solar power, set these to black on black, with Bright and Flash unticked.

**Air** The amount of air in the room. 162 (the maximum value) means "air is not limited".

**Rigor Mortis** If this is ticked, guardians in the room will not move until all the items in the room are collected.

**No Kamikaze** If this is ticked, when Willy dies he loses any items he collected since entering the room. This is intended for compatibility with Manic Miner-style rooms, but a sadistic game designer may decide to use it in other circumstances.

**Bonus Room** If this is ticked, each item collected gives Willy an extra life.

**Escalator** Tick this to make all ramps in the room into escalators. There is a second tickbox to control whether you want them to be 'up' or 'down' escalators.

**Land softly** (aka "Fall any height"). If this is ticked, Willy cannot die through falling (in this room).

### 14.3 Room-specific graphics

JSW64 variants allow some space in the room definition to be used for 16x16 graphics (eg, for a portal). Variants X,Y,Z and [ can store a complete set of frames (eg, for a custom sprite for Willy).

To edit these sprites, choose "Sprites" from the room options menu. You can then edit the sprites in the room, in the same way that you can edit the systemwide sprites.

### 14.4 Conveyor animation

As in Manic Miner, the conveyor animation in JSW64 games is independent of the actual conveyor. Unlike when you are editing a Manic Miner game, drawing conveyor cells does not update the animation (or vice versa). You have to do the animation separately; press V and then draw the animation over the blocks. With this system it's quite possible to have a conveyor that goes in one direction and animates in the other!

### 14.5 Final Barrier

JSW64 contains two graphical buffers that can be used to draw particular levels (as the title screen is used in Manic Miner to draw the Final Barrier). To set a room to use one or other of these buffers (or with the top/bottom half of the title screen), select "Final Barrier" from the room options menu. A screen will then appear showing the room, with two lists. One selects the buffer to use for the top half of the room; the other selects the bottom.

## 15 Guardians

In JSW64, as in Manic Miner, guardians are set up individually in each room. The number of guardian slots available varies depending on the JSW variant you are using, and can be further customised by room.

The guardian choice screen shows a list of guardian slots; each one is shown as "Grd" if there is a guardian in it, or "-" if it is empty. Click on the screen to set the start position of the guardian; right-click to get the "Guardians" menu. The start frame is set in the guardian editor rather than using ALT+1 - ALT+8.

Unlike in JSW48 / JSW128, guardians in JSW64 are not shared between rooms.

## 15.1 Guardian editor

The guardian editor behaves just like the JSW128 one, with these exceptions:

- There are two “Animation” fields. The left-hand one is the start frame of the guardian.
- The arrow editor allows you to set the arrow colour, and the bitmap pattern used by the middle of the arrow.
- Extra guardian types are supported: Skylabs, Angry Eugenes, Triggers, Switches and Vanishing Walls.

### 15.1.1 Skylabs

A Skylab behaves like a vertical guardian, except that it does not animate until it reaches the end of its travel. At this point, it starts again at the beginning, 8 columns to the right.

The Skylabs are based on those in Manic Miner. However, these Skylabs can also go up. If so, the first bound is the bottom bound and the second bound is the top (the other way round from all other vertical guardians).

### 15.1.2 Angry Eugenes

Angry Eugene behaves like a vertical guardian until his Y-coordinate equals either of its bounds (it is therefore usual to set both bounds to the same value). Once Eugene has reached his destination, he stops.

Be warned: If the bound value cannot be reached exactly (eg: Y-coordinate is 4, bounds are both 50, speed is 4) then Eugene will just keep going, probably colliding with something undesirable.

There are three Eugene types - normal, multicolour, and multicolour 2. The second multicolour option exactly duplicates the way Manic Miner does a multicoloured Eugene; it will only work in a room with a limited air supply.

### 15.1.3 Triggers

A trigger is used to modify the behaviour of the subsequent guardian in the table. When you edit a trigger, you will see two buttons:

**Source** is the event that causes the trigger to activate. It can be one of:

- Willy collecting the last item in the room.
- An angry Eugene stopping.
- A switch being turned on.
- A wall vanishing.

**Effect** brings up another guardian editor, in which you can decide exactly what will happen when the trigger takes effect. The trigger can only change bytes 0 and 4-7 of a guardian; so you can't change position, colour or animation settings.

If the trigger is last in the guardian list, then a bright red warning will be displayed on the guardian editor. If you save a game with a trigger at the end of the list, then the game will probably crash when the trigger is activated.



#### 15.1.4 Switches

A switch is a guardian, but it's harmless for Willy to touch it. At any time, it is turned off or on; if Willy touches it, he may change its state.

By itself, a switch just sits there. But other guardians can monitor the state of a switch and take action based on its setting.

The options when editing a switch are:

**Sprite** The switch graphic. Unlike the other guardians, the switch is only 8 pixels high (it is still 16 pixels wide). The top half of its sprite graphic is the "off" image; the bottom half is the "on" image.

**Colour/Bright** as normal.

**Animation** chooses which sprite in the page will draw the switch.

**On/Off** sets the initial state of the switch

**Stays off** If this is ticked, then once the switch is turned off it can't be turned on.

**Stays on** If this is ticked, then once the switch is turned on it can't be turned off.

If you leave both the "Stays" options unticked, then while Willy is touching the switch it will rapidly alternate between states. If both are ticked the switch becomes a harmless ornament.

#### 15.1.5 Vanishing Walls

A vanishing wall section, when activated, removes a section of the room - one character wide, and an arbitrary height. Like the trigger, it only activates after another event has taken place, and the same "Source" button is used to select this. The other control on the screen - height - selects how many wall pixels will be removed.

It is possible to use a vanishing wall as a delay system, by putting it on top of air cells. It then has no visible effect, but still completes a fixed time after it was started.

#### 15.1.6 Stoppers

Guardians which appear in the list after a stopper do not appear in the game. The intention is to have a trigger followed by a stopper; when the trigger is activated, the stopper is changed to a blank guardian - at which point the guardians after the stopper are drawn.

There are two types of stopper. The first one inhibits movement and drawing, so that the extra guards aren't visible. The second one only inhibits movement - so you can have a 'rigor mortis'-like effect on only some guardians in a room.

### 15.2 Guardian table

In JSW64, a room's guardian list can have a variable number of slots. To amend this (either to increase the number of guardians in the room, or to decrease it and use the memory for something else), right-click in the guardian choice screen and select "Table". You will then see a memory map of the room, coloured as follows:

**Red** Guardian table

**Green** Free space

**Blue** Used for other purposes

**Cyan** Used for sprites

There is also a +/- control which adjusts the number of slots in the guardian table. As you alter this, the memory map will be updated to match.

If you change the size or position of a room's guardian table, cross-references in triggers and opening walls will become incorrect and must be set up again.

## 16 Teleport editor

This works exactly like the JSW48 / JSW128 teleport editor, with one extension. From the right-click menu, you can choose to have a visual effect when Willy enters a teleporter:

**No special effect** behaves like JSW48 and JSW128; teleporters act as a jump-cut. This is the proper setting to use if you are using teleporters to simulate switches.

**Screen flashes** makes the screen fade to dark blue (as it does when you enter a portal in Manic Miner).

**Air runs down** makes the air remaining decrease to zero. This has no effect if the room doesn't have a limited air supply.

**Both the above** does both, one after the other - like a true portal.

## 17 Start position / Portal editor

This editor is similar to the JSW48 / JSW128 one, but it also allows you to add a portal to each room. Portals work the same way as in Manic Miner; they're inactive until Willy has collected all the items in the room, and then he can step into one and arrive anywhere else in the game. Here's a feature comparison:

JSW48/128 teleporters	Portals
As many as you like per room	0 or 1 per room
Invisible	Visible
Always active	Active once items collected, or when flashing
Optional extension	Part of game engine

Keypresses you can use here are:

**B** Set whether portal is bright.

**D** Set the portal destination. You will first be asked for the room; then you can place Willy's arrival position within it.

**F** Set whether portal flashes. If the portal is flashing Willy can go to the next level whether or not he has collected all the items in this room.

**G** Choose the sprite to use for the portal.

**I** Set the ink colour for the portal.

**M** Set the position of Maria.

**P** Set the paper colour for the portal.

**T** Set the position of the toilet.

**W** Set Willy's start position.

**X** Set the position of the portal.

To remove a portal from a room, press G and select "No portal". If you're adding a portal, it's advisable to set the colour before choosing the sprite.

### 17.1 Portal special effect

As with teleporters, it's possible to set the special effect that occurs when you enter a portal. To choose the effect, right-click and select "Portal effect". It can be one of:

**No effect** Willy is instantly transported to the portal destination.

**Screen flashes** The screen fades to blue but the air (if limited) does not run down.

**Screen flash + air** The screen fades and the air runs down.

The portal effect is set by room, rather than systemwide.

## 18 JSW64 Good advice

- If Willy stands on a trampoline in a room with Superjump turned on, he will shoot straight up until he either:
  1. Arrives in the room above;
  2. Loses a life;
  3. Hits something solid.
- Make sure the Solar Power beam doesn't go off the screen, or the game engine will crash. Block it with Earth if necessary.
- Unlike in JSW48 and JSW128, you can't combine the behaviour of different cells by setting them to the same attribute.
- If you change Willy's colour to Magenta, Green, Cyan or Yellow, then multiple items in the same position will be reduced to one single item when Willy enters the room; the others will be auto-collected.

## Part IV

# Editing Jet Set Willy 2

## 19 Editing Games

The game editor behaves in a similar way to the JSW48 one.

Note that JSW2 stores its rooms in a compressed format. Whenever you save or playtest, JSWED will generate the compressed data (you'll see a progress bar); if the result won't fit, then JSWED will tell you; your changes *will not be saved*. You must then reduce the size of your game until it does save<sup>1</sup>.

On tape (and therefore .TAP), JSW2 is encrypted. This is why you can only edit snapshots from after the game was loaded.

## 19.1 Game

This page has two options on it - the number of rooms in the game, and the number of items Willy has to collect to win (unlike in other JSW games, he doesn't necessarily have to collect every item). The original JSW2 has 134 rooms, but it may be possible to squeeze in a few more.

## 19.2 Message

This page contains messages printed by the game - the title screen message, the "Game Over" message, the "score" message and so forth. You can edit them as you like; the program will not allow you to enter more text than will fit. The "Trip switch" message should end with "Off"; the game will replace the last two characters with "n" if the switch is turned on rather than off.

## 19.3 Sprites

The sprite editor screen is divided into two parts. The bottom half contains a list of sprites; select these using the left and right cursor keys (use PageUp/PageDown/Home/End to move faster) and click or press ENTER on the chosen sprite. You can also scroll the list with the mouse, by clicking on the arrows in the bottom corners.

Once you have chosen a sprite, switch to the editor using TAB and use the four cursor keys to move the cursor, SPACE to toggle the cell colour (or use the mouse). The context menu for this screen allows you to reflect, rotate, move, invert and clear the bitmap.

Press CTRL+Z or CTRL+U to undo changes to the current sprite. Press it again to undo the undo.

## 19.4 Cells

In JSW2, the graphics for cells (the basic building blocks of a room) are defined in a global list rather than each room having its own cell graphics. This editor allows you to edit the cell graphics, in a similar manner to the way that sprites are edited. Under the bitmap editor grid are two colour choosers to set the foreground and background colours of the cell, and a checkbox which allows the graphic to be marked as inverted. If a graphic is marked as inverted, the dark and light pixels will be swapped before the game is started for the first time. Don't ask me why JSW2 does this.

---

<sup>1</sup>In fact, JSWED's compressor appears to do a better job than the one that JSW2 was originally compressed with. If you load JSW2 into JSWED and then save it, you end up with free space that wasn't there before.

## 19.5 The Room Editor

The room editor is similar to the JSW48 version, but the following functions are different:

### 19.5.1 Guardians

JSW2 has only two types of guardian - conventional guardians, and arrows. The menu allows you to add either type, or to edit or delete existing guardians. There is no provision for changing the start frame on this screen; but you may set the start position of guardians both horizontally and vertically (rather than just horizontally, as in the original JSW48).

### 19.5.2 Editing cells

Rather than editing individual cell graphics, you have to choose the graphic for each cell type from the list of all possible cell graphics.

JSW2 supports 9 elements rather than the four in JSW48. These are:

- Air
- Water
- Earth
- Fire
- Ramps (both directions)
- Conveyors (both directions)
- Collectable items. You can have up to 16 items per room; there is no separate global limit.

### 19.5.3 Room options

Selecting “Room options” takes you straight to the room property screen. You can set the following values:

- The name of the room
- Whether the room has a rope.
- Whether the conveyor is animated. If the conveyor cells in a room are not all contiguous (ie: there are two or more separate conveyors) then you shouldn't tick this box.
- Whether to animate the third row of the conveyor graphic as well as the top row.
- The patch code for the room. There are 26 possible patch codes in JSW2:
  1. Central Cavern: If Willy has won, jump on the spot repeatedly.
  2. Lift 1: Draw lifts using the 1st pair of lift definitions.

3. Lift 2: Draw lifts using the 2nd pair of lift definitions.
4. Lift 3: Draw lifts using the 3rd pair of lift definitions.
5. Lift 4: Draw lifts using the 4th pair of lift definitions.
6. The Trouble With Tribbles: Moving floor segments.
7. Is not used in Spectrum JSW2, and appears to have no effect.
8. Rocket Room. When Willy reaches particular coordinates, the central section of the room takes off and Willy is transported to the room above.
9. The Bathroom. Draws the toilet. If Willy hits it and the game is won, then teleport him to room 133 (Central Cavern). Also, make Willy run to the right if the game is won.
10. Master Bedroom. Removes Maria if 150 or more items have been collected. If Willy is standing on a right-moving conveyor (ie, the bed) then start him moving to the right.
11. Beam Me Down Spotty: Enables teleporters to be used in this room.
12. Belfry: Draws ropes above vertical guardians.
13. Eggoids: Makes diagonal guardians reverse when they hit the top or bottom of the screen.
14. The Yacht: Deals with the Yacht sailing away.
15. Trip Switch: Handles what happens when Willy touches the Trip Switch. The left conveyor is used for the “off” switch, and the right conveyor for the “on” switch.
16. Rigor Mortis: If all items have been taken from room 75, sets the first two guardians in the room to have X steps of -1 and +1, and counter values of 28, respectively.
17. Crypt Switch. Adjusts the X step and counter of the first guardian in the room.
18. Foot Room. Puts a foot at the top of the room, which drops when all items have been taken from room 103.
19. Lift 5: Draw lifts using the 6th pair of lift definitions.
20. First Landing: Make ‘fire’ cells flash. Also, make Willy run to the right if the game is won.
21. Lift 6: Draw lifts using the 7th pair of lift definitions.
22. Deserted Isle: Handle all the complicated behaviours of that complicated room. Allows teleporters to be used.
23. Macaroni Ted: Make Willy run to the right if the game is won.
24. Dumb Waiter: Draw lifts using the 5th pair of lift definitions. Also, make Willy run to the right if the game is won.
25. Highway to Hell: Moving floor segments.
26. Is not used in Spectrum JSW2, and appears to have no effect.

## 19.6 Room Specials Editor

This screen allows you to edit Willy's start position, the layout of the Cartography room, and those features provided by patch vectors - lifts, teleporters and the toilet in the bathroom.

Features other than teleporters and cartography are handled like guardians, and can be edited in the same way as normal guardians in the room editor.

In the Cartography room, Water cells will be drawn with a superimposed number showing which room the cell corresponds to. Numbers of 100 or more are split onto two lines.

Keypresses you can use here are:

**1-4** Select guardian to edit. Willy's starting position, the Bathroom toilet, and lifts are all edited in the same way as guardians.

**5-8** Select teleporter to edit. See below for notes on teleporter editing.

**S** Make this room the 'start' room. You can then set Willy's initial position.

**C** Make this room the Cartography room (if it isn't already). It is then possible to edit the room that each water cell corresponds to - click on a water cell, and choose the room number. In the original JSW2, room 108 is the Cartography room.

### 19.6.1 Editing teleporters

- You can only edit a teleporter if the current room's patch code is 11 or 22. Otherwise the teleporter list will be blank.
- Teleporters in this room are shown in black; those in other rooms are shown in blue.
- In the room preview, teleporters are drawn with a number in the lower left-hand corner to show which teleporter they are (JSW2 only allows four teleporters in total).
- To edit a teleporter, select it. Then click where it should be put; choose the destination room from the list, and the position within it on the following screen.

## 19.7 The Title Screen Editor

The title screen editor for JSW2 is similar to the title screen editor for JSW48 (or to the other screen editors in colour attribute mode). Instead of selecting "bright" and "flash", you select the type of tile to show ("\", "/" or blank). This replaces the rather clumsy mapping of colours to tile types in JSW48.

The JSW2 title screen is smaller than the JSW48 one, and starts and ends in the middle of a line. Areas you can't change are shown in dark grey; in the game, they'll be black.

You can't manually select flashing colours, but it's possible to replace exactly one combination of ink and paper by a flashing attribute. To set this up, right-click the screen and select "Colours". You will see a screen showing:

**Map this:** The ink and paper that will be replaced. In the original JSW2, this is yellow on yellow.

**To this:** The ink, paper and flash values that will be substituted. In the original JSW2, this is flashing magenta on yellow.

It is possible to edit the 'slope' graphics by right-clicking and selecting "Tiles". Images can be imported and exported as usual.

## 20 The Guardian editor

### 20.1 Guardian behaviour notes

- In JSW2, guardians are set up by room; changing a guardian in one room will not affect that guardian in any other room.
- Guardians have 'primary' and 'secondary' movement values. If 'secondary' is zero, the guardian will be horizontal or vertical; if it is not, the guardian will be diagonal. The primary movement value can either be left/right, or up/down.

### 20.2 Editor overview

When a guardian is being edited, the top half of the screen shows a preview of the result, while the bottom half contains controls. The controls are:

Control	Description
Sprite	Select the sprite(s) used by this guardian.
X	Initial position, X-coordinate (note 1).
Y	Initial position, Y-coordinate.
L/R?	Ticked if primary movement is left/right.
Counters	See note 2.
Colour	JSW2 only supports 4 guardian colours. You can tell it was originally for the CPC, can't you?
Animation	0-7. Roughly, the number of sprite frames it uses (note 3)
Oneway	If ticked, the guardian is unidirectional (meaning that it is only drawn moving in one of its two possible directions) and is drawn in white ink.
Movements	Primary and secondary movement values.
!	Edit the guardian as hex (section 8.7)

**Note 1** X-coordinates allow values to be entered with 2-pixel precision. But if the primary direction of movement is vertical, these will be rounded to 8 pixels. For guardians whose primary motion is horizontal, the pixel position is used to start the animation at the appropriate frame.

**Note 2** JSW2, like Geoff Mode, uses a frame counter. When the guardian is first drawn, the counter is given the value in the left-hand box. The guardian heads in its initial direction until the counter reaches zero; then the guardian reverses, and the counter is given the value in the right-hand box. All subsequent reversals use the right-hand box.



**Note 3** The animation values most likely to be used are 0 (no animation); 1 (2-frame animation); 3 (4-frame animation) and 7 (4-frame animation, with separate 'left' and 'right' (or 'up' and 'down') sprites). Bits 0 and 1 of this value are a frame mask; bit 2 is set for separate left/right sprites.

## A JSW64 for machine-code programmers

The internal data format and mechanisms of JSW48 are very well known (which is one reason why there are many more JSW editors than, for example, Mutant Monty editors). This section is an attempt to document JSW64 to a similar level of detail.

### A.1 JSW64 Room Formats

Notes:

1. In the format description below, fields marked 'reserved' may be used by future versions of the JSW64 engine. Fields marked 'unused' will not; they are intended for your own data.
2. For more information on the Manic Miner 'Oxygen' and 'Portal' features, see Andrew Broad's Manic Miner room format. Type [http://www.geocities.com/andrewbroad/spectrum/willy/mm\\_format.html](http://www.geocities.com/andrewbroad/spectrum/willy/mm_format.html) into the Internet Archive Wayback Machine at <http://www.archive.org> to find it.
3. Hex numbers are denoted with a leading '#'.

Formats W, Y and [ use a 4-bit room map, so they can all specify 16 block types. But rooms in W and Y only have space for 13 block types, so the other three are defined globally (in bank 7).

#### A.1.1 Bytes #00-#B5 for formats V and X

**#00-#68** Up to 13 guardian instances for the room, terminated with #FF.

**#69-#6D** Meanings of the following 8 cell types.

First byte low nybble = type of first cell

First byte high nybble = type of second cell

2nd byte low nybble = type of third cell

etc.

**#6E-#B5** Attributes and bitmap for the 8 cell types

Cell types are:

**0** Air

**1** Water

**2** Earth

**3** Fire

**4** Ramp \

**5** Ramp /

- 6** Conveyor <<
- 7** Conveyor >>
- 8** Crumbling floor
- 9** Trampoline
- 10** Trap
- 11-15** reserved.

#### **A.1.2 Bytes #00-#B5 for formats W, Y and Z**

**#00-#40** Up to 8 guardian instances for the room, terminated with #FF.

**#41-#B5** Attributes and bitmap for 13 cell types

#### **A.1.3 Bytes #00-#B5 for format [**

**#00-#20** Up to 4 guardian instances for the room, terminated with #FF.

**#21-#25** Reserved.

**#26-#B5** Attributes and bitmap for 16 cell types

#### **A.1.4 Bytes #B5-#FF (all formats)**

**#B6-#D5** The room name, ASCII.

**#D6** Conveyor animation, and the Final Barrier flags.

**Bits 5,6,7:** Source of bottom half of room image

**Bits 2,3,4:** Source of top half of room image

Sources are:

- 0:** Normal (generated from room graphics)
- 1:** Title screen, top half
- 2:** Title screen, bottom half
- 3:** Screen buffer 1
- 4:** Screen buffer 2
- 5,6,7:** Reserved.

**Bit 1,0:** Conveyor settings

- 0:** Animate left
- 1:** Animate right
- 2:** All conveyors in room deactivated
- 3:** All conveyors in room are sticky

**#D7-#D8** Conveyor animation position

**#D9** Conveyor animation length

**#DA-#DB** Address of the guardian instance table used by this room. Normally #8000, but if a larger guardian instance table is needed it will be the address of that table, either in the room record or elsewhere in memory. The original guardian table will then be available for your own purposes.

**#DC** Solar Power beam entry point. Low 5 bits are X-coordinate; high 3 bits are Y-coordinate (solar power must start in the top 8 lines).

**#DD** Attribute of the solar beam. If this is zero, the room has no solar power. Note that unlike in Manic Miner, the solar beam does not require a green background. If you want a harmless solar beam, you'll have to set the room not to limit air supply.

**#DE** Border and flags

**Bits 0-2:** Border

**Bits 3-5:** Willy's colour

**Bit 6:** Rigor Mortis. If set, guardians are stationary until all objects in the room have been collected.

**Bit 7:** Superjump.

**#DF** Oxygen supply, major value. See Andrew Broad's Manic Miner room format for full details of this field.

**#E0** Oxygen supply, minor value. Must be a multiple of 4, or #FF if the room does not have a limited air supply.

**#E1** Item bitmap

**#E9-#EC** Exits

**#ED** Willy's sprite page. If less than #80, uses the standard Willy.

**#EE-#EF** The address of the 16x16 graphic to use for the portal. Set to 0 if this room does not have a portal.

**#F0-#F1** Position of portal attribute (based at #5C00).

**#F2-#F3** Position of portal bitmap (based at #6000).

**#F4** Portal attribute. If it is flashing the portal can be entered even if there are still items to be collected.

**#F5** Portal destination room number.

**#F6-#F7** Portal destination: Position of Willy attributes (based at #5C00).

**#F8** Portal destination: Willy's new Y-coordinate (based at #6000).

**#F9-#FA** 'Portal' patch vector

**#FB-#FC** 'Room setup' patch vector

**#FD-#FE** 'Main loop' patch vector

**#FF** More flags:

**Bit 0** set if Willy loses the items he's collected in the current room when he loses a life.

**Bit 1** set if escalators are down, 0 if up.

**Bit 2** set if ramps are escalators.

**Bit 3** set if Willy can fall any height in this room.

**Bits 4,5,6** reserved.

**Bit 7** set if this is a bonus room (each item gives Willy an extra life)

#### **A.1.5 Bytes #100 onwards: Format V**

**#100-#13F** Unused. Available to designer for portal graphic or patch vector(s).

**#140-#1FF** Room layout, 3 bits per cell.

#### **A.1.6 Bytes #100 onwards: Format W**

**#100-#1FF** Room layout, 4 bits per cell.

#### **A.1.7 Bytes #100 onwards: Format X**

**#100-#33F** Unused. Available to designer for portal graphic, patch vector(s), sprites, guardians, or all of the above.

**#340-#3FF** Room layout, 3 bits per cell.

#### **A.1.8 Bytes #100 onwards: Format Y**

**#100-#2FF** Unused. Available to designer for portal graphic, patch vector(s), sprites, guardians, or all of the above.

**#300-#3FF** Room layout, 4 bits per cell.

#### **A.1.9 Bytes #100 onwards: Format Z**

**#100-#1FF** Unused. Available to designer for portal graphic, patch vector(s), sprites or guardians.

**#200-#3FF** Room layout, 8 bits per cell. Unknown attributes take the 'water' graphic.

#### **A.1.10 Bytes #100 onwards: Format [**

**#100-#2FF** Unused. Available to designer for portal graphic, patch vector(s), sprites, guardians or all of the above.

**#300-#3FF** Room layout, 4 bits per cell.

### **A.2 Cell Types Table (formats W, Y, Z)**

In these formats, the meaning of each cell is determined globally rather than by room. There is a 16-byte table at #F4FF in bank 7. Each byte defines the meaning of a cell (so byte 0 refers to the first cell in the room definition, byte 1 to the second, and so on).

Cell types are:

**0** Air

**1** Water

- 2 Earth
- 3 Fire
- 4 Ramp \
- 5 Ramp /
- 6 Conveyor <<
- 7 Conveyor >>
- 8 Crumbling floor
- 9 Trampoline
- 10 Trap
- 11-15 reserved.

After these 16 bytes are the attributes and bitmaps for the three "global" cells – the ones which are the same in every room.

### A.3 Cell Types Table (format I)

The cell type table is at #F519 (again, in bank 7) and is 16 bytes long. There are no global cell definitions.

### A.4 JSW64 Guardians

The guardians in JSW64 are based on the JSW128 guardians, but with a few added features. A guardian definition is 8 bytes long; I refer to them as GB0-GB7.

The low 4 bits of GB0 give the guardian type. This is one of:

- 0 Blank
- 1 Horizontal
- 2 Vertical
- 3 Rope
- 4 Arrow
- 5 Diagonal 1
- 6 Diagonal 2
- 7 Vertical, colour-cycling
- 8 Extended guardian. Top 4 bits give subtype:

**#08:** Skylab. Behaves like a vertical guardian, except that at the end of its travel it 'crashes' and then reappears 8 columns to the right (wrapping, of course). Other bytes as vertical guardian, except that byte 7 is 'crash' position and byte 6 is 'restart' position. So for an upward skylab, byte 7 must be less than byte 6.

**#18:** Angry Eugene. Other bytes as vertical guardian, except that byte 1 is colour; high bits must be zero. An Angry Eugene goes to the end of his travel and stops dead.

**#28:** Angry Eugene (colour-cycling based on air supply)

**#38:** Angry Eugene (colour-cycling using JSW64 method).

**#88:** Trigger.

**Byte 0:** #88

**Bytes 1-2:** Address of a flag. When this flag becomes zero, bytes 3-7 are applied to the next guardian in the table; and byte 0 of this guardian is cleared so it doesn't happen again.

**Byte 3:** New byte 0 of next guardian.

**Bytes 4-7:** New bytes 4-7 of next guardian.

**#98:** Switch.

**Byte 0:** #98.

**Byte 1:** Ink.

**Byte 2:** X-coordinate and frame. The top half of the guardian sprite will be the 'off' switch and the bottom half will be the 'on' switch.

**Byte 3:** Y-coordinate.

**Byte 4:** State. 1=Off 0=On.

**Byte 5:** Sprite page.

**Byte 6:** If it's off and Willy hits it, change state to this.

**Byte 7:** If it's on and Willy hits it, change state to this.

**#A8:** Opening wall.

**Byte 0:** #A8.

**Bytes 1-2:** Address of flag (same as for the Trigger).

**Byte 3:** Number of vertical lines this will remove.

**Byte 4:** State. Initialised by JSWED to 1. It will become 0 when all lines are removed.

**Byte 5:** Number of lines removed so far. Initialised by JSWED to 0.

**Byte 6:** X-coordinate of wall to open.

**Byte 7:** Y-coordinate of wall to open.

**9** - Horizontal, colour-cycling

**#A** - Vertical

**#B** - Reserved

**#C** - Reserved

**#D** - Diagonal 1, colour-cycling

**#E** - Diagonal 2, colour-cycling

**#F** - Vertical, colour-cycling

#### A.4.1 Horizontal and Diagonal Guardians

These guardian types behave like the standard JSW48 horizontal guardian.

Diagonal ones use GB4 as their vertical speed.

The top 3 bits of GB7 have these meanings:

**Bit 5:** Half speed. Guardian will only move every other frame.

**Bit 6:** Half-speed ticker (used when the game is running).

**Bit 7:** Double speed. Guardian will be animated twice per frame. This means you only get two animation images instead of four, but the guardian now goes twice as fast as Willy!

#### A.4.2 Ropes

Set bit 6 of GB0 to have a stationary rope. JSW64 has been patched so that a rope definition fits in 8 bytes; a rope can be followed by other guardians than arrows and it won't corrupt them.

#### A.4.3 Arrows

GB1 is the ink colour to use for the arrow. Bits 3-7 of this byte will also be ORed to the screen, so it's best to leave them as 0.

GB3 is the bitmap for the middle line of the arrow.

### A.5 Patch Vectors

There are three patch vectors in the room data. In order, these are:

- Portal vector. This is called when Willy enters a portal. The Patch Vector Jump-block (below) provides two possible implementations of this; or you could write your own.
- Note that when you return from the Portal vector, the game will assume you have moved to a new room and set things up accordingly. If you want Willy to stay in the same room, return with

```
POP BC  
RET
```

- Room Setup vector. This is called when Willy enters a new room, after the room attributes and bitmap patterns have been generated. This can be used to put custom graphics in a room (like the Final Barrier does in Manic Miner). The default implementation is just to return.
- Main Loop vector. This is called in the main game loop, just after the screen has been drawn and just before Willy moves. It corresponds to the patch vector in 'Geoff Mode' JSW.

### A.5.1 Patch Vector Jumpblock

The following entry points are provided as helper functions for patches that you write. More may be added later.

The jumpblock is at a fixed location, and this will stay fixed in future JSW64 versions.

```
#869F RET
```

For rooms without patches, the Room Setup and Main Loop patch vectors should be set to this address.

```
#86A0 JP portal_with_fx_and_air
```

This is the default portal patch vector. It takes Willy to the location specified in the room's portal settings. The portal special effect occurs (as in Manic Miner); and if the room has a limited air supply, it will be run down to zero with a whining noise (again, as in Manic Miner).

```
#86A3 JP portal_with_fx
```

As above, but does not do the 'air-running-down' thing.

```
#86A6 JP portal_without_fx
```

As above, but with no special effects at all (like the JSW128 teleporter).

```
#86A9 JP ptl_fx
```

This performs the portal special effect only.

```
#86AC JP airend
```

This performs the air-running-down-to-zero effect.

```
#86AF JP ptl_both
```

This performs the above two effects, one after the other.

```
#86B2 RST 0
```

The last entry in the jump block is always RST 0. If more entries are added, they will go at #86B2 and the RST 0 will move to the new end of the jump block.

### A.5.2 Music Jumpblock (original music player)

The following entry points are provided as helper functions for patches that might want to amend the game music system.

The jumpblock is at a fixed location, and this will stay fixed in future JSW64 versions. It is also present in JSW128 (HL9 and later).

```
#FEC0 JP tuneoff
```

This will stop the in-game tune playing. Returns NZ if the tune was not playing, Z if it was.



#FEC3 JP tuneon

This will set the tune going. Note that the tune pointers at #FE8E must have been set to point to the start of the selected tune.

#FEC6 JP newtune

Change the in-game tune. Enter with HL pointing to the 6-byte header of the new in-game tune. If no tune is playing, nothing obvious happens; if a tune is playing, it is stopped and the new tune starts.

#FEC9 DEFS 6

A copy of the 6-byte header of the last tune selected with NEWTUNE (#FEC6). This is the tune that will be played when the current in-game tune comes to an end (used to make the tune into a continuous loop).

### A.5.3 Music Jumpblock (SoundTracker)

When the SoundTracker player is loaded, the music jumpblock is slightly different, because of the way SoundTracker handles looping.

#FEC0 JP tuneoff

This will stop the in-game tune playing. Returns NZ if the tune was not playing, Z if it was.

#FEC3 JP tuneon

This will set the tune going. Note that the tune must previously have been initialised by one of the three entry points below.

#FEC6 JP newtune

Change the in-game tune. Enter with HL pointing to the SoundTracker file to play. If no tune is playing, nothing obvious happens; if a tune is playing, it is stopped and the new tune starts.

#FEC9 JP newsingle

As #FEC6; in addition, the player will be set to play the tune once and then stop.

#FECC JP newloop

As #FEC6; in addition, the player will be set to play the tune on a loop.

## A.6 Guardian Patch Vectors

It is possible to define new guardian types, or change the behaviour of existing guardian types (JSW64:Manic Miner does, for example). The JSW64 engine supports up to 30 guardian types, the behaviour of which is defined in four tables:

**#84D2-#84F1:** Table of 16 movement routines, for guardian types 0-15 (types 0 and 8 should not be altered). The movement routine updates the guardian's position and colour, but does not usually test for collisions. At the end of a movement routine, jump to the movement handler for type 0; if you RETurn instead, guardians after your custom guardian will not move.

**#EC66-#EC85:** Table of 16 drawing routines, for guardian types 0-15 (types 0 and 8 should not be altered). As with movement, you should jump to the type 0 drawing routine rather than RETurn. If Willy should lose a life, jump to MAN-DEAD1, #90B7.

**#EC98-#ECB7:** Table of 16 movement routines for guardian types #08, #18, #28...

**#ECB8-#ECD7:** Table of 16 drawing routines for guardian types #08, #18, #28...

A standard JSW64 game leaves guardian types #0B, #0C, #48, #58, #68, #78, #D8, #E8 and #F8 undefined. JSW64:Manic Miner defines #58, #68 and #D8 as custom guardian types, though only #D8 is used in the game itself. The "Droplet" patch defines #78.

## A.7 Memory Map

This section lists memory areas or addresses that might be useful to a machine-code programmer writing JSW64 patch vectors or other custom code. The list does not attempt to cover every change between JSW48 and JSW64.

In normal use, memory bank 0 is present at #C000. The other banks are mainly used to store data, though some room-initialisation code is present in bank 7.

**#8200-#82FF:** In formats V and W, this memory is free for the game designer to put sprites or other data in (JSWED does not support this yet). In formats X, Y, Z and [ it contains the third page of the currently loaded room definition. Note that in formats X, Y and [, this is part of the 'free space' in the room, and may contain patch vectors, guardians or sprites.

**#8300-#83FF:** In formats V and W, this memory is free for the game designer to put sprites or other data in. In formats X, Y, Z and [ it contains the fourth page of the current room definition. In format X (but not Y, Z or [) the first 64 bytes of this page are part of the 'free space' in the room.

**#8553:** Number of objects in the current room that haven't yet been collected.

**#85C9:** Engine variant, 'V' to '['. Not used by the game, but essential for JSWED.

**#85CA:** Number of items in the game (in JSW48 and JSW128 this is at #A3FF).

**#8640-#869E:** Memory map, as in JSW128, but starts 12 bytes sooner.

**#87B2:** Call a function in memory bank 7. Enter with HL=address of function. DE, IX and IY will be passed unchanged to the function; AF and BC are used by this routine.

**#8D33-#8D52:** Runtime cell type table - expanded from the table in the room definition (variants V,X) or the global table (other variants). Two bytes per cell type; the first is the attribute, and the second is the behaviour (0=Air, 1=Water etc.). This could be changed by patch vector code at run time. Note that in variants V and X, only the first 8 entries in the table are used.

**#97E0-#97FF:** The attributes used to draw the 'air remaining' bar.

**#C000-#C0FF:** In formats V and W, contains object locations. In variants X, Y, Z and [ this is unused. It's possible to extend the font into this area if you want characters 0-31 to be printable.

**#C100-#C3FF:** The font.

### A.7.1 Banks 1,3,4,6

These memory banks hold the room definitions. Bank 1 holds the first 32 rooms (V and W); or the first 16 rooms (other variants). Then Bank 3 holds the next 32/16 rooms, and so on.

### A.7.2 Bank 7 memory map

**#C000-#D5FF:** Title screen

**#D600-#DAFF:** Scrolly message

**#DB00-#DB05:** Bank 7 jumpblock (JSW64 internal, unlikely to be useful in patch vectors).

**#DB06-#DD93:** Room decompression code (size varies, depending which variant we're talking about)

**#DD94-#E2FE:** Reserved

**#E2FF-#F4FE:** The two 'top third' screens. Each is formed of 2k bitmap and 256 bytes attributes.

**#F4FF-#F529:** Cell map and global cell definitions

**#F52A-#FFFF:** Title screen music

## B Format of exported room definitions

The file format used for import/export is XML.

### B.1 Overall format

The root node must be <rooms>, in the namespace "http://www.seasip.demon.co.uk/Jsw/jswedrooms.dtd". Don't try to download a DTD from that address, by the way, because there isn't one there.

```
<?xml version="1.0"?>
<rooms xmlns="http://www.seasip.demon.co.uk/Jsw/jswedrooms.dtd">
```

The <rooms> node contains one or more <room> nodes, each with a *number* attribute:

```
<room number="0">
<room number="2">
```

It will probably also contain one or more <sprite> nodes.

## B.2 <room> node

A <room> node corresponds to one room (surprise!). It contains:

- A title - eg: <title> The Off Licence </title>
- A <layout> section.
- An optional <exits> section.
- An optional <items> section.
- An optional <guardians> section.
- An optional <portal> section.
- An optional <hints> section.

### B.2.1 <layout>

The <layout> section contains the grid of fixed cells composing the room. It begins with a <cells> section:

```
<cells count="nn">
  <cell number="xx" behaviour="behaviour" ink="i" paper="p"
    bright="b" flash="f">bitmap</cell>
  ...
</cells>
```

Each cell has the following values:

**number** is the 2-digit hex number used to represent that cell in the room bitmap (which follows)

**behaviour** is one of 'air', 'water', 'earth', 'fire', 'lramp', 'rramp', 'lconveyor', 'rconveyor', 'crumbly', 'trampoline' and 'trap'.

**ink/paper/flash/bright** are the attributes in which to draw this cell. Flash/Bright can be omitted if they are 0.

**bitmap** is treated as a sequence of eight hex bytes defining the cell graphic.

After the <cells> comes a <bits> node. This has width and height attributes, giving the size of the room layout (all JSW/MM games have width=32, height=16). Between <bits> and </bits> should be 2\*width\*height hex digits (plus any whitespace you want; it will be ignored). Each pair of hex digits corresponds to a character cell, working from left to right and top to bottom.

### B.2.2 <exits>

The <exits> section is a single node:

```
<exits left="1" right="2" up="3" down="4"/>
```

The numbers refer to other <room>s, which may or may not be present in the file.

### B.2.3 <items>

The <items> section, if present, can contain:

- A <cell> node. Between <cell> and </cell> are 16 hex digits, giving the 8 rows of an 8x8 bitmap.
- One or more <item> nodes, each giving the coordinates of an item (eg: <item x="10" y="4" /> )

### B.2.4 <guardians>

The <guardians> section describes the guardians in the room. It has a type value which describes the guardian storage format; currently the only supported types are 'jsw' and 'jsw2', giving: <guardians type="jsw"> or <guardians type="jsw2">. There may be multiple <guardians> sections, containing the same guardians rendered in different formats.

Within the section, each guardian is described by a node such as:

```
<vg frame="0" fastanim="1" ink="5" bright="1" framemask="3" x="10"
  startframe="4" y="48" speed="8" limit1="16" limit2="208">
  <spriteref frame="0" xref="be80"/>
  <spriteref frame="1" xref="bea0"/>
  ...
</vg>
```

The guardian nodes supported are:

**blank** A blank guardian.

**hg** A horizontal guardian.

**vg** A vertical guardian.

**gg** A JSW2 generic guardian.

**rope** A rope.

**arrow** An arrow.

**dgnw** A diagonal guardian going Northwest.

**dgne** A diagonal guardian going Northeast.

**skylab** A Skylab.

**eugene** An angry Eugene.

**trigger** A trigger.

**switch** A switch.

**wall** An opening wall.

**stopper** A stopper.

Those guardians which use sprites contain <spriteref> nodes, describing what sprite is used for each frame. The actual graphics are contained in <sprite> nodes further down the file. Sprite IDs are Hex numbers; up to 6 digits are allowed.

### B.2.5 <portal>

The <portal> section is present if there's a portal in the room:

```
<portal x="28" y="10" ink="7" paper="1" bright="1"
      target="46" targetx="2" targety="13" sprite="218100"/>
```

and describes the colour and position of the portal, where it takes you to, and what sprite should be used to draw it.

### B.2.6 <hints>

The <hints> section contains information which is not supported by all game engines, but may help with accurate reconstruction of a room. Within <hints> will be one or more <hint> nodes; each one has a *type* attribute plus any other necessary attributes.

For example, exporting a JSW48 room can produce up to three hints:

Hint type	Additional parameters	Meaning
conveyor	x, y, length, direction	Conveyor settings. Present if the room has a conveyor.
ramp	x, y, length, direction	Ramp settings. Present if the room has a ramp.
escalator	direction	Present if the room has a conveyor ramp.

If these hints are not present, the conveyor and ramp details will be reconstructed from the room <layout>. This will do a pretty good job but it doesn't support "stopped" or "sticky" conveyors.

Some of the hints will also include sprite IDs. For example, exporting a Manic Miner game will store references to the Kong and Eugene sprites in this section. These are expressed in the same way as for guardians - as <sprite ref> nodes within the appropriate <hint>.

## B.3 <sprite> node

For each sprite frame referred to by other parts of the file, there's a <sprite> node:

```
<sprite base="00e3e0" w="16" h="16">
  00 00 00 00 00 00 03 c0 0c 30 12 48 2a 54 5f fa
  f6 7f 47 e2 23 c4 10 08 0c 30 03 c0 00 00 00 00
</sprite>
```

The 'base' is the sprite's identifier, as used elsewhere in the file. The "w" and "h" give the dimensions of the sprite image. Within the node, there should be  $((w+7)/8)*h*2$  hex digits, giving the bitmap.

## C Assembly file format

The Assembly file format is the most suitable format for editing a tune file or creating a new one (though that's not saying much). It's in the form of an assembly language listing, compatible with Udo Munk's Z80ASM assembler. The only directives used (and the only ones that will be recognised on import) are:

```
org defw defb end
```

A tune specified in Assembly format should be laid out as follows:

```
org      0f700h
```

The org (origin) directive should be the first one in the file. The value following it should be greater than 4000h but is otherwise comparatively unimportant. In the case of tunes saved by JSWED, it gives the address where the tune was saved from.

The next three directives must be the addresses of the three channels:

```
defw     Gf706
defw     Gf751
defw     Gf752
```

Usually these are stored as label names, but you could use numerical addresses if you wanted. JSWED outputs labels of the form Gxxxx, but you might want to use more memorable names - eg:

```
defw     channel1
defw     channel2
defw     channel3
```

Following the channel addresses, the remainder of the file describes the tune data. For example:

```
GF706:  defb     0FEh      ;Call subroutine
        defw     Gf732
        defb     0FEh      ;Call subroutine
        defw     Gf742
        defb     03eh,00ah
        defb     03bh,00ah
        defb     03eh,014h
        defb     03dh,028h
```

The first line in the listing has a label. You can put a label on any line of the file, but it only makes sense to put them at the start of channels and at the start of subroutines.

Each channel is formed of a series of instructions, constructed using defb and defw directives. The first byte is a command byte, which may then be followed by parameter bytes. These can be on the same line or on different lines; JSWED doesn't care.

Any text on a line after a ; is ignored.

```
end
```

JSWED's output files end with the 'end' directive, but this is optional.

Command byte	Number of parameter bytes	Meaning
000h	1	Rest. The parameter byte is the duration in jiffies.
001h-0F8h	1	Play a note. The parameter byte is the duration in jiffies. The command byte is the pitch in semitones; 48 is middle C.
0F9h	1	Set waveform effect (like 'W' in a PLAY command)
0FAh	1	Set waveform period (like 'X' in a PLAY command)
0FBh	1	Set the volume (like 'V' in a PLAY command) If you want to use waveforms, set the volume to 16 first.
0FCh	3	Loop - used to repeat a preceding section. The first and second parameter bytes should both be set to the loop count. The third is the relative address of the start of the section to repeat.
0FDh	0	Return from a sub-tune.
0FEh	2	Call a sub-tune, which must end with a 0FDh byte. The 2 parameter bytes are the address of the sub-tune.
0FFh	0	End of channel. All channels must end with this.

Note that on the import side, JSWED is very restricted compared to a real assembler. You can't use EQU directives, conditional assembly or arithmetic. If you need any of these, use a proper assembler.