

MINE MINER

Remaking a classic



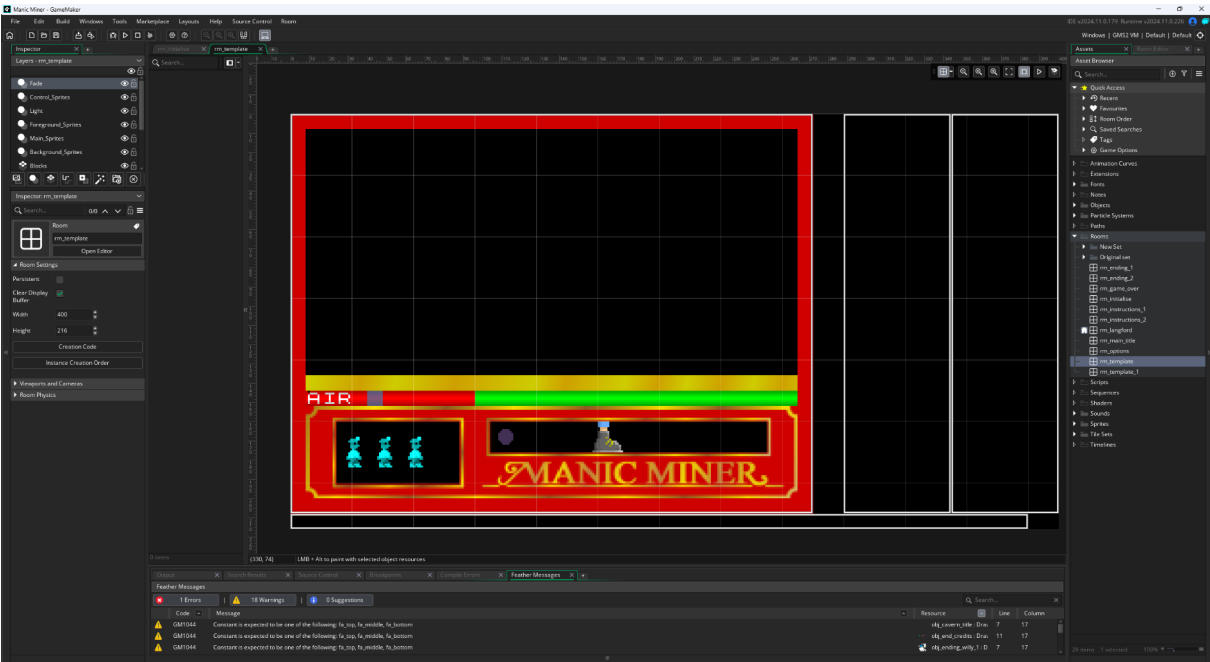
LANGFORD
PRODUCTIONS

REMAKING MANIC MINER

First of all, a remake of the Spectrum all-time classic *Manic Miner* was not my intention. I was nearing the end of another project and waiting for some game music from a very helpful chap in Portugal and was chatting to a fellow 'game remaker' who I had met on Facebook (you may know him for his recent remake of *Scuba Dive* - one that I am definitely very envious of)! We both use different software to create our games and I put together some extremely simple platformer code to show him how I go about it using Gamemaker 2. I then wondered how easy it would be to turn that code into Manic Miner and thought I'd expand this demo to make The Central Cavern just to see how it looked. Manic Miner had been remade very successfully for PC before so remaking the whole game was not initially on the cards at all.

Then I thought "What the Hell".

I've been using Gamemaker to create games now for over 10 years. Manic Miner is my 21st completed game and I would say that I've still only scratched the surface of what it can do. There are literally thousands of commands in GML of which I probably only use just a fraction, but along the way I tend to add two or three more commands to my repertoire and discover new, easier ways of getting things done which previously were far more complicated for me.



Manic Miner screen template in Gamemaker

The first thing to do was to set up a template for the screen, which looked very much like the one above, but obviously with 'placeholder' graphics rather than what you see here. This contains everything that appears on every level in the game - the border, the air indicator, the scoreboard and lives remaining display. The boot in this template is... well... some of you will know, some will not. The outlined white rectangles are various sections of the screen which will be shown at different positions on your game monitor. The two areas on the right will be displayed each side of the main game screen which would be centered. I'd originally thought of having a fancy border or backdrop in those but decided against it as I went on. The main viewport containing the game is 272 pixels wide and 208 high and is magnified 5 times to make it fit better onto a 1920x1080 monitor. Once the screen was set up, I could create a generic wall, floor and 'kill block', put them on the screen and set about coding Miner Willy's movements.

MINER WILLY

The most important thing to me when making Manic Miner was that the movement of Miner Will had to be spot on. The game is so well known and different people have their own tried and tested ways of getting through each screen that if the movement was anything other than 100% accurate it would be like playing a different game altogether.

Originally I had the game running at 60 frames per second, which meant that for Willy to move at roughly the same speed as the Spectrum version he would move just 0.2 pixels every frame. At first I coded all this as I would any other game. I was pleased that it was accurate and continued with setting up all the remaining screens and enemies. I'd got to the point where all 20 original screens were in place before I realised that this player movement was not good enough at all. I deleted the Miner Willy code altogether, changed the game to run at 12 frames per second and r-wrote the Miner Willy code altogether.

The drop in frame rate meant that I also had to also change every enemy code in the game so far plus other bits and pieces such as the air indicator.

In the original Manic Miner, wherever Miner Willy is on the screen he will always be at the correct animation screen. For example, when at the very edge of a platform he will always be at animation frame 4 (see the animation frames above). This was a huge headache for me! In my original code he was moving 0.2 pixels every game frame and this made it extremely difficult to get the animation frames right. Slowing the game to 12 frames per second meant I could move Willy 2 pixels every frame just as he does in the Spectrum version.



```
obj_willy: Step
254
255 } else {
256   movetry = 0;
257 }
258
259 if x1 > x && sprite_index != spr_willy_left sprite_index = spr_willy_left;
260 if x1 < x && sprite_index != spr_willy_right sprite_index = spr_willy_right;
261
262 tmpx = (floor(x / 8) * 8);
263 tmpx2 = x - tmpx;
264
265 if sprite_index = spr_willy_right {
266   if tmpx2 = 0 ii = 1;
267   if tmpx2 = 2 ii = 0;
268   if tmpx2 = 4 ii = 3;
269   if tmpx2 = 6 ii = 0;
270 }
271
272 if sprite_index = spr_willy_left {
273   if tmpx2 = 0 ii = 0;
274   if tmpx2 = 2 ii = 1;
275   if tmpx2 = 4 ii = 0;
276   if tmpx2 = 6 ii = 3;
277 }
278
279 image_index = ii;
280
281
```

Above is the code that tells the Miner Willy sprite which frame of animation should be displayed. Though it looks pretty simple now, I think getting to this was one of the biggest headaches in creating the game (this was before I got to the conveyors!).

Lines 260 and 261 pretty much just say if he has moved right, show the willy facing right sprite, and if he has moved left, show the willy facing left sprite.

Lines 263 and 264 then work out where Willy is in relation to an 8x8 pixel grid (each platform section being 8 pixels wide). Lines 268 to 275 then decide which frame of animation should be used depending on that position and line 281 displays that animation frame.

So Willy can now move left and right - he now needs to jump. In any other platformer my jumping code would be pretty generic, with a calculated jump arc which pretty much auto adjusts the Y coordinate and the X coordinate. In Manic Miner the jump arc was very specific and had to be replicated in my version. To work it out, I loaded up the original in Spectaculator and repeatedly paused the game, took screenshots, skipped to the next frame, took another screenshot and so on, then loaded all the screenshots into my paint program to see how far he had moved in each frame. I then coded each frame movement specifically as shown below.

```
163 if jumping = 1 xmove = movexy;
164 if jumping = 1 || jumping = 2 ymove = -4;
165 if jumping = 3 || jumping = 4 ymove = -3;
166 if jumping = 5 || jumping = 6 ymove = -2;
167 if jumping = 7 || jumping = 8 ymove = -1;
168 if jumping = 9 || jumping = 10 ymove = 0;
169 if jumping = 11 || jumping = 12 ymove = 1;
170 if jumping = 13 || jumping = 14 ymove = 2;
171 if jumping = 15 || jumping = 16 ymove = 3;
172 if jumping = 17 ymove = 4;
173 if jumping = 18 {
174     ymove = 4;
175     xmove = 0;
176 }
```

‘xmove’ the amount to add to his x coordinate and is always going to be 2, -2 or 0 depending on if he is jumping left, right or straight up. ‘jumping’ is increased by 1 for each frame of the jump. So line 164 says if we are at jump frame 1 or jump frame 2 then he should move 4 pixels up. At frame 3 and 4 he should move 3 pixels up and so on. At the last frame, frame 18, Willy moves 4 frames down and doesn’t move left or right at all which is why xmove is set to 0.

Phew! That’s enough about getting Miner Willy to move about I think. I won’t go into how I got him to interact with conveyor belts etc as it gets far too complicated. Suffice to say that how Miner Willy interacts with these things is completely different to any other platformer I’ve made.

SETTING UP THE LEVELS

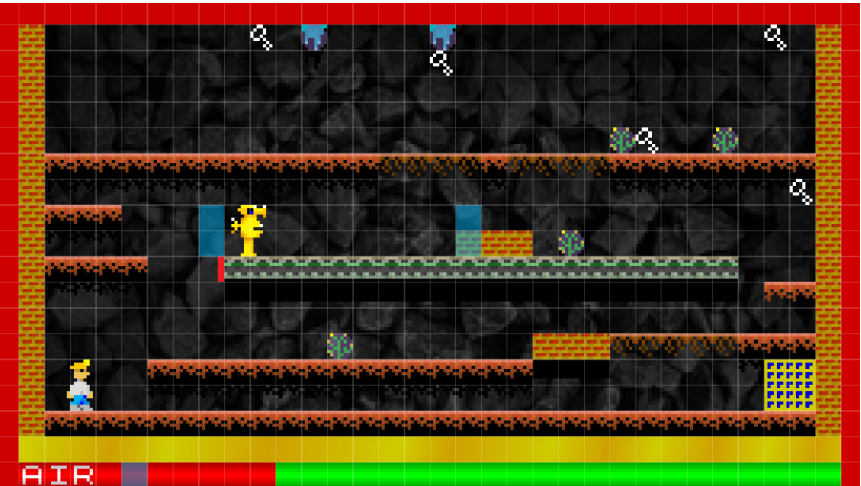


To create any game in Gamemaker you need to create tiles, sprites, objects, rooms and sounds. Each object is assigned a sprite (if needed) and contains the GML code for what you want that object to do. I’ll get onto that in a bit when I talk about the Manic Miner enemies.

Blocks are an efficient way to lay out the screen. On the left are the blocks needed for level 1, The Central Cavern (In the end, I did not display the cavern name with blocks, so in this case I did not use the yellow letter blocks).

Along the top row you can see the blocks for the solid wall, the regular platform, two “kill” blocks and the red border. There are also blocks for the air gauge and shadows which I used on some screens to make the platforms stand out more from the backgrounds.

Here is the “room” for level 1. The rooms are created in layers of backgrounds, tiles, and objects. In this level there is a pixel background, on top of that is a tile layer for the shadows, then another tile layer for the blocks. Then there is an object layer (this is on top of the tile layer as Willy has to move in front of the platforms). There is a foreground objects layer which contains the finish block.



This is above the layer Willy is on as Willy moves behind the finish block. Above that is what I call a control layer. Each object can be coded to interact with other objects, so I create this layer solely for that purpose. During gameplay this layer is invisible. I have an object for walls, an object for platforms and ‘kill’ squares and objects for left and right conveyors. You will also see a square next to “AIR” which is the air gauge. The blue blocks to the left and right of the conveyor belt in this screen are used in the code to tell the enemy where to turn and start walking in the opposite direction. Which brings us on to...

ENEMIES

The enemies in Manic Miner are all pretty much alike and simple to code. Each object has separate sections of code. The main ones are CREATE, STEP and DRAW. To the left is the create step for the enemy in level 1. The CREATE step is run once when the screen is first displayed. All it really does is set up the required variable for this object. In this case it sets the initial x movement to 2 pixels (moving right), the image_speed command tells it how fast to animate the sprite and I also set up a variable ‘is’ to contain that speed; the reason will become clear in the STEP code.

```
obj_enemy_01_1: Create
1 // Description: Insert description here
2 // You can write your code in this editor
3
4 xmove = 2;
5 image_speed = 1;
6 is = 0;
```

```
obj_enemy_01_1: Step
1 // Description: Insert description here
2 // You can write your code in this editor
3
4 if instance_exists(obj_death_fade) { finished = 1;
5     image_speed = 0;
6     exit;
7 }
8
9 if instance_exists(obj_pause) {
10     if image_speed != 0 {
11         is = image_speed;
12         image_speed = 0;
13     }
14     else {
15         if image_speed = 0 image_speed = is;
16     }
17 }
18
19 if move > 0 {
20     if collision_rectangle(x + 5, y - 14, x + 6, y, obj_blocker, 0, 1) {
21         xmove = 0 - xmove;
22         image_speed = -1;
23     }
24 }
25 else {
26     if collision_rectangle(x - 6, y - 14, x - 5, y, obj_blocker, 0, 1) {
27         xmove = 0 - xmove;
28         image_speed = 1;
29     }
30 }
31 x += xmove;
```

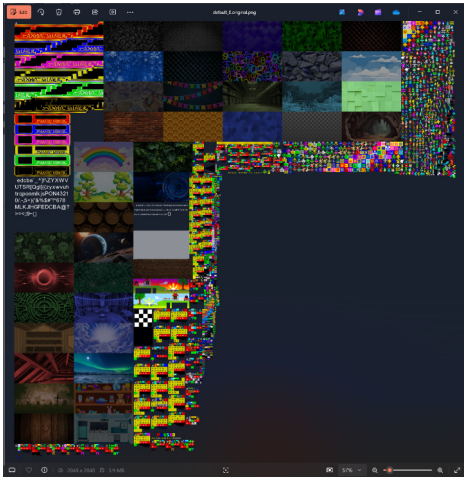
Here is the STEP code for the same enemy. It’s quite short, as I said the enemies are all pretty simple in Manic Miner. Lines 4 to 7 just check if the Willy has died. When Willy dies an object called “obj_death_fade” is created so these lines check if that object exists - if it does, the sprite stops animating and the step is exited before the movement code is run. Lines 9 to 17 check if there is an object called “obj_pause”. Pausing a game is not simple in Gamemaker! If it does it will set variable ‘is’ to the current image speed, set the image speed to 0 and exit the code. When obj_paused has gone, it will set the image speed back to ‘is’.

Then comes the main movement code - lines 19 to 29. Lines 19 to 23 check if the sprite is moving right, if it is it will check the area just to the right of the sprite for a blocker (the blue blocks mentioned above). If it finds one it will flip the sprite and set it off moving to the left. Lines 24 to 29 do the same for when the sprite is moving left. The last line just updates the X coordinate depending on variable ‘xmove’. Simple!

DRAW code is not needed in the enemy code. The sprite is drawn on the current screen by default. DRAW code is needed in objects such as the air gauge where a rectangle is drawn on screen the size of which depends on the amount of air remaining.

The original enemy sprites were gathered by endless pause/screenshot/unpauses of the game on an emulator as I did with the jump arc. They were loaded into Gamemaker and coloured in the built-in sprite designer.

There’s not much more to say about how the enemies work in Manic Miner, all are pretty much the same whether moving horizontally and vertically, they just move at different speeds which is all set up in that ‘xmove’ command. So to complete this page, just for fun, on the right are all the graphics, backgrounds, blocks and sprites used in the 40 levels of this version of Manic Miner. Better get your microscope out if you want to find your favourite enemy!



PUTTING IT ALL TOGETHER

This, for me, is always the fun bit. Adding title screens, instructions, music and fancy stuff.

In most of my games, the music is usually made for me by very generous and talented individuals such as Ricardo Vieira or Pedro Pimenta. I'm able to read music (very slowly), but would not even know where to start with regard to coming up with new music for games.

Fortunately, in Manic Miner, the tunes used are well known classics of which sheet music is pretty easy to track down with a quick search on Google. So I downloaded the free version of Musescore and off I went. The only creative part in this process was to decide what instruments I wanted to use for each tune.

Flute

Tuba

Harpsichord

$\text{♩} = 80$



In the case of “In the Hall of the Mountain King” - the original in-game tune - I went for a flute, a tuba and a harpsichord, though I doubt those instruments sound much like their equivalents in Musescore. The main title tune is “The Blue Danube” and, in the final stages of creating the game I decided to use two different classical tunes for the new screens and for the mixed mode. These were “The Radetzky March” and “Für Elise”. The keyboard on the title screen does actually play the correct notes for “Blue Danube” apart from one note which was too high for the on-screen keyboard!



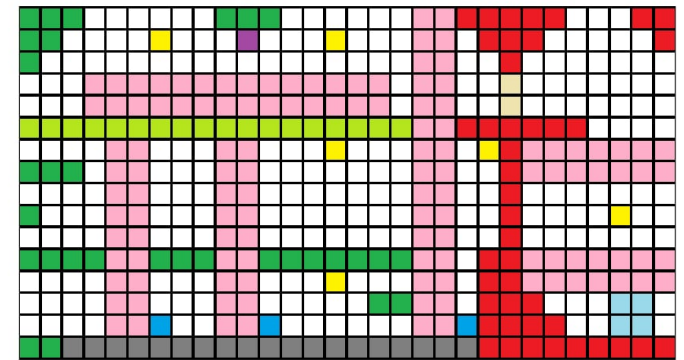
The Miner Willy on the title screen I drew using Serif DrawPlus X8. This software has now been superceded by Affinity, but I've been using DrawPlus for so long now I have yet to pluck up the courage to jump over to Affinity - even though I've now owned the Affinity software for around two years!

Though I would class myself as a bit of an artist, I'm not too hot with pixels. I'm much more at home with pencil and paper and/or vector graphics. I managed quite well to adapt the sprites used in game, but larger subjects I don't fare too well on. For the nice scene on the title screen and on the last level (and on the next page here) I called upon the services of Paul Weller of Sunteam who created the gorgeous pixel art in Steve Watson's excellent Spectrum remakes. Anyone who has played Steve's versions of Scuba Dive and Olli & Lissa will be well aware of the amazing talent for pixels that Paul has.



Above: Final cavern graphic created by Paul Weller, also used on the title screen. For the last “new” level I flipped this and changed the colours to make it a sunset.

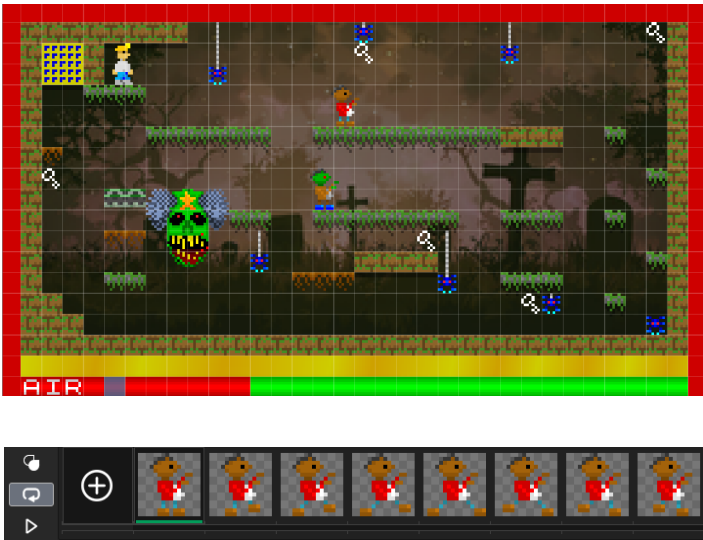
Those extra twenty levels.... Well, they're a bit of a mish-mash really. A few of them were just mirror versions of original caverns, for instance “The Bee Side” is a mirror of “The Central Cavern” with an added enemy and the original enemy moving a little further than before. Some of the screens were inspired by rooms in Jet Set Willy, often with slight changes. A few screens were plundered from the many, many unofficial remakes of Manic Miner of which I spent hours watching walk-through videos! These I also made a few subtle layout changes and added my own enemies. The remaining levels were designed from scratch either by me, or by my good friend John Davies, who definitely has a knack for this whole level design thing. It was John who helped greatly with suggestions on my Project ZX games - they certainly wouldn't have been the same without him and they definitely wouldn't have been so difficult (though he may deny that bit). Many of the sprites in the new levels were lifted from Jet Set Willy, some from other Spectrum platform games and some are original Langford creations!



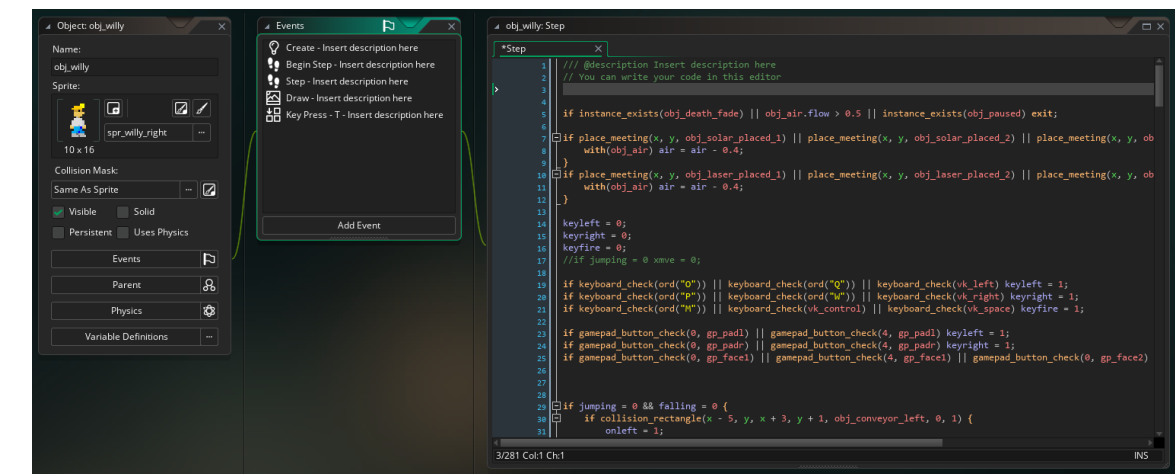
■ SOLID ■ JUMP-THRU ■ CRUMBLE ■ CONVEYOR ■ FATAL ■ KEY ■ SWITCH ■ SWITCH BLOCK

For the screens contributed by John, I sent him the grid, he would beaver away at it and send it back, all nicely coloured and coded. The example on the left is for the “Leap of Faith” level. The pink squares represent where the enemies will move. John providing some screens was a great help as level design is not my strongest talent! With John living in Indonesia and me in London, what on Earth would I have done before the internet and Facebook messenger!?

Miner sequels I found to be a bit too fragmented and tricky to find your way around. Plus many incorporated non-standard features like falling off the bottom of the screen and reappearing at the top, so for the ones I used I adapted them to suit the Manic Miner rules and to tidy them up a bit. I also gave them new names and themes. One of my favourites is “Darkness Falls Across the Land” - an homage to Michael Jackson's Thriller. I was particularly proud of the little MJ and zombie traipsing back and forth doing the ‘Thriller’ walk. And that big scary face? Yes there were no sprites that size in Manic Miner but it was one of my favourites from Jet Set Willy.



AND THAT’S THAT



Levels designed and drawn, background graphics done, sprites drawn, code written, music and polish added and play testing done by Steve and John, it probably sounds from this doc that everything was carefully planned and ran very smoothly. But no game I’ve ever written ever goes like that. I’m sure if a professional Gamemaker coder were to ever inspect my code they’d either laugh or have a heart attack, but it works and that’s all I care about! Code for the main player sprite started off in such an orderly fashion until I got to all the quirks and weirdness of the original player movement. Then it became a patchwork of extra lines written here and there, sections commented out and replaced by new bits and pieces and random code put here and there to cater for player movement that just wasn’t quite right. And this continued after I posted the game and watched videos made of the game, when I noticed other quirks in the player movement that I hadn’t spotted before.

My messy unstructured code is the main reason why I never return to update games once they’re done and dusted. After a few weeks away from them I can never find my way around the code again!

And that’s it for Manic Miner. Onwards and upwards to the next project!



A HELPING HAND

Cheating should really not be encouraged, but Manic Miner would just not be Manic Miner without that little secret code that, at the time was probably revealed in just about every Spectrum or home computer magazine there was!

But if you missed it, if you’re too young to know about it, or if Google is just not playing ball here it is.



That’s the one! Type that number whilst in play and a lovely boot will appear in the scoreboard area. Not only do you now have infinite lives, but you can magically teleport to any screen of the game. Hold down the number keys shown in the codes below to jump straight to that screen. It’s not entirely above board, but I won’t tell anyone if you don’t.

	ORIGINAL SET	CODE	NEW SET
01	The Central Cavern	6	The Bee Side
02	The Cold Room	16	Tomorrow’s World
03	The Menagerie	26	A Brief Encounter
04	Abandoned Uranium Workings	126	The Full Monty
05	Eugene’s Lair	36	Willy’s Tool and Pipe Works
06	Processing Plant	136	The Horrors of Henhouse Harry
07	The Vat	236	The Ice Cream Vat
08	Miner Willy Meets the Kong Beast	1236	Clash of the ZX Titans
09	Wacky Amoebatrons	46	Central Control System
10	The Endorian Forest	146	The Wine Cellar
11	Attack of the Mutant Telephones	246	The Milky Mars Bar Way
12	Return of the Alien Kong Beast	1246	ZX Titans: The Rematch
13	Ore Refinery	346	Darkness Falls Across the Land
14	Skylab Landing Bay	1346	Doctor Who?
15	The Bank	2346	Leap of Faith
16	The Sixteenth Cavern	12346	A Right Palaver
17	The Warehouse	56	The Potting Shed
18	Amoebatrons Revenge	156	The Souvenir Shop
19	Skylab Landing Bay	256	No, Mr Willy, I Expect You to Die
20	The Final Cavern	1256	Sunset on Willy Street

Quick! Your best friend is coming up the stairs! You don’t want him to catch you cheating, you want him to think you made it past ‘The Warehouse’ all by yourself. Press 9 and 0 together to rid the screen of that tell-tale boot and switch the cheat off again.

Unfortunately, it’s always said that cheats never prosper, and in this case that most certainly is true. If you use the cheat at any point at all during the game - even if you just activate it to see the boot then immediately deactivate it again, Big Brother is watching you. You won’t see the real happy ending on completing the game. Instead you will be judged on your dubious morals and be treated to a ‘bog-wash’ into the bargain.

Just to end on a childish note...
Here's a picture of my Willy!

