

All Five Games (SMAS+SMW) ACE

Introduction

This is an any% speedrun setup for all five games in Super Mario All-Stars + Super Mario World. The SMB3 ACE glitch was discovered by [Mally](#) and additional research for the credits warp, including the highly-optimized SMB3 payload was done by [MrCheeze](#). This route requires two multitaps, four controllers, and some way of holding down buttons on those controllers.

The main route contained in this document involves three separate sets of red shell spits. However, with additional clamping and/or controller swapping, the entire route can be done much faster with only a single set of red shell spits. For information on those routing changes, see the section on [No Shellcode ACEs](#).

For the slower but more accessible single-controller route, see [this document](#).

Setup

- USA Super Mario All-Stars + Super Mario World Cartridge
- Two multitaps in 5P mode, each with two controllers plugged into the first two slots. (For alternate controller configurations see [No Shellcode ACEs](#))
 - Port 1 Controller 1: Gameplay controller
 - Port 2 Controller 1: E0 00 = A + X + L
 - Port 1 Controller 2: 20 E4 = L + B + Y + Select + Down
 - Port 2 Controller 2: 00 60 = Y + Select
- If you're using the Bizhawk emulator, you must menu **Config** → **Cores** → **SNES** → **BSNES** to select the BSNES core, then menu **SNES** → **Controller Configuration** menu to enable multitaps in both ports. Then, you can use this chart to figure out which controller is which:
 - Player 1 = Port 1 Controller 1
 - Player 2 = Port 1 Controller 2
 - Player 5 = Port 2 Controller 1
 - Player 6 = Port 2 Controller 2

Instructions

1. Start a new game of SMB3 on save file C.
2. Without doing anything else, press **start** then **Save and Quit**.
3. Start a new game of SMW.
4. Complete YI2, getting Yoshi and passing through the midway checkpoint to become big.
5. Complete YI3, getting fire power.

Powerup Incrementation

6. Perform [Powerup Incrementation](#) twice in Y13. (For faster first-pass Powerup Incrementation strategies, see [this doc](#))

To perform powerup incrementation:

- a. **High jump** off the brown platform here.



- b. Tap **A** to jump off of Yoshi about here.



- c. Turn around midair above the parakoopa and go **left** to land on it. Mario must have leftward velocity or the koopa will fall the wrong way. It should fall to the left to land on the ?-blocks.



- d. Land on the koopa again and grab its shell.



- e. Stand here, aligned with the edge of the Yoshi block.



- f. Tap **jump**, and on your way down at this height, throw the shell **up** to dupe the Yoshi block. If you hit the block without duping it, get on Yoshi, **start+select** and try again from the beginning. If you miss the block altogether, just try again.



- g. Quickly re-press **X** or **Y**. You'll want to time it so that your fireball doesn't hit the shell.
- h. Once you've regrabbed the shell, throw it **up** to spawn a second Yoshi.



- i. Turn to the **right** before re-pressing **X** or **Y** so that your fireball doesn't destroy the red shell.
- j. Run a few pixels **right** before performing a high **jump** to get on the Yoshi. That little bit of velocity before jumping will increase your jump height enough to be able to mount Yoshi directly from the jump.



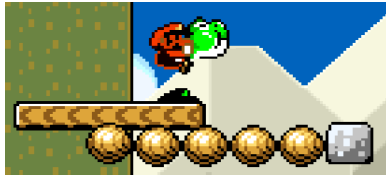
- k. If you land on an invisible Yoshi, quickly tap **B** then **A** to dismount and remount the visible Yoshi.
- l. Run to the **right** off the block.
- m. Turn around to face **left**, toward the red shell on the ground.
- n. **Grab** the shell in Yoshi's mouth.
- o. **Jump** to the **right** and **dismount** Yoshi so that he falls into the pit.



- p. **High jump** so that you can land straight on the invisible Yoshi on top of the duped Yoshi block.
 - i. It won't look like you're on a Yoshi, but you'll hear the Yoshi mounting sound, and Mario will get stalled out in midair for a second.
- q. Once Mario's feet are on the ground, tap **right** and **jump**. this will avoid the invisible Yoshi becoming visible while you're on the ground, making you clip into the ground.
- r. Go **left** and **jump** onto the brown platform.
- s. Stand, facing **right**, at this pixel. You can **pause** to help identify whether you're on the right pixel or not.



- t.
- u. When the platform is as far left as it'll go, and the chain holding it is pointed directly left, press **X/Y** to spit the null sprite and perform powerup incrementation.



- i. You can be off by about 4 frames early or late, creating a ~9-frame window for the spit. If you spit outside the window, the game will likely crash.
- 7. After the second powerup incrementation, ditch Yoshi and **start+select**.

RAM Manipulation

- 8. Enter YI2 without Yoshi.
- 9. Jump on the leftmost koopa on the ledge and grab its shell
 - a. You can use the shell from the ground at the beginning of the level to clear out as many koopas as you want besides the target koopa.



- 10. Hit the Yoshi block to spawn Yoshi.
- 11. Stand at this pixel next to the Yoshi block



12. Hold **down** and let go of **run** to drop the shell.
13. Hold **R** and let go of all other buttons to scroll the screen all the way right.
14. **Run** left 17 pixels to this location (holding **left** for exactly 12 frames will usually accomplish this).
 - a. If you run too far/too short, it's not a problem as long as the screen scroll doesn't change, just get into this position while holding the shell.



15. Throw the shell **up** and immediately **jump** onto Yoshi and move left so that you don't pick it back up or stomp it. This will create a white splat that sets RAM address \$02F4 to \$18.
16. Get Yoshi to this pixel, facing left



17. **Screen scroll** all the way right.

18. Press **start+select**. This will despawn the shell on the ground to set RAM address \$03D8 to \$4C.

Powerup \$15 ACE

19. Re-enter Y12 with Yoshi.

Setting up the sprite x-coordinate table with Arbitrary Code: The sprite x-coordinate table has 12 sprite slots, numbered 0 through B, two of which (A and B) are set aside for special sprites like the goal tape and glitched berries. When a new sprite is loaded, it picks the highest available sprite slot to occupy. For example, when you enter the level with Yoshi, first the red shell on the ground occupies slot 9, then Yoshi occupies slot 8.

The spits are done starting at slot 0 (seen at address \$E4) and iteratively going up. To do this, you need to make sure all of the remaining koopas/shells are loaded, and that you grab whatever sprite was the last to load (since it has the lowest slot number). You can always move away to unload all of the koopas/shells, and reload them all, to be sure that you're grabbing the lowest available sprite slot. If you can keep extra koopas from despawning, sometimes you don't need to reload them all to do the next spit, as long as you're able to grab the next-most recently spawned koopa in the next-lowest sprite slot.

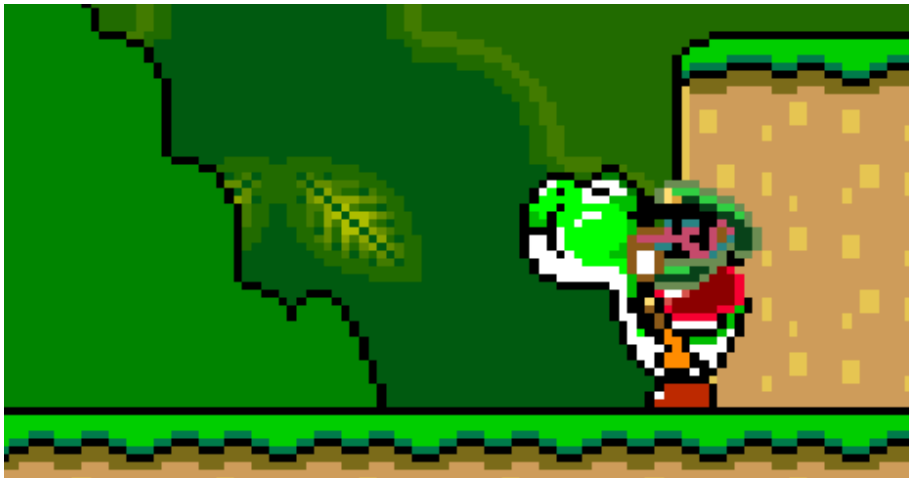
The sprite's final x-coordinate when spitting it out is the location where the fireballs spawn in front of Yoshi's mouth. If Yoshi faces right, the x-coordinate will be 32 pixels higher than if he faces left. Only the low byte of the sprite's x-coordinate matters, so every 256-pixel interval is the same for our purposes.

20. Without killing any other koopas/shells, grab the rightmost koopa in line in Yoshi's mouth.

21. Stand here and spit so that address \$E4 becomes \$84.



22. Stand here and pick up the red shell in Yoshi's mouth and spit it out. Address \$E5 becomes \$19.

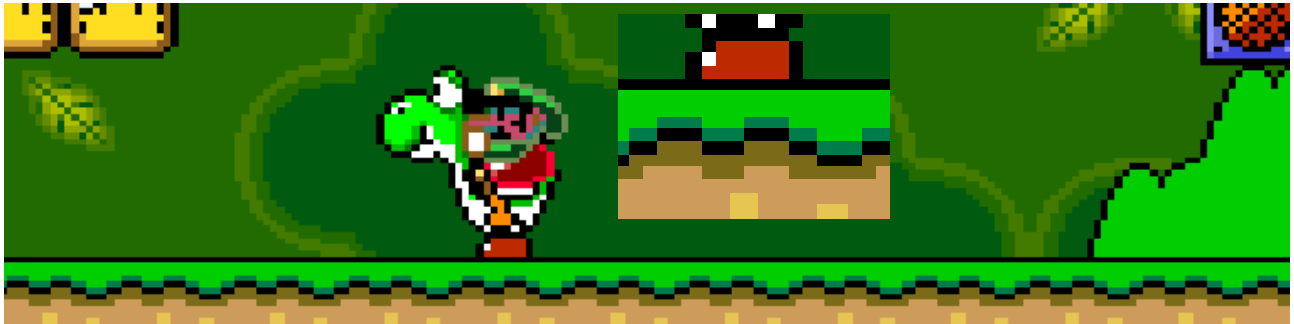


23. If the leftmost walking koopa is still on-screen, grab it in Yoshi's mouth. Otherwise, go a bit left (to despawn any remaining koopas) and run right and grab the rightmost walking koopa.

24. **Spit** here. Address \$E6 becomes \$60.



25. Run **right** and position Yoshi facing left at this pixel.



26. **Dismount** Yoshi and stand at this pixel under the Yoshi block, facing either direction.



27. **Screen scroll** all the way **left**.

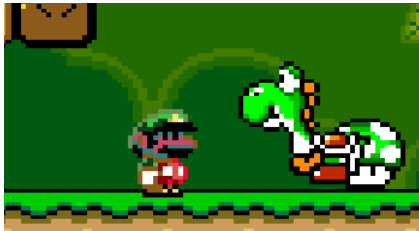
28. Hold **down** and **run**, then hold **left**, then hold **B**. Don't let go of any of those buttons until you come to a stop. This will set up the screen scroll so that the top right Yoshi egg fragment will despawn on a certain animation frame, with a specific vertical velocity. When it does, \$1829=\$18 and \$182A=\$42.



29. As the top right eggshell fragment is about to despawn, start running **right**.



30. About a tile before you touch Yoshi, hold **B** to perform a tall jump. Your goal is to hit the left side of the message box to halt Mario's rightward progress.



31. As you approach the message box, let go of all buttons and hold the two shoulder buttons **L** and **R**.

32. When the mushroom touches you, the first ACE will happen. If Yoshi is positioned correctly, \$3D0=\$05 and \$3D4=\$0F. These can be off by 1 if the camera over-scrolled by a pixel when you ran into the message box, but that should be okay. This will set Mario's powerup state at address \$19 to \$15.



33. Get back on Yoshi and **start+select**.

34. *Optionally:* Remove the clamp from **X** on port 2 controller 1.

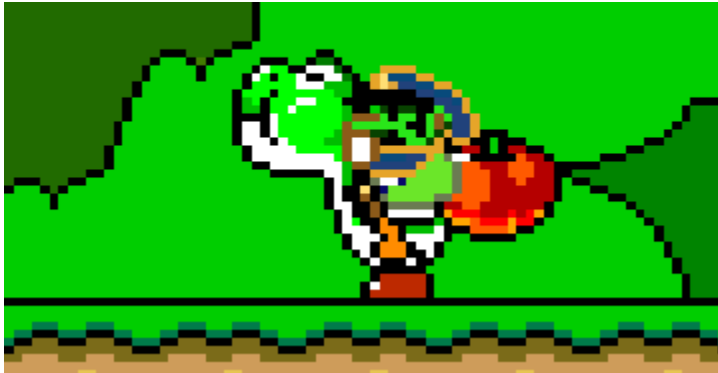
SMB3 Save Injection

35. Re-enter Y12 with Yoshi.

36. Glitch the first berry by going to the right of it, running **left**, and **tonguing** right before Yoshi would eat it.

37. Run **right** past the info box and glitch another berry.

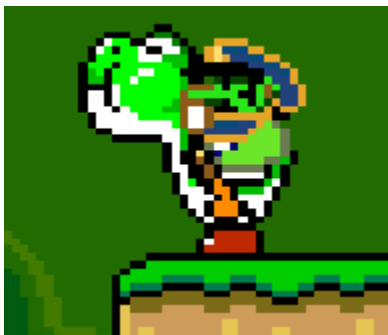
38. Run **left** and **grab** the shell on the ground. Stand here and **spit** it. Address \$E4=\$98



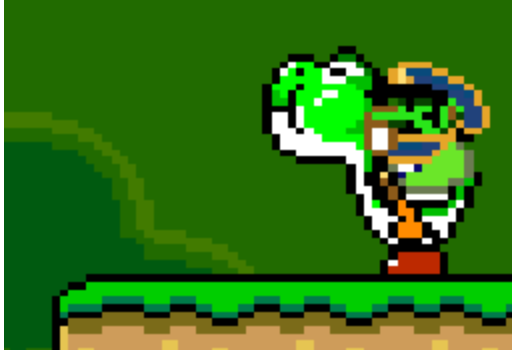
39. Go **right** and **grab** the rightmost koopa. Stand here and **spit** it. \$E5=\$8F



40. Run **right** to despawn the koopas, then run **left**, **grab** the leftmost koopa and **spit** it here. \$E6=\$26



41. **Grab** the next koopa and **spit** it here. \$E7=\$43 (\$03, \$23, and \$63 would also work)



42. **Grab** the next koopa and **spit** it around here. This spit is imprecise, and just has to fall in a 16 pixel range, so that $\$E8=\$F0-\$FF$



43. **Grab** the next koopa and **spit** it here. $\$E9=\$E6$



44. **Grab** the next koopa and **spit** it here again. $\$EA=\$E6$



45. **Grab** the next koopa and **spit** it here. $\$EB=\60



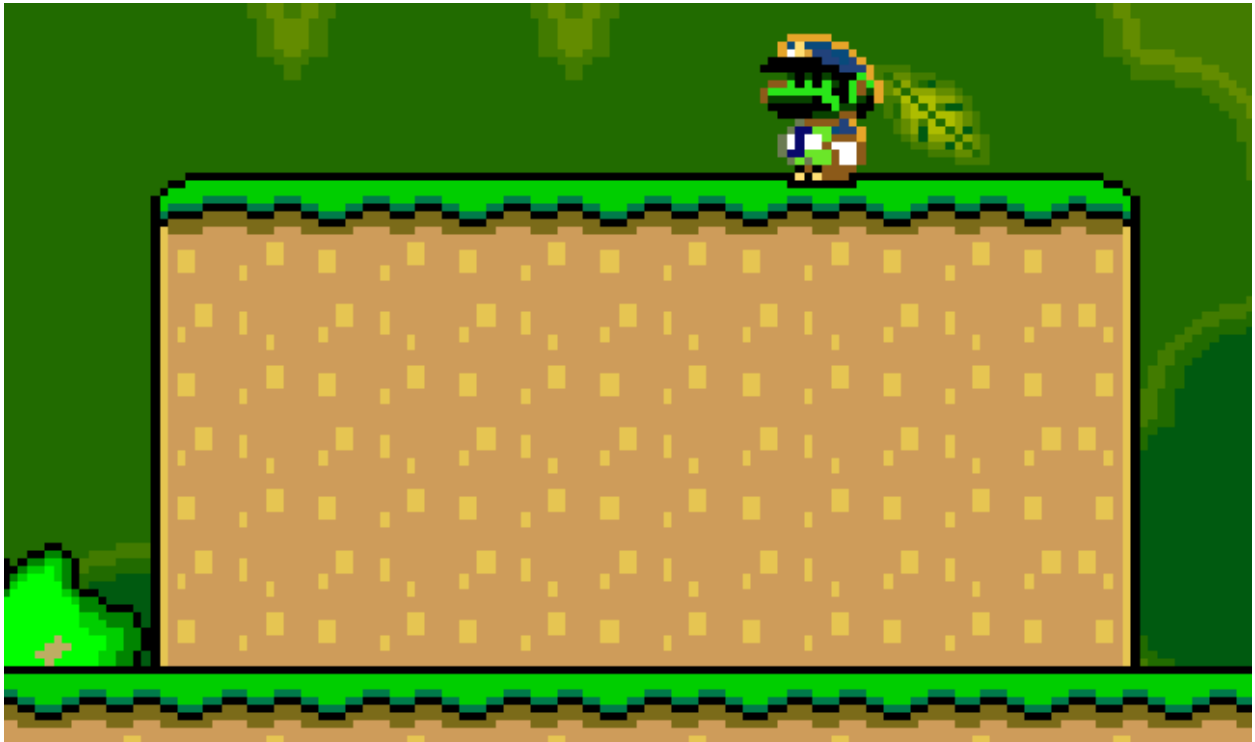
46. Place yoshi here, facing left, and **dismount**.



47. Stand here under the Yoshi block. **Scroll** the screen all the way **left**. Hold **down** and **run**, then hold **left**, then hold **B**. Don't let go of those buttons until Mario comes to a stop. \$1829=\$4218.



48. **Screen scroll right** so that Mario is relatively centered. (Don't do this too early or the eggshell fragment won't despawn correctly.)
49. Stand on this spot on the hill, with the screen scrolling left with Mario's movement.
 - a. If you need to fix the camera scroll, while facing left, **screen scroll right** then **left**.
 - b. \$300 should be \$04 and \$304 should be \$0E. If not, Yoshi is positioned incorrectly, or your camera scroll isn't correct. You can also be one pixel to the left of this.



50. Clamp **A** on your gameplay controller. If you haven't already, remove the clamp from **X** on port 2 controller 1 (but leave the clamps on **shoulder L** and **A**)

- a. Or, optionally clamp both **X** and **A** on your gameplay controller. Doing this will make it not matter whether or not you're pressing other buttons (other than the shoulder buttons) when collecting the mushroom.

51. You'll need to hold sets of buttons on the port 2 controller 1 dpad, then press **select** on the main controller.

- a. Each time the mushroom touches you, the ACE will run, and write the next byte of the SMB3 save file.
- b. You must not be pressing **select**, or any other buttons besides **A**, on the gameplay controller when the mushroom touches you, or the game will crash.
 - i. If you clamped both **X** and **A**, it's fine to be pressing **B** and/or **select** when collecting the mushroom. This can greatly speed up the \$00 inputs, since you can collect multiple mushrooms per jump safely.
- c. You must be holding the indicated set of buttons as well as **shoulder L** and **A** on controller 2 when the mushroom touches you.

Hex	Port 2 Controller 1	-Release	+Hold
55	down+right+start+Y		+down right start Y
B5	down+right+start+select+B	-Y	+B select
16	down+left+start	-right select B	+left
10	start	-down left	
20	select	-all	+select
82	left+B	-all	+left B
E5	down+right+select+Y+B	-left	+down right select Y
E0	select+Y+B	-down right	
00 (x29)		-all	
85	down+right+B		+down right B
08	up	-all	+up
09	up+right		+right
01	right	-up	
69	up+right+select+Y		+right select Y
02	left	-all	+left
04	down	-all	+down
04	down		
75	down+right+select+start+Y		+right select start Y
F2	left+select+start+Y+B	-down right	+left B
95	down+right+start+B	-left select Y	+down right
02	left	-all	+left
A0	select+B	-all	+select B
06	down+left	-all	+down left
C2	left+Y+B	-down	+Y B
20	select	-all	+select
98	up+start+B	-all	+up start B
54	down+start+Y	-up B	+down Y
70	select+start+Y	-down	+select
11	right+start	-select Y	+right

54	down+start+Y	-right	+down Y
00		-all	
00			
22	left+select		+left select
CA	up+left+Y+B	-select	+up Y B
81	right+B	-up left Y	+right
00 (x10)		-all	
F0	select+start+Y+B		+select start Y B

52. Get on Yoshi and **start+select**.

53. Remove the clamp from **A** on the gameplay controller, and add a clamp to **select** on port 2 controller 1 (leaving the clamps on **L** and **A** as well).

a. Optionally clamp **left** on port 2 controller 1 as well, for a faster credits warp.

SMW Credits Warp

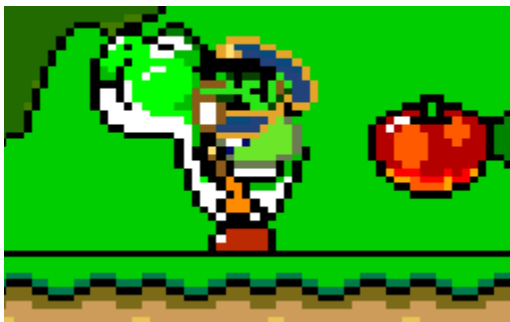
54. Re-enter YI2 with Yoshi.

55. Glitch the berry at the beginning of the level.

56. Grab the rightmost koopas and spit here. \$E4=\$98



57. Run **right** to the edge of the hill to despawn the koopas, then run **left** and grab the shell on the ground. Spit here. \$E5=\$92



58. Grab the rightmost koopas and spit here. \$E6=\$3D



59. Grab the next koopa and spit here. \$E7=\$60



60. Run **right** past the message box and glitch another berry.

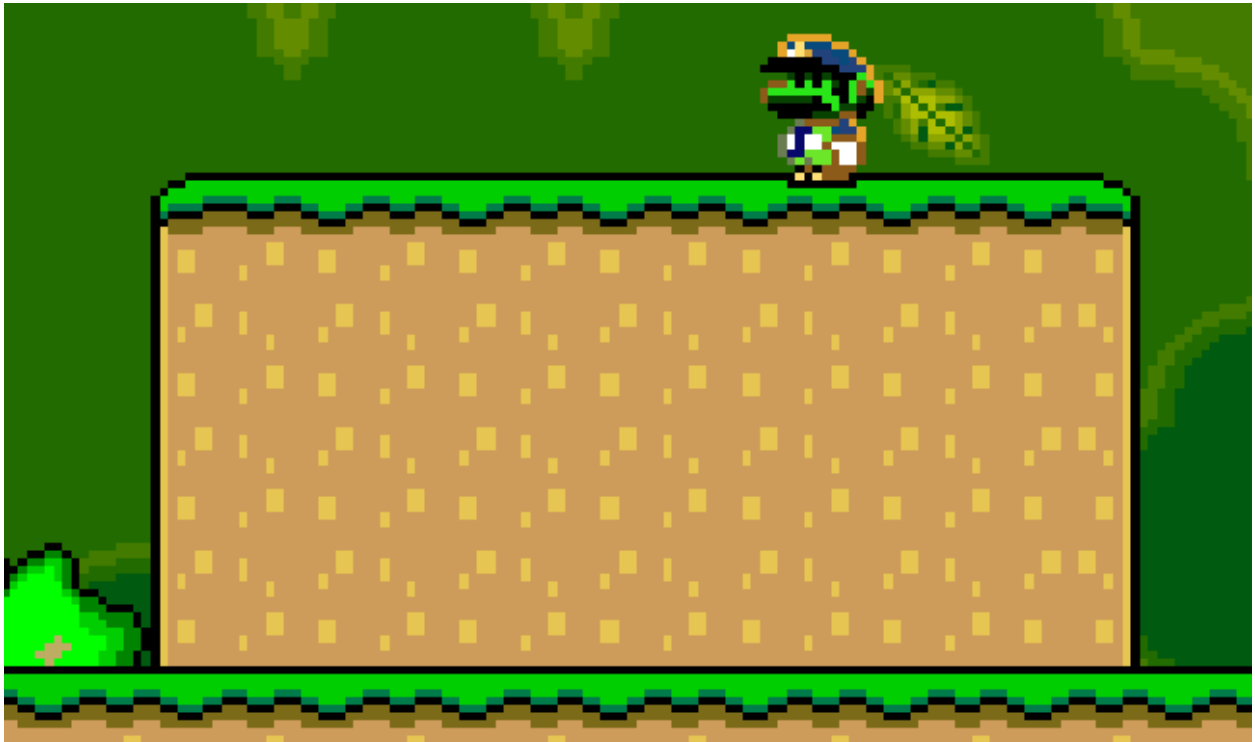
61. Place yoshi here again, facing left, and dismount.



62. Stand here under the Yoshi block again. Scroll the screen all the way left. Hold **down** and **run**, then hold **left**, then hold **B**. Don't let go of those buttons until Mario comes to a stop. \$1829=\$4218.



63. **Screen scroll right** so that Mario is relatively centered. (Don't do this too early or the eggshell fragment won't despawn correctly.)
64. Stand on this spot on the hill again, with the screen scrolling left with Mario's movement.
 - a. If you need to fix the camera scroll, **screen scroll right** then **left**.
 - b. \$300 should be \$04 and \$304 should be \$0E. If not, Yoshi is positioned incorrectly, or your camera scroll isn't correct.



65. Tap select to release the mushroom, then press and hold **A** to collect it. Make sure you aren't pressing any other buttons. Credits should play.
66. Reset

67. Remove the clamp from **select** on port 2 controller 1 (you can remove the other clamps from it, too, if you want).

The Remaining 4 Games

68. Start SMB3 save file 3.

- a. To launch SMB1 at 8-4 with debug mode, hold **down** on port 2 controller 1, press **Y** to open the item bar, and press **A** (not B!) to use the first item.
 - i. Use **select** to toggle noclip and **A** to give yourself a powerup.
 - ii. If you enable noclip while on the ground and get your height to exactly \$F0, you can skip all the pipes (and therefore the water section). Found by [WolfAeterni](#) and [ThreeCreepio](#).
- b. To launch SMB2J at D-4 with debug mode, hold **up+right** on port 2 controller 1, hold **right** on the gameplay controller, press **Y** to open the item bar, and press **A** (not B!) to use the first item.
 - i. Use **select** to toggle noclip and **A** to give yourself a powerup.
- c. To launch SMB2U at glitched level 7-6 with debug mode, hold **right** on port 2 controller 1, hold **left** on the gameplay controller, press **Y** to open the item bar, and press **A** (not B!) to use the first item.
 - i. You'll need to go through at least one door then die. The game will then put you in 7-2.
 - ii. Use **X** to give yourself star power, and **A** to toggle noclip.
 - iii. Make sure you use **X** to give yourself star power shortly before killing Wart. The game uses music ending as a trigger to allow you to enter the final doorway, but because the SMB3 sounds are still loaded, the music that will play there doesn't end. The star music does end, though, allowing you to proceed.
- d. To launch the SMB3 credits, press **Y** to open the item bar, and press **B** (not A!) to use the first item. Press **Y** to close the item bar and **enter** 1-1. The credits will then play.

Technical Overview

For a technical overview of Powerup Incrementation, see [this video](#). Powerup incrementation is used twice to reach powerup state 5.

When Mario collects a powerup, the game uses a lookup table to find a pointer to the code it should run. With Mario in powerup state 5 collecting a 1-up, or Mario in powerup state \$15 collecting either a 1-up or mushroom, the function pointer lookup is out-of-bounds, and happens to jump to \$3103D0, which is in a portion of RAM responsible for OAM (Object Attribute Memory, used for drawing sprites to the screen). OAM is divided into four byte "tiles", each with:

- X-coordinate
- Y-coordinate
- Tile number (where on the sprite sheet its graphics can be found)
- Properties (things like mirror horizontal, mirror vertical, Z-priority, color palette)

The game sets aside five OAM tiles per sprite, and the memory at \$3D0 is responsible for the first tile for sprite slot 8. By positioning Yoshi's on-screen x-coordinate, the two OAM x-coordinates for his sprite tiles will be either \$05/\$0F or \$04/\$0E, both of which allow the game to reach \$3D8, while maintaining the processor's negative flag (important for later).

At \$3D8, there's leftover tile data for a koopa shell that was despawned by using start+select, with x-coordinate \$4C. Since that OAM tile is no longer being used, the game sets its y-coordinate to \$F0 (offscreen). Its tile ID was \$8C. This makes the next instruction JMP \$8CF0, which jumps into ROM at a misaligned instruction. The code there happens to be JSR \$02F0, which will jump to \$02F0, then later execution will be returned to this part of ROM.

\$02F0 is a different part of the OAM table, responsible for drawing white "splat" animations. Because we most recently created one of those by kicking a shell up, the x-coordinates of the first two tiles are \$10 and \$18. The first instruction is BPL \$F0, which is ignored since the negative flag is set. The first tile's tile id is \$7C and its properties bits are \$20. This means the next instruction is JMP (\$1820,x).

The x register is 9 because the 1up/mushroom that Mario collected was in slot 9. \$1829 is in the middle of the minor extended sprite y-velocity table. Because Yoshi's egg shell fragments were near the right edge of the screen, the top right fragment despawned early with a y-velocity of \$18. The bottom left fragment despawned naturally with a y-velocity of \$42. The contents of \$1829 are \$4218, so execution jumps to \$4218.

\$4218 are hardware registers called the Joypad Autoread registers. They automatically read data from the first two controllers of each multitap every frame. The first byte of each controller is AXLR0000 and the second byte is BYETUDLR. These controllers have buttons held down to form code. In this run they always include JSR \$00E4. \$00E4 is the table for sprite x-coordinate low bytes. By spitting out red shells at certain spots, this table was filled with additional code.

See the [No Shellcode ACEs](#) section for alternate ACE paths that utilize the controllers rather than shells for the powerup \$15 and credits warp ACEs.

First ACE

Joypad Autoread Registers

```
BMI $00
CPX #$00
JSR $00E4
RTS
```

Sprite x-coordinate low byte table at direct page \$00E4

```
STY $19
RTS
```

Since the Y register contains \$15 at this point, this ACE stores the value \$15 to address \$19, setting Mario's powerup state to \$15.

Second ACE (Repeated for Code Injection)

Joypad Autoread Registers

```
BRA $00 / CPY #varies
LDY #varies
```

```
JSR $00E4
RTS
```

Sprite x-coordinate low byte table at direct page \$00E4

```
TYA
STA $700326
INC $E6
RTS
```

This ACE is triggered by pressing select to drop the mushroom from the item reserve box, and collecting it while holding A on the gameplay controller. Each time it's triggered, a different set of buttons are held on port 2 controller 1, and so a different LDY #const loads a different value into the Y register, which then gets written to the destination address. The destination starts out at \$700326, but each time the ACE is run, the low byte of the destination gets incremented by 1.

Bank \$70 is SRAM, the battery backed RAM used to store save files. Banks \$71-7D and \$F0-\$FF are all mirrors of bank \$70. Likewise pages \$23, \$43, \$63 (and in banks \$F0-\$FF, pages \$83, \$A3, \$C3 and \$E3) are all mirrors of page \$03 of SRAM, because there are only \$2000 bytes of SRAM on the SMAS+SMW cartridge. \$700326 is the beginning of Mario's item bar storage. The values written here are:

```
// Payload written by MrCheeze
// Preconditions: A=1D81, X=01, $02 is nonzero, $04 is zero
.db $55 // Glitched item, use to trigger payload
LDA $16,X // A = 00 01 02 80 for SMB1, SMB2j, SMB2, SMB3
BPL notsmb3 // +20
BRL $FE6D // E5 E0

notsmb3:
STA $08
ORA #$01 // 01 01 03
ADC #$02 // 03 03 05
TSB $04
ADC $F2,X // 07 0C 06
STA $02,X // store to $03
LDY #$06
REP #$20
TYA // 16 bit transfer
MVN $1170 // copy $0001-$0008 over $700007-00000E
MVN $0000 // fix data bank, and copy wram over itself in a loopy way. $08
                eventually gets copied to $D0 (game to boot)
JSL $0081CA
```

Followed eventually by the save file's checksum, \$F000.

The final SMW ACE is

Third ACE

Joypad Autoread Registers

```
BRA $00  
LDY #$20  
JSR $00E4  
RTS
```

Sprite x-coordinate low byte table at direct page \$00E4

```
TYA  
STA ($3D)  
RTS
```

Address \$3D always contains \$0100, so this stores the value \$20 to \$100, which is the address responsible for the current game mode. \$20 is a mid-credits game mode.

The SMB3 payload is accessed by using the first item in Mario's item bar. Its item ID is \$55, which is an invalid ID. When used, it runs code from an out-of-bounds pointer table lookup, and jumps to \$1D81, which happens to be the RAM responsible for Mario's item bar, starting at the second item. The payload here can perform several tasks.

By using the item with B, the payload will jump to \$FE6D, which writes to two addresses and then returns to normal gameplay. The first address, when nonzero, will complete the current level as soon as it can, and return to the overworld. The second address, when nonzero, will override loading the overworld with playing the credits. By entering 1-1, the first flag is checked and immediately the level is completed, and then instead of loading the overworld the credits are played.

By using the item with A, the d-pad can be used to control which game is loaded by the subroutine at \$0081CA. A neutral d-pad will load SMB1, holding right will load The Lost Levels, and holding left will load SMB2. The d-pad on port 2 controller 1 is read through a mirror at \$F3, and is used to control which world is loaded in the target game. The first controller d-pad also controls which level of that world is loaded in the target game, with level 4 being loaded for SMB1 and The Lost Levels (targeting 8-4 and D-4 respectively), and level 6 being loaded for SMB2 (targeting 7-6, a glitched level that can be used to quickly get to the middle of level 7-2). The payload also sets \$700007 nonzero, which enables debug mode in the target game, and \$700006 nonzero, which enables control style B.

Alternate RAM Manipulation for \$18 and \$4C

To get \$02F4=\$18 and \$03D8=\$4C, you can follow these alternate steps:

1. Enter YI2 without Yoshi.
2. Jump on the leftmost koopa on the ledge and grab its shell
 - a. You can use the shell from the ground at the beginning of the level to clear out as many koopas as you want besides the target koopa.



3. Hit the Yoshi block to spawn Yoshi.
4. Stand at this pixel under the Yoshi block



5. Hold **down** and let go of **run** to drop the shell.
6. Hold **R** and let go of all other buttons to scroll the screen all the way right.
7. **Run** left 17 pixels to this location (holding **left** for exactly 12 frames will usually accomplish this).
 - a. If you run too far/too short, it's not a problem as long as the screen scroll doesn't change, just get into this position while holding the shell.



8. Throw the shell **up** and immediately run **right** so that you don't pick it back up. This will create a white splat that sets RAM address \$2F4 to \$18.
9. Get on Yoshi
10. Jump **left** to the hill beside these blocks.
11. Jump back **right** onto the blocks, letting go of **run** but holding **right** to slow your speed down as you approach (so that you don't overshoot them). This will reset the camera scroll. Yoshi's head should be under the item box now.
12. Come to a stop on this pixel without overshooting
 - a. If you overshoot, get back to this pixel then **screen scroll** all the way **left**, then **screen scroll right** so that you're centered again. This will reset the screen scroll.



13. Press **start+select**. This will despawn the shell on the ground to set RAM address \$3D8 to \$4C.

No Shellcode ACEs

If you perform the no-shellcode ACE for both Powerup \$15 and Credits Warp, you'll only need to perform one set of shell spits, and won't need to re-enter YI2 between any of the ACEs. You can also use a spare item reserve box mushroom to fill a sprite slot rather than glitching a second berry. This route can be seen [here](#).

No-Shellcode Option for Powerup 15 ACE

During the first ACE, by setting up the controllers at the beginning of the run, and holding:

- Port 1 Controller 1: 20 04 = **L + Down**
- Port 2 Controller 1: A0 94 = **A + L + Start + B + Down**
- Port 1 Controller 2: 10 E4 = **R + B + Y + Select + Down**
- Port 2 Controller 2: 00 60 = **Y + Select**

The joypad autoread registers contain the code:

```
JSR $A004  
STY $10,X  
CPX $00  
RTS
```

JSR \$A004 will immediately return, doing nothing of any importance. Then, since X=\$9 and Y=\$15, STY \$10,X will set Mario's powerup state at address \$19 to value \$15. CPX \$00 will do nothing, and then RTS will return control back to the game code. This requires minimal reclamping for the SMB3 save file injection, just removing **start+B+down** on port 2 controller 1, and reclamping the shoulder from **R** to **L** on port 1 controller 2.

Skipping the shell code also allows for performing the first ACE without re-entering YI2:

<https://youtu.be/f3VfnXX7qhk>. Alternatively, you can setup as normal, glitching two berries before performing this ACE, then continue to perform the spits and ACEs for the SMB3 code injection. This will avoid the need to re-enter YI2 for the SMB3 ACE, and redo the shell fragment despawn.

No-Shellcode Option for Credits Warp ACE

- Port 1 Controller 1: 20 28 = **Up + Select + L**
- Port 2 Controller 1: 90 A0 = **R + A + B + Select**
- Port 1 Controller 2: 40 91 = **X + B + Start + Right**
- Port 2 Controller 2: 20 60 = **L + Y + Select**

The joypad autoread registers contain the code:

```
JSR $9028
LDY #$40
STA ($20),Y
RTS
```

JSR \$9028 will load direct page \$64 into the accumulator, then push A, do some graphics stuff, and pull A. Since \$64 contains the value \$20 in YI2, A will get the value \$20. Then, we set Y to \$40. Address \$20 contains \$00C0, so STA (\$20),Y will write \$20 to \$0100 and then RTS will return control back to the game code. On the next frame the game will be in gamemode \$20, which is credits.

After finishing the SMB3 save file injection, there is no need to re-enter YI2 or move at all. Either reclamp or swap controllers to match the above configuration. Then, hold **up**, then **shoulder L**, then **select** on the main controller, and wait for the mushroom to touch Mario (making sure not to scroll the screen left).

Full Clamp Flow

Starting configuration: [JSR A004; STY \$10,X; CPX \$00; RTS]

- Controller 2: **A + L + Start + B + Down**
- Controller 3: **R + B + Y + Select + Down**
- Controller 4: **Y + Select**

After the Powerup 15 ACE, before code injection: [CPY/CPX #imm; LDY #imm; JSR \$00E4; RTS]

- Controller 1: Add clamp to **X + A**
- Controller 2: Remove **Start + B + Down**
- Controller 3: Swap the shoulder clamp from **R** to **L**

After finishing code injection, before credits warp: [JSR \$9028; LDY #\$40; STA(\$20),Y; RTS]

- Controller 1: Remove clamp from **X + A**
- Controller 2: Swap to **R + A + B + Select**
- Controller 3: Swap to **X + B + Start + Right**
- Controller 4: Clamp **L**

To reset:

- Controller 2: Swap back to **A + L**, add **Start + B + Down**
- Controller 3: Swap back to **L + B + Y + Select + Down**, swap shoulder clamp from **L** to **R**
- Controller 4: Remove clamp from **L**

Four Players

PI15 ACE: [JSR A004; STY \$10,X; RTS]

- Controller 1: **L + Down**
- Controller 2: **A + L + Start + B + Down**
- Controller 3: **R + Y + Select**
- Controller 4: *None*

Code injection: [CPY #imm; LDY #imm; JSR \$00E4; RTS]

- Controller 1: **X + A + Select + B**
- Controller 2: **A + L + varies**
- Controller 3: **L + B + Y + Select + Down**
- Controller 4: **Y + Select**

Credits warp: [JSR \$9028; LDY #\$40; STA(\$20),Y; RTS]

- Controller 1: **L + Up + Select**
- Controller 2: **R + A + B + Select**
- Controller 3: **X + B + Start + Right**
- Controller 4: **L + Y + Select**

Save File Injection Destination Reset

In case you want to restart the save file injection process, these two ACEs can reset the injection destination address contained at \$00E6 to \$700320. If you want to reset it additional times, only the second ACE is necessary. You'll need to write \$00 six times to get the write destination back to the beginning of the payload.

Store \$E6 to direct page \$30

- | | | |
|-------------|-------|---|
| • LDY #\$E6 | A0 E6 | A + L + B + Y + Select + Down + Left |
| • BRA #\$01 | 80 01 | A + Right |
| • \$00 STY | 00 84 | B + Down |
| • \$30 RTS | 30 60 | L + R + Select + Y |

Store \$20 to \$00E6 (via indirect addressing at direct page \$30)

- | | | |
|--------------|-------|--|
| • JSR \$04 | 20 04 | L + Down |
| • \$A0 LDA | A0 A9 | A + L + B + Select + Up + Right |
| • #\$20 STA | 20 92 | L + B + Start + Left |
| • (\$30) RTS | 30 60 | L + R + Select + Y |