This is the old code repository. The content of this document has been re-organized and moved here:
https://sites.google.com/site/sm64gameshark/

This document will not be maintained.

Harper Code™

Unless otherwise stated, all codes and explanations in this document were created by Harper. The "Coin count reset," "Star select fix for sublevels," and Addendum were made by Hellool. Level select codes were made by Deftek.

Level Reset Code (USA) (by harper)

(L button to reset) (US)

```
D033AFA1 0020      <- Condition to activate reset level code
8033B248 0002      <- Reset level code  (put 0003 to not destroy elevators, ttc will freeze)
8029CEC0 0000      <- Awesome code respawns most things
8029CE9C 0000      <- Respawns the rest of the shit
```

Restore Health on Reset (L Button) (US)

```
D033AFA1 0020      <- Condition to restore health
8033B21E 0008      <- Health
```

Star Select (US)

Note: (You need **L to reset level** *enabled*, but optionally you may disable the "respawns most things" code if you need space such as if you have no expansion pak)

```
D033AFA1 0030      <- Star Select Screen (Press R then press L, then release R while holding L)
8033B239 0004      <- (This does not escape sub levels, but resets everything without the
```

other two lines labeled "respawns most things". You don't need to disable those codes but you can if you need to. Takes longer than the other reset, but resets wdw water level, camera issues, mobs and even allows selecting a different star)

Star Select fix for sublevels (US) (by hellool & abitalive)

Note: (This does not replace Star Select code. You need Star Select and Level Reset code *enabled)*

(16bit version, for GameShark V3.0+)

```
D033AFA1 0030
80370000 0030
D233AFA1 0030
80370000 0000
D033B249 000D
80370001 000D
D233B249 000D
80370001 0000
D1370000 300D
8133B24A 020A
D1370000 3000
8133B24A 010A
```

(8bit version, for GameShark V2.x)

For everything EXCEPT THI

```
D033AFA1 0030
8033B24A 0001
D033AFA1 0030
8033B24B 000A
```

For THI (do NOT use this version for anything else, it will behave unexpectedly and may crash your game)

```
D033AFA1 0030
8033B24A 0002
D033AFA1 0030
8033B24B 000A
```

Coin Count Reset (US) (by hellool)

(16bit version, for GameShark V3.0+)

```
D033AFA1 0020
8133B262 0000
D033AFA1 0020
8133B218 0000
```

(8bit version, for GameShark V2.x)

```
D033AFA1 0020
8033B262 0000
D033AFA1 0020
8033B263 0000
D033AFA1 0020
8033B218 0000
D033AFA1 0020
8033B219 0000
```

Level Reset Code (JP) (by harper)

(L button to reset) (JP)

```
D0339C31 0020      <- Condition to activate reset level code
80339ED8 0002      <- Reset level code  (put 0003 to not destroy elevators, ttc will freeze)
8029C740 0000      <- Awesome code respawns most things
8029C71C 0000      <- Respawns the rest of the shit
```

Restore Health on Reset (L Button) (JP)

```
D0339C31 0020      <- Condition to restore health
80339EAE 0008      <- Health
```

Star Select (JP)

Note: (You need **L to reset level** *enabled*, but optionally you may disable the “respawns most things” code if you need space such as if you have no expansion pak)

```
D0339C31 0030      <- Star Select Screen (Press R then press L, then release R while holding L)
80339EC9 0004      <- (This does not escape sub levels, but resets everything without the
```

other two lines labeled “respawns most things”. You don’t need to disable those codes but you can if you need to. Takes longer than the first, but resets wdw water level, camera issues, mobs and even allows selecting a different star)

Star Select fix for sublevels (JP) (by hellool & abitalive)

Note: (This does not replace Star Select code. You need Star Select and Level Reset code *enabled)*

(16bit version, for GameShark V3.0+)

```
D0339C31 0030
80370000 0030
D2339C31 0030
80370000 0000
D0339ED9 000D
80370001 000D
D2339ED9 000D
80370001 0000
D1370000 300D
81339EDA 020A
D1370000 3000
81339EDA 010A
```

(8bit version, for GameShark V2.0+)

For everything EXCEPT THI

```
D0339C31 0030
80339EDA 0001
D0339C31 0030
```

```
80339EDB 000A
```

For THI (do NOT use this version for anything else, it will behave unexpectedly and may crash your game)

```
D0339C31 0030
80339EDA 0002
D0339C31 0030
80339EDB 000A
```

Coin Count Reset (JP) (by hellool)

(16bit version, for GameShark V3.0+)

```
D0339C31 0020
81339EF2 0000
D0339C31 0020
81339EA8 0000
```

(8bit version, for GameShark V2.x)

```
D0339C31 0020
80339EF2 0000
D0339C31 0020
80339EF3 0000
D0339C31 0020
80339EA8 0000
D0339C31 0020
80339EA9 0000
```

Level Select Codes

This code loads any of the 15 main levels and the 3 bowser levels.
(16bit, for Gameshark v3.0+)
Note: (Must have level reset codes and star select codes *enabled* with the sublevels fixed)

do input for the level you want, then go to star select with the star select code
(e.g. to go to hmc hold dpadright then hit CRIGHT then go to level select)

PRESS BOTH — Insert from Left to Right ->

JP

Button 1	Button 2	Stage	First Line	Second Line
DPADUP	CUP CRIGHT CDOWN CLEFT	BOB WF JRB CCM	D1339C30 0808 D1339C30 0801 D1339C30 0804 D1339C30 0802	80339ED9 0009 80339ED9 0018 80339ED9 000C 80339ED9 0005

DPADRIGHT	CUP CRIGHT CDOWN CLEFT	BBH HMC LLL SSL	D1339C30 0108 D1339C30 0101 D1339C30 0104 D1339C30 0102	80339ED9 0004 80339ED9 0007 80339ED9 0016 80339ED9 0008
DPADDOWN	CUP CRIGHT CDOWN CLEFT	DDD SL WDW TTM	D1339C30 0408 D1339C30 0401 D1339C30 0404 D1339C30 0402	80339ED9 0017 80339ED9 000A 80339ED9 000B 80339ED9 0024
DPADLEFT	CUP CRIGHT CDOWN CLEFT	THI TTC RR	D1339C30 0208 D1339C30 0201 D1339C30 0204 D1339C30 0202	80339ED9 000D 80339ED9 000E 80339ED9 000F
DPADLEFT DPADUP DPADRIGHT		BitDW BitFS BitS	D1339C30 0200 D1339C30 0800 D1339C30 0100	80339ED9 0011 80339ED9 0013 80339ED9 0015

PRESS BOTH

Insert from Left to Right ->

US

Button 1	Button 2	Stage	First Line	Second Line
DPADUP	CUP CRIGHT CDOWN CLEFT	BOB WF JRB CCM	D133AFA0 0808 D133AFA0 0801 D133AFA0 0804 D133AFA0 0802	8033B249 0009 8033B249 0018 8033B249 000C 8033B249 0005
DPADRIGHT	CUP CRIGHT CDOWN CLEFT	BBH HMC LLL SSL	D133AFA0 0108 D133AFA0 0101 D133AFA0 0104 D133AFA0 0102	8033B249 0004 8033B249 0007 8033B249 0016 8033B249 0008
DPADDOWN	CUP CRIGHT CDOWN CLEFT	DDD SL WDW TTM	D133AFA0 0408 D133AFA0 0401 D133AFA0 0404 D133AFA0 0402	8033B249 0017 8033B249 000A 8033B249 000B 8033B249 0024
DPADLEFT	CUP CRIGHT CDOWN CLEFT	THI TTC RR	D133AFA0 0208 D133AFA0 0201 D133AFA0 0204 D133AFA0 0202	8033B249 000D 8033B249 000E 8033B249 000F
DPADLEFT DPADUP DPADRIGHT		BitDW BitFS BitS	D133AFA0 0200 D133AFA0 0800 D133AFA0 0100	8033B249 0011 8033B249 0013 8033B249 0015

Death Codes

8bit (for **V2.x**) Death code (For gamesharks below 3.0) (Press R and L to activate)
`D033AFA1 0030`
`8133B21E 00FF`

or

`D033AFA0 0004` (dpad down to activate)
`8133B21E 00FF`

The reason gamesharks below 3.0 cannot use the other death code is because they do not support codes starting with D1,D2 and D3. Anything on the dpad is a 16 bit value. You need a D1 code to let the dpad 16 bit value activate the code. So anything activated with the Dpad will not work on a gameshark below 3.0, Unless you change the conditional to 33AFA0 (-1) and delete the last two digits. So dpad down goes from 0400 to 0004 (further explanation in addendum).

This means if you find a code with D1 but have a gameshark that is below 3.0 and the code does not work. Replace the code with D0 but it will only read the last two digits in the value part of the code. 16 bit will read all four digits, 8 bit will only read the last two digits. So find a button combo that does not take 3-4 digits, find one with two digits using this button value graph.

XXYY 8 Bit(D0 CodeType) will read YY
XXYY 16 bit(D1 CodeType) will read XX and YY

So choose a button with two digits instead of three and replace D1 with D0.

```
BUTTON_C_RIGHT = 0x0001,
BUTTON_C_LEFT  = 0x0002,
BUTTON_C_DOWN  = 0x0004,
BUTTON_C_UP    = 0x0008,
BUTTON_R       = 0x0010,
BUTTON_L       = 0x0020,
BUTTON_D_RIGHT = 0x0100,
BUTTON_D_LEFT  = 0x0200,
BUTTON_D_DOWN  = 0x0400,
BUTTON_D_UP    = 0x0800,
BUTTON_START   = 0x1000,
BUTTON_Z       = 0x2000,
BUTTON_B       = 0x4000,
BUTTON_A       = 0x8000
```
(copied from kazes gameshark doc, thank you kaze)

JP - US Conversion

For most codes, to convert between the US code and JP you must add or subtract 0x1370 (that is, 1370 in hex), from the address. To do this, open up calculator, set to programmer mode, set to hex mode. Then enter the address (Which is the middle 6 digits of the code) and either add 1370 to go from JP to US or subtract 1370 to go from US to JP. Example:
D033AFA1 0020 (US) → D0339C31 0020 (JP). (33AFA1 - 1370 = 339C31 in hex)

There seem to be some codes where this offset is incorrect, but it works for almost all of them.

VC

For now Until I update this VC page go here.http://marioruns.com/forum/viewtopic.php?f=20&t=281#p2304
Just copy and paste the VC codes inside your cheats .txt on the SDCard and use your favorite cheat manager to create a .gct with it.

Old Codes

USA

Reset Camera. (Dpad up before resetting the level. Do not stand next to a door, or just reset twice.)

```
D033AFA0 0008      <-Condition to reset camera (Dpad up)
8033C6D4 0010      <-Resets Camera, 03 is flight cam, 01 is most levels default.
```

JP

Reset Camera. (Dpad up before resetting the level. Do not stand next to a door, or just reset twice.)

```
D0339C30 0008      <-Condition to reset camera (Dpad up)
8033B364 0010      <-Resets Camera, 03 is flight cam, 01 is most levels default.
```

Notes

To practice Vanish Cap
You also may optionally put 0003 instead of 0002 for the value for the reset level code it will not destroy elevators and you can practice Vanish cap BUT TTC will freeze. The code will destroy elevators when set to 0002.(default)

Reset Camera Code issues
You need to reset the camera in BBH away from a door because the door holds a trigger that forces the camera mode. Or you can reset, then reset the camera and reset again to be centered like normal.

Swimming is a constant trigger that forces the camera so you need to reset the level, reset the camera and then reset the level again if you want it perfect or just reset the level and then reset the camera but it won't be perfect just like you entered the level.

The camera gets reset to JRB default value of 10. BBHs camera is actually 04 but it gets forced to 04 right before you enter the front door. So youll notice the camera jerk down slightly when entering the front door of BBH. The JRB camera seems to be the best one to reset to though. Most levels reset it to 01, but BBH allows the JRB camera. So youll notice the jerk when you enter the mansion.

If youd like it to be more accurate. You may add the BBH camera in a different Dpad then the JRB one.

(This code is not necessary, just use the default one when in doubt)

Example:
`D033AFA0 0008` <-Condition to reset camera (Dpad up)
`8033C6D4 0010` <- JRB camera
`D033AFA0 0004` <-Condition to reset camera (Dpad down)
`8033C6D4 0004` <-BBH Camera
`D033AFA0 0002` <-Condition to reset camera (Dpad left)

TTC Time
You'll probably want to change the buttons for these.
US
`D033AFA0 0008` (dpad up)
`80361259 0003` (time stopped)
`D033AFA0 0001` (dpad right)
`80361259 0000` (time slow)
`D033AFA0 0004` (dpad down)
`80361259 0002` (time random)
`D033AFA0 0002` (dpad left)
`80361259 0001` (time fast)

JP
`D0339C30 0008` (dpad up)
`8035FEE9 0003` (time stopped)
`D0339C30 0001` (dpad right)
`8035FEE9 0000` (time slow)
`D0339C30 0004` (dpad down)
`8035FEE9 0002` (time random)
`D0339C30 0002` (dpad left)
`8035FEE9 0001` (time fast)

Old Deftek Code JP
`D1339C30 0209` (dpad left+c right+c up)
`8035FEE9 0001` (time fast)
`D1339C30 0205` (dpad left+c right+c down)
`8035FEE9 0003` (time stopped)

00 = Slow (3 O'Clock)
01 = Fast (9 O'Clock)
02 = Random (6 O'Clock)
03 = Stopped (12 O'Clock)
04 = Very Slow

WDW Water Level
You'll probably want to change the buttons for these.
US
`D033AFA0 0004` (dpad down)
`80330F3C 44AB` (waterlevel low)
`D033AFA0 0002` (dpad left)

80330F3C 44BB (waterlevel mid)
D033AFA0 0008 (dpad up)
80330F3C 44CB (waterlevel high)

JP
D0339C30 0004 (dpad down)
8132FFDC 44AB (waterlevel low)
D0339C30 0002 (dpad left)
8132FFDC 44BB (waterlevel mid)
D0339C30 0008 (dpad up)
8132FFDC 44CB (waterlevel high)

00000000-44ACCCCC = Low
44ACCCCD-44C7FFFF = Mid
44C80000-7F800000 = High
7F800001-FFFFFFFF = Low

wdw water levels are affected by the height you jump into a painting, and also by the height of the painting in the castle. so you could jump into other paintings and affect the water height in wdw

Close JRB Cannon
This code will close the cannon in JRB, but also mark the 7 stars in CCM as collected in the process. Use the code that matches the save slot you're playing. Press Dpad Right+A at any time to close the cannon, and then level reset (if you're in JRB). GS 3.0+ required.

US
File A
D133AFA0 8200 (dpad left+a)
8020770F 007F (write to flag)

File B
D133AFA0 8200 (dpad left+a)
8020777F 007F (write to flag)

File C
D133AFA0 8200 (dpad left+a)
802077EF 007F (write to flag)

File D
D133AFA0 8200 (dpad left+a)
8020785F 007F (write to flag)

JP
File A
D1339C30 8200 (dpad left+a)
80207B0F 007F (write to flag)

File B
D1339C30 8200 (dpad left+a)
80207B7F 007F (write to flag)

File C
`D1339C30 8200` (dpad left+a)
`80207BEF 007F` (write to flag)

File D
`D1339C30 8200` (dpad left+a)
`80207C5F 007F` (write to flag)

Other
for THI, you could map dpaddown for tiny island by setting subzone to 2
hit dpaddown while you are in star select
use R+L to set the subzone back to 1
loading levels without a subzone 2 will freeze the game

Addendum - Explanation of codes

Hey guys, I decided to add this to the code document because I've gotten quite a few questions from people asking how people find/make these codes or how certain ones work. I will not attempt to offer a specific explanation for the reset and star select screen codes since I don't understand them 100% enough to confidently explain their workings.

- General GS code guide

(Skip past if you only care about coin count reset or sublevel fix codes)
People have this idea in their head that GS codes are some arcane, complex topic that only the most advanced computer geeks can figure out. The truth is they are quite simple as long as you have a basic understanding of how memory and addresses work. Any 12 digit gameshark code line is broken into three parts: the prefix, the address, and the value (all in hexadecimal A=10, B=11, C=12, D=13, E=14. F=15 for those who don't know). If you've ever played around with a hex editor like cheatengine (perhaps to cheat at an old flash game? heh), a GS is basically that in hardware form. Lets break down a couple of lines. Take the restore health code (US):
(credit to Harper [I think])
D0 / 33AFA1 / 0020 <- Condition to restore health
80 / 33B21E / 0008 <- Health

We'll look at each line individually, as they provide examples of the two types of codes appearing in this document. The first line is a trigger line, which is specified by the prefix D0 or D1.
First Line: D0 / 33AFA1 / 0020 ← You'll see this exact line many times in this document.
Prefix: D0. This tells the GS hardware to continuously check the address for the given value, and when and only when it detects that value, execute the line immediately following.
Address: 33AFA1. This hex address may be the most used in all of SM64 GS code creation, as it is the address of the game's current button input (with a caveat, see below with 8/16 bit).
Value: 0020. The hex value of 20 (0x20, or 32 in decimal) is what the GS checks for at the above address. In this case, 20 in hex corresponds to the L button being pressed. So when you press L with this code enabled, the GS knows to execute whatever the line below it is.

Second Line: 80 / 33B21E / 0008
Prefix: 80. The prefix 80 tells the GS to continuously (every frame?) set the value at the address to the given value. This is how any "infinite x" code works or any code that constantly modifies a

game value such as the invisible mario code. If we put this line without the trigger line right above it, we'd be giving mario infinite health. Instead it only happens when L is being pressed.
Address: 33B21E. This address is where mario's health is stored.
Value: 0008. You set the health to 8 (full). Pretty simple to understand.

difference between 16 and 8 bit codes. Each address in memory only stores an 8 bit value, which means only a number from 0-255 (0000 - 00FF) can be stored there. There are some cases, like for using d-pad down as an input trigger, where this is not sufficient, because the game actually uses a 16 bit value. Gamesharks before 3.0 could not directly change a 16 bit value, so you had to work around it. GS 3.0 and beyond can, but it is still common practice to avoid making 16bit only codes, that way more people can use them.

Lets look at the 16 bit death code for an explanation. (taken from siglemic.net resources)
D1 / 33AFA0 / 0400
80 / 33B21E / 0000

The second line is the same as it would be on an older GS. 80 prefix means modify the address. 33B21E is mario's health (same as above) which is set to 0 for an instant kill. It's the first line that makes this code unusable on GS 2.x. First understand that the input value in SM64 is actually two consecutive 8 bit addresses, 33AFA0 and 33AFA1. Each can store 00-FF, so together they can store 0000-FFFF, a 16 bit value.

When you only want to use the L button as input, you use a value of 0020. Since the first two digits are unused, you can basically ignore the first "half" of the address and just change the second half, 33AFA1, to 20, which is why it works on an older GS even though it's technically a 16 bit value. If you want to use d-pad down, which has an input code of 0400, there are two ways of doing it. First you can use a 16 bit modification if you have a V3.x GS. The D1 prefix is the same as D0 except it tells the GS to look at the current line AND the next one (address + 1) and check for a 16 bit value. That way we can check for an input value of 0400.

The second way is pretty clever. If you look at the value 0400 you notice the last two digits are unused. So we can actually just change the *first* part of the address to 0004 using a D0 (8 bit check). That's where we get this 8 bit version that does the exact same thing.
D033AFA0 0004
8133B21E 00FF.

Anyway if you want to know more just google gameshark code making guide or something. Here's an explanation of some of the codes on this document:

- Coin Count Reset Explanation

The most common question I get about this code is, why doesn't it work with <insert code here> code? To that I don't have a solid answer. For some people it's lack of Expansion Pack, but I'm not entirely sure why some code interactions fail and others don't. The GS is a pretty crappy piece of hardware. Anyway, here's the code (US, 16 bit)

D033AFA1 0020
8133B262 0000
D033AFA1 0020
8133B218 0000

The first thing you notice is there are two identical lines. We now know the reason is because they both just check for an L input. There is no way of assigning multiple lines to a single L button check, so we have to do it this way (in pairs). The two lines that actually *do* something are:
8133B262 0000
8133B218 0000

These are both 16 bit (0000-FFFF) values, even though I don't see why they need to be, since you can only have up to 255 coins. . . Now when some people tried to make this code originally their mistake was trying to only change one of these two values, which interact closely with each other. (I may have gotten these addresses swapped, its been awhile)
The first address, 33B262, stores your actual coin count at any given frame. I'll call this your ticking coin count.
The second address, 33B218, stores the amount of coins you *should* have eventually. I'll call this your coin max.
The reason why this matters is that you don't accumulate coins instantly. When you pick up a blue coin, your coin max increases by 5 instantly, but your ticking coin count does not. Instead, it ticks up by 1 coin each frame until it reaches the new coin max, which is certainly noticeable in game if you pay attention. Trying to set only the ticking coin count to 0 and not the coin max results in the coin count going to 0 but then immediately continuing to increase back to the old value. Kind of funny but useless. Trying to set the coin max to 0 does nothing, since your ticking coin count at that point is probably equal to your coin max (unless you reset it while your coins are counting up). Setting both to 0 simultaneously proved to be the solution.

The 8-bit version of this code works by just splitting every pair of values into two pairs, one for each half of the 16 bit value. That's why there are 8 lines instead of 4. This is a common technique for converting codes to be useable on GS 2.x.

- Star Select Screen Sublevel Fix Explanation

OK, this one is pretty simple. When you use the star select screen code, you're changing the state of the game to that screen. Doing so resets many of the values in the level to the default, but not the sublevel value which tells the game what part of the level to spawn you in (because the game is designed so that re-entry takes you back to the last sublevel you were in).

Using a debug code, I went around and checked what sublevel value the game had for each level. I found that in all of them except THI, the value was 0001 for the normal starting area, and 0002 for a sublevel if it existed (I did not check every sublevel, but I did check THI, TTM, SL, SSL, LLL). In THI, the huge world was 01 but you want to start in the tiny world which was 02. THI also had a problem with sublevel entry points, so I set that to the default value of 0A in both cases just to be safe. Here's the normal code (US, 8-bit)

D033AFA1 0030
8033B24A 0001 ← only change for THI is to change 0001 to 0002
D033AFA1 0030
8033B24B 000A

Again the first and third lines are identical and just check for an input value of 0030 which corresponds to R+L being pressed simultaneously.
33B24A is the address for the sublevel code which we set to 01 (or 02 for THI).
33B24B is the sublevel entry point code which we set to 0A which is the initial entry point. If you don't change this, the game will use the last point of entry into the level (messes you up for

practicing THI100 for example). I couldn't find another example of where this code is necessary, but it's a good idea to include it anyway just in case, unless your GS can't handle too many codes at once. In that case you can remove the second pair of lines from the Non-THI code.

Well that's all I have to say, thanks for reading this drawn out explanation and good luck on your practice!
Regards,
-Hellool