

# Super Mario Sandbox Manual

Version 0.4

## Contents

About the game .....	1
Disclaimer.....	1
Controls.....	2
General controls .....	2
Player controls .....	2
Editor controls.....	2
Editor instructions.....	3
General.....	3
Block/Item menu .....	3
Block/Item subtype effect.....	4
Pipe warping .....	5
Upcoming features.....	5
Known issues.....	5
Credits .....	6
Libraries credits.....	6

## About the game

- This game was created by MightySlurm <https://mightyslurm.itch.io/>
- Visit <https://mightyslurm.itch.io/super-mario-sandbox>

## Disclaimer

Super Mario Sandbox is a fan game and is not affiliated or endorsed by Nintendo. All rights to Nintendo Characters and themes are held by Nintendo, please show you support of Nintendo by purchasing Nintendo Games and Game systems.

## Controls

Default controls are not customizable yet, sorry, but they will be in future updates :)

### *General controls*

Reset level	R
Quit game	ESC
Quit without saving	ESC + SHIFT
Toggle pause	P
Toggle fullscreen mode	F11
Toggle camera mode	V

### *Player controls*

Reset level	R
Run/grab/fireball	X
Jump	C/Y/Z/Spacebar
Up/Down/Right/Left	Directional arrows

### *Editor controls*

<b>General</b>	
Toggle editor	E
Move camera	W/A/S/D
Change zoom level	CTRL + Scroll
Change level theme	L
Save level	F1
Save level with name	F2
Load level	F3
Move player to cursor position	M
Change player powerup status	T
Set cursor as new starting position	SHIFT + M
Generate new random level	CTRL + R
Generate new empty flat level	CTRL + SHIFT + R
<b>Level block/item edit</b>	
Set block at cursor position	Left Mouse Click / Touch Tap
Erase block at cursor position	Right Mouse Click / CTRL + Left Mouse Click / CTRL + Touch Drag
Undo change	CTRL + Z
Redo change	Can't do it yet sorry...
Select level section	SHIFT + Left Mouse Click + Drag Cursor/ SHIFT + Touch Drag
Delete selected section	DELETE
Copy selected section	CTRL + C

Paste copied section	CTRL + V
Fill selected section with current block/item	Left Mouse Click on selected section
<b>Special editor controls</b>	
If pipe item is currently selected, rotate pipe by 90 degrees counterclockwise	ENTER
Increase current subtype by 1	F9
Decrease subtype by 1	F8
Increase subtype by 32	F7
Decrease subtype by 32	F6
Block/Item selection shortcuts	Try number keys, then combine with SHIFT and CTRL :)

## Editor instructions

Unfinished properties are marked with “(!)”

### *General*

- You can control the player while in edit mode, allowing you to test the feasibility of the jumps
- The editor supports touch input, alongside mouse and trackpad input
- Example levels can be found in the folder “Data/Worlds”
- there can be infinite checkpoints, after the player dies he starts from the last touched checkpoint. However, there can be only one flagpole per level.
- It is not possible yet to fit multiple subworlds in one level, this feature will be available in a future update
- some example levels have background blocks (mushroom stems, etc). While it is possible to select them with shift+1,2,3,4,5, the current system is deprecated. I'm working on a better system to deal with background blocks, it will be available in a future update

### *Block/Item menu*

The block/item menu consists of a menu bar with shortcuts and a drop-down menu with all the existing blocks and items.

You can change a shortcut item with a right click on a wanted block/item from the drop-down menu, followed by a right click on the shortcut item in the menu bar you want to change.

The block/items are organized in three categories: **blocks**, **enemies** and **objects**, as it is seen in the drop-down menu (“items” stands for “enemies and objects”). Each block/item has a **type** attribute, which defines *which* block/item it is. Additionally, each has also a **subtype**, is a value between 0 and 127 with 2 additional *modes*: 1 and 2. The subtype is displayed as a number close to the cursor position with a gray color when the subtype is in mode 1, and a red color when it's in mode 2. This value defines some other properties of the item/block, which are detailed in the next chapter.

## Block/Item subtype effect

The subtype value and mode has different effect depending on the block/item.

For **coin/breakable blocks**:

Subtype value is <b>0</b>	nothing is contained in the block
Subtype value between <b>1</b> and <b>31</b>	the value is the number of coins contained in the block
Subtype value between <b>32</b> and <b>63</b> (!)	the block contains a powerup. Value=0 means an <i>incremental</i> powerup, values 1 and 2 are respectively a mushroom and a fire flower. Currently the other values do nothing
Subtype value between <b>64</b> and <b>95</b> (!)	The block contains an object, look at the screen to see which :)
Subtype value between <b>69</b> and <b>127</b> (!)	The block contains an enemy, look at the screen to see which
In addition to the subtype value, if the subtype is in <b>mode 2</b> (displayed red), the block is a <b>hidden block</b> : it's invisible until the player hits it with the head	

For **all other blocks**:

The subtype **mode** should always be 1 (and not be set to 2), else the block will be an invisible block... which doesn't make sense. Thus for the rest of the block types the mode will be ignored

For **conveyor belts**:

Subtype value between <b>0</b> and <b>5</b>	the value controls the direction and speed of the belt, try it out
For all other values it has an undefined behavior (thus avoid it, really)	

For **pipes**

Subtype value is <b>0</b>	The pipe is composed of normal solid blocks and it is <b>not</b> a warping pipe
Subtype value is between <b>1</b> and <b>63</b>	The pipe is a <b>warping pipe</b> and the subtype value denotes its <b>address</b>
All other subtype values and the mode 2 have not defined behavior and should not be used	

For **enemies**

Subtype mode is <b>1</b> (thus displayed gray)	Enemy is spawned <i>once</i> at the item position
Subtype mode is <b>2</b> (thus displayed red)	An enemy is periodically spawned at the item position
The subtype value has no effect	

For **all other items**

- The subtype value has no effect

### *Pipe warping*

Pipe warping is controlled by the **address** of the pipe, which is defined by the subtype (see chapter above). Non-warping pipes have no address.

Warping pipes are *entered* by staying in contact with its entrance and by pressing the directional arrow corresponding to the direction of the entrance.

- If there is **no** other pipe in the level with the **same** address, the pipe behaves exactly as a non-warping pipe (thus the player doesn't warp)
- In the other case, the player is warped to the *left-most other* pipe with the same address. There can thus be multiple warping pipes with the same address in one level, and their behavior is still well defined

### Upcoming features

- **Multiplatform support:** currently the source code relies on the win32 API for the window management, that's why this game is currently only supported on windows. I plan to switch to SDL2 in future updates and support also Linux and MacOS.
- **Online level sharing:** I'm planning to develop an online level sharing service, so that everyone can share their own levels and play and rate the levels from all the other players.
- **More features:**
  - New themes/tilesets
  - More enemies and objects/items
  - Possibility to fit multiple subworlds in one level and warp between them
- **Create a whole game:** it will be possible to create a whole game made out of your levels, not only single levels

### Known issues

- In fullscreen mode, game works best on a 16:9 display, otherwise there will be some stretching
- Expect random crashes sometimes... this is still only a beta version:) (tip: save your level regularly)
- Sometimes after the player dies, the items in a question mark block disappear right after spawning
- If there are too many bomb explosions there might be crashes
- Sometimes a bomb explosion causes the generation of HUGE particles. This is kinda fun :)

- There is a glitch in the collision detection with spike blocks. Try to find it and exploit it ;)
- There is a glitch when detecting whether the player or the enemy has stomped the other, very similar to the original Super Mario Bros. Sometimes it helps the player, other times not so much...

## Credits

- MightySlurm, for all the source code and many edits/additions to music, sound effects and sprites
- Nintendo, for the original Super Mario Bros. and Super Mario Maker music, sound effects and sprites. And well, also for the creation of the Super Mario franchise, all its games and their gameplay/mechanics that we all love
- All rights to Nintendo Characters and themes are held by Nintendo, please show you support of Nintendo by purchasing Nintendo Games and Game systems.

## Libraries credits

The game was written entirely in C++

The code relies on the following libraries:

- Microsoft Windows's Win32 API for the window management
- OpenGL for rendering
- GLEW (OpenGL Extension Wrangler Library)
- SOIL (Simple OpenGL Image Library) for loading image files
- OpenAL for audio
- freealut for loading audio files