RAINBOWZ EDITOR MANUAL

TABLE OF CONTENTS

Contents

INTRODUCTION 6

QUICK START GUIDE 6

- LOADING YOUR ROM 6
- TILE EDITING 6
  - REGULAR 6
  - STAMP 6
  - HIGHLIGHT 8
- CAMERA EDITING 8
  - MOUSE EDITING 8
  - TEXT EDITING 8
- ENTITY EDIT 9
- ENTRANCE EDITING 9
- BANANA EDITING 9
- PLATFORM EDITING 9

DETAILED HOW-TO 10

- TILE EDITING (IN-DEPTH) 10
  - TRAIL 10
  - AUTOFILL 12
- CAMERA EDITING (IN-DEPTH) 12
- ENTITY EDIT (IN-DEPTH) 13
  - BUTTONS 14
  - COMMON ENTITIES 15
  - NEW ENTITIES 15
- ENTRANCE EDITING (IN-DEPTH) 17
- BANANA EDITING (IN-DEPTH) 17
- PLATFORM EDITING (IN DEPTH) 18

FILE 21

- LOAD 21
- LOAD (RESTORE FROM BACKUP – END) 21
- LOAD (RESTORE FROM BACKUP – START) 21

SAVE 21
SAVE AS ROM 21
RESTORE BACKUP 21
RENAME VERSION 21
IMPORT 21
EXPORT 22
MOD (IGNORE THIS) 23
EMU 23
SELECT EMU PATH 23
LAUNCH (F7) 23
SELECT HEX EDITOR PATH 24
LAUNCH HEX 24
EDIT 24
TILEMAP 24
ENTITY INIT EDITOR 24
PALETTE EDIT 25
SNAP 25
TRANSPARENT 25
LEVEL 25
CLEAR TILEMAP 25
CLEAR CAMERA MAP (HORIZONTAL) 25
CLEAR DEFAULT ENTITIES 25
CLEAR BANANA MAP 25
ORDER ENTITIES 25
SET END OF LEVEL 25
MISC 25
ASM MODS 25
OVERWORLD EDIT 26
PERCENTAGE 26
COPYRIGHT EDIT 27
TEXT EDITOR 27
REVERSE SQUAWKS 27
QUICK FAKE CREDITS 27

LEVEL ATTRIBUTES 27
CUSTOMLEVEL ATTRIBUTES 27
MUSIC PICKER 27
STARTING KONG 28
WORLD TERMINATOR 28
KONG ATTRIBUTES 28
SAVE AFTER EACH LEVEL 28
EXTERNAL 28
OBJECT SCRIPT LOOKUP 28
BG IMPORTER 29
SPRITE EDIT 29
MUSIC TRACK EDIT 29
ANIMATION EDIT 29
K ROOL EDIT 29
BOSS EDIT 29
BACKUPS 29
AUTO-BACKUPS 29
BACKUP - END 30
BACKUP – START 30
SAVE-BACKUPS 30
RESTORING SAVE-BACKUPS 30
RESTORING SAVE-BACKUPS MANUALLY 30
PLATINUM (*.FMF) 31
EXPORTING 31
IMPORTING 32
ENTITY EDITOR 32
PALETTE EDIT 34
TEXT EDITOR 35
LEVEL ATTRIBUTES 37
LOAD 37
LISTBOX 37
APPLY 37
APPLY SANITY 37

DONE 38
ALL &=, |=, OR ^= 38
CUSTOM ATTRIBUTES 38
CREDITS 38

INTRODUCTION

To get started, you need to use a 1.0 U version of Donkey Kong Country for the SNES. This editor backs up often, so you don't lose work. This backs up in various ways too. See the BACKUPS section for more details. That being said, it is advised that you should still keep your own backups. This is not super polished, so there are many ways you can still break things. All values are in hex unless otherwise specified.

If you are more of a visual learner, some of the features are demonstrated on this playlist.

While not required, a tip on patreon is always appreciated.

For any updates, please go here.

QUICK START GUIDE

*****NOTE *****

Vertical level can be buggy! Use caution when editing.

LOADING YOUR ROM

Open the program and load your rom. Use the regular load option and select a proper DKC ROM. Header vs headerless doesn't matter, and this will recognize most dkc1 1.0 U ROMS.

Once you load the ROM, use the dropdown to select a level, then press 'Select.'

TILE EDITING

Since this is the quick-start guide, only Regular, Stamp, and Highlight will be covered here. See the TILE EDITING (IN-DEPTH) for a full explanation of features.

REGULAR

Here, you have the option of arranging tiles in a finely tuned manner. You can click any tile to highlight it. Then you can click anywhere in the level to place. You can check the "Grid" box on the main window to see exactly how tiles line up. Further, in the tile window, you have the option to place x or y flipped tiles. This only works with tiles (any form of tile editing), but you can press ctrl+Z to undo any tile change.

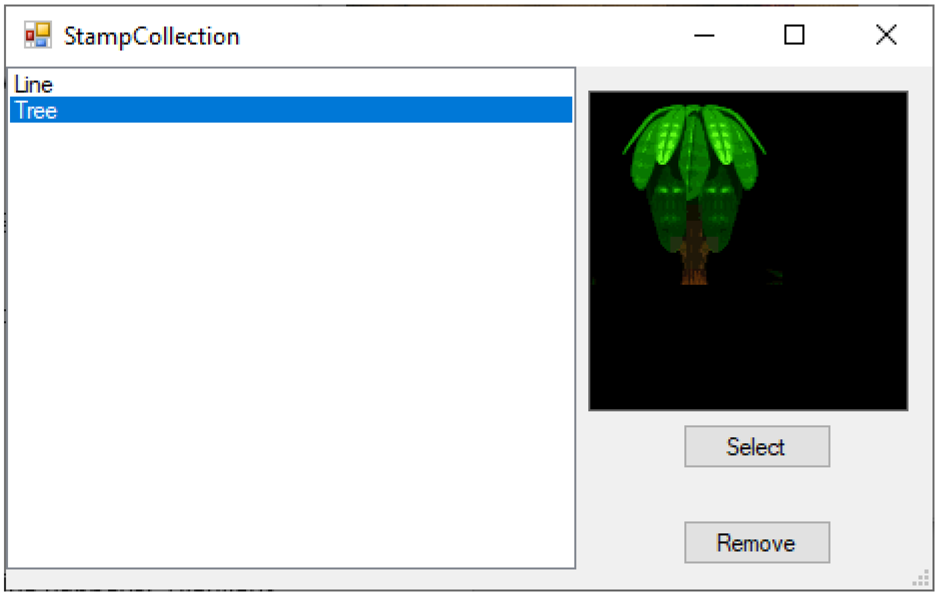
STAMP

Stamps are a really powerful feature. A 5x5 'sandbox' is presented where anything could be designed without affecting the level. To go about creating a stamp, first click any tile in the 5x5. Then click whichever tile you desire (including x and y flips). Do this for as many tiles as you'd like. An empty tile won't be written. After designing the 5x5, left click in the level to get a preview. Right click to place the tiles. There are a few buttons in Stamp to talk about.

Clear – This just empties the canvas for you.

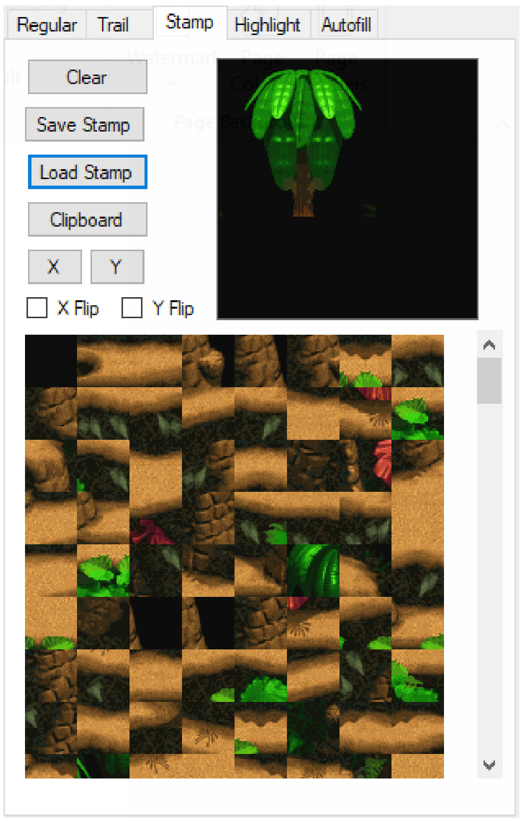
Save Stamp – This gives you a chance to save whichever stamp you want with a unique name. Careful! Repeat names will be overwritten!

Load Stamp – This will load any previously saved stamps, and each stamp has an image preview. By left clicking, you select and preview a stamp. You can either press 'Select' or right-click to choose a stamp. You also can remove stamps.



Clipboard – You also have the option to import anything you previously copied in the next section. NOTE: Only the first 5x5 will be considered!

X/Y – These buttons flip the entire stamp around over the specified axis.



HIGHLIGHT

The highlight tab is also very powerful and will likely be where most of the designing happens. 'Highlight' Is the default option. When selected, if you left click and drag in the level, a group of tiles is highlighted. After highlighting, you can press either of the 3 buttons below. Cut copies and deletes. Copying also adds the highlight to history. More on that later. By switching to 'Paste' mode, a left-click in level shows you a preview of what is on the 'clipboard'. A right-click places from the 'clipboard'. The X/Y buttons flip the whole copy for you.

'History' is a unique feature. Whenever you copy something, an index is stored on that list (if it is a unique copy!). Clicking on any index will cause a quick 2 second preview of that paste to appear. Also, 'History' will always remember the last thing you copied and will recall it whenever a new instance of the program is loaded. This was done so you can copy across programs. So, if you like a structure from a different rom, you could copy it and use it in your current mod.

CAMERA EDITING

The following sections only work if the 'Edit {Section}' is checked, which is only available if 'View {Section}' is checked. There are two ways to edit.

MOUSE EDITING

There's not much to say here. You can drag edges by holding them and moving your mouse.

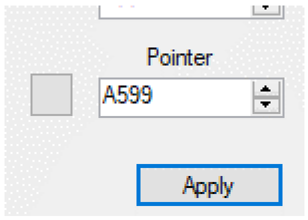
TEXT EDITING

Text editing is a bit more advanced and will be covered in CAMERA EDITING (IN-DEPTH)

ENTITY EDIT

Some confusion may arise as everyone call these sprites. That leads to confusion between the gfx and behavior, so these will be referred to as entities here. The most important thing to note is that entities should be ordered by x position for things to work well. There is a feature in this program that orders the entities on level reload. On the main form, click Level>Order Entities. This flags it so on level reload entities are ordered. That option is remembered, so on future instances that option is checked automatically. There is a slight bug with ordering, but 99% of the time the bug won't be encountered. See ENTITY EDIT (IN-DEPTH) for details.

Most entities are type 1, so those are fine. Some entities have a different type though and need that type. Gnawty Wheels and swinging ropes are examples of this. Script sets use type 5 and are only good where they are in vanilla. To learn more about this go to ENTITY EDIT (IN-DEPTH).



Pressing the button to the left here brings up a popup of every entity in the game. After navigating to the desired entity, either double-click, right-click, or press enter to select that pointer. Then be sure to hit apply or press enter to make the changes.

There are quite a few other buttons, but those can be ignored for now. Those will be discussed later.

ENTRANCE EDITING

Entrances can be dragged around with the mouse. For those pesky off-screen entrances, you can mess with values directly. This is expanded on in ENTRANCE EDITING (IN-DEPTH).

BANANA EDITING

You can move banana groups with the mouse. To change the banana group, just select one that you like from the dropdown. To change these via text, see BANANA EDITING (IN-DEPTH).

PLATFORM EDITING

This refers to the path that tracks take (like in 5-2). This option is available everywhere but vertical levels, and then only in levels that have the track entity. Basically, you can move points with the mouse. There is more to it further explained in PLATFORM EDITING (IN DEPTH).

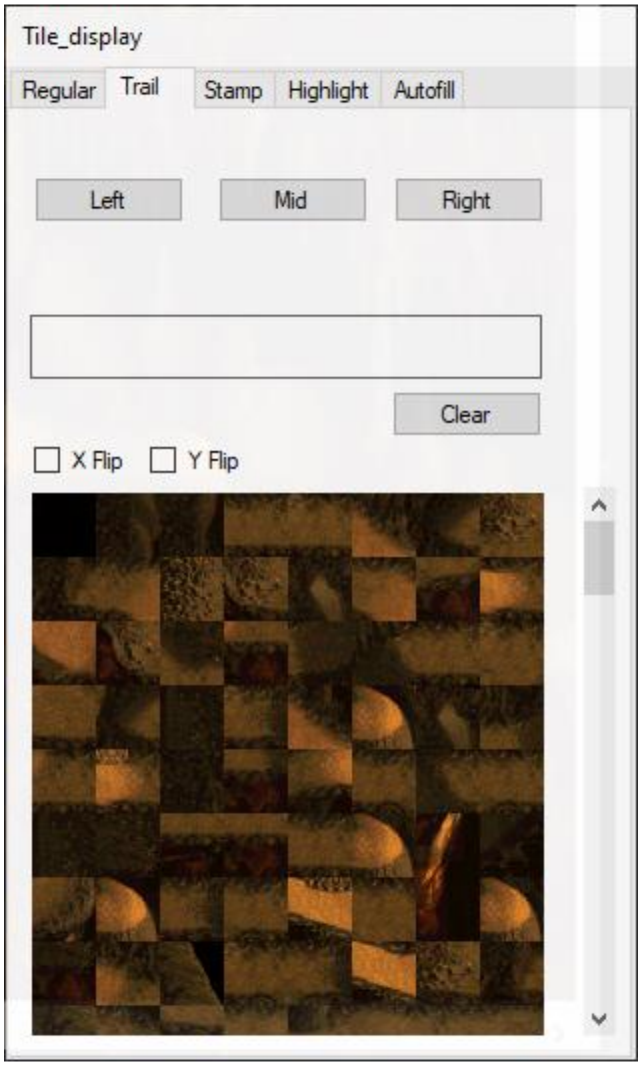
DETAILED HOW-TO

TILE EDITING (IN-DEPTH)

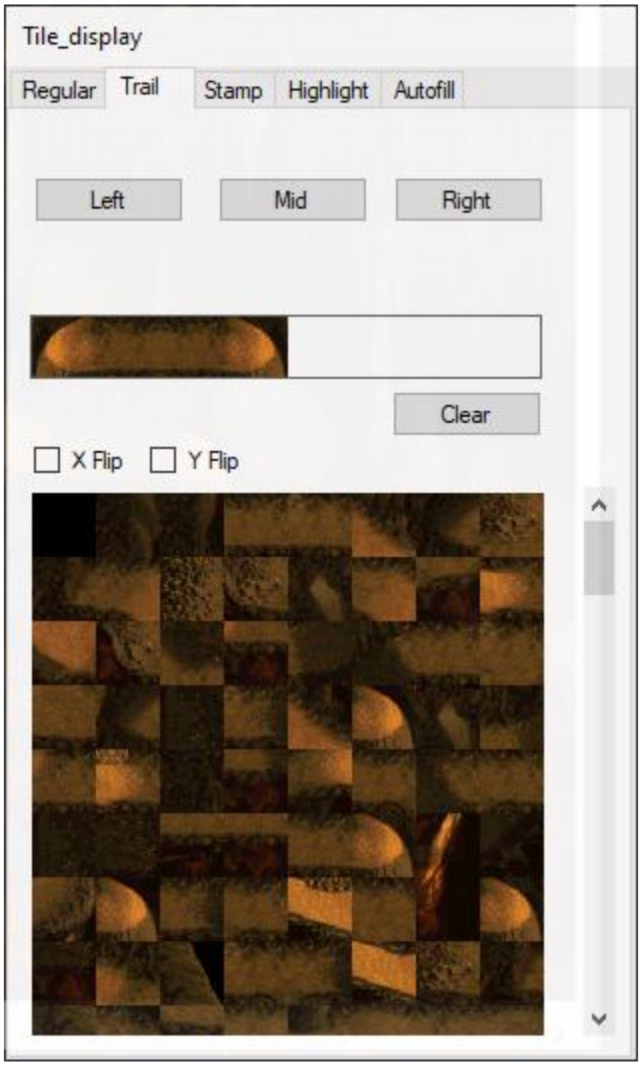
This guide is not going to repeat the info in the quick-start section. Rather, this will expand on things. This editor can be powerful if used properly. 'Export mapchip' will be discussed in the PLATINUM (*.FMF) section.

TRAIL

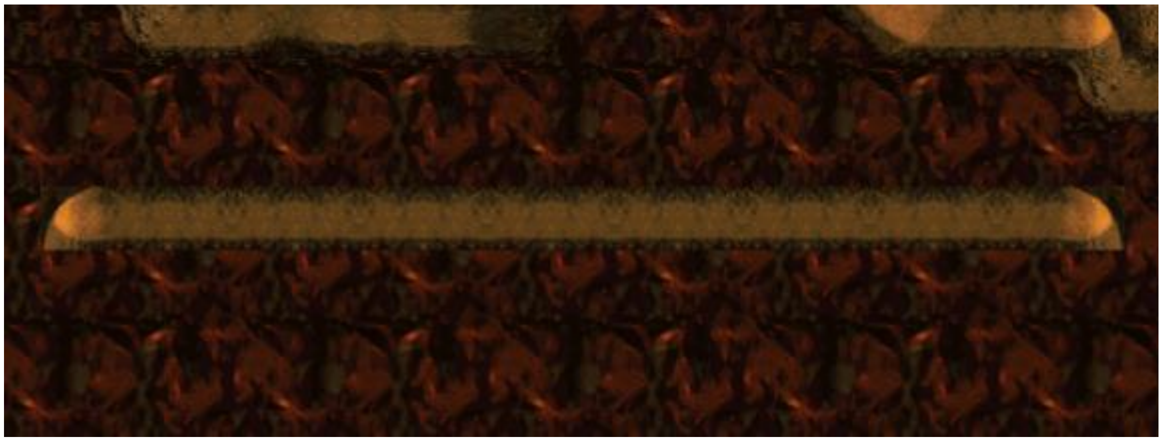
Trail is a powerful feature that was inspired from RPG Maker.



Basically, you can construct a prototype like this

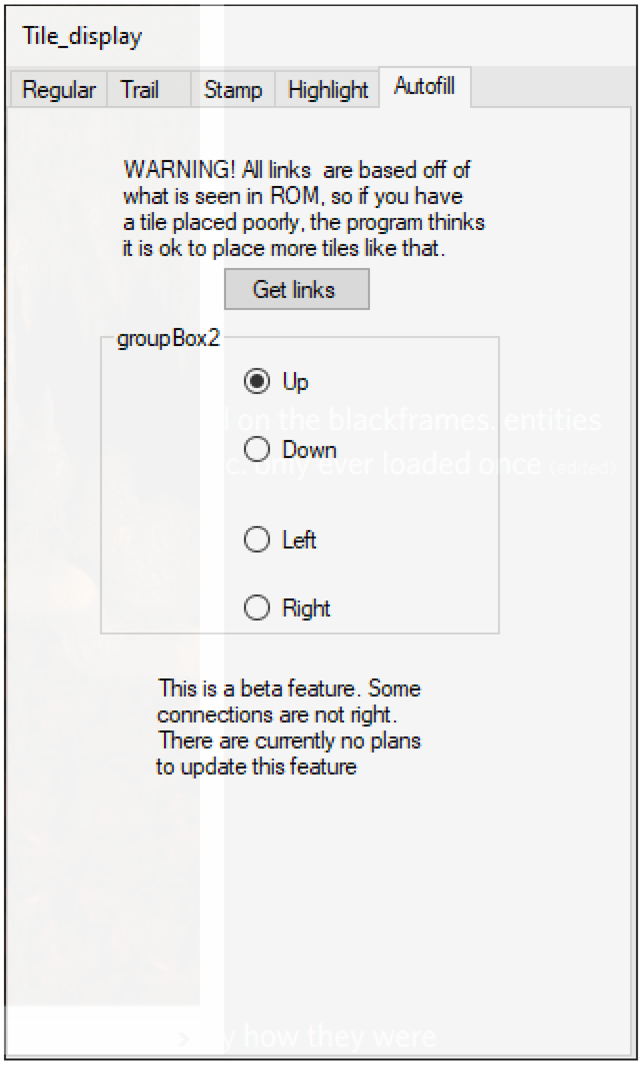


And left click and hold+drag in level to get a structure like this



AUTOFILL

Autofill is a beta feature left in with no plans to finish it. In level, click 'Get links' and choose a direction below. Clicking on a tile now in the main window will automatically place a tile for you in the direction you specified before.

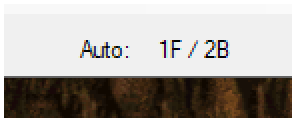


WARNING! All links are based off of what is seen in ROM, so if you have a tile placed poorly, the program thinks it is ok to place more tiles like that.

[✓ Up] Down Left Right

This is a beta feature. Some connections are not right. There are currently no plans to update this feature

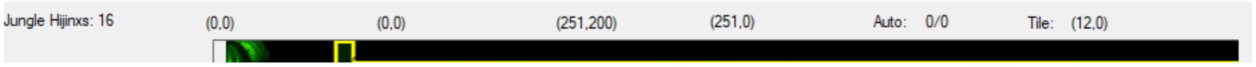
After you click, this text value is updated. The number on the right, in hex, is the total number of connections found. The number on the left, in hex, is the current index of the selected tile from an array of connections. Undo (Ctrl + Z) is great here.



Auto: 1F / 2B

CAMERA EDITING (IN-DEPTH)

Only drag and drop editing was discussed before. Basic text editing is intuitive. The camera map is composed of 6-byte long entries (3 ints). The first int is the coordinate of the left edge of the camera. The next int represents the top, and the final, the bottom. 0 on the bottom starts at the middle of the screen. If you notice, numbers go down from the top of the screen. Both bananas and cameras do this. To make positioning easier, look at the 4th set of digits along the top here.



Jungle Hijinxs: 16 (0,0) (0,0) (251,200) (251,0) Auto: 0/0 Tile: (12,0)

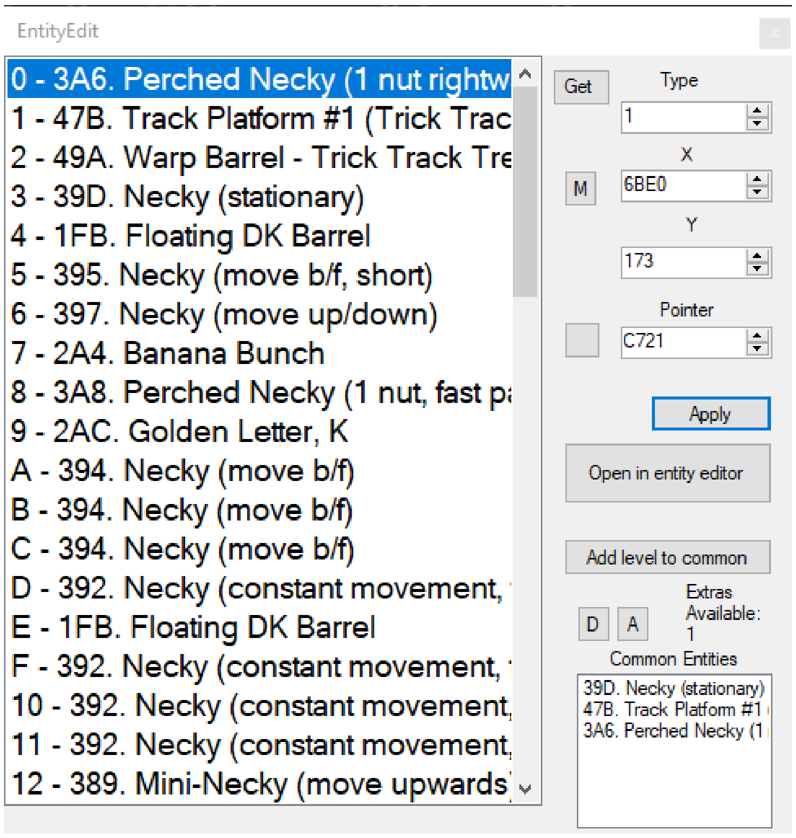
That is the coordinates from the top of the screen of the cursor. You can use this to aid in positioning camera boxes. Camera top starts where you specify and ends at the middle. Camera bottom starts at the middle and ends where you specify. The camera goes to the right until it encounters the next box.

If you want to apply multiple cameras at the same time, you can use the multiple option at the bottom. Select an option for which value you want to repeat, and then press 'Set [Start, end).' This is standard notation that says start at 'Start' inclusively and go up to but not including 'end.' This will then apply the value you specified to every camera in the range you specified.

ENTITY EDIT (IN-DEPTH)

Entity editing is huge! Vertical entity edit vs horizontal entity edit are two separate things for now. This covers all the features of horizontal entity editing. Vertical is nothing but a lite version of horizontal.

Pressing 'Enter' does the same thing as pressing apply.



EntityEdit

0 - 3A6. Perched Necky (1 nut rightw
1 - 47B. Track Platform #1 (Trick Trac
2 - 49A. Warp Barrel - Trick Track Tre
3 - 39D. Necky (stationary)
4 - 1FB. Floating DK Barrel
5 - 395. Necky (move b/f, short)
6 - 397. Necky (move up/down)
7 - 2A4. Banana Bunch
8 - 3A8. Perched Necky (1 nut, fast pa
9 - 2AC. Golden Letter, K
A - 394. Necky (move b/f)
B - 394. Necky (move b/f)
C - 394. Necky (move b/f)
D - 392. Necky (constant movement,
E - 1FB. Floating DK Barrel
F - 392. Necky (constant movement,
10 - 392. Necky (constant movement,
11 - 392. Necky (constant movement,
12 - 389. Mini-Necky (move upwards

Type: 1
X: 6BE0
Y: 173
Pointer: C721

Extras Available: 1

Common Entities
39D. Necky (stationary)
47B. Track Platform #1
3A6. Perched Necky (1

There is a lot to take in on this screen. Let me first explain how the game reads entities. Entities are composed of 8 bytes. Bytes 0 + 1 are its type (high byte is always 0), bytes 2 + 3 are the x position of the entity, bytes 4 + 5 are y, and bytes 6 + 7 are which entity it is. Look at Simion32's list of objects to see where the codes come from.

Entity type is very important! It says how to handle a specific entity. For example, a value of 0x0002 tells the game not to reserve any vram for this entity, like with exits. A type of 0x000d says to reserve an extra chunk of vram, for bigger entities such as the track platforms.

BUTTONS

GET

Gets the hex offset of the selected entity in ROM and copies to clipboard.

M

Sets the x coordinate to that of the last entity + 1 or at the very end of the entity map.

P

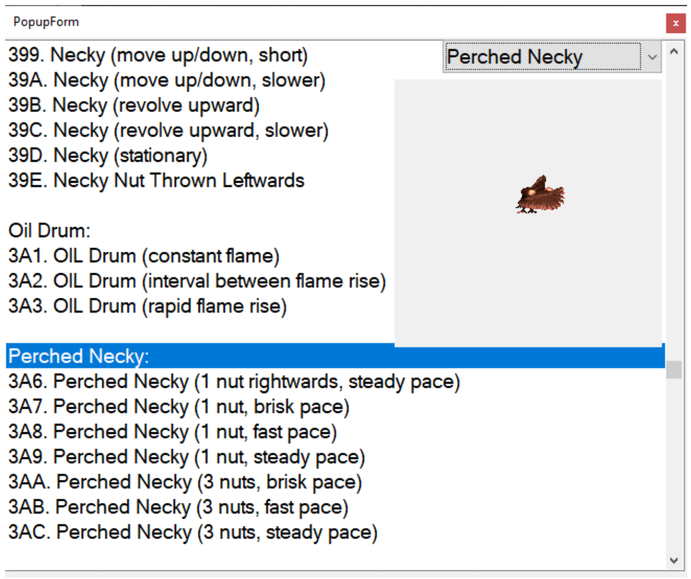
Opens the current entity's palette that they point to in the PALETTE EDIT.

APPLY

This program doesn't care about what you use for inputs, it just places what values are in the boxes. *So, make sure your data is accurate*. Also, on apply, that specific pointer is added to the list of common.

[ ]

Clicking this opens a new popup like the following:



399. Necky (move up/down, short)
39A. Necky (move up/down, slower)
39B. Necky (revolve upward)
39C. Necky (revolve upward, slower)
39D. Necky (stationary)
39E. Necky Nut Thrown Leftwards

Oil Drum:
3A1. OIL Drum (constant flame)
3A2. OIL Drum (interval between flame rise)
3A3. OIL Drum (rapid flame rise)

Perched Necky:
3A6. Perched Necky (1 nut rightwards, steady pace)
3A7. Perched Necky (1 nut, brisk pace)
3A8. Perched Necky (1 nut, fast pace)
3A9. Perched Necky (1 nut, steady pace)
3AA. Perched Necky (3 nuts, brisk pace)
3AB. Perched Necky (3 nuts, fast pace)
3AC. Perched Necky (3 nuts, steady pace)

This is a list, broken down by type, of every available entity type in the game. There is a preview for any entity that has a palette/default animation. Some entities have some more info, like barrel cannons.

To select an entity, highlight an entity and either: right-click, double-click, or press enter to update your entity pointer value. On Exit entities as well as the custom Exits, the type has been automatically changed to 2.

OPEN IN ENTITY EDITOR

See the ENTITY EDITOR section for details. This takes the current pointer and feeds it as an arg to the entity editor.

ADD LEVEL TO COMMON

Adds every entity in the level, minus those with type 2, to the COMMON ENTITY section.

D

If there is more than 1 entity in the level, this deletes entities. Or rather, just shifts to the end and replaces all with zeroes. The game reads the entity map until it encounters type 0. 0 Means no more entities are read.

A

If there is space, another entity is added. This looks for 00s at the end of a level and counts how many multiples of 8 there are. This asks you how many entities you would like to add, then it inserts those entities for you at the current screen location.

COMMON ENTITIES

Any entity you applied is added to this list. By clicking on apply, the point and type of the entity you input are temporarily saved. Clicking on an entity in common sets the values of your saved entity (type and pointer). This is useful if you want to reuse many entities in a level. If you highlight an entry and right-click it, that entry is removed from the list of common entities.

NEW ENTITIES

There are a bunch of new entities unlike anything DKC has seen before! These are not all thoroughly tested yet, so use at your own risk.

EVIL ANIMALS – These are K Rool possessed animal buddies that hurt you on contact. Rambi will run, turn around at walls, and hurt enemies. Expresso will run, turn around at walls, and jump when you do. And Winky runs and homes in on you.

EVIL ENEMIES – These entities are red. On contact, hit or kill, the current Kong will be damaged.

FRIENDLY ENEMIES – These entities are green. On contact, hit or kill, the current Kong will not be damaged and instead will get a slight boost up.

CUSTOM PODOBO – Here, you have 3 options to select from.

- A stationary podobo that sits on the ground.
- A mini-drum that launches them up
- An invisible floor spawn, that works like a mini-drum but starts at the bottom of the screen.

THWOMP – In this case, a red mini-drum only slams down and hurt you upon going underneath it.

PIT LAUNCH – Here, you have 2 options to select from.

- Pit Launch (Klaptrap) spawns and launches new jumping Klaptraps.
- Pit Launch (Army) spawns and launches a new Army Insta-Roll.

ARMY INSTA-ROLL – This is the same as the original army with one key difference. This army starts in the rolling phase.

KAIZO BLOCK – This works almost identically to the Kaizo blocks found in Mario games. Since there are no blocks in DKC, this uses ab upside-down tire with a cave platform on top of it.

VARIOUS POWER UPS – Here, you have 4 options to select from.

- Swim powerup transforms you into the swimming state for 2.5 seconds. This is useful to get to higher places. Additionally, on contact, this resets your y momentum and gives you a slight boost in y.
- Jump powerup gives you an additional (1 extra max) jump until you hit the ground. This is also useful to get to higher places.
- Roll powerup gives you an additional (3 extra max) roll until you are standing on the ground. This is useful to get to faraway places.
- Speed powerup doubles your speed in the x direction for 10 seconds. Your current Kong sparkles too, indicating that the speedup is active.

RESPAWNING STUFF – While 1 new entity, this works on multiple entities. This can be used to respawn any of the 4 powerups listed below. Alternatively, you have the option to respawn 2-barrel types.

WARP BARRELS – There are 4 new custom warp barrels. The name pretty much tells what each does. R is a right-warp while L is left. A '1' or '2' indicates how many screens away the warp is. So R2 is a rightward 2 screen warp.

MINCER SWITCH BARREL – This places a barrel and a mincer on the screen. These 2 objects are tied together. When you touch the barrel, the barrel switches on and the mincer moves left. After a bit of time, the mincer travels back right.

CANNONBALL LAUNCH – Here, you have 3 options to select from.

- Custom cannon ball spawns a blast barrel that shoots a cannonball left in a little over 2 second increments.
- Custom cannon ball spawns a blast barrel that shoots a cannonball right in a little over 2 second increments.
- Custom cannon ball spawns a cannonball that falls down in a little over 2 second increments.

MINI NECKY SPITTING TOKENS – Here, you have many options to select from.

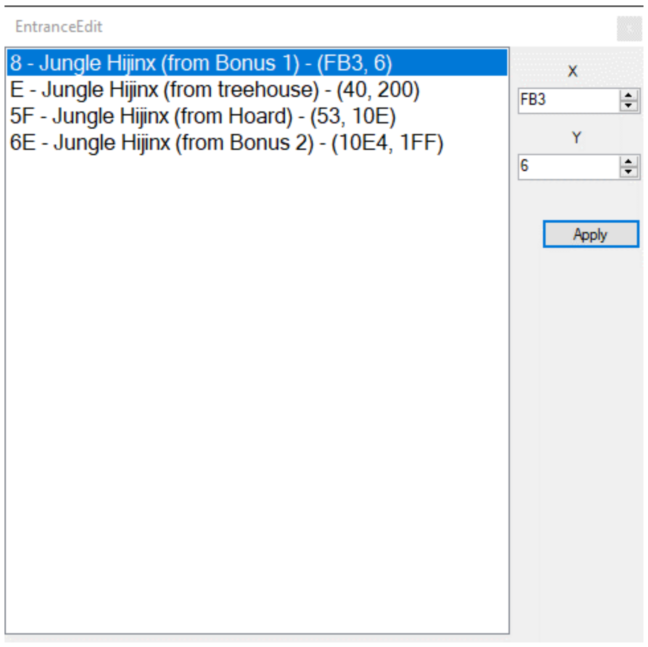
There are many varieties of mini-neckies. A, B, + C all signify a different type of mini-necky (number of shots, movement pattern) as well as each letter later in the alphabet represents a faster speed. All tokens have a normal launch pattern EXCEPT the red Winky token. That one is a homing token (ie it accelerates towards you).

CUSTOM STEEL KEG (SHELL JUMP) – This is meant in a way to mimic shell jumps. This recolors a steel keg and throwing has no animation, so throwing is instant. The ability to ride tin cans makes gaining height possible by jumping off tin cans in the air.

CUSTOM CAVE PLATFORM (CIRCLE) – This acts as a normal game platform except for its circular movement.

ENTRANCE EDITING (IN-DEPTH)

Text editing entrances is super simple to get your head around. Each entrance is shown in a list and represented by 4 bytes (2 ints). Those entrances are labeled and positioned without the y inverted. The X and the Y coordinates refer to a coordinate in the selected level. The only caveats to this, are that 1-1 entrance and all checkpoints are missing. This is because those entrances are tied to entities. The 1-1 entrance has a special entity script, and all the checkpoints have their entrances tied to it. At least in this program they are. They are not linked on the game end and can be changed via hex editing.

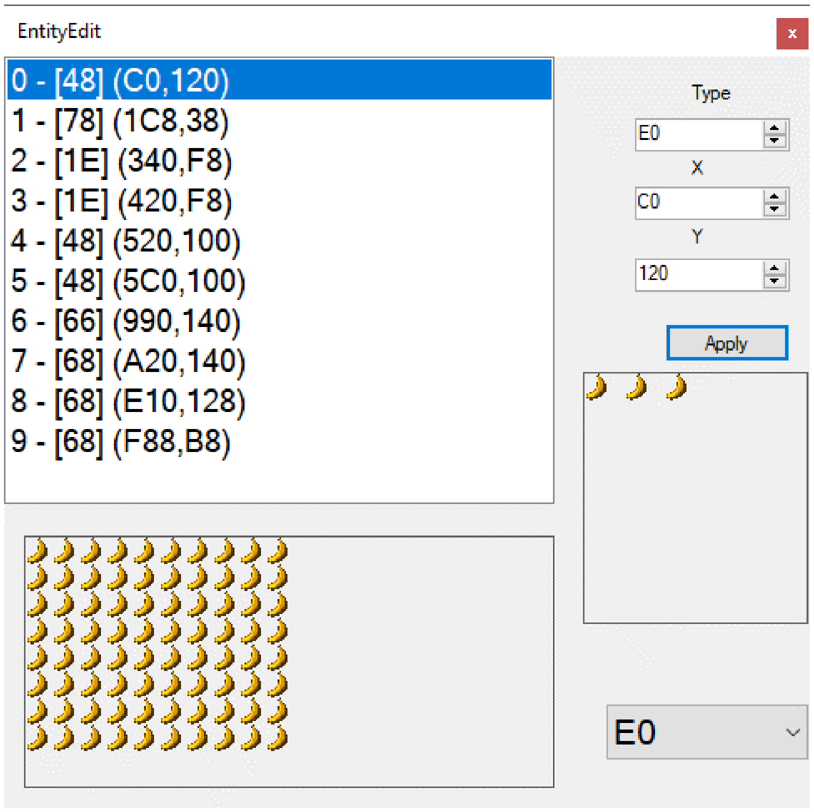


8 - Jungle Hijinx (from Bonus 1) - (FB3, 6)
E - Jungle Hijinx (from treehouse) - (40, 200)
5F - Jungle Hijinx (from Hoard) - (53, 10E)
6E - Jungle Hijinx (from Bonus 2) - (10E4, 1FF)

X: FB3
Y: 6

BANANA EDITING (IN-DEPTH)

Banana text is like cameras, in that the y coordinate is inverted. But that is where the similarities end.


EntityEdit

0 - [48] (C0,120)
1 - [78] (1C8,38)
2 - [1E] (340,F8)
3 - [1E] (420,F8)
4 - [48] (520,100)
5 - [48] (5C0,100)
6 - [66] (990,140)
7 - [68] (A20,140)
8 - [68] (E10,128)
9 - [68] (F88,B8)

Type: E0
X: C0
Y: 120

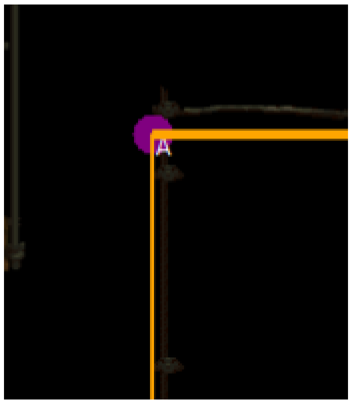
Fortunately, this program does all the hard work for you! If you would like to learn more of the specifics, Simion32 explains it well here.

It is because of his research that this editor even can edit bananas!

Positioning the bananas is self-explanatory. On the editor window, the bottom has a lookup feature, and after you apply a type, you can see exactly which one is used.

PLATFORM EDITING (IN DEPTH)

Platform editing is actually fairly straightforward. You may have noticed with ‘view’ that the thickness of the yellow lines between platform path points represents speed.



Platform paths are essentially arrays of destinations that the platform will visit. Each index is 4 or 8 bytes, depending on the command.

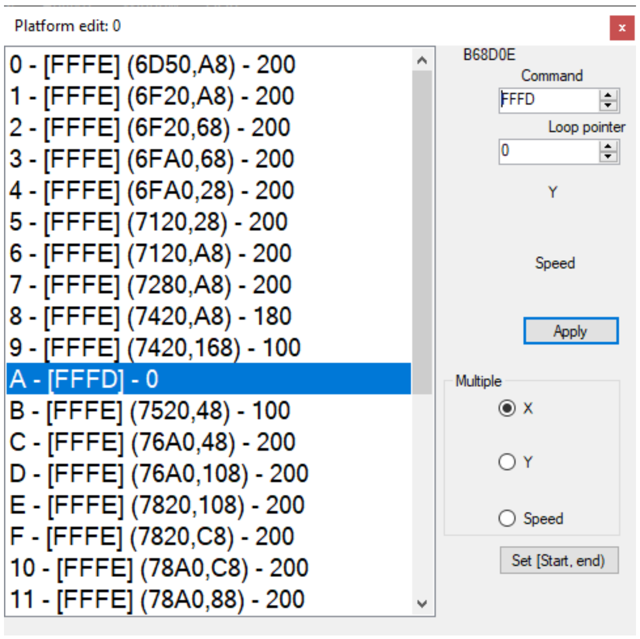
If command is fffe, the game reads 8 bytes. 0+1 are command (fffe), 2+3 are speed, 4+5 are x coords, 6+7 are y coords. When the game sees command fffe, it moves the platform directly to those coords read at the read speed.

If command is fffd, the game reads 4 bytes. 0+1 are command (fffd) and 2+3 is the loop pointer address. When the game sees command fffd, it goes to the address at the loop pointer and executes that instead.

If command is ffff, the game reads 2 bytes. 0+1 are command (ffff). When the game sees command ffff, it knows that the platform script is done.

For all else, the game reads 4 bytes. Bytes 0+1 are x coords and 2+3 are y coords. When the game sees anything else for command, speed is retained from the previous index and the game goes directly to the read coords.

To make looping easier, it has been implemented so that you can optionally just enter in the path index instead of the actual pointer. If you change the command to fffd and hit enter, you get this:



Platform edit: 0

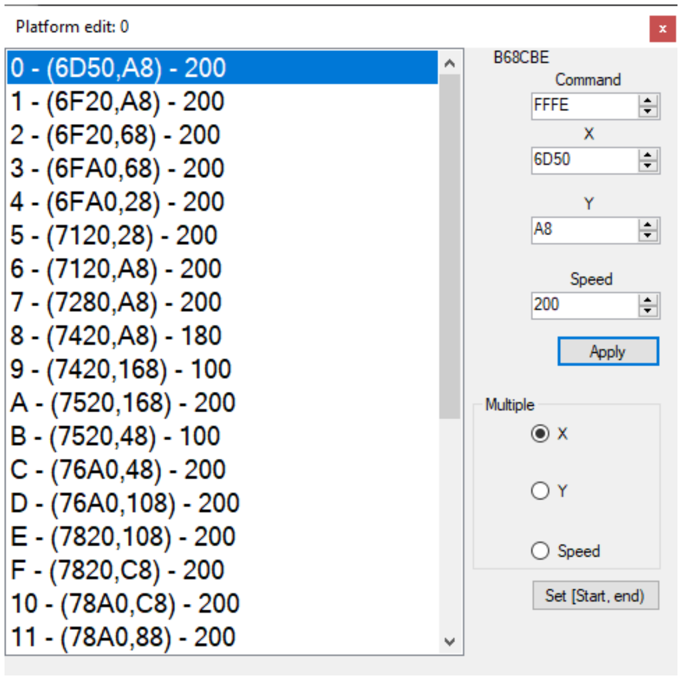
```
0 - [FFFE] (6D50,A8) - 200
1 - [FFFE] (6F20,A8) - 200
2 - [FFFE] (6F20,68) - 200
3 - [FFFE] (6FA0,68) - 200
4 - [FFFE] (6FA0,28) - 200
5 - [FFFE] (7120,28) - 200
6 - [FFFE] (7120,A8) - 200
7 - [FFFE] (7280,A8) - 200
8 - [FFFE] (7420,A8) - 180
9 - [FFFE] (7420,168) - 100
A - [FFFD] - 0
B - [FFFE] (7520,48) - 100
C - [FFFE] (76A0,48) - 200
D - [FFFE] (76A0,108) - 200
E - [FFFE] (7820,108) - 200
F - [FFFE] (7820,C8) - 200
10 - [FFFE] (78A0,C8) - 200
11 - [FFFE] (78A0,88) - 200
```

B68D0E

Command: FFFD

Loop pointer: 0

Here, you can enter in the path index (the first number on the left) into the loop pointer box. If you press enter, this automatically changes for you. Be careful that the loop start and loop end have the same height! DKC tends to overshoot height otherwise.


Platform edit: 0

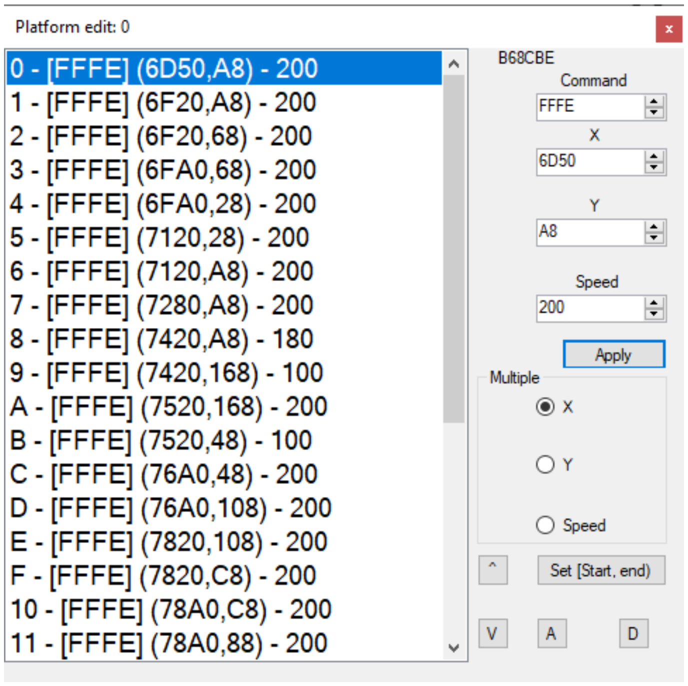
0 - (6D50,A8) - 200
1 - (6F20,A8) - 200
2 - (6F20,68) - 200
3 - (6FA0,68) - 200
4 - (6FA0,28) - 200
5 - (7120,28) - 200
6 - (7120,A8) - 200
7 - (7280,A8) - 200
8 - (7420,A8) - 180
9 - (7420,168) - 100
A - (7520,168) - 200
B - (7520,48) - 100
C - (76A0,48) - 200
D - (76A0,108) - 200
E - (7820,108) - 200
F - (7820,C8) - 200
10 - (78A0,C8) - 200
11 - (78A0,88) - 200

B68CBE
Command: FFFE
X: 6D50
Y: A8
Speed: 200
Multiple: (● X) ○ Y ○ Speed

The label in the top right gives you the 24-bit address of that index in rom. The last 4 digits are what you would use for the loop pointer.

Multiple works just how multiple did in CAMERA EDITING (IN-DEPTH).

The coordinates for the very first platform index are tied to the entity the path belongs to.


Platform edit: 0

0 - [FFFE] (6D50,A8) - 200
1 - [FFFE] (6F20,A8) - 200
2 - [FFFE] (6F20,68) - 200
3 - [FFFE] (6FA0,68) - 200
4 - [FFFE] (6FA0,28) - 200
5 - [FFFE] (7120,28) - 200
6 - [FFFE] (7120,A8) - 200
7 - [FFFE] (7280,A8) - 200
8 - [FFFE] (7420,A8) - 180
9 - [FFFE] (7420,168) - 100
A - [FFFE] (7520,168) - 200
B - [FFFE] (7520,48) - 100
C - [FFFE] (76A0,48) - 200
D - [FFFE] (76A0,108) - 200
E - [FFFE] (7820,108) - 200
F - [FFFE] (7820,C8) - 200
10 - [FFFE] (78A0,C8) - 200
11 - [FFFE] (78A0,88) - 200

B68CBE
Command: FFFE
X: 6D50
Y: A8
Speed: 200
Multiple: (● X) ○ Y ○ Speed

Four new buttons have since been added.

^ – Moves the highlighted index up one

V – Moves the highlighted index down one

A – Adds a new index

D – Deletes an index

WARNING!

Before using the loop feature, press ‘Select’ from the main screen to refresh path addresses.

FILE

LOAD

Load any ROM

LOAD (RESTORE FROM BACKUP – END)

When the program closes, it writes a backup to the disk. If you chose not to save your previous day’s work, you can restore the last rom here.

LOAD (RESTORE FROM BACKUP – START)

When a ROM is first loaded, a backup copy is saved. So, if you tweak a ROM, sav, then change your mind and want to undo everything, you can do that here.

SAVE

Overwrites whatever file you loaded. Optionally, CTRL+s does the same thing. This works any time.

SAVE AS ROM

Allows you to specify the file name and the directory and saves there. This only works after a level has been loaded.

RESTORE BACKUP

Expanded on in the BACKUPS section

RENAME VERSION

Expanded on in the BACKUPS section

IMPORT

TILEMAP

Takes the *.bin file previously exported and imports it into place. Size is NOT verified with this, so you can easily overshoot.

OBJECT MAP

Takes the *.bin file previously exported and imports it into place. Size is NOT verified with this, so you can easily overshoot.

CAMERAS (HORIZONTAL)

Takes the *.bin file previously exported and imports it into place. Size is NOT verified with this, so you can easily overshoot.

BANANA MAP

Takes the *.bin file previously exported and imports it into place. Size is NOT verified with this, so you can easily overshoot. ONLY WORKS WITH FILES EXPORTED WITH THIS PROGRAM.

PLATINUM TILEMAP

Please go to PLATINUM (*.FMF) for more info.

ENTITY INIT BANK

Takes the *.bin file previously exported and imports it into place. Size is NOT verified with this, so you can easily overshoot.

ENTRANCES

Takes the *.bin file previously exported and imports it into place. Size is NOT verified with this, so you can easily overshoot. Every 7 bytes are read. The first 3 are address, the next 2 are x coordinates, and the final 2 are y coordinates.

STAGE

This imports a large file. If present, the file includes tilemap, object map, horizontal cameras, bananas, paths, level attributes for all related levels, stage boundaries, stage music for all related levels, stage coordinates, and a few bulk related values such as reverse squawks, Kong start, and quick K Rool Kredits.

BULK

Bulk imports a huge file that contains all main stages and some other stuff, minus bonuses and bosses. Extra stuff includes the entity script bank, the stage names, all background palettes, all object palettes, all of the Kong Family text, the regular credits, and the Kredits.

EXPORT

ENTITY INIT BANK

Exports the entity most of the script bank. Specifically, 0x6fff bytes starting at 0xb59000.

TILEMAP

Exports the tilemap as bytes in exactly the same format the game uses.

OBJECT MAP

Exports the object map in bytes the way the game reads.

CAMERAS (HORIZONTAL)

Exports the cameras in bytes the way the game reads.

BANANAS

Exports the object map in bytes the way the game reads. The name of the file is custom to this as well.

ENTRANCES

Exports the entrances in bytes as follows:

For each entrance export 7 bytes

- 3 for address
- 2 for x coordinates
- 2 for y coordinates

.FMF (PLATINUM)

Please go to PLATINUM (*.FMF) for more info.

STAGE

This exports a large file. If present, the file includes tilemap, object map, horizontal cameras, bananas, paths, level attributes for all related levels, stage boundaries, stage music for all related levels, stage coordinates, and a few bulk related values such as reverse squawks, Kong start, and quick K Rool Kredits.

BULK

Bulk exports a huge file that contains all main stages and some other stuff, minus bonuses and bosses. Extra stuff includes the entity script bank, the stage names, all background palettes, all object palettes, all of the Kong Family text, the regular credits, and the Kredits.

MOD (IGNORE THIS)

Truly ignore this option. This option does some weird things to levels based on the author’s needs at the time.

EMU

This set of options allows you to open you current modified ROM in a third-party program. Basically, if drag and drop works in opening a ROM, this program will work too.

SELECT EMU PATH

Here, you can select the exe used for an emulator. Warning! Many emulators are a pain to get working with this feature. The path you choose is always remembered so you don’t have to select each time.

LAUNCH (F7)

You can also highlight the main window and press F7 to launch. This locally saves a copy of your ROM named ‘test.smc’ you can play.

SELECT HEX EDITOR PATH

This is identical to the ‘SELECT EMU PATH’ button.

LAUNCH HEX

This does a couple things different from the ‘LAUNCH (F7)’ button. On LAUNCH HEX, the main program also writes ‘test.smc’ locally to be loaded externally, but then enters in a loop state constantly asking if you are finished. When you specify that you are finished, that local file is opened in this program automatically. So this a good option when you prefer to edit and save you file in a hex editor. This was implemented so that this plays nice with the hex editor of your choice.

EDIT

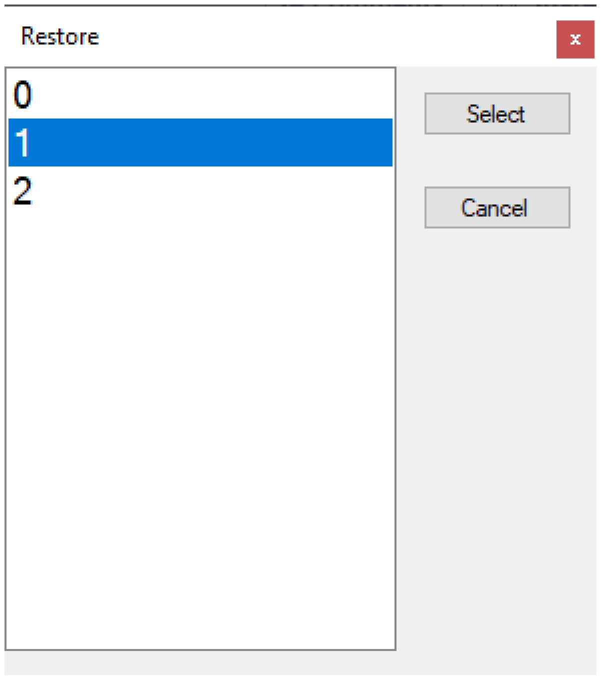
TILEMAP

UNDO (CTRL + Z)

Undo the last tilemap change you made.

RESTORE

A list is generated of all previous backups allowing you to restore whichever one you choose.



This program creates a tilemap backup every time you click ‘Select’ to reload a level. Backups are based on the level, so 1-1 could have 30 backups while 1-2 has 2.

ENTITY INIT EDITOR

See the ENTITY EDITOR section for details.

PALETTE EDIT

See PALETTE EDITOR for details. After completing, ‘Select’ is clicked again for an instant preview.

SNAP

This is the amount entities will move by when you drag them with the mouse. Especially useful for bonus walls, when you want to move the entity but also want to keep its relative position in the tile.

TRANSPARENT

This opens up a color dialog window so you can select which color you would like transparent to appear as. The ‘Select’ button is pressed to offer instant feedback.

LEVEL

CLEAR TILEMAP

Puts tile 0 in every tile in the tilemap.

CLEAR CAMERA MAP (HORIZONTAL)

Maximizes every camera in the level from a Y perspective, effectively removing all camera restrictions.

CLEAR DEFAULT ENTITIES

Changes all entities of type 1, f, and d to that of a banana bunch.

CLEAR BANANA MAP

Replaces the entire banana map with zeroes.

ORDER ENTITIES

This is just a flag that is checked when you click ‘Select,’ and if flagged orders entities for you. Also, this is saved so no need to recheck this.

SET END OF LEVEL

This does just what it sounds like. The only thing to note is this doesn’t work in vertical levels.

MISC

ASM MODS

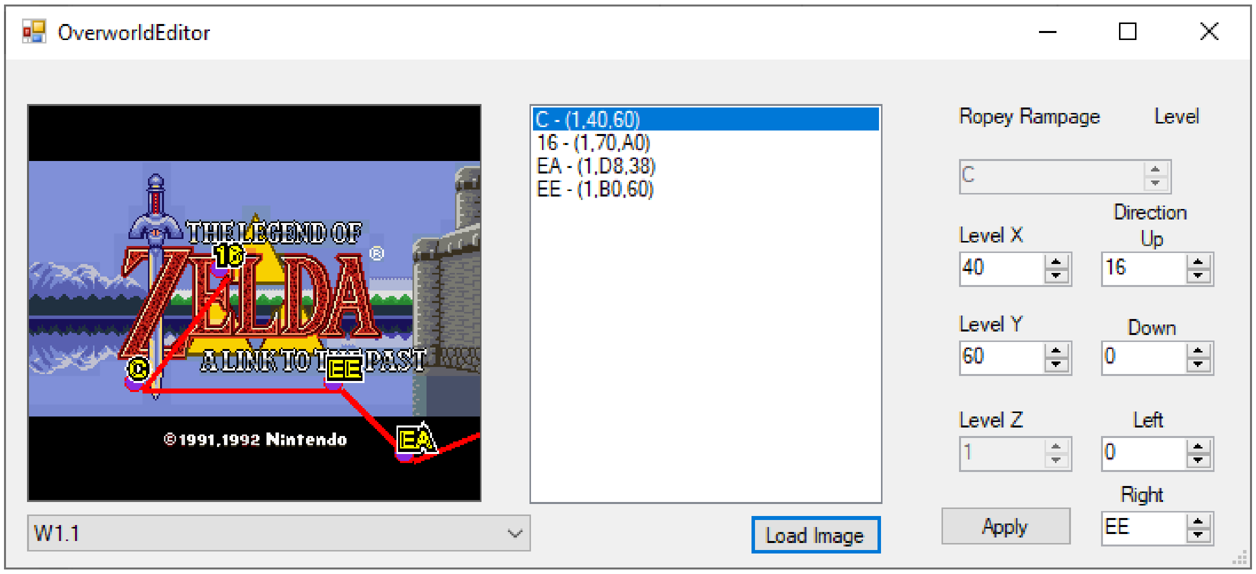
ASM mods are the core of what makes this program great. Don’t worry about applying them ASM updates are optionally applied in multiple occasions. Such as:

1. When a custom entity is applied
2. When level attributes are opened
3. When custom level attributes are opened
4. When the percent editor is opened

Otherwise, you need to refresh by clicking this button if you have a newer ASM folder of files to apply. This means you can create a hack completely free of ASM if you please!

OVERWORLD EDIT

This displays most of the assets involved in drawing the overworld. This allows you to just change the x and y coordinates of stages. All of the background images reflect what is in rom. For example:



C - (1,40,60)
16 - (1,70,A0)
EA - (1,D8,38)
EE - (1,B0,60)

Ropey Rampage Level: C

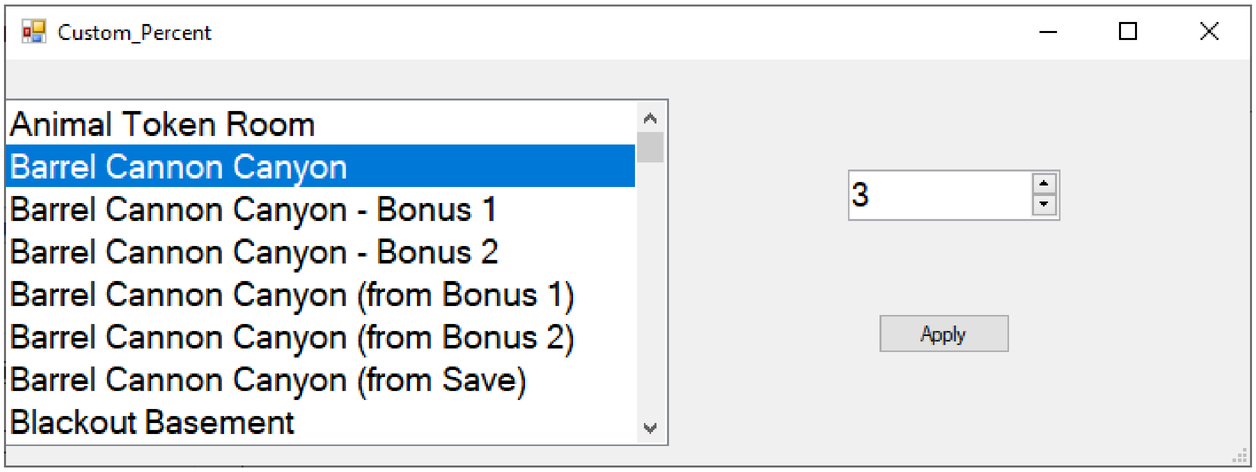
Level X: 40
Level Y: 60
Level Z: 1

Direction
Up: 16
Down: 0
Left: 0
Right: EE

PERCENTAGE

PERCENT CHANGE

This tool provides you with each stage and the number of percentage points associate with each stage.



Animal Token Room
Barrel Cannon Canyon
Barrel Cannon Canyon - Bonus 1
Barrel Cannon Canyon - Bonus 2
Barrel Cannon Canyon (from Bonus 1)
Barrel Cannon Canyon (from Bonus 2)
Barrel Cannon Canyon (from Save)
Blackout Basement

3

These are percent points. Notice Barrel Cannon Canyon has a 3 but Bonus 1 has 0. Percentage points aren’t counted in bonuses. They are tied to bonus doors/barrels. Therefore, they are in the base stage (which is why the points are all on the base stage). Max % is calculated by counting all this up. Percentage is 1 for finishing the stage, and 1 each for bonuses. To change max percent, just change what is here.

CUSTOM MODS SCH1EY (CELESTE)

Not relevant. This changes what stage percentage goes to.

COPYRIGHT EDIT

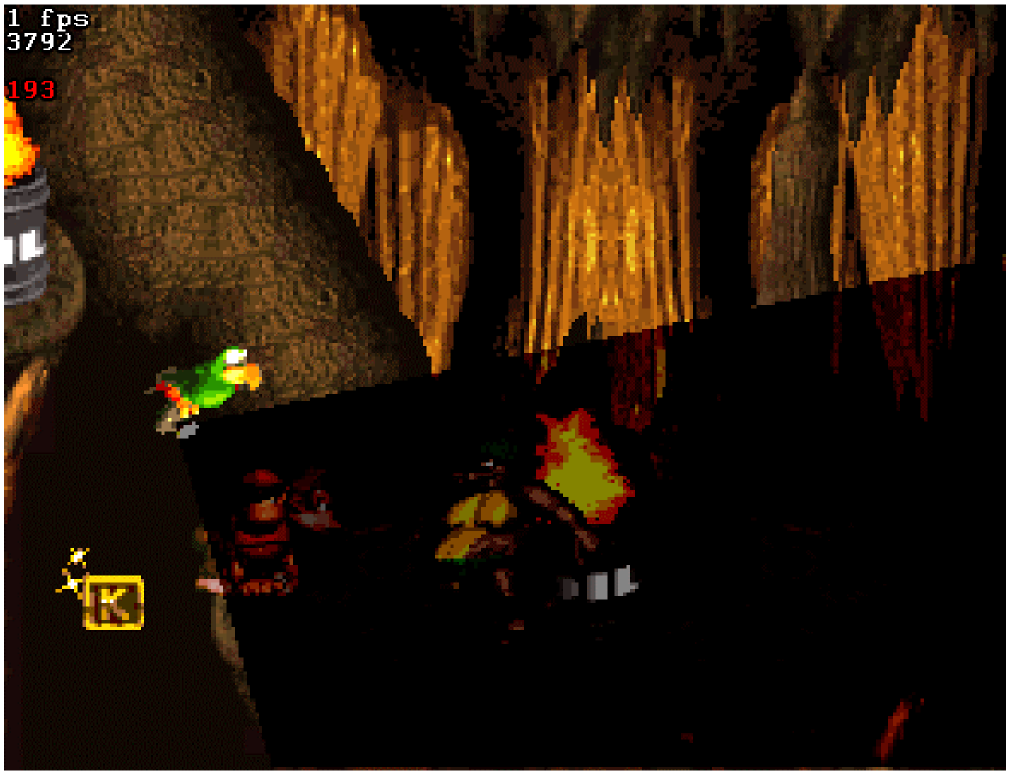
This tool reads the current text for copyright and displays it full using ROM assets. Year is written automatically, as well as NINTENDO always being there. You can apply up to 7 different names on copyright.

TEXT EDITOR

Please see TEXT EDITOR for more details.

REVERSE SQUAWKS

Toggling this option changes the squawks behavior. When in 'reverse' mode, the stage is lit, and squawks shines darkness instead.



QUICK FAKE CREDITS

Checking this changes the game so when K Rool starts his fake credits, they immediately finish so you still get a break, but it is only ~3 sec instead of +30s.

LEVEL ATTRIBUTES

Please see LEVEL ATTRIBUTES for more details.

CUSTOMLEVEL ATTRIBUTES

Please see LEVEL ATTRIBUTES for more details.

MUSIC PICKER

This option allows you to choose the track to play in a stage. 'Apply (Sanity)' just applies the change to all versions of the level. Because of the way DKC is coded, Jungle Hijinx has many level codes for the same level. This is expanded upon in the LEVEL ATTRIBUTES section. Sanity basically just takes every level that has the same name and applies the changes to that code.

STARTING KONG

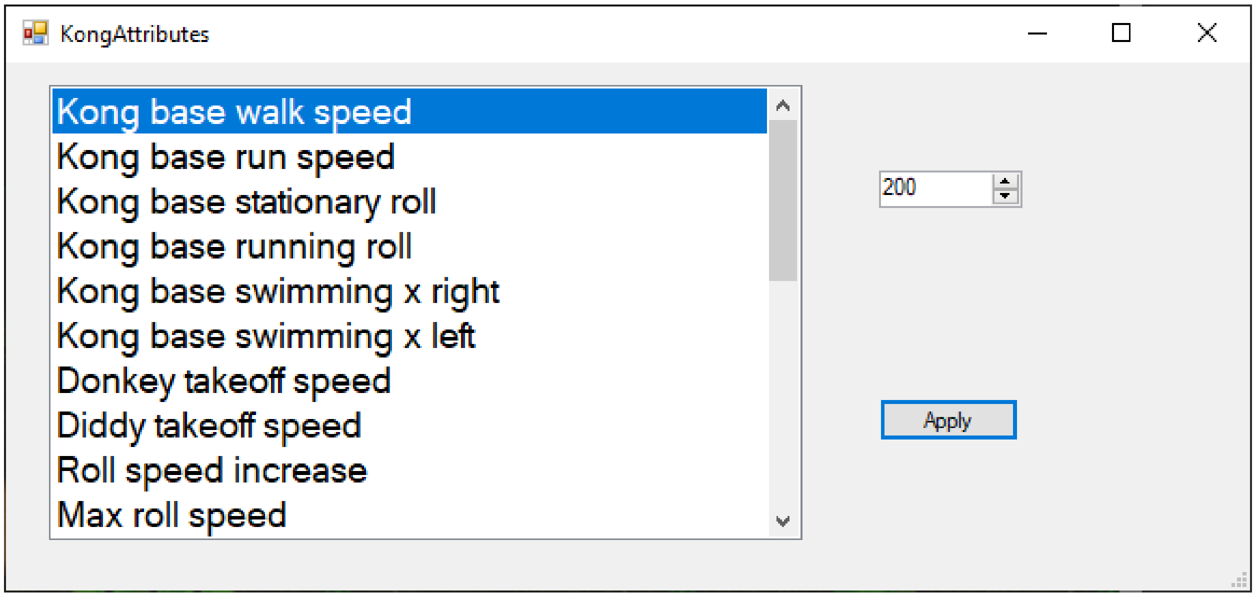
This option simply changes which Kong you start the game as.

WORLD TERMINATOR

This option flags a particular as the final stage of that world. There is a slight glitch in that funky doesn't work correctly if the boss is not used.

KONG ATTRIBUTES

This option allows you edit various Kong attributes, like roll speed or takeoff speed (jump height). Values too high will break the game. Please use responsibly.



Kong base walk speed
Kong base run speed
Kong base stationary roll
Kong base running roll
Kong base swimming x right
Kong base swimming x left
Donkey takeoff speed
Diddy takeoff speed
Roll speed increase
Max roll speed

200

SAVE AFTER EACH LEVEL

This tells the game to save after each level, so life management isn't a thing.

EXTERNAL

These are all tools coded already and working but are not all implemented into this program.

OBJECT SCRIPT LOOKUP

This feature is a more advanced tool. This provides you with the object start address based on its $d45 id code. After entering in $d45, click 'Get' or press enter. To copy the code to clipboard, select an entry and click copy.

BG IMPORTER

Coming soon!

SPRITE EDIT

Coming soon!

MUSIC TRACK EDIT

Coming soon!

ANIMATION EDIT

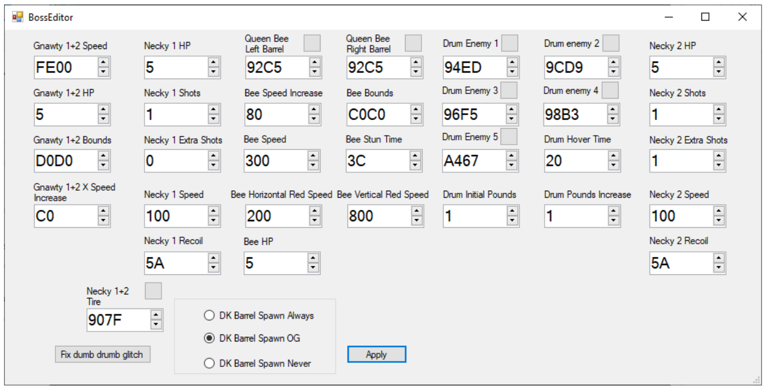
Coming soon!

K ROOL EDIT

Description coming soon!

BOSS EDIT

There are options to edit bosses here. *Most* of these edits can be done in the entity editor. This tool is just a more organized way.



BossEditor

Gnawty 1+2 Speed: FE00	Necky 1 HP: 5	Queen Bee Left Barrel: 92C5	Queen Bee Right Barrel: 92C5	Drum Enemy 1: 94ED	Drum enemy 2: 9CD9	Necky 2 HP: 5
Gnawty 1+2 HP: 5	Necky 1 Shots: 1	Bee Speed Increase: 80	Bee Bounds: C0C0	Drum Enemy 3: 96F5	Drum enemy 4: 98B3	Necky 2 Shots: 1
Gnawty 1+2 Bounds: D0D0	Necky 1 Extra Shots: 0	Bee Speed: 300	Bee Stun Time: 3C	Drum Enemy 5: A467	Drum Hover Time: 20	Necky 2 Extra Shots: 1
Gnawty 1+2 X Speed Increase: C0	Necky 1 Speed: 100	Bee Horizontal Red Speed: 200	Bee Vertical Red Speed: 800	Drum Initial Pounds: 1	Drum Pounds Increase: 1	Necky 2 Speed: 100
	Necky 1 Recoil: 5A	Bee HP: 5				Necky 2 Recoil: 5A

Necky 1+2 Tire: 907F

○ DK Barrel Spawn Always
[✓ DK Barrel Spawn OG]
○ DK Barrel Spawn Never

Pressing any of those blank squares brings up the entity selector. One important thing to note is that the ASM patch fixes the dumb drum glitch. Finally, there are a few radioboxes that tells the game what to do about DK barrel spawns.

BACKUPS

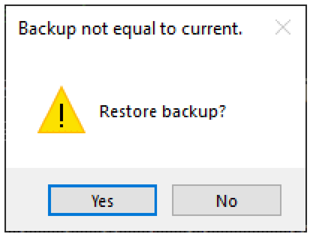
This program backs up often. So you *should* never lose work. If something catastrophic happens to your ROM, you could always export the stage and import the stage on a clean version of DKC.

AUTO-BACKUPS

These backups happen when you load ROM or select a level. Backup – end is written to on level select or program close. Backup – start is written to on ROM load.

BACKUP - END

You may get this on program load:



Backup not equal to current.

Restore backup?

This happens because when the program loads, it automatically tries to load the last ROM you used. If that ROM does not equal the ROM saved as a backup, this appears. This was implemented to counter power losses or other factors that would cause you to lose work. You can optionally load that backup by pressing FILE>LOAD (RESTORE LAST BACKUP - END).

BACKUP – START

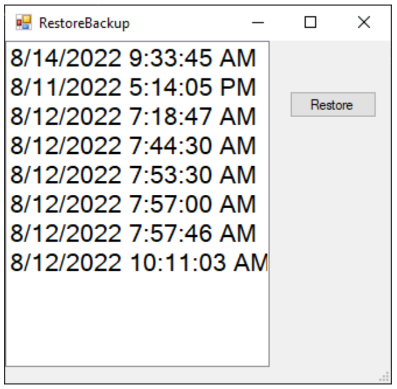
This feature is useful if you break something. If the program is open a while and you make and save a lot of changes, and either don't like those changes or find something is broken, you can restore your ROM to the state it was in when you first loaded the program. You can accomplish this by clicking FILE>LOAD (RESTORE LAST BACKUP - START).

SAVE-BACKUPS

When you save the game, this saves a backup of your ROM in the directory of your current header. To change which directory your save backups go to, click FILE>RENAME VERSION

RESTORING SAVE-BACKUPS

When you press FILE>RESTORE BACKUP, you are presented with this window:



8/14/2022 9:33:45 AM
8/11/2022 5:14:05 PM
8/12/2022 7:18:47 AM
8/12/2022 7:44:30 AM
8/12/2022 7:53:30 AM
8/12/2022 7:57:00 AM
8/12/2022 7:57:46 AM
8/12/2022 10:11:03 AM

This shows the date and time of the last backup performed. You can restore by either highlighting and clicking restore, or by highlighting and right clicking.

RESTORING SAVE-BACKUPS MANUALLY

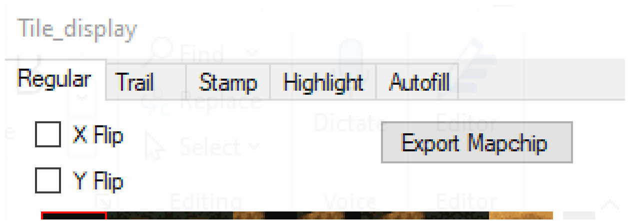
Go to the directory where you put these folders and enter 'Backup Version.' Next you have new folders based on the name you chose. If you are unsure, just sort by modified date and enter. There should be a bunch of '.bac' files there. Take the one you want, rename it, and change the extension to '.smc', and now you have a ROM!

PLATINUM (*.FMF)

EXPORTING

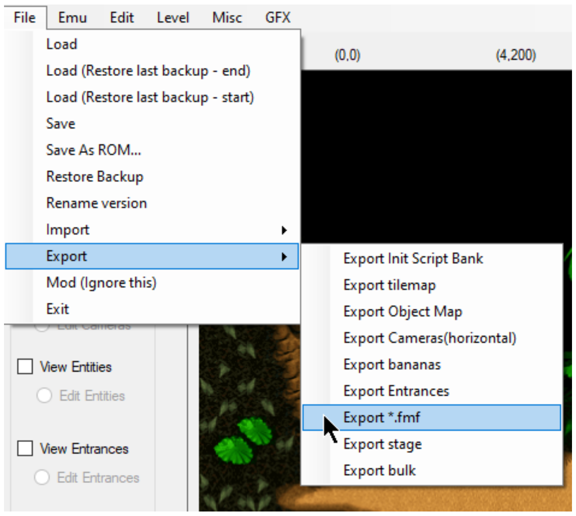
MAPCHIP –

The mapchip is very important! The *.fmf file you export depends on how the mapchip is constructed. You can export all needed mapchips by selecting a level for edit and pressing EXPORT MAPCHIP on TILE_DISPLAY.



FMF –

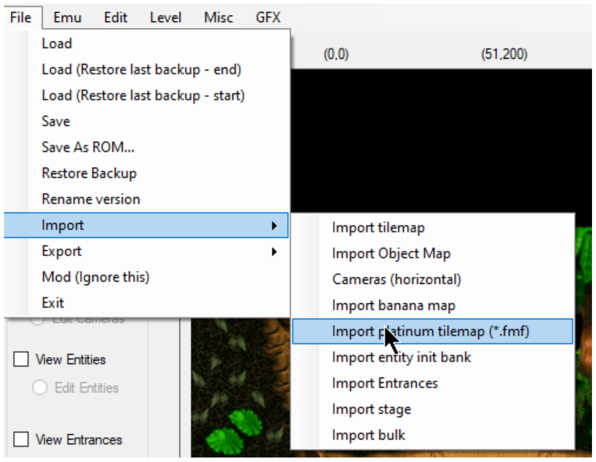
To export a tilemap to use in platinum, click this:



REMINDER! This only works with the provided mapchip!

IMPORTING

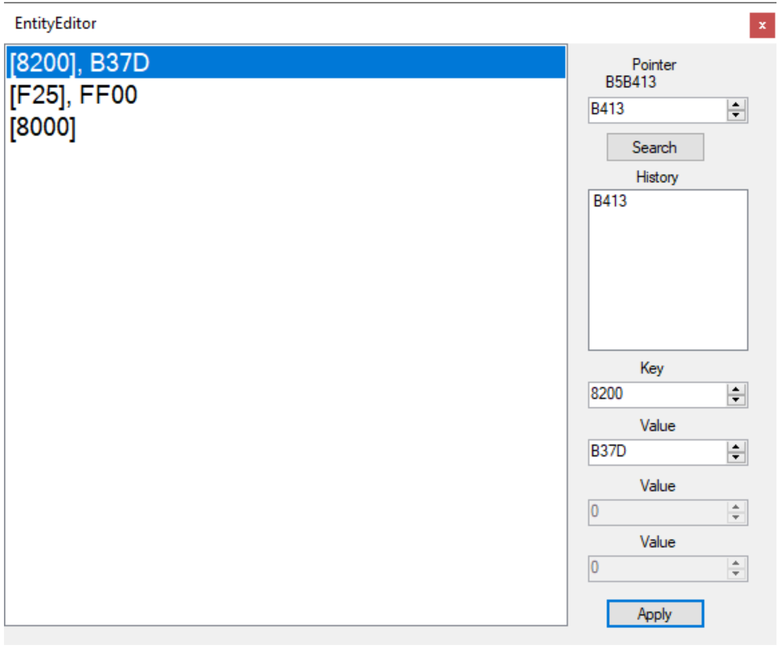
Not much to say here. Just load a level, click this, and select the right fmf.



It should be mentioned that this woks on vertical levels too.

ENTITY EDITOR

If you open this from the drop down, b413 is the pointer loaded. If you instead opened this from entityedit, the pointer of the highlighted entity is used. On open, you might see something like this:



[8200], B37D
[F25], FF00
[8000]

Pointer B5B413: B413
History: B413
Key: 8200
Value: B37D
Value: 0
Value: 0

[XXXX] – denotes an array/key. The number after is the value to be stored in that array. HOWEVER, an [XXXX] at or above 8000 does something special. Check this post from Mattrizzle to see what many do! If you click any one, you could edit key-values.

Common keys:

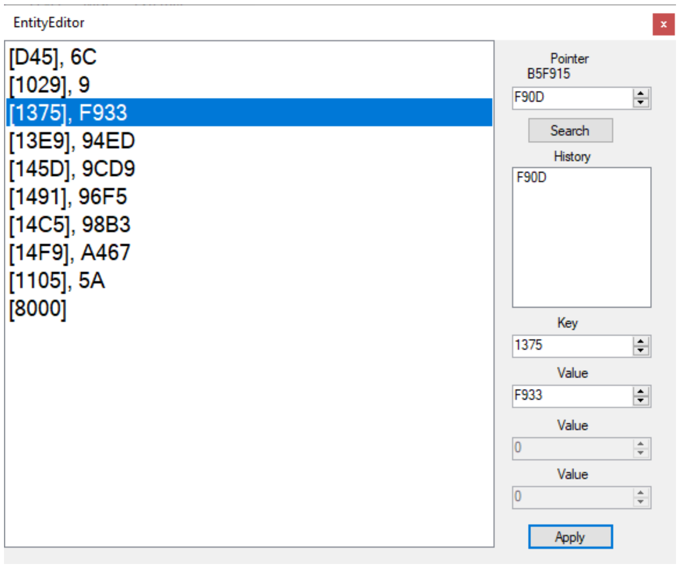
[8200] – Jump to object data at the value given (call)

[8100] – Default animation

[8800] – Palette pointer

[8000] – Return or end

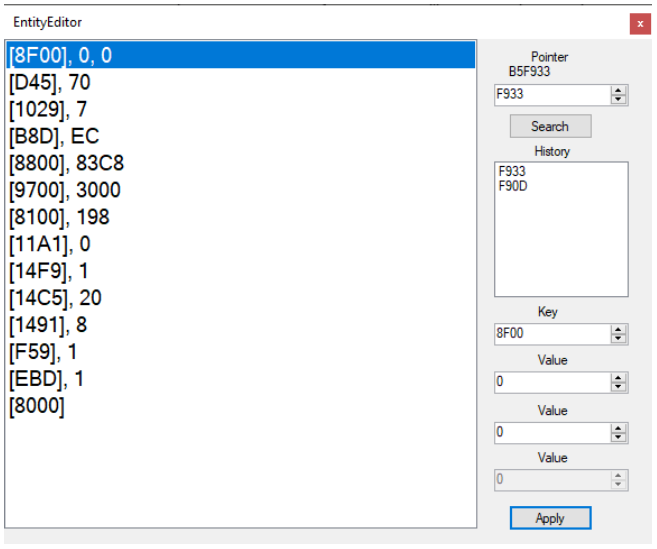
Sometimes, keys hold values pointing to other entities. If you double-click any index (or right click), the highlighted value is automatically searched for. This is really useful for navigating to values that have the key [8200]. If you try that on a value with key [8800], that pointer is fed to the palette editor. If you try that on a value with key [8100], that pointer is fed to the animation editor. An example of how this is used follows:



```
[D45], 6C
[1029], 9
[1375], F933
[13E9], 94ED
[145D], 9CD9
[1491], 96F5
[14C5], 98B3
[14F9], A467
[1105], 5A
[8000]
```

Pointer B5F915: F90D
History: F90D
Key: 1375
Value: F933
Value: 0
Value: 0

When you open up dumb drum in the entity editor, you get the following screen. The main dumb drum entity (and all bosses for that matter) represent the boss room. So the room has those attributes. [1375] points to the actual dumb drum boss entity. Right-clicking on that bring me here:



```
[8F00], 0, 0
[D45], 70
[1029], 7
[B8D], EC
[8800], 83C8
[9700], 3000
[8100], 198
[11A1], 0
[14F9], 1
[14C5], 20
[1491], 8
[F59], 1
[EBD], 1
[8000]
```

Pointer B5F933: F933
History: F933, F90D
Key: 8F00
Value: 0
Value: 0
Value: 0

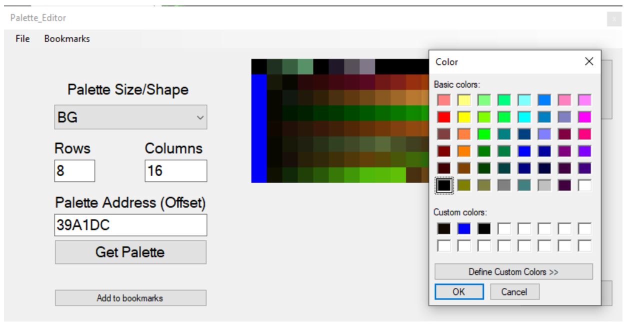
Here is where you can change specific properties of dumb drum. For example, it's palette is at 83c8. [14f9], in this case, points to which phase dumb drum should start on (1-5). It is NEVER recommended to change keys.

Highlight a value in 'history' and either double or right-click to load that specific pointer.

PALETTE EDIT

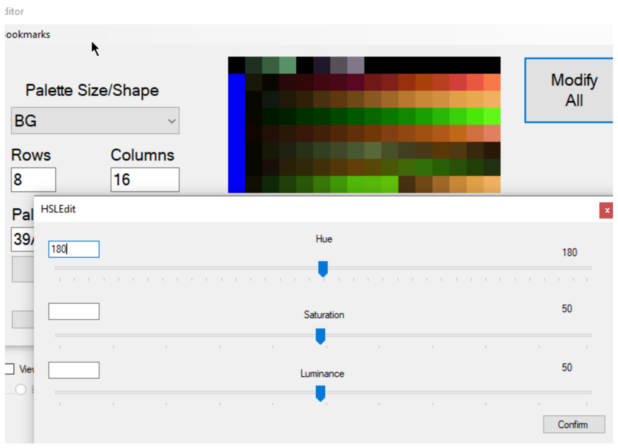
This is a built-in palette editor. The addressing here uses addresses more like what you might find in a hex editor. This is because this came from an outdated program. However, the abilities are still powerful! This has a few different editing options.

First, if you click on any color in the palette, you can manually edit the colors as so:

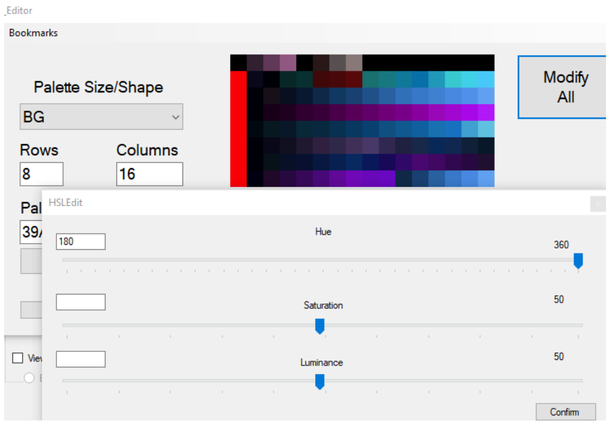


If you hadn't noticed, the 'custom colors' section contain some colors already. These are the previous 3 colors in the palette, in case you wanted similar colors.

Second, if you click 'modify all' this window shows up:



This modifies the HSL values of all the colors at once. The entire palette is updated in the program instantly like so:



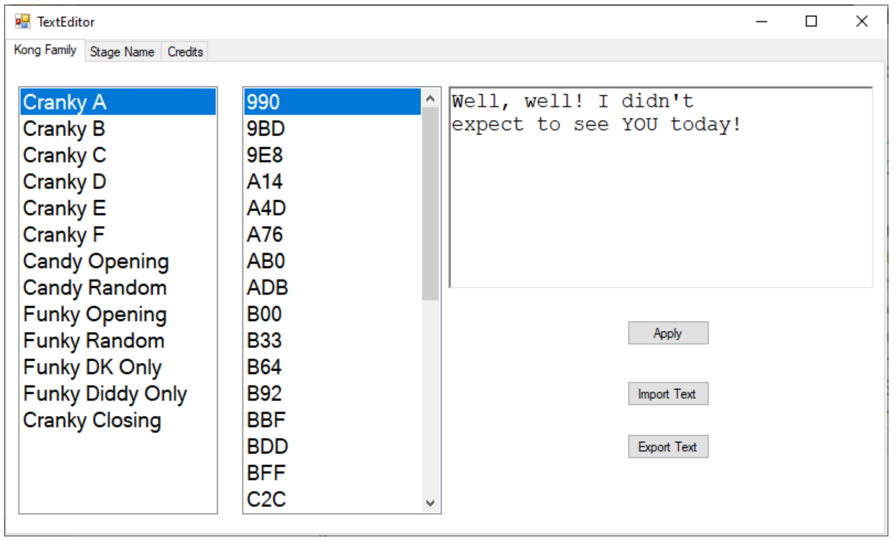
This makes it simple to make changes to multiple colors at once. HSL won't be explained here. There's always Google for that.

The third method of editing is exporting the palette as an image, modifying in a third-party program, and importing it back in.

Finally, click 'Write'. ROM is not written to until you press this.

TEXT EDITOR

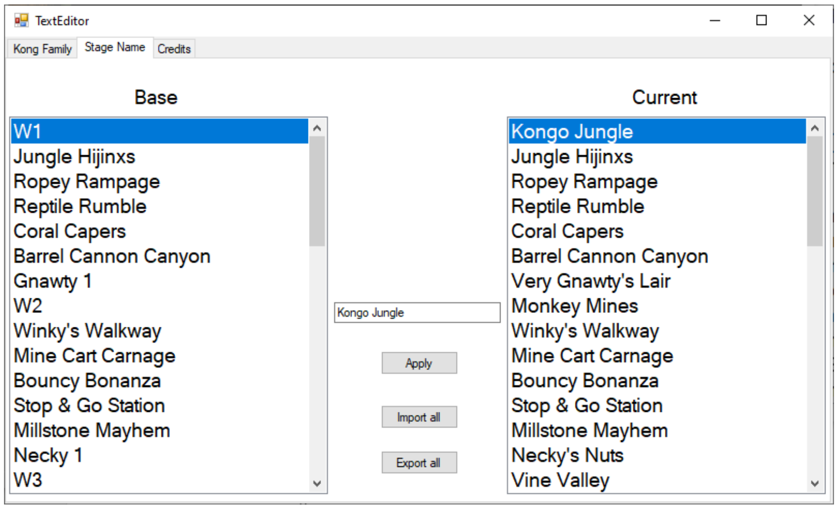
This is a fairly self-explanatory feature, so this won't be gone over in-depth. That being said though, there are still a few things to mention here.



Cranky A	990
Cranky B	9BD
Cranky C	9E8
Cranky D	A14
Cranky E	A4D
Cranky F	A76
Candy Opening	AB0
Candy Random	ADB
Funky Opening	B00
Funky Random	B33
Funky DK Only	B64
Funky Diddy Only	B92
Cranky Closing	BBF
	BDD
	BFF
	C2C

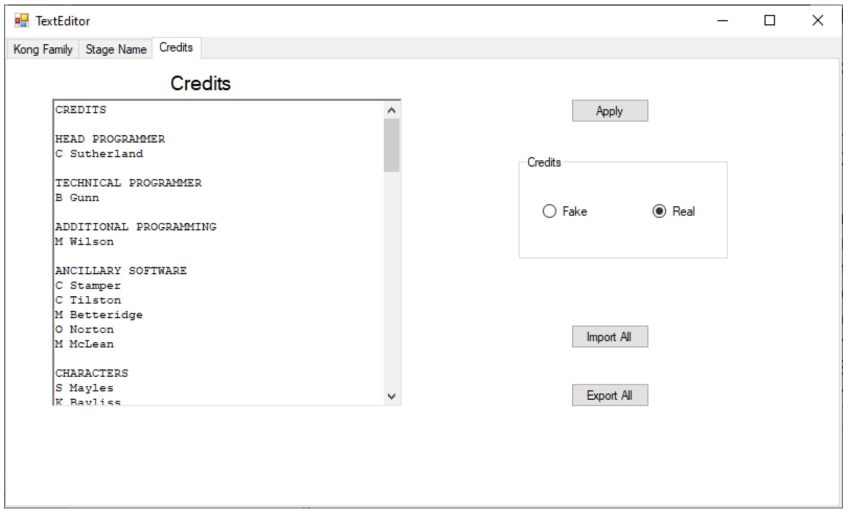
```
Well, well! I didn't
expect to see YOU today!
```

Those characters in the second column are text pointers. Careful about using too much text. This program does not verify line length or rows. A good idea is to stay under 30 characters and 4 rows.



Base	Current
W1	Kongo Jungle
Jungle Hijinxs	Jungle Hijinxs
Ropey Rampage	Ropey Rampage
Reptile Rumble	Reptile Rumble
Coral Capers	Coral Capers
Barrel Cannon Canyon	Barrel Cannon Canyon
Gnawty 1	Very Gnawty's Lair
W2	Monkey Mines
Winky's Walkway	Winky's Walkway
Mine Cart Carnage	Mine Cart Carnage
Bouncy Bonanza	Bouncy Bonanza
Stop & Go Station	Stop & Go Station
Millstone Mayhem	Millstone Mayhem
Necky 1	Necky's Nuts
W3	Vine Valley

Next are stage names. The left holds a list of what the stages represent normally. The right represents what the stages are currently. This overwrites German text for anything changed. A similar limitation to the Kong Family text's limitations applies here. You want to stay under ~30 characters. Pressing 'Enter' is the same as 'Apply' here.



```
CREDITS

HEAD PROGRAMMER
C Sutherland

TECHNICAL PROGRAMMER
B Gunn

ADDITIONAL PROGRAMMING
M Wilson

ANCILLARY SOFTWARE
C Stamper
C Tilston
M Betteridge
O Norton
M McLean

CHARACTERS
S Mayles
K Bayliss
```

Credits: Fake [✓ Real]

Finally, the credits can be edited. A common concern is of credit length. Both real and fake credits can be as long or as short as you want if the byte length of data is satisfied. For example, if credits were 1 line saying, "HELLO WORLD", a valid credit sequence would be.

H

E

L

L

O

W

O

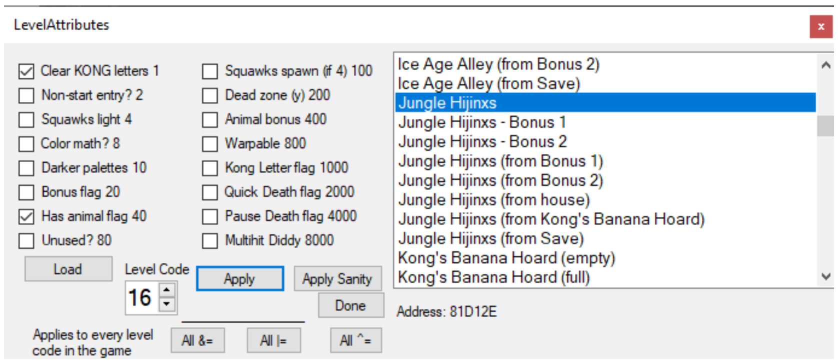
R

L

D

LEVEL ATTRIBUTES

This is a pretty powerful little tool that flags a particular level code as having that attribute. The last 4 are all custom and require ASM to work properly.



LevelAttributes

[✓] Clear KONG letters 1
[ ] Non-start entry? 2
[ ] Squawks light 4
[ ] Color math? 8
[ ] Darker palettes 10
[ ] Bonus flag 20
[✓] Has animal flag 40
[ ] Unused? 80
[ ] Squawks spawn (if 4) 100
[ ] Dead zone (y) 200
[ ] Animal bonus 400
[ ] Warpable 800
[ ] Kong Letter flag 1000
[ ] Quick Death flag 2000
[ ] Pause Death flag 4000
[ ] Multihit Diddy 8000

Level Code: 16

Ice Age Alley (from Bonus 2)
Ice Age Alley (from Save)
Jungle Hijinxs
Jungle Hijinxs - Bonus 1
Jungle Hijinxs - Bonus 2
Jungle Hijinxs (from Bonus 1)
Jungle Hijinxs (from Bonus 2)
Jungle Hijinxs (from house)
Jungle Hijinxs (from Kong's Banana Hoard)
Jungle Hijinxs (from Save)
Kong's Banana Hoard (empty)
Kong's Banana Hoard (full)

Address: 81D12E

The box on the right has every code available in the game, minus the top 26. You may notice that there are several similar codes, like Jungle Hijinx and Jungle Hijinx (from Bonus 1). This was done so things would be different upon return. This format was greatly improved upon in future titles. What this means, is you can have different attributes after certain parts!

LOAD

This loads any level code in the level code box. This is useful if you know the code you want to use and don't feel like finding it in the list. Any code you search for is found for you in the listBox.

LISTBOX

This holds all the level codes used in the game. There are a couple of bizarre unused ones. Clicking any stage, valid or not, automatically puts that level code in the level code box and loads it for you.

APPLY

All of the checkboxes are combined into one number and then written to rom.

APPLY SANITY

This button first applies to the current level code, then applies the same to every related level code. This was implemented if you want the same attributes in all variations of a level. NOTE! This is only available on the current level you are editing.

DONE

This button does nothing but close 'Level Attributes' for you.

ALL &=, |=, OR ^=

These symbols are programming related and won't be explained here. What this does though, is applies that operation to almost every level in the game. This excludes 8 stages from the list:

- Animal Bonus
- Kong's Banana Hoard (empty)
- Kong's Banana Hoard (full)
- Credits
- Expresso Bonus
- Enguarde Bonus
- Rambi Bonus
- Winky Bonus

CUSTOM ATTRIBUTES

KONG LETTER FLAG – This just makes it so all Kong letters are required to beat a level. If you get to the ending without all Kong letters, the ending will be replaced with a mincer.

QUICK DEATH FLAG – This effectively takes out lives. Flagging this makes it so upon death, the game is tricked into thinking you were in a bonus and changes the return destination.

PAUSE DEATH FLAG – This works best when paired with quick death. This makes it so if you hold 'R' on start select, instead of going to the overworld immediately, you are killed. But worry not! This does not steal any Kongs from you!

MULTIHIT DIDDY – This is a way to effectively remove Donkey as a playable character. On levels flagged with this, a Diddy head is placed on the bottom left of the screen next to a number. This shows how many hits you have remaining. A DK barrel replenishes your extra hits just like the original (+1 if you have no hits, +0 if you have 1 extra).

ANIMAL TOKEN RESTART – This is meant to accompany token spitting mini-neckies where if 3 of a single variety are collected, you are started at the last checkpoint.

CREDITS

This couldn't have been done without some help from some of these people:

- Myself086
- H4v0c21
- Kingizor
- Sch1ey
- Mattrizzle
- Simion32
- Snakpak
- Fathlo23

- The DKC Atlas community and Discord
- DKCSpeedruns.com