

Hacking Knockoffaly

Knockoffaly is designed to be the ultimate configurable game. Using the features of the editor you can alter the data structures and use the provided resources to give your board a different look. This is for people who want to go beyond that. You don't need to be a super hacker; but some limited hacking skills would be useful. ResEdit and/or a color bitmap (paint) program is also extremely useful. Always work on a copy of your Knockoffaly stacks just to be sure!

General hacking

Knockoffaly is not locked. Feel free to browse the scripts and see how it works. I set the user level to 3 at first because that's all we need to play. You can get the message box through the back door. [The second die button is a back door. If you COMMAND-CLICK it, it pops up the message box.](#) You can then set the user level to 5, reset the menubar, whatever. Obviously if you make changes you may ruin it. There are a small number Knockoffaly routines that are compiled for speed reasons. The commented source is NOT included just for compactness' sake. If you want the Hypertalk source to those, email me and I'll send it to you. (They're not really very interesting.)

Adding more token icons

Using resEdit or any of a number of available Hypercard Resource manipulation tools you can browse through the ICON and ICN# resources of files that you have. If you find any you like, select them, copy them, and paste them into your [Resource Stack](#). They will then be available as icon choices to anyone editing using your copy of the Resource Stack.

Adding additional sounds

If you have Hypercard or ResEdit you can browse your software collection for snd resources and copy them to your [Resource Stack](#). The additional sounds will then be available to anyone editing using your copy of the Resource Stack.

Dependencies

[Adding icon or snd resources makes a dependency on your altered Resource stack.](#) The configuration only records the [reference](#) to the icon or sound and not its actual data. If you share a configuration using extra icons or sounds with someone else, the next person will not be able to see the additional icons or hear the new sounds unless they have a copy of your altered Resource Stack. [Platypus is not responsible for the performance of altered Resource Stacks!](#)

Adding more “color icons”

You probably know that the “color icons” on the Knockoffaly boards are not really icons, but merely small pict. You can of course make your own small pict. I provide a small PICT file called [templates](#). It has four rectangles in it. The larger two are for vertical and horizontal non property squares. They will be placed so they cover the whole square. The smaller two are for property squares. They are placed so that they do not cover the color bars.

Draw your miniature artwork inside these rectangles. When you turn them into PICT resources, select just inside the boundaries for copying to the clipboard. Do not select the black outlines; they are guidelines. When you use resEdit to create PICT resources from the clipboard, the appropriate stack to copy them to is [KnockEditors](#).

Once you have used your copy of KnockEditors to draw a board with original PICTs, you can share the board with other Knockoffaly players. They don't need the PICTs copied into their copy of KnockEditors; in fact, they don't even need a copy of KnockEditors, as the PICT graphics are incorporated into the board file.

If you don't want to bother with individual PICT resources you can just edit the board file directly in a painting program, drawing your little pictures inside the squares. Of course those “PICT”s will just be part of the board, and will not be available as separate PICT's from the editor.

Examining game or configuration files with a text editor

Game files and configuration files are nothing but text files. Knockoffaly assigns special types to them and special icons (in System 7) to help you and Knockoffaly tell them apart. The Open Config routine can only open files of type CNFG, and the Open Game routine can only open files of type KNKF. Unfortunately, this renders them so that Text editors and Word Processors are unable to open them. There is a routine accessible from the Knockoffaly and the KnockEditor that lets you change them back to TEXT files in case you want to see what's in them. It is called [fudgeType](#). There is no menu path to it; you have to type “fudgeType” into the message box. It presents you with a standard open file dialog, and allows you to choose to set the file type to TEXT, KNKF, or CNFG. When you're done looking at the file, be sure to fudgeType it back to what it was, or Knockoffaly won't be able to open it. Of course if you tell it a KNKF is a CNFG or vice versa, Knockoffaly will belch error messages. With power comes responsibility.

SETTING DEFAULT GAME

I debated long and hard about this one. Since you have so much flexibility, it is of course possible for you to have a bad configuration. I would not want casual players of Knockoffaly to inadvertently wipe out their very well tested default configuration

permanently with an invalid, or just plain bad, game. So you have the option to change the default, but you have to know what you're doing in order to do it. There is no menu path to it. In Knockoffaly, bring up the message box with the back door (Command-Click the right most die button.) In the message box, type **SetDefault**. The current configuration will permanently replace the default one. Although I encourage you to copy and share the Knockoffaly game and the Resource Stack (even altered), **you are specifically prohibited from sharing any altered versions of Knockoffaly itself including copies with the default overwritten!**

Keep a backup of the original to give to your friends. If you want them to try your configuration, share it with them as a configuration file.

GAME CONFIGURATION

You will probably not need to text edit in game files too much, since it is easier to just have Knockoffaly create a new one if you don't like the ones you have. The main purpose of editing game files will probably be to alter your balance in games where cheating is not allowed. Whatever. Game file format is included here, for completeness.

Line 1: The signature, the number of groups, and the player number of the owner of each group. -1 if the group is not totally owned by a single player yet. Separated by a space.

The next forty lines are game data about each square. Each line consists of three items, separated by comma:

Item 1: the owner of this square

Item 2: the number of improvements on it

Item 3: "M" if it is mortgaged, empty otherwise.

The next line is p: the number of players

The next line is the number of the current player

The next p lines are player descriptors.

The next line is a blank separated list of indices relative to their respective start indices of each of the Fate sets. The number of items here is dependent upon the number of fate sets.

The remaining lines are blank separated lists of shuffled indices of the fate sets. The exact number of lines depends on how many fate sets there are. The exact number of items in each line depends on the number of fate twists in that set.

Player Descriptor format:

A player descriptor consists of 13 items separated by commas.

Item 1: The player's name

Item 2: a blank separated list of square numbers that are owned by this player.

Item 3: Assets. (not used)

Item 4: "In Jail" if the player is incarcerated

Item 5: contains "Chance" and a the number of the fate set it came from if the player has a get out of jail free "card."

Item 6: the Try number for getting out of jail.

Item 7: The player's cash

Item 8: the player's current location

item 9: active or bankrupt

item 10: money owed if player has negative balance.

item 11: the number of the player this player owes money to. This is who gets his properties if he declares bankruptcy

item 12: offset of the player's token from the first token.

item 13: Dummy, not used.