NethackDS Manual

Brett Kosinski

March 6, 2024

Contents

1 The Introduction

NetHackDS is a port of the game NetHack to the Nintendo DS™video game system. For those unfamiliar with the game NetHack, it might be best to first familiarize yourself with the basic game itself. You can find detailed information on NetHack, the game, at the project home page: `http://www.nethack.org`.

NetHack is typically referred to as a roguelike game. In this sense, it features a hero wandering around in a virtual world, acquiring equipment, items, and money, in the hopes of completing some goal or quest. In the case of NetHack, the goal is to retrieve the Amulet of Yendor from the depths of the dungeon, whereupon you must return and sacrifice the amulet to your god. It sounds simple on the surface, but the path to ascension is a long and arduous one, frought with many perils. Not the least of which is the destructive nature the game often has on grades, job performance, and so forth.

Traditionally, NetHack is played on a computer, and as such, makes extensive use of the keyboard as a vehicle for user interaction. This makes translating the game to a portable device particularly challenging as compared to other game porting efforts. However, I hope I have succeeding in creating something that is both easy and enjoyable to play, both for experienced NetHack players, as well as neophytes.

2 The User Interface

2.1 Overview

The game display in NetHackDS is divided between the two screens of the Nintendo DS™as follows:

1. On the bottom screen is rendered the game map, where a viewport into the world is rendered. It is here that the dungeon is displayed, along with the hero, monsters, items, etc.
2. On the top screen is rendered the status display, where information about the hero is shown, as well as a miniature rendering of the current level.

Of course, the bottom screen is also used to interact with the user in a number of other ways. Most importantly, it is on the bottom screen that the command window is rendered, through which the user may direct their actions in the game. Additionally, menus and text windows are displayed there.

2.2 Game Map

The game map displays a subset of the current visible level. Among the notable features are:

1. The hero, who is highlighted by either a graphical or textual cursor (depending on configuration).
2. Dungeon features, such as walls, floors, doors, hallways, furniture, etc.
3. Monsters, both friendly and unfriendly.
4. Items, such as gold, scrolls, potions, wands, armor, weapons, and so on and so forth.

Note, the visible portion of the current level may be panned by either dragging with the stylus, or using keys bound to the various map panning functions, as described in section 3.1.

Additionally, the user may directly control the game in a number of ways using the touch-screen. For more details, see section 3.2.

2.3 Status Display

The status display, present on the top screen, provides various information about the player and the game. Among the key features visible, there is:

1. The minimap, which shows a scaled down view of the current level map.
2. The player status lines, which provide information about various player attributes, as well as the current dungeon level, and so forth.
3. The game message window. This presents various information about events occuring within the game.

2.3.1 The Minimap

The minimap shows a wealth of information in a compact display. On it you will see rendered a red box, which represents the current viewport visible on the bottom screen, as well as any dungeon features visible or discovered, presented as coloured regions. The various colours correspond to different features on the map, and include:

Green	The Hero
Red	Monsters
Purple	Pets
Dark Yellow	Doors
Black	Floors
Blue	Stairs
Light Yellow	Altars
White	Walls, Furniture, etc.

2.4 Menus and Text

There are a few basic user interface elements present in NetHackDS. Among the most important are the menu and text window displays. As you would expect, these are used to present modal, scrollable lists of textual information or items that the user may select. Typically, these displays are activated in response to actions taken by the user (for example, displaying the item inventory).

Controls in menu and text windows are fairly simple:

Right	Page Down
Left	Page Up
Up	Move Item Selector Up
Down	Move Item Selector Down
X	Select Item or Increment Counter
Y	Deselect Item or Decrement Counter
A	Dismiss Menu/Text Display
B	Cancel Menu/Text Display
Select	Select All Items
Select+R or Select+L	Deselect All Items

Note, menus are often divided into subsections by section titles. Selecting/deselecting these titles will affect all items within that subsection.

Additionally, menu and text windows can be interacted with using the stylus. In particular:

- Tapping an item in a menu will select it or increment it's counter.
- Tapping an item while holding L or R will deselect it or decrement it's counter.
- Tapping section headers (eg, "Armor", "Weapons", etc) will act on all items in that section.
- Tapping the arrows at the top/bottom of the screen will page the display up/down.
- Tapping the checkmark icon will dismiss the menu (equivalent to pressing A).
- Tapping the X icon will cancel the menu (equivalent to pressing B).

2.5 The Command Window

Like all roguelikes, NetHack's gameplay is centered around a vocabulary of verbs. These verbs define all actions which can be taken in the game, and in NetHack, this vocabulary is large. Very large. Of course, NetHack's traditionally keyboard-based control scheme makes such a game possible, as verbs can be mapped to various keys on the keyboard. However, on a portable device, things are not so easy, and so the command window provides an interface to the set of verbs available in the game.

In the default configuration, the command window is activated by pressing and holding the left trigger button. This brings up a display of verbs organized in columns, which may then be selected by using the joypad or the touchscreen. When the joypad is used, the direction keys can be used to highlight verbs, and selection is done by pressing the X or A buttons.

Additionally, actions may be repeated some number of times based on a count provided by the user. To indicate a repeated action is desired, tap-and-hold the action using the stylus until a keyboard appears. Then type in the repeat count desired, and press the Enter button on the keyboard.

3 The Control Scheme

3.1 The Joypad

The default NetHackDS control scheme is set up as follows:

Up	Move Up
Down	Move Down
Left	Move Left
Right	Move Right
R+Up	Pan Map Up
R+Down	Pan Map Down
R+Left	Pan Map Left
R+Right	Pan Map Right
A	Pick Up
B	Search
X	Open
Y	Kick
L	Display Command Window

3.2 The Touchscreen

In addition to using the joypad, the user may also interact with the game map using the touchscreen. For example:

- Tapping on an adjacent, empty square will cause the hero to move into that square.
- Tapping on a non-adjacent square will cause the hero to move to that location (or as close as possible to it).
- Tapping on an adjacent, non-empty square will cause the hero to interact with the item in that square.
- Tapping on the hero while they are standing over something will cause the hero to interact with the item in question.

In the last two cases, the type of interaction will depend upon the item in question. For example:

- Tapping an adjacent monster will cause the hero to attack it.
- Tapping an adjacent door will cause the hero to attempt to open it.

- Tapping the hero while standing over stairs will cause the hero to climb them.
- Tapping the hero while standing over a container (box, bag, etc), will cause them to try and loot it.
- Tapping the hero while standing over a sink or fountain will cause them to attempt to drink from it.

And so forth. Lastly, in some instances, the game may prompt the user for directional input in order to complete a command (eg, zapping a wand, throwing a weapon, etc). In this case, the map can be tapped in order to indicate the direction. Specifically:

- Tapping an adjacent square will input the corresponding direction (ie, tapping the upper-right square will indicate a direction of up-right).
- Tapping the hero will indicate the hero (equivalent to '.').
- Tapping-and-holding on the hero (press and hold the stylus on the hero) will indicate the floor (equivalent to '>').

4 The Configuration System

Because tastes among users strongly vary regarding control schemes, display features, and so forth, NetHackDS is, above all, designed to be highly customizable. As such, the game provides a deeply flexible keybinding system, as well as a wide array of features which may be enabled to disabled as the user desires, in addition to the standard configurable features available in NetHack.

4.1 The Configuration File

Much of the configuration in NetHackDS is managed through a file called "defaults.nh", present in the NetHackDS directory. This text file, which can be modified with a simple text editor, exposes a wide array of configurable settings. Here I will limit the discussion to major NetHack options relevant to NetHackDS, and NetHackDS-specific settings. The remaining options are documented on the NetHack website.

The general format of the configuration file is a set of directives and values, separated by an equals sign ('='). The directives may be one of:

OPTIONS	Game option directive, used to turn on or off various features.
MENUCOLOR	Used to define colors for menucolor mode.
CHORDKEYS	Defines the set of chord keys (for key binding, see below).
HELPLINE1 HELPLINE2	Controls the key help displayed in the status window.

Note, multiple copies of each directive may be present in the file, and are all honoured.

In the case of OPTIONS lines, multiple game options may be specified. Options are separated by commas, may be preceeded by an exclamation point ('!') to indicate negation, and for compound options (options which take an operand), the option name and value are separated by a colon (':') character. For example:

OPTIONS=autopickup,pickup_types:$,!menucolors

In this case, we have three options defined. The first is a simple boolean option, the second is an option taking a value, in this case the string '$', and the third is a negated boolean option.

4.1.1 Map Display Settings

Traditionally, NetHack was played on Unix terminals, and as such, the entire game was rendered in simple ASCII graphics. However, as the game progressed, optional tile-based graphics were added, which are typically used in Windows, X11, and other graphical ports. That said, purests such as myself still prefer the ASCII graphics version, and as such, NetHackDS supports both ASCII and tile-based graphics modes, the latter being the default setting.

The following options affect how the map is rendered:

tile_file:<file_name>	A 16- or 256-color BMP file containing the graphics tiles to use.
tile_width:<width>	The width of each tile (must be a multiple of 8).
tile_height:<height>	The height of each tile (must be a multiple of 8).
color	In text mode, specifies that color should be used.
ibmgraphics	In text mode, specifies that IBM graphics characters should be used.
cursor:<cursor_mode>	In text mode, how to display the cursor. 0=always, 1=not on hero, 2=never.

Note, text mode is enabled simply by omitting the tile_file option from the configuration.

4.1.2 User Interface Settings

There are a number of settings available which can be used to fine-tune the user interface.

Option Name	Description	Default
cmdwindow	Determines whether or not the command window or a virtual keyboard should be used for command input.	on
holdmode	Determines whether or not the command key toggles, or simply displays, the command window.	on
doubletap	If enabled, item selection with the stylus requires two taps.	off
hpmon	If enabled, colours the hitpoint monitor based on level of damage.	on
menucolors	If enabled, allows user-customizable colouring of menu items.	on
mapcolors	If enabled, certain dungeon features are coloured different (eg, walls in the gnomish mines, etc).	off

4.1.3 Compass Mode

Compass Mode is an alternative stylus control scheme, inspired by (and code swiped from) the iRogue project. In this mode, one can consider the screen as being divided into a set of eight wedges, centered around the hero or the center of the screen, depending on the configuration settings. Tapping within a given wedge indicates the hero should move in that direction. Tapping near the compass center results in a single step. Tapping further away results in running.

Compass Mode is controlled using the 'compassmode' option, which takes one of the following values:

0	Disabled
1	Relative Mode (compass is centered on the hero)
2	Absolute Mode (compass is centered on the middle of the screen)

4.2 The Key Binding System

As has been alluded to previously, NetHackDS sports an extremely flexible keybinding system, which is controlled through a combination of settings in the configuration file, as well as an in-game key configuration system.

4.2.1 Key (har har) Concepts

On the standard PC keyboard, keys such as Alt and Shift are unusual. These keys, when pressed, trigger no action on their own[1]. Instead, they alter the behaviour of other keys.

[1]Okay, that's not strictly true. For example, in the Windows operating system, Alt, when pressed then released, focuses the menu in the current window. Similarly, in NetHackDS, chord keys may have actions bound to them if they are pressed then released.

Similarly, the NetHackDS key binding system differentiates keys based on whether or not they are "chord keys" (to Unix users, these are more familiarly known as "meta" keys), keys which can be combined with other keys for binding purposes, or regular keys.

But, unlike a PC, which has a strict set of chord keys, NetHackDS allows the flexibility to control which keys are chord keys, in addition to controlling which commands are bound to which keys/key combinations, which allows for incredibly flexible key binding.

Lastly, there is one "special" key in NetHackDS, the command key. It is this key which brings up the command window. And, again, there is an option for controlling which key is mapped to this functin.

4.2.2 Key Configuration Settings

Key configuration in NetHackDS is controlled by one configuration directive, CHORDKEYS, and one configuration option, cmdkey. CHORDKEYS specifies the list of chord keys, and is defined as a comma-separated list of key names. For example:

CHORDKEYS=up,down,left,right

would define the various directions on the D-Pad as chord keys (and is essentially identical to the DSCrawl key arrangement). Valid key names include: up, down, left, right, a, b, x, y, select, start, r, and l.

Additionally, the cmdkey option takes, as a value, a key name, and defines which key is used to open the command window. For example:

OPTIONS=cmdkey:start

would define the Start button as the key which brings up the command window.

4.2.3 Key Binding

The actual process of binding keys (mapping keys or key combinations to actions) is done in-game using the "Key Config" action in the command menu (or F1, if the command window has been disabled). Upon selecting the "Key Config" action, the user is asked to press a key or key combination for binding purposes. This can be any key or a combination of chord key(s) and a regular key. After the key or key combination to be bound is pressed, a menu is presented offering a number of command subgroups, including:

Movement	Various Hero movement operations, including regular movement, running, and fight operations
Game Command	Standard game commands. Brings up the command window (or virtual keyboard, if the command window is disabled).
Toggle Option	Boolean game options. Binding a key to one of these causes the option to be toggled when the key is pressed.
Map Panning	Commands for moving the map viewport around.
No Command	Effectively "unbinds" the key in question.

Note, when binding game commands, repeat and extended commands are perfectly valid selections.

4.3 Other Data Files

There are a number of data files which NetHackDS makes use of. The following is a list of those files along with a simple description of their function and format.

font.bdf	BDF-format font file used for menus, text, status display, etc.
map.bdf	BDF-format font file used for the text-based map graphics. Note, the font characters in this file are padded out to a multiple of 8 when used in-game.
kbd.pal	The palette file for the virtual keyboard graphics.
map.pal	The palette file for the text-based map graphics.
minimap.pal	The palette file for the minimap.
text.pal	The palette file used for menus, text, status display, etc.

4.3.1 Palette File Format

The palette files used in NetHackDS are simple text files. Any blank lines, or lines prefixed with a pound ('#') are ignored. All other lines must be bare six-digit hexadecimal values (lower-case characters only) representing RGB triplets.

5 The Credits and Acknowledgements

NetHackDS wouldn't exist without the invaluable contributions from many others. Below are just a few without whom this port would not exist:

- This code has it's roots in the first NetHackDS port, created by Stuart Pernsteiner (aka "Wosret"). While the actual implementation is essentially a rewrite, I borrowed much (such as the keyboard handling code), and benefitted greatly from his initial porting effort.

- Of course, there's the NetHack DevTeam, without which this port wouldn't exist (and I would've spent many hours doing far more productive things with my time).
- The Devkitpro folks, and all those who've contributed to the project. In particular, libnds has been unsurprisingly invaluable.
- Chism for his work on libfat.
- Masscat for his GDB stub. I probably would've given up without it.
- The dswifi developers, without whom Masscat's stub would've been useless.
- Oddly, the Foobillard developers. I snagged their BDF font rendering and PPM handling code. :)
- Stumpy, for his superb ANSI fontsets, from which I scavenged.
- Tobias Jung, for making ProFont available.
- The authors of the surprisingly portable PCRE, without which menucolors support would've been a non-starter.
- Many users for providing valuable feedback and feature suggestions.