

NAME

dgenrc — file containing settings for *dgen(1)*

SYNOPSIS

\$HOME/.dgen/dgenrc

DESCRIPTION

The file *\$HOME/.dgen/dgenrc* is parsed by *dgen(1)* when the emulator is started. It is used to set controller keys, as well as other characteristics of the emulation. The contents of this file may be overridden with the contents of another similarly-formatted file, via the *-r* commandline switch.

FILE FORMAT

Each rc file consists of an unlimited number of lines, which each have the format 'fieldname = value'. A line may also be a comment, if it begins with the hash mark (#) character.

Each *fieldname* is prepended by a name, which identifies the type of this field:

*key_** A key value. May be set to a key identifier listed in the “KEY IDENTIFIERS” section below.

*joy_** A joystick/joypad value. May be set to a joystick identifier listed in the “JOYSTICK IDENTIFIERS” section below.

*mou_** A mouse action. May be set to a mouse identifier listed in the “MOUSE IDENTIFIERS” section below.

*bool_** A boolean value. "false", "no", and "0" values are taken as false, while "true", "yes", and any number except 0 are taken as true.

*int_** An integer value, greater than or equal to 0.

*str_** A string value, can be empty.

Some fields take special value sets, which are addressed in their respective sections. None of the field names or values are case-sensitive.

The fields fall under a few basic groups. They are listed below, with their default values in brackets ([]):

All of them can be modified interactively from the prompt, as described in *dgen(1)*.

CONTROLLERS

key_pad1_up [up]

key_pad1_down [down]

key_pad1_left [left]

key_pad1_right [right]

key_pad1_a [a]

key_pad1_b [s]

key_pad1_c [d]

key_pad1_x [q]

key_pad1_y [w]

key_pad1_z [e]

key_pad1_mode [backspace]

key_pad1_start [return]

Map keys to the first Genesis controller. Each of these fields has a corresponding "key_pad2" field, to map to the second controller.

joy_pad1_up [joystick0-axis1-min]
 joy_pad1_down [joystick0-axis1-max]
 joy_pad1_left [joystick0-axis0-min]
 joy_pad1_right [joystick0-axis0-max]
 joy_pad1_a [joystick0-button0]
 joy_pad1_b [joystick0-button3]
 joy_pad1_c [joystick0-button1]
 joy_pad1_x [joystick0-button6]
 joy_pad1_y [joystick0-button4]
 joy_pad1_z [joystick0-button5]
 joy_pad1_mode [joystick0-button9]
 joy_pad1_start [joystick0-button8]

Map joystick/joypad buttons to the first Genesis controller. Each of these fields has a corresponding "joy_pad2" field, to map to the second controller.

bool_joystick [true]

Use joysticks to emulate the controllers. Note that the keyboard keys will still work if this value is set. This field is only available if you have joystick support enabled.

int_mouse_delay [50]

Number of milliseconds to wait after the last mouse motion event to release buttons bound to such events.

USER-DEFINED BINDINGS

bind_{keysym} *action*

bind_{joypad} *action*

bind_{mouse} *action*

Defines a new keyboard, joystick/joypad or mouse binding to an arbitrary action. These variables use the *keysym* format as defined in “KEY IDENTIFIERS”, the *joypad* format as defined in “JOYSTICK IDENTIFIERS”, or the *mouse* format as defined in “MOUSE IDENTIFIERS”. When *action* is prefixed with "key_", "joy_" or "mou_", it becomes an alias to the corresponding variable in “CONTROLLERS”. Otherwise, it is interpreted as if entered at the prompt, and can be used to modify variables or execute commands (see *dgenrc(5)* for more information). No bindings are defined by default.

bind_{keysym|joypad|mouse} [{keysym|joypad|mouse} [...]] *action*

Alternate syntax that supports combining several identifiers to perform *action*. Controller types can be mixed. Identifiers are separated by spaces. To avoid syntax errors, spaces must be properly escaped or quoted.

AUDIO

bool_sound [true]

Enable the sound subsystem.

int_soundrate [44100]

Sound frequency to play at, in hertz (Hz).

int_soundsegs [8]

Number of sound segments to use for sound buffering. Lower values guarantee low latency. Increment this only if the sound becomes choppy.

int_soundsamples [0]

Size of the system sound buffer, in samples (samples are the sound unit, sound rate is how many of them are played every second). Specifying 0 automatically chooses the safest value. If you experience sound issues int_soundsegs is unable to solve, try to change this. Increasing it will cause noticeable audio lag (it is unfortunately often required on slower machines).

int_volume [100]

Volume level, in percent. Values above 100 cause distortion.

key_volume_inc [=]

key_volume_dec [-]

joy_volume_inc []

joy_volume_dec []

mou_volume_inc []

mou_volume_dec []

Bindings for volume control.

bool_mjazz [false]

MJazz option - puts 2 more FM chips in the Megadrive for a sort of 22 channel sound boost. Can sound good. Slows things down a lot.

VIDEO

int_depth [0]

Color depth (bits per pixel). Allowed values are 0 (automatic), 8, 15, 16, 24 and 32. Ignored in OpenGL mode.

int_width [-1]

int_height [-1]

Desired window width and height.

bool_opengl [true]

Use the OpenGL renderer, if it is available.

bool_opengl_stretch [true]

Let OpenGL stretch the screen.

bool_opengl_linear [true]

Use GL_LINEAR for textures filtering instead of GL_NEAREST.

bool_opengl_32bit [true]

Use 32 bit textures. They require more memory but are usually faster than 16 bit textures.

bool_opengl_square [false]

Use square textures. Wastes a lot of memory but may solve OpenGL initialization failures.

bool_fullscreen [false]

Try to run fullscreen, if possible.

int_scale [-1]

int_scale_x [-1]

int_scale_y [-1]

Amount by which to scale the rendered screen from the default resolution. See scaling filters.

bool_aspect [true]

Retain original aspect ratio when resizing window.

key_fullscreen_toggle [alt-enter]

joy_fullscreen_toggle []

mou_fullscreen_toggle []

Button to toggle fullscreen mode (this may do nothing if SDL doesn't support fullscreen toggling on your platform.)

int_info_height [-1]

Height of the text area at the bottom of the screen, in pixels. This also affects the font size. Values smaller than the minimum font size make DGen redirect text to stdout instead. The default value of -1 makes DGen choose the proper height.

bool_fps [false]

Display the current number of frames per second.

bool_buttons [false]

Display pressed buttons. Can be used to help configuring them.

bool_swab [false]

Swap bytes in the video output. Sometimes useful when video output is located on a different system. This is implemented as a CTV filter which must be compiled-in to work.

bool_doublebuffer [true]

Toggle double buffering. Enabling this should prevent screen tearing from happening. Disabling this may improve the number of displayed frames per second on some systems.

bool_screen_thread [false]

When enabled, a separate thread is created to offload the displaying of frames. This is only useful on slower machines where flipping video buffers takes time, especially when V-sync is enabled and doing so blocks until the next frame without consuming CPU time (sometimes the case when bool_doublebuffer is enabled). This currently has no effect when OpenGL is enabled and only works if multi-threading support is compiled-in.

SAVE STATES

key_slot_X [X]

joy_slot_X []

mou_slot_X []

Sets the current save-state slot to number X.

key_slot_next [f8]

joy_slot_next []

mou_slot_next []

Switch to the next save-slot.

key_slot_prev [f7]

joy_slot_prev []

mou_slot_prev []

Switch to the previous save-slot.

key_save [f2]

joy_save []

mou_save []

Saves state to the current slot.

key_load [f3]

joy_load []

mou_load []

Loads state from the current slot.

MISCELLANEOUS KEYS

key_fix_checksum [f1]

joy_fix_checksum []

mou_fix_checksum []

Fixes the checksum value. Some older games will freeze with a red screen if the ROM has been hacked or modified with Game Genie codes. If it does, pressing this, and resetting should fix the problem.

key_quit [escape]

joy_quit []

mou_quit []

Exit DGen or switch to the next ROM on the command-line.

key_craptv_toggle [f5]

joy_craptv_toggle []

mou_craptv_toggle []

Toggles Crap-TV image filters. These filters aren't available in 8 bpp mode.

key_scaling_toggle [f6]

joy_scaling_toggle []

mou_scaling_toggle []

Toggles scaling algorithms. See scaling_startup below.

key_reset [tab]

joy_reset []

mou_reset []

Restart the Genesis emulation.

key_cpu_toggle [f11]

joy_cpu_toggle []

mou_cpu_toggle []

Switch CPU emulators. The x86 assembly CPU emulator StarScream is fast, but has glitches which affect a few games. Switching to the slower Musashi core will fix these problems, at a speed penalty.

key_z80_toggle [f10]

joy_z80_toggle []

mou_z80_toggle []

Switch Z80 emulators. MZ80 is a bit faster than CZ80, particularly in its assembly version (only available for x86), but CZ80 works with more games. This key can also disable Z80 emulation.

key_stop [z]

joy_stop []

`mou_stop []`
 Pause emulation, so you can concentrate on real life for a few seconds. :)

`key_game_genie [f9]`

`joy_game_genie []`

`mou_game_genie []`
 Enter a Game Genie or Hex code. This key also works in stopped mode.

`key_screenshot [f12]`

`joy_screenshot []`

`mou_screenshot []`
 Take a screenshot. Not available in 8 bpp mode.

`key_prompt [:]`

`joy_prompt []`

`mou_prompt []`
 Pause emulation and display interactive prompt. Also works in stopped mode.

`key_debug_enter ['']`

`joy_debug_enter []`

`mou_debug_enter []`
 Break into the debugger. Only meaningful if debugger support is compiled-in.

PREFERENCES

`int_hz [60]`
 Video refresh rate. The default is 60 as in NTSC consoles.

`bool_pal [false]`
 When true, a PAL console is emulated. This should be used in combination with `int_hz` above for 50Hz emulation.

`region ['']`
 U for America (NTSC), E for Europe (PAL), J for Japan (NTSC), X for Japan (PAL), or empty space for autodetection (the default). Overrides `bool_pal` and `int_hz`.

`str_region_order [JUEX]`
 Regions DGen is allowed to emulate when autodetection is enabled, ordered by preference.

`emu_m68k_startup [musa]`
 Useful when both Musashi and StarScream are compiled-in. This option selects the default emulator to use ("musa" for Musashi, "star" for StarScream, "none" for neither). See `key_cpu_toggle`.

`emu_z80_startup [cz80]`
 Useful when both CZ80 and MZ80 are compiled-in. This option selects the default emulator to use ("cz80", "mz80" or "none", if you want to disable it altogether). See `key_z80_toggle`.

`bool_autoload [false]`
 Automatically load the saved state from slot 0 when DGen starts.

`bool_autosave [false]`
 Automatically save the saved state to slot 0 upon exit. Setting both of these fields true, you can exit DGen, and automatically start a game where you left off when you start it again.

`bool_autoconf [true]`
 Automatically dump the current configuration to `dgenrc.auto` before exiting. This file is always loaded before `dgenrc` at startup.

`bool_frameskip` [true]

Automatically skip frames, when it is necessary to maintain proper emulation speed. You may want to disable sound or set `int_nice` to a nonzero value when setting this to false.

`int_nice` [0]

If set to a non-zero value, DGen will call *usleep*(3) with the specified parameter after rendering each frame. This will slow the program down (if it is running too fast on your computer), and allow the operating system to reclaim some CPU time.

`ctv_craptv_startup` [off]

CTV filter to use by default. Available filters are "blur", "scanline", "interlace" and "swab".

`scaling_startup` [stretch]

Scaler to use when display resolution is larger than original screen. Available filters are "stretch", "scale", "hqx", "hqx stretch", "scale2x", "scale2x stretch" and "none".

`bool_show_carthead` [false]

Show cartridge header info at startup.

`bool_raw_screenshots` [false]

Generate unfiltered screenshots.

`str_rom_path` ["roms"]

Directory where DGen should look for ROMs by default. It's relative to DGen's home directory, unless an absolute path is provided.

DEBUGGING

`bool_vdp_hide_plane_a` [false]

`bool_vdp_hide_plane_b` [false]

`bool_vdp_hide_plane_w` [false]

`bool_vdp_hide_sprites` [false]

Hide various planes during frame rendering. Require VDP debugging to be compiled-in.

`bool_vdp_sprites_boxing` [false]

`int_vdp_sprites_boxing_fg` [0xffff00] (yellow)

`int_vdp_sprites_boxing_bg` [0x00ff00] (green)

Toggle sprites boxing and configure its colors. Require VDP debugging to be compiled-in. "fg" is for sprites with the high priority bit set, "bg" is for the others. Color format is 0xRRGGBB.

EXAMPLES

See the file "sample.dgenrc" in the DGen/SDL distribution.

KEY IDENTIFIERS

A key identifier can have the prefixes "shift-", "ctrl-", "alt-" and "meta-", or any combination thereof, to require that the specified modifier be pressed in combination with the key. For example, the identifier "alt-enter" would correspond to holding down the Alt key while pressing Enter.

The "shift-" modifier only works with keys that don't generate symbols (such as arrow keys). Otherwise their UTF-8 representation must be used directly.

The numbers "0" through "9" ("kp_0" through "kp_9" for the numeric keypad), letters "A" through "Z", and function keys "F1" through "F12" map to their key equivalents. In addition, the following identifiers map to their similarly-labeled key counterparts. Identifiers on the same line map to the same key:

escape	
backspace	
tab	
capslock	caps_lock
lshift	shift_l

rshift	shift_r
lctrl	ctrl_l
lmeta	meta_l
lalt	alt_l
ralt	alt_r
rmeta	meta_r
rctrl	ctrl_r
return	enter
space	
scrollock	scroll_lock
numlock	num_lock
insert	
home	
page_up	pageup
delete	
end	
page_down	pagedown
left	
right	
up	
down	
kp_home	
kp_up	
kp_pageup	kp_page_up
kp_left	
kp_right	
kp_end	
kp_down	
kp_pagedown	kp_page_down
kp_insert	
kp_delete	kp_period
kp_enter	
kp_divide	
kp_minus	
kp_multiply	
kp_plus	

JOYSTICK IDENTIFIERS

Like key identifiers, joystick (or joypad) identifiers describe a joystick event. Each detected joystick is numbered starting from 0. Three different event types are supported.

Buttons: joystickX-buttonY

For joystick/joypad number X, button number Y. "button" can be abbreviated as "b".

Axes: joystickX-axisY-min, joystickX-axisY-max

For joystick/joypad number X, axis number Y, and its position, which is either "min" (also "n", "negative") or "max" (also "p", "positive"). "axis" can be abbreviated as "a".

Hats: joystickX-hatY-{direction}

For joystick/joypad number X, hat number Y and direction. "hat" can be abbreviated as "h". Possible directions are "up", "right", "down", "left".

MOUSE IDENTIFIERS

These identifiers describe a mouse event. Each detected mouse is numbered starting from 0. Two different event types are supported.

Buttons: mouseX-buttonY

For mouse number X, button number Y. "button" can be abbreviated as "b".

Motions: mouseX-{direction}

For mouse number X and direction. Possible directions are "up", "right", "down", "left".

SEE ALSO

dgen(1)

AUTHORS

This manual page was written by Joe Groff <joe@pknet.com>.

Updated by zamaz <zamaz@users.sourceforge.net>.