

dwm: config.h

config.def.h

```

/* See LICENSE file for copyright and license details. */

/* appearance */
static const char font[] = "-*-terminus-medium-r-*-*-14-*-*-*-*-*-*";
//couleur des bordures de fenêtres inactives :
static const char normbordercolor[] = "#515151";
//couleur d'arrière plan barre / tags inactifs / dwmstatus:
static const char normbgcolor[] = "#191919";
//couleur de police des tags inactifs et de police de dwmstatus:
static const char normfgcolor[] = "#FFE400";
//couleur des fenêtres actives:
static const char selbordercolor[] = "#D70751";
//couleur d'arrière plan des tags actifs
static const char selbgcolor[] = "#515151";
//couleur de police des tags actifs:
static const char selfgcolor[] = "#33FF00";

static const unsigned int borderpx = 1; /* border pixel of windows */
static const unsigned int snap = 32; /* snap pixel */
static const unsigned int systrayspacing = 1; /* systray spacing */
static const Bool showsystray = True; /* False means no systray */
static const Bool showbar = True; /* False means no bar */
static const Bool topbar = True; /* False means bottom bar */

/* tagging */
static const char *tags[] = { "Ω", "Δ", ">", "Π", "¿", "√", "≈", "χ", "[]" };
/* static const char *tags[] = { "sog", "www", "mx", "grf", "mzk", "xmp", "wrk", "dwl", "stk" }; */

static const Rule rules[] = {
    /* class      instance    title        tags mask    isfloating
monitor */
    { "Dwb",      NULL,        NULL,        1 << 1,      False,
-1 },
    { "Iceweasel", NULL,        NULL,        1 << 1,      False,
-1 },
    { "Mutt",      NULL,        NULL,        1 << 2,      False,
-1 },
    { "Mcabber",   NULL,        NULL,        1 << 5,      False,
-1 },

```

```

    { "MC",                NULL,        NULL,        0,        False,
-1 },
    { "MOC",               NULL,        NULL,        1 << 4,   False,
-1 },
    { "Gimp",              NULL,        NULL,        1 << 3,   False,
-1 },
    { "Inkscape",          NULL,        NULL,        1 << 3,   False,
-1 },
};

/* layout(s) */
static const float mfact      = 0.70; /* factor of master area size
[0.05..0.95] */
static const int nmaster      = 1;    /* number of clients in master
area */
static const Bool resizehints = False; /* True means respect size hints
in tiled resizals */

static const Layout layouts[] = {
    /* symbol      arrange function */
    { "[]=",       tile },    /* first entry is default */
    { "><=",       NULL },    /* no layout function means floating
behavior */
    { "[M]",       monocle },
};

/* key definitions */
#define MODKEY Mod4Mask
#define TAGKEYS(KEY,TAG) \
    { MODKEY,       KEY,      view,           {.ui = 1 \
<< TAG} }, \
    { MODKEY|ControlMask, KEY,      toggleview,      {.ui = 1 \
<< TAG} }, \
    { MODKEY|ShiftMask,   KEY,      tag,            {.ui = 1 \
<< TAG} }, \
    { MODKEY|ControlMask|ShiftMask, KEY,      toggletag,       {.ui = 1 \
<< TAG} },

/* helper for spawning shell commands in the pre dwm-5.0 fashion */
#define SHCMD(cmd) { .v = (const char*[]){ "/bin/sh", "-c", cmd, NULL }
}

/* commands */
static const char *dmenucmd[] = { "dmenu_run", "-fn", font, "-nb",
normbgcolor, "-nf", normfgcolor, "-sb", selbgcolor, "-sf", selfgcolor,
NULL };
static const char *termcmd[]  = { "x-terminal-emulator", NULL };

static Key keys[] = {
    /* modifier                                key          function          argument
*/

```

```

    { MODKEY,
dmenucmd } },
    { MODKEY|ShiftMask,
termcmd } },
    { MODKEY,
    { MODKEY,
} },
    { MODKEY,
} },
    { MODKEY,
} },
    { MODKEY,
} },
    { MODKEY,
-0.05} },
    { MODKEY,
+0.05} },
    { MODKEY,
    { MODKEY,
    { MODKEY|ShiftMask,
    { MODKEY,
&layouts[0]} },
    { MODKEY,
&layouts[1]} },
    { MODKEY,
&layouts[2]} },
    { MODKEY,
    { MODKEY|ShiftMask,
    { MODKEY,
~0 } },
    { MODKEY|ShiftMask,
~0 } },
    { MODKEY,
} },
    { MODKEY,
} },
    { MODKEY|ShiftMask,
} },
    { MODKEY|ShiftMask,
} },
    TAGKEYS(
0)
    TAGKEYS(
    TAGKEYS(
2)
    TAGKEYS(
3)
    TAGKEYS(
4)
    TAGKEYS(
    TAGKEYS(

```

```

XK_p,      spawn,      {.v =
XK_Return, spawn,      {.v =
XK_b,      togglebar,  {0} },
XK_k,      focusstack, {.i = +1
XK_j,      focusstack, {.i = -1
XK_i,      incnmaster, {.i = +1
XK_d,      incnmaster, {.i = -1
XK_h,      setmfact,   {.f =
XK_l,      setmfact,   {.f =
XK_Return, zoom,       {0} },
XK_Tab,    view,       {0} },
XK_c,      killclient, {0} },
XK_t,      setlayout,  {.v =
XK_f,      setlayout,  {.v =
XK_m,      setlayout,  {.v =
XK_space,  setlayout,  {0} },
XK_space,  togglefloating, {0} },
XK_agrave, view,       {.ui =
XK_agrave, tag,        {.ui =
XK_comma,  focusmon,   {.i = -1
XK_period, focusmon,   {.i = +1
XK_comma,  tagmon,     {.i = -1
XK_period, tagmon,     {.i = +1
XK_ampersand,
XK_eacute,
XK_quotedbl,
XK_apostrophe,
XK_parenleft,
XK_minus,
XK_egrave,

```

```

TAGKEYS(                                XK_underscore,
7)
TAGKEYS(                                XK_ccedilla,
8)
    { MODKEY|ShiftMask|Mod1Mask,      XK_q,      quit,      {0} },
      { ControlMask|Mod1Mask,          XK_i,      spawn,
SHCMD("dwb") },
      { ControlMask|Mod1Mask,          XK_m,      spawn,
SHCMD("sakura --class=Mutt -x 'mutt -y'") },
      { ControlMask|Mod1Mask,          XK_p,      spawn,
SHCMD("st -c Mcabber -e mcabber") },
      { ControlMask|Mod1Mask,          XK_g,      spawn,
SHCMD("gimp") },
      { 0,                             0x1008ff2d, spawn,
SHCMD("slock") },
      { 0,                             0x1008ff02, spawn,
SHCMD("xbacklight -inc 10") },
      { 0,                             0x1008ff03, spawn,
SHCMD("xbacklight -dec 10") },
      { ControlMask|Mod1Mask,          XK_z,      spawn,
SHCMD("st -c MOC -e mocp") },
      { ControlMask|Mod1Mask,          XK_a,      spawn,
SHCMD("sakura --class=MC -x mc") },
      { ControlMask|Mod1Mask,          XK_e,      spawn,
SHCMD("sakura -x emacs") },
      { ControlMask|Mod1Mask,          XK_l,      spawn,
SHCMD("bash /home/sogal/bin/launcher") },
      { 0,                             0x1008ff13, spawn,
SHCMD("amixer -q set Master 2dB+ unmute") },
      { 0,                             0x1008ff11, spawn,
SHCMD("amixer -q set Master 2dB- unmute") },
      { 0,                             0x1008ff14, spawn,
SHCMD("mocp --toggle-pause") },
      { 0,                             0x1008ff16, spawn,
SHCMD("mocp --previous") },
      { 0,                             0x1008ff17, spawn,
SHCMD("mocp --next") },
      { 0,                             0xff61,      spawn,
SHCMD("scrot -z") },
      { 0,                             0x1008ff41, spawn,
SHCMD("st -e htop -u sogal") },
      { 0,                             0x1008ff59, spawn,
SHCMD("arandr") },
      { 0,                             0x1008ff8f, spawn,
SHCMD("camorama") },
      { ControlMask|Mod1Mask,          0xffff,    spawn,
SHCMD("sh /home/sogal/bin/arret") },

};

/* button definitions */

```

```

/* click can be ClkLtSymbol, ClkStatusText, ClkWinTitle, ClkClientWin,
or ClkRootWin */
static Button buttons[] = {
    /* click          event mask      button      function
argument */
    { ClkLtSymbol,      0,           Button1,     setlayout,
{0} },
    { ClkLtSymbol,      0,           Button3,     setlayout,
{.v = &layouts[2]} },
    { ClkWinTitle,      0,           Button2,     zoom,
{0} },
    { ClkStatusText,    0,           Button2,     spawn,
{.v = termcmd } },
    { ClkClientWin,     MODKEY,      Button1,     movemouse,
{0} },
    { ClkClientWin,     MODKEY,      Button2,
togglefloating, {0} },
    { ClkClientWin,     MODKEY,      Button3,
resizemouse,    {0} },
    { ClkTagBar,        0,           Button1,     view,
{0} },
    { ClkTagBar,        0,           Button3,     toggleview,
{0} },
    { ClkTagBar,        MODKEY,      Button1,     tag,
{0} },
    { ClkTagBar,        MODKEY,      Button3,     toggletag,
{0} },
    { ClkStatusText,    0,           Button4,     spawn,
SHCMD("amixer -q set Master 2dB+ unmute") },
    { ClkStatusText,    0,           Button5,     spawn,
SHCMD("amixer -q set Master 2dB- unmute") },
};

```

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

http://debian-facile.org/utilisateurs:sogal:config:dwm_config.h



Last update: **03/11/2015 21:03**