Configure the tiling window manager yabai for macos

Ronald (Ryy) Glenn Thomas

2024-12-31

Table of contents

1 Introduction

Yabai is a tiling window manager for macOS.

What is a tiling window manager? According to Wikipedia, “In computing, a tiling window manager is a window manager with an organization of the screen into mutually non-overlapping frames, as opposed to the more common approach (used by stacking window managers) of coordinate-based stacking of overlapping objects (windows) that tries to fully emulate the desktop metaphor”

Specifically, yabai is a tiling window manager for macOS. The user interface allows close control of windows, spaces, and displays. It automatically modifies the window layout throughout yur workspace using a binary space partitioning algorithm.

The github page for yabai is here.

Figure 1: quarto

2 Installation

Yabai can be install using homebrew. The following commands will install yabai and start the service.

```
brew install koekeishiya/formulae/yabai
# start or restart yabai
yabai --start-service
yabai --restart-service
```

Keyboard shortcuts can be defined with skhd or any other suitable software you may prefer. ## .yabairc

```
#!/usr/bin/env bash
sudo yabai --load-sa
yabai -m signal --add event=dock_did_restart action="sudo yabai --load-sa"
set -x

# ====== Variables =============================

declare -A gaps
declare -A color

gaps["top"]="12"
gaps["bottom"]="12"
gaps["left"]="12"
gaps["right"]="12"
gaps["inner"]="12"

color["focused"]="0xE0808080"
color["normal"]="0x00010101"
color["preselect"]="0xE02d74da"


# ===== Loading Scripting Additions ============

# See: https://github.com/koekeishiya/yabai/wiki/Installing-yabai-(latest-release)#macos-big-su
# sudo yabai --load-sa
# yabai -m signal --add event=dock_did_restart action="sudo yabai --load-sa"
# ===== Tiling setting =========================

yabai -m config layout                      bsp

yabai -m config top_padding                 "${gaps["top"]}"
yabai -m config bottom_padding              "${gaps["bottom"]}"
yabai -m config left_padding                "${gaps["left"]}"
yabai -m config right_padding               "${gaps["right"]}"
yabai -m config window_gap                  "${gaps["inner"]}"

yabai -m config mouse_follows_focus         on
yabai -m config focus_follows_mouse         on

yabai -m config window_topmost              off
yabai -m config window_opacity              off
yabai -m config window_shadow               float

yabai -m config window_border               on
yabai -m config window_border_width         2
yabai -m config active_window_border_color  "${color["focused"]}"
yabai -m config normal_window_border_color  "${color["normal"]}"
yabai -m config insert_feedback_color       "${color["preselect"]}"

yabai -m config active_window_opacity       1.0
yabai -m config normal_window_opacity       0.90
yabai -m config split_ratio                 0.50

yabai -m config auto_balance                on

yabai -m config mouse_modifier              fn
yabai -m config mouse_action1               move
yabai -m config mouse_action2               resize



set +x
printf "yabai: configuration loaded...\\n"
```

3 skhd

```
# experimental
# # move window to display left and right
shift + alt - s : yabai -m window --display west; yabai -m display --focus west;
shift + alt - g : yabai -m window --display east; yabai -m display --focus east;
# opens iTerm2
alt - return : "${HOME}"/bin/open_iterm.sh


# Navigation
alt - h : yabai -m window --focus west
alt - j : yabai -m window --focus south
alt - k : yabai -m window --focus north
alt - l : yabai -m window --focus east

# Moving windows
shift + alt - h : yabai -m window --warp west
shift + alt - j : yabai -m window --warp south
shift + alt - k : yabai -m window --warp north
shift + alt - l : yabai -m window --warp east

# Move focus container to workspace
shift + alt - m : yabai -m window --space last; yabai -m space --focus last
shift + alt - p : yabai -m window --space prev; yabai -m space --focus prev
shift + alt - n : yabai -m window --space next; yabai -m space --focus next
shift + alt - 1 : yabai -m window --space 1; yabai -m space --focus 1
shift + alt - 2 : yabai -m window --space 2; yabai -m space --focus 2
shift + alt - 3 : yabai -m window --space 3; yabai -m space --focus 3
shift + alt - 4 : yabai -m window --space 4; yabai -m space --focus 4

# Resize windows
lctrl + alt - h : yabai -m window --resize left:-50:0; \
                  yabai -m window --resize right:-50:0
lctrl + alt - j : yabai -m window --resize bottom:0:50; \
                  yabai -m window --resize top:0:50
lctrl + alt - k : yabai -m window --resize top:0:-50; \
                  yabai -m window --resize bottom:0:-50
lctrl + alt - l : yabai -m window --resize right:50:0; \
                  yabai -m window --resize left:50:0

# Equalize size of windows
lctrl + alt - e : yabai -m space --balance

# Enable / Disable gaps in current workspace
lctrl + alt - g : yabai -m space --toggle padding; yabai -m space --toggle gap

# Rotate windows clockwise and anticlockwise
alt - r         : yabai -m space --rotate 270
shift + alt - r : yabai -m space --rotate 90

# Rotate on X and Y Axis
shift + alt - x : yabai -m space --mirror x-axis
shift + alt - y : yabai -m space --mirror y-axis


# Float / Unfloat window
shift + alt - space : \
    yabai -m window --toggle float; \
    yabai -m window --toggle border

# Restart Yabai
shift + lctrl + alt - r : \
    /usr/bin/env osascript <<< \
        "display notification \"Restarting Yabai\" with title \"Yabai\""; \
    launchctl kickstart -k "gui/${UID}/homebrew.mxcl.yabai"

# Make window native fullscreen
alt - f         : yabai -m window --toggle zoom-fullscreen
shift + alt - f : yabai -m window --toggle native-fullscreen
```

References:

GitHub - koekeishiya/yabai: A tiling window manager for macOS based on binary space partitioning

Tiling window manager - Wikipedia

MacOS - Yabai Configuration | Bryce Smith

Configuring Yabai for Focus

youtube Yabai configuration updates 2024