eepp / **jome** Public

An emoji picker desktop application

MIT license

27 stars 3 forks

Star Notifications

Code Issues 3 Pull requests Actions Security Insights

master Go to file

eepp Add missing 😈 and 👿 emojis ... 25 days ago 109

View code

jome 😏

Table of Contents

jome (*joh·mee*) is a ⌨️ centric emoji picker 🖥️ application.

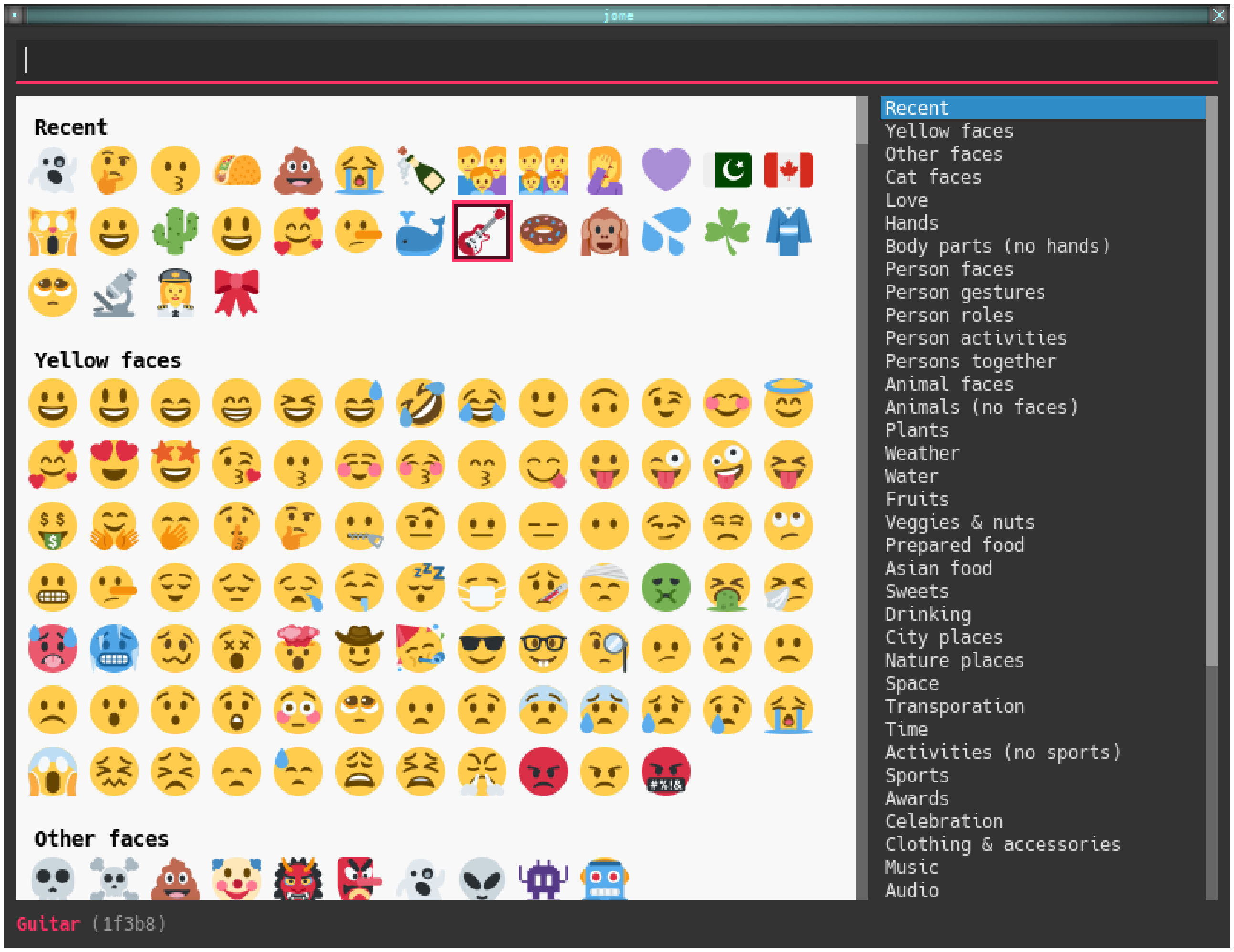
You can also pick an emoji with the 🖱️, don't worry.

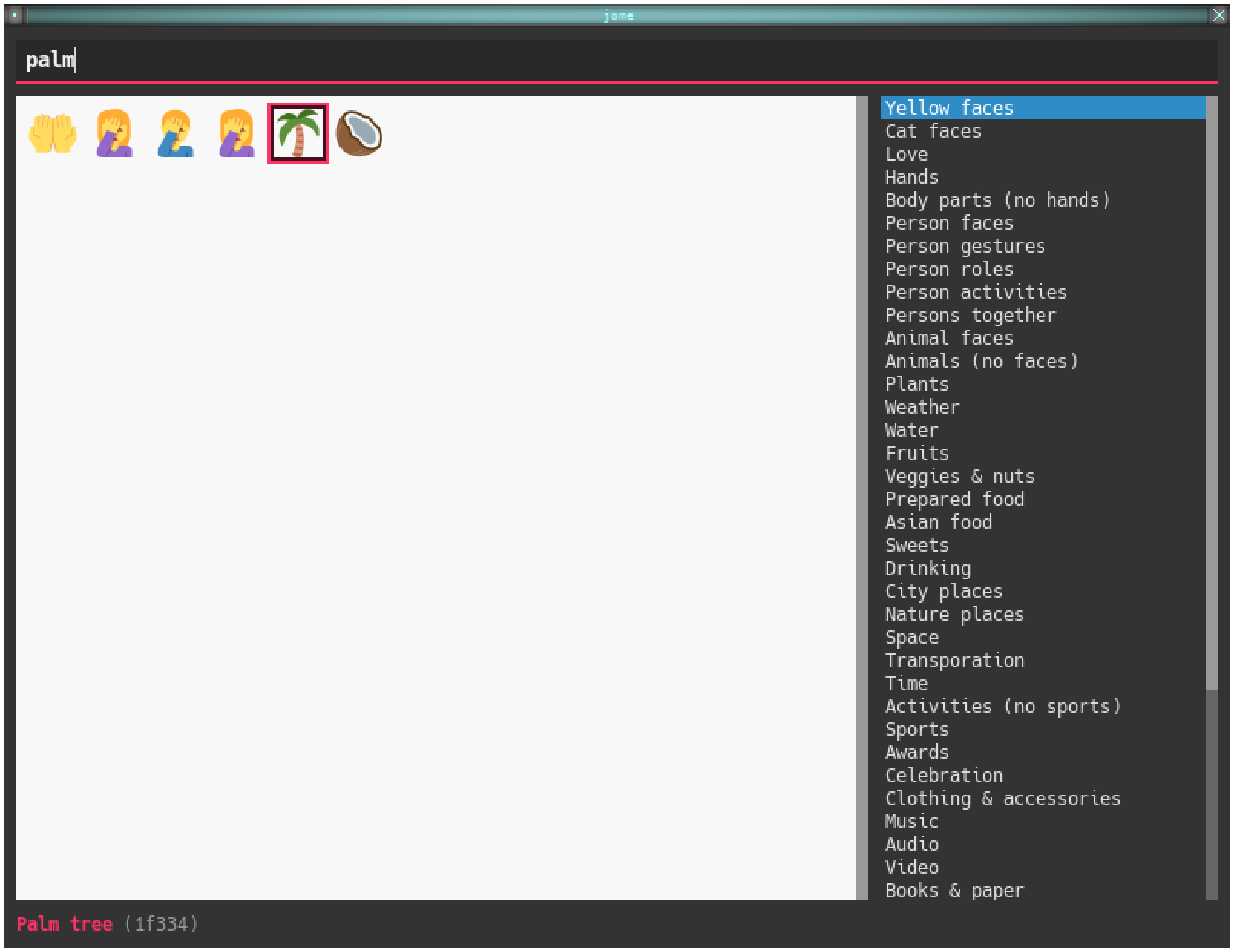
jome has most of the interesting emojis of Emoji 13.1.

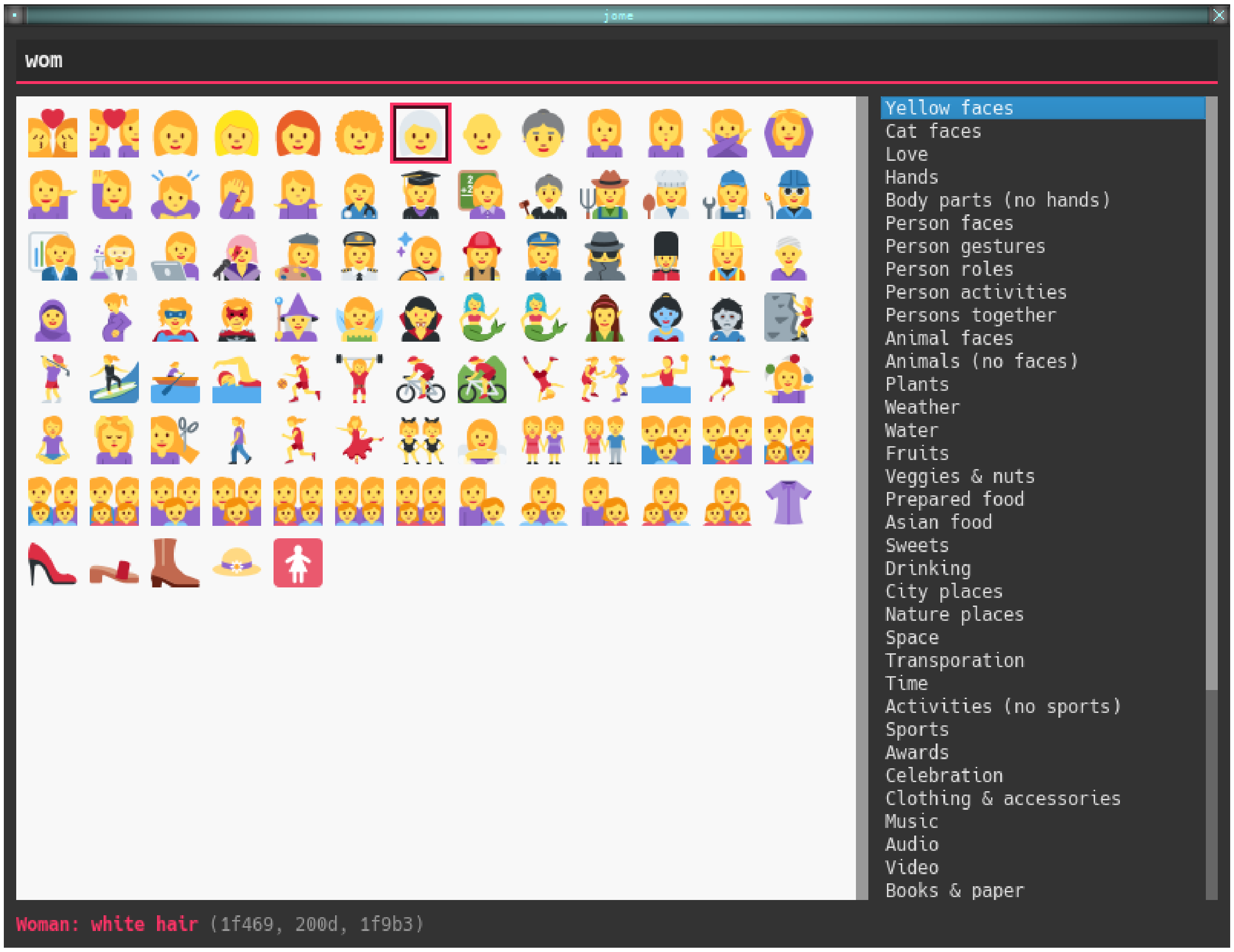
I'm not a fan of the usual very broad categories of emojis which do not intersect so I made my own categories. A given emoji can be found in more than 1️⃣ category. For example, 🐬 is found in both the *animals (no faces)* and *water* categories. I find that it's easier to 🔍 by theme than by very general category. Feel 🆓 to suggest more categories.

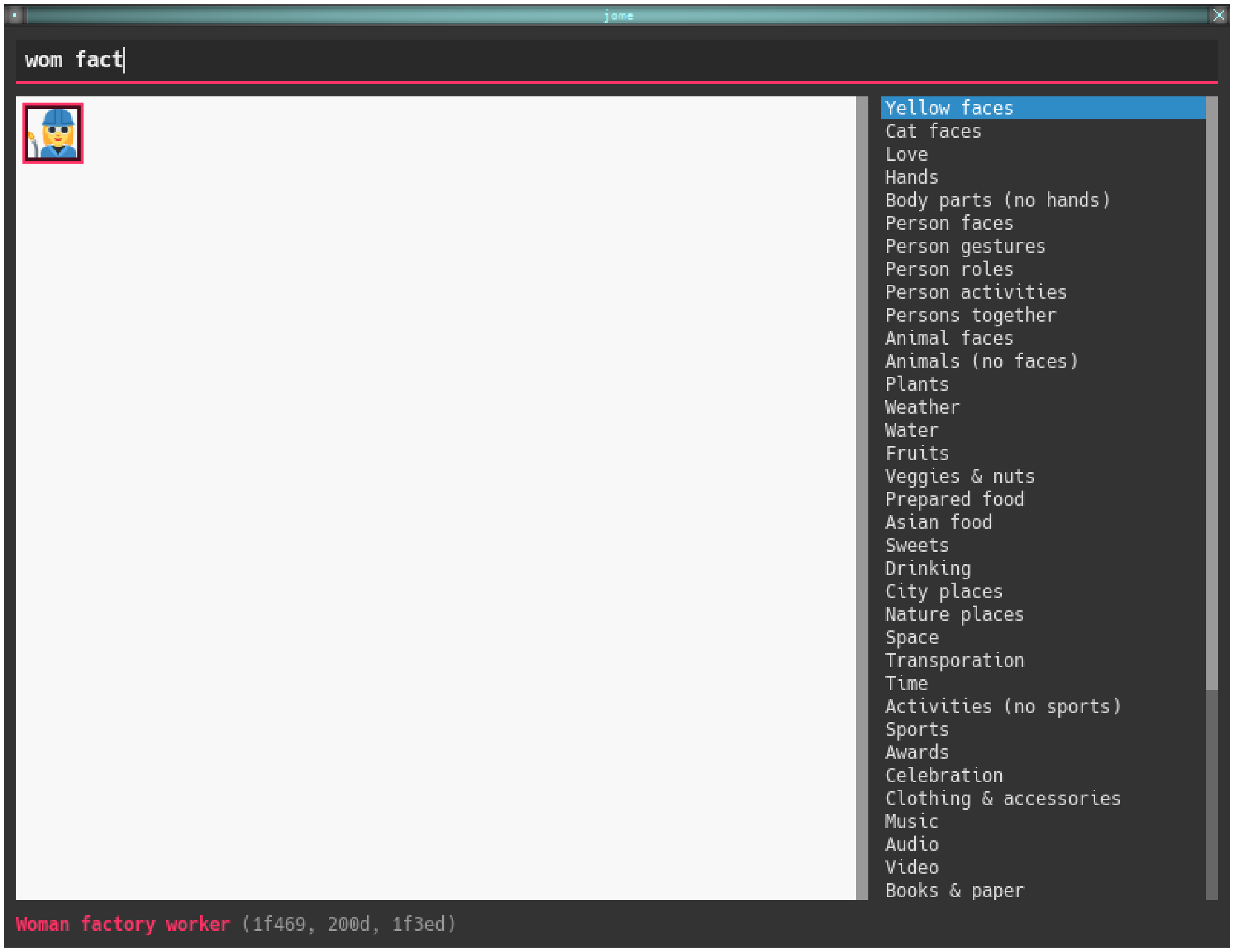
jome is currently only tested on 🐧.

Preview

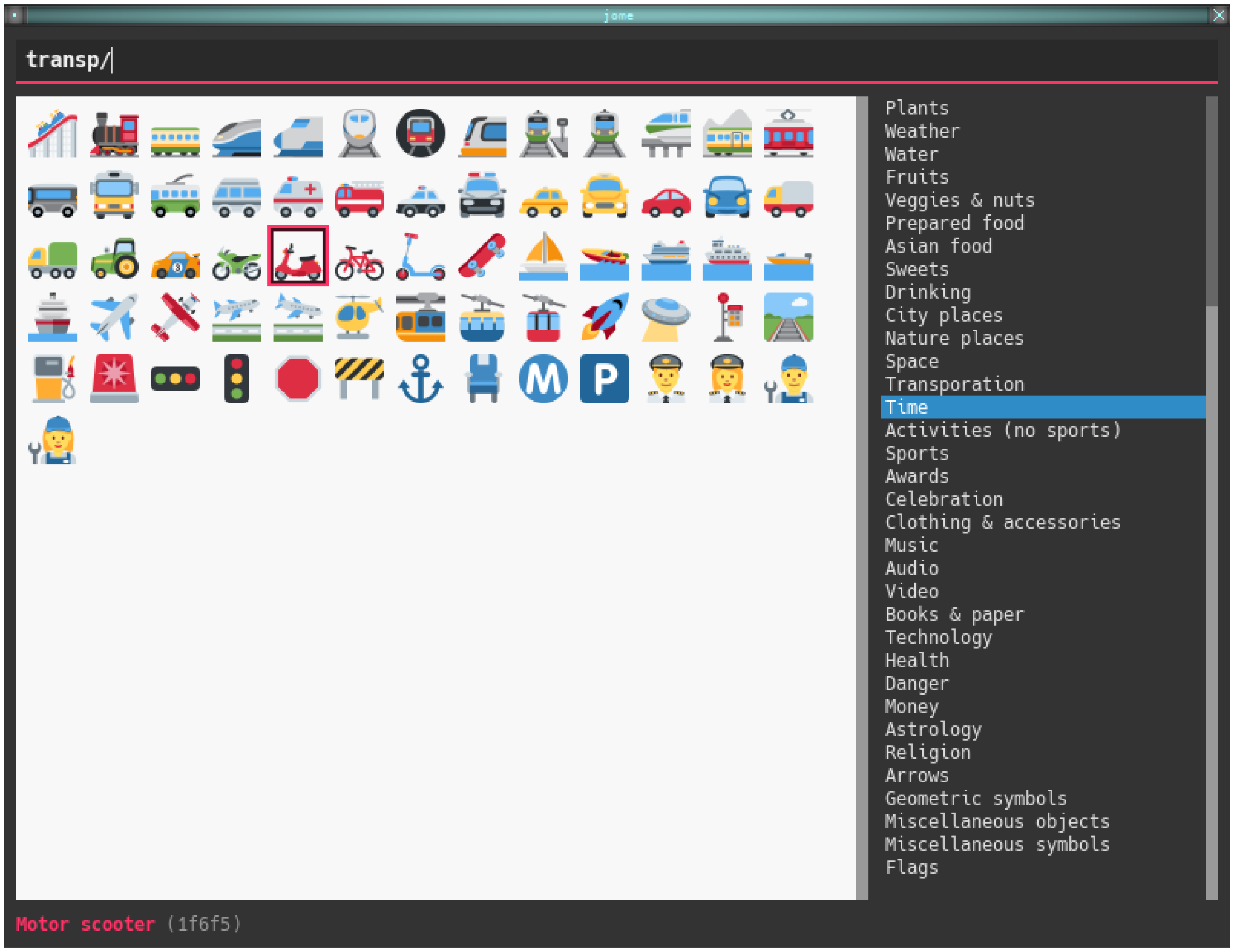


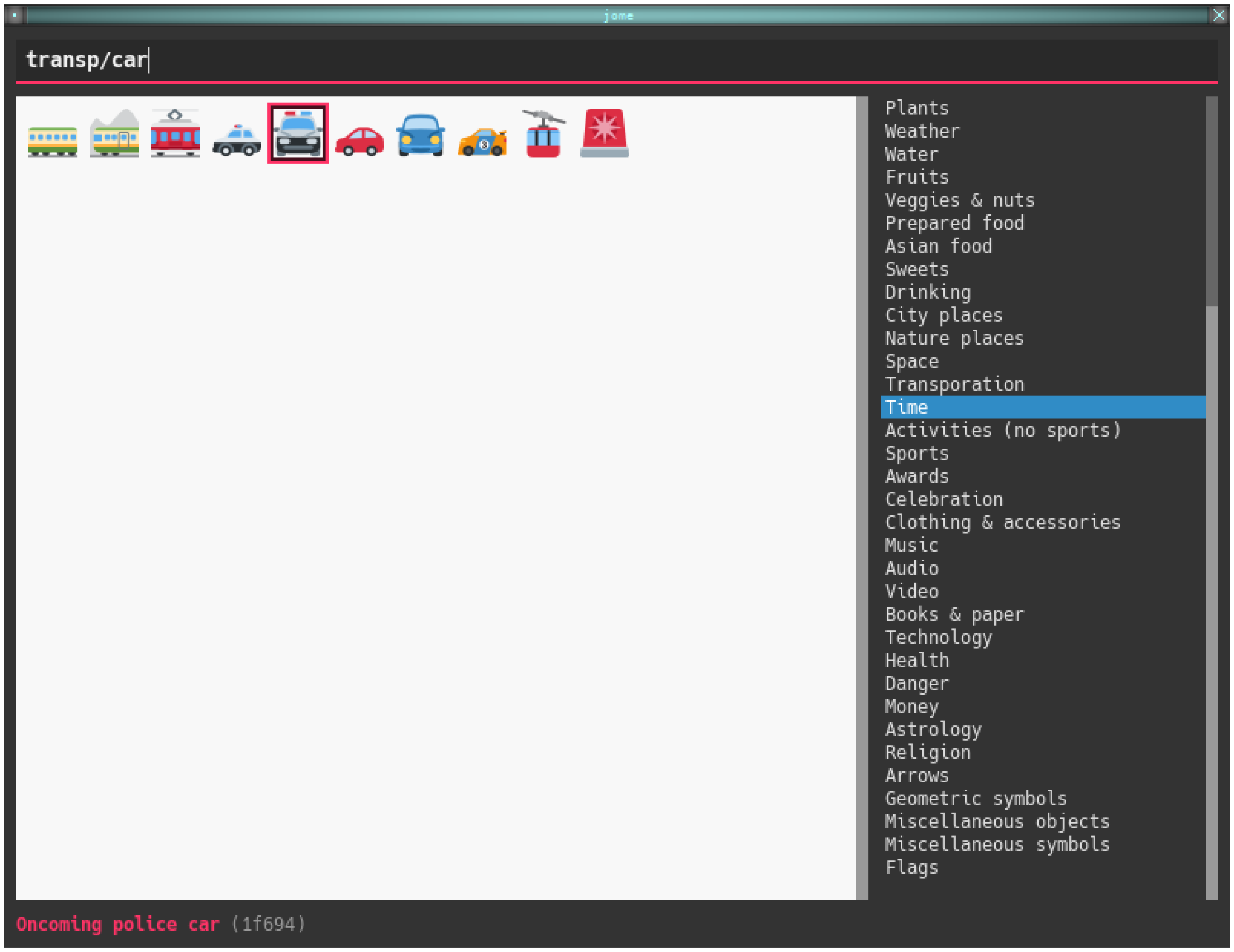

jome
palm
Yellow faces
Cat faces
Love
Hands
Body parts (no hands)
Person faces
Person gestures
Person roles
Person activities
Persons together
Animal faces
Animals (no faces)
Plants
Weather
Water
Fruits
Veggies & nuts
Prepared food
Asian food
Sweets
Drinking
City places
Nature places
Space
Transporation
Time
Activities (no sports)
Sports
Awards
Celebration
Clothing & accessories
Music
Audio
Video
Books & paper
Palm tree (1f334)

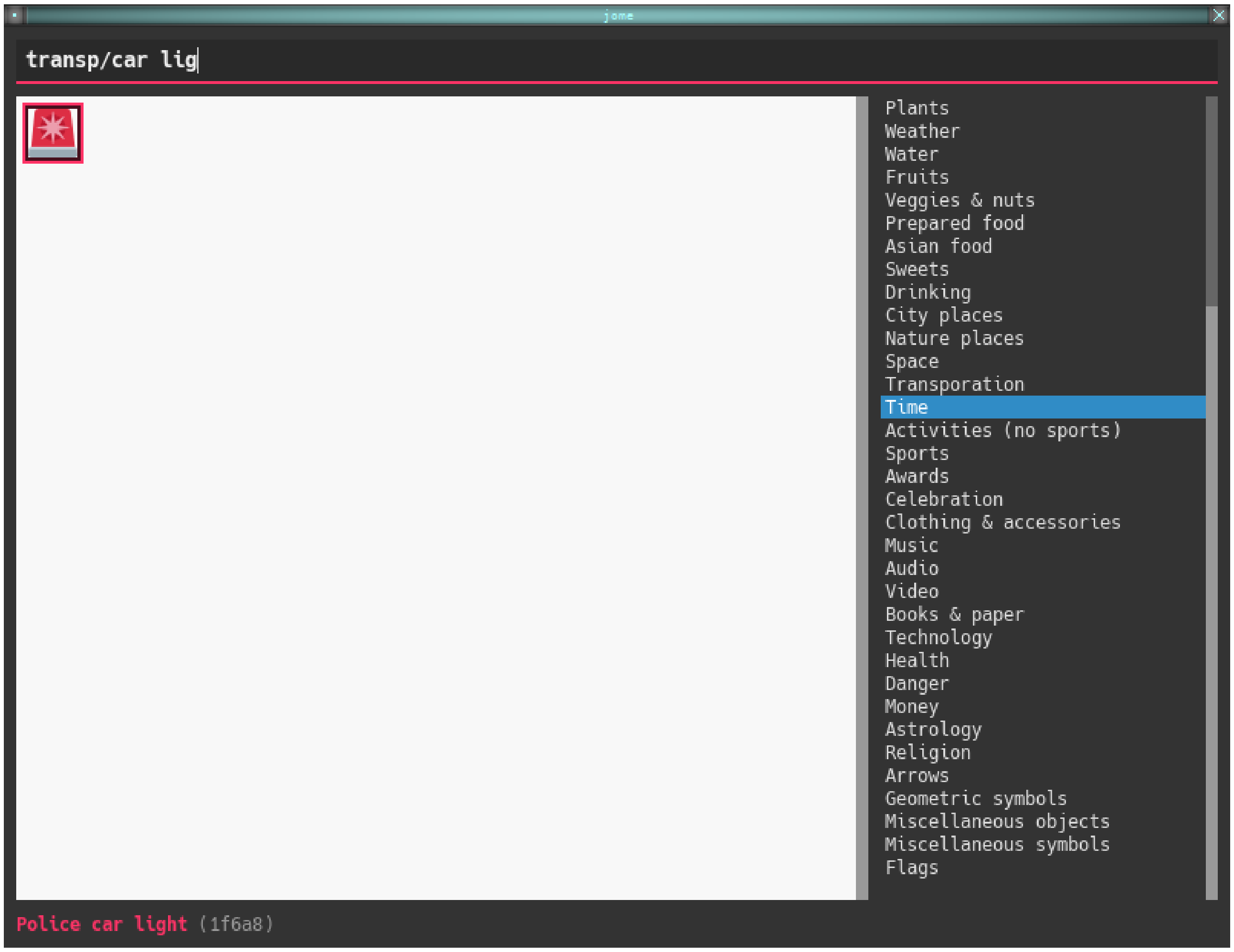

jome
wom
Yellow faces
Cat faces
Love
Hands
Body parts (no hands)
Person faces
Person gestures
Person roles
Person activities
Persons together
Animal faces
Animals (no faces)
Plants
Weather
Water
Fruits
Veggies & nuts
Prepared food
Asian food
Sweets
Drinking
City places
Nature places
Space
Transporation
Time
Activities (no sports)
Sports
Awards
Celebration
Clothing & accessories
Music
Audio
Video
Books & paper
Woman: white hair (1f469, 200d, 1f9b3)



Woman factory worker (1f469, 200d, 1f3ed)


jome
transp/
Plants
Weather
Water
Fruits
Veggies & nuts
Prepared food
Asian food
Sweets
Drinking
City places
Nature places
Space
Transporation
Time
Activities (no sports)
Sports
Awards
Celebration
Clothing & accessories
Music
Audio
Video
Books & paper
Technology
Health
Danger
Money
Astrology
Religion
Arrows
Geometric symbols
Miscellaneous objects
Miscellaneous symbols
Flags
Motor scooter (1f6f5)


jome
transp/car
Plants
Weather
Water
Fruits
Veggies & nuts
Prepared food
Asian food
Sweets
Drinking
City places
Nature places
Space
Transporation
Time
Activities (no sports)
Sports
Awards
Celebration
Clothing & accessories
Music
Audio
Video
Books & paper
Technology
Health
Danger
Money
Astrology
Religion
Arrows
Geometric symbols
Miscellaneous objects
Miscellaneous symbols
Flags
Oncoming police car (1f694)



transp/car lig

Plants
Weather
Water
Fruits
Veggies & nuts
Prepared food
Asian food
Sweets
Drinking
City places
Nature places
Space
Transporation
Time
Activities (no sports)
Sports
Awards
Celebration
Clothing & accessories
Music
Audio
Video
Books & paper
Technology
Health
Danger
Money
Astrology
Religion
Arrows
Geometric symbols
Miscellaneous objects
Miscellaneous symbols
Flags

Police car light (1f6a8)

🏗

You need:

- CMake ≥ 3.1.0
- A C++14 compiler
- Boost ≥ 1.58 (only to 🏗)
- Qt 5 (*Core*, *GUI*, *Widgets*, and *Network* modules)

🏗 and install jome

```
$ mkdir build
$ cd build
$ cmake -DCMAKE_BUILD_TYPE=release ..
$ make -j$(nproc)
$ sudo make install
```

Note	You need to *install* jome for it to find the correct data 📄. If you don't want to

install it on your system, use `-DCMAKE_INSTALL_PREFIX=path/to/install/directory` when you run `cmake`.

Usage

jome's purpose is to help you *pick* an emoji.

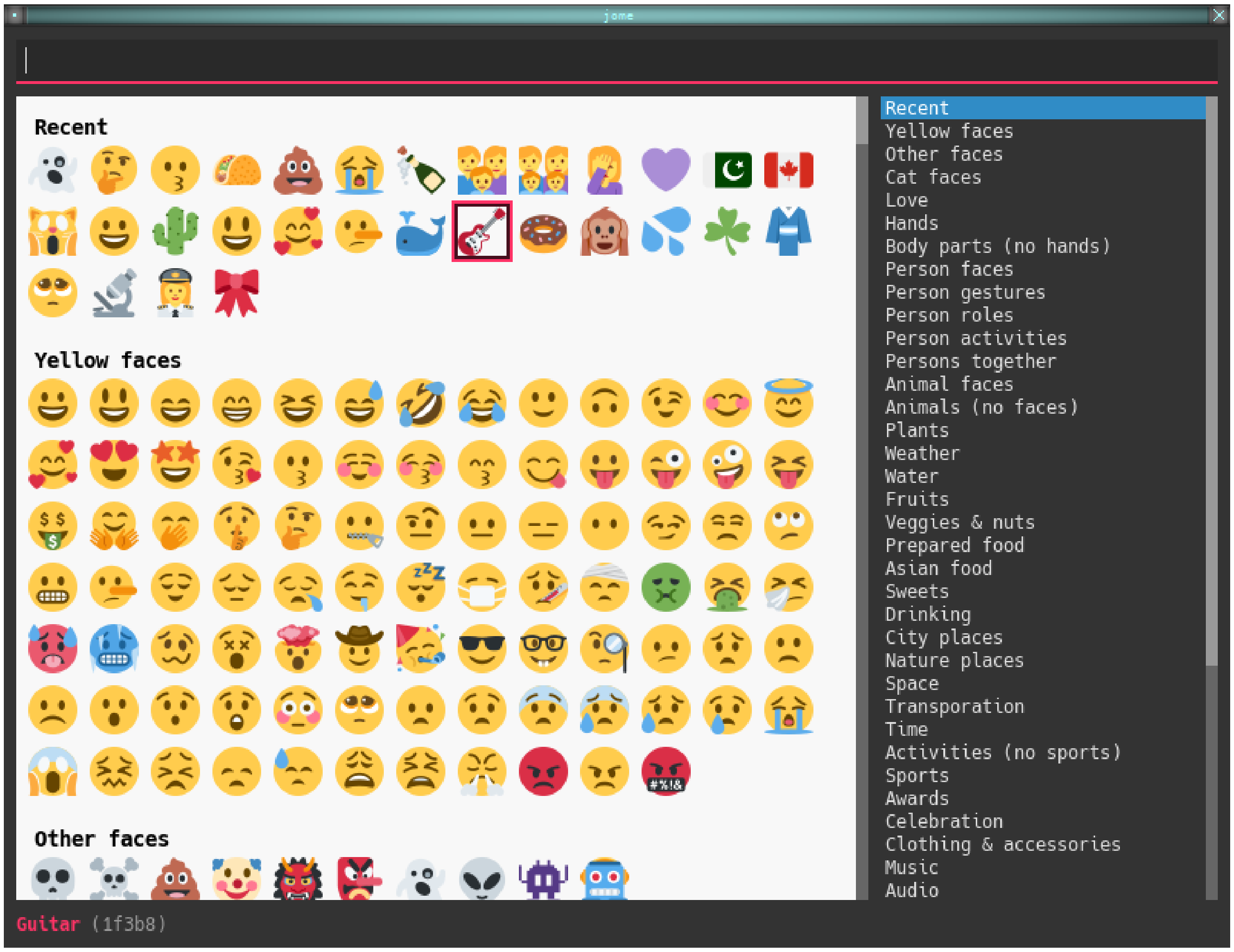
When you accept an emoji (with the ⌨️ or with the 🖱️), jome 🖨️ the UTF-8 emoji or the Unicode codepoints (see the `-f` option), with an optional prefix (see the `-p` option) for each codepoint, to the standard output. Additionally, jome can:

- Execute a custom command which receives the UTF-8 emoji or the Unicode codepoints, with an optional prefix for each codepoint, as its last argument(s). See the `-c` option.
- Send the UTF-8 emoji or the Unicode codepoints, with an optional prefix for each codepoint, in response to a client which requested picking an emoji. See the `-s` option.

If you close the window (you can 👇 **Escape** to do this), then jome 🖨️ nothing to the standard output and executes nothing.

If you don't start jome in server mode (`-s` option) and you don't specify the `-q` option, then jome immediately quits after you accept an emoji or close the window.

README.adoc



There are 4 sections:

🔍 *box* (⬆)

Input box where you can ⌨️ a query to 🔍 emojis.

Emojis

All emojis (with an empty 🔍 box) or 🔍 results.

When there's at least 1 emoji, there's always a selected emoji with a 🔴 box around it.

🖱️ an emoji to accept it.

Hover an emoji to update the ⬇ emoji info text temporarily.

Use the `-d` option to make the background behind emojis dark.

Category list

List of available categories.

When all emojis are 👁️ (the 🔍 box is empty), 🖱️ a category name to scroll to this emoji category.

The first category, *Recent*, is a special category with the recently accepted emojis.

Emoji info text (⬇️)

Name and Unicode codepoints of the selected or hovered emoji.

🔍 emojis

The power of jome is its 🔍 box.

When you launch jome, the 🔍 box is focused, and it should stay focused unless you browse emojis manually with the intention of accepting one with the 🖱️.

The format of a query is 1️⃣ of:

- *`TERMS`*
- *`CAT/`*
- *`CAT/TERMS`*

where:

`CAT`

Partial name of categories in which to 🔍.

`TERMS`

Space-separated list of 🔍 terms.

For an emoji to be part of the results, at least 🔍 of its keywords must contain *all* the 🔍 terms.

Select and accept an emoji

To select an emoji, use the following ⌨️:

⬅️, ➡️, ⬆️, ⬇️

Go ⬅️/➡️/⬆️/⬇️.

Ctrl+⬅️, Ctrl+➡️

Go ⬅️/➡️ 5️⃣ emojis.

Page ⬆️, Page ⬇️

Go ⬆️/⬇️ 10 rows.

Home

Go to the first emoji.

End
Go to the last emoji.

To accept the selected emoji, 👇:

Enter
Accept the selected emoji with the default skin tone (if applicable).

F1, F2, F3, F4, F5
If the selected emoji supports skin tones, accept the selected emoji with a light, medium-light, medium, medium-dark, or dark skin tone.

To cancel, 👇 **Escape** or close the window.

Go to Emojipedia page

To go to the Emojipedia 📃 of the selected emoji, 👇 **F12**.

To go to the Emojipedia 📃 of any emoji with the 🖱️, right-click it and click "Go to Emojipedia page".

Command-line options

`-f FMT`
Set the output format to `FMT`:

`utf-8` (default)
UTF-8 emoji.

`cp`
Space-separated Unicode codepoints (hexadecimal).

Example: `1f645 200d 2642 fe0f`

`-p PREFIX`
Set the prefix to be prepended to each Unicode codepoint.

For example, with `-f cp` and `-p U+`: `U+1f645 U+200d U+2642 U+fe0f`.

`-n`
Do not 🖨️ a newline after 🖨️ the emoji or codepoints.

`-c CMD`
When you accept an emoji, execute command `CMD` 20 ms *after* closing the jome window.

jome interprets *`CMD`* like a 🐚 does, so you can have arguments too.

`CMD` receives the UTF-8 emoji or the Unicode codepoints (depending on the `-f` option) with their optional prefix as its last argument(s).

Examples with xdotool:

```
$ jome -c 'xdotool type'
$ jome -f cp -p U -c 'xdotool key --delay 20'
```

`-q`

Do not quit when you accept an emoji.

By default, when you accept an emoji (with the ⌨️ or with the 🖱️), jome:

1. 🖨️ the accepted emoji or its codepoints to the standard output.
2. Hides its window.
3. **Optional**: Executes a command (see the `-c` option) after 20 ms.
4. **If not running in server mode**, quits (see the `-s` option).

With the `-q` option, jome does not hide its window and does not quit when you accept an emoji so that you can make it 🖨️ multiple emojis and/or execute a command multiple ⏱️ with multiple emojis without restarting the application.

You cannot specify the `-q` and `-s` options at the same ⏱️.

`-s NAME`

Start jome in server mode and set the server name to *`NAME`*.

On Unix, this creates the socket 📄 `/tmp/`*`NAME`* which must *not exist* before you start jome.

You cannot specify the `-s` and `-q` options at the same ⏱️.

`-d`

Use a dark background for emojis.

`-w WIDTH`

Set the width of individual emojis to *`WIDTH`* pixels, amongst 16, 24, 32 (default), 40, or 48.

Server mode

jome features a server mode to avoid creating a process (a Qt window can be quite long to create) every ⏲️ you need to pick an emoji. With this mode, you can 👁️ the jome window instantaneously.

To start jome in server mode, use the `-s` option to specify the server name:

```
$ jome -s mein-server
```

This creates a local server named `mein-server`. On Unix, it creates the socket 📄 `/tmp/mein-server`.

Important	On Unix, the server mode won't work if the socket 📄 already exists. Remove the 📄 before you start jome in server mode:
	
	`$ rm -f /tmp/mein-server`
	`$ jome -s mein-server`

When jome starts in server mode, it does not 👁️ its window. Instead, it ⏳ for a command sent by the client, `jome-ctl`. To 👁️ the window:

```
$ jome-ctl mein-server
```

When you accept an emoji, `jome-ctl` 🖨️ what jome also 🖨️ to the standard output and quits with exit code 0️⃣. Therefore, the output format of `jome-ctl` is 🎛️ by the options passed to `jome`.

If you cancel jome (press **Escape** or close the window), `jome-ctl` 🖨️ nothing and returns with exit code 1️⃣.

In server mode, jome does not quit once you accept an emoji or cancel: it hides the window and keeps 👂. To make it quit gracefully, which also removes the socket 📄:

```
$ jome-ctl mein-server quit
```

You don't need to use what `jome-ctl` 🖨️ to the standard output. You can use jome in server mode with the `-c` option to make jome execute a command itself. For example:

```
$ rm -f mein-server
$ jome -s mein-server -c 'xdotool type'
```

Then, bind a ⌨️ shortcut to:

```
$ jome-ctl mein-server
```

⌨️ the accepted emoji

Here are Bash 📜 to ⌨️ the accepted emoji with xdotool.

Non server mode

With `xdotool key`

```
#!/usr/bin/bash

codepoints="$(jome -f cp -p U)"

if [ $? -ne 0 ]; then
    exit 1
fi

xdotool key --delay 20 $codepoints
```

With `xdotool type`

```
#!/usr/bin/bash

emoji="$(jome)"

if [ $? -ne 0 ]; then
    exit 1
fi

xdotool type "$emoji"
```

Server mode

With `xdotool key`

```
#!/usr/bin/bash

socket_name="jome.socket.$(id -u)"

if ! pidof jome &>/dev/null; then
```

```
    rm -f "/tmp/$socket_name"
    jome -s "$socket_name" -f cp -p U -c 'xdotool key --delay 20' & disown

    while [ ! -e "/tmp/$socket_name" ]; do
        sleep .1
    done
fi

jome-ctl "$socket_name"
```

With `xdotool type`

```
#!/usr/bin/bash

socket_name="jome.socket.$(id -u)"

if ! pidof jome &>/dev/null; then
    rm -f "/tmp/$socket_name"
    jome -s "$socket_name" -c 'xdotool type' & disown

    while [ ! -e "/tmp/$socket_name" ]; do
        sleep .1
    done
fi

jome-ctl "$socket_name"
```

Build

To build and install jome:

1. Make sure you have:

   Build time dependencies

   - A C++-14 compiler
   - CMake ≥ 3.1
   - Qt 5 development tools
   - Boost ≥ 1.58

   Run time dependency
   Qt 5

2. Create a build directory and make it your current working directory:

```
$ mkdir build
$ cd build
```

3. Create the build files:

```
$ cmake .. -DCMAKE_BUILD_TYPE=release
```

   See the `CMAKE_INSTALL_PREFIX` CMake variable to control the installation prefix.

4. Build jome:

```
$ make
```

5. Install jome:

```
$ sudo make install
```

Releases

4 tags

Packages

No packages published

Languages

● C++ 96.0% ● Python 2.8% ● CMake 1.2%