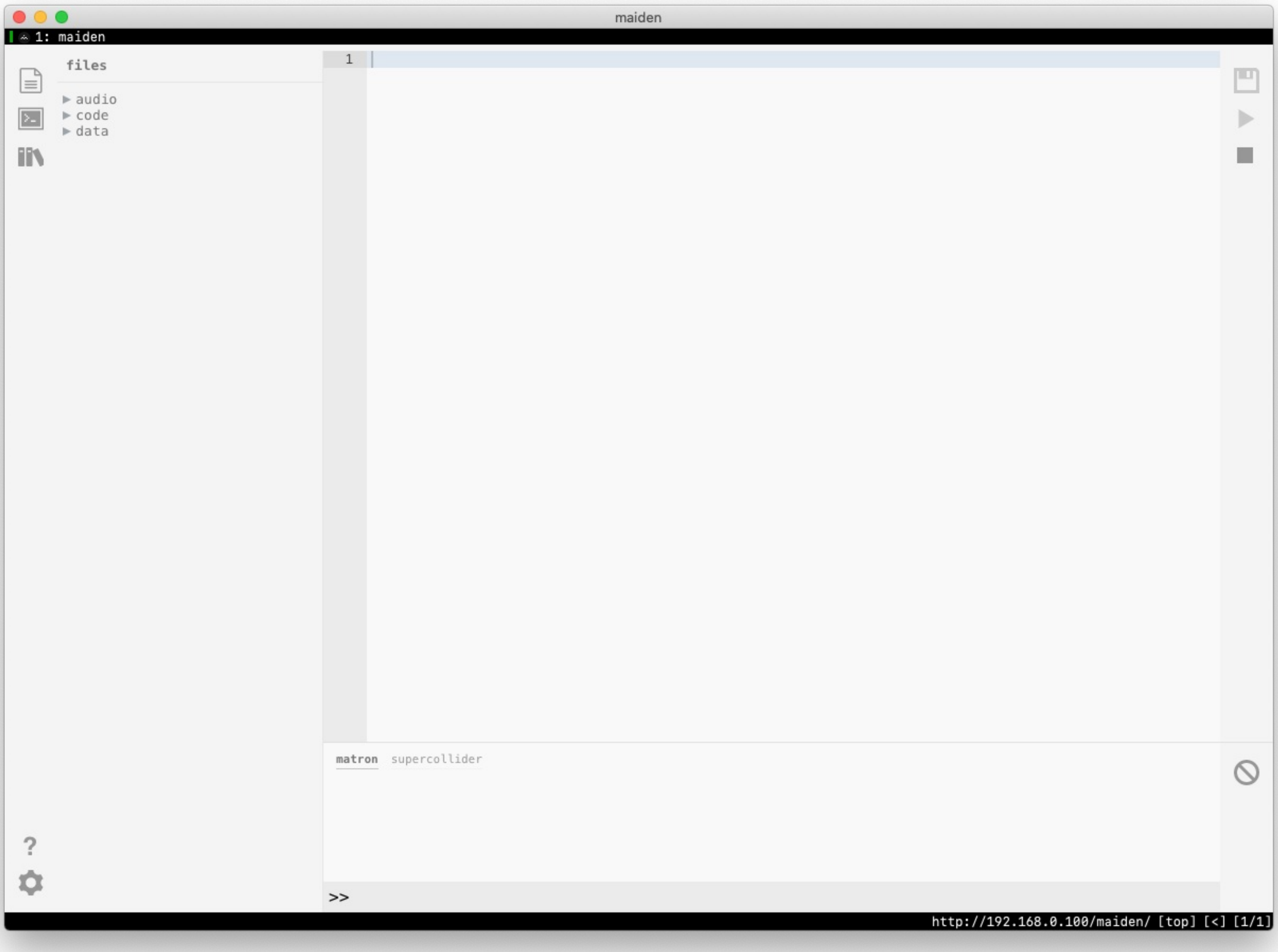
maiden

▼ sections

maiden is a browser-based portal for norns. It can be accessed via a norns-hosted hotspot, or if norns and your browser are on the same WiFi network.

To dive in, point your web browser at `norns.local` to see the maiden interface. If the site is not found, try connecting directly to the IP address shown on the norns screen, for example: `192.168.0.100`.

> Windows: if neither of these URLs resolve, try `IP-ADDRESS-OF-YOUR-NORNS/maiden/`, eg. `192.168.1.100/maiden/`



The interface includes a meta-navigator in the far-left sidebar, which from bottom-to-top allows you to:

- access the *project manager*, where you can manage, discover, and install community scripts on your norns
- toggle the *repl* (read-eval-print-loop), where your scripts + the system both print useful information
- toggle the *file viewer*, where you can view and select scripts to edit
  - note that maiden only has read/write access to files within `/home/we/dust` : `audio` , `code` and `data` – no other system files can be accessed.
  - use SFTP for accessing files outside of `/home/we/dust`

Let's start with the *project manager*, so we can download some new community scripts!

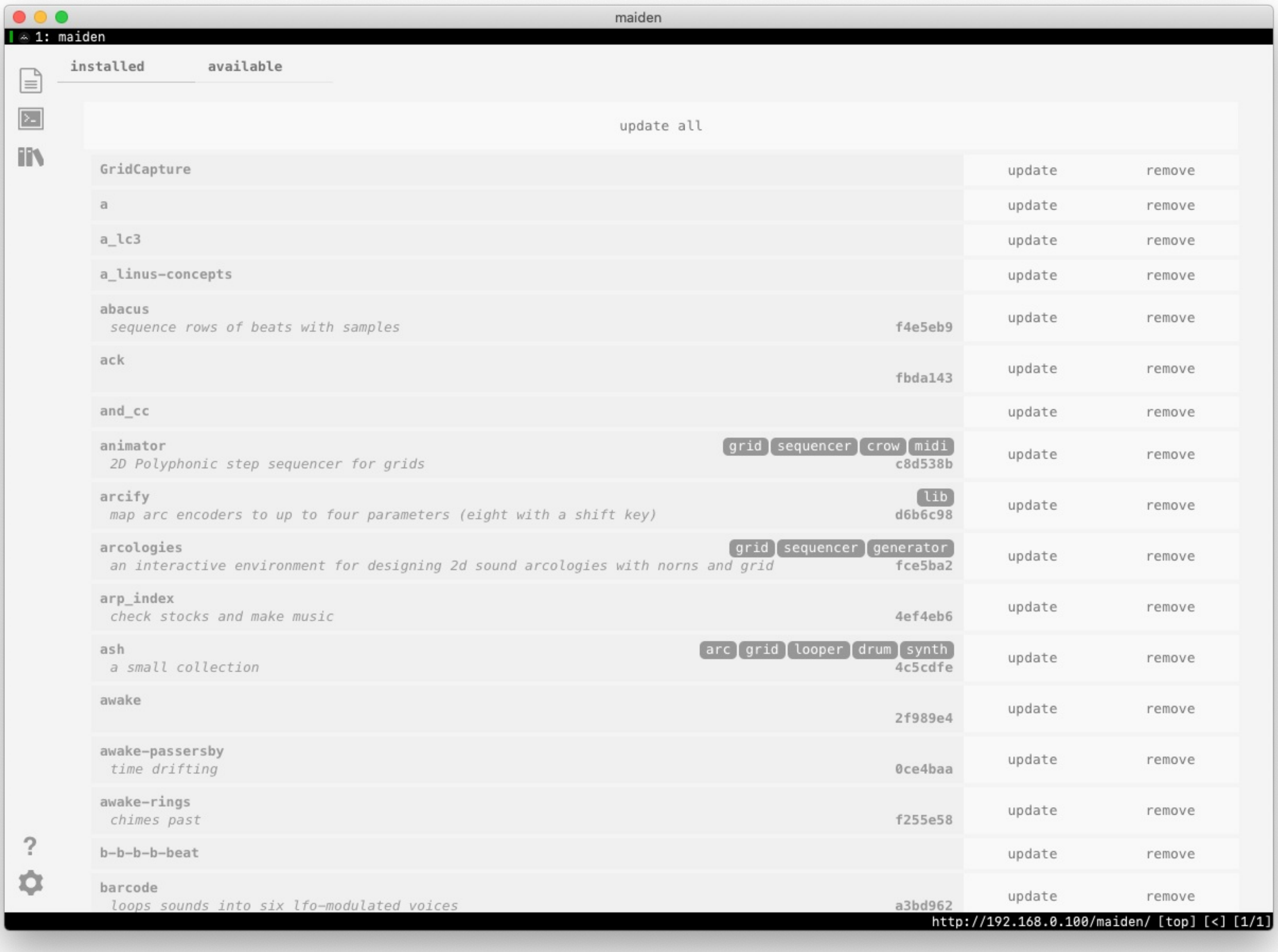
project manager

maiden features a project manager to help you discover and download new projects. Projects contain both engines and scripts.

You can access both the *base* (projects from monome) and *community* (projects from other artists) repositories via the books icon in the left-sidebar.

installed

This tab shows which projects are currently installed on your norns.



GridCapture			update	remove
a			update	remove
a_lc3			update	remove
a_linus-concepts			update	remove
abacus *sequence rows of beats with samples*		f4e5eb9	update	remove
ack		fbda143	update	remove
and_cc			update	remove
animator *2D Polyphonic step sequencer for grids*	grid sequencer crow midi	c8d538b	update	remove
arcify *map arc encoders to up to four parameters (eight with a shift key)*	lib	d6b6c98	update	remove
arcologies *an interactive environment for designing 2d sound arcologies with norns and grid*	grid sequencer generator	fce5ba2	update	remove
arp_index *check stocks and make music*		4ef4eb6	update	remove
ash *a small collection*	arc grid looper drum synth	4c5cdfe	update	remove
awake		2f989e4	update	remove
awake-passersby *time drifting*		0ce4baa	update	remove
awake-rings *chimes past*		f255e58	update	remove
b-b-b-b-beat			update	remove
barcode *loops sounds into six lfo-modulated voices*		a3bd962	update	remove

```
http://192.168.0.100/maiden/ [top] [<] [1/1]
```

Each entry has two actions: **update** and **remove**.

If you choose to update a project which you downloaded through maiden, please note that local modifications you have made will be overwritten. If you wish to retain multiple versions of a project, please reference the SFTP guide.

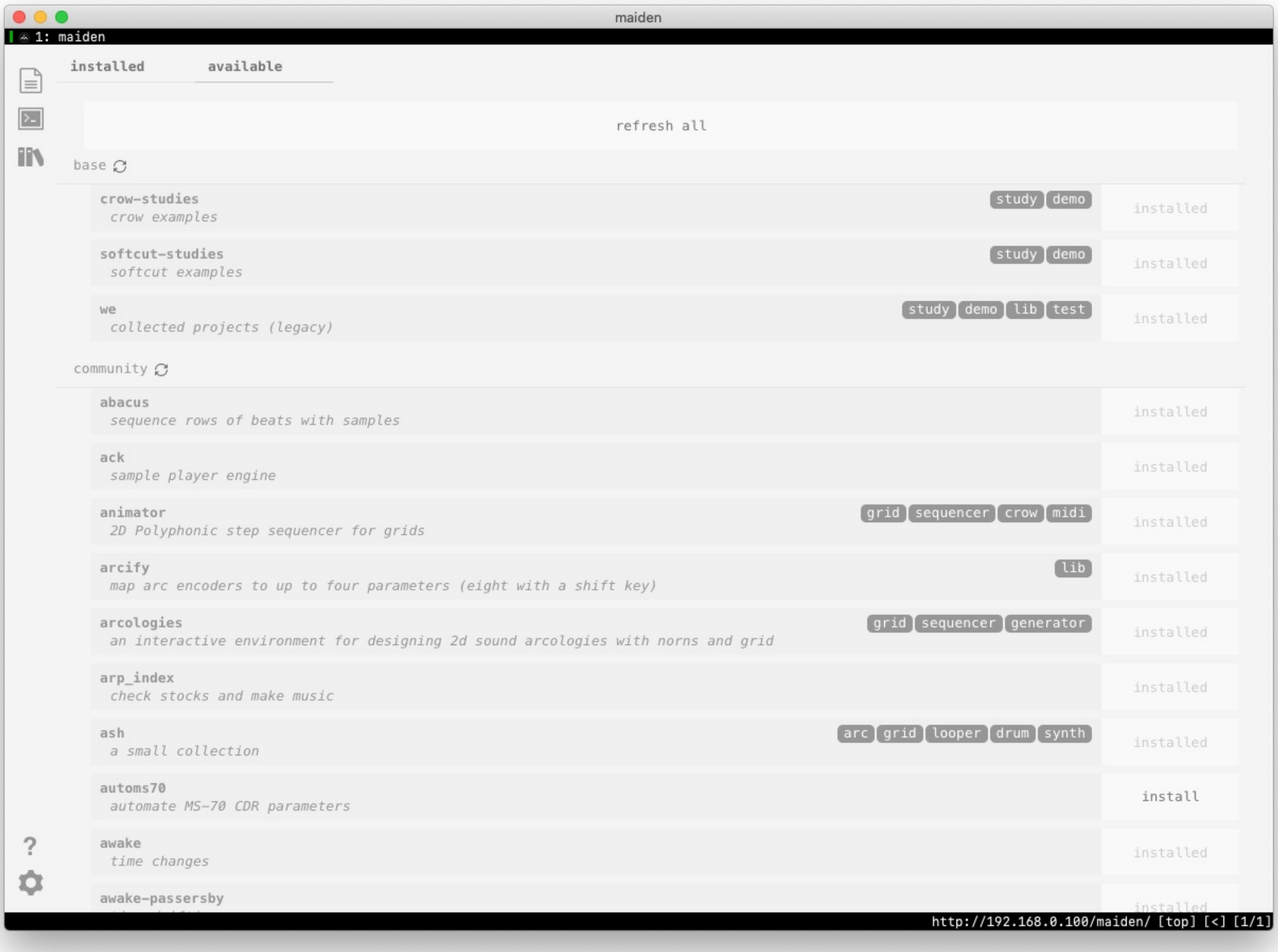
Once you update a project through the PROJECT MANAGER, you'll see a commit number listed on the right of the project's tile (like *34d225b*). Click a project's commit number to be brought to the project's GitHub page, where you can learn more about the project and verify that the version you have is the latest.

> If you are updating a project through the PROJECT MANAGER that was not installed by using the PROJECT MANAGER, you will receive an error that the project cannot be found in the catalog. Please delete the previously installed version and reinstall through PROJECT MANAGER, which establishes the necessary git files for future updates.

available

This tab shows which projects are available through the *base* and *community* repositories.

Whenever maiden is loaded, it automatically refreshes both catalogs. If a script is released after you've loaded maiden, just press the `refresh all` button at the top of the page and all new entries will be added.



maiden

1: maiden

installed available

refresh all

base

crow-studies
crow examples
study demo
installed

softcut-studies
softcut examples
study demo
installed

we
collected projects (legacy)
study demo lib test
installed

community

abacus
sequence rows of beats with samples
installed

ack
sample player engine
installed

animator
2D Polyphonic step sequencer for grids
grid sequencer crow midi
installed

arcify
map arc encoders to up to four parameters (eight with a shift key)
lib
installed

arcologies
an interactive environment for designing 2d sound arcologies with norns and grid
grid sequencer generator
installed

arp_index
check stocks and make music
installed

ash
a small collection
arc grid looper drum synth
installed

automs70
automate MS-70 CDR parameters
install

awake
time changes
installed

awake-passersby

http://192.168.0.100/maiden/ [top] [<] [1/1]

Many projects will have informational tags like **crow**, **drum**, **looper**, as well as a project description. Please note that the **lib** tag is specifically used to indicate that a project includes both a script *and* an engine, which will require a device restart.

Each entry has an **INSTALL** action, which can be used to install the selected script.

You will have to restart norns if a freshly-installed project contains an engine

repl

Messages are printed in the bottom panel. There are two tabs: matron is the main lua environment, and sc is supercollider which is the engine environment.

You can use the bottom prompt to type commands which will be interpreted by the system. For example:

```
print("hello there")
```

will display the expected message in the window above.

The clear icon to the right will clear the current messages.

If you need to restart the matron/crone environment for any reason (ie, the menu system is not accessible), you can issue a command via the REPL:

```
;restart
```

This will disconnect maiden, but once matron has restarted you can reconnect.

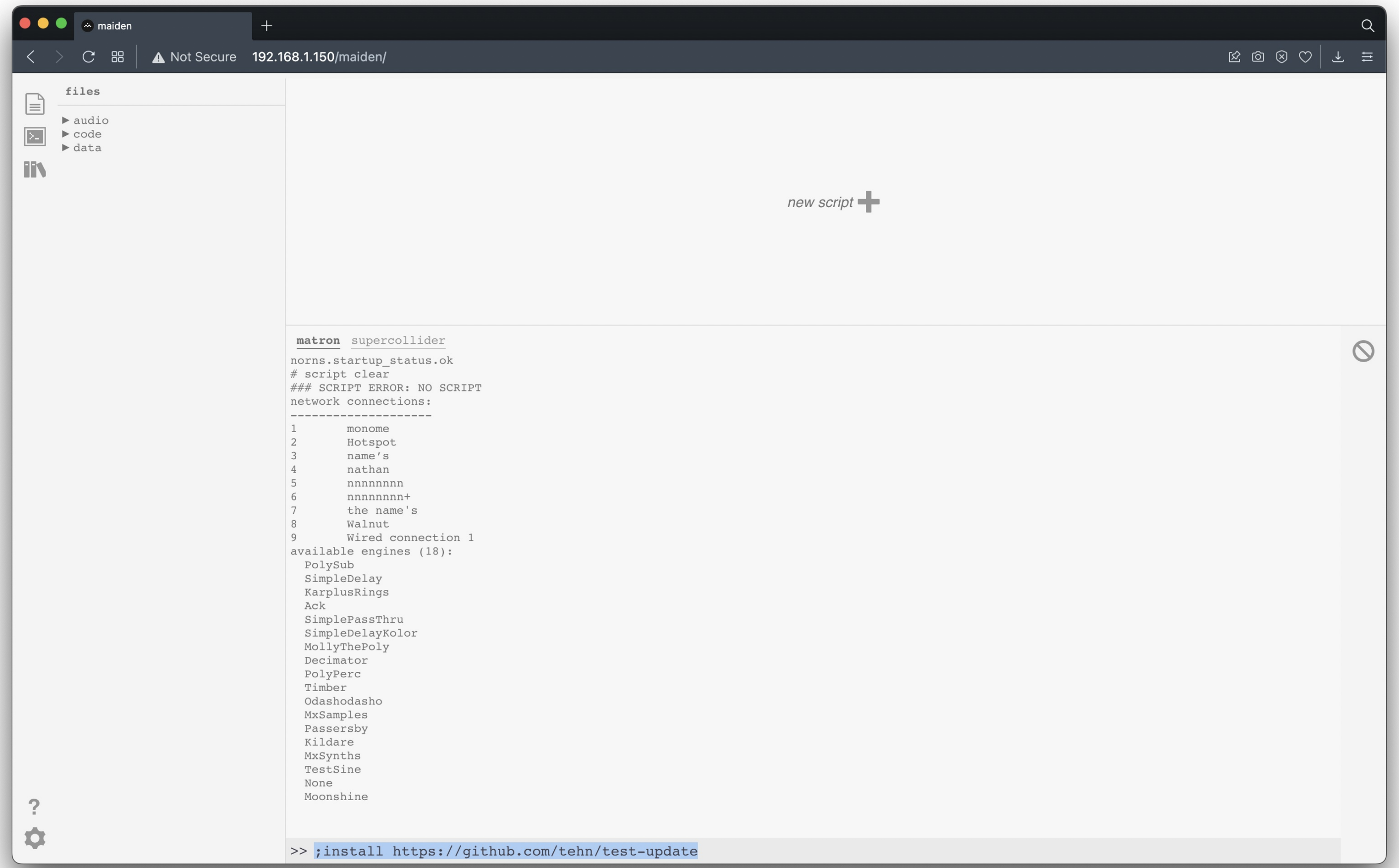
installing scripts from GitHub (WiFi connection required)

Sometimes, scripts don't make it into the community catalog. To fetch a script that's only hosted on a developer's GitHub:

- copy the URL of the lone project (eg. `https://github.com/tehn/test-update`)
- in maiden's REPL, execute:

```
;install https://github.com/tehn/test-update
```

eg.



```
norns.startup_status.ok
# script clear
### SCRIPT ERROR: NO SCRIPT
network connections:
--------------------
1       monome
2       Hotspot
3       name's
4       nathan
5       nnnnnnnn
6       nnnnnnnn+
7       the name's
8       Walnut
9       Wired connection 1
available engines (18):
  PolySub
  SimpleDelay
  KarplusRings
  Ack
  SimplePassThru
  SimpleDelayKolor
  MollyThePoly
  Decimator
  PolyPerc
  Timber
  Odashodasho
  MxSamples
  Passersby
  Kildare
  MxSynths
  TestSine
  None
  Moonshine
>> ;install https://github.com/tehn/test-update
```

If the fetch was successful, you'll see:

```
starting...
installed "test-update"!
```

You will have to restart norns if a newly-installed project contains an engine

If the fetch was unsuccessful, the cause is that a script is already installed with the same name. You'll see:

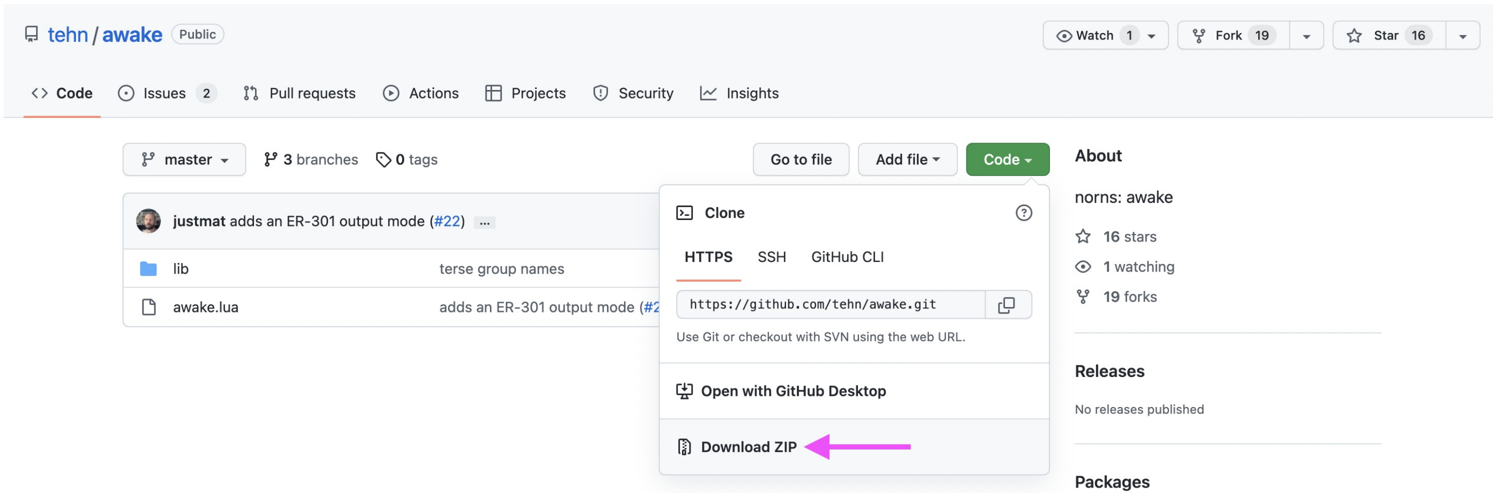
```
starting...
install failed: project test-update already exists in /home/we/dust/code
```

In which case, you just need to remove the redundant script and re-fetch.

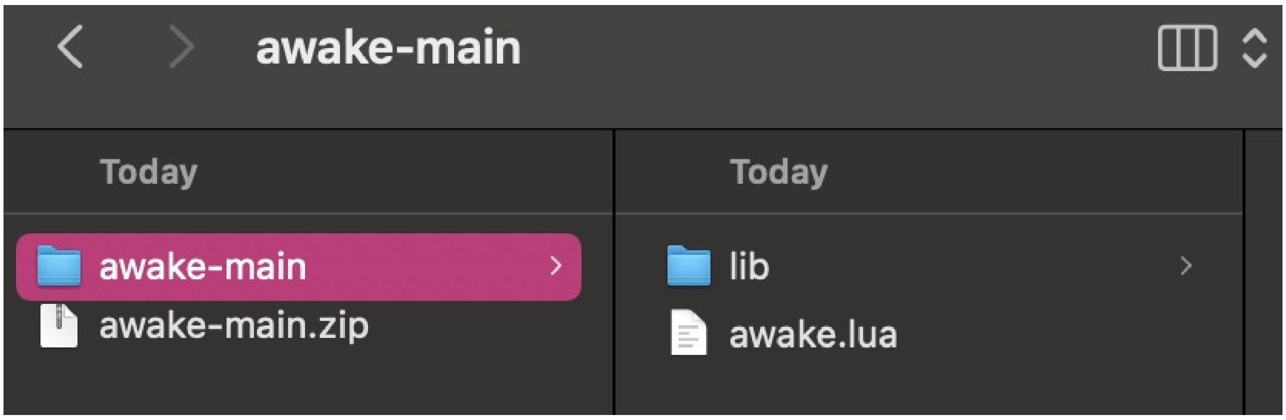
manually installing scripts

If you're unable to connect to maiden + its project manager, you can still install scripts manually via SMB or SFTP by downloading the files onto a non-norns computer and migrating them to the `code` folder on norns.

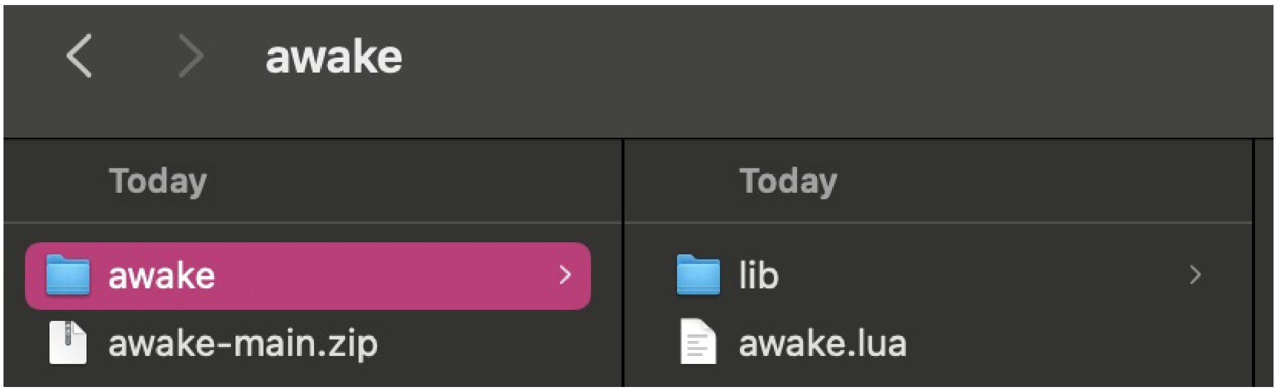
To get started, locate the GitHub repository for the script you wish to install and download the project's zip file to your non-norns computer:



Unzip the file onto your non-norns computer:



Rename the resulting folder to remove any instance of `-main` or `-master` – in other words, we want the script folder to only reflect the script's name:



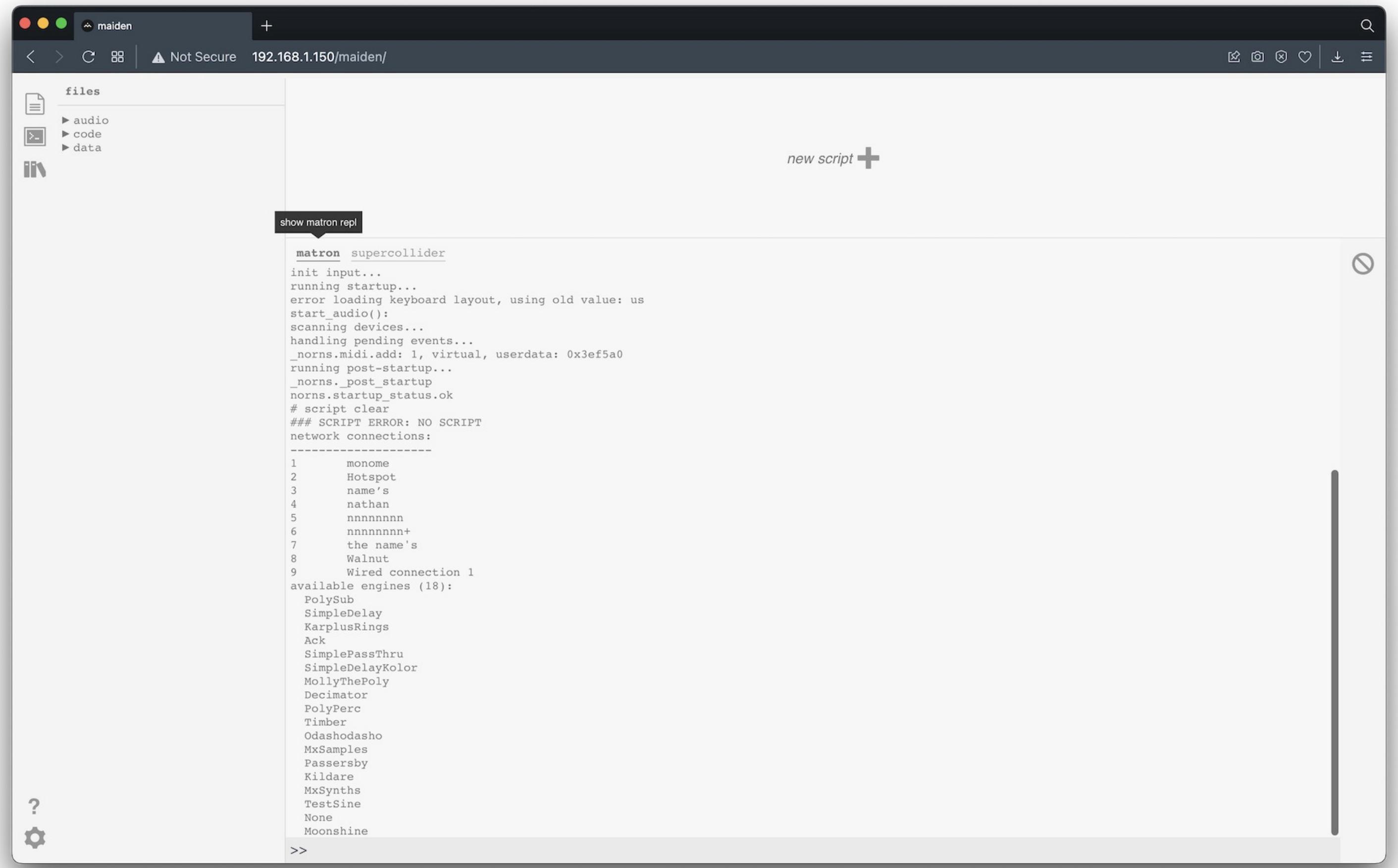
Now, you're ready to migrate the folder to norns! Connect to norns via SMB or SFTP and drag + drop the renamed script folder to the `code` folder of your norns, alongside all the other scripts on your device. Restart your unit to ensure any engine changes are registered and you should be able to start playing!

troubleshooting

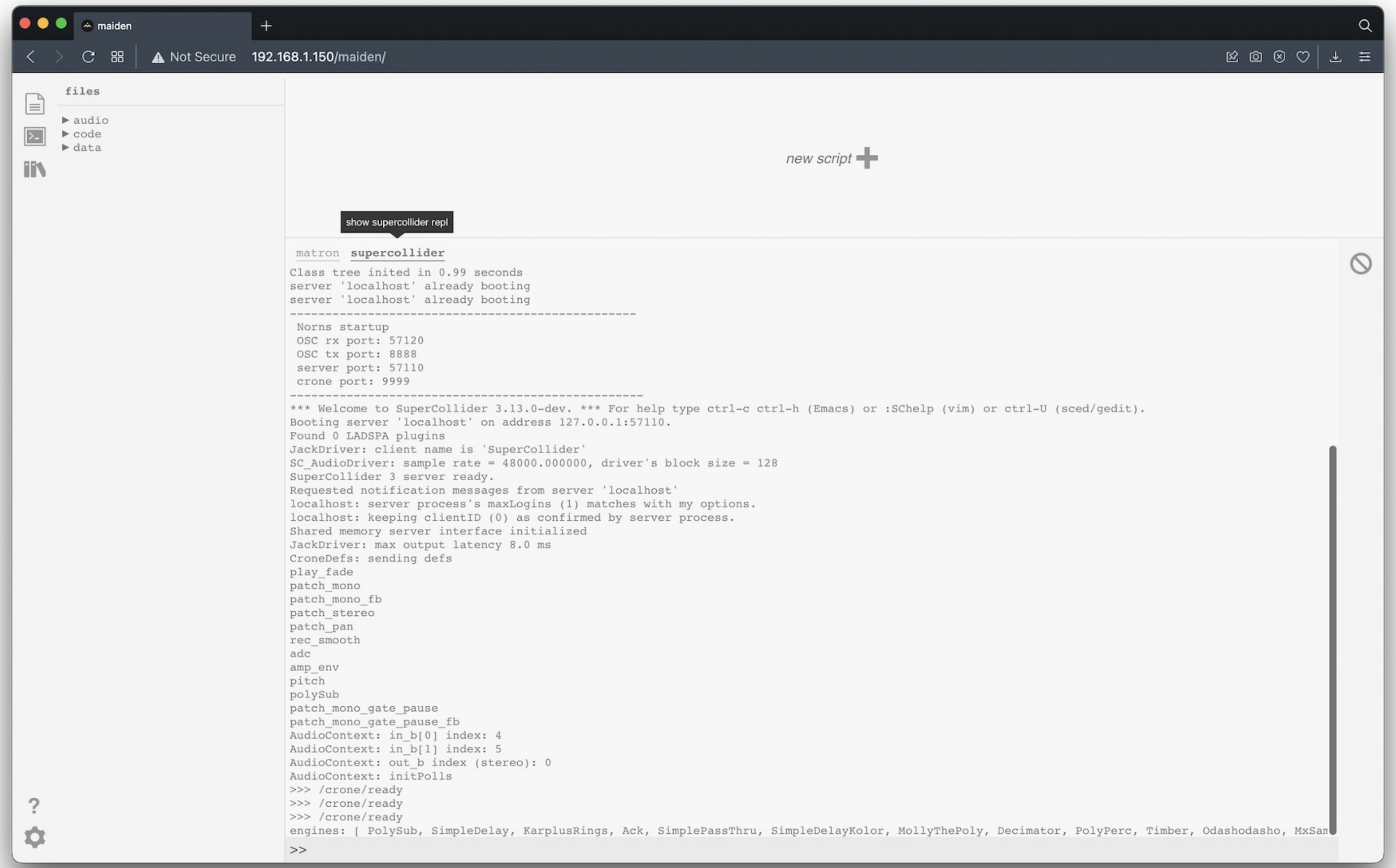
Along your norns journey, you may encounter errors printed to the norns screen like:

- `error: load fail`
- `error: missing <EngineName>`
- `error: SUPERCOLLIDER FAIL`

These errors are straightforward to address when their cause is known – but since the norns screen is only 128 x 64 pixels, robust error logging must find a different avenue. This is where maiden's `matron` (which manages the Lua scripting layer) and `supercollider` (which manages the synth engine layer) tabs come into play:



```
init input...
running startup...
error loading keyboard layout, using old value: us
start_audio():
scanning devices...
handling pending events...
_norns.midi.add: 1, virtual, userdata: 0x3ef5a0
running post-startup...
_norns._post_startup
norns.startup_status.ok
# script clear
### SCRIPT ERROR: NO SCRIPT
network connections:
--------------------
1       monome
2       Hotspot
3       name's
4       nathan
5       nnnnnnnn
6       nnnnnnnn+
7       the name's
8       Walnut
9       Wired connection 1
available engines (18):
  PolySub
  SimpleDelay
  KarplusRings
  Ack
  SimplePassThru
  SimpleDelayKolor
  MollyThePoly
  Decimator
  PolyPerc
  Timber
  Odashodasho
  MxSamples
  Passersby
  Kildare
  MxSynths
  TestSine
  None
  Moonshine
>>
```



```
Class tree inited in 0.99 seconds
server 'localhost' already booting
server 'localhost' already booting
-------------------------------------------------
 Norns startup
 OSC rx port: 57120
 OSC tx port: 8888
 server port: 57110
 crone port: 9999
--------------------------------------------------
*** Welcome to SuperCollider 3.13.0-dev. *** For help type ctrl-c ctrl-h (Emacs) or :SChelp (vim) or ctrl-U (sced/gedit).
Booting server 'localhost' on address 127.0.0.1:57110.
Found 0 LADSPA plugins
JackDriver: client name is 'SuperCollider'
SC_AudioDriver: sample rate = 48000.000000, driver's block size = 128
SuperCollider 3 server ready.
Requested notification messages from server 'localhost'
localhost: server process's maxLogins (1) matches with my options.
localhost: keeping clientID (0) as confirmed by server process.
Shared memory server interface initialized
JackDriver: max output latency 8.0 ms
CroneDefs: sending defs
play_fade
patch_mono
patch_mono_fb
patch_stereo
patch_pan
rec_smooth
adc
amp_env
pitch
polySub
patch_mono_gate_pause
patch_mono_gate_pause_fb
AudioContext: in_b[0] index: 4
AudioContext: in_b[1] index: 5
AudioContext: out_b index (stereo): 0
AudioContext: initPolls
>>> /crone/ready
>>> /crone/ready
>>> /crone/ready
engines: [ PolySub, SimpleDelay, KarplusRings, Ack, SimplePassThru, SimpleDelayKolor, MollyThePoly, Decimator, PolyPerc, Timber, Odashodasho, MxSam
>>
```

If you run into *any* errors using a script (either your own or someone else's), maiden will print messages to each of these REPLs, depending on which layer is experiencing an issue. This extends beyond the script's initialization – if a variable is miscalculated during play and causes instability within a script, for example, maiden will present these errors as well.

collecting logs

If an error occurs during play, it can be easy to assume others will know what's wrong if you describe the general message (eg. "I'm excited to play with this script, but norns says 'error: SUPERCOLLIDER FAIL' when I try to run it"). However, those who want to help will only be able to if you share a more complete report of errors with them.

If we want to see a more encompassing realtime view of error messages from the running script, mods, and SuperCollider, we can log into our our norns directly via SSH and issue the following command:

```
journalctl -f
```

This will not only show the last few system messages (including errors), but it will update as new ones occur.

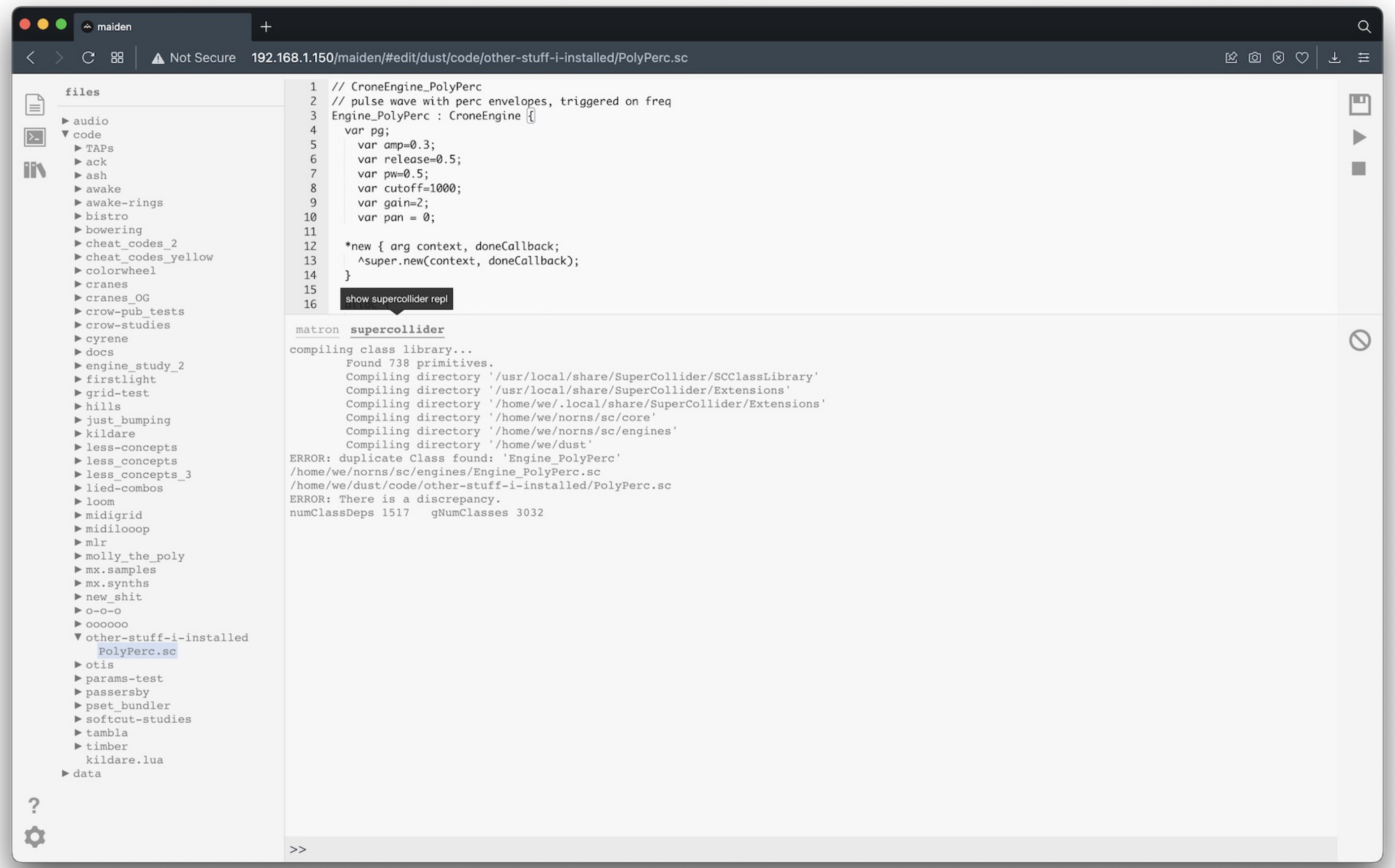
When we're done, we can close the stream by hitting `Ctrl+C`. Be sure to also close the SSH connection to your norns by executing `exit`.

nb. If you are experiencing the issue for the first time, power-cycle your device before attempting to gather logs.

Providing this information will make it easier for others to understand the specific causes of the trouble you're experiencing and empower them to suggest helpful next steps.

example reading

While you collect this information, you might also find that certain situations produce very clear messages – for example, here's the `supercollider` REPL's output during a common occurrence of `error: SUPERCOLLIDER FAIL` / `error: DUPLICATE ENGINES`:



```
// CroneEngine_PolyPerc
// pulse wave with perc envelopes, triggered on freq
Engine_PolyPerc : CroneEngine {
  var pg;
    var amp=0.3;
    var release=0.5;
    var pw=0.5;
    var cutoff=1000;
    var gain=2;
    var pan = 0;

  *new { arg context, doneCallback;
    ^super.new(context, doneCallback);
  }
```

```
compiling class library...
	Found 738 primitives.
	Compiling directory '/usr/local/share/SuperCollider/SCClassLibrary'
	Compiling directory '/usr/local/share/SuperCollider/Extensions'
	Compiling directory '/home/we/.local/share/SuperCollider/Extensions'
	Compiling directory '/home/we/norns/sc/core'
	Compiling directory '/home/we/norns/sc/engines'
	Compiling directory '/home/we/dust'
ERROR: duplicate Class found: 'Engine_PolyPerc'
/home/we/norns/sc/engines/Engine_PolyPerc.sc
/home/we/dust/code/other-stuff-i-installed/PolyPerc.sc
ERROR: There is a discrepancy.
numClassDeps 1517   gNumClasses 3032
```

```
compiling class library...
  Found 738 primitives.
  Compiling directory '/usr/local/share/SuperCollider/SCClassLibrary'
  Compiling directory '/usr/local/share/SuperCollider/Extensions'
  Compiling directory '/home/we/.local/share/SuperCollider/Extensions'
  Compiling directory '/home/we/norns/sc/core'
  Compiling directory '/home/we/norns/sc/engines'
  Compiling directory '/home/we/dust'
ERROR: duplicate Class found: 'Engine_PolyPerc'
/home/we/norns/sc/engines/Engine_PolyPerc.sc
/home/we/dust/code/other-stuff-i-installed/PolyPerc.sc
ERROR: There is a discrepancy.
numClassDeps 1517   gNumClasses 3032
```

While the first 8 lines may not mean much (they get printed every time SuperCollider loads on norns), there *is* a cluster of errors which can be deciphered:

```
ERROR: duplicate Class found: 'Engine_PolyPerc'
/home/we/norns/sc/engines/Engine_PolyPerc.sc
/home/we/dust/code/other-stuff-i-installed/PolyPerc.sc
```

When chunked, we can better notice the information presented by SuperCollider:

- there's a duplicate of `PolyPerc`
- the first one is at `/home/we/norns/sc/engines/` , named `Engine_PolyPerc.sc`
- the other is at `/home/we/dust/code/other-stuff-i-installed/` , named `PolyPerc.sc`

It may help to remind that maiden accesses and manages files and folders within `/home/we/dust/` (`audio` , `code` and `data`), but not any other system folders. So, the `/home/we/dust/code/other-stuff-i-installed/` location must have been created when another script was installed. Also, `PolyPerc` is installed by default on norns at `/home/we/norns/sc/engines/` , which is a location we cannot access via maiden.

To resolve the issue, we'll either want to delete the one installed in the `/home/we/dust/code/other-stuff-i-installed/` folder or delete that whole folder altogether.

file viewer

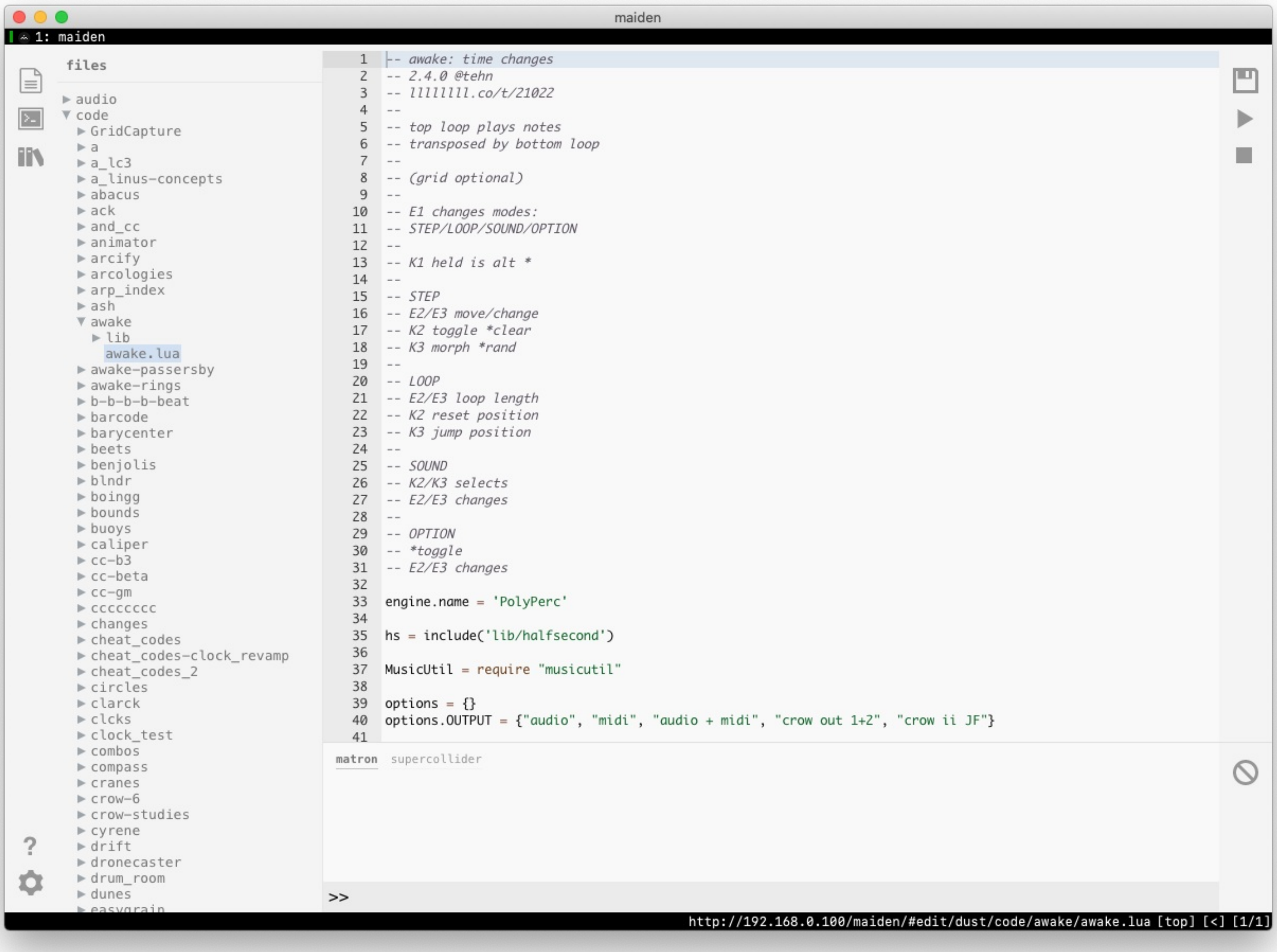
The bulk of the *file viewer* is dedicated to the EDITOR, where you can view and edit the code of a selected script.

This panel lets you select the text you're editing in EDITOR.

There are top bar icons for various actions: **New**, **Delete**, **Duplicate**, **New Folder**, and **Rename**.

nb. If you rename a file, make sure to retain its extension (eg. `.lua`) as you replace the filename. Otherwise, maiden will not know what type of file it is and will not load it as expected.

The `>` 's can be expanded to reveal a file tree. When you select a file, it will show in the EDITOR:



```
files

▶ audio
▼ code
  ▶ GridCapture
  ▶ a
  ▶ a_lc3
  ▶ a_linus-concepts
  ▶ abacus
  ▶ ack
  ▶ and_cc
  ▶ animator
  ▶ arcify
  ▶ arcologies
  ▶ arp_index
  ▶ ash
  ▼ awake
    ▶ lib
      awake.lua
  ▶ awake-passersby
  ▶ awake-rings
  ▶ b-b-b-b-beat
  ▶ barcode
  ▶ barycenter
  ▶ beets
  ▶ benjolis
  ▶ blndr
  ▶ boingg
  ▶ bounds
  ▶ buoys
  ▶ caliper
  ▶ cc-b3
  ▶ cc-beta
  ▶ cc-gm
  ▶ cccccccc
  ▶ changes
  ▶ cheat_codes
  ▶ cheat_codes-clock_revamp
  ▶ cheat_codes_2
  ▶ circles
  ▶ clarck
  ▶ clcks
  ▶ clock_test
  ▶ combos
  ▶ compass
  ▶ cranes
  ▶ crow-6
  ▶ crow-studies
  ▶ cyrene
  ▶ drift
  ▶ dronecaster
  ▶ drum_room
  ▶ dunes
  ▶ easygrain
```

```lua
-- awake: time changes
-- 2.4.0 @tehn
-- llllllll.co/t/21022
--
-- top loop plays notes
-- transposed by bottom loop
--
-- (grid optional)
--
-- E1 changes modes:
-- STEP/LOOP/SOUND/OPTION
--
-- K1 held is alt *
--
-- STEP
-- E2/E3 move/change
-- K2 toggle *clear
-- K3 morph *rand
--
-- LOOP
-- E2/E3 loop length
-- K2 reset position
-- K3 jump position
--
-- SOUND
-- K2/K3 selects
-- E2/E3 changes
--
-- OPTION
-- *toggle
-- E2/E3 changes

engine.name = 'PolyPerc'

hs = include('lib/halfsecond')

MusicUtil = require "musicutil"

options = {}
options.OUTPUT = {"audio", "midi", "audio + midi", "crow out 1+2", "crow ii JF"}
```

```
>>
```

maiden
1: maiden
matron
supercollider
http://192.168.0.100/maiden/#edit/dust/code/awake/awake.lua [top] [<] [1/1]

editor

This is where you can edit the selected script.

To the right there is a bar with three icons: disk is **SAVE**, **PLAY** will run the current script, **STOP** will stop and clear the currently running script.

These actions are bound to the following keyboard shortcuts:

- CMD/CTRL- `S` : save
- CMD/CTRL- `P` : play
- CMD/CTRL- `.` : stop

The editor can be configured for various modes (default, vim, emacs) in addition to tab size and light/dark mode. Click the gear icon at the bottom left of the screen. For more about maiden's keybindings, see the docs in the maiden repository.

realtime line evaluation

maiden's editor also supports realtime line evaluation, which opens up new avenues for live coding from maiden.

To evaluate a line of code, place the cursor on the line and (on your keyboard) execute:

- `command` + `return` : Mac
- `Ctrl` + `Enter` : Windows / Linux

To evaluate many single lines of code at once, highlight each line and execute the eval key combo – maiden will automatically register the line breaks as discrete commands.

Want to give it a try? Open a new file in maiden and paste this in, then start executing:

```
engine.load('PolyPerc') -- load the PolyPerc engine

base = 440 -- controls the base frequency, which can be modified as the sequence runs!

-- execute this entire block:
function notes()
  while true do
    engine.pan(math.random(-1,1))
    clock.sync(1)
    engine.hz(base * math.random(3))
    clock.sync(1/3)
    engine.hz(base / math.random(6))
    clock.sync(2/3)
    engine.hz((base*3) / math.random(6))
  end
end
--- ^ that will randomly create notes if it's part of a clock coroutine, which is next!

seq = clock.run( notes ) -- start a seq clock and assign it the 'notes' function
clock.cancel(seq) -- cancel the last-started 'seq'

params:set("clock_tempo", 94 * math.random(3)) -- speed up / slow down

-- play with 'base'!
-- try canceling the clock and redefining the 'clock.sync' values in 'notes'!
```

programming reference

The bottom left ? icon can be used to navigate to the onboard programming reference.

You can manually open the API reference at `norns.local/doc` .

Also see the scripting reference.

advanced access

If you prefer to integrate norns into your existing IDE workflow, you can load maiden's REPL and manage projects through more bare-metal tools.

terminal REPL

To access maiden's REPL from a terminal session, SSH or screen into norns and execute:

```
maiden repl
```

Command Line Interface

Nearly all of the project management operations exposed in the maiden web UI can be accomplished on the Command Line Interface (CLI).

To access:

```
ssh we@norns.local
...
maiden/maiden
web editor for norns scripts

Usage:
  maiden [command]

Available Commands:
  catalog     manage the script catalog
  help        Help about any command
  project     manage dust projects
  repl        matron/crone repl
  server      run the backend server for maiden
  version     print maiden version

Flags:
      --config string   use specific config file
      --debug           enable debug logging
  -h, --help            help for maiden

Use "maiden [command] --help" for more information about a command.
```

The maiden backend server also has sub-commands:

```
ssh we@norns.local
...

#
# the catalog sub-command is useful for updating the "we" and "community"
#
~/maiden/maiden catalog help
manage the script catalog

Usage:
  maiden catalog [command]

Available Commands:
  init        init an empty catalog file
  list        list projects
  update      update configured catalogs

Flags:
  -h, --help   help for catalog

Global Flags:
      --config string   use specific config file
      --debug           enable debug logging

Use "maiden catalog [command] --help" for more information about a command.
```

```
#
# the project sub-command is installing and updating project directories
```

```
#
~/maiden/maiden project help
manage dust projects

Usage:
  maiden project [command]

Available Commands:
  install     install a script/project
  list        list installed script/project(s)
  push        push a git based project
  remove      remove a project dir
  update      update a script/project

Flags:
  -h, --help   help for project

Global Flags:
      --config string   use specific config file
      --debug           enable debug logging

Use "maiden project [command] --help" for more information about a command.
Have further usage questions? Visit the [norns: maiden](https://llllllll.co/t/norns-maiden/14052) topic on lines.
```

---

help