Integrated Development Environments (IDEs)

Adam Boult (www.bou.lt)

March 23, 2024

Contents

I vi family 2

1 vim 3

2 Neovim 5

II Emacs 6

3 emacs 7

III RStudio 13

4 RStudio 14

IV JupyterLab and Jupyter notebooks 15

5 JupyterLab 16

V Visual Studio Code 17

6 Visual Studio Code 18

VI Tmux 19

7 tmux and screen 20

VII Retro games 22

8 GB Studio 23

9 NESmaker **24**

VIII Other games **25**

10 godot **26**

11 PICO-8 **27**

12 Unity **28**

13 Unreal **29**

IX Copilots **30**

14 Github Copilot **31**

Part I

vi family

Chapter 1

vim

1.1 Introduction

1.1.1 Scripts

vim has scripts as opposed to ed/vi

page on vim script around

folding text

1.1.2 Text highlighting

1.1.3 Line wrapping

Can turn on or off with:

```
:set wrap
:set unwrap
```

1.1.4 Splitting window

Split horizontal.

```
:split
```

Split vertical

```
:vsplit
```

to navigate between windows

```
ctrl-W h/j/k/l
```

to resize

```
ctrl-W +-<>
```

to reset sizes

```
ctrl-W =
```

to swap window position

```
ctrl-W HJKL
```

1.1.5 Terminal

to verical split and add terminal to top on

```
:term
```

1.1.6 File tree

To explore file tree in command mode:

```
:Explore
```

1.1.7 Opening other files

As in vi we can do

```
:e[dit] FILE_PATH
```

```
:vi[sual] FILE_PATH
```

However these now support autocompletion.

In addition we can also use, but this seems like a legacy feature?

```
:open FILE_PATH
```

Chapter 2

Neovim

2.1 Introduction

2.1.1 Introduction

nvtree: Ctrl+n to open up. can ”c” to copy file or folder. ”p” to paste it.

nvchad: Ctrl-x in terminal mode to enter commands :bprev and :bnext to switch buffers (tabs). can also use tab and alt tab cd: /path/to/folder/ to change root folder on nvtree kill buffer with SPC x

Part II

Emacs

Chapter 3

emacs

3.1 Introduction

3.1.1 Editing text, saving and exiting

.emacs file; changing screens; opening files; search; saving

emacs mode buffer. has info on file and mode

text folding

major modes, including text mode

3.1.2 Saving and exiting

exit:

```
C-x C-c
```

Save:

```
C-x C-s
```

3.1.3 Searching

search forward:

```
C-s
```

search back:

```
C-r
```

3.1.4 Copy and paste

emacs copy paste

3.1.5 Undo and redo

3.1.6 Buffers and reading other files

read file:

```
C-x C-f
```

switch buffer:

```
C-x b
```

kill buffer

```
C-x k
```

3.1.7 dired

File browser. Installed by default.

```
C-x d
```

3.1.8 Macros

Allows you to see value of variable set:

```
M-x describe-variable
```

M-x concept. runs macros? C-sfsjfls is shortcuts for macros? so for each thing should have the M-x and C-ses equivalent?

3.2 Working with multiple windows, and shells

3.2.1 Windows

close window:

```
C-x 0
```

close other windows:

```
C-x 1
```

split on x axis:

```
C-x 2
```

split on y axis:

```
C-x 3
```

switch windows:

```
C-x o
```

Make window wider:

```
C-x }
```

Make window thinner:

```
C-x {
```

balance window size:

```
C-x +
```

make window smaller:

```
C-x -
```

3.2.2 Shell

Start a shell:

```
M-x shell
```

3.2.3 Renaming buffers and multiple shells

Can name shell so if get a second one it won't be the same.

```
C-u M-x shell
```

3.3 GUI and terminal

3.3.1 GUI and terminal

to start in terminal mode:

```
emacs -nw
```

3.3.2 GUI only: Font size and zooming

zoom out

```
C--
```

zoom in

```
C-+
```

3.4 Emacs daemon

3.4.1 Emacs daemon

3.5 Built in modes

3.5.1 Org mode

customising using conf.org file

org mode use of * use of + use of TODO: set with:

```
C-c C-t
```

see agenda files to be in agenda stored in org-agenda-files

to change state and record note:

```
C-u C-c C-t
```

cycle through agenda files:

```
C-’
```

add current file to agenda files:

```
C-c [
```

remove:

```
C-c ]
```

view agenda:

```
M-x org-agenda
```

change theme:

```
M-x load-theme
```

3.5.2 eww (Emacs Web Wowser)

Installed by default.

3.5.3 Tramp for remote access

Built into Emacs

3.6 Language specific modes, including Python and Javascript

3.6.1 Python mode

to get python interactive shell from within python mode

```
C-c C-p
```

"run-python" macro

python.el provides python mode. built into emacs

elpy (to evaluate script):

```
C-c C-c
```

jedi for autocompletion

3.6.2 Javascript

3.7 Customising Emacs and Emacs Lisp (Elisp)

3.7.1 Introduction

```
~/.emacs
~/.emacs.el
~/.emacs.d/init.el
```

3.8 Installing additional packages

3.8.1 Package mangement and MELPA and ELPA repositories

```
M-x list-packages
```

i/I d/D x to execute the activity

3.8.2 Treemacs

treemacs: R to rename yf to copy file cf to create file cd to create directory

treemacs-select-window to go to treemacs window

3.8.3 Magit

magit-status to see magit status

magit: press s to stage a file

For git

3.8.4 Projectile

projects, projectile projectile-add-known-project; projectile-switch-project

3.8.5 Company-mode

COMPlete ANY

3.8.6 Need to install: Evil mode

Use vi style bindings.

Part III

RStudio

Chapter 4

RStudio

4.1 Introduction

4.1.1 Introduction

Part IV

JupyterLab and Jupyter notebooks

Chapter 5

JupyterLab

5.1 Introduction

5.1.1 Introduction

Part V

Visual Studio Code

Chapter 6

Visual Studio Code

6.1 Introduction

6.1.1 Introduction

ctrl P

use of @

use of #

Ctrl G

alt up or down to move line up or down

ctrl / to toggle comments on lines Ctrl ‘ to open terminal

F5 run code

Extensions

Part VI

Tmux

Chapter 7

tmux and screen

7.1 Introduction

7.1.1 Introduction

Ctrl-b then something

7.1.2 Working with multiple windows

c - create new window

w - choose window

0,1,2 etc - choose window 0 etc

, - rename

7.1.3 Working with multiple panes

“ - split vertically

% - split horizonatally

o - next pane

arrows - change pane

7.1.4 Sessions

d detach

```
tmux ls
tmux rename-session
tmux new -s session_name
```

Can attach to a session rather than start a new one.

`tmux attach`

tmux kill-session

can preserve sessions over reboots?

7.1.5 Plugins

Part VII

Retro games

Chapter 8

GB Studio

8.1 Introduction

8.1.1 Introduction

Chapter 9

NESmaker

9.1 Introduction

9.1.1 Introduction

Part VIII

Other games

Chapter 10

godot

10.1 Introduction

10.1.1 Introduction

Chapter 11

PICO-8

11.1 Introduction

11.1.1 Introduction

Chapter 12

Unity

12.1 Introduction

12.1.1 Introduction

Chapter 13

Unreal

13.1 Introduction

13.1.1 Introduction

Part IX

Copilots

Chapter 14

Github Copilot

14.1 Introduction

14.1.1 Introduction

Can be used with VS Code, vim, neovim, emacs.