Building modern desktop apps in Go

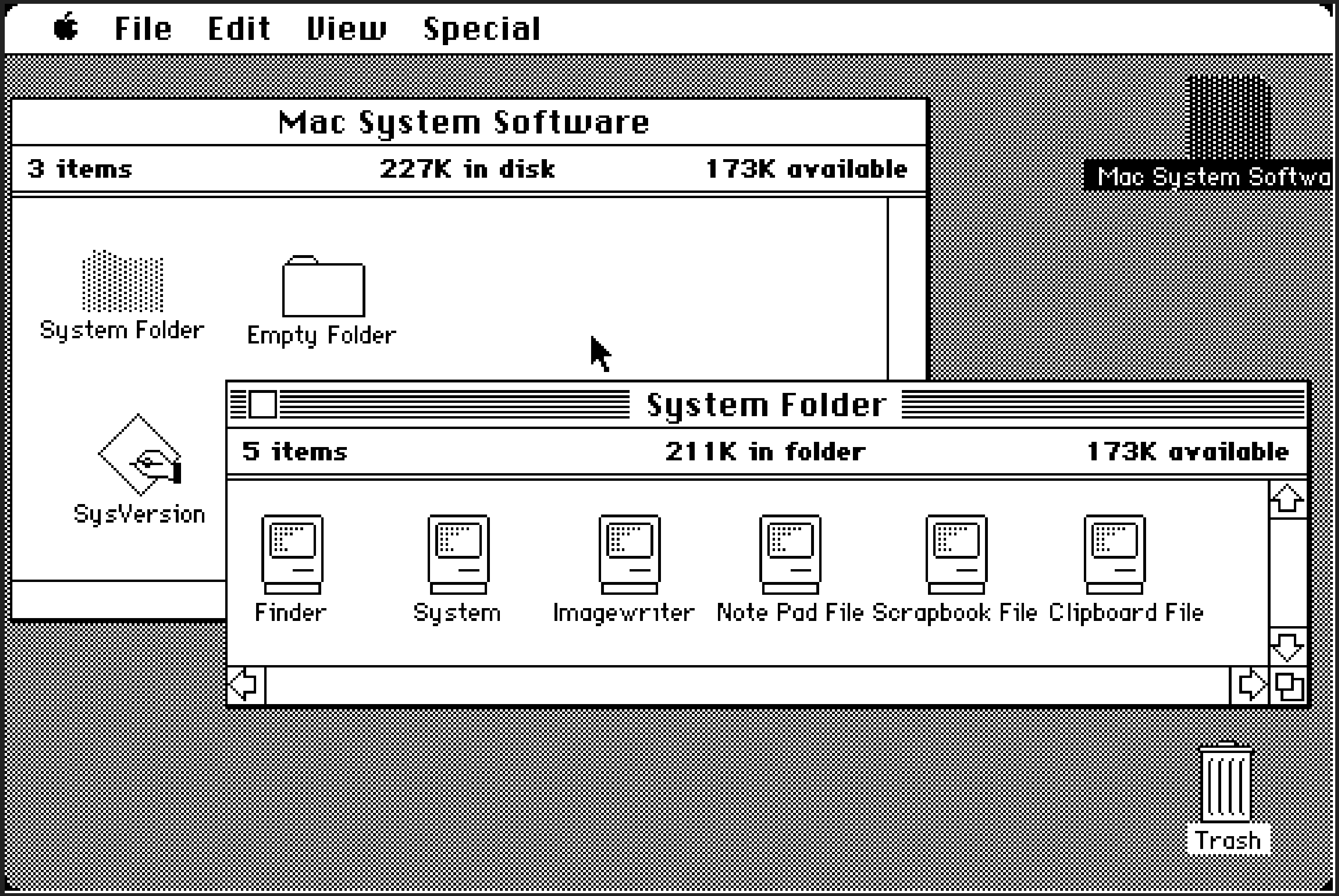
Serge Zaitsev / zserge.com / @zsergo

Mobile?
Wearable?
Desktop!

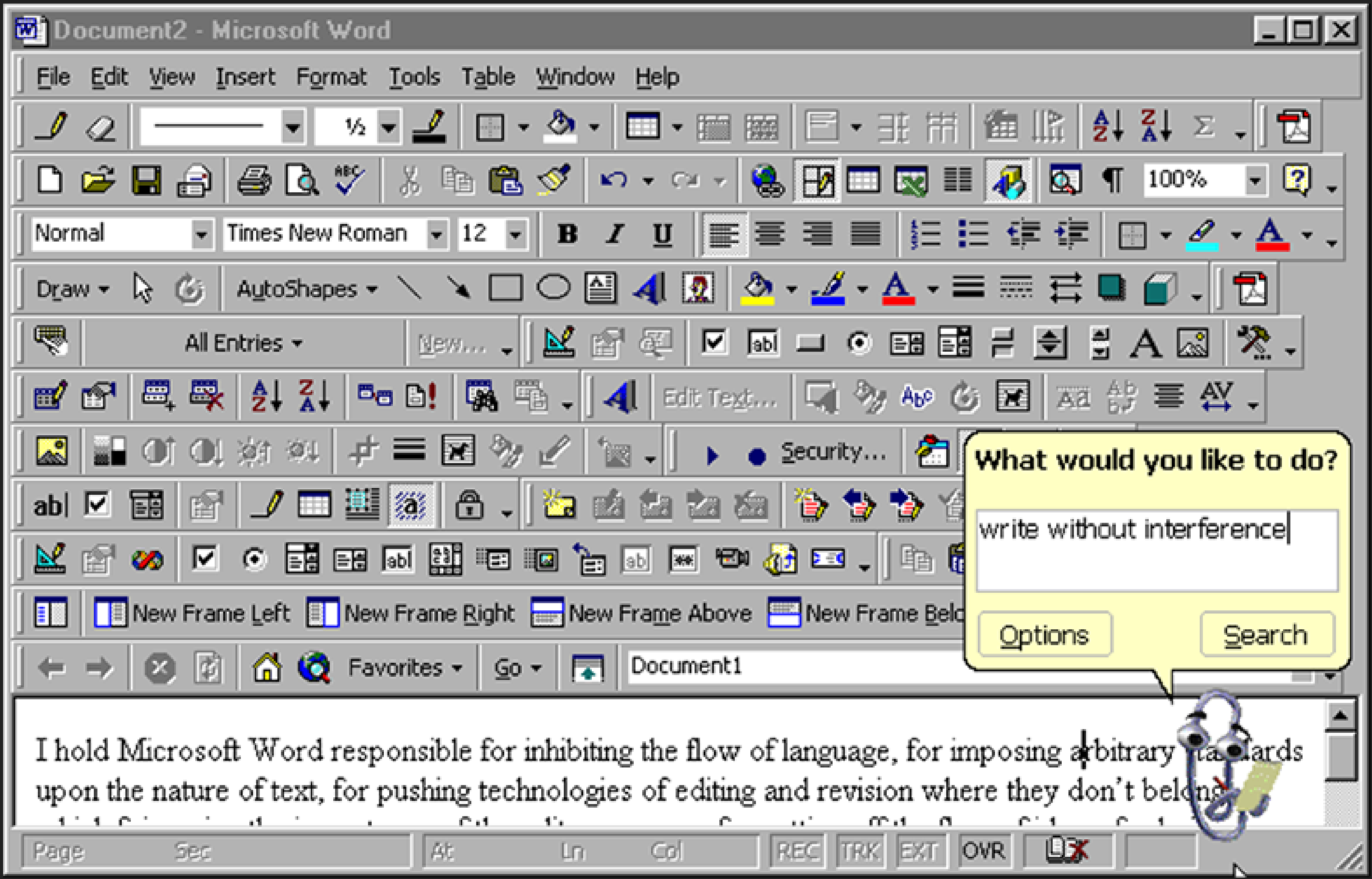
For getting job done with high performance, security and privacy.

WIMP Era

Window, Icon, Menu, Pointer. Ruling the world since 1973.

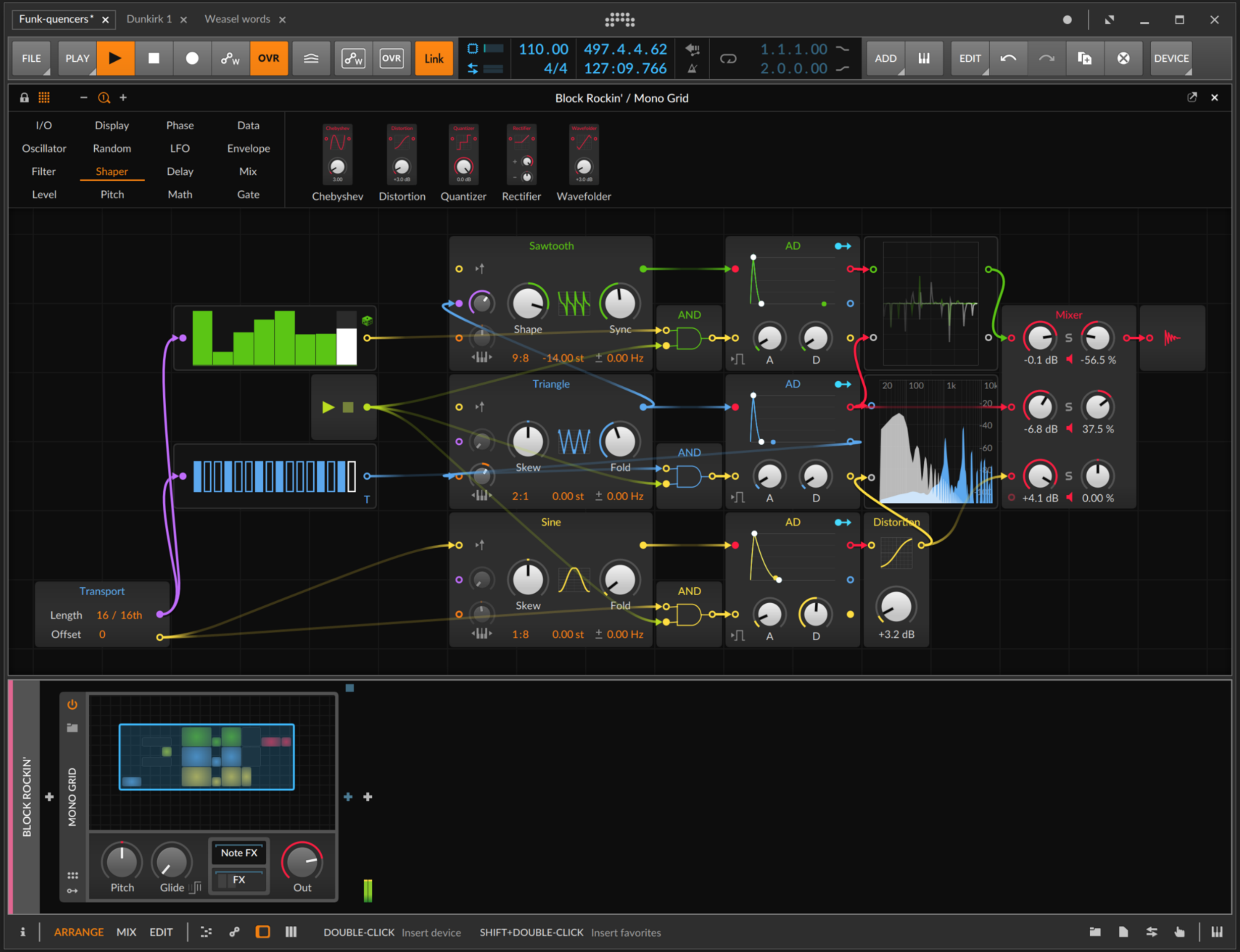


The Golden Age of WIMP UI


Document2 - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Normal
Times New Roman
12
100%
Draw
AutoShapes
All Entries
Security...
New Frame Left
New Frame Right
New Frame Above
Favorites
Go
Document1
What would you like to do?
write without interference
Options
Search
Page Sec At Ln Col REC TRK EXT OVR

I hold Microsoft Word responsible for inhibiting the flow of language, for imposing arbitrary standards upon the nature of text, for pushing technologies of editing and revision where they don't belong

Post-WIMP Era (still evolving)



Well-designed and attractive typography makes content meaningful.

Animation makes important parts stand out and helps with micro-interactions.

Onboarding, common for web and mobile apps, helps more than traditional "F1" button.

Visual trends may change and it should not be hard to update the UI without rewriting it all.

Electron

Is

Bad.

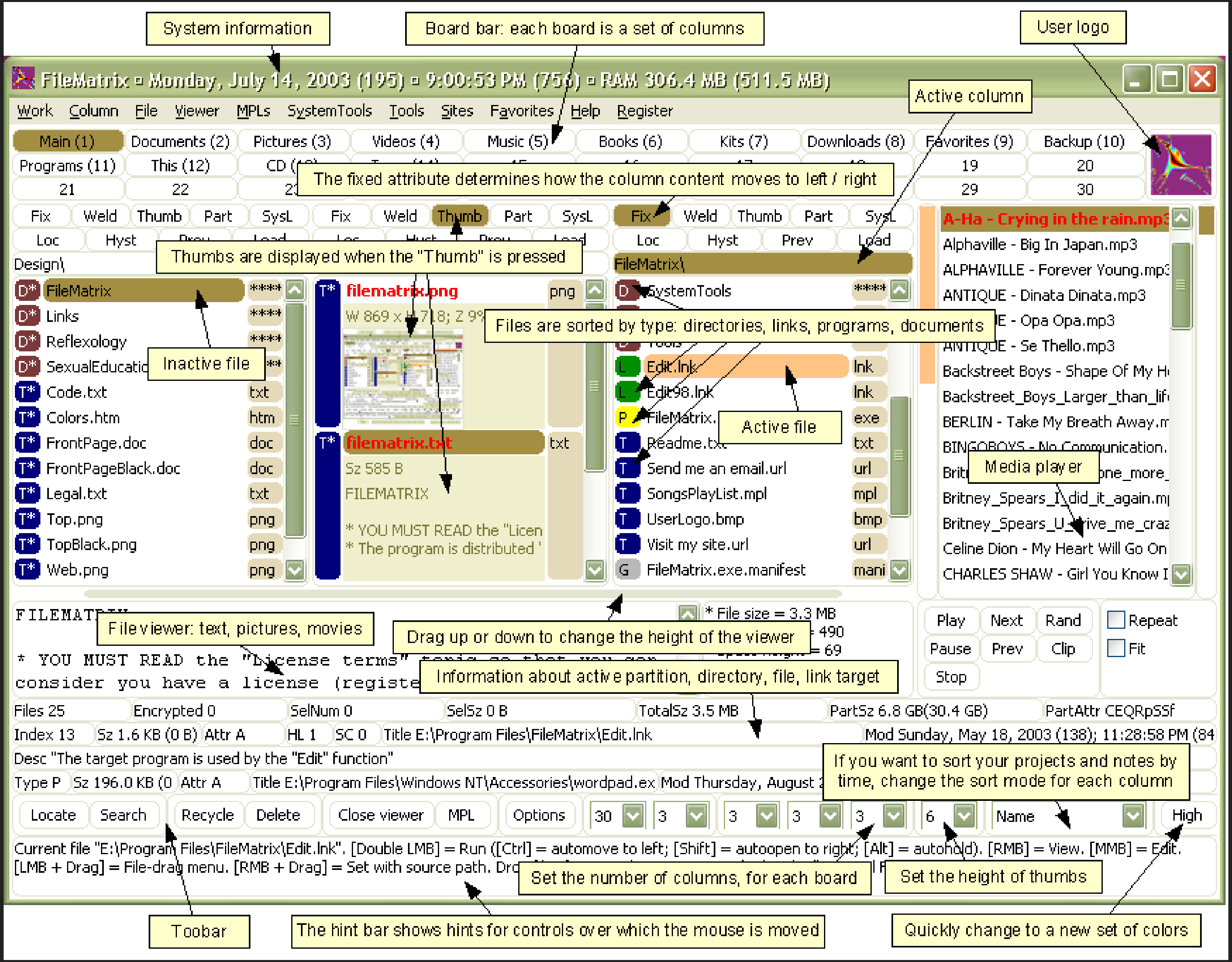
What do we want?

Common UI API for all platforms because we are lazy.

Support for existing UI frameworks and UI kits (material, flat, etc) because we often can't design UI.

Painless development, debugging, packaging and distribution because we have more important things to do.

When programmers make UI


System information
Board bar: each board is a set of columns
User logo
Active column
The fixed attribute determines how the column content moves to left / right
Thumbs are displayed when the "Thumb" is pressed
Files are sorted by type: directories, links, programs, documents
Inactive file
Active file
Media player
File viewer: text, pictures, movies
Drag up or down to change the height of the viewer
Information about active partition, directory, file, link target
If you want to sort your projects and notes by time, change the sort mode for each column
Set the number of columns, for each board
Set the height of thumbs
Toobar
The hint bar shows hints for controls over which the mouse is moved
Quickly change to a new set of colors

FileMatrix ¤ Monday, July 14, 2003 (195) ¤ 9:00:53 PM (756) ¤ RAM 306.4 MB (511.5 MB)

Work Column File Viewer MPLs SystemTools Tools Sites Favorites Help Register

Main (1) Documents (2) Pictures (3) Videos (4) Music (5) Books (6) Kits (7) Downloads (8) Favorites (9) Backup (10)
Programs (11) This (12) CD 19 20
21 22 29 30

Fix Weld Thumb Part SysL | Fix Weld Thumb Part SysL | Fix Weld Thumb Part SysL
Loc Hyst | Loc Hyst Prev Load

Design\ | FileMatrix\

D*	FileMatrix	****
D*	Links	****
D*	Reflexology	****
D*	SexualEducatio	**
T*	Code.txt	txt
T*	Colors.htm	htm
T*	FrontPage.doc	doc
T*	FrontPageBlack.doc	doc
T*	Legal.txt	txt
T*	Top.png	png
T*	TopBlack.png	png
T*	Web.png	png

T* filematrix.png png
W 869 x H 718; Z 9
T* filematrix.txt txt
Sz 585 B
FILEMATRIX
* YOU MUST READ the "Licen
* The program is distributed

D	SystemTools	****
	Tools	
L	Edit.lnk	lnk
L	Edit98.lnk	lnk
P	FileMatrix.	exe
T	Readme.txt	txt
T	Send me an email.url	url
T	SongsPlayList.mpl	mpl
T	UserLogo.bmp	bmp
T	Visit my site.url	url
G	FileMatrix.exe.manifest	mani

A-Ha - Crying in the rain.mp3
Alphaville - Big In Japan.mp3
ALPHAVILLE - Forever Young.mp3
ANTIQUE - Dinata Dinata.mp3
E - Opa Opa.mp3
ANTIQUE - Se Thello.mp3
Backstreet Boys - Shape Of My H
Backstreet_Boys_Larger_than_lif
BERLIN - Take My Breath Away.m
BINGOBOYS - No Communication.
Brit one_more_
Britney_Spears_I_did_it_again.m
Britney_Spears_U_drive_me_craz
Celine Dion - My Heart Will Go On
CHARLES SHAW - Girl You Know I

```
FILEMAT

* YOU MUST READ the "License terms"
consider you have a license (registe
```

* File size = 3.3 MB
490
= 69

Play Next Rand Repeat
Pause Prev Clip Fit
Stop

Files 25 | Encrypted 0 | SelNum 0 | SelSz 0 B | TotalSz 3.5 MB | PartSz 6.8 GB(30.4 GB) | PartAttr CEQRpSSf
Index 13 | Sz 1.6 KB (0 B) Attr A | HL 1 | SC 0 | Title E:\Program Files\FileMatrix\Edit.lnk | Mod Sunday, May 18, 2003 (138); 11:28:58 PM (84
Desc "The target program is used by the "Edit" function"
Type P | Sz 196.0 KB (0 | Attr A | Title E:\Program Files\Windows NT\Accessories\wordpad.ex | Mod Thursday, August

Locate Search | Recycle Delete | Close viewer MPL | Options | 30 | 3 | 3 | 3 | 3 | 6 | Name | High

Current file "E:\Program Files\FileMatrix\Edit.lnk". [Double LMB] = Run ([Ctrl] = automove to left; [Shift] = autoopen to right; [Alt] = autohold). [RMB] = View. [MMB] = Edit.
[LMB + Drag] = File-drag menu. [RMB + Drag] = Set with source path. Dr

Option 1.

Reuse some modern browser that is already installed.

Option 2.

Reuse the browser engine that comes with the OS (and every desktop OS now comes with a decent browser engine).

Lorca

github.com/zserge/lorca

Lorca

Chrome DevTools Protocol

Bind Go functions to JS

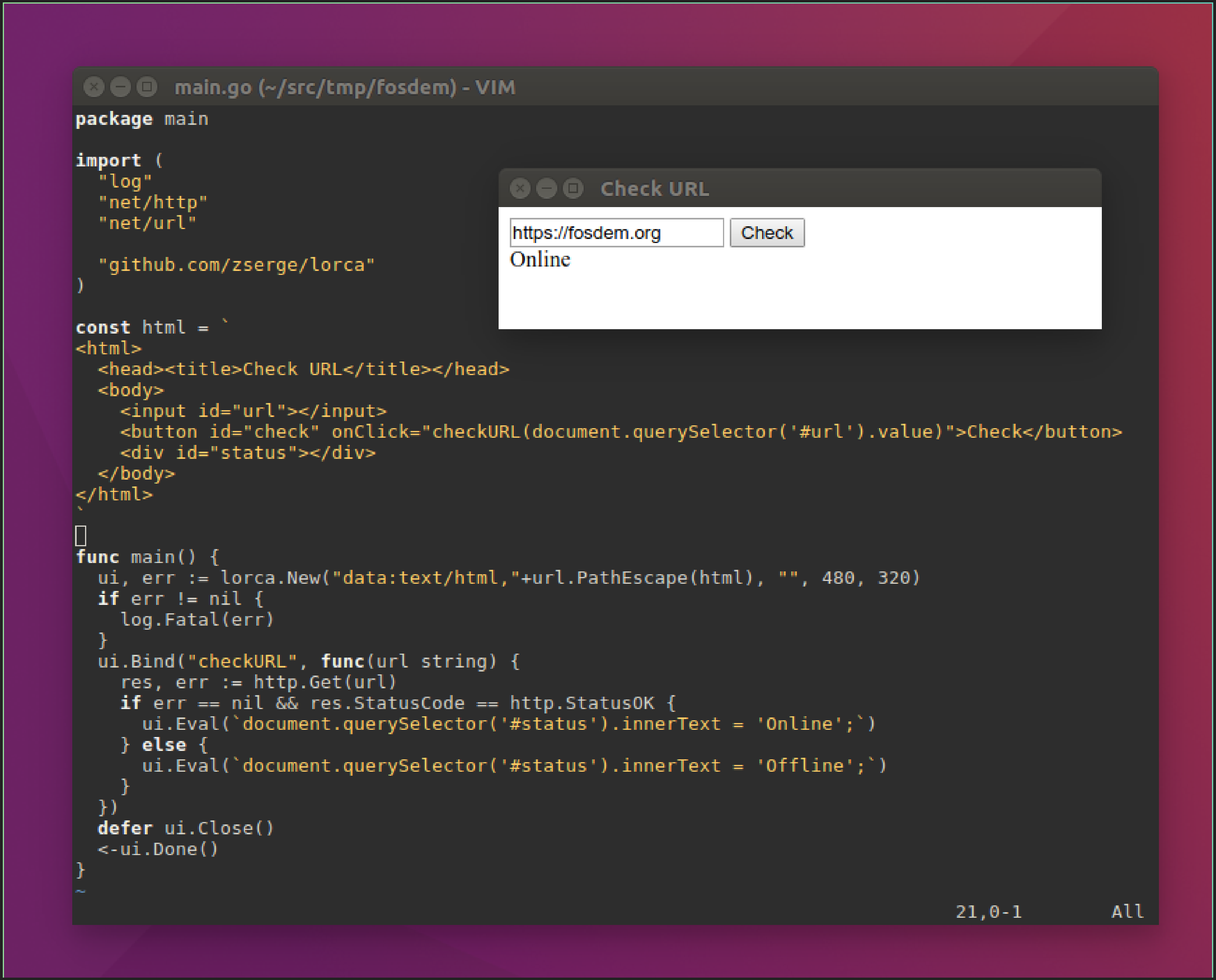
Call JS from Go

Control native window

API

```
ui, _ := lorca.New(...)
ui.Load(url)
ui.Bind("someFunc", func() {})
five := ui.Eval("2+3").Int()
<-ui.Done()
```

Example



main.go (~/src/tmp/fosdem) - VIM

```go
package main

import (
  "log"
  "net/http"
  "net/url"

  "github.com/zserge/lorca"
)

const html = `
<html>
  <head><title>Check URL</title></head>
  <body>
    <input id="url"></input>
    <button id="check" onClick="checkURL(document.querySelector('#url').value)">Check</button>
    <div id="status"></div>
  </body>
</html>
`

func main() {
  ui, err := lorca.New("data:text/html,"+url.PathEscape(html), "", 480, 320)
  if err != nil {
    log.Fatal(err)
  }
  ui.Bind("checkURL", func(url string) {
    res, err := http.Get(url)
    if err == nil && res.StatusCode == http.StatusOK {
      ui.Eval(`document.querySelector('#status').innerText = 'Online';`)
    } else {
      ui.Eval(`document.querySelector('#status').innerText = 'Offline';`)
    }
  })
  defer ui.Close()
  <-ui.Done()
}
~
                                                                21,0-1        All
```

Check URL

https://fosdem.org

Online

Lorca!

Minimal (1KLOC). One dependency (websocket lib)

Simple API.

ES6 and modern CSS without Babel.

Decent debugger.

Lorca?

Window can't have fixed size.

Window global menu can't be controlled.

It still behaves more like a browser rather than an app.

WebView

github.com/zserge/webview

GTK+
GtkWebkit2

WinAPI
MSHTML (OLE)

Cocoa
WKWebView

WinAPI
EdgeHtml (winrt)

Window

Set title.

Set size.

Optional: other window flags (border, full-screen mode, transparency)

Minimize/maximize/restore

…

(BYOF - bring your own features)

Browser

Load arbitrary URL (including data URIs).

Initialize with JS code when new page is loaded (before DOM is ready).

Evaluate JS code any time later.

Call native callback with a string argument from JS.

Go

```
ui, _ := webview.New()
ui.SetTitle("Hello")
ui.Load(url)
ui.Bind("foo", func(s string) {})
defer ui.Close()
ui.Run()
```

C++

```
webview w;
w.set_title("Hello");
w.load(url);
w.bind("foo", [](string arg){})
w.run()
```

Lorca	WebView
Disk: ~10MB	Disk: ~10MB
RAM: ~80MB	RAM: ~6MB
CPU: 2%	CPU: ~1.3%

Note: benchmarks are useless, just try it yourself.

Questions?

github.com/zserge/lorca

github.com/zserge/webview