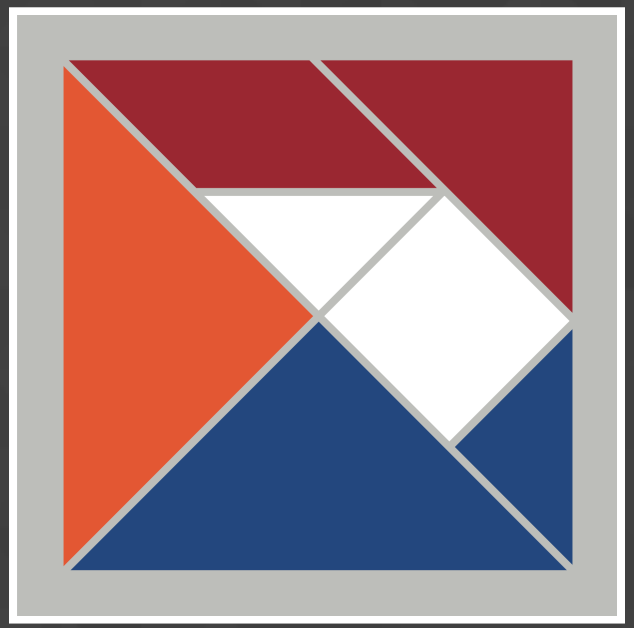
INITIAL ELM NETHERLANDS

2016-08-03, Mats Stijlaart (@stil4m)

@ElmNetherlands




SMART SEARCH & MATCH


AVISI

INTRODUCTION

- Elm and friends
- Build a community
- Adjust to the needs
- Have some fun!

ELM NETHERLANDS
+
ELM AMSTERDAM

TODAY

WHAT?

BUZZWORD BINGO

- Functional
- Immutable
- Statically typed
- Strongly typed
- Pure
- Memoization
- Virtual DOM

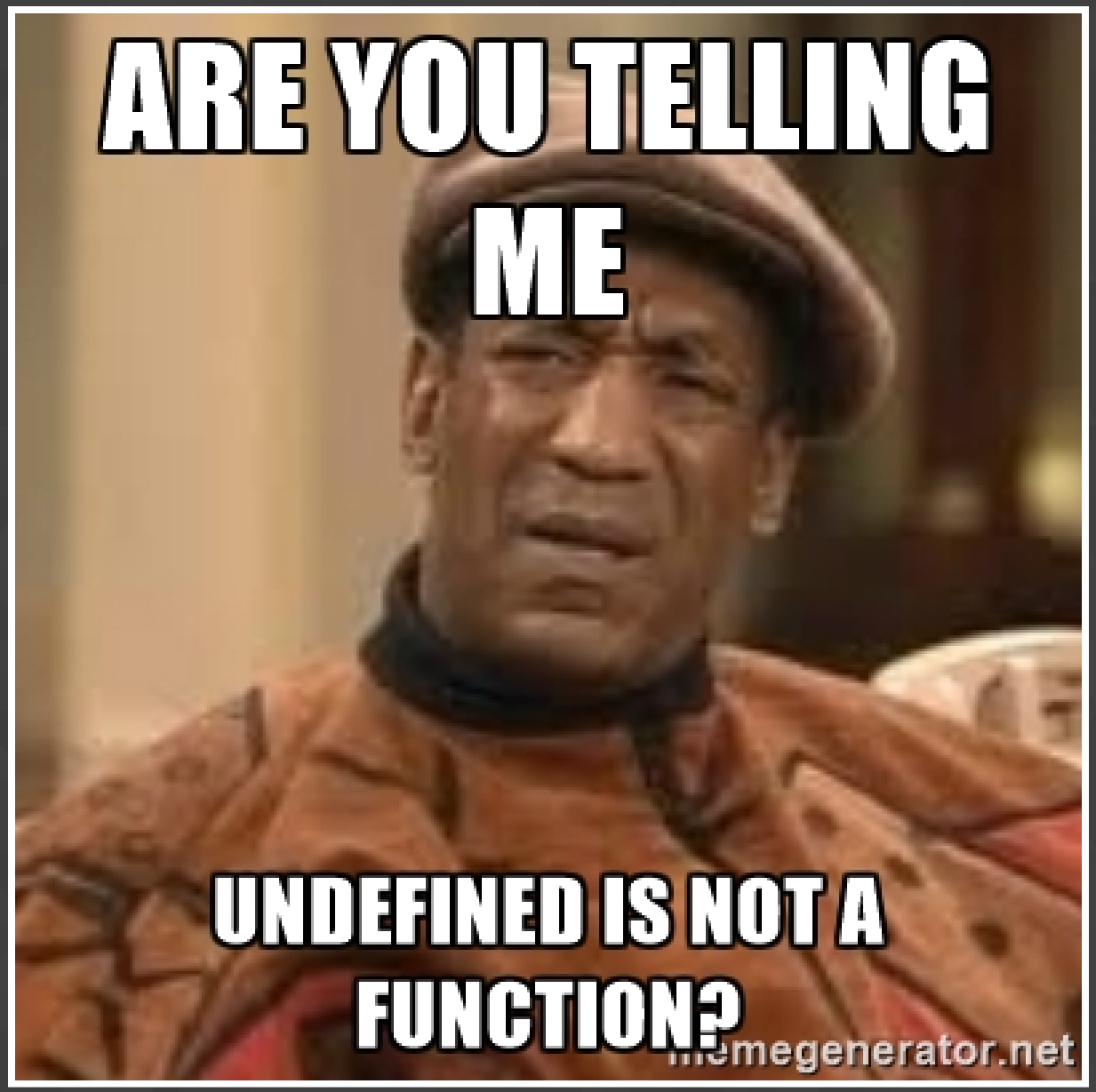
VE RI TAS

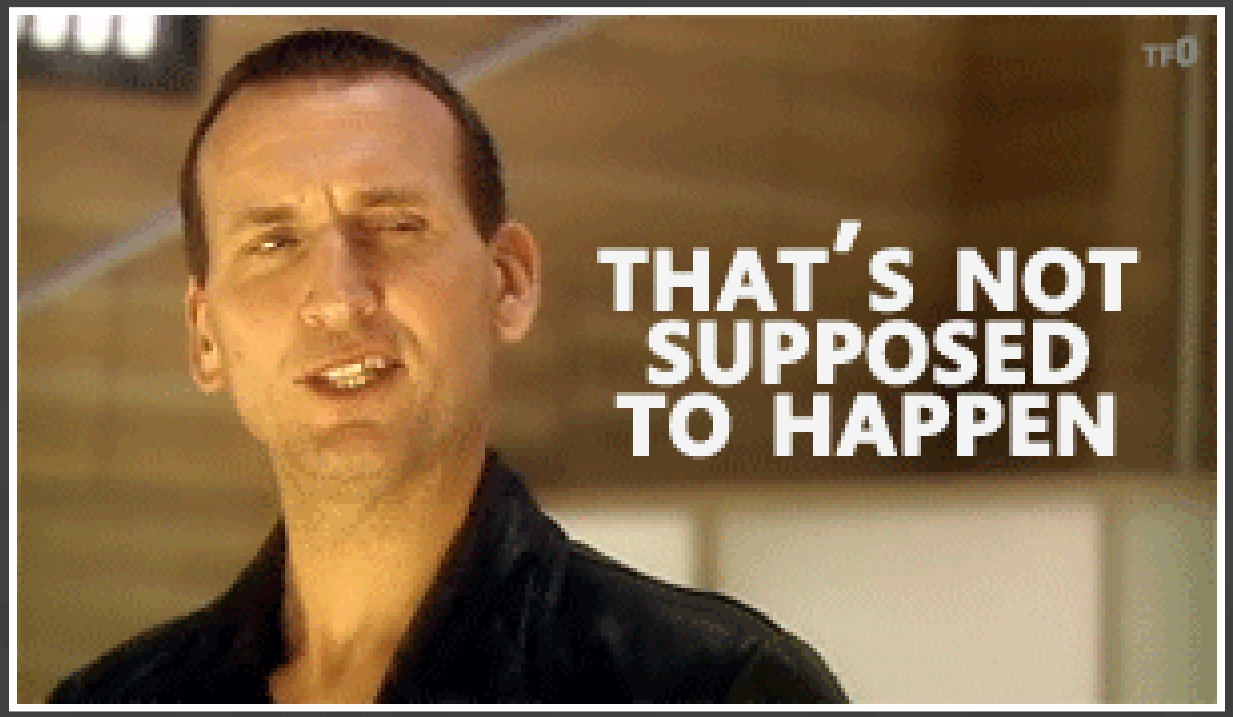
HARVARD

Prezi

noredink

WHY?


ARE YOU TELLING ME
UNDEFINED IS NOT A FUNCTION?
memegenerator.net


THAT'S NOT
SUPPOSED
TO HAPPEN

Runtime exceptions reduced to (nearly) zero

Batteries included

- The Elm Architecture (TEA)
- No HTML and JS, just Elm

Designed to solve common problems.

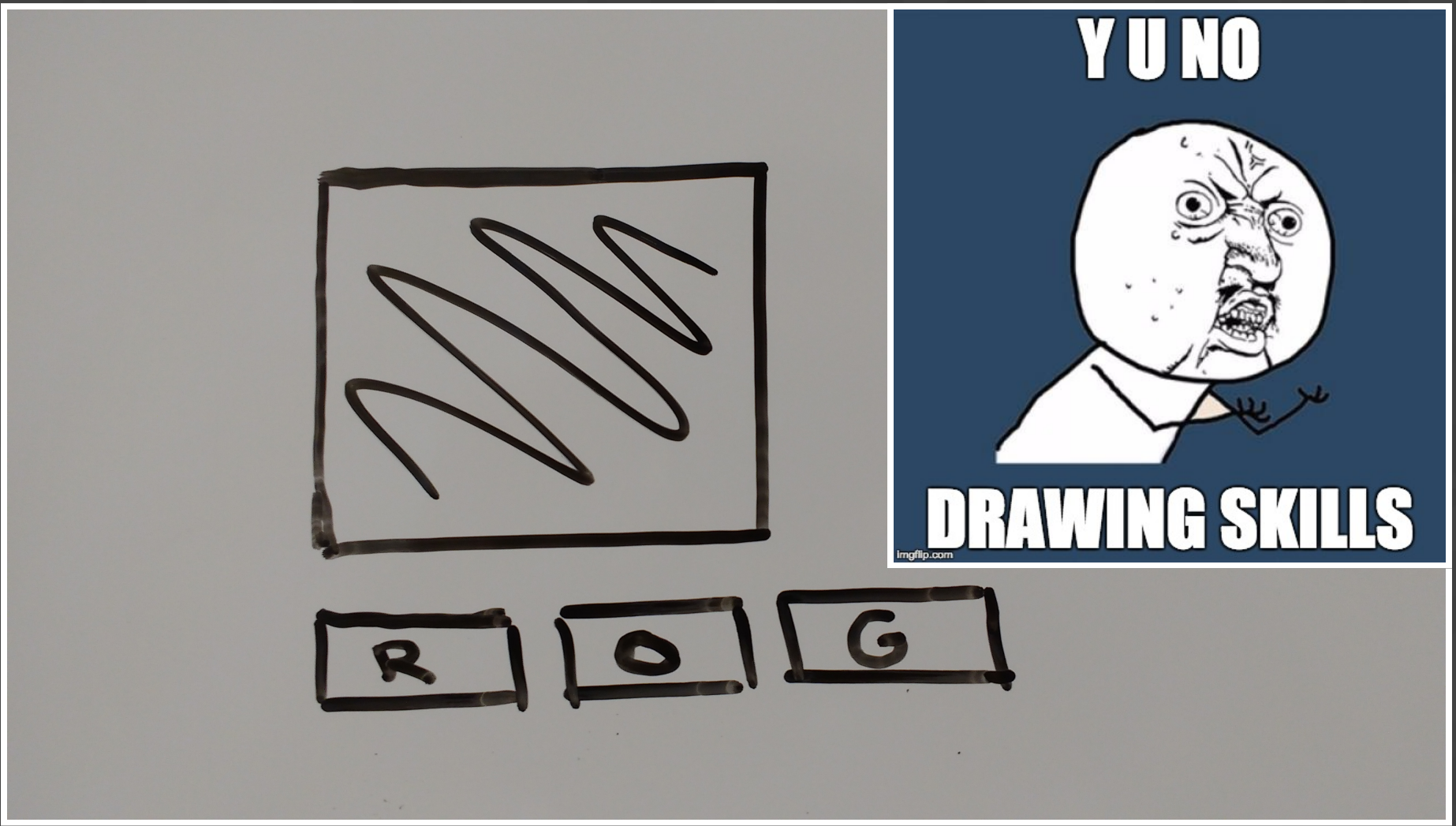
- Package manager
- Formatting

We can assume certain things from our software, thus it would require less testing.

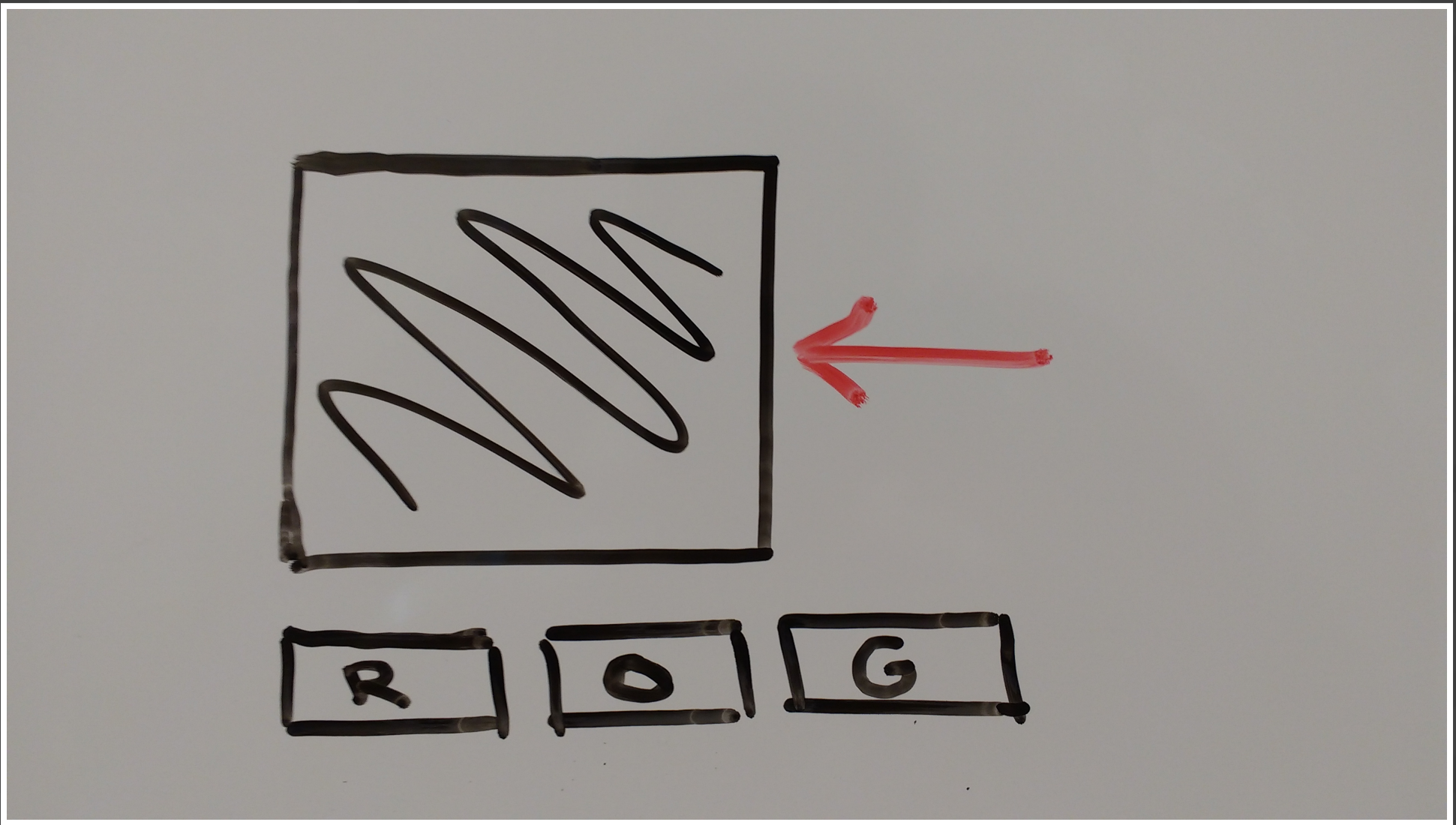
DEMO

We are going to build a simple application using The Elm Architecture

A TRAFFIC LIGHT



MODEL THE PROBLEM

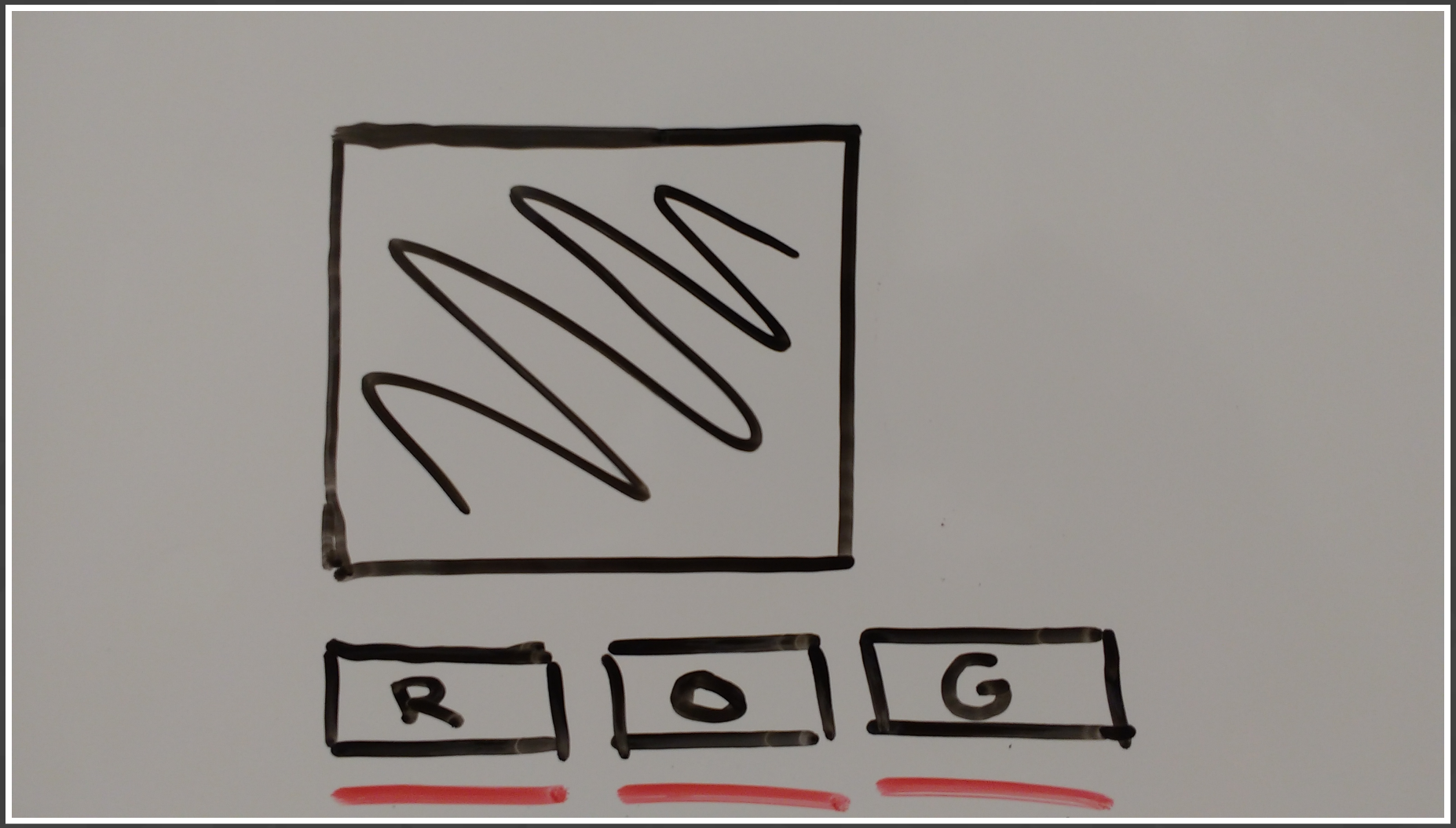


MODEL THE PROBLEM

```
type alias Model =
    Color

type Color
    = Green
    | Orange
    | Red
```

WHICH ACTIONS SHOULD I HANDLE?



WHICH ACTIONS SHOULD I HANDLE?

```
type Msg
    = SetGreen
    | SetOrange
    | SetRed
```

MODEL, VIEW AND UPDATE

INITIAL MODEL

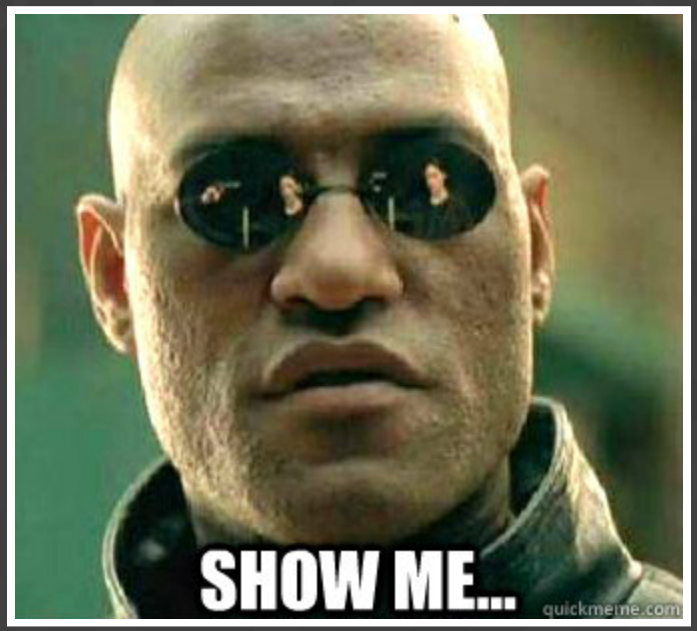
```
model : Model
model =
    Green
```

VIEW

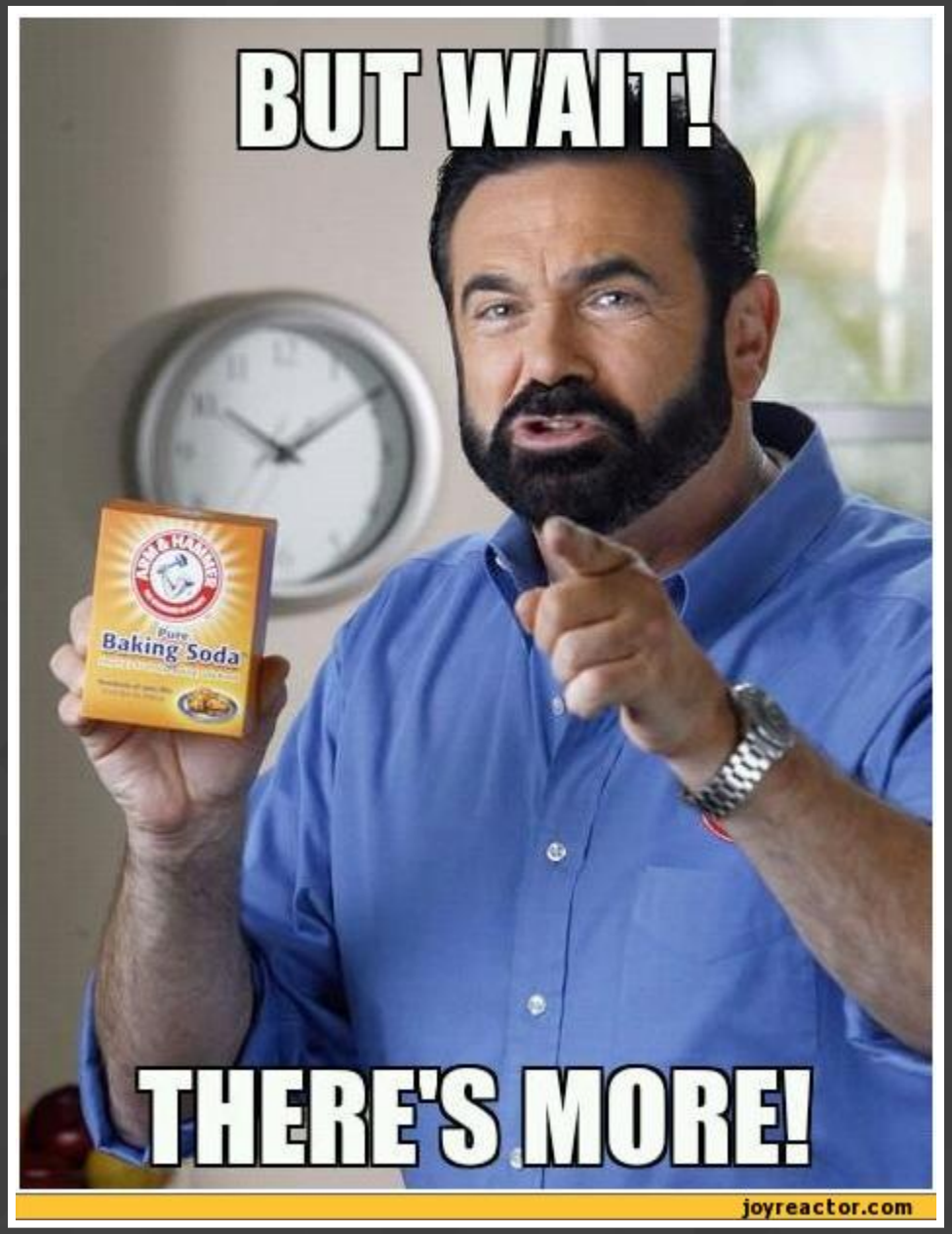
```
view : Model -> Html Msg
view model =
    let
        colorText =
            colorToString model
    in
        div []
            [ div [ style (lightStyle colorText) ] []
            , button [ onClick SetGreen ] [ text (colorToString Green) ]
            , button [ onClick SetOrange ] [ text (colorToString Orange) ]
            , button [ onClick SetRed ] [ text (colorToString Red) ]
            ]
```

UPDATE

```
update : Msg -> Model -> Model
update msg model =
    case msg of
        SetGreen ->
            Green
        SetOrange ->
            Orange
        SetRed ->
            Red
```


SHOW ME...
quickmeme.com

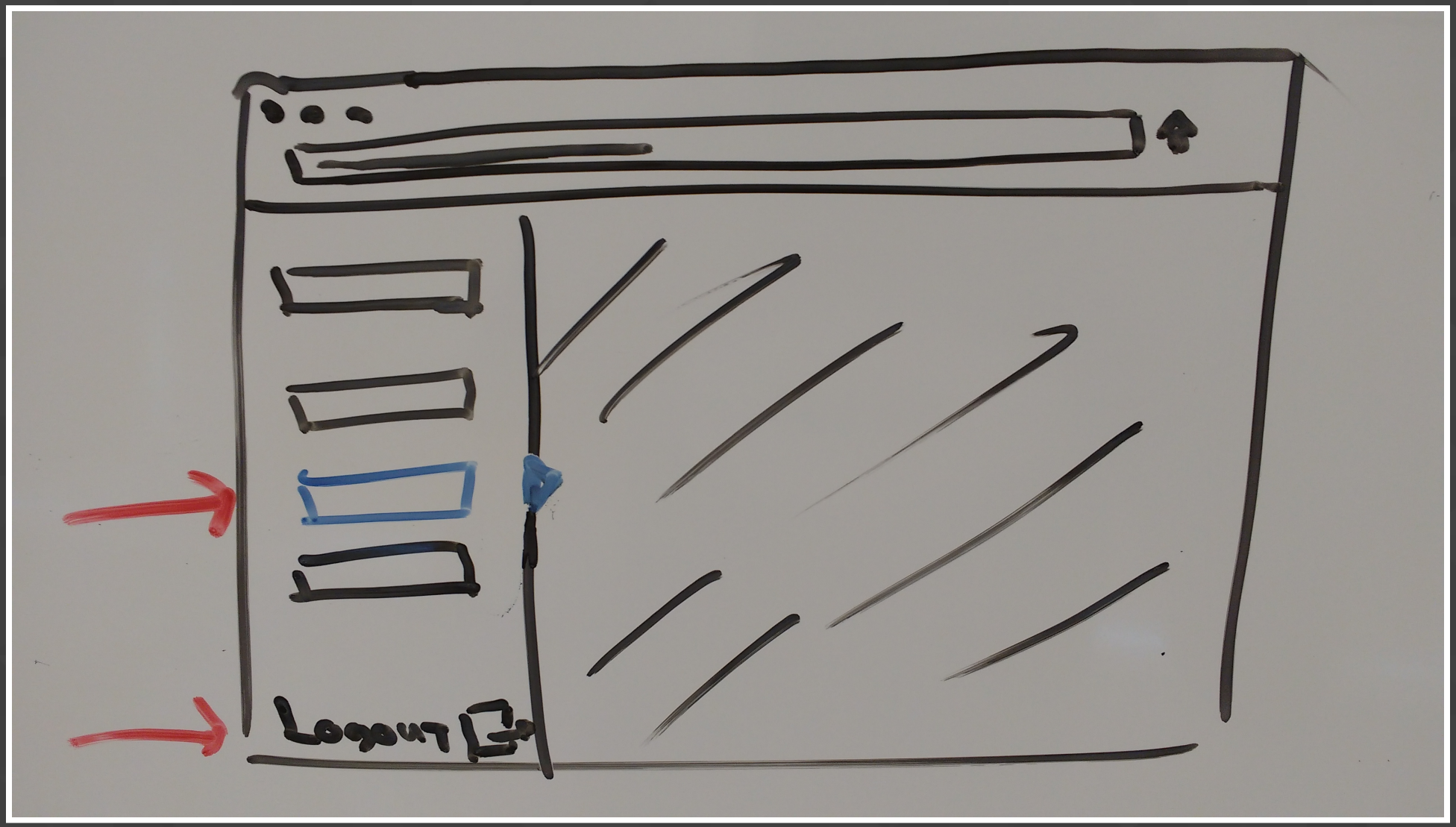

AT FIRST YOU HAD MY CURIOSITY
BUT NOW YOU HAVE MY ATTENTION
quickmeme.com

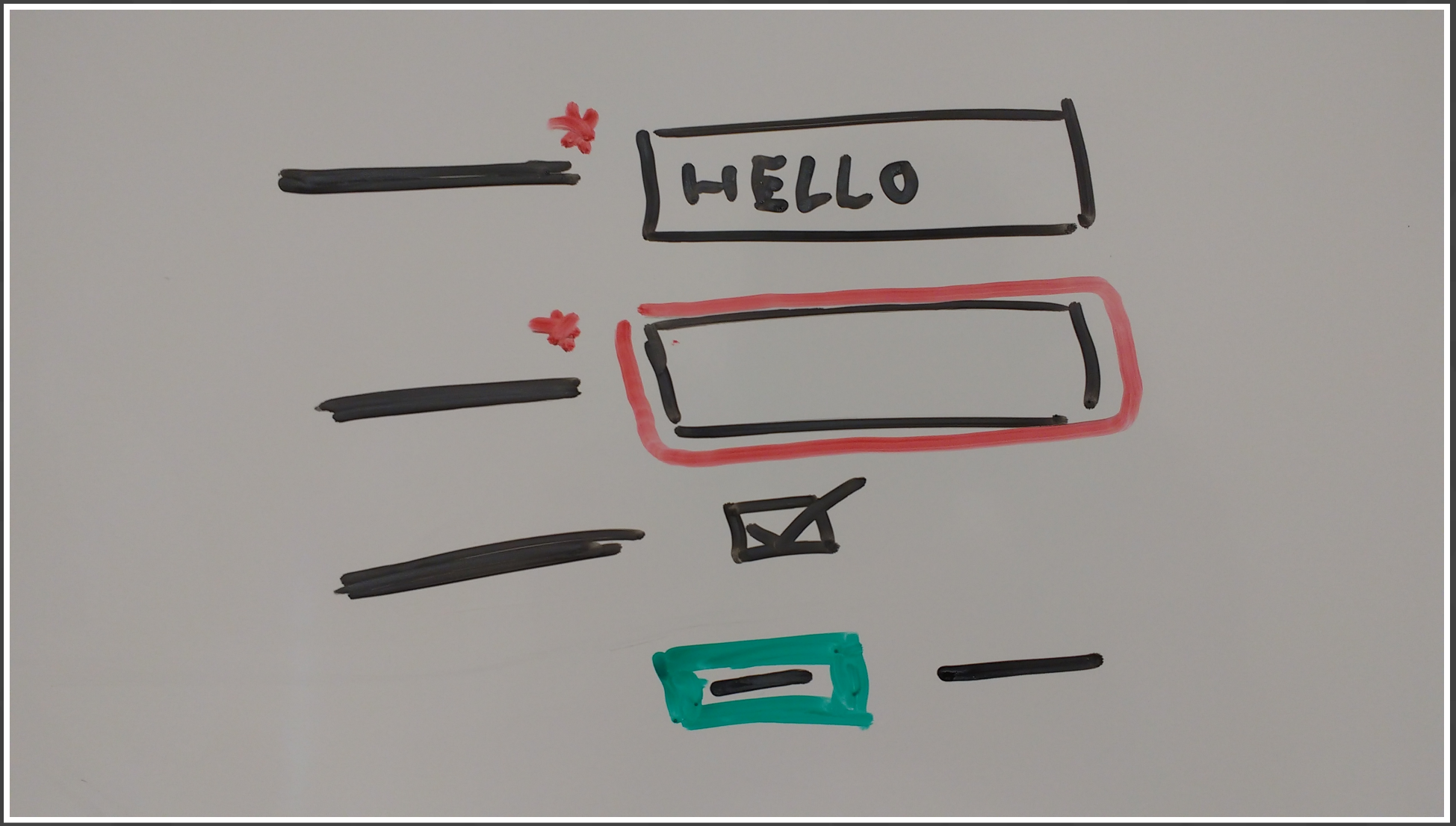

BUT WAIT!
Pure
Baking Soda
THERE'S MORE!
joyreactor.com

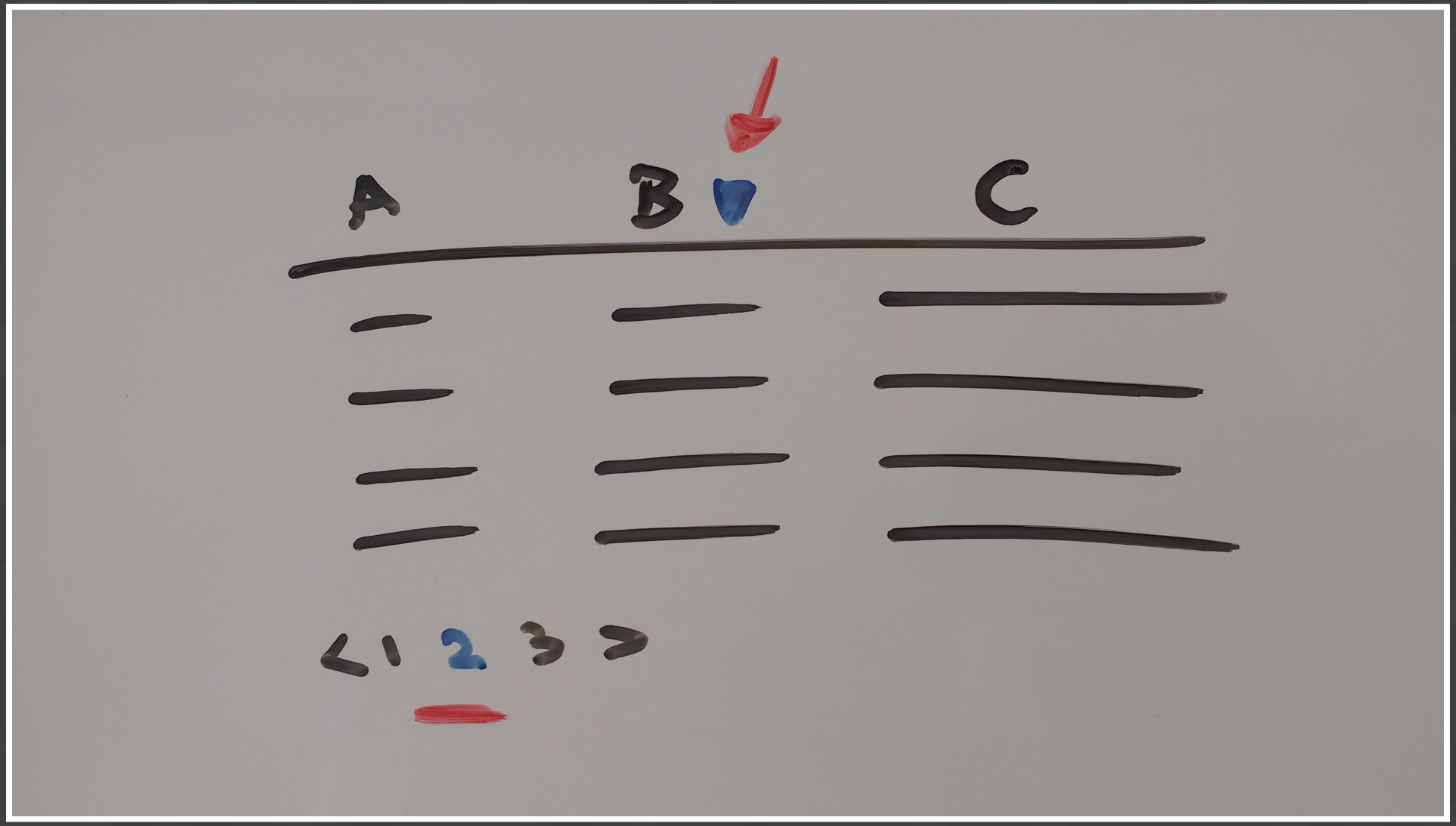
- ~~Easy~~ Safe to extend
- Reusable
- Purity enables property based testing
- Encapsulation

RECAP

- We cant break it, only implement the specs wrong
- It is modular
- Easy and safe to extend
- The Elm Architecture: `model`, `view` and `update`


Logout


HELLO


A
B
C
< 1 2 3 >

'WHY NOT'

- 0.X -> 0.Y -> 0.Z
- Documentation may be scarce
- It must fit the model
- 'There is a library for that'

SOURCE

- http://elm-lang.org
- https://github.com/isRuslan/awesome-elm
- http://www.elmweekly.nl/
- Elm Lang Slack
- https://github.com/stil4m/elm-netherlands-meetup

QUESTIONS?

AVISI