Lecture 5:
Designing smart systems

Using statistical methods to anticipate user needs and actions with Bayesian strategies

Overview of the course

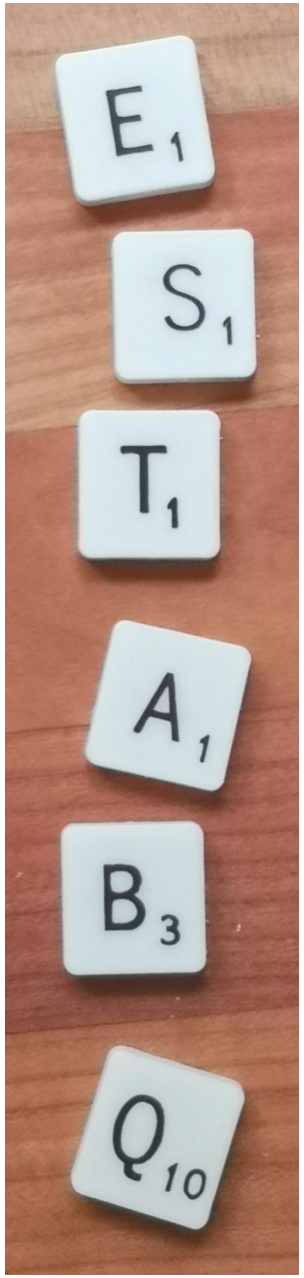
- Theory driven approaches to HCI
- Design of visual displays
- Goal-oriented interaction
- Designing efficient systems
- **Designing smart systems**
- Designing meaningful systems (guest lecturer)
- Evaluating interactive system designs
- Designing complex systems

Uniform text entry

Apple UK

Information gain per key press

$$h(x_i) = \log \frac{1}{p(x_i)}$$



The q?

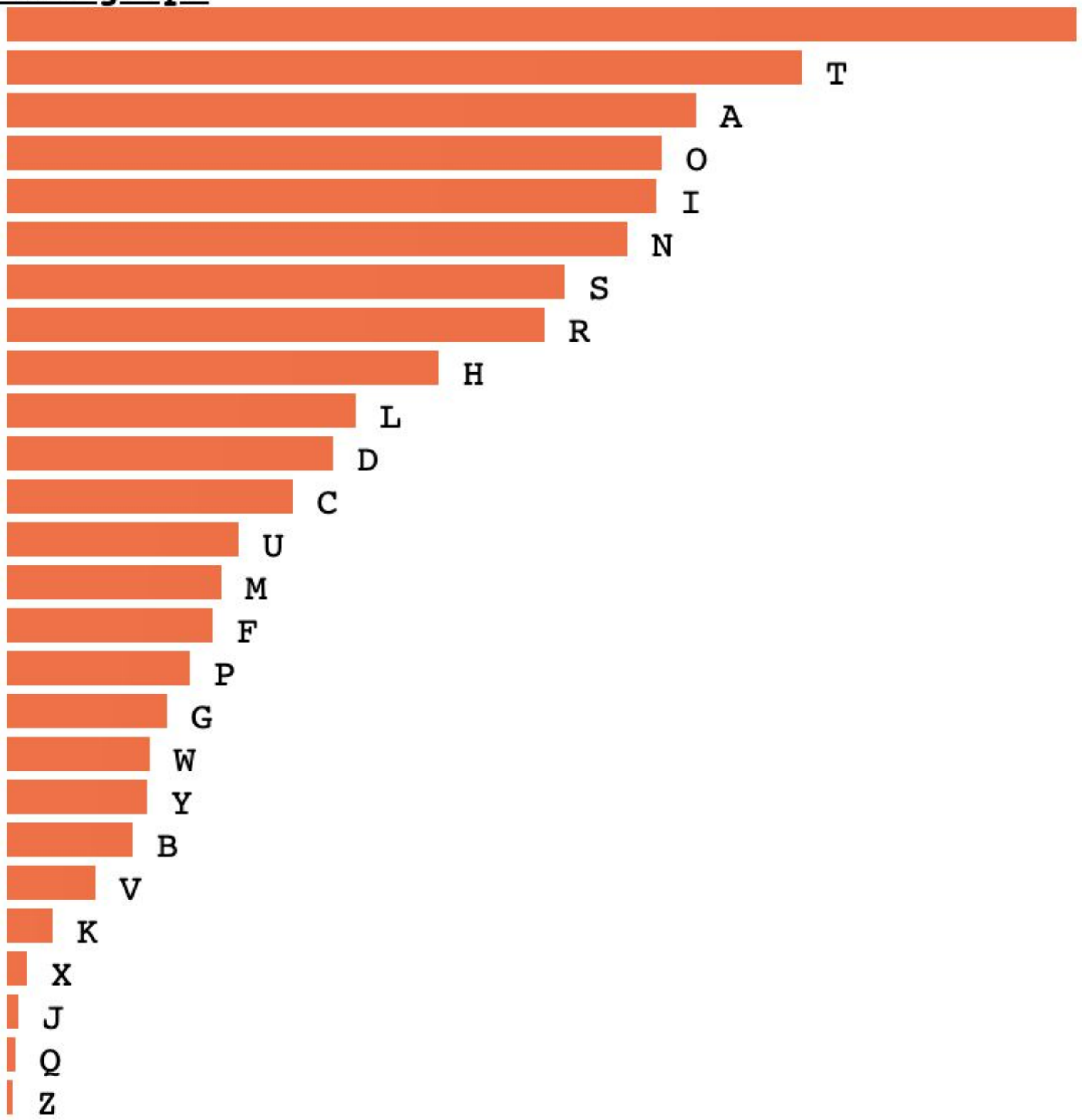
Information gain per key press

"As you are aware, E is the most common letter in the English alphabet, and it predominates to so marked an extent that even in a short sentence one would expect to find it most often"

The Adventure of the Dancing Men,

Sir Arthur Conan Doyle

LET	COUNT	PERCENT	bar graph
E	445.2 B	12.49%	E
T	330.5 B	9.28%	T
A	286.5 B	8.04%	A
O	272.3 B	7.64%	O
I	269.7 B	7.57%	I
N	257.8 B	7.23%	N
S	232.1 B	6.51%	S
R	223.8 B	6.28%	R
H	180.1 B	5.05%	H
L	145.0 B	4.07%	L
D	136.0 B	3.82%	D
C	119.2 B	3.34%	C
U	97.3 B	2.73%	U
M	89.5 B	2.51%	M
F	85.6 B	2.40%	F
P	76.1 B	2.14%	P
G	66.6 B	1.87%	G
W	59.7 B	1.68%	W
Y	59.3 B	1.66%	Y
B	52.9 B	1.48%	B
V	37.5 B	1.05%	V
K	19.3 B	0.54%	K
X	8.4 B	0.23%	X
J	5.7 B	0.16%	J
Q	4.3 B	0.12%	Q
Z	3.2 B	0.09%	Z

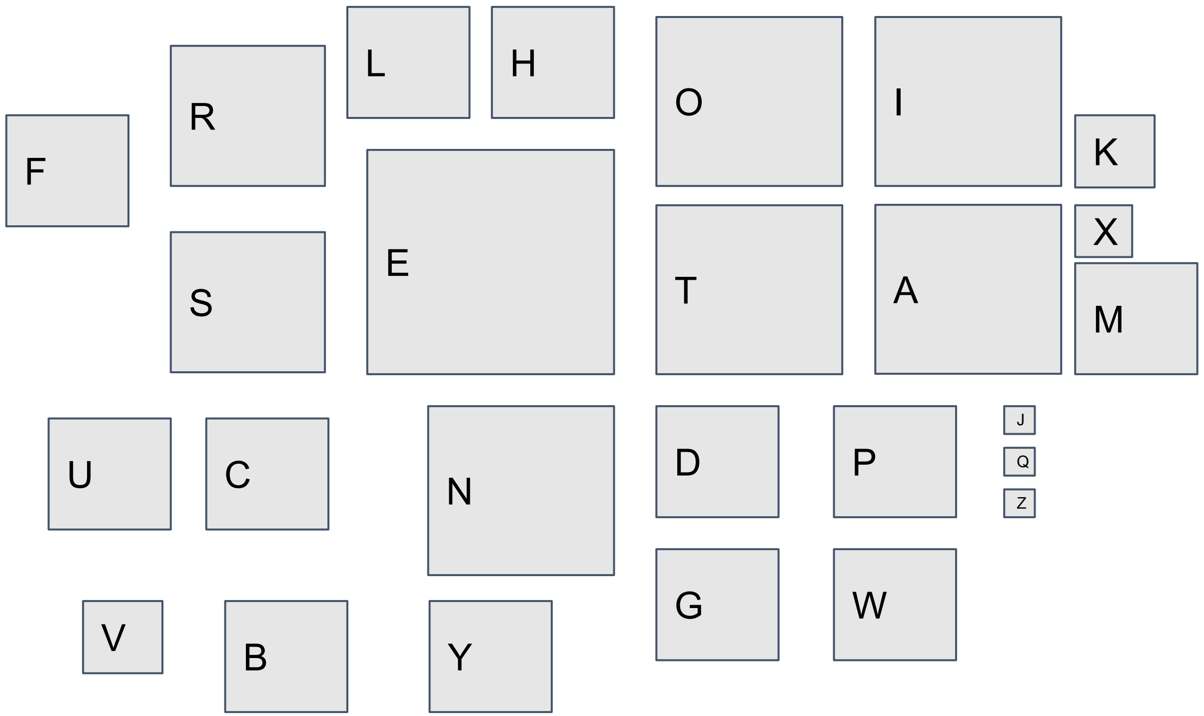


https://norvig.com/mayzner.html

Hacking Fitt's Law: "semantic pointing"

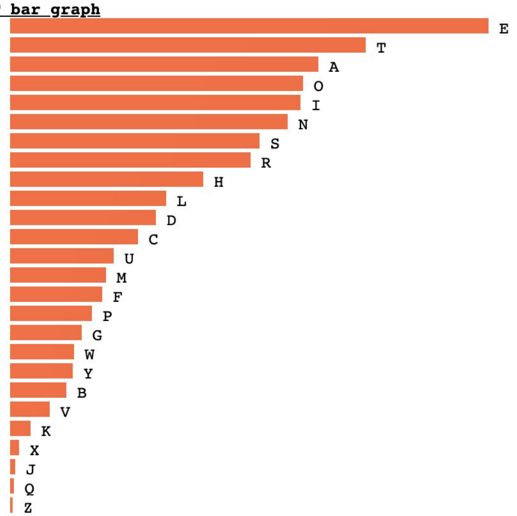
Renaud Blanch, Yves Guiard and Michel Beaudouin-Lafon. **Semantic Pointing: Improving Target Acquisition with Control-Display Ratio Adaptation.** In *Proceedings of CHI 2004*, pages 519-526, Vienna - Austria, April 2004.

Simple application of Fitts Law



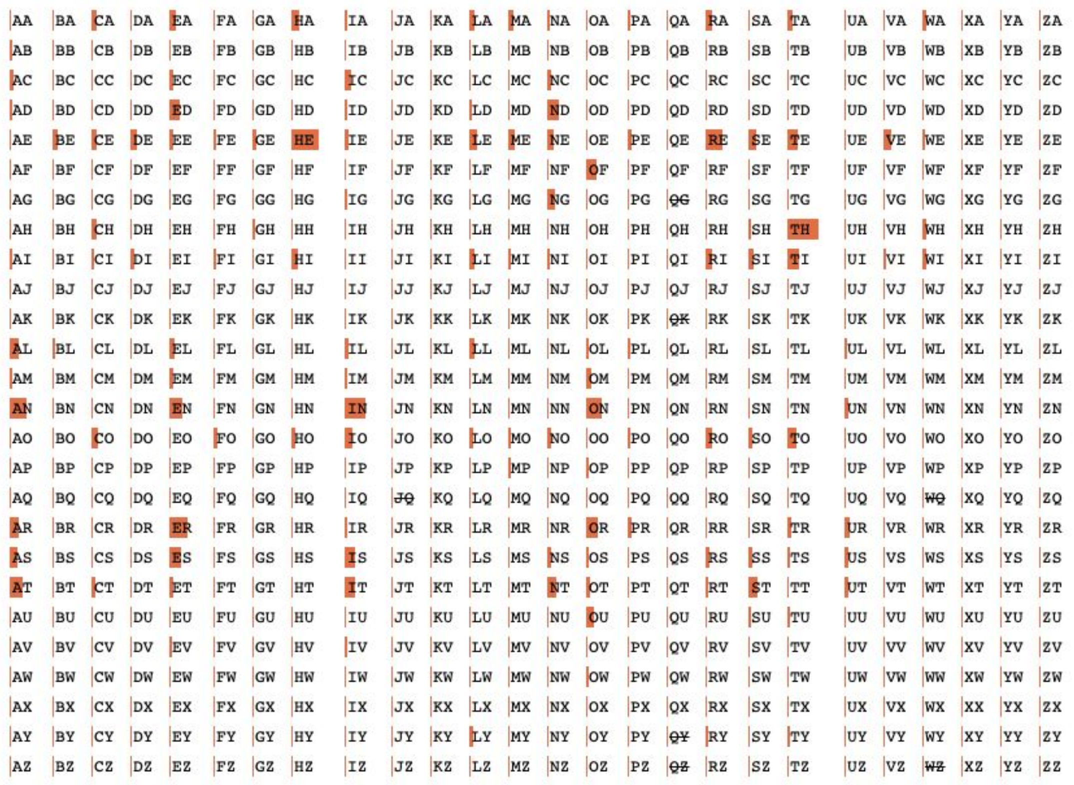
LET	COUNT	PERCENT
E	445.2 B	12.49%
T	330.5 B	9.28%
A	286.5 B	8.04%
O	272.3 B	7.64%
I	269.7 B	7.57%
N	257.8 B	7.23%
S	232.1 B	6.51%
R	223.8 B	6.28%
H	180.1 B	5.05%
L	145.0 B	4.07%
D	136.0 B	3.82%
C	119.2 B	3.34%
U	97.3 B	2.73%
M	89.5 B	2.51%
F	85.6 B	2.40%
P	76.1 B	2.14%
G	66.6 B	1.87%
W	59.7 B	1.68%
Y	59.3 B	1.66%
B	52.9 B	1.48%
V	37.5 B	1.05%
K	19.3 B	0.54%
X	8.4 B	0.23%
J	5.7 B	0.16%
Q	4.3 B	0.12%
Z	3.2 B	0.09%

bar graph



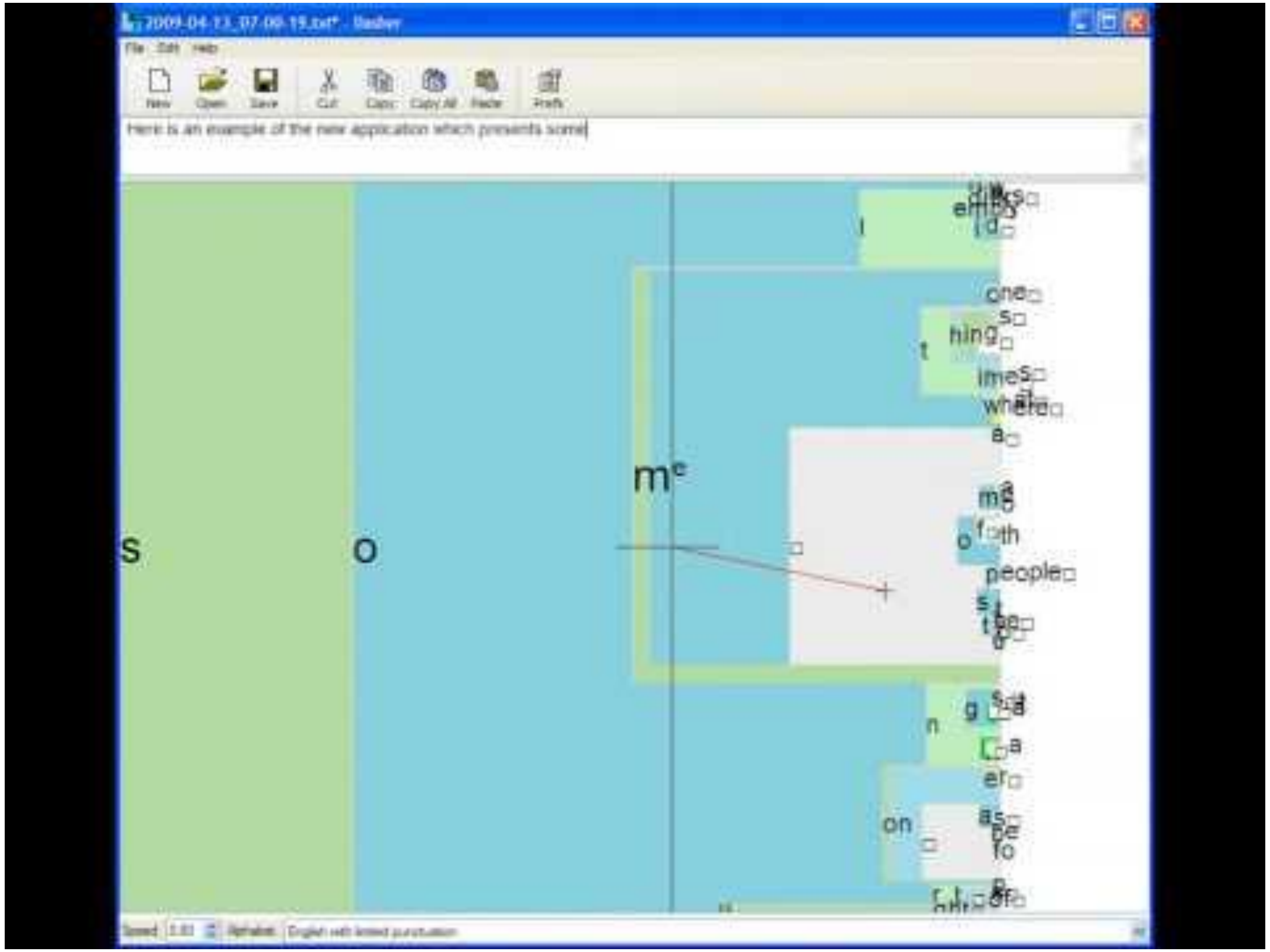
What's wrong with this?

Bigrams



Increasing the depth of the language model allows for a further optimisation, accounting not only for “pointing” target width, but distance of travel ...

Building a system based on relative frequencies



Dasher
(https://www.youtube.com/watch?v=FLalNywdHxU)

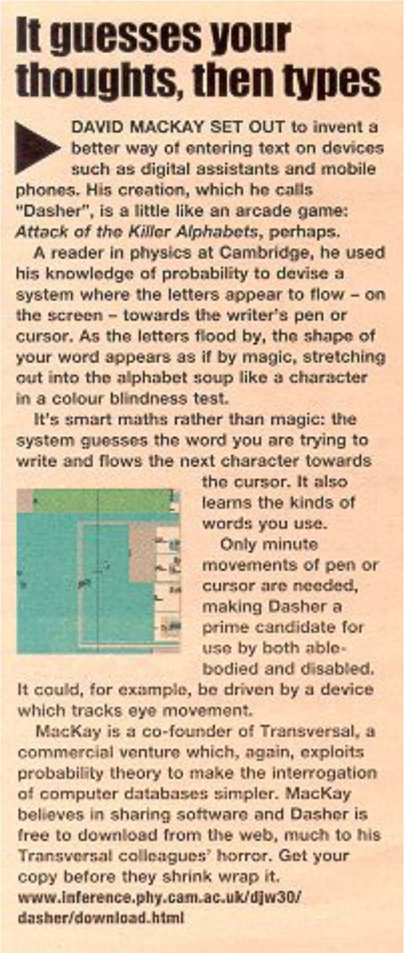
Some lessons from Dasher

- Turning an information theoretic model into a user interface requires a lot of creativity

  => Part II / Part III Human-Centred AI module

- In many cases simple models (nGrams + smoothing) are as - or more - effective than complex ones (neural nets)

- Supporting even famous software, useful for marginalised groups is hard



It guesses your thoughts, then types

DAVID MACKAY SET OUT to invent a better way of entering text on devices such as digital assistants and mobile phones. His creation, which he calls "Dasher", is a little like an arcade game: *Attack of the Killer Alphabets*, perhaps.

A reader in physics at Cambridge, he used his knowledge of probability to devise a system where the letters appear to flow – on the screen – towards the writer's pen or cursor. As the letters flood by, the shape of your word appears as if by magic, stretching out into the alphabet soup like a character in a colour blindness test.

It's smart maths rather than magic: the system guesses the word you are trying to write and flows the next character towards the cursor. It also learns the kinds of words you use.

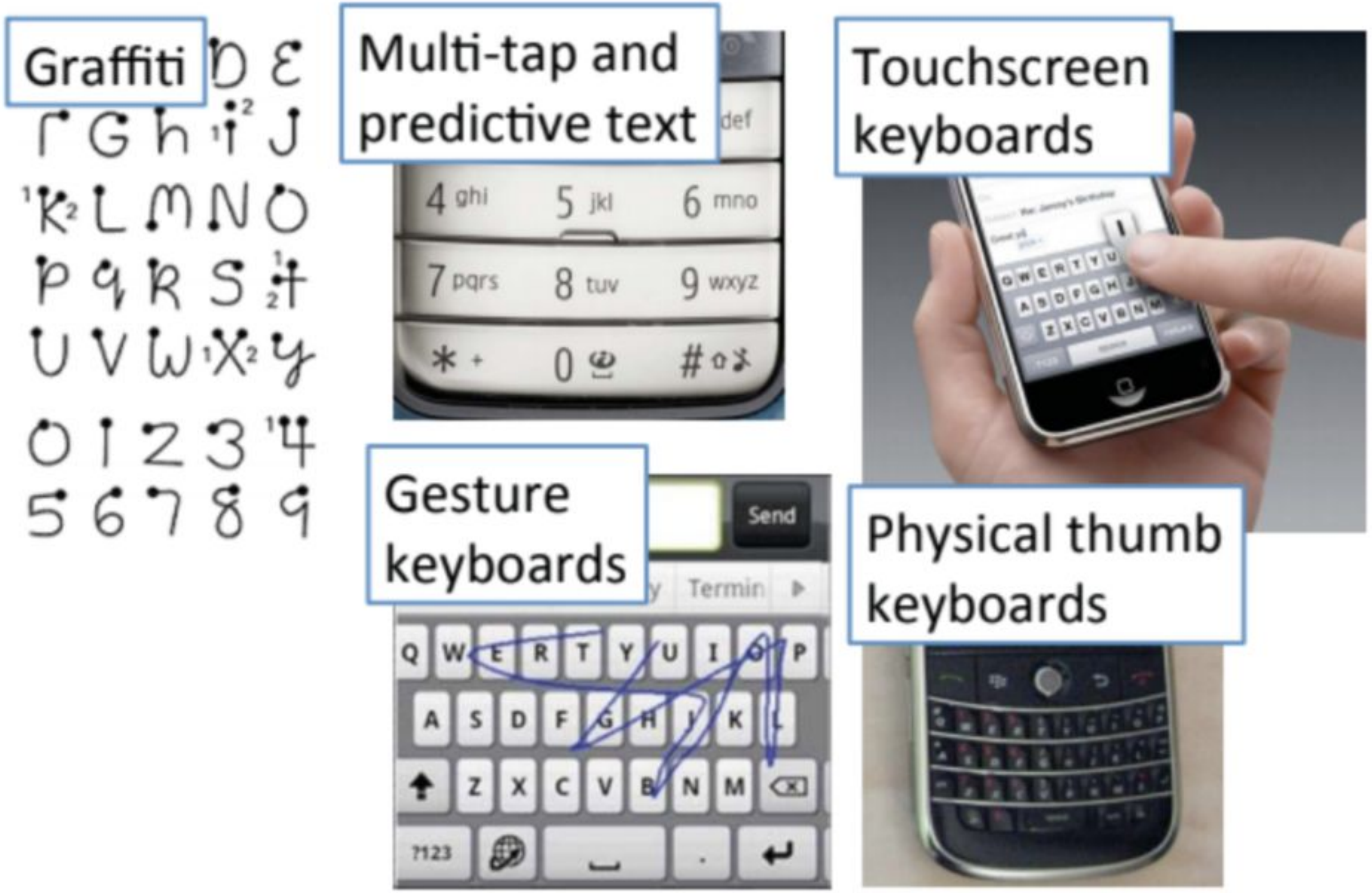
Only minute movements of pen or cursor are needed, making Dasher a prime candidate for use by both able-bodied and disabled. It could, for example, be driven by a device which tracks eye movement.

MacKay is a co-founder of Transversal, a commercial venture which, again, exploits probability theory to make the interrogation of computer databases simpler. MacKay believes in sharing software and Dasher is free to download from the web, much to his Transversal colleagues' horror. Get your copy before they shrink wrap it.
www.inference.phy.cam.ac.uk/djw30/dasher/download.html

(The Financial Times, February 2002)

Deploying smart interfaces

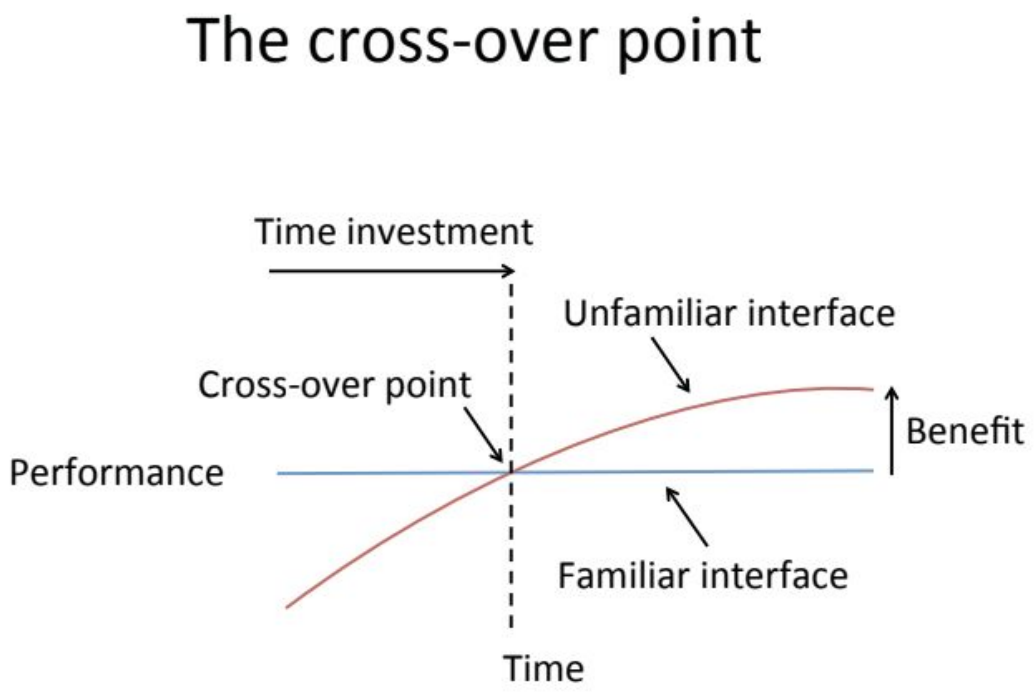
(from Per Ola Kristensson, Cambridge Professor of Interactive Systems Engineering)



Deploying smart interfaces

(from Per Ola Kristensson, Cambridge Professor of Interactive Systems Engineering)

- Entry and error rate
- Learning curve, familiarity and immediate efficacy
- Form factor, presentation, time and comfort
- User engagement
- Visual attention and cognitive resources
- Privacy
- Single vs Multi-character entry
- Specification vs Navigation
- One/Two handed
- Task integration
- Robustness
- Device independence
- Computational demands
- Manufacturing and support cost
- Localisation
- Market acceptance



Deploying smart interfaces

(from Per Ola Kristensson, Cambridge Professor of Interactive Systems Engineering)

- From closed to open-loop
  - Avoid the need for a visual feedback loop
- Continuous novice-to-expert transition
  - Avoid explicit learning
- Path dependency
  - Avoid redesigning the interaction layer
- Flexibility
  - Enable users to compose and edit in a variety of styles without explicit mode switching
- Efficiency
  - Let users' creativity be the bottleneck

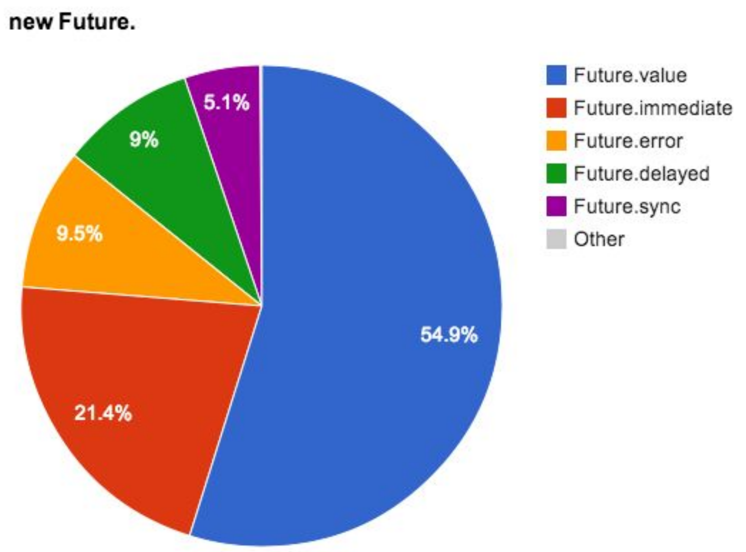
Artificial languages

```
new Future.?
```

```
New Future.aaaaa()
New Future.aaaab()
New Future.aaaac()
New Future.error()
…
New Future.aaaad()
New Future.aaaae()
New Future.value()
```

Artificial languages

`new Future.?`



Liveness becomes Entelechy, Church et al

Ordering code completion suggestions

A simple scheme for predicting code completions:

```
void main() {
    Stopwatch sw = new Stopwatch();
     sw. // <--- What goes here?
}
```

```
elapsed
elapsedMicroseconds
elapsedMilliseconds
elapsedTicks
Frequency
hashCode
isRunning
noSuchMethod
Reset
runtimeType
Start
Stop
toString
```

Ordering code completion suggestions

We calculate:

```
P(completion = “reset”    | context = “void main() { Stopwatch sw = new Stopwatch(); sw.”)
P(completion = “start”    | context = “void main() { Stopwatch sw = new Stopwatch(); sw.”)
```

…

And the usual:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)},$$

Ordering code completion suggestions

$$P(completion = ? \mid context = \text{"..."}) \propto P(context = \text{"..."} \mid completion = ?)\ P(completion = ?)$$

Feature vector

Completion c	Count of seen completions	P(completion)
start	10	0.5
reset	5	0.25
elapsed	5	0.25

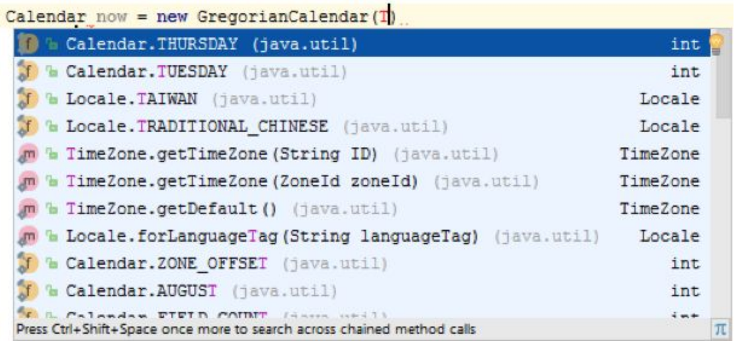
Ordering code completion suggestions

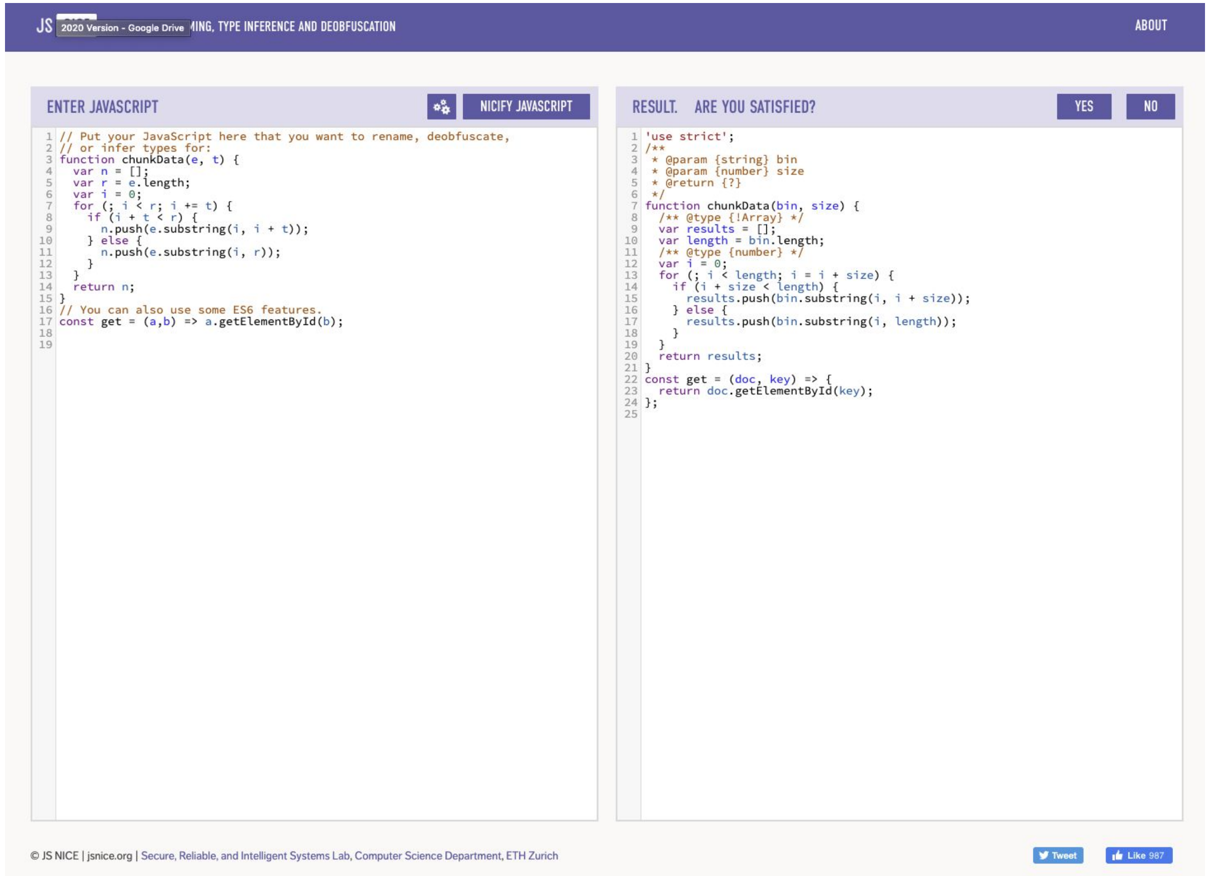
$$P(completion = c \mid context = \text{"..."}) \propto P(context = \text{"..."} \mid completion = c)\ P(completion = c)$$

Completion c	P(completion==c | context) ∝	Order
start	0.9 * 0.5 = 0.45	0
reset	0.4 * 0.25 = 0.1	1
elapsed	0.2 * 0.25 = 0.06	2

Completion c	Feature	Feature value	Count
start	"First-Use"	true	9
		false	1
reset	"First-Use"	true	2
		false	3
elapsed	"First-Use"	true	1
		false	4

Some progress in information efficient IDEs





IntelliJ (Jetbrains)
JSNICE (ETH Zurich)

Where are we going with GPT/CoPilot?

From *Moral Codes* + joint PPIG / Lund AI meeting in 2023:

- Good: allow connection to contextual knowledge
- Poor: doesn’t support visual design notations
- Good: more efficient predictive text, ideally software reuse, cross-PL
- Poor: rife with plagiarism and IP infringement
- Good: accessible to non-programmers and learners
- Poor: hallucinations even worse than in natural language (subtle errors that look right), non-repeatable behaviour

Is CoPilot the next compiler for an even higher-level language? (Remember FORTRAN the Formula Translator).

There is lots of detailed work in our group - e.g. Michael Lee’s project for Part II HCAI module.

Real Co-pilot

Airbus autopilot

https://docs.flybywiresim.com/pilots-corner/a32nx-briefing/flight-deck/glareshield/fcu/

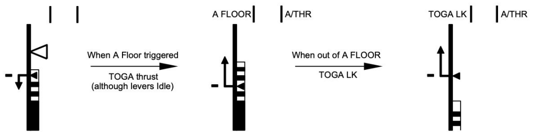
Airbus autopilot

ALPHA FLOOR

When the aircraft's angle-of-attack goes beyond the ALPHA FLOOR threshold, this means that the aircraft has decelerated significantly (below ALPHA PROT speed): A/THR activates automatically and orders TOGA thrust, regardless of the thrust lever position.
The example below illustrates that:

- The aircraft is in descent with the thrust levers manually set to IDLE.
- The aircraft decelerates, during manual flight with the FD off, as indicated on the FMA.

Speed scale and FMA indications in a typical A floor case



When the speed decreases, so that the angle-of-attack reaches the ALPHA FLOOR threshold, A/THR activates and orders TOGA thrust, despite the fact that the thrust levers are at IDLE.
When the aircraft accelerates again, the angle-of-attack drops below the ALPHA FLOOR threshold. TOGA thrust is maintained or locked. This enables the flight crew to reduce thrust, as necessary. TOGA LK appears on the FMA to indicate that TOGA thrust is locked. The desired thrust can only be recovered by setting A/THR to off, with the instinctive disconnect pushbutton.
ALPHA floor is available, when the flight controls are in NORMAL LAW, from liftoff to 100 ft RA at landing. It is inhibited in some cases of engine failure.

Approach Speed

Page 1 of OPS DATA explains how the speed correction is applied. The approach speed increment should be added to the VREF (shown as VLS on the PERF APPR page) for **Flap FULL.** In addition, provided that the resultant VAPP does not exceed VREF +20kt, one third of the headwind component should be added to this figure.

The resultant speed should be inserted, if possible, in the VAPP field on the PERF APPR page and bugged on the standby airspeed indicator. If the situation requires the speed to be Selected, rather than Managed, then the speed calculated above can be set on the FCU.

Insertion of the calculated VAPP on the PERF APPROACH page will ensure that if Managed speed is available, the correct approach speed will be flown. Also the benefits of GS mini will be available, even though the aircraft is landing in an abnormal configuration.

For example, a DUAL ADR FAULT requires a direct law landing flown in Config 3, using a VAPP of VLS **Flap FULL** plus 10kt, plus one third of the head wind component, subject to the 20kt limit described above.

https://www.737ng.co.uk/a320training.pdf

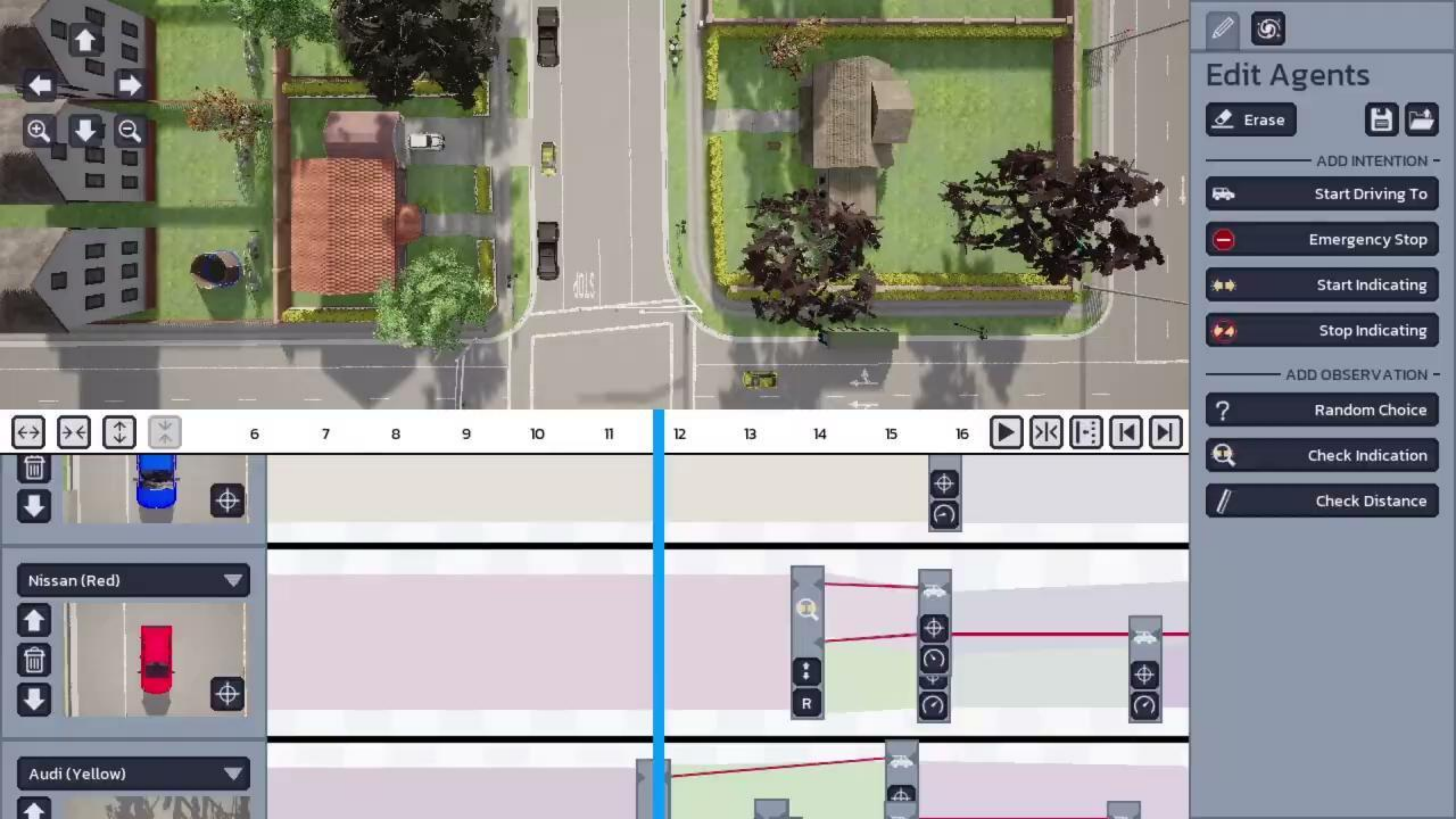
Tesla autopilot

Design considerations

- Situational awareness
- Interaction style for configuring future behaviour
- Degree of knowledge of the behaviour of the system

Some local research for hybrid human/AI futures:
Multiverse Explorer

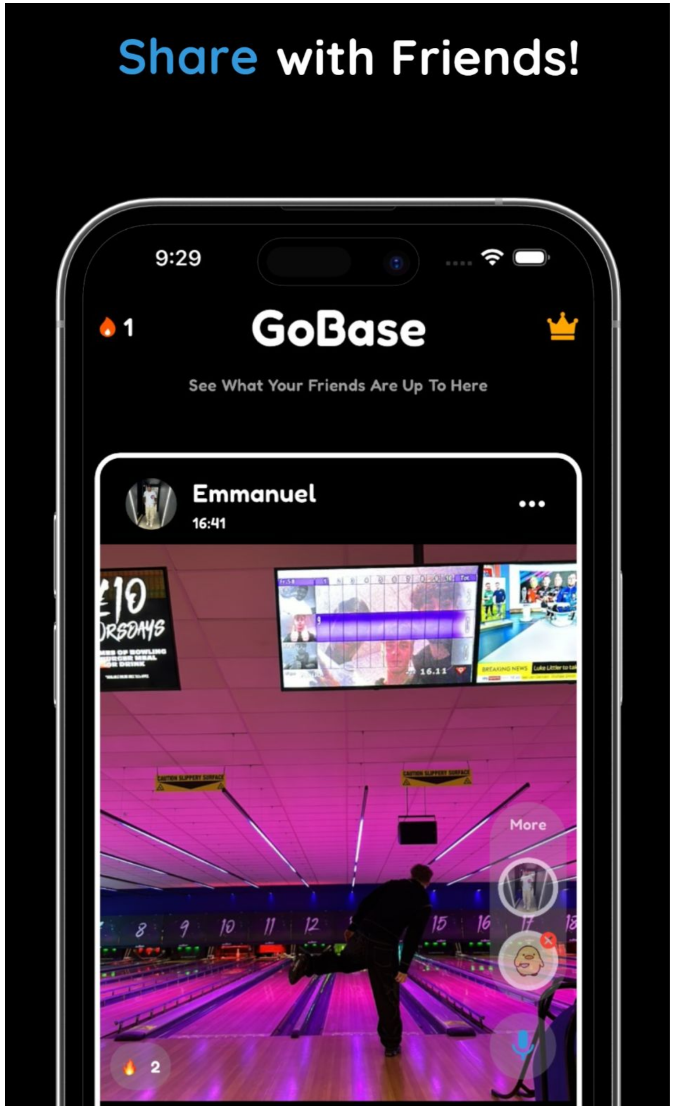
(Tesla Model 3, 2023, Personal image)

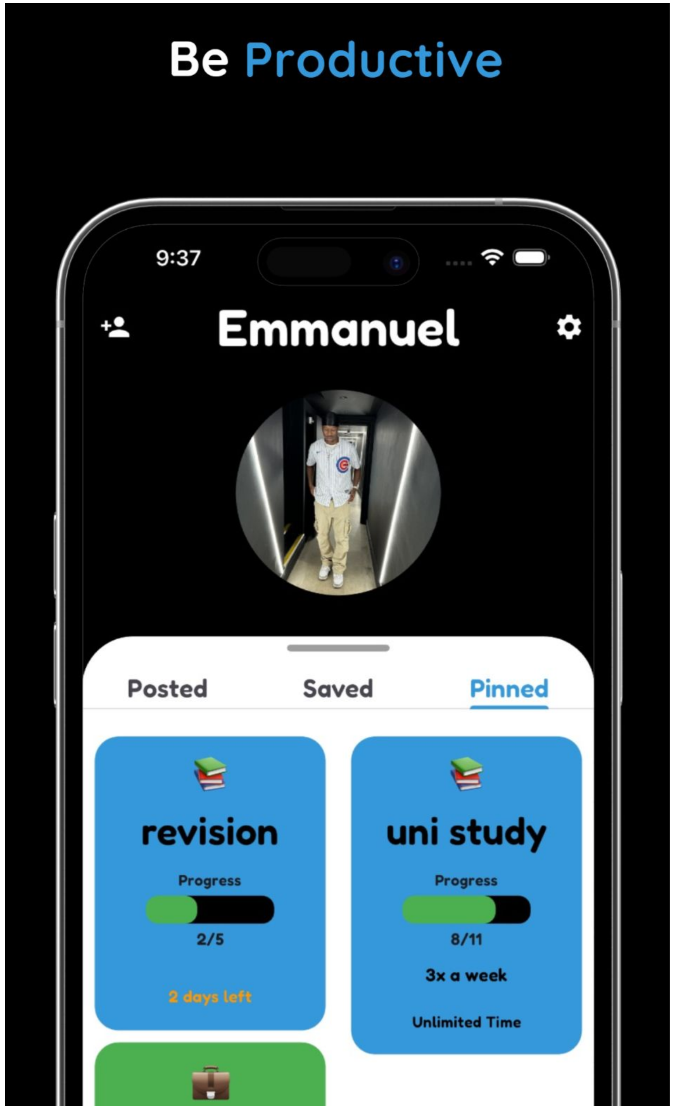


Edit Agents
Erase
ADD INTENTION
Start Driving To
Emergency Stop
Start Indicating
Stop Indicating
ADD OBSERVATION
Random Choice
Check Indication
Check Distance
6 7 8 9 10 11 12 13 14 15 16
Nissan (Red)
Audi (Yellow)
R

The programming analogy challenge 2024:

Example #4: eX-Twitter (we look at Emmanuel Adesola's *GoBase*)





The programming analogy challenge 2024:

Example #4: eX-Twitter (we look at Emmanuel Adesola's *GoBase*)

Is "social" actually a function? In eX-Twitter, any actual work is only a side-effect of networking interactions, meaning bullshit is maximised and message value is reduced to near-zero.

Elon Musk told our PM Rishi Sunak that after these tasks have been automated by AI, nobody will need to work any more.

In Emmanuel's GoBase, the social network delivers proof of work. How does the functional programming paradigm account for side-effects in the world outside the computer, and can the same approach be used to fix the bullshit to value ratio in the non-digital world?