

Open source communities for artists

Lessons learned from the Tracery Project

Dr. Kate Compton
CROSS Incubator Fellow



Tracery: generate text, graphics and more

Tracery is a super-simple tool and language to generate text, by [GalaxyKate](#). It's been used by middle school students, humanities professors, indie game developers, professional bot makers, and lots of others.

In open source, we feel strongly that to really do something well, you have to get a lot of people involved. -Linus Torvalds

Part I

About “contributors”

Who is an open source contributor?

- People who write the code
- People who write the documentation
- People who port the tool to other languages
- People who make connecting tools, pipelines, platforms
- People who write tutorials or teach workshops
- People who fundraise, organize or provide long-term stability
- People who make art with the tool

Who is an open source contributor?

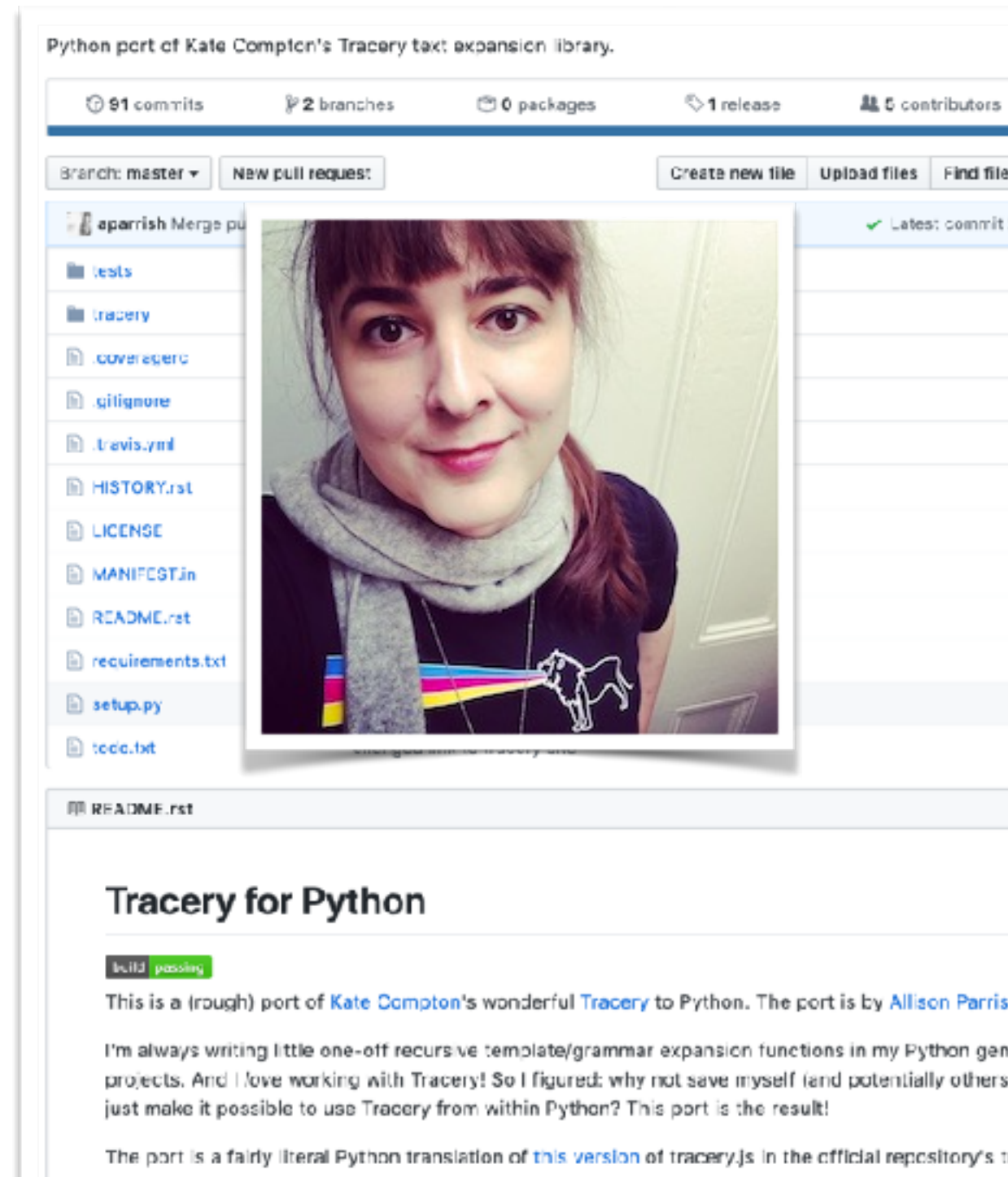
- People who write the code
- People who write the documentation



Who is an open source contributor?

- People who port the tool to other languages

Allison Parrish



Who is an open source contributor?

- People who make connecting tools, pipelines, platforms

Cheap Bots, Done Quick!

This site will help you make a Twitterbot! They're easy to make and free to run.

To use it, [create a Twitter account](#) for your bot to run under and then sign in below. The bots are written in [Tracery](#), a tool for writing generative grammars developed by [Kate Compton](#). This site is run by [George Buckenham](#) - they can be contacted at vtwentyone@gmail.com. You can support this site on [Patreon](#).

 Sign in w



@v21 George Buckenham

Who is an open source contributor?



Shawn Graham (@electricarchaeo)



- People who write tutorials or teach workshops



Aaron Reed (@aaronareed)



Who is an open source contributor?

- People who fundraise, organize or provide long-term stability



CROSS
cross.ucsc.edu

The logo for CROSS (Center for Research in Open Source Software) features the word "CROSS" in a bold, green, sans-serif font. The letter "O" is stylized with a white grid pattern. Below the word, the website address "cross.ucsc.edu" is written in a smaller, dark blue font.

botwiki

The image shows the homepage of Botwiki, a website for bot enthusiasts. The main heading "botwiki" is in a large, bold, dark grey font. Below it, there are four main sections: "create", "learn", "about", and "explore". Each section has a brief description. At the bottom, there is a search bar and a row of social media icons (Facebook, Twitter, YouTube, and a generic icon) followed by the handle "@botwikidotorg".

explore

What are bots, and where to find them.

create

Learn how to make fun, friendly online bots.

learn

Read our collection of articles and essays.

about

What is Botwiki, and who made it.

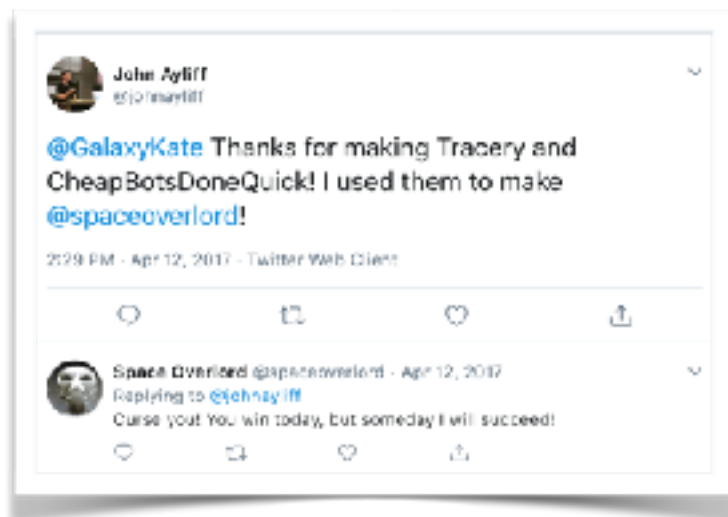
Search...



Who is an open source contributor?



- People who make art with the tool



Who is an open source contributor?

- People who write the code
- People who write the documentation

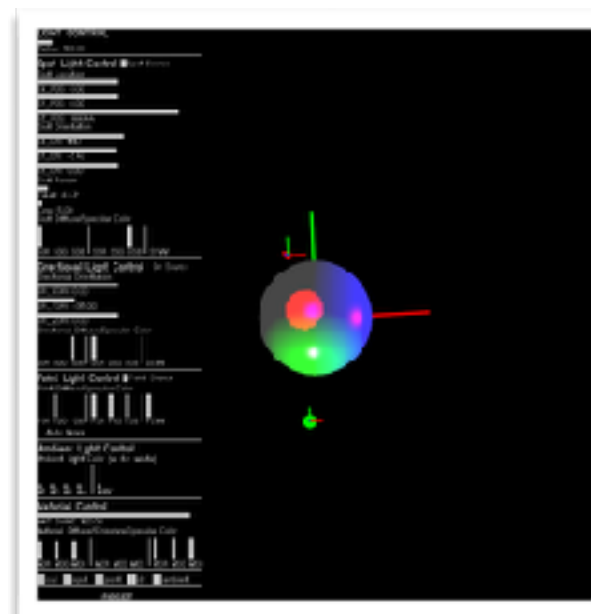
Not on the GitHub repo!

- People who port the tool to other languages
- People who make connecting tools, pipelines, platforms
- People who write tutorials or teach workshops
- People who fundraise, organize or provide long-term stability
- People who make art with the tool

Convening for open source tools for the arts

(June 2018, Minneapolis)

- processing.org
- three.js
- wekinator
- openframeworks
- tracery

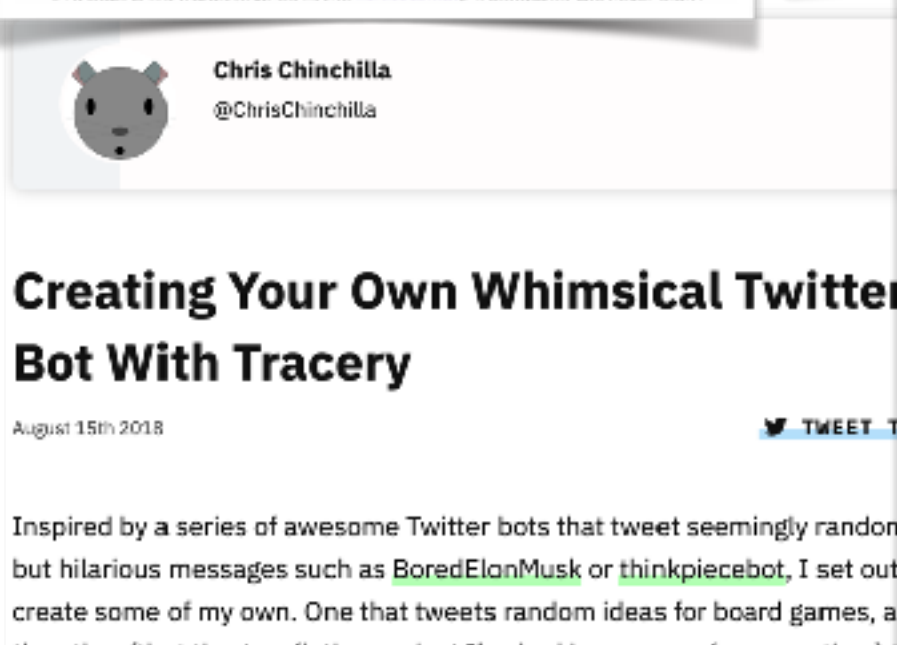
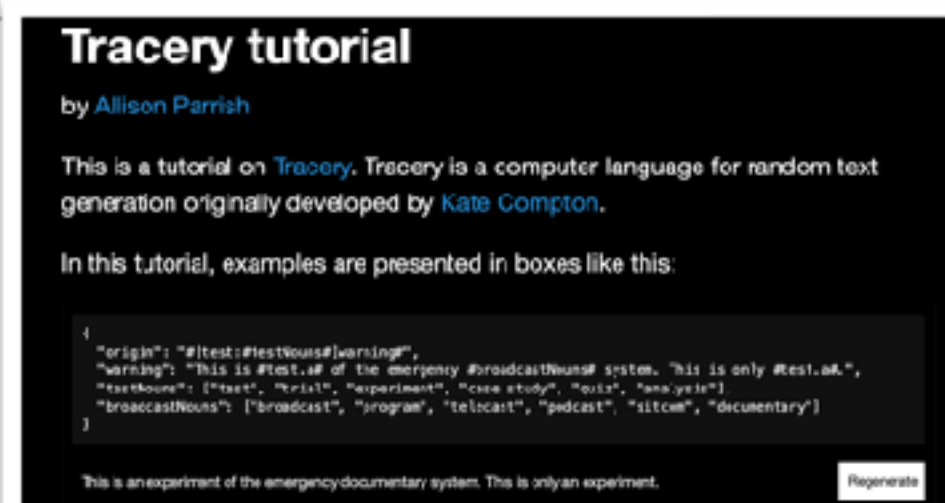
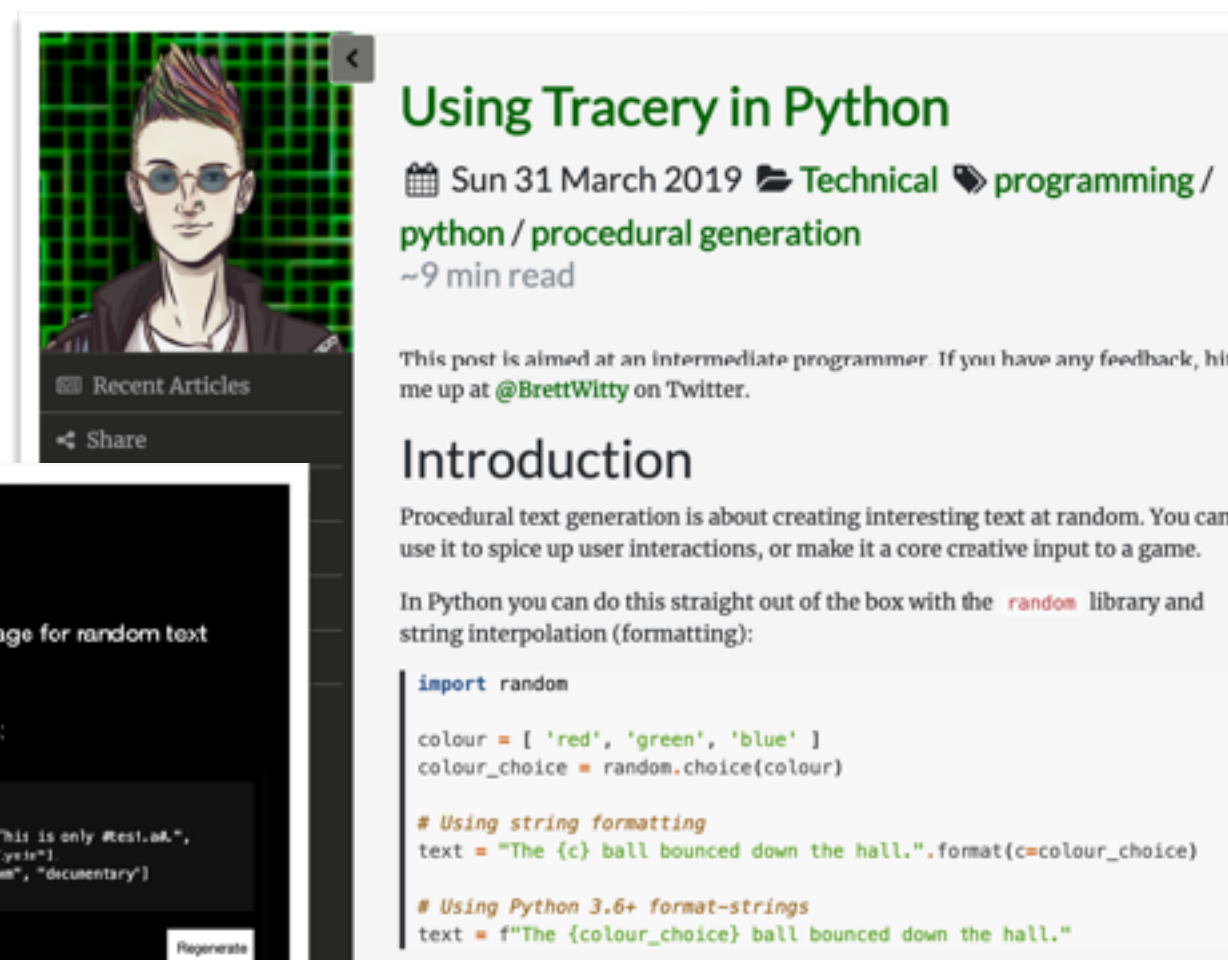


Kinds of outreach

- How can you reach people you don't have access to?
 - Different departments
 - Different fields
 - Different language
 - Different needs
 - Age/interest/community

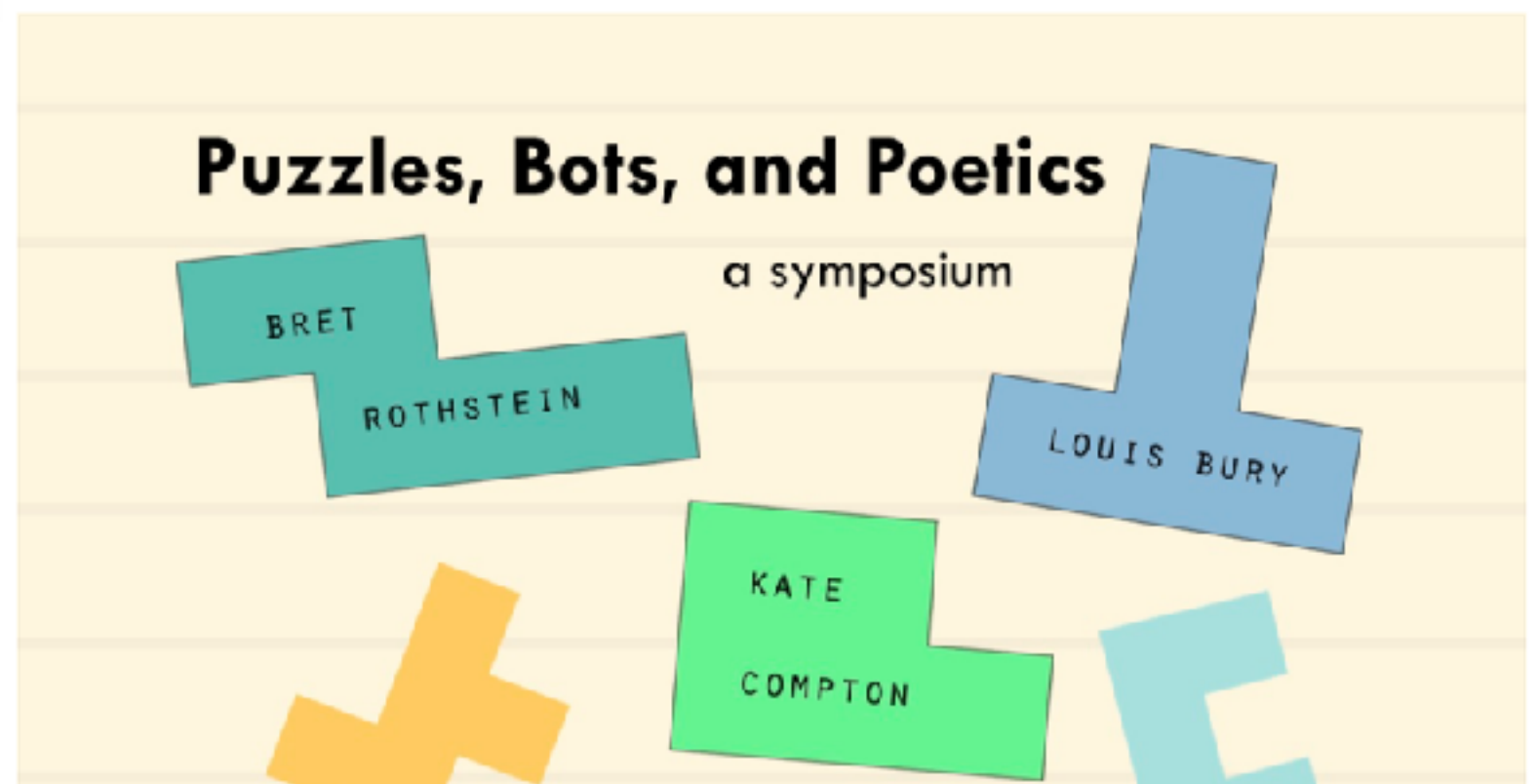


Kinds of outreach



Kinds of outreach

- Go where people are
- Respect their time and interest



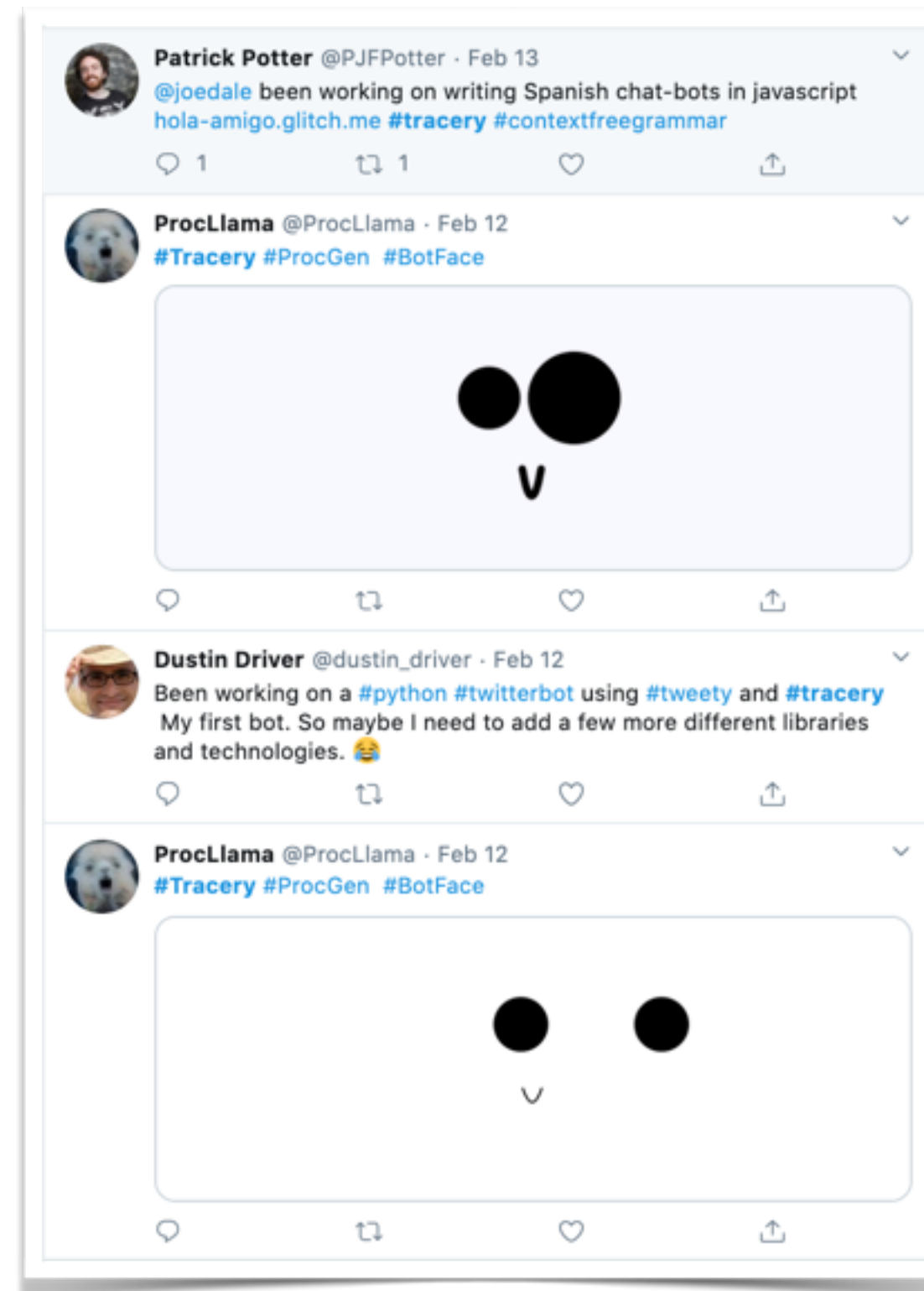
Kinds of outreach

- Evangelist
- Tinker



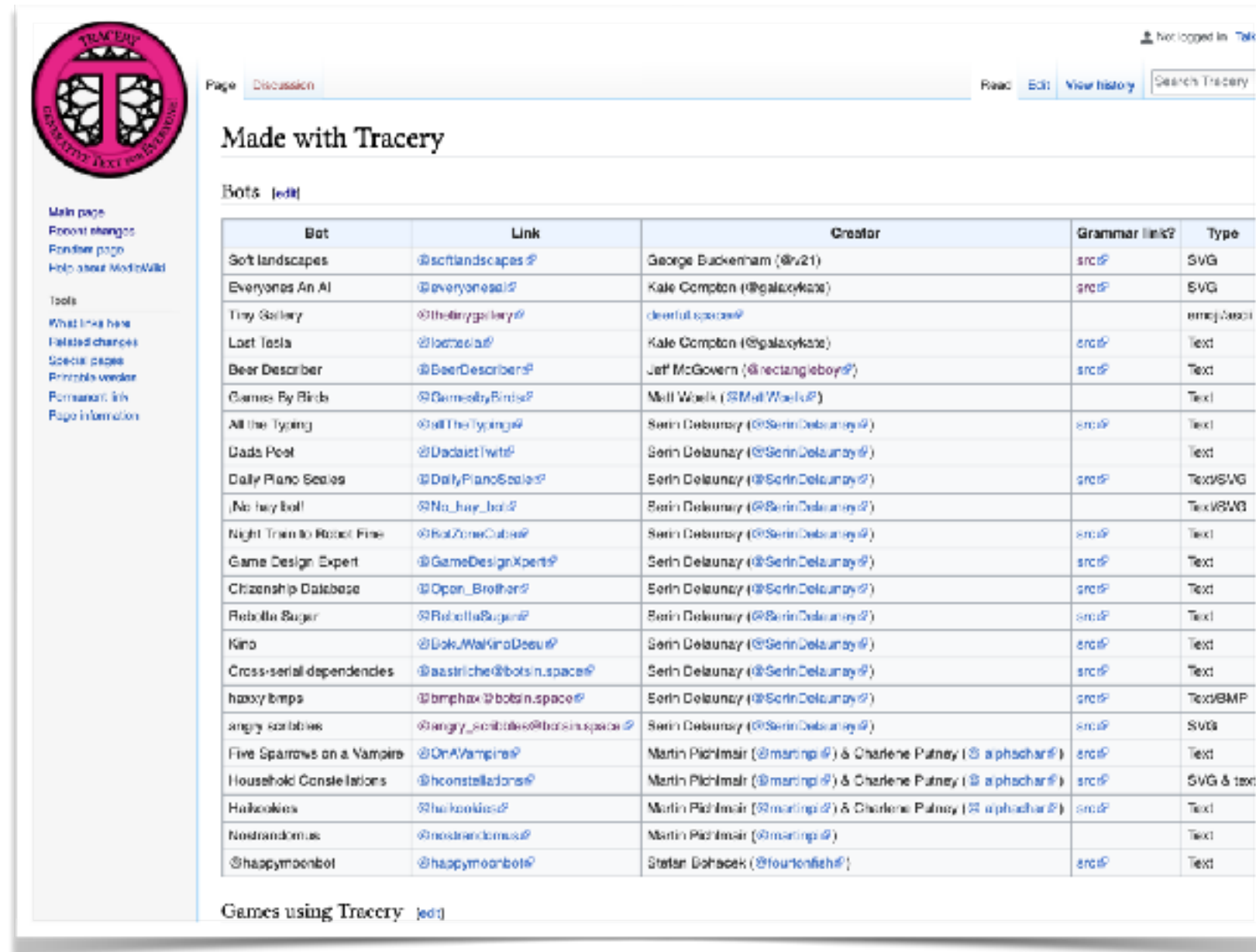
Useful things to have

- A place where people can talk
- A way for people to talk to you
 - A way for *non-technical* people to contact you
- A known hashtag



Useful things to have

- A site where people can put up art
- Or share source code (and link to remixes)
- Or annotate and curate lists



The screenshot shows the Tracery website interface. On the left is a sidebar with navigation links: Main page, Recent changes, Random page, Help about MediaWiki, Tools, What links here, Related changes, Special pages, Printable version, Permanent link, and Page information. The main content area is titled "Made with Tracery" and features a "Bots" section with a table listing various bots. The table has columns for Bot, Link, Creator, Grammar link?, and Type. Below the bots section is a "Games using Tracery" section.

Bot	Link	Creator	Grammar link?	Type
Soft landscapes	@softlandscapes	George Buckenham (@rv21)	src	SVG
Everyones An AI	@everyonesai	Kale Compton (@galaxykale)	src	SVG
Tray Gallery	@thetraygallery	deanfullspace		img/ascii
Lost Tesla	@lostmelod	Kale Compton (@galaxykale)	src	Text
Beer Describer	@BeerDescriber	Jeff McGovern (@rectangleboy)	src	Text
Games By Birds	@GamesByBirds	Matt Waelk (@MailWaelk)		Text
All the Typing	@allthetyping	Serin Delaunay (@SerinDelaunay)	src	Text
Dada Poet	@DadaistTwit	Serin Delaunay (@SerinDelaunay)		Text
Daily Piano Seals	@DailyPianoSeals	Serin Delaunay (@SerinDelaunay)	src	Text/BMP
No hay ball	@No_hay_ball	Serin Delaunay (@SerinDelaunay)		Text/BMP
Night Train to Robot Fine	@BotZoneCubed	Serin Delaunay (@SerinDelaunay)	src	Text
Game Design Expert	@GameDesignXperts	Serin Delaunay (@SerinDelaunay)	src	Text
Citizenship Database	@Open_Brother	Serin Delaunay (@SerinDelaunay)	src	Text
Rebelle Super	@RebelleSuper	Serin Delaunay (@SerinDelaunay)	src	Text
Kino	@DokuWaKinoDeu	Serin Delaunay (@SerinDelaunay)	src	Text
Cross-serial dependencies	@aastriche@botsin.space	Serin Delaunay (@SerinDelaunay)	src	Text
hazy bmps	@bimphax@botsin.space	Serin Delaunay (@SerinDelaunay)	src	Text/BMP
angry scribbles	@angry_scribbles@botsin.space	Serin Delaunay (@SerinDelaunay)	src	SVG
Five Sparrows on a Vampire	@OnAVampire	Martin Pichlmair (@martinpi) & Charlene Patney (@aphaschar)	src	Text
Household Constellations	@hconstellations	Martin Pichlmair (@martinpi) & Charlene Patney (@aphaschar)	src	SVG & text
Haikookies	@haikookies	Martin Pichlmair (@martinpi) & Charlene Patney (@aphaschar)	src	Text
Nostrandomus	@nostrandomus	Martin Pichlmair (@martinpi)		Text
@haapmoonbot	@haapmoonbot	Stefan Bohacek (@fourtenfish)	src	Text

Games using Tracery [\[edit\]](#)

Useful things to have

- A site where people can put up art
- Or share source code (and link to remixes)
- Or annotate and curate lists



Useful things to have

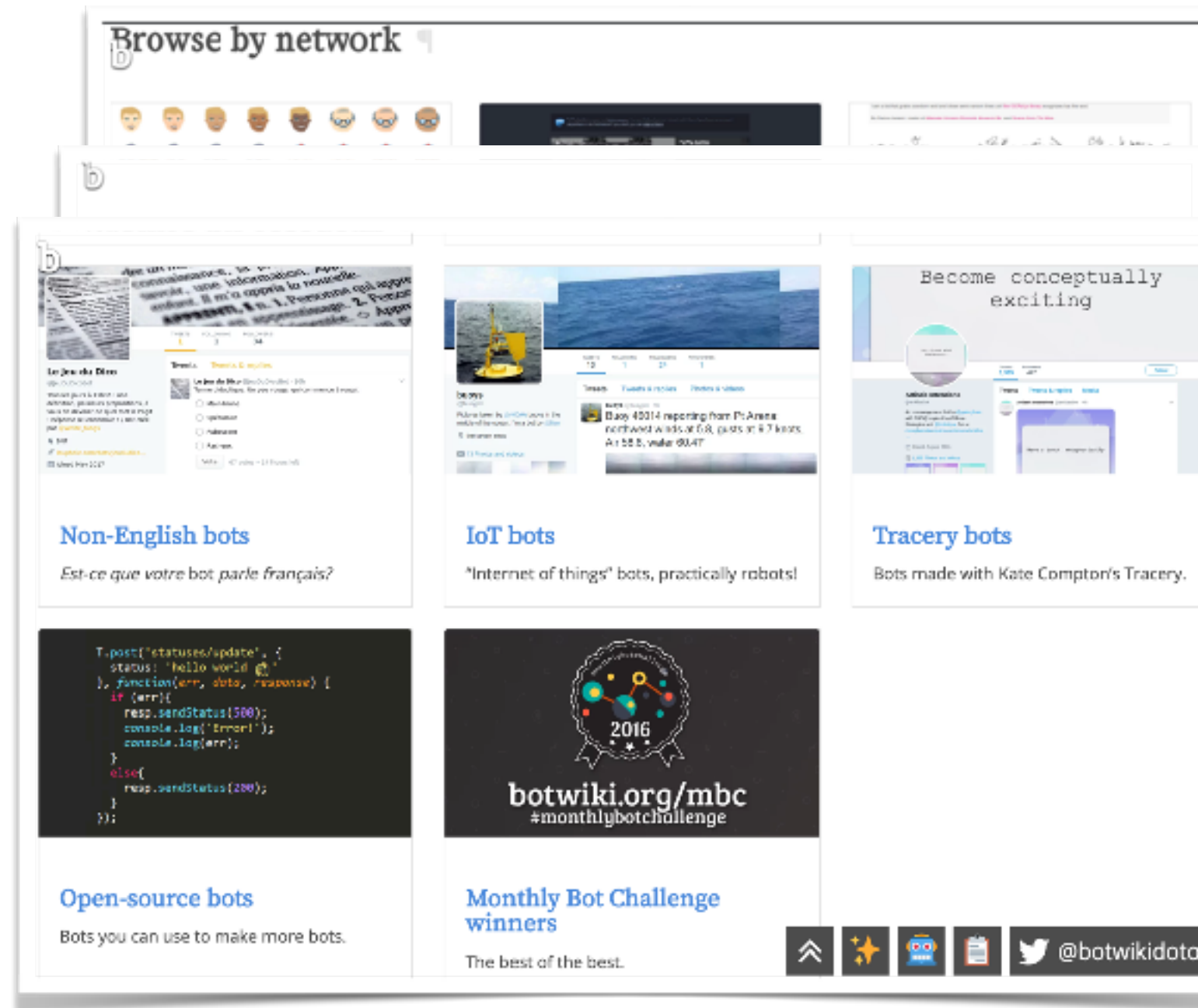
- A site where people can put up art
- Or share source code (and link to remixes)
- Or annotate and curate lists



<https://cheapbotsdonequick.com/source/katesbot>

Useful things to have

- A site where people can put up art
- Or share source code (and link to remixes)
- Or annotate and curate lists



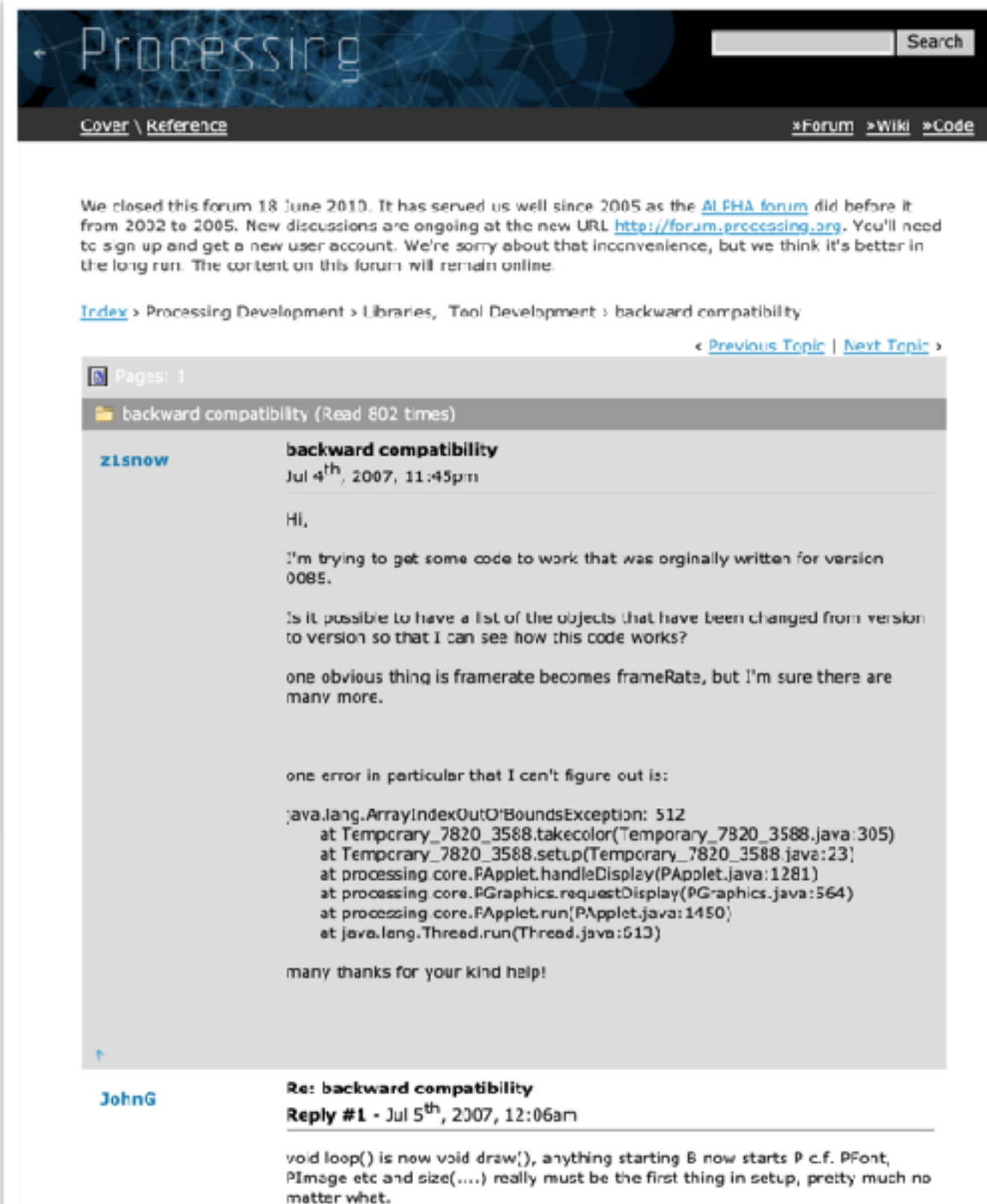
Why do people need a place to gather?

- A place to look for answers
 - (“has this been ported to LISP?”, “is there any French documentation?”, “how do I ...”)
- A place to invest in things
 - *you* care about
 - (kinds of use cases, types of users, languages and nationalities, fandoms, fields of study)
- “Off-label” uses
- Missing community features



What do we owe artists?

- Backwards compatibility
- Documenting broken compatibility
- Not breaking links
- Human-readable formats
- “Forever” museum-quality versions



The screenshot shows a forum post on the Processing website. The header includes the Processing logo, a search bar, and navigation links for Cover, Reference, Forum, Wiki, and Code. The main content area contains a notice about the forum's closure and a breadcrumb trail leading to the current thread. The thread is titled 'backward compatibility' and was posted by user 'zlsnow' on July 4th, 2007. The post discusses the user's difficulty in running code originally written for version 0085 and asks for a list of changes between versions. A stack trace for a 'java.lang.ArrayIndexOutOfBoundsException' is provided. The post concludes with a thank you. A reply from user 'JohnG' is partially visible at the bottom, dated July 5th, 2007, and mentions changes to 'void loop()' and 'void draw()'.

Processing

Cover \ Reference

Forum Wiki Code

We closed this forum 18 June 2010. It has served us well since 2005 as the [ALPHA forum](#) did before it from 2002 to 2005. New discussions are ongoing at the new URL <http://forum.processing.org>. You'll need to sign up and get a new user account. We're sorry about that inconvenience, but we think it's better in the long run. The content on this forum will remain online.

Index > Processing Development > Libraries, Tool Development > backward compatibility

< Previous Topic | Next Topic >

Pages: 1

backward compatibility (Read 802 times)

zlsnow

backward compatibility
Jul 4th, 2007, 11:45pm

Hi,

I'm trying to get some code to work that was originally written for version 0085.

Is it possible to have a list of the objects that have been changed from version to version so that I can see how this code works?

one obvious thing is framerate becomes frameRate, but I'm sure there are many more.

one error in particular that I can't figure out is:

```
java.lang.ArrayIndexOutOfBoundsException: 512
    at Temporary_7820_3588.takecolor(Temporary_7820_3588.java:305)
    at Temporary_7820_3588.setup(Temporary_7820_3588.java:23)
    at processing.core.PApplet.handleDisplay(PApplet.java:1281)
    at processing.core.PGraphics.requestDisplay(PGraphics.java:564)
    at processing.core.PApplet.run(PApplet.java:1450)
    at java.lang.Thread.run(Thread.java:613)
```

many thanks for your kind help!

JohnG

Re: backward compatibility
Reply #1 - Jul 5th, 2007, 12:06am

void loop() is now void draw(), anything starting B now starts P c.f. PFont, PImage etc and size(...) really must be the first thing in setup, pretty much no matter what.

What do we owe artists?

- Backwards compatibility
 - Documenting broken compatibility
- Not breaking links
- Human-readable formats
- “Forever” museum-quality versions

EMAIL

Email-based works rely on email as a tool of communication or provide methods for subverting or transforming the purpose and experience of email.

SCAREMAIL - BEN GROSSER

Keywords: code, javascript, activist, email

TATUAJE - RODOLFO JM, LEONARDO ARANDA, GABRIELA GORDILLO, ET AL.

Keywords: hypertext, email

FANFICTION

Works of fanfiction, or transformative works, take inspiration from other fictional universes to build new stories and commentary. The tradition of fanfiction has long existed in digital fiction communities.

FRED AND GEORGE - FLOURISH KLINK

Keywords: remix, generative, fanfiction, poetry, javascript

TAKEI, GEORGE - MARK SAMPLE

Keywords: remix, generative, fanfiction, poetry, javascript

FLASH

Adobe's Flash platform offers tools for combining scripted behaviors with animation through a browser plug-in. Flash dominated the interactive web, but its use has faded thanks in large part to the rise of mobile platforms.

@GUA_UM CONTO DIGITAL - ANTONIO ABERNÚ

Keywords: activist, flash

ANACRÓN: HIPÓTESIS DE UN PRODUCTO TODO - AUGUSTO MARQUET AND GABRIEL WOLFSON

Keywords: poetry, hypertext, flash

ARS POETICA - ZENON FAJFER

Keywords: poetry, flash

ASK ME FOR THE MOON - JOHN DAVID ZUERN

Keywords: flash

What do we owe artists?

- A file is a file.....right?
- Can you download a copy from the hosting service? (not in Mario Maker)
- Can you edit the file in someone else's software? (not Photoshop files)
- Can you, a human, read these files? Could you write your own editor? (not for MS Word docs)
- Can someone else build code to **run** these files?
- Would you feel safe running strangers' files on your system?



Part I

Language features

How does adoption happen

- You make a good tool
- Everyone likes it



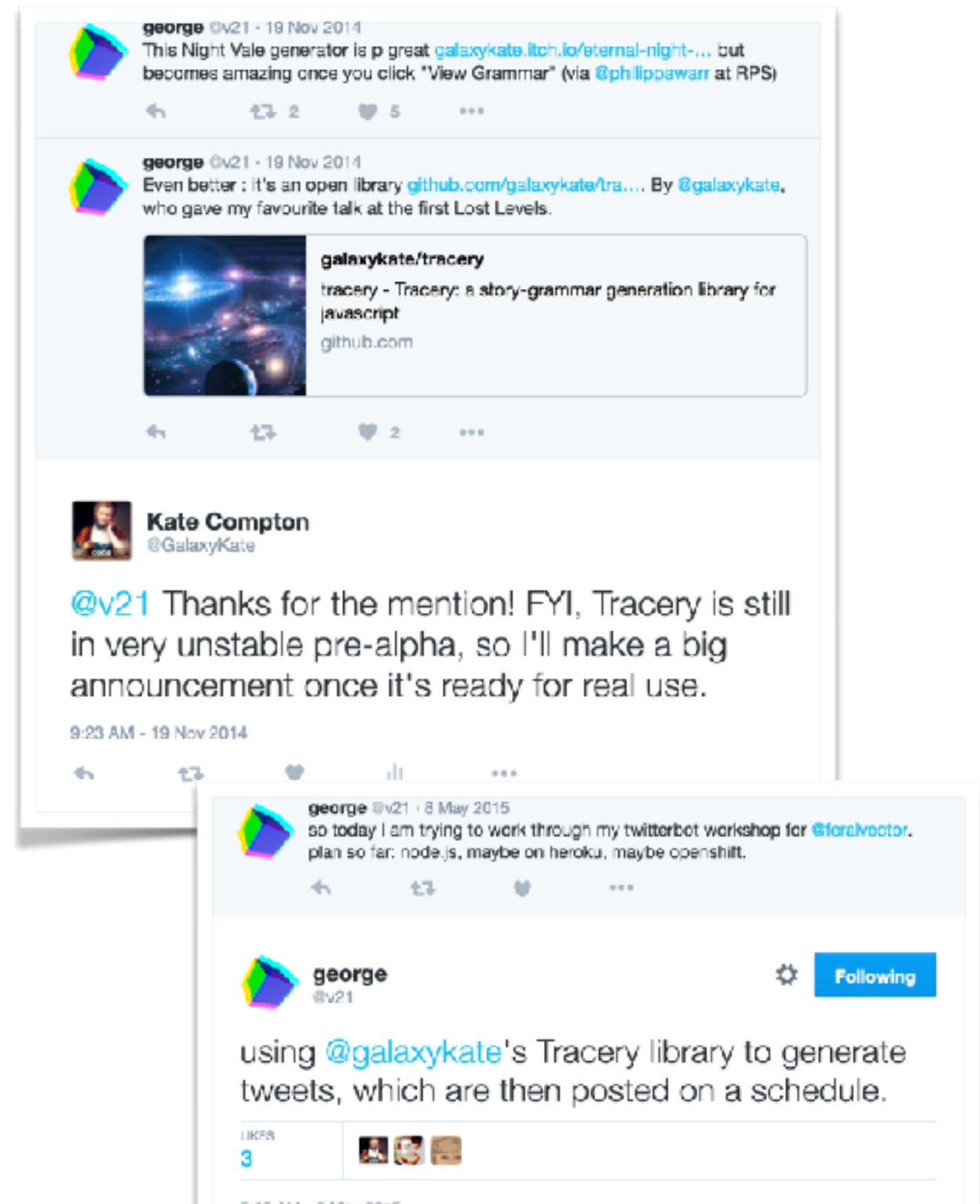
How does adoption happen

- You make a good tool
- Someone is looking for a tool like that at that time
- They have enough time to try it out
- Nothing breaks immediately



How does adoption happen

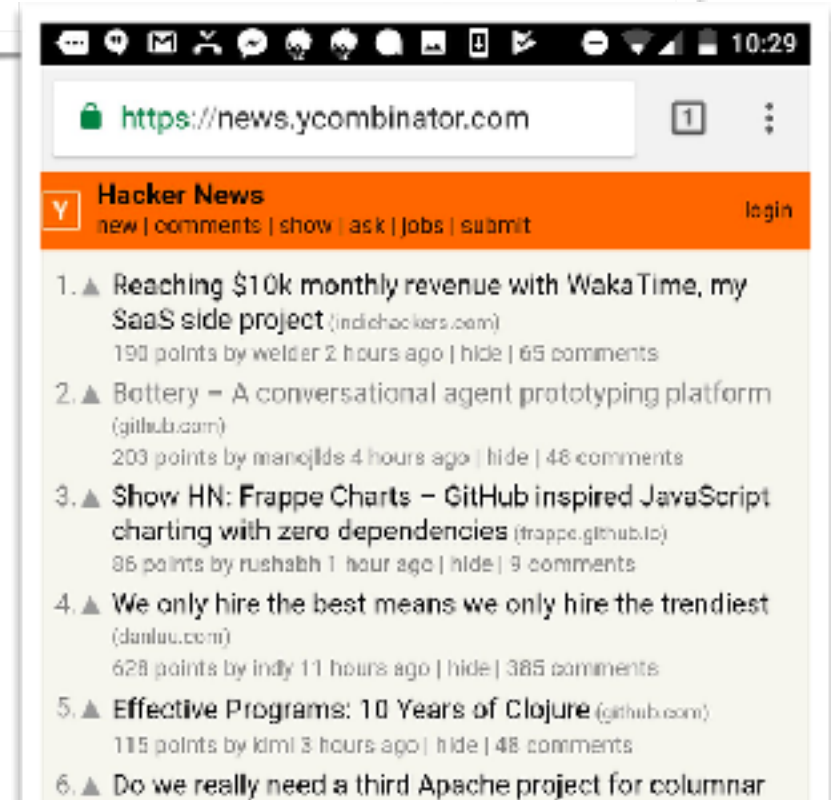
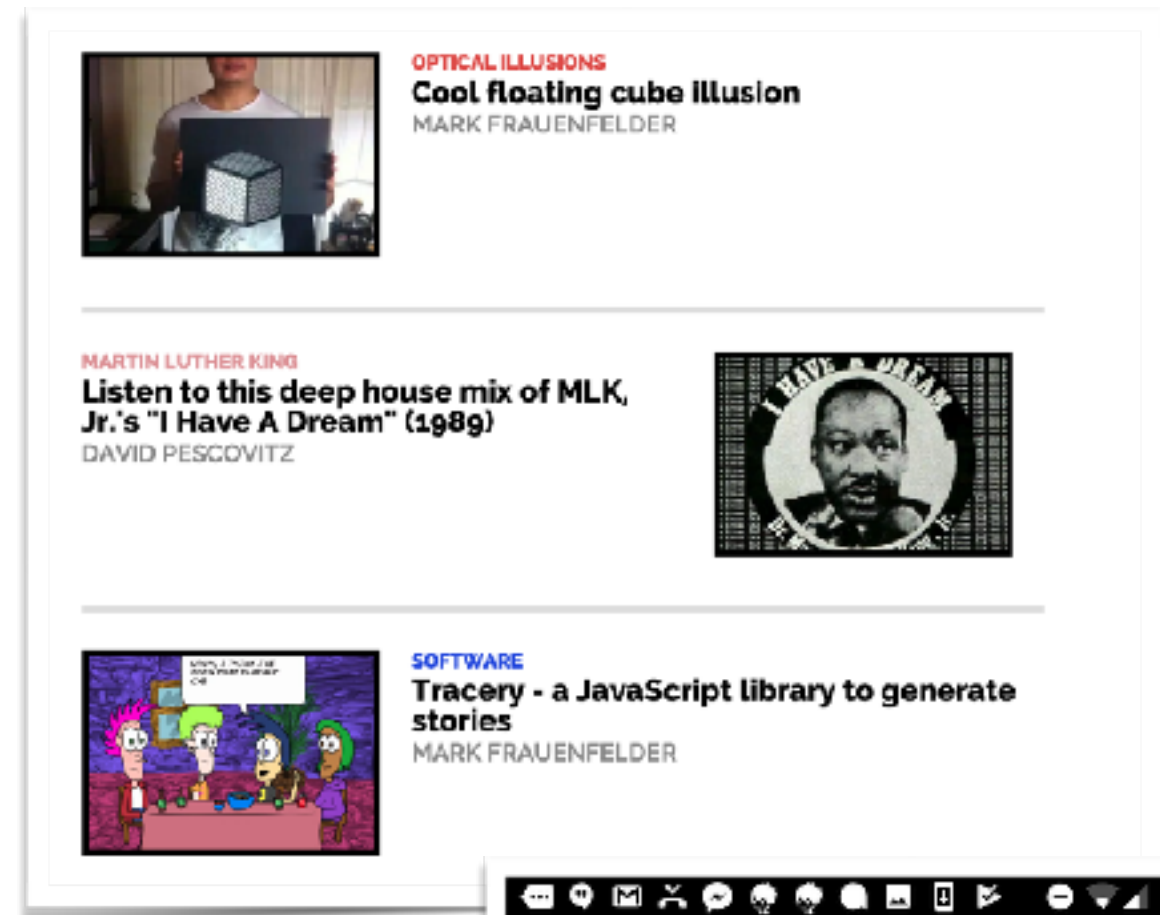
- You make a good tool
- Someone is looking for a tool like that at that time
- They have enough time to try it out
- Nothing breaks immediately



How does adoption happen

Lessons:

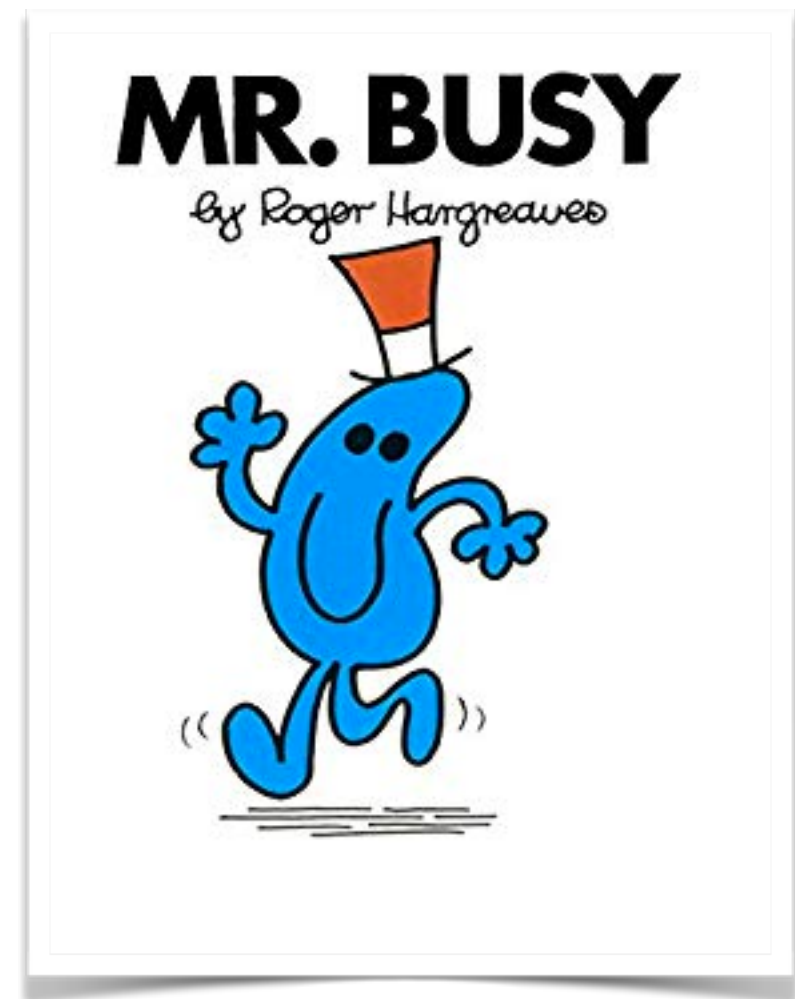
- You aren't ready when it happens
- “not done yet” won't stop it from happening
- but “email me for the current version” will



Everyone is too busy

What is the absolute minimum time needed for someone to prove to themselves that your product works?

- journalist
- CTO / tech worker
- high-ranking professor
- new student



Everyone is too busy

What is the absolute minimum time needed for someone to prove to themselves that your product works?

- journalist
- CTO / tech worker
- high-ranking professor
- new student

The screenshot shows the 'Crystal Code Palace' Tracery tutorial interface. It features a pink-themed layout with a sidebar on the left containing a 'Crystal Code Palace' logo and navigation links like 'Tutorials', '@galaxykate', and 'more from galaxykate'. The main content area is divided into three sections, each with a title, explanatory text, a JSON grammar snippet, and a 'reroll' button.

1: Generating text!
Tracery uses grammars to generate text. A replacement grammar takes a starting symbol, and replaces it with one of several rules. On the right (or below, depending on your screen size), you can see the json object that represents this very simple grammar. Farther right is the current start symbol ('animal') and a list of possible texts that it can generate. Press the reroll button a few times to generate more. Not very interesting, yet, so lets get more complicated!

```
{
  "animal": ["unicorn", "raven", "sparrow", "scorpion", "coyote", "eagle", "zebra"]
}
```

start symbol: animal
raven
scorpion
lizard
sparrow
zebra
eagle
zebra

2: Rules within rules
Rules can call rules! Take a look at the rule for 'sentence': 'The #color# #animal# of the #natureNoun# is called #name#'. Each of the words between two hashtags is a symbol to be replaced. Try adding your name to the list of names, surrounded by quotation marks like all the other options. It will now appear in some of the generated text! Notes: JSON is very very fussy. There must be a comma between every option, but none after the last option.

```
"sentence": ["The #color# #animal# of the #natureNoun# is called #name#", "The #color# #animal# of the #natureNoun# is called #name#"]
"color": ["orange", "blue", "white", "black", "grey", "purple", "indigo", "turquoise"]
"animal": ["unicorn", "raven", "sparrow", "scorpion", "coyote", "eagle", "zebra"]
"natureNoun": ["ocean", "mountain", "forest", "cloud", "river", "tree", "sky"]
"name": ["Arjun", "Yuuma", "Darcy", "Mia", "Chiaki", "Izzi", "Azra", "Lina"]
```

start symbol: sentence
The purple duck of the
The turquoise unicorn
The grey sparrow of the
The turquoise sparrow
The purple zebra of the
The indigo duck of the
The black sparrow of the

3: Adding your own symbols
Try adding a new symbol and a set of replacement rules. How should this story end? You could add a 'mood' symbol so that Darcy the blue raven is [happy, sad, angry].

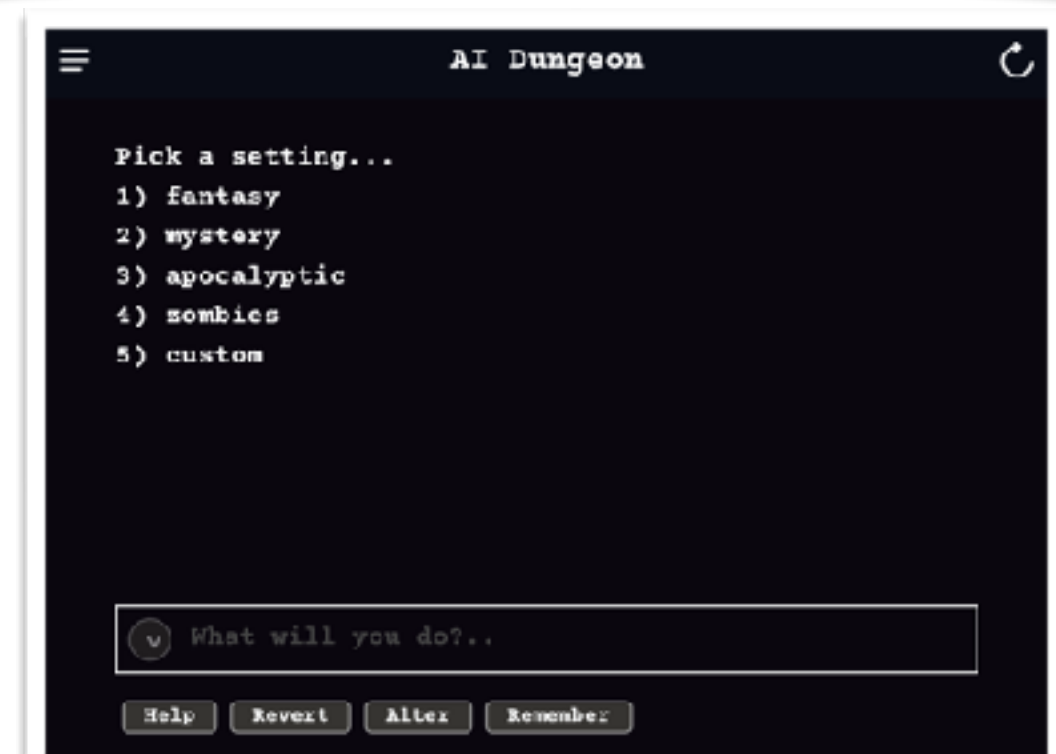
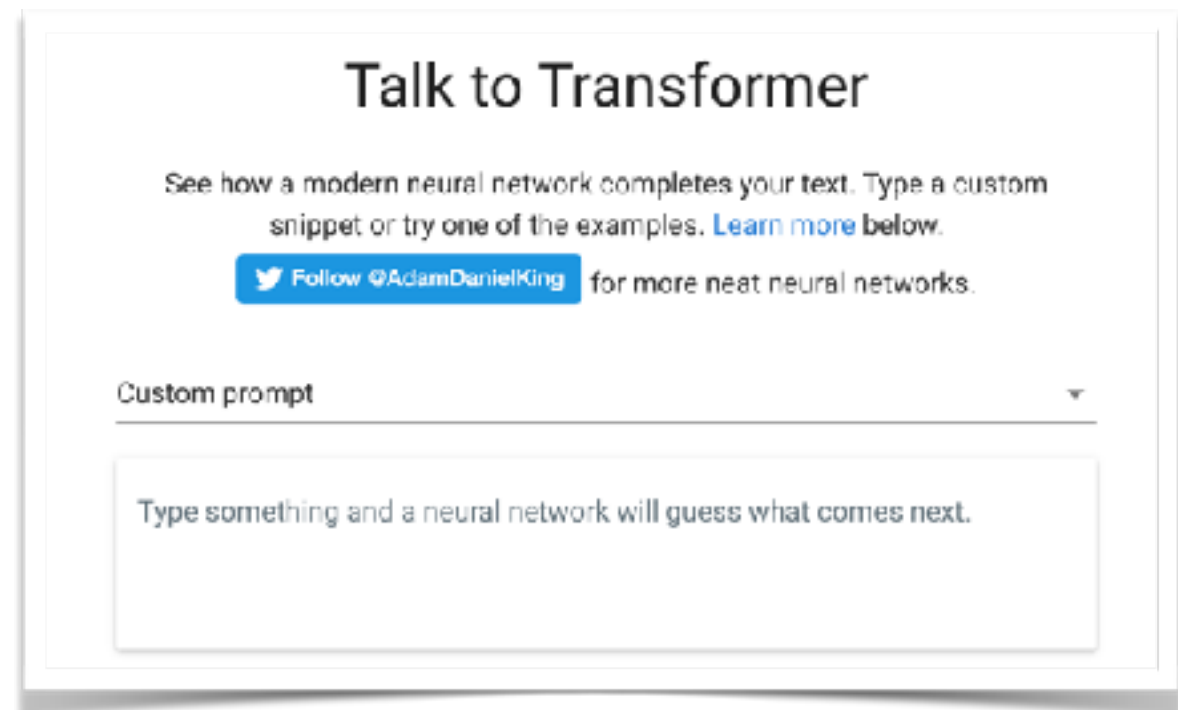
```
"sentence": ["#name# the #color# #animal# was ... something."]
"name": ["Arjun", "Yuuma", "Darcy", "Mia", "Chiaki", "Izzi", "Azra", "Lina"]
"color": ["orange", "blue", "white", "black", "grey", "purple", "indigo", "turquoise"]
"animal": ["unicorn", "raven", "sparrow", "scorpion", "coyote", "eagle", "zebra"]
```

start symbol: sentence
Chiaki the purple lizard
Izzi the turquoise eagle
Darcy the turquoise unicorn

Everyone is too busy

What is the absolute minimum time needed for someone to prove to themselves that your product works?

- journalist
- CTO / tech worker
- high-ranking professor
- new student



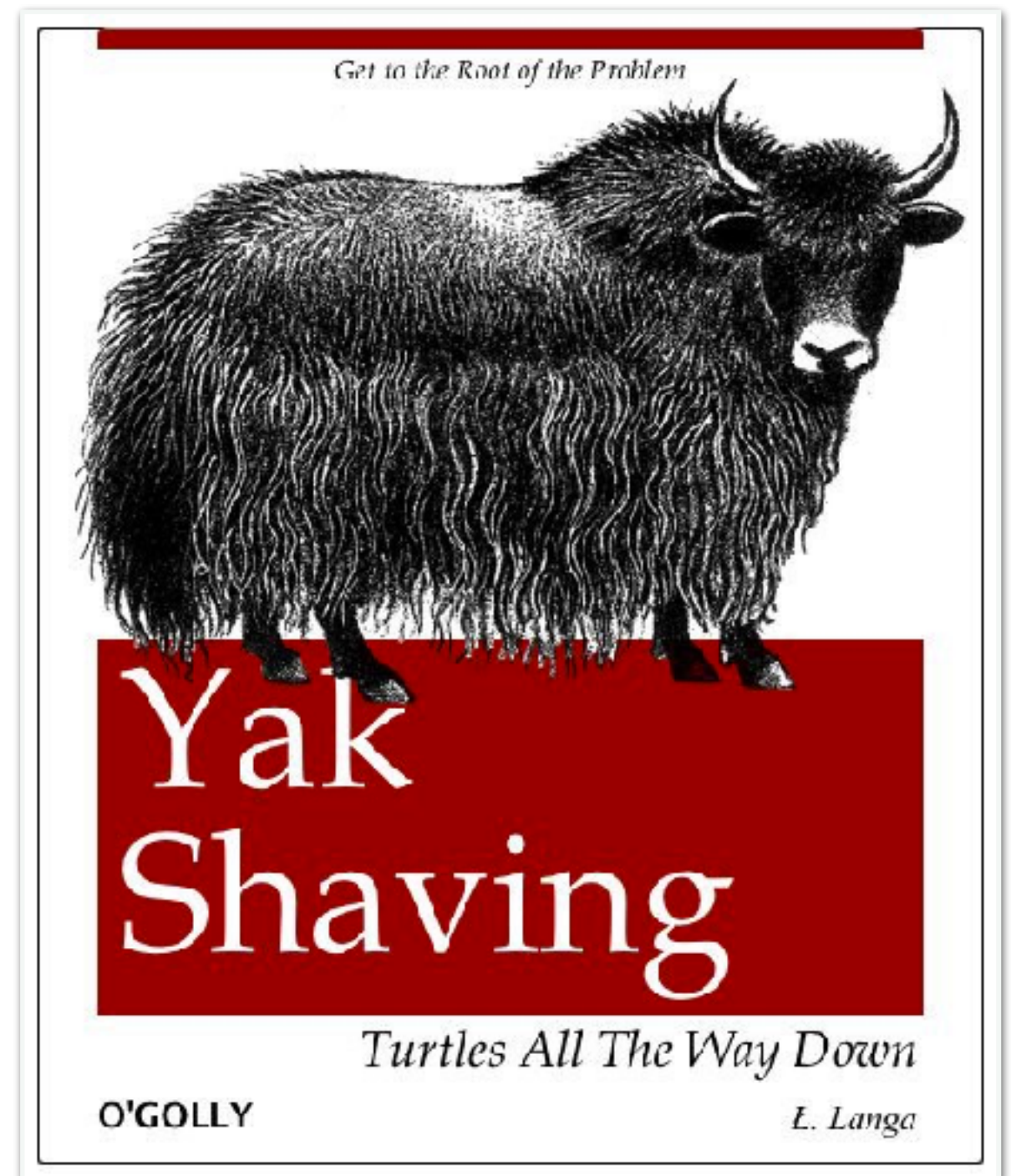
Everyone is too busy

What is the absolute minimum time needed for someone to embed your tool?

What are the dependencies?

Does this require a package manager? A conda container? A server?

Each additional complexity divides your users by 10



Language pri

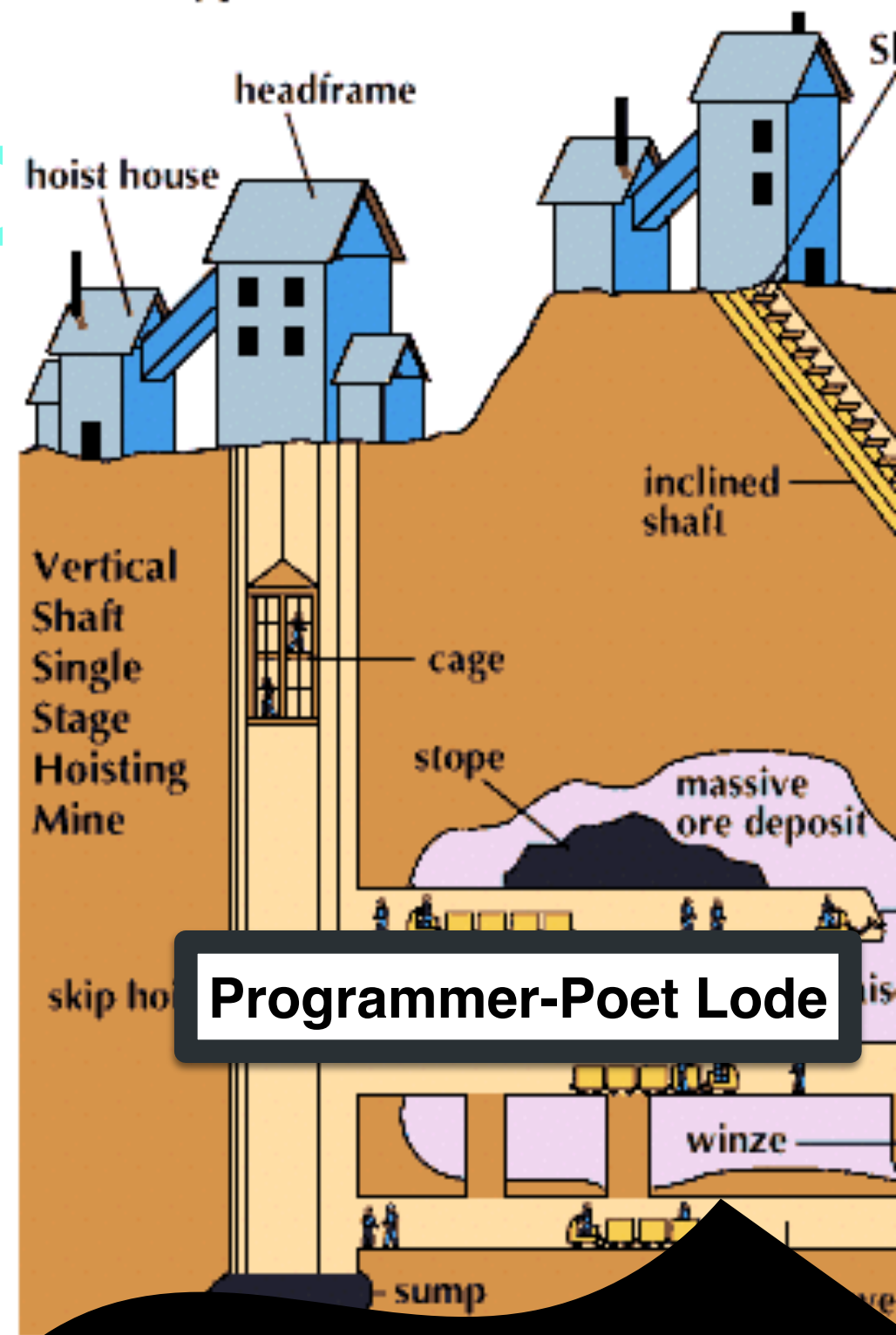
What is the absolute minimum time needed for someone to embed your tool?

What are the dependencies?

Does this require a package manager? A conda container? A server?

Each additional complexity divides your users by 10

Some Types of Mines



Programmer-Poet Lode

Lower Poet Lode

Make 1000 things, fast and bad

What is this new system for?

You don't know!

Guess from old genres

You can make one prototype

(it'll be wrong)

The 10th will tell you more

Diversity in users creates
diversity in creativity and use
cases

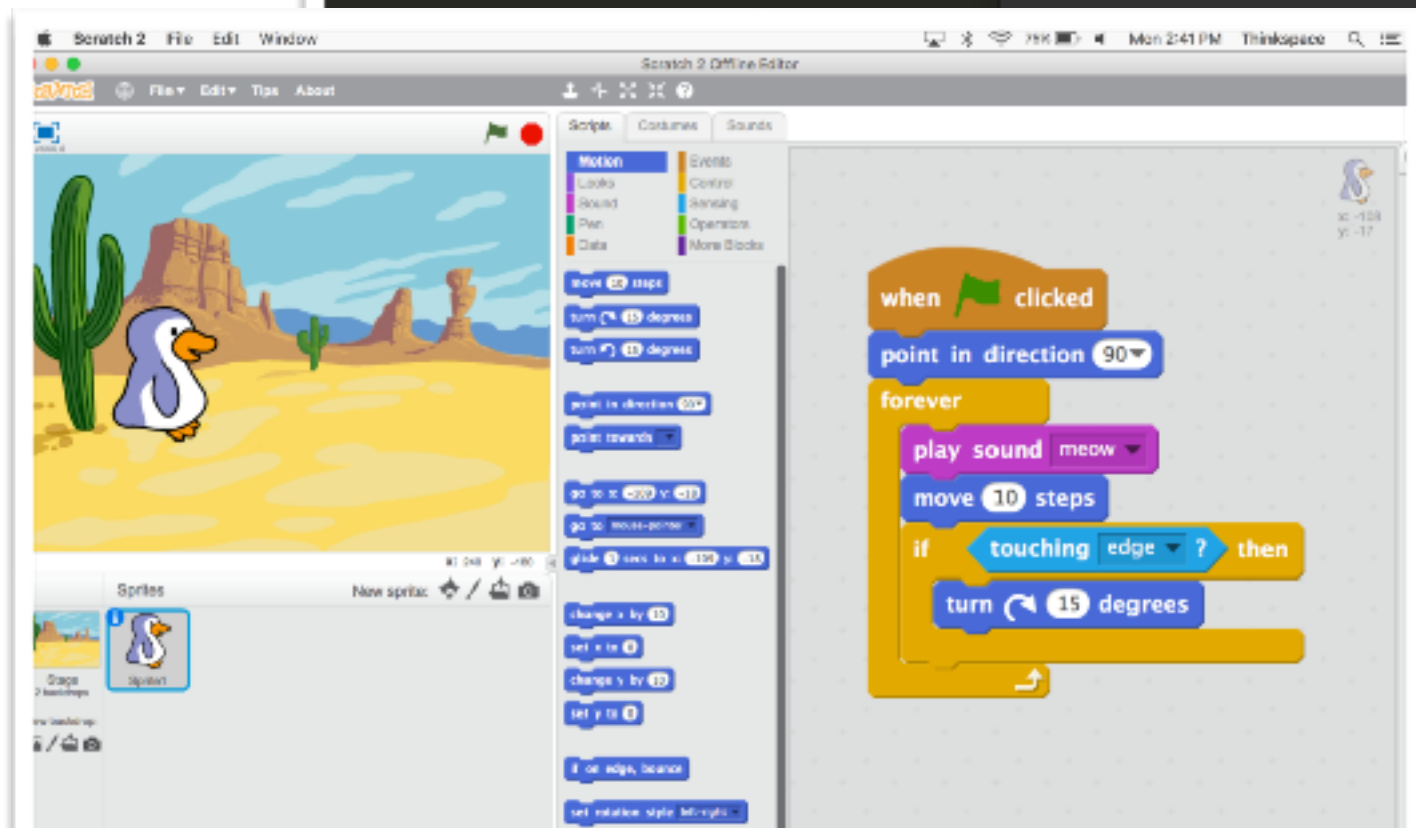


Language features for fast creation

- Scaffolded
- Unfoldable
- Little

```
<div class="container-fluid">  
  <div class="row">  
    <div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">  
      <h1>Title</h1>  
      <a href="#" class="btn">  
    </div>  
  </div>  
</div>
```

btn	Bootstrap
btn-block	Bootstrap
btn-danger	Bootstrap
btn-group	Bootstrap
btn-info	Bootstrap



Language features for fast creation



- Scaffolded
- Unfoldable
- Little

This is my webpage

This is **my** webpage

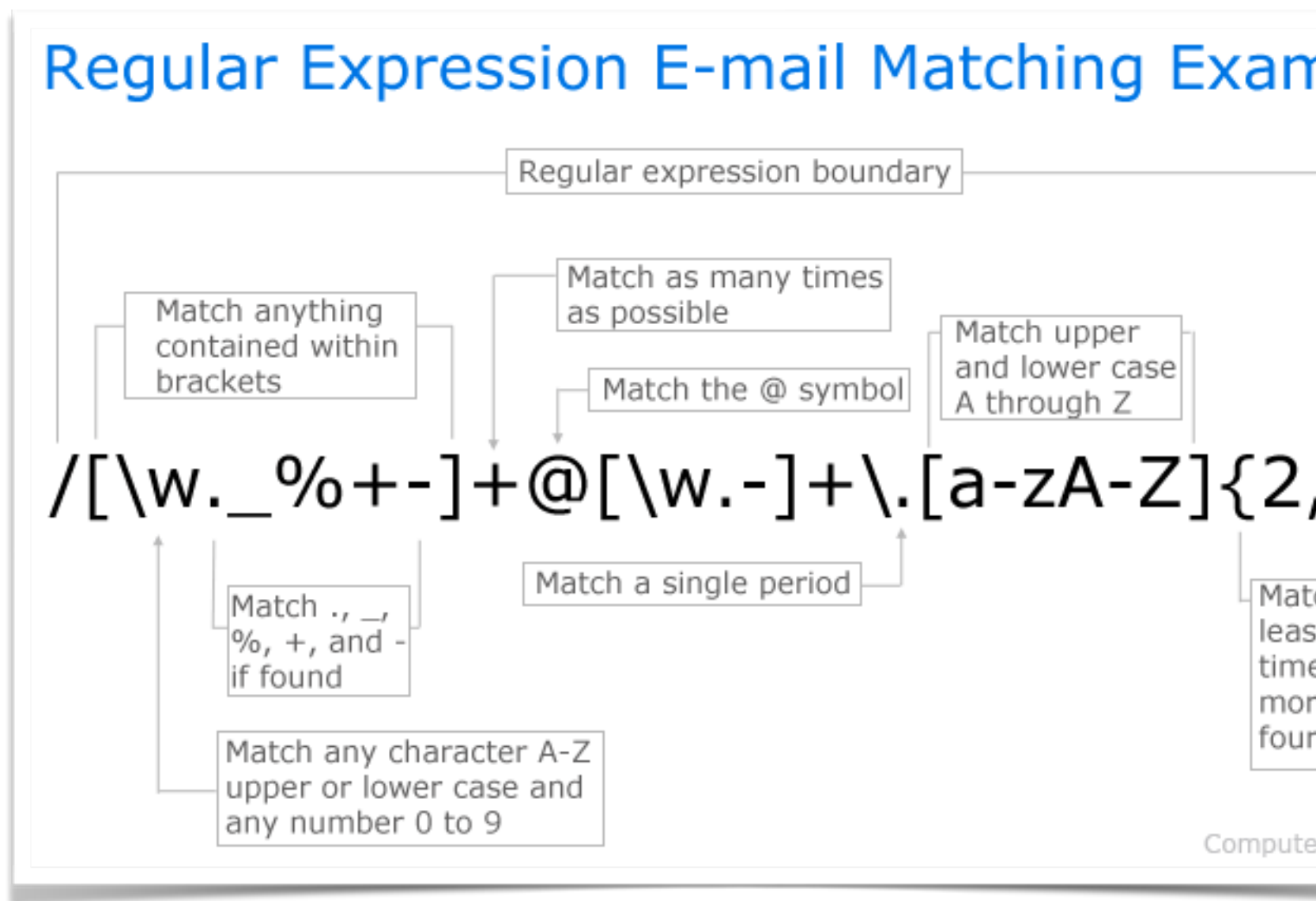
This is **my** [webpage](\"webpage\")

This is **my** webpage

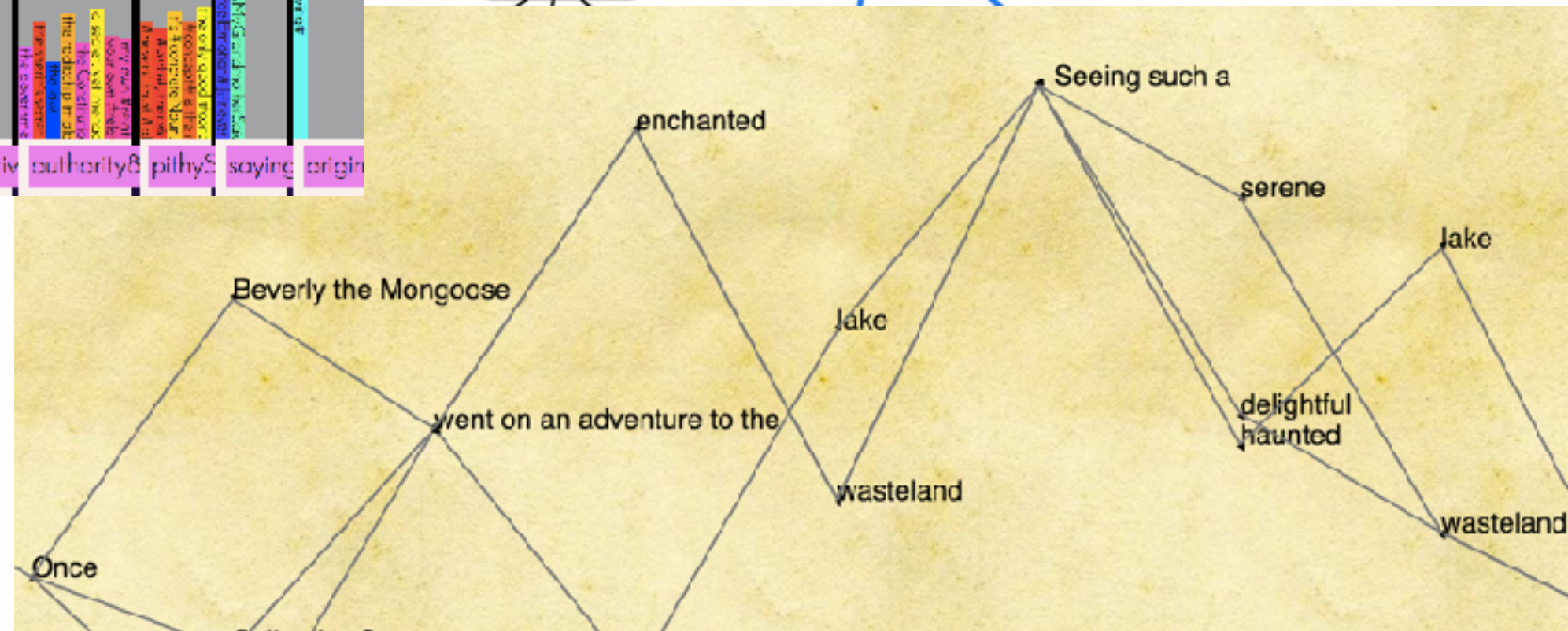
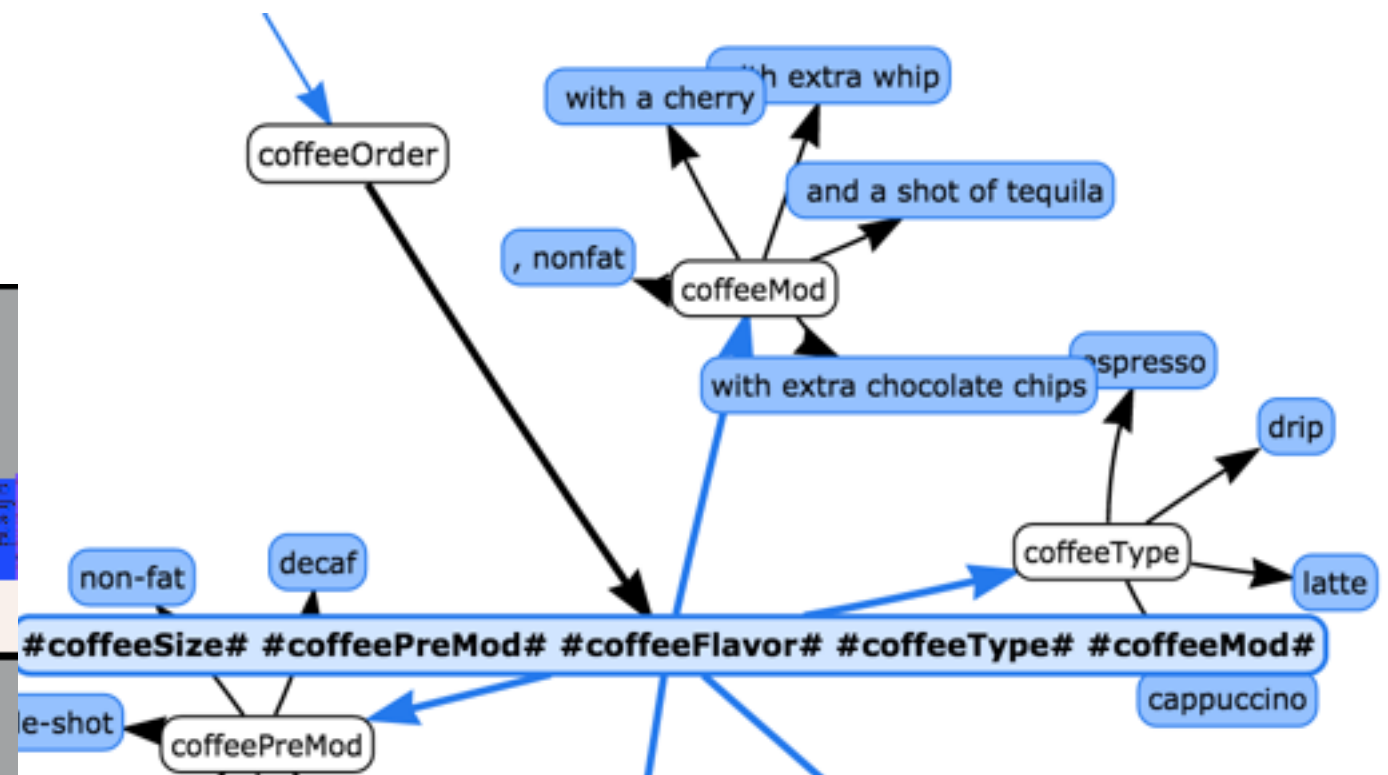
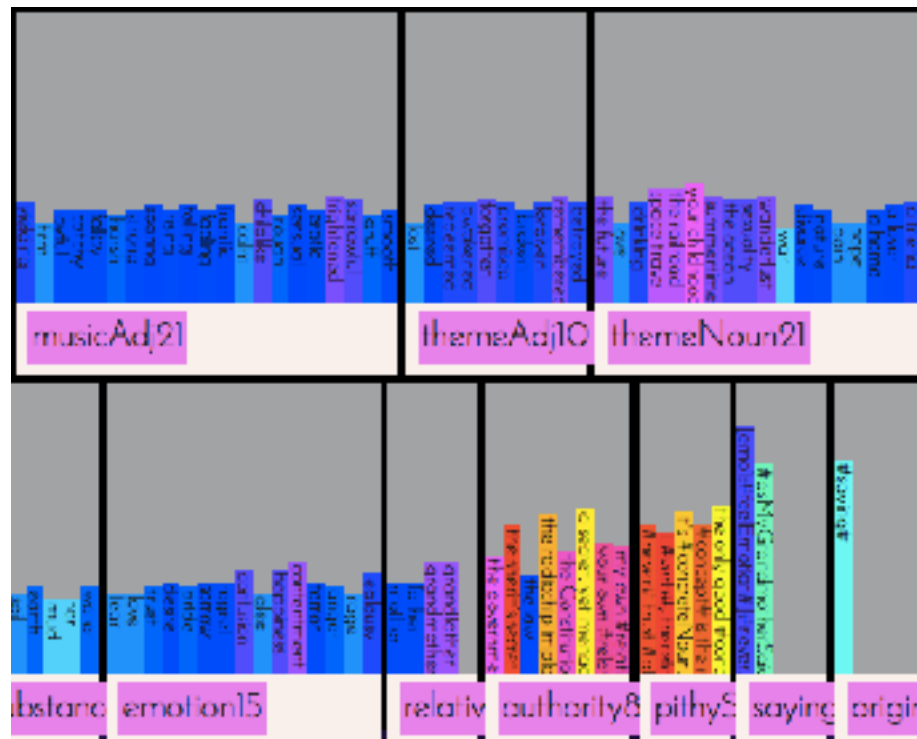
This is >my [webpage](\"webpage\")

Language features for fast creation

- Scaffolded
- Unfoldable
- Little

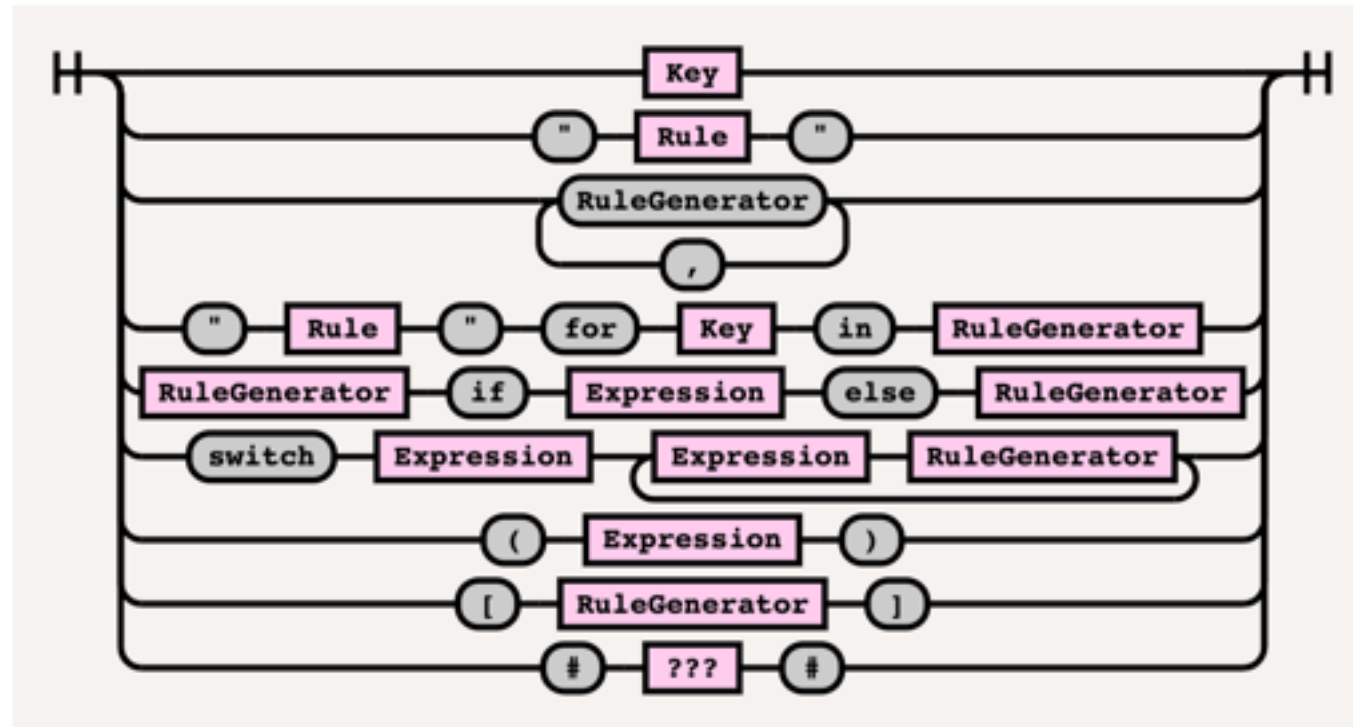


What didn't work?

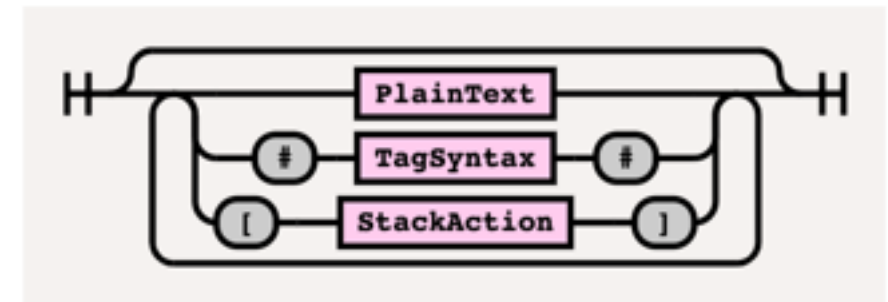


What needs more work?

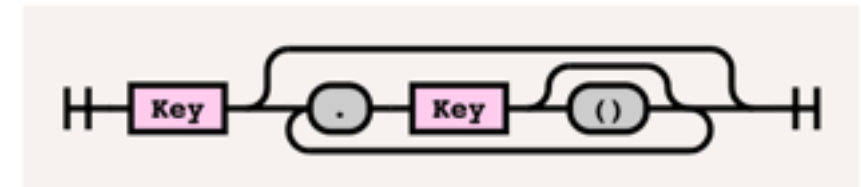
Rule Generator



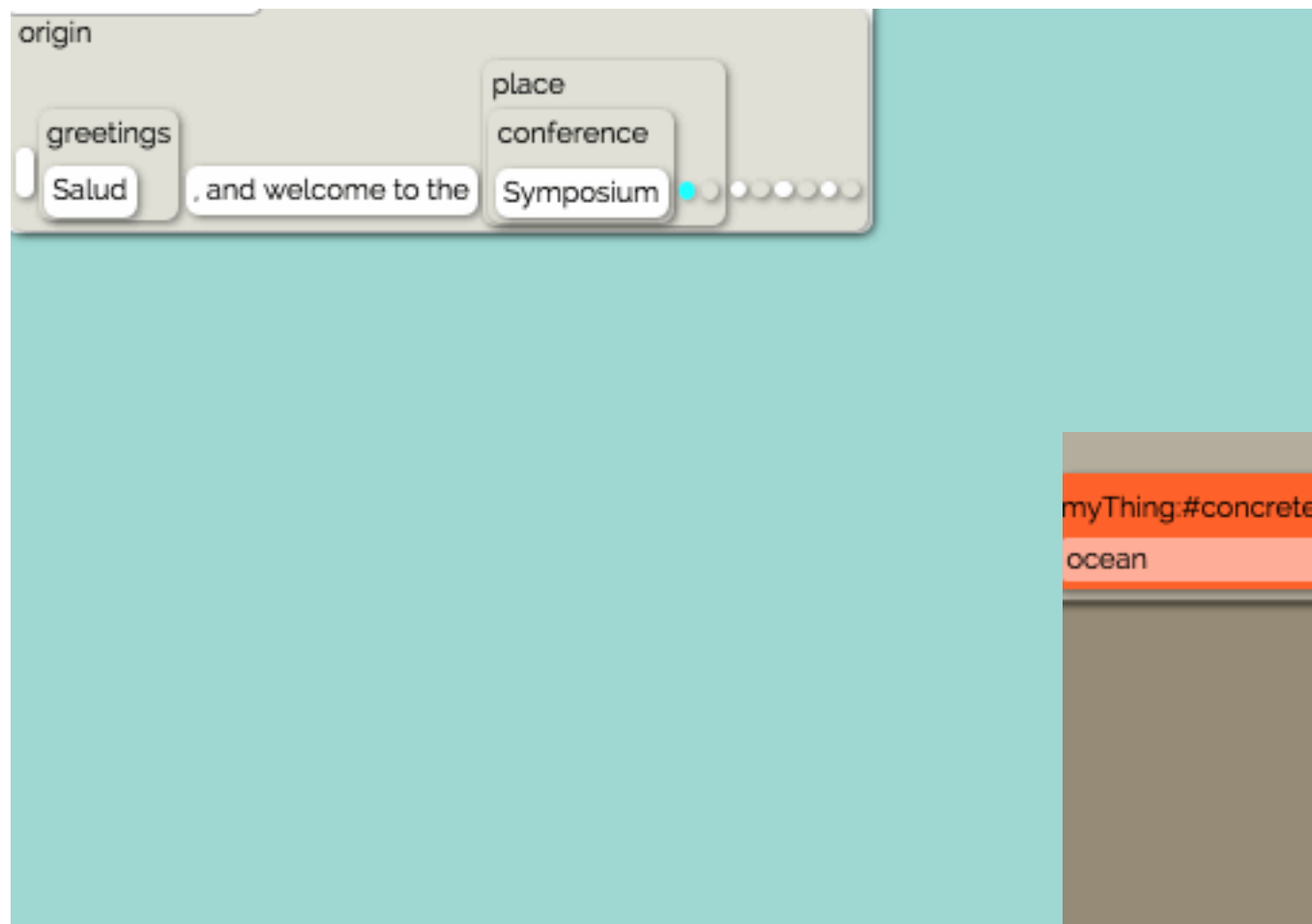
Rule



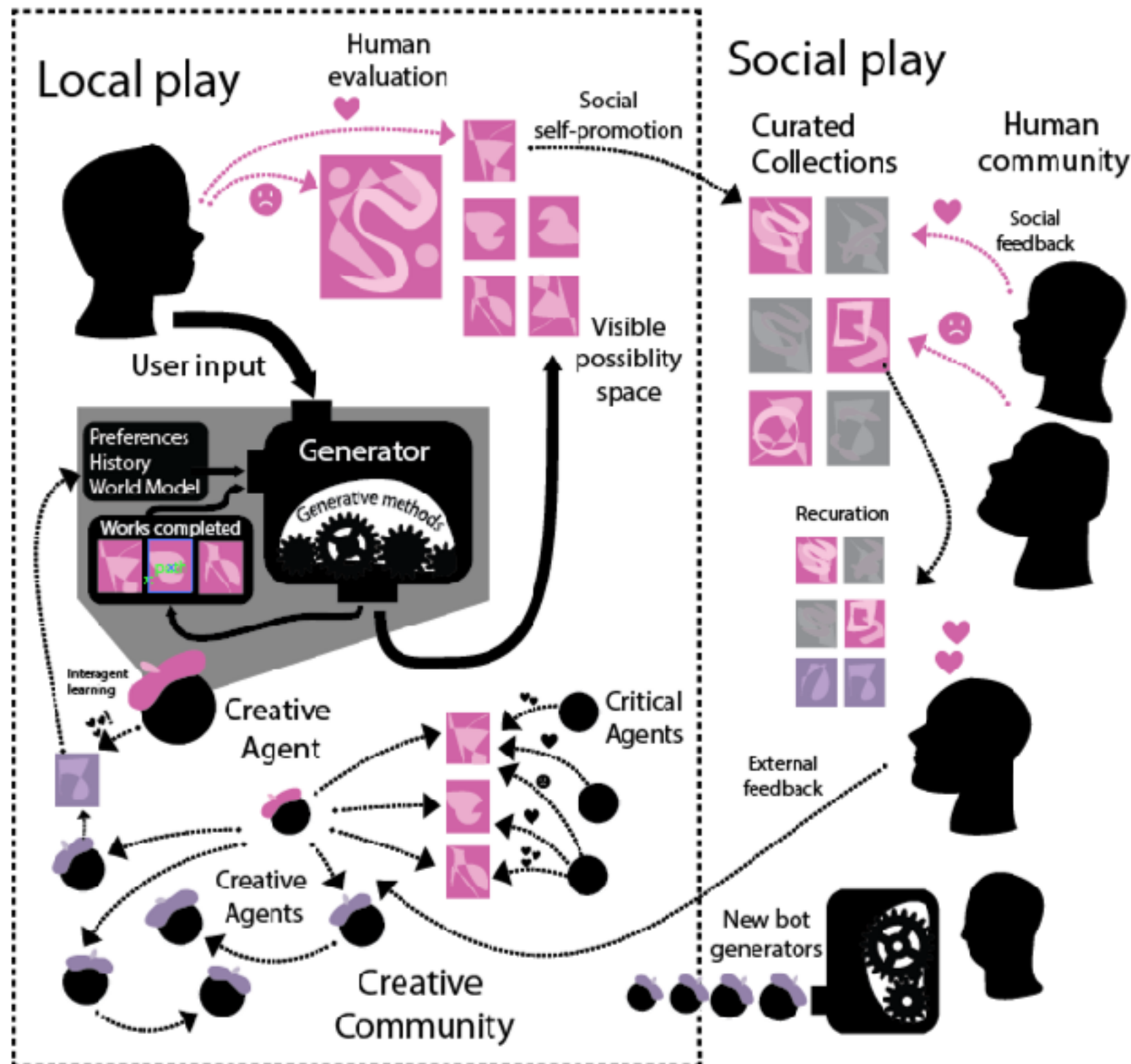
Tag Syntax



What needs more work?



How do predict user needs?



How do predict user needs?

A **metagrammar** generates new art-generating grammars

```
...  
shape: ["circle", "triangle", "rect"...]  
Color: ["blue", "magenta", "grey"...]  
ArtGrammar: ["...SetColor: #Color#  
    MakeShape: #someShapeType#..."]  
...
```

```
...  
SetColor: "turquoise"  
MakeShape: "rect"  
...
```

```
...  
SetColor: "magenta"  
MakeShape: "circle"  
...
```

A new **bot** is made by combining an art-generating grammar and a generated or authored heuristic

```
...  
SetColor: ["grey", "red"]  
MakeShape: "triangle"  
...
```

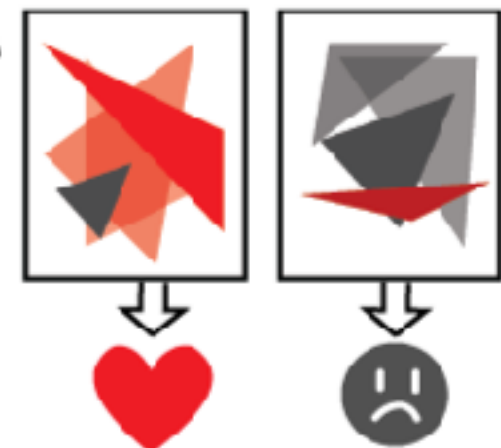
Many possible
(arbitrary)
heuristics

likes: pink (% of red or pink pixels in the art)

likes: full canvas
(100 - % white pixels)

likes: many round things
(# of times "circle" or "ellipse"
appears in svg output)

**Make and
evaluate art**



How do predict user needs?



Ranjit Bhatnagar @ranjit · May 31

Makin' graphics with @GalaxyKate's
Tracery brightspiral.com/tracery/



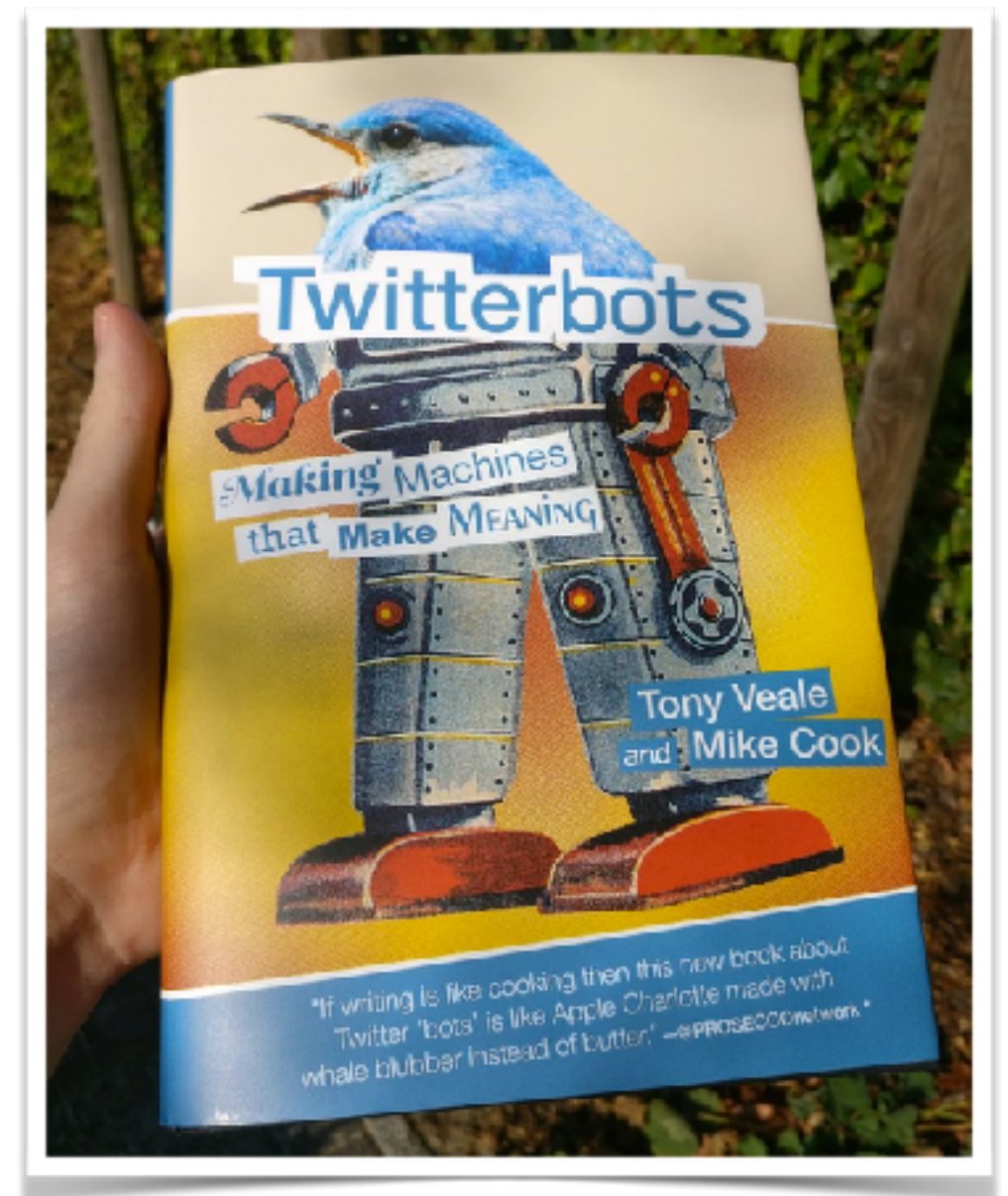
← ↺ 4 ★ 10 ⋮

[View photo](#)

How do predict user needs?

Lessons:

- Users may discover features
- Users may come up with terrible workarounds
- Empty spaces will get filled
- Design for surprise and flexibility
- How does modding work?



Lessons from Tracery/Chancery

- Users will define what the tool can make
- Design for openness:
 - What can happen with the tool
 - Where the tool can be embedded (in 2045!)
 - Where source files come from, and where they go
- THERE IS NO TIME!
 - Respect user time, and user attention, and user self-esteem

