

CSCI 334:  
Principles of Programming Languages

Lecture 14-1: Project Ideas

Instructor: Dan Barowy

**Williams**

Outline

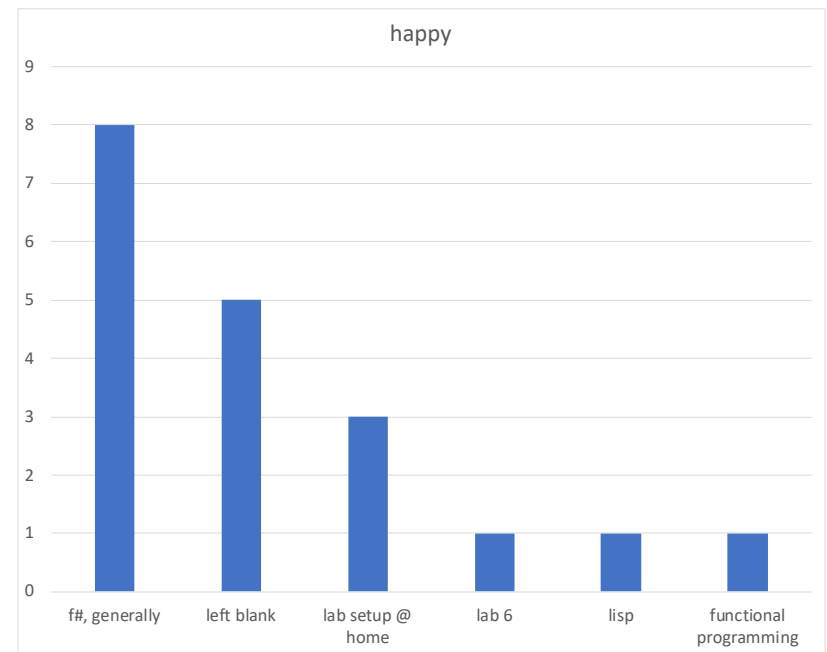
Feedback

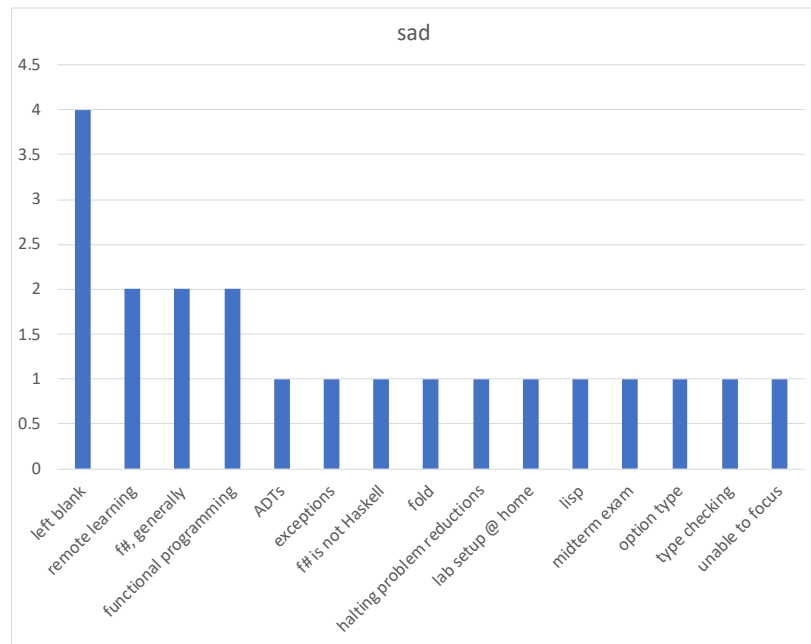
Weekly TODOs

Cool languages

Last year's final projects

Feedback



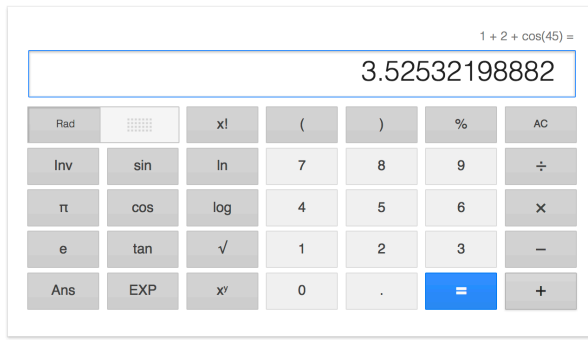


Reminder: Four TODOs each week

- Lab assignment      5-10 hours
- Quiz                      <20 minutes
- Activity                   <20 minutes
- Feedback                <5 minutes

Inspiration for Projects

Scientific Calculator



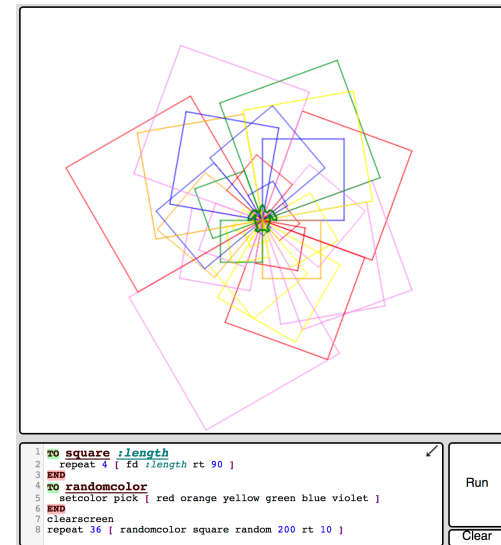
<https://www.google.com/search?q=google+calculator>

$$1 + 2 * 3 - 1$$

Should support:

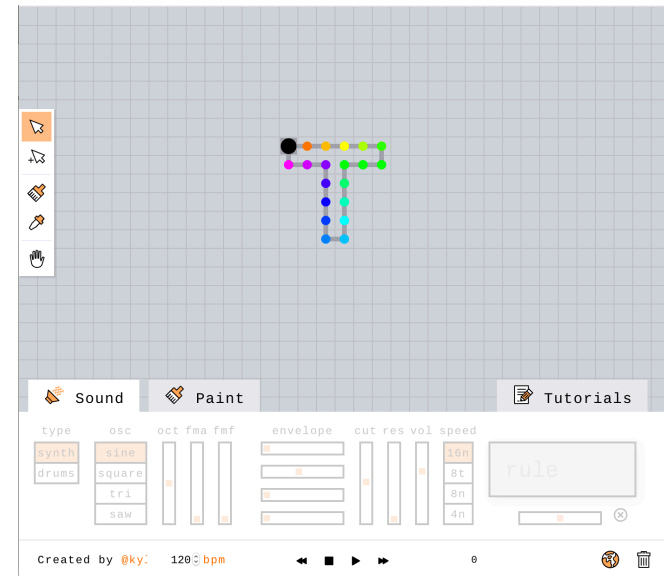
- infix expressions, or
- variables, or
- user-defined functions, or
- add some other interesting feature of your own design.

## Logo Interpreter



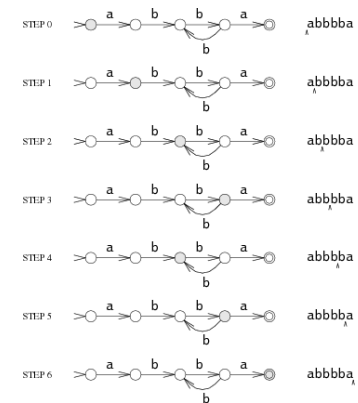
<https://www.calormen.com/jslogo/>

# Turtle Audio



<http://www.turtle.audio/>

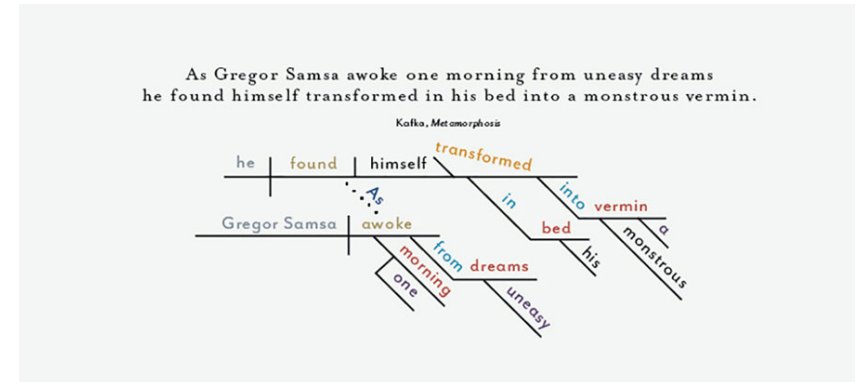
# Regular Expressions



<https://swtch.com/~rsc/regexp/regexp1.html>



# Auto Sentence Diagramming



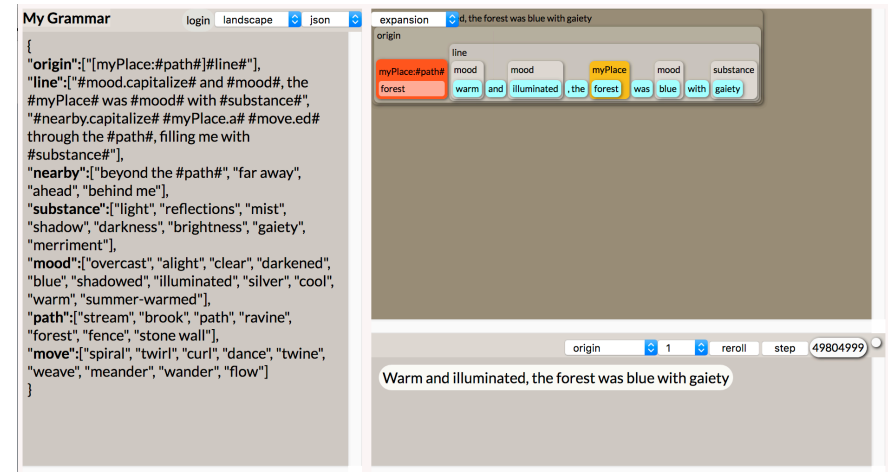
<https://www.npr.org/sections/ed/2014/08/22/341898975/a-picture-of-language-the-fading-art-of-diagramming-sentences>

# BASIC

```
710 IF PR(IPS,3)=1 THEN PRINT "Your victim is badly wounded" :: RETURN
720 PRINT "You killed your victim"
730 PR(IPS,1)=0 :: IVW=B+IPS
740 VP(IVW)=IKM :: IPS=0
750 RETURN
760 PRINT "Possible commands:"
770 FOR I=1 TO 15
780 PRINT H$(I)
790 NEXT I
800 RETURN
810 IF IVW<>0 THEN B40
820 PRINT "There is nothing to take"
830 RETURN
840 IF BLL=0 AND IKM<25 THEN PRINT
:: RETURN
850 PRINT "I take the ";V$(IVW);"
860 B(IBM)=IVW
870 IBM=IBM+1
880 VP(IVW)=0
890 IVW=0
900 RETURN
910 INPUT "What would you like to do"
920 LNG=LEN(D$)
930 IVWB=0
940 FOR I=1 TO 12
950 IF D$=SEG$(V$(I),1,LNG)THEN IVW
960 NEXT I
970 IF IVWB<>0 THEN 1000
980 PRINT "I do not understand you"
```

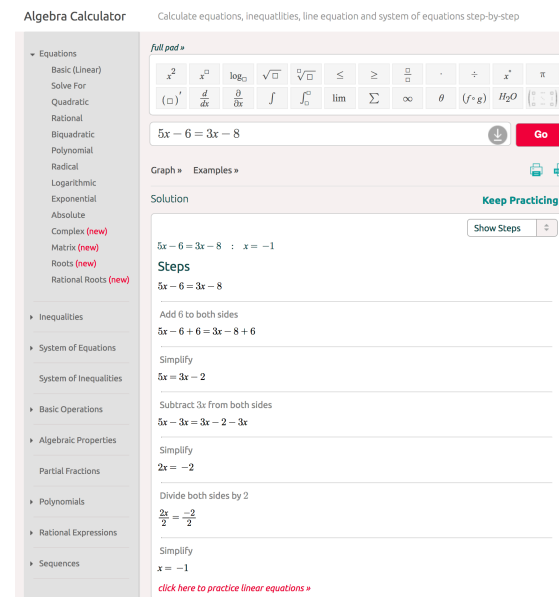
# A BNF parser and generator

```
<expr> ::= <var>
         | <abs>
         | <app>
<var>   ::=  $\alpha \in \{a\dots z\}$ 
<abs>   ::= ( $\lambda$ <var>.<expr>)
<app>   ::= (<expr><expr>)
```



<http://tracery.io/editor/>

# An algebra solver



<https://www.symbolab.com/solver/algebra-calculator>

Chuck

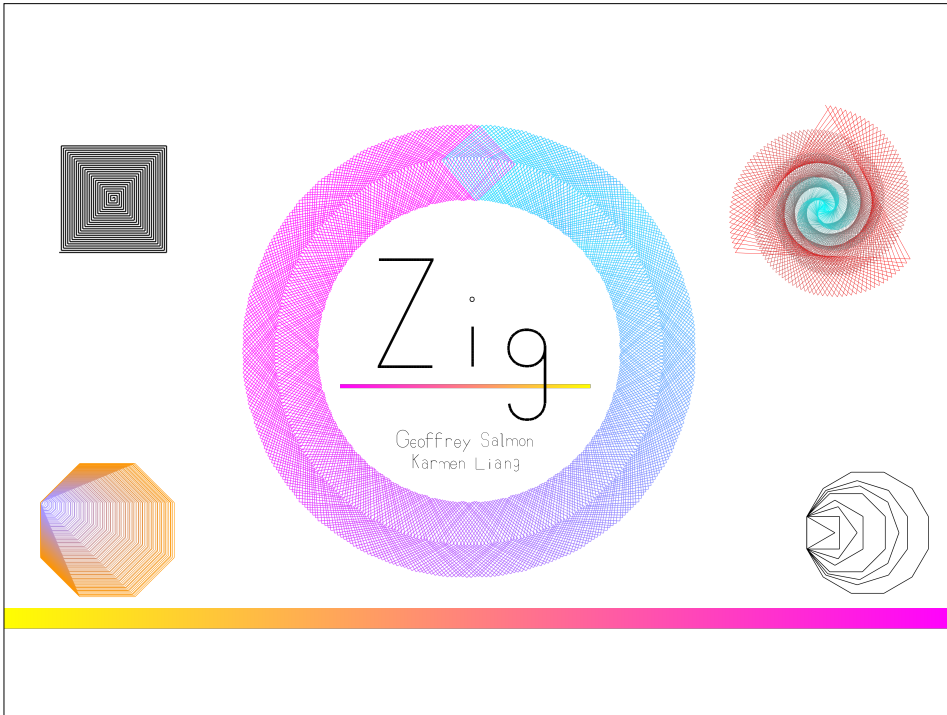
```
SinOsc ge => dac;  
  
while( true )  
{  
  Math.random2f(30,1000) => ge.freq;  
  .5::second => now;  
}
```

[https://www.ted.com/talks/ge\\_wang\\_the\\_diy\\_orchestra\\_of\\_the\\_future](https://www.ted.com/talks/ge_wang_the_diy_orchestra_of_the_future)

<http://chuck.cs.princeton.edu/>





Highlights from Last Year

Zig







## Vector graphics made easy

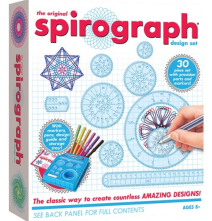
Goal: A readable and transparent interface  
(also fun)

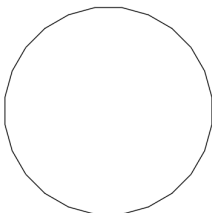
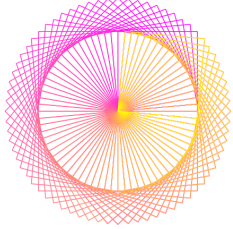

ahead 50; clockwise 90; ahead 50; clockwise 90;

ahead 50; clockwise 90; ahead 50; clockwise 90


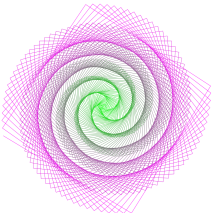
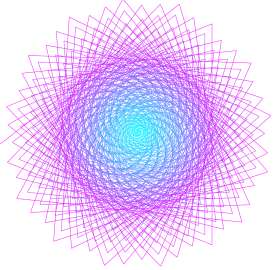


## Examples

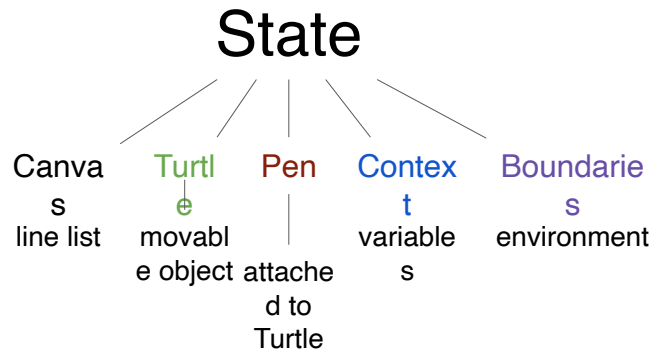
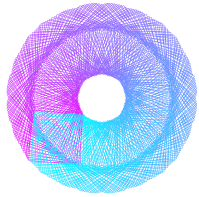




## Design Principles

- Moving turtle and pen
- Simple linear commands
  - Loops
  - Variables
  - Abbreviation

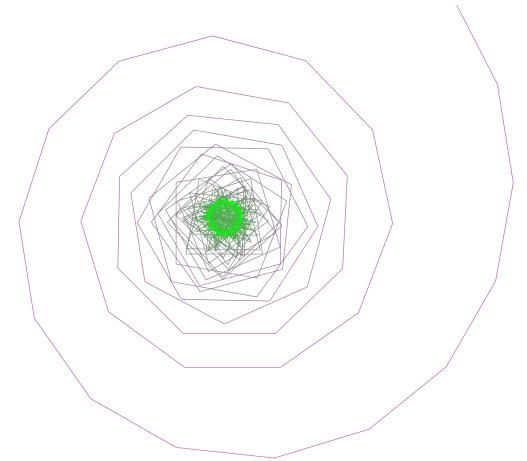




Behind the scenes



Future work

- Functions
- Lexical scoping
- Interactivity
- Eraser & screen color



F# Natural

**F Natural** : A language for music generation

—  
By Margaret Allen and Phoebe Huang

## Program Example

```
harmony h1 = {0, 4} -> h2 | END  
harmony h2 = {2, 5} -> h1
```

```
HarmonySet HS = {h1, h2}
```

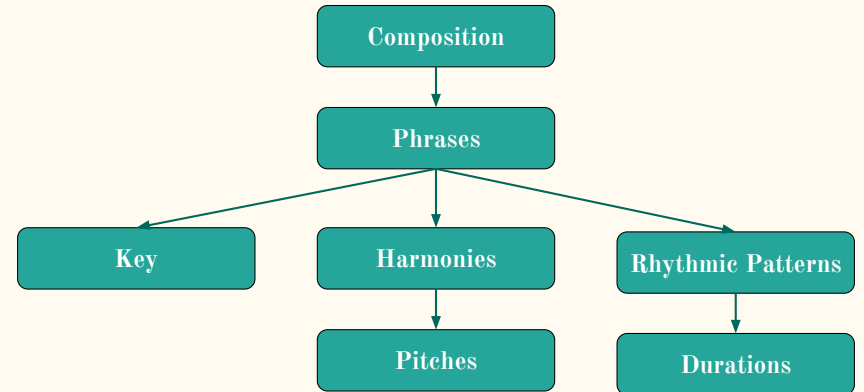
```
rhythm r1 = (2,2)  
rhythm r2 = (4,4,2)  
rhythm r3 = (4,8,8,4,4)
```

```
RhythmSet RS1 = {r1, r2}  
RhythmSet RS2 = {r2, r3}
```

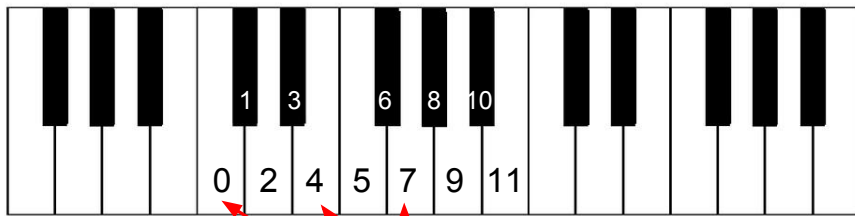
```
Phrase p1 = {C, HS, RS1}  
Phrase p2 = {G, HS, RS2}
```

```
Composition = (p1, p2, p1)
```

## Musical Grammar

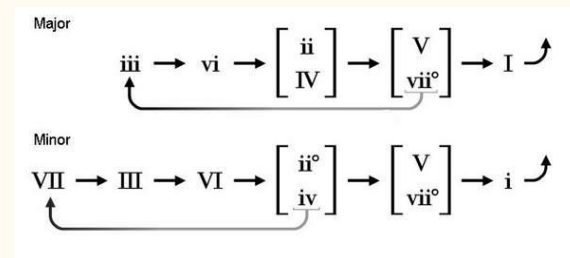


## Harmony in Key of C

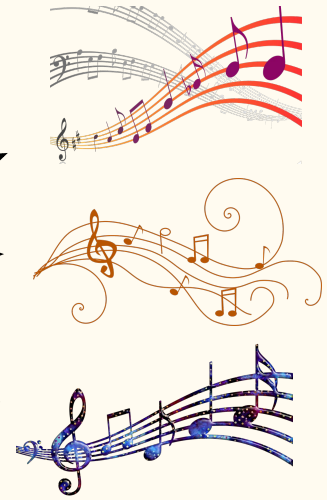
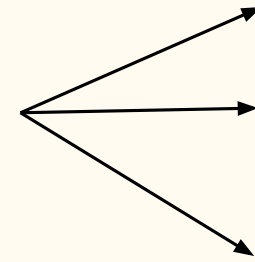
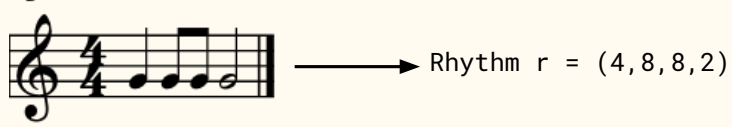
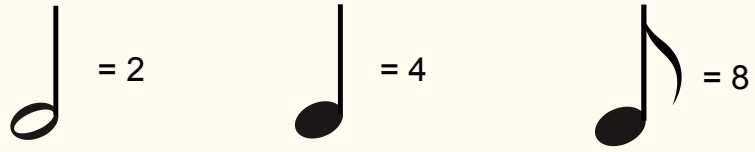


harmony I = {0, 4, 7}

## Harmonic Rules



# Rhythm

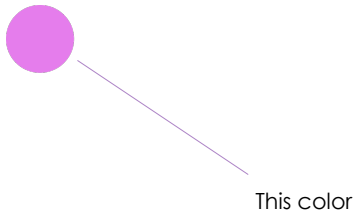


ColorScheme

ColorScheme

Noah Cowit

## What color is (229,125, 236)?



## Program: "p random (0,0,0)"

- "random" modifies color to a random value
- "p" prints color to color palate

Output :

"Parsed"

"random (0,0,0)"

"

(229,125,236)

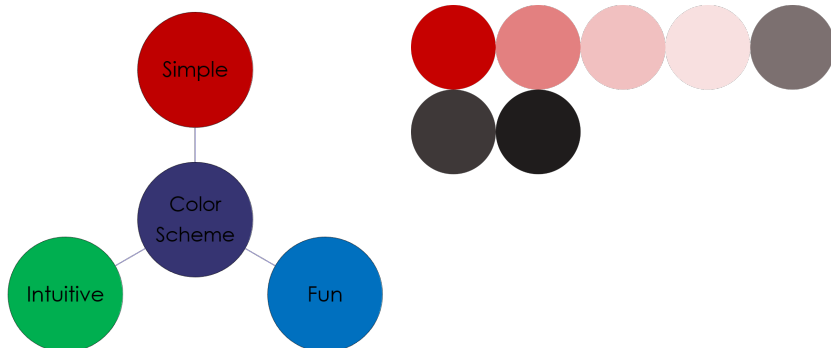
print: rgb(229,125,236)

-return value of "random"

Function

-return value of "p" Function

## What is ColorScheme?



## We want a darker color

- "p sub (229,125, 236)"

- print: rgb(114,62,118)





Yay Colors!



## Recap & Next Class

This lecture:

Administrivia

Project ideas

Next lecture:

SQL