Computational Form

JUSTIN BAKSE

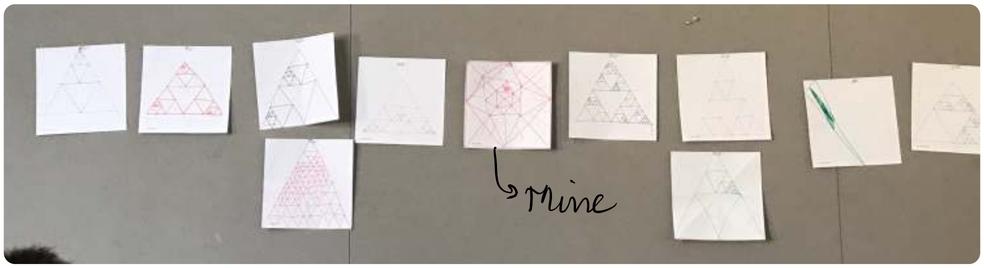
LEGEND

🟥	: Assignments & Homework
🟨	: Storyboarding / Brainstorming
🟩	: Artist / Designer to look up
🟦	: Events & prospects to explore

25 Jan 2019

✱ CLASS WEBSITE : COMPFORM.NET

- Marathon of constant working - 5 assignments a week
  - ↳ NO FINALS, NO WRITTEN ASSIGNMENTS
- 2 min paper & Game makers toolkit
  - ↳ Great resources
- Coding Model : Breadth first & Depth first
- Assignment : Designing the Sierpinski triangle



"In this class, we will make things that make things."

- This class is NOT about a computer.
  - ↳ Computation exists physically as well.

- John Cage
  - ↳ 4 min 33sec of no music — think of the space around

- Idea:
  - ↳ Showing wireless network overlay on top of real world. (AR)

- Comp Form Reference: Procedurally generated games: Minecraft, Dwarf Fortress

- About the class: Make. Make. Make. Don't think of the concept.

- Idea:
  - ↳ Create procedurally generated data & create a story around that — for Storytelling with Data.

- Idea:
  - ↳ Create projects in Comp Form using Laser Cut, Textile, & other stuff at the Making Center.
    - ↳ Think about Procedurally Generated Art

      → Following a ritual

- Idea:
  - ↳ Using ML database of images & creating procedural art with it. [CITE THE REFERENCES]

- Time spent on each sketch: Min - 60 min; Max - 120 min.

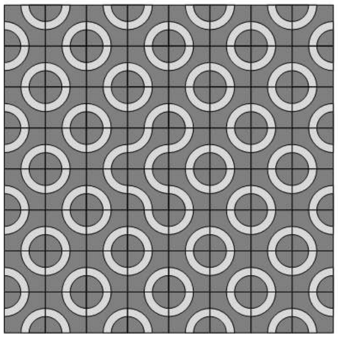
◦ Theme for this week's assignment : JPG images

§ TILE SETS

◦ Batman Game : NES (Nintendo Entertainment System)

↳ Tile based geaphics (64 bit system)

◦ Truchet Tiles → Rotatable Tiles



01 Feb 2019

- Next week's after class : Discovering OOP in JavaScript.
- Added features on the website (Compform.net)
  - ↳ Comment with emoji

- Cyriak Animator

☆ RANDOM VALUES

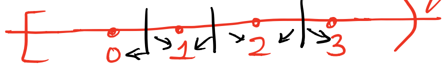
Q : Considering the tile set as a tool, how do tools impact authorship?

↳ the related project by Brooke Smith

- Random is very valuable in generating sth. Random can mean unplanned, unexpected, unpatterned, uncontrolled, or unpredictable.

- Generating Random Numbers
  - ↳ JS : Math.random() → Value range [0, 1)
    including 0, excluding 1 ←
  - ↳ Creating a range :
    random() * 5 + 10
  - ↳ integers → Math.<u>floor</u>()

- Be careful about random gener^{n}:
  - ↳ round does not do equally likely option



  - ↳ Math.floor(Math.random()* 7);
    - ↳ this would give a 0. So, it won't work for a rolling die.

- "min" f^{n} in p5 → min(a, b);
  - ↳ lower value of a & b

- Bell curve/Normal distribution → very common in nature

- randomGaussian() → f^{n} in p5 that gives a value very randomly, 1, -1, 2, 3, 100000, 5, 6, ---
  - ↳ use carefully

- Randomize process of thought: Dice vs Deck
  - Dice: normally distributed
  - Deck: ↳ reliable

- To give a 10% likelihood in a f^{n}, you can use:

```
if (random() < 0.1) {
    console.log("Rare think");
}
```

- Use random just once in a code. Rolling it multiple times would not give the right probability.

Deck → 1

Normal bias → 3

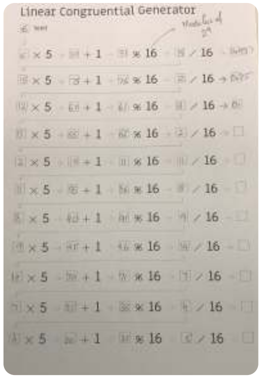
High → 5

Low → 2

Pure random → 4

Think about diff^t types of random.

- Pseudo random values → created using a deterministic process.
  - ↳ Common method : LCG (Linear Congruential Generator)



Linear Congruential Generator

seed

```
[illegible] × 5 → [illegible] + 1 → [illegible] % 16 → [illegible] / 16 → [illegible]
[illegible] × 5 → [illegible] + 1 → [illegible] % 16 → [illegible] / 16 → [illegible]
[illegible] × 5 → [illegible] + 1 → [illegible] % 16 → [illegible] / 16 → [illegible]
[illegible] × 5 → [illegible] + 1 → [illegible] % 16 → [illegible] / 16 → □
[illegible] × 5 → [illegible] + 1 → [illegible] % 16 → [illegible] / 16 → □
[illegible] × 5 → [illegible] + 1 → [illegible] % 16 → [illegible] / 16 → □
[illegible] × 5 → [illegible] + 1 → [illegible] % 16 → [illegible] / 16 → □
[illegible] × 5 → [illegible] + 1 → [illegible] % 16 → [illegible] / 16 → □
[illegible] × 5 → [illegible] + 1 → [illegible] % 16 → [illegible] / 16 → □
[illegible] × 5 → [illegible] + 1 → [illegible] % 16 → [illegible] / 16 → □
[illegible] × 5 → [illegible] + 1 → [illegible] % 16 → [illegible] / 16 → □
```

- Modulus operator (%) → gives remainder
  - ↳ 13 % 5 → 3
  - ↳ 3 % 5 → 3

- RandomSeed → control the type of random nos. we want to get.
  - ↳ Trick : give the random seed a random number

- Using colorMode
  - ↳ colorMode (HSB, 1)
    - ↳ Play with it

  Playing with colors on Hue, Saturation, Brightness

✲ Assignment : Theme : Random ; Make sketches & post

↳ Recreate the work of one of the artists (like Mondrian)

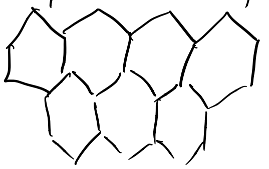
08 Feb 2019

* Group question : Does computationally generated art have an inherent aesthetic ?

↳ Almost always perfect (pixel sharp)

↳ Aesthetic varies with the perception of the artist.

↳ A lot of geometry (similar to nature)



→ honey comb

ants move in a line

↳ An example artwork that justifies aesthetic : path of a bouncing ball — Batool Ak

* PARAMETERS

- Parameter space : a set of all possible combin^{ns} of values for the parameters of a sys
  ↳ eg : 8 boolean parameters → $2^8 = 256$ possibilities
- Samey-ness : Procedurally generated art is not the same. Its <u>almost</u> the same.
- Parametric Design : Creating a system that creates a design — rules to design a layout.
  ↳ eg : Input height of a bicycle rider $\xrightarrow[o/p]{}$ makes a bicycle

- Benefits of Parameteriz^{n}
  - ↳ Better code, better UX, better results

✯ INTERFACE

↳ Think about the parameters you are using for designing an interface

eg:
```
function rect(x, y, w, h) {
}
```

↳ Think about which parameters you want to expose.

↳ Which parameters are optional?

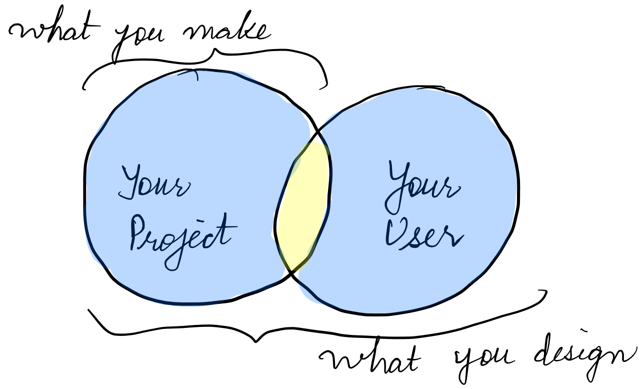
↳ Which value should be accepted for each parameter.

are these parameters method oriented or goal oriented?

- Interface types → UI
  → API

§ Idea

↳ Creating a painting canvas with parameters (Similar to photoshop)

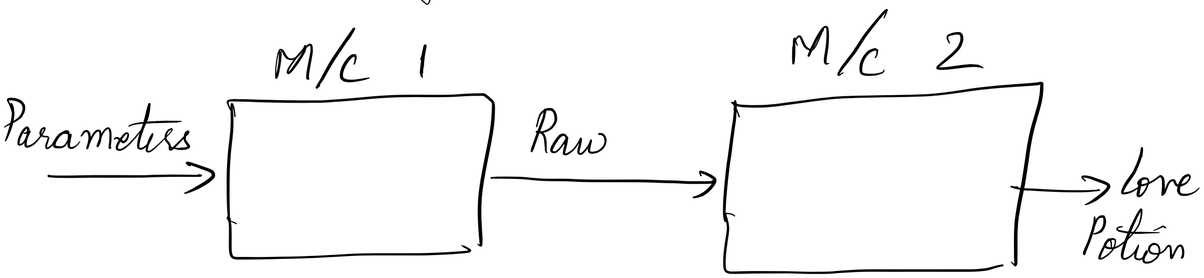


- Imperative programming vs Declarative programming
  - ↳ eg: JavaScript
  - ↳ eg: SQL

Procedural

A way of thinking about an interface

- Group Activity : Fictional Machines
  - ↳ Love Potion Mixer
    - ↳ Parameters : Goal ① Degree of love (obsession)
      - ② Duration
      - Hidden ③ Subject
      - ④ Activⁿ time
      - Method ⑤ Form
        - ↳ Density
      - ⑥ Flavours
    - ↳ User Type : Machine (a part of the chain)



- Choose 3 parameters for your user
  - → Method oriented : Form (Density)
  - → Goal oriented : Degree of love
  - → Hidden : Subject

- Datatypes
  - → Degree of love : num degreeOfLove
    : 0 – 1
  - → Form (Density) : num density
    0 – 1
    (liquid) (solid)
  - → Subject : person class subject
    [object]

- Use p5 dom library for adding interactivity (sliders, say) on a web sketch.
- createP(' '); → writing a paragraph.
- color picker : var colorPicker = createInput("#....", "color");

- Randomly picking

  ↳ if (random() < 0.5) { square_size = 10; }

  else { square_size = 50; }

  ↳ Ternary operator

  square_size = (random() < 0.5) ? 10 : 50;

chance (let large_chanceN = 55;)

Normalized value from 0 to 1 (clue)

- Using "let" for variable declaration

  ↳ works only within ()

  ↳ for global working, use "var".

- Using modulo to choose b/w rings → choosing b/w 0, 1, 0, 1, ...

— x —

- Challenge

  ↳ Create a face generating tool

- Do 5 sketches on Parameters (share min & map sketch)

15 Feb 2019

§ Homework Review with Alonso & Josefina

↳ Choose a sketch that you think started with a plan. What benefits did starting with a plan have?

① Anna's project : Drawing with natural language

↳ The end result is structured

— ∞ —

- Idea : Use shaders for creating gradients.

- Next week
  ↳ Extra class on node.js.

☆ Upcoming field
  ↳ Natural lang. processing

— ∞ —

☆ NOISE
↳ Different from random( ).

- Activity : Create random fn from an array w/o using random fn

BRAINSTORMING

$\boxed{1.}$567 Math.floor ( )

11.6876 → Time input

randomValues [_]

CLASS Discussion

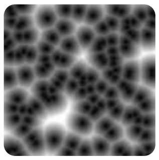
```
↳ fill(randomValues[i], randomValues[i+1], randomValues[i+2]);
  i += 3;
  if(i === 90) {
      i = 0;
  }
```

— ∞ —

☆ Note: Consider choosing noise fn whenever you feel like using random. Noise gives "controlled" variation.

- ○ Noise in p5: Perlin Noise (created by Ken Perlin for the film Tron)
- ○ Diff^t types of noise f^{ns}:

Perlin

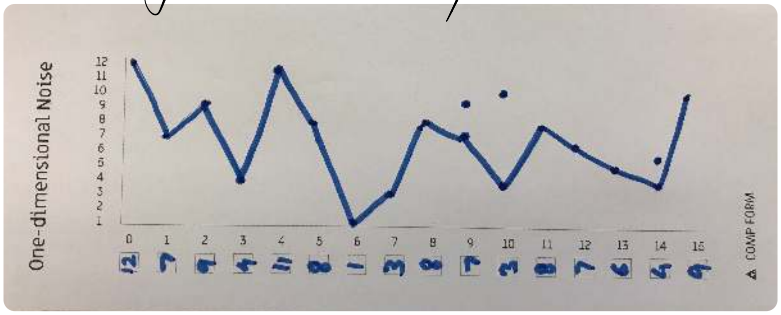

worely

Value

Improvement over Perlin noise: Simplex Noise

- ○ Noise is used very often to make terrain (mountains), clouds, wood texture
- ○ randomSeed() applies to all. Noise gives better control.

Activity : Linear Interpolation

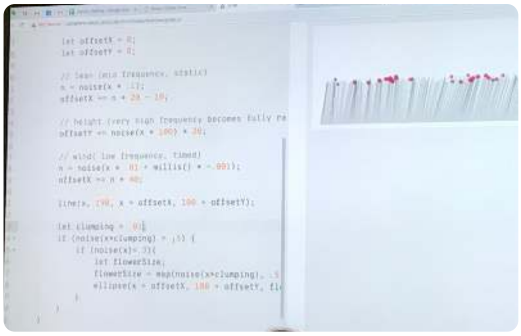


- Benefits of Noise
  - Looks good
  - The variation in Perlin Noise is band-limited
    - ↳ ≡ high pass & low pass filters
  - Visually isotropic — looks the same when rotated.
  - We can control the freq. of values with noise.
    - ↳ + Amplitude, Octave
      - ↳ No. of details from one value to the other.

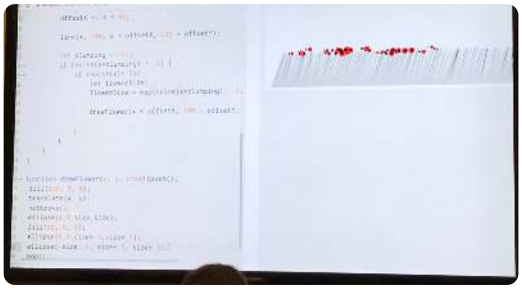
— x —

- Noise doesn't generate a random value. So, it needs a parameter.
  - ↳ noise(x, y, z);
    - ↳ how fast we vary these inputs determines the freq.
- Noise range : generally 0 – 1
- More fns : noiseDetail(); , noiseSeed();

- Noise frequency
  - ↳ noise (x * 0.001, y * 50);
    - ↳ low freq → x , More freq → y



```
let offsetX = 0;
let offsetY = 0;

// lean (mid frequency, static)
n = noise(x * .1);
offsetX += n * 20 - 10;

// height (very high frequency becomes fully ra

offsetY += noise(x * 100) * 20;

// wind( low frequency, timed)
n = noise(x * .01 + millis() * .001);
offsetX += n * 40;

line(x, 190, x + offsetX, 100 + offsetY);

let clumping = .01
if (noise(x*clumping) > .5) {
    if (noise(x)>.3){
        let flowerSize;
        flowerSize = map(noise(x*clumping), .5
        ellipse(x + offsetX, 100 + offsetY, fl
    }
}
}
```



```
    offsetX += n * 40;

    line(x, 190, x + offsetX, 100 + offsetY);

    let clumping = .01;
    if (noise(x*clumping) > .5) {
      if (noise(x)>.3){
        let flowerSize;
        flowerSize = map(noise(x*clumping), .5

        drawFlower(x + offsetX, 100 + offsetY,

      }
    }
  }
}

function drawFlower(x, y, size){push();
fill(255, 0, 0);
translate(x, y);
noStroke();
ellipse(0, 0, size, size);
fill(0, 0, 0);
ellipse([illegible] size [illegible]);
ellipse(-size [illegible] size [illegible] size [illegible]
pop();
```

- Challenge for next week
  - ↳ Generate a Treasure Map (Check syllabus)

22 Feb 2019

§ Today's group: Noah & Kevin

↳ Choose a sketch that's unlike others. Why so?

↳ Felix's p5 sound note envelope

↳ why: numbers, sound

○ Create a sketch by using random sketches in noise.

☆ STRATEGIES

↳ About how do we get better at programming

↳ Thinking about strategies & tactics for common problems.

Strategy	Tactics	Primitives
↳ Big picture plan	↳ Small plan	↳ Building blocks

○ "Strategy without tactics is the slowest route to victory. Tactics without strategy is the noise before defeat."

— Probably not Sun Tzu.

○ Primitives are atomic ≡ indivisible (smallest unit)

↳ usually never broken down

- Trick : Recognized the abstract similarity b/w problems.
  ↳ eg : bouncing ball → change posⁿ + collision detection
  Similar situations : music notes (high - low)
  : ball color pulse (bright - dark)

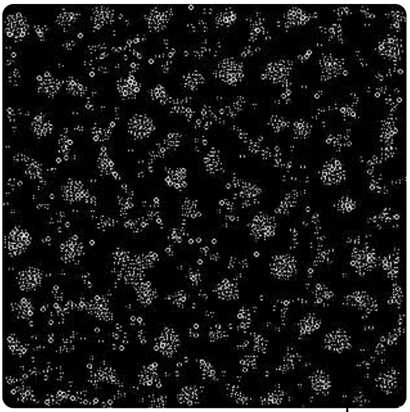
- Recognize common problems through tactics

★ Activity

① Ellipse

② Clustering
- no overlap
- limited qty
- good spacing

How was this made?



Think about the tactic
↳ putting dots on a square

- Diffᵗ tactics : PLACEMENT

① Random placement
② Grid placement
③ Noise placement
④ Proximity Cell Placement
⑤ Stamp Placement

- Diff^t tactics : DISPLACEMENT

⑥ Random Displacement
⑦ Noise Displacement
⑧ Relaxation Displacement
⑨ Noise Culling

- Noise tends to be center biased.
  ↳ So, we see noise patterns clustred to the edges.

- Assignments
  ① Recreate 2 sketches on the website : Req^d challenge
    ↳ + Put your spin on it — 2 more sketches
  ② Do all in-class challenges.
  ③ Read Ch-1 from PGC.
  ④ Create a physical object from code
    ↳ 3D printing, laser cutting, CNC mills

01 Mar 2019

§ Class Discussion

- Choose a work that combines computational & manually-created form.
  - Scratches by Janice & Calligraphic Brush Stroke by Brad

—— end of foundation unit ——

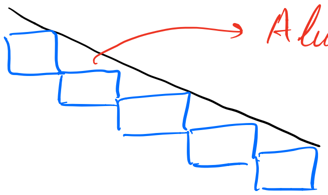
☆ Today's topic : PIXEL DATA

↳ get() & set() is really slow

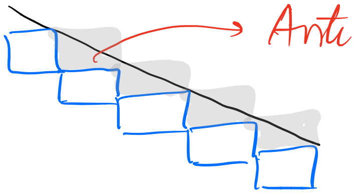
↳ So, Pixel[] array is used

Next few weeks — understanding vectors, rasters, etc.

- When we create an ellipse on canvas, we are creating a vector shape on a pixel grid.
  ↳ This process is called Rasterization.

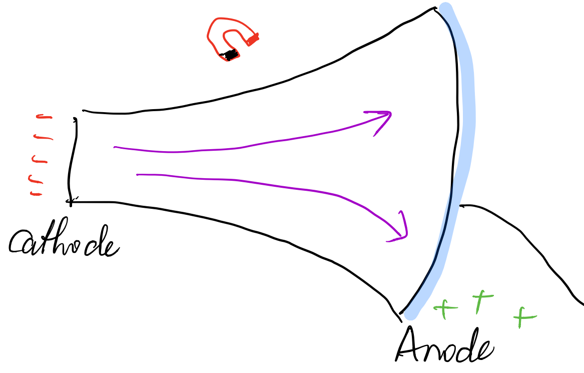
- 

  Aliasing effect : trying to make a continuous signal discrete.

- 

  Anti-aliasing : creating a smoother version of a continuous line by adding details

✱ About Cathode Ray Tube



- electrons going from cathode to anode
- Magnet directs the region where e⁻ hits the anode.

→ Mg layer that lights up

- Tearing Effect in Video Games
  - ↳ Refreshing ½ of the screen with new values from RAM & the bottom ½ with old values.

- Documentary : Racing the beam
  - ↳ Discusses CRT & other displays

- In JS, if we draw outside the size of canvas, it would extend the array. That's a bug.

— ∞ —

✱ Using pixels [ ] array

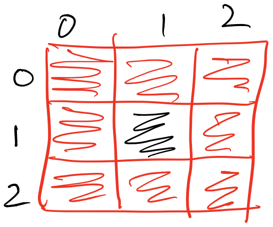
↳ ◦ get() reads every pixel everytime — very slow

↳ eg : a 1920×1080 img will have 4 trillion pixels to process — browser would crash

- Using a f^n for bypassing the safety feature of get()

↳
```
function getQuick (img, x, y) {
    var i = (y * img.width + x) * 4;
    return [
        testImage.pixels[i],
        testImage.pixels[i],
        testImage.pixels[i],
        testImage.pixels[i],
    ];
}
```

logic



The pixel we are going to:

(row * width) + col

(1 * 3) + 1 = $\boxed{4}$.

- each color has 4 values (RGBA). So, multiply by 4 to shift through each pixel.

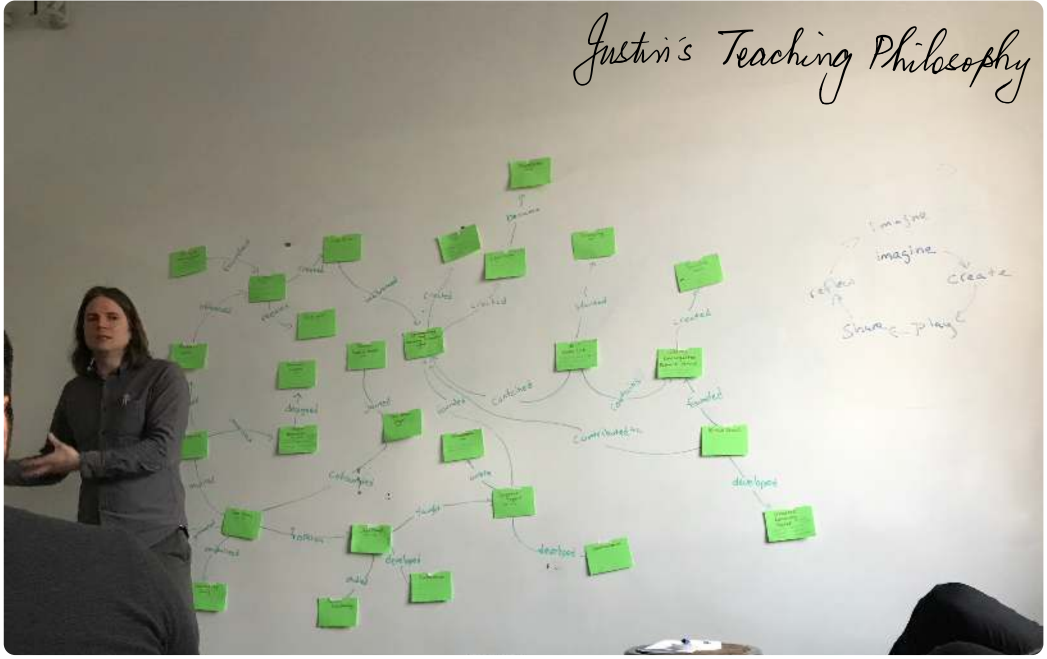
↳ ⇒ ((row * w) + col) * 4

✲ ASSIGNMENT

- Create a sketch entirely from pixel mapping.
- Create a sketch by loading an image & make sth out of it.
- CHALLENGE: create a generational image (see website)

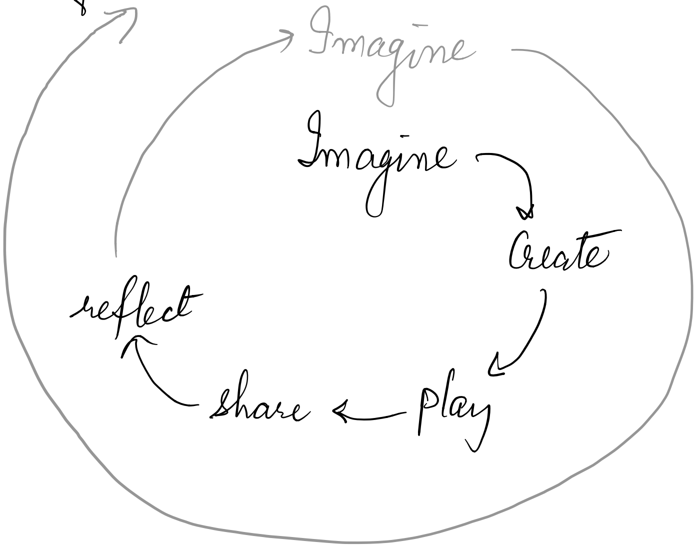
08 Mar 2019

☆ Mindmap Filling Activity



Justin's Teaching Philosophy

§ Creative Learning Spiral



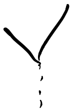
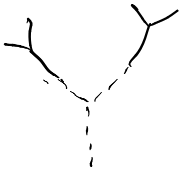
☆ Today's topic : TURTLE GRAPHICS

- Museum of Computer History – Seattle
  ↳ Code on old computers

- Drawing a line in p5 vs Turtle graphics

p5 → locating the line is easier

Turtle graphics → the length of the line is explicitly in the code.

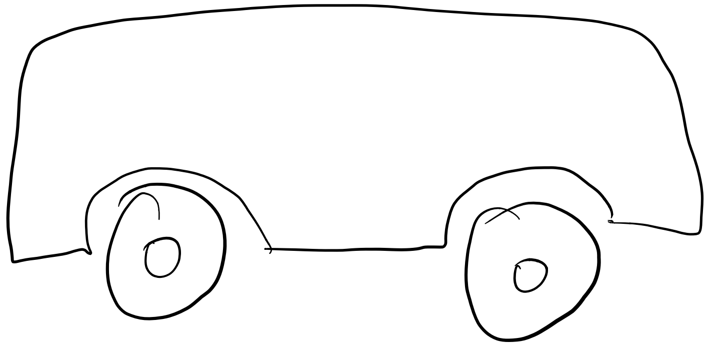
- Drawing a tree in turtle :
  - Draw a line |
  - Draw a branch over it 
  - Draw a tree on each of these branches 
  - Keep doing that.

15 Mar 2019

☆ Today's group : Noah

- → Choose a work that filters an image : Noah's Z-space
- → Choose your favorite turtle challenge response : Anna's Jellyfish sketch

☆ Activity : Create a car using felt



- Benefits of felt : Dimensions | easy to sketch
  vs paper sketch : More abstract | realistic

- Disadvantages : Pasting objects is hard | limited to 2D

- What qualities emerge? : Assembly
  Easy to do as a team

- How would you animate? : Stop motion | Flipbook

§ Today's topic : VECTOR GRAPHICS

↳ Using Paper.js

- SVG format : Using XML script to draw shapes
- We don't use vector displays. So, we have to rasterize the image to show on the screen.
  - ↳ Vector to raster — easy
  - ↳ Raster to vector — not perfect

✲ This week : HYBRID WORKFLOW

Use paper.js to create an SVG.

Then, load that file in Ai & create sketch with it.

✲ Paper.js

- ↳ Developed by Jürg Lehni, once a faculty at Parsons
  - ↳ Created Hektor, a spraypainting robot
- ↳ Really good for procedural vector graphics.
- ↳ Useful for making a drawing tool s/w.
- ↳ Check the website paperjs.org for details

- Conceptual shift b/w paper.js & p5
  - ↳ Fill can be written "AFTER" drawing the object — paper
    - ↳ Deferred rendering ≡ Document Object Model

⋮

↳ Using felt ≡ Paper.js ; Pen + Paper ≡ p5.js

- Let & const doesn't work in paperscript. It only uses var.
- I cannot draw an object. I just create it & then the whole scene is drawn in the end.
- Great for creating stuff for laser cutting .

☆ Assignments — Coding Challenges

① Create a poster

② Laser cut it

29 Mar 2019

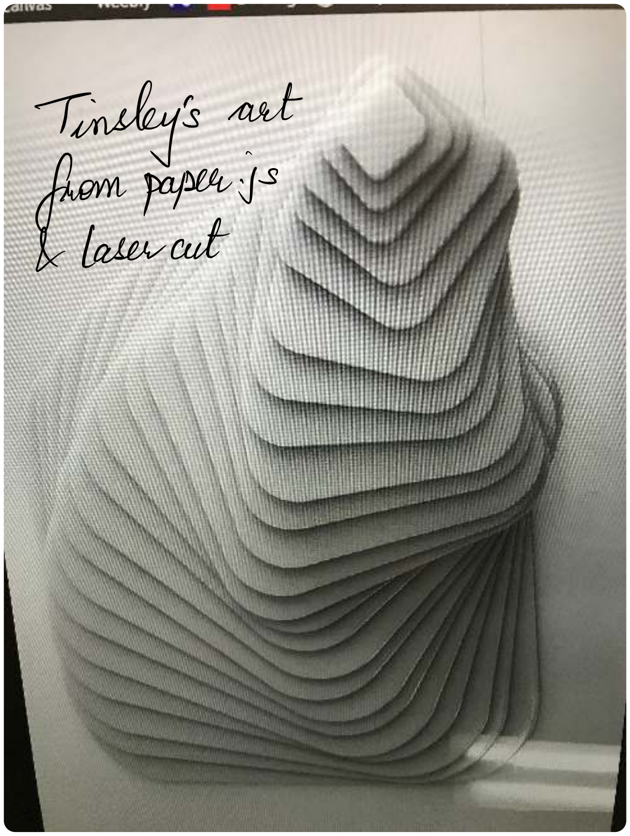
☆ Paper.js vs other tools

- Integration with other tools
- Scalable – vecter
- Clean – graphically → shadows, boolean oper^{ns} on shapes.
- Customizability
- Method - diff^{t}

① Hokusai waves, Comp Form 3D
② Anna's 4 layer piece
③ Stepping Stone Pyramid Gradient : Tinsley

- Risograph machine : A quick way to do screenprinting
  ↳ A screen-printer compressed in an X-ray machine.

- Framing : Actions you do to suggest how you expect them to see your work



Tinsley's art from paper.js & laser cut

Today's topic : Animation

- → Use anything to animate : Unity, Maya, C4D, AE, ...
- → Try to sync with music

- Smear frames : Showing several frames at once to highlight the impact of movement.
  eg: Done in Disney Cartoons

- Laika : A company that does Stop Motion Animation using 3D models.

- Playdead : A game – advanced & better version of Limbo.
- For VR, 60 fps is not enough. Min. used : 90 fps.
  - ↳ convention.
  - ↳ low framerate, lag, etc – reasons for getting nausea.
- Types of animation : pre-rendered vs real-time
  - ↳ very little time to draw frames

★ Game Loop vs Draw loop
↳ Simplified form of Game loop

- When drawing a frame takes too long, s/w drops the frame & goes behind schedule.
  - ↳ If it shows the frame immediately as it buffers, we may get "TEARING" on the screen.
  - ↳ Frame Count in p5.js ⟶ don't use to sync with music
    - ↳ ∵ hardware refresh rate is different
    - ↳ So, use millis() for real-time animation.
    - ↳ For pre-rendered animation, frameCount works best.
      - ↳ caring about the frame I'm making

★ Export pre-rendered animation this week.

☆ Map function

Map is a great function to map "time" to position, size, color, etc.

☆ Sine – for smooth motion.

☆ Modulo (%) function – great to limit the millis ( ) to a particular value.

☆ FFmpeg – a command line utility to export image sequences

☆ Weekly Challenge: Create a CompForm Bumper
↳ Refer MTV Bumper
↳ Details on the website

05 Apr. 2019

✯ Today's group : Noah & Felix

§ Today's topic : GENERATING TEXT

✯ Use RiTa text library
(previously done for Word map)

✯ Read the paper written by Alan Turing :
Computing Machinery & Intelligence → compform.net/text/

- Turing Test : was called The Imitation Game by Alan Turing
- ABC & JAM formats : Text formats to generate music .
- Computer programs are made out of text
  ↳ ⇒ you can create a generative text to create a program that runs a program .
- Quine : a program that generates a copy of text .

✯ Pentametron : a pair of words that are rhyming
↳ 10 syllables (usually)

↳ Shakespearean style of iambic writing .

- Memes → (were called) Image memes → (were called) cat macros

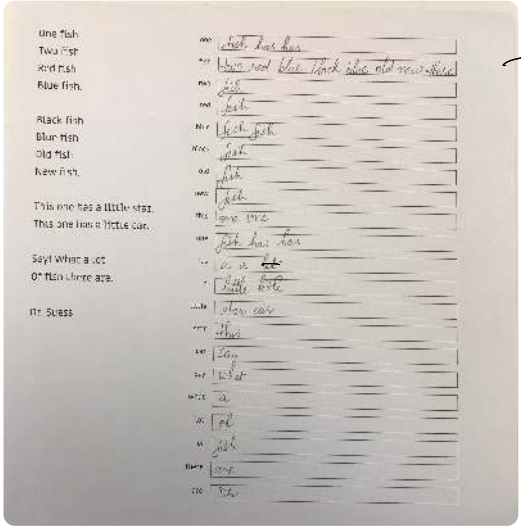
- NaNoGenMo 2018 : creating novels through code.

—x—

- Using Javascript template literals using back-ticks :
  `I am a ${noun}!`;

✬ MARKOV CHAINS

↳ models that can be applied to any series of symbols



One fish
Two fish
Red fish
Blue fish.

Black fish
Blue fish
Old fish
New fish.

This one has a little star.
This one has a little car.

Say! What a lot
Of fish there are.

Dr. Suess

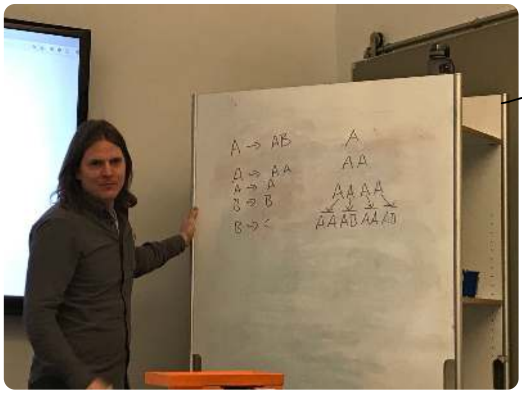
one	fish has has
fish	two red blue black blue old new there
two	fish
red	fish
blue	fish fish
black	fish
old	fish
new	fish
this	one one
one	fish has has
has	a a
a	little little
little	star car
star	this
car	say
say	what
what	a
lot	of
of	fish
there	one
are	[illegible]

→ start with any word.
↓
the next word random
↓
use this word to take the next word in the sequence.

eg : one fish
↳ this one
↳ has a

✬ CONTEXT FREE GRAMMAR

- Not all text strings are valid HTML.
  ↳ eg : <div><b>Hi there!</div></b>
  ↳ Invalid



→ substitution grammar

↳ In context free → does not check for other symbols. Only looks at the symbol it is modifying.

- Tracery – text modifier js language
  ↳ #sth → which has rules of change.

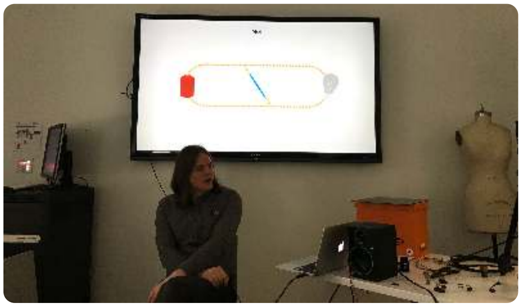
★ Challenge
↳ Create a bad short story using computation. → details on compform.net/text/

- Twine – graphical engine for making – choose your adventure text stories.

12 Apr. 2019

§ This week's topic: Sound

☆ Act 1 : Carefully arranged Sound & lightning



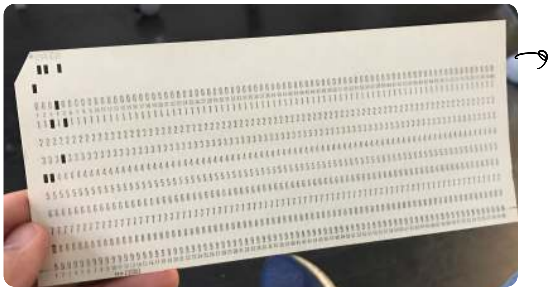
- $3000 Apple 2C : 11,500 transistors, 1.023 MHz
- $30 Arduino Uno : 100,000+ transistors, 16 MHz
- Mac Pro : 1B+ Transistors, 2,3,4 GHz

☆ Act 2 : A Universal Lever With which to Move The World

- Computer should have 3 parts :
  ① Operations ② Memory ③ Input & Output

  Input → Instructions & Data

  Output → Data



→ Punch cards.

- ElectroBoom : ESD (Electrostatic Discharge)

☆ Act 3 : Light

- Richard Feynman's YouTube video on light
- UV Imaging : How the Sun Sees You

- An LED produces electromagnetic radiation at very specific wavelengths.
  ↳ ⇒ A blue LED is not emitting white light.

→ The controversy
↳ How our eyes perceive colors.

→ IR Camera

✯ Act 4 : Sound

- Video about Hearing & How it Works | Med-El.

Video about Hearing and How It Works | MED-EL

- We are better at perceiving freq with audio

IMAGE VS SOUND CODES

✯ Similarity

- Outside Source
- Var declarⁿ reqᵈ
- Loading media code is similar
- Both have setup & preload
- Both written in JS

Differences

- Positioning not reqᵈ.
- Time component only for audio
- Diffᵗ file extensions
- loop fⁿ for audio, frame rate for image
- Procedural vs Object Oriented Style Programming

Sound

Image

- Change playback rate
  - ↳ slower : lower pitch
  - ↳ faster : Higher pitch
- Mod function to create audio drum beats
  - ↳

```
if(frameCount % 15 === 0){
    hat.play();
}
if(frameCount % 60 === 0){
    drum.play();
}
```

- p5 has audio recording feature
- FFT (Fast Fourier Transform)
  - ↳ Converts Time Input from sound & convert in frequency
- For creating music, tone.js is better than p5 sound.

This week's challenges :

1. Choose a 15 sec video clip & create a sound for that — eg: Charlie Chaplin
2. Choose a 15 sec audio clip & create a visual for that sound — use FFT.
3. Create both video & audio with code.

19 Apr. 2019

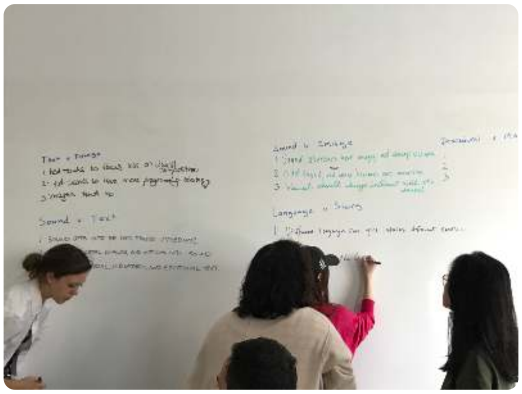
§ Today's group : Noah & Anna

✲ Activity : View work from last weeks
↳ compare it & write 3 insights (Sound vs image)
↳ Phrase in 7 words

✓ ① Sound + Img = more immersive → Add sound & img becomes more immersive

✓ ② Sound has to sync with visual

✓ ③ Sound needs visuals → Visual → should always Interact With the Sound



§ Today's topic : MUSIC (using tone.js)
↳ Different aspects of creating computational music :

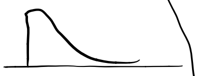
① Algorithmic music

② Aleatoric music → how the composition will sound like is not pre-decided.

③ Generative music

Reference :
① Six Marimbas – Steve Reich → There's a structure to the music creation
② John Zorn : Cobra

✪ Activity : Compare 2 musical instruments
↳ Piano & Clarinet
↳ Sound



→ More variation

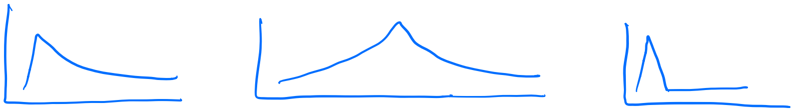
→ flexible

→ More windy feeling in clarinet

wave pattern (sine, square, triangle, sawtooth)
(Oscillator)

- Timbre : a term used to differentiate sound from 1 instrument to that from another
(anything that isn't pitch or loudness)

- Other qualities to judge the sound : Attack (A), Decay (D), Sustain (S) & Release (R) — ADSR
(Envelope)



→ Shape of a sound

- Additive Synthesis : Adding waves to get a complex sound
↳ Other terms : Subtractive synthesis, Wavetable synthesis, Sampled Synthesis, FM & AM Synthesis

- Code : synth.toMaster( ); → to listen to a sound
- Speaker plays at 44,100 Hz
- In JS, we create an oscillator & have that fill the buffer

  makeSound (255)

  [- - - - - -]

  ↳ 255 values in the future

☆ Activity : What would you need in JS to create a melody?

↳ how would you represent data?

Note
- → pitch
- → dur^n
- → loudness
- → rest time

Sequence : Array of null objects

+ Arrays of arrays

Verses

↑

Phrase

↑

Note

↑

Parameters

- Constraints :

  Scale, major, minor,

- Playing with music : Use MIDI (easy math)

  → 30, 31, 60, ---

  Use Scientific Pitch Notation (Supported by tone.js)

  → C3 D♭4 E#4 ....

- Const melody = [

  ["E4", "4n."], ["D4", "8n"], ... ["rest", "1m"]];

  → Quarter note + 1/2 × (Quarter note)

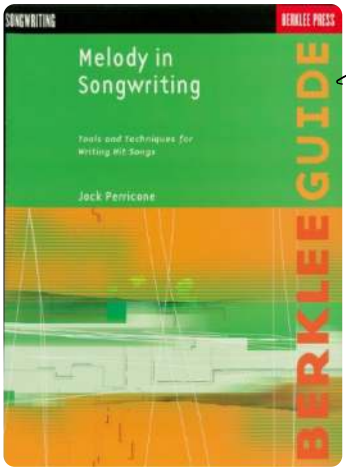
✯ Activity : How to create Sound machines ?

1. Composition structure (picking random notes & filtering it to a set of keys)

12 notes → Main array

↳ 7 note → Subset of notes

2. Choose a note using a noise f^n instead of random
3. Markov Chain : tight association b/w states – having a structure



SONGWRITING

BERKLEE PRESS

Melody in Songwriting

Tools and Techniques for Writing Hit Songs

Jack Perricone

BERKLEE GUIDE

→ Guide to create a melody

↳ Planning the target characteristics

1. Use C-major
2. Use 4/4 time
3. Only ½ & ¼ notes
4. No rests

— — — — —

- Term : "Degree" : Position in a scale : 0, 1, 2, – –
- Concat f^n → combine & flatten melodies from difft arrays into one.

return [ ].concat(a, b, a, c);

✯ Challenge : Record & make a song

26 Apr. 2019

☆ Using OpenSCAD.

↳ Constructive modeling (not done in Maya)
(More of Fusion, CAD)

→ Create code that generates a model
↳ Diff^{t} way to construct 3D models

- Activity: Discuss a challenge response which doesn't use code.
  ↳ Andrew Levinson's Midi Music Piece

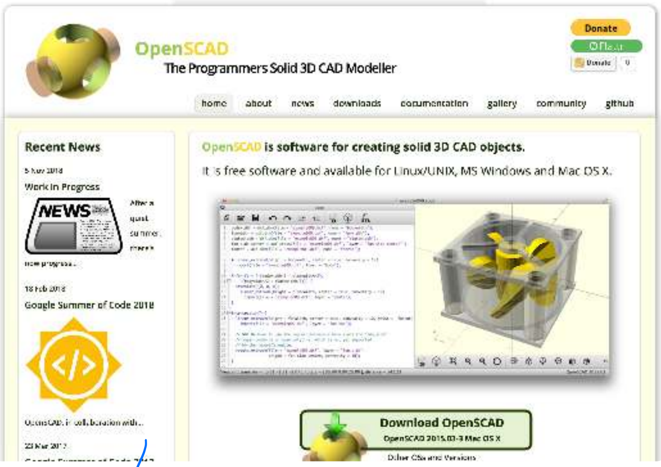
- Movie reference: Rosencrantz & Guildenstern are Dead

- Great project: Build a 3D printer from scratch.
  ↳ You learn about axis.

☆ Assignment ideas — 3D

① Major Major 3D form
② CNC Cut (Major Major)
③ Autodesk 3D Character: Mixamo
④ Saransh's project – human molecules
⑤ VR–Unity 3D setup
⑥ Sketches in OpenSCAD

- OpenSCAD: Using constructive solid geometry
  ↳ with Boolean operations to create complex shapes.



Recent News

5 Nov 2018

Work in Progress

After a quiet summer, there's new progress...

18 Feb 2018

Google Summer of Code 2018

OpenSCAD, in collaboration with...

23 Mar 2017

Google Summer of Code 2017

OpenSCAD is software for creating solid 3D CAD objects.

It is free software and available for Linux/UNIX, MS Windows and Mac OS X.

↳ OpenSCAD.org

OpenSCAD is its own language.

If you want JS version of OpenSCAD

↳ OpenSCAD.net

- OpenSCAD is not like C [C language family]
- Terms : Difference – operator
  cylinder – action

✗ PROGRAMMING PARADIGMS

- Functional vs Procedural programming

Functional → how you are organizing your thoughts in a program

Organize logic using functions

Procedural → Organize steps into procedures

- Imperative vs Declarative Programming

Imperative → Steps req^{d} to achieve a goal (eg: JS)

Declarative → detailing out the result than the steps to do.

→ Specific^{n} of the result (eg: OpenSCAD) (eg: SQL)

- Object Oriented Programming
  - → Groups related data & procedures using objects
- Logic programming
  - → stating a set of rules – bunch of statements
  - eg: constraint solving
- Immutable vs Mutable Data
  - eg OpenSCAD
  - → variable values assigned last get that value.
- Prefix vs Infix Notation
  - Prefix → Operator comes first
    - $* 2 \ 3 \rightarrow 6$
  - Infix → Operator in b/w operands
    - $2 * 3 \rightarrow 6$
- Named vs Positional Parameters
  - Positional → cylinder(4, 1, true)
  - Named → cylinder({h: 4, r: 1, center: true});

—∞—

QUIZ

1. Programming paradigm: F^{nal}, procedural, object oriented, logic, imperative, declarative
2. Paradigm for JS: f^{nal}, Imperative, Procedural, OOP, Declarative, logic
3. Paradigm for OpenSCAD: Declarative, pr~~oce~~dural, functional
4. Infix notation
5. Prefix notation

⑥ * 10 2 in prefix notation

⑦ Giving instructions — S1, S2 , recipe

⑧ Immutable : Last var value

: Cannot change the value

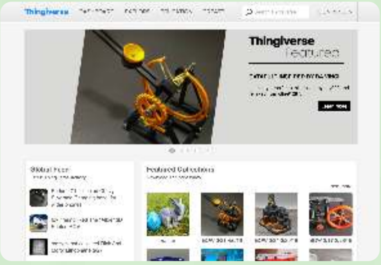
— ∞ —

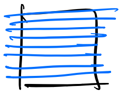
§ OpenSCAD — Notes

- OpenSCAD : Use "%" to render an object as a ghost
- $fn, $fa, $fs : Control no. of edges
- # → make an object red : Useful for selection
- echo → print / console log
- module in OpenSCAD ≡ function in JS.
- for loop written differently
  - ↳ for loop for any no. of variables

☆ Parametric OpenSCAD

☆ Reference for downloading a 3D model for printing



- 3D printer works on Slicer  & G-code → coding for 3D printing

☆ Houdini : Node based 3D modeling program for VFX.

↳ Industry program

FULLY PROCEDURAL

[illegible]

☆ This week's challenge : Build a castle.

03 May 2019

§ Today's group : Noah & Me (Table 3)

☒ ACTIVITY

3D vs Music

3D sketches are visual experiences not auditory

Visual

3D Visual ✓

Auditory

Spatial sound absent ✗ → R&L → Mono.

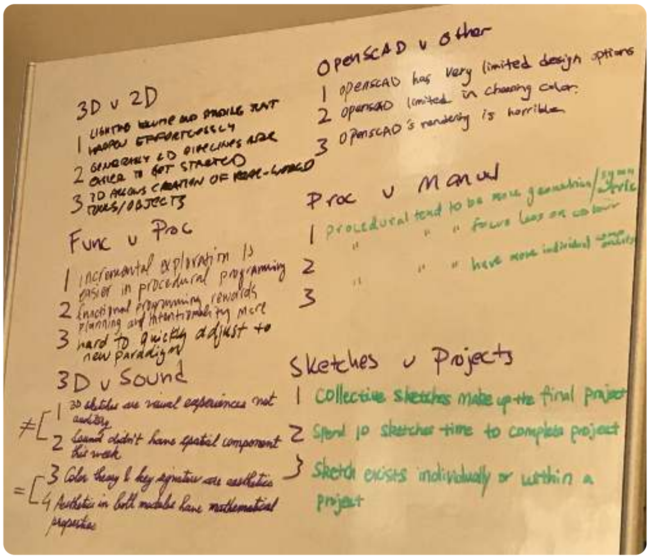
Music didn't have spatial component this week

Similarity

Color theory → key signature

Colors theory & key signature are aesthetics

Aesthetics in both modules have mathematical properties



3D v 2D

1 LIGHTING VOLUME AND SHADING THAT HAPPEN EFFORTLESSLY

2 GENERALLY 2D OPTIONS ARE EASIER TO GET STARTED

3 3D ALLOWS CREATION OF REAL-WORLD TOOLS/OBJECTS

Func v Proc

1 incremental exploration is easier in procedural programming

2 functional programming rewards planning and intentionality more

3 hard to quickly adjust to new paradigm

3D v Sound

#[1 3D sketches are visual experiences not auditory

2 Sound didn't have spatial component this week

=[3 Color theory & key signature are aesthetics

4 Aesthetics in both modules have mathematical properties

OpenSCAD v other

1 openSCAD has very limited design options

2 openSCAD limited in choosing color.

3 OpenSCAD's rendering is horrible

Proc v Manual

1 procedural tend to be more geometric/symmetric

2 " " focus less on colour

3 " " have more individual [illegible]

Sketches v Projects

1 Collective sketches make up the final project

2 Spend 10 sketches time to complete project

3 } Sketch exists individually or within a project

§ This Week's Topic : MICROGAMES

- A Minigame : 1-2-Switch → Interesting mini-game Play it & see why it's so weird.

- Wario Ware : example of Microgame.

- Group these interactive artefacts into

Games

Monopoly
Chess
Super Maio Bros
Pac Man
++

Toys

Lego
Dolls
++
Tops

Tools

Photoshop
Hammer
Telephones
++
++

- Brainstorm at least 10 Defining or Characteristic features of games

Defining: must be present eg: birds have wings

Characteristic: commonly true eg: birds can fly

① Parameters / Rules (ii)

② Start & End — Tempo (iii) → Over time

③ Setting (vi)

④ Player(s) (i)

⑤ Win / Loss (vii)

⑥ Interaction (v)

⑦ Diff't aesthetic - suspended norms (viii)

⑧ Outcomes (iv) → ≡ Goals

⑨ Visual (x)

⑩ Levels (ix)

↳ Rank them in the order of importance

Others : Agency/Choice
: Instructions
: Emotional Engagement
: Resources
: Mastery / Ability to improve
: Addiction
: Emergent play
: Obstruction

— × —

◦ Activity : Approaching Learning New Libraries
↳ 8 tactics you used to start learning tools

① Documentⁿ Page (ii)

② Video tutorials (i)

③ Example code (iii)

④ Blog (stackexchange) (iv)

(1b) Programming structure
↳ Procedural vs final

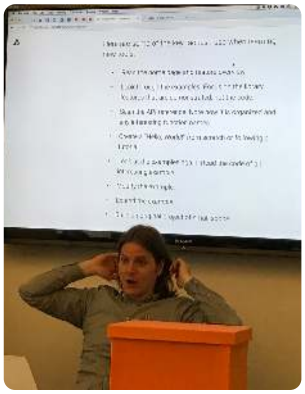
⑤ Looking at the source code for library

⑥ Implementⁿ methods & platforms (v)

⑦ Error reporting (auto) — console

⑧ Self-application

↳ Top 5



—✕—

§ Using p5.play

Example : Kick Ass Bookmarklet

→ More work in setup & less in draw function

✲ Coding Challenge : Couch Co-op

- 2 plays sitting next to each other & competing

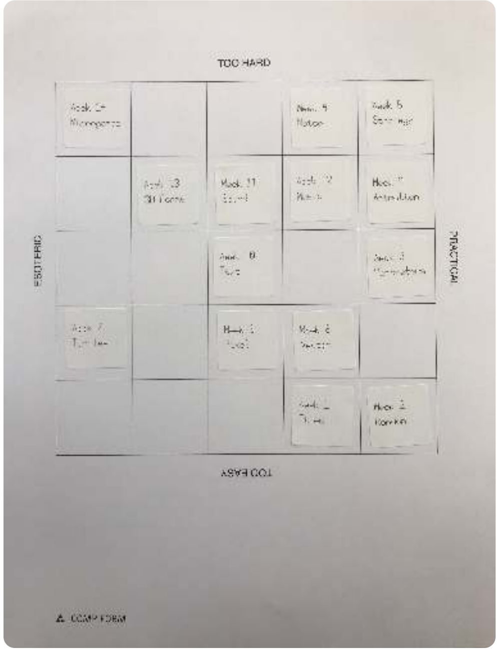
↳ example : Lovers in Spacetime .

✲ Piskel :

Tool to create pixel art .

10 May 2019

✡ About Microgame
- Well documented
- Interactive sketches



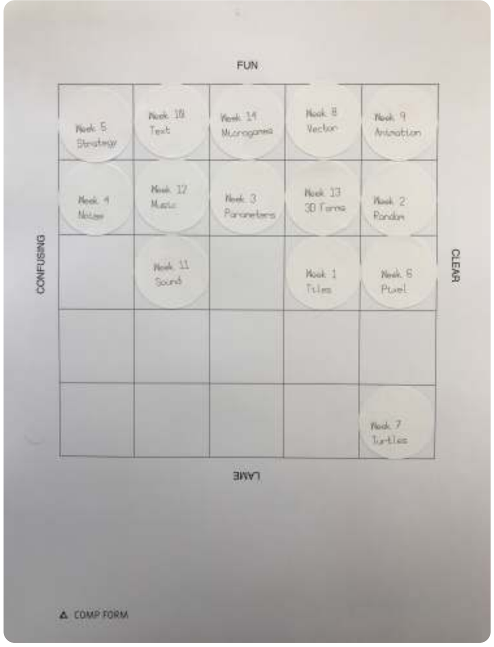
TOO HARD

ESOTERIC / PRACTICAL

Week 14 Microgames			Week 4 Noise	Week 5 Strategy
	Week 13 3D Forms	Week 11 Sound	Week 12 Music	Week 9 Animation
		Week 10 Text		Week 3 Parameters
Week 7 Turtles		Week 6 Pixel	Week 8 Vector	
			Week 1 Tiles	Week 2 Random

TOO EASY

COMP FORM



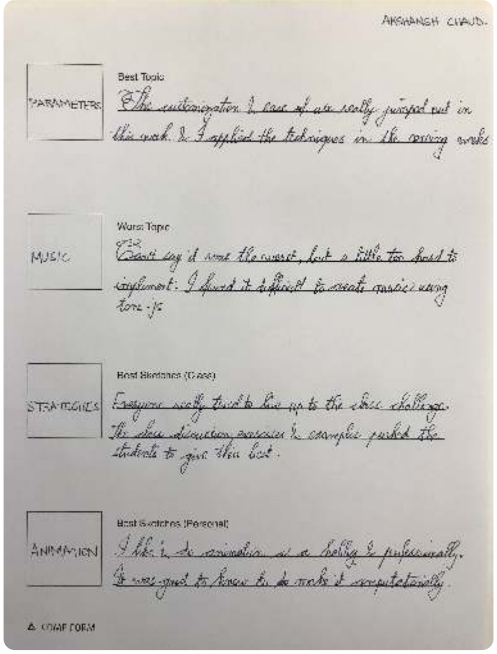
FUN

CONFUSING / CLEAR

Week 5 Strategy	Week 10 Text	Week 14 Microgames	Week 8 Vector	Week 9 Animation
Week 4 Noise	Week 12 Music	Week 3 Parameters	Week 13 3D Forms	Week 2 Random
	Week 11 Sound		Week 1 Tiles	Week 6 Pixel
				Week 7 Turtles

LAME

COMP FORM



AKSHANSH CHAUD.

PARAMETERS

Best Topic

The customization & ease of use really jumped out in this week & I applied the techniques in the coming weeks

MUSIC

Worst Topic

Can't say it was the worst, but a little too hard to implement. I found it difficult to create music using tone.js

STRATEGIES

Best Sketches (Class)

Everyone really tried to live up to the class challenge. The class discussion, exercises & examples pushed the students to give their best.

ANIMATION

Best Sketches (Personal)

I like to do animation as a hobby & professionally. It was great to know to make it computationally.

COMP FORM

☆ Top 3 problems with this class

1. Class Timing/Day
2. Discussion on the mindmap activity
3. Too many sketches during finals
4. No collaboration in sketches
5. No communic^n b/w classes
6. Some topics are too narrow/similar
7. Sketch format too self directed

↳ Solve the problem 1 : No room for deep dive

↳ Time change cannot work : constraint

1. 3 tickets — choose wisely
   ↳ Week with lower sketches → added constraint on no. of elements used.
2. Deep dive allowed for imp. topics - strategies & parameters
3. Make a sketch from some other week for a week that isn't that imp
4. One week on personal expression

end of course