

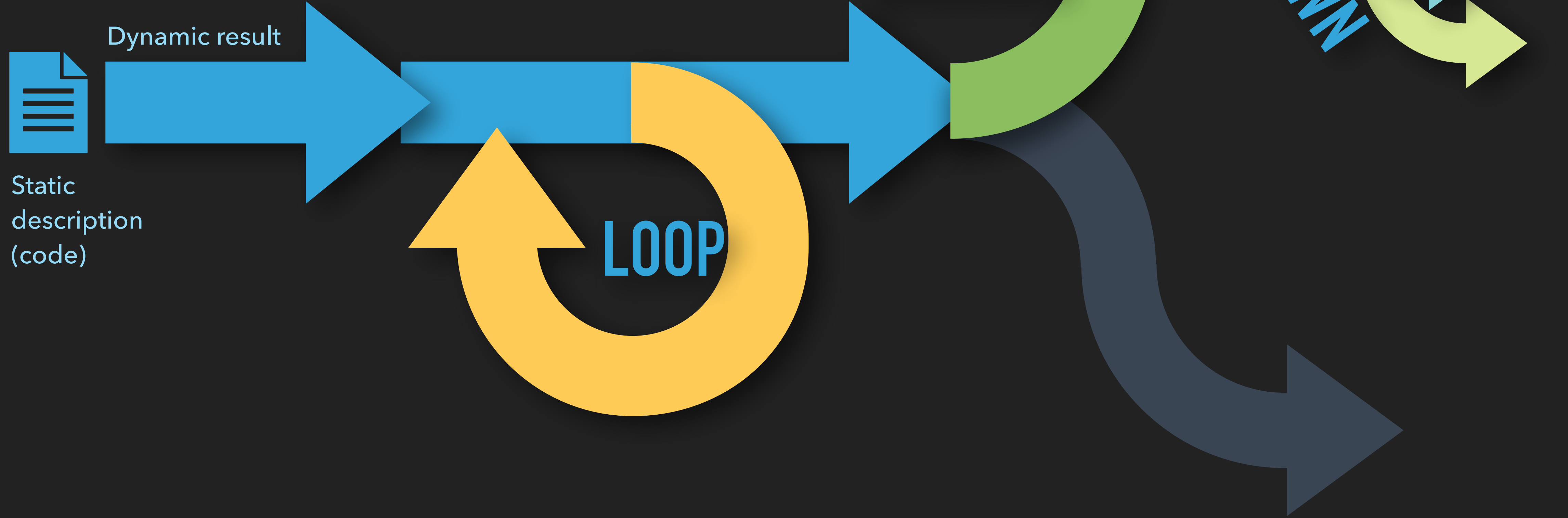
**EXISTING
IN
TIME**





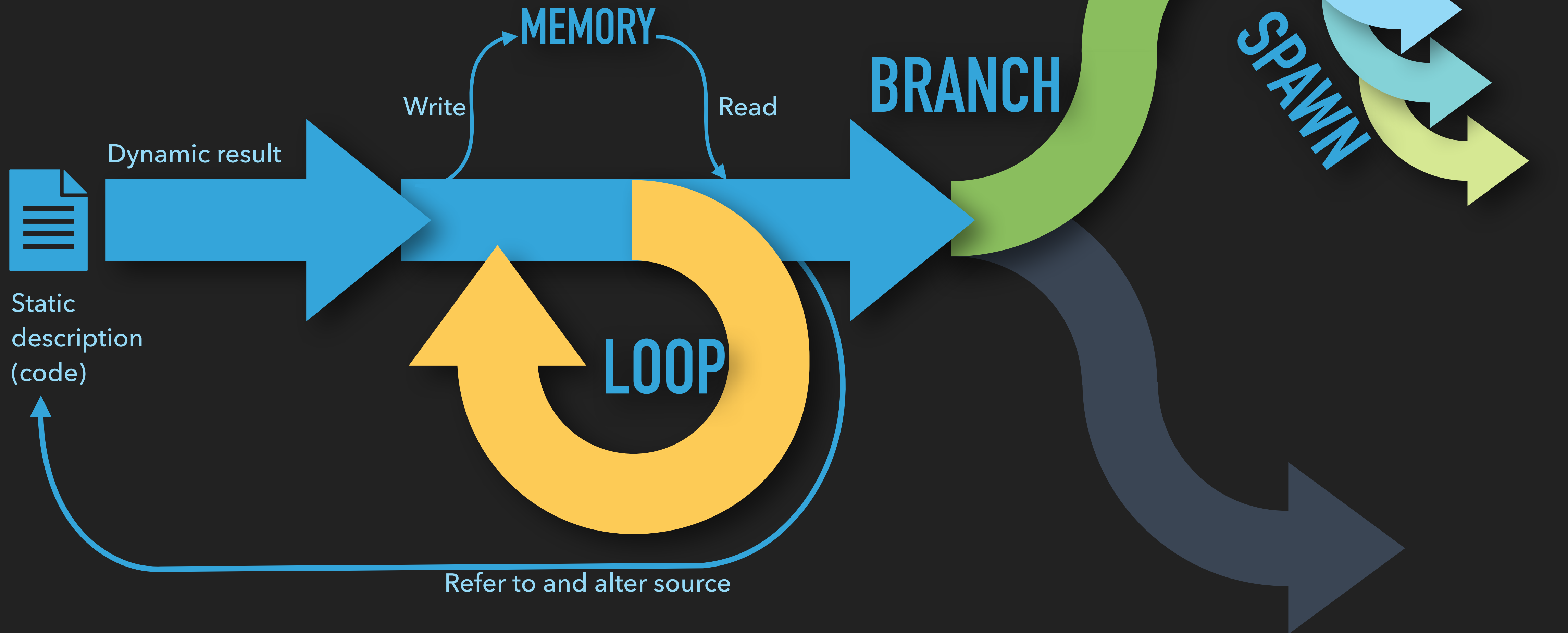
CODE

More situated in time than linear media or static artifacts



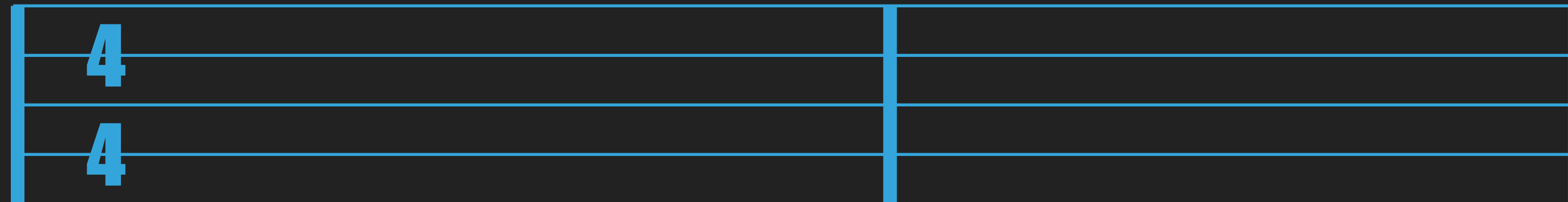
CODE

More situated in time than linear media or static artifacts
Self referential (input and output is same thing)



CODE MUSIC

Situated in time, with loops and branches, but not self-referential



CODE MUSIC

Symbols for representing time

TIME SIGNATURE

1/4 NOTE

1/8 NOTE

The image shows a musical score for a piece titled "No. 16." The score is written in 3/4 time and consists of three systems of music. The first system is in a key signature of one sharp (F#) and features a forte (*f*) dynamic. The second system is in a key signature of one flat (Bb) and features a fortissimo (*ff*) dynamic. The third system is in a key signature of two flats (Bb, Eb) and features a piano (*p*) dynamic. The score includes various musical notations such as treble and bass clefs, time signatures, notes, rests, and dynamic markings. Annotations in blue highlight specific elements: a box around the time signature "3/4" in the first system, a vertical line pointing to a quarter note in the first system, a vertical line pointing to an eighth note in the first system, a horizontal line pointing to a quarter rest in the first system, and a vertical line pointing to a half note in the third system.

1/4 REST

1/2 NOTE

CODE MUSIC

Symbols for structuring execution

LOOP DELIMITER SYMBOL

REPEAT FIRST SECTION ONCE

The image displays a musical score for 'Nº 16.' in 3/4 time, marked *ff*. The score is divided into three sections. The first section is highlighted in blue and contains a loop delimiter symbol (a vertical bar with a double bar line) at its beginning. The second section is highlighted in green and contains a loop delimiter symbol at its end. The third section is also highlighted in green and contains a loop delimiter symbol at its end. The score includes various musical notations such as notes, rests, and dynamic markings (*fz*, *p*, *ff*).

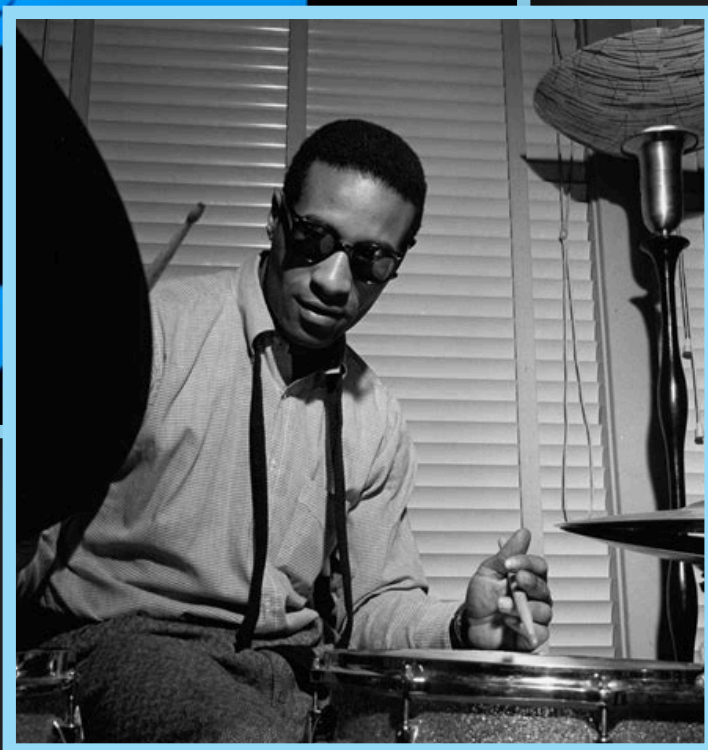
REPEAT SECOND SECTION, USE ALTERNATE ENDING SECOND TIME

CODE MUSIC

Simple outline can be decoded in sophisticated way by specially-trained agents aka 'musicians'



SAXOPHONE COLOSSUS
SONNY ROLLINS



Medium Swinging
Latin

St. Thomas

Sonny Rollins

$\text{♩} = 105$ **A** C^6 E_{MI}^7 A^7 D_{MI}^7 G^7 C^6
 (tenor, 8^{va} b.)

C^6 E_{MI}^7 A^7 D_{MI}^7 G^7 C^6

$E_{MI}^7(b5)$ B^b7 A^7 D_{MI}^7 $A^b7(\#5)$ G^7

C^7 C^9/E F^6 $F^{\#o7}$ C^6/G G^7 C^6

B (Solos) C^6 A^7 D_{MI}^7 G^7 C^6 (fine)

C^6 A^7 D_{MI}^7 G^7 C^6

$E_{MI}^7(b5)$ A^7 D_{MI}^7 G^7

C^7 C^9/E F^6 $F^{\#o7}$ C^6/G G^7 C^6

Solos may swing.

After solos, D.C. al fine.
Head is played twice before & after solos.



dedicated to
Keith Cary and Robert Roux

Music with Timing Devices*

for any number of players**

1.
2.
3.
4.
5.

(quasi-Modal) (quasi-Modal/atonal)

Tempo changes at arrows - Tempo constant within arrows

gradual dynamic changes. Utilize indicated spectrum

abrupt dynamic changes. Utilize indicated spectrum

dim. with last grains of sand

abrupt stop with sand

Go out with a abrupt stop

*Timing devices = 3 min. hourglass-type egg timers (out of 14 timing devices tested, 200 percent were inaccurate. Considered desirable.)
observed.

Reed Maxson

Davis
April
1974

** If more than one player participates, each player may play a different line, or some combination of this arrangement may be realized. With over five players, some or all lines will be necessarily doubled, tripled, etc. Any number of timing devices, up to not more than one device per player, may be used. Devices may be started together or in series at five to twenty-five second intervals. Upon reaching the second inversion point, players may jump to corresponding point in another line. Read left to right.

Krzysztof Penderecki: *Threnody for the Victims of Hiroshima* (1960)

16

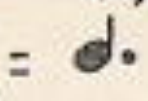
1-12
12Vn
13-24
12Vn
1-10
10VI
1-10
10Vc
1-4
8Cb

18'' 20''

* flageolet tones

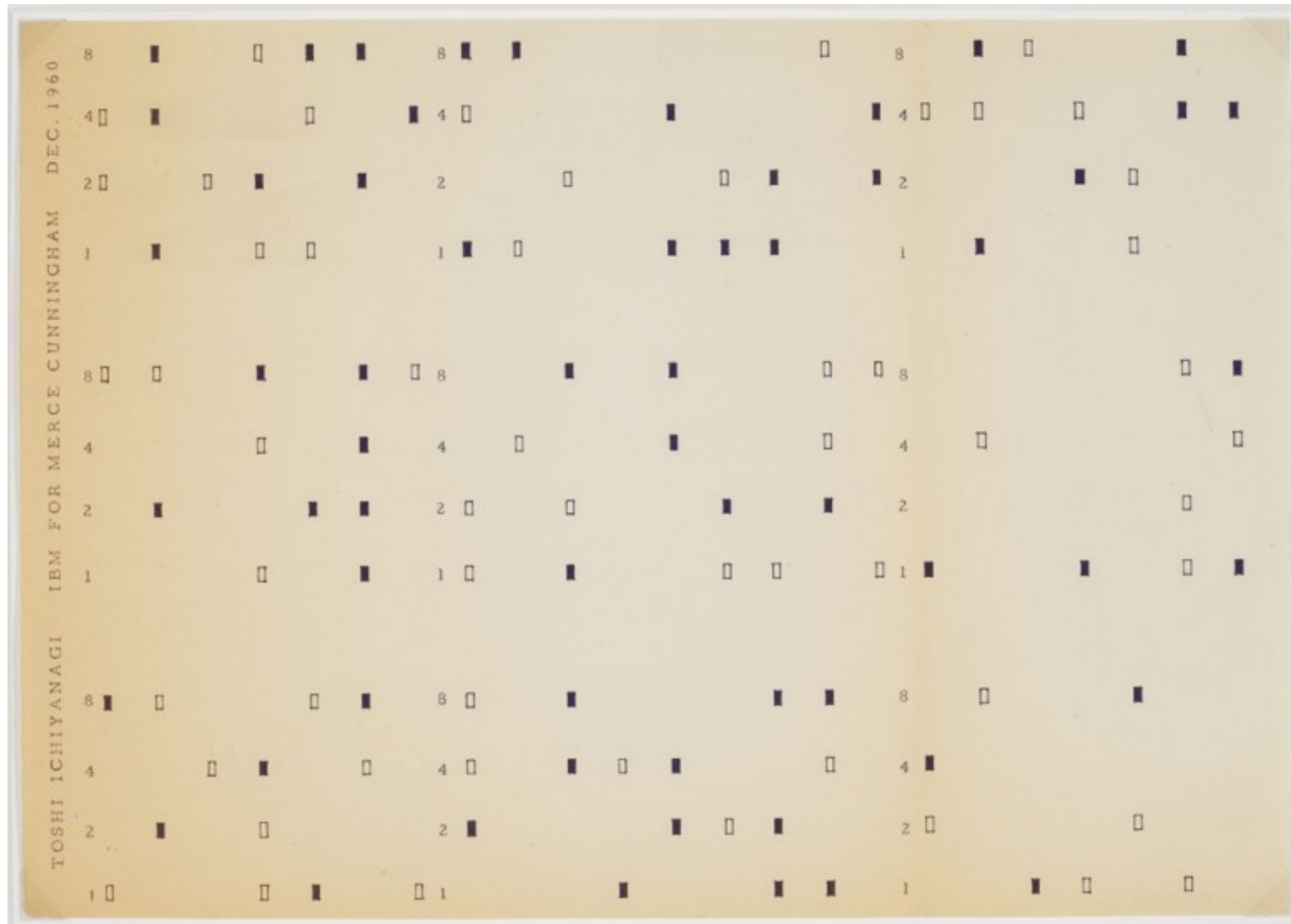


Castagnettes.

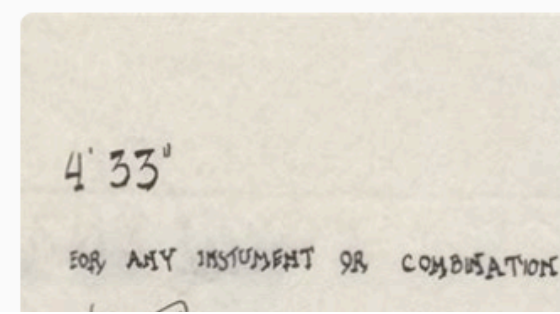
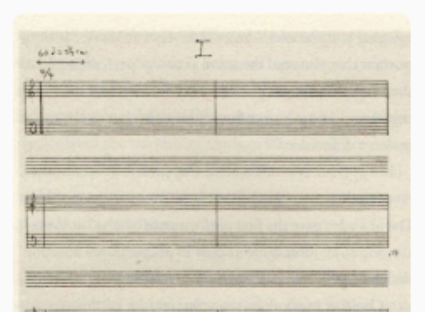
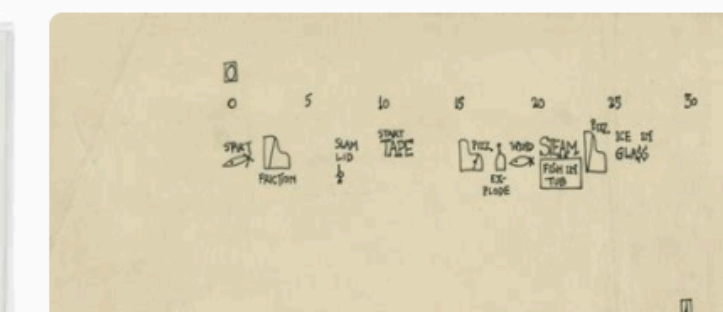
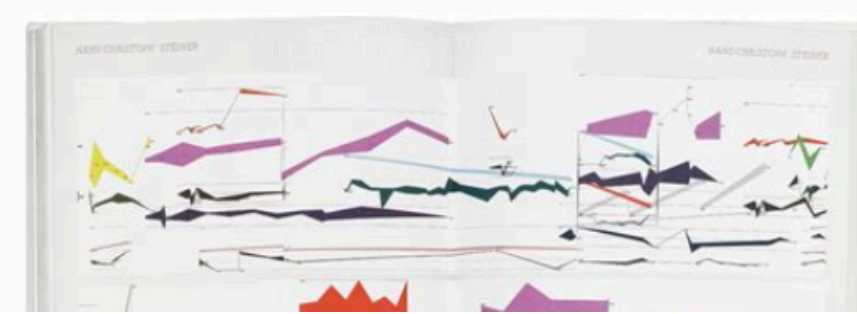
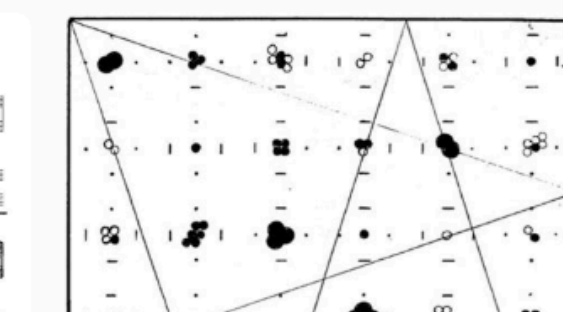
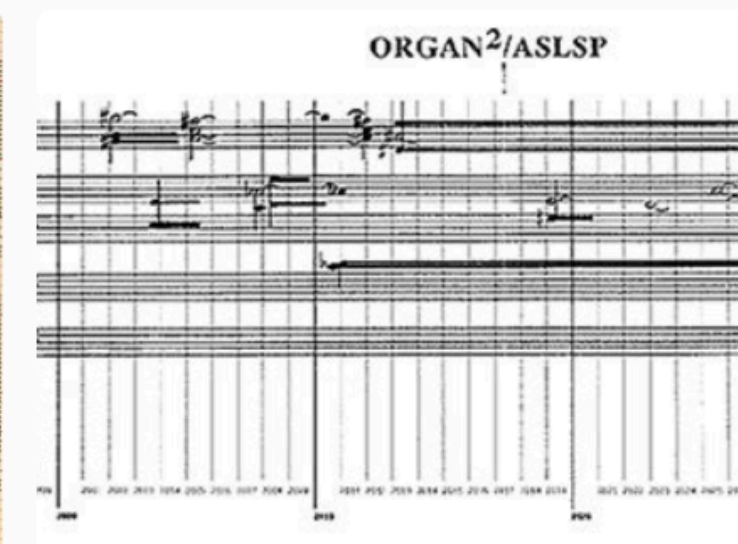
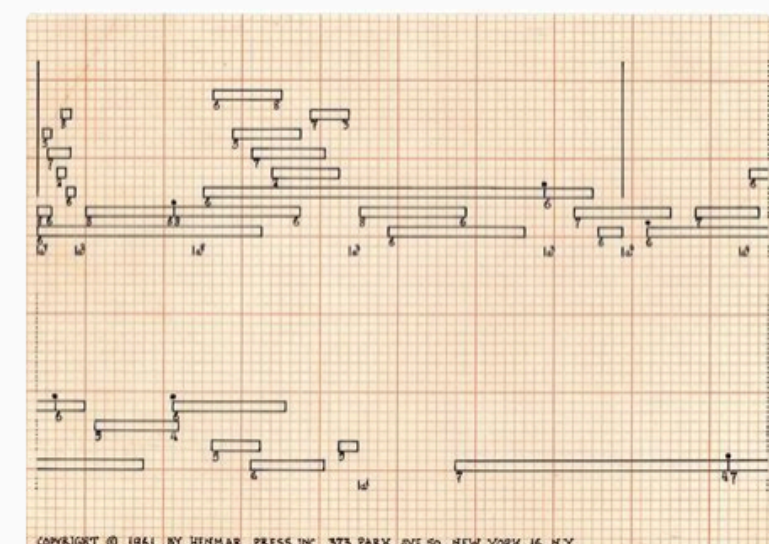
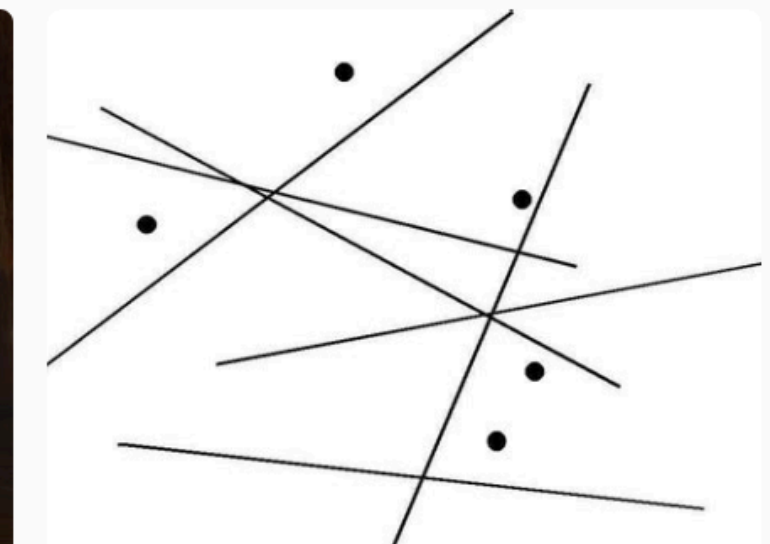
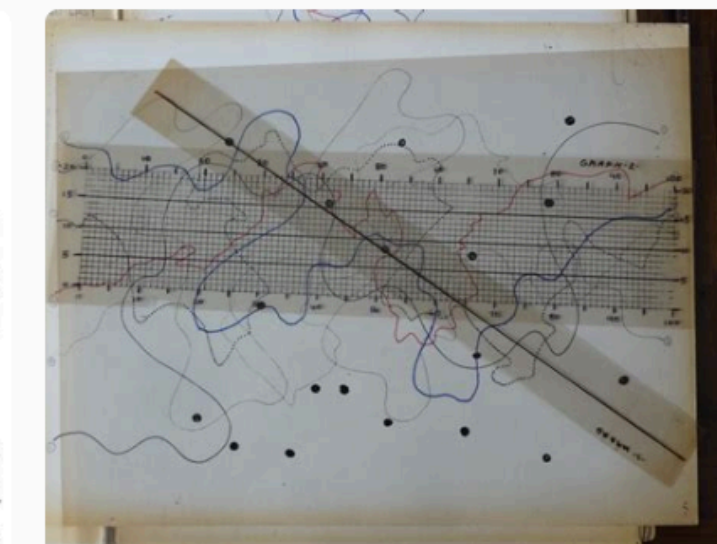
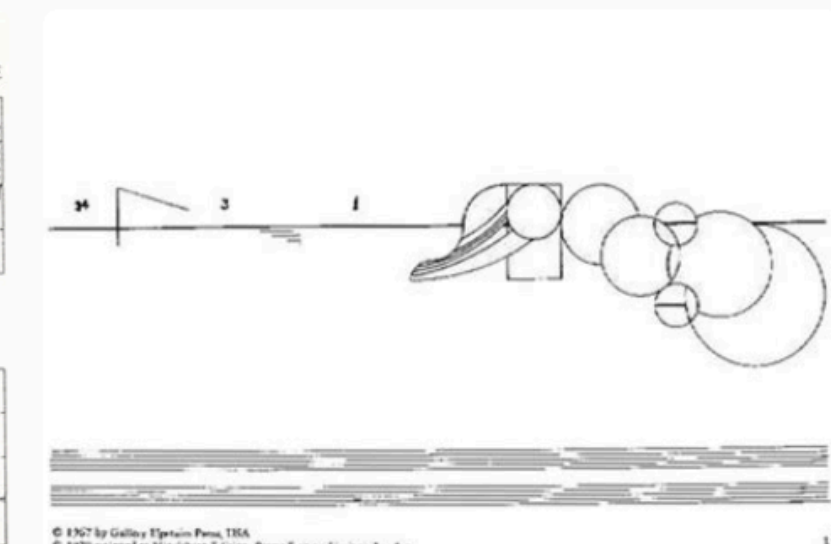
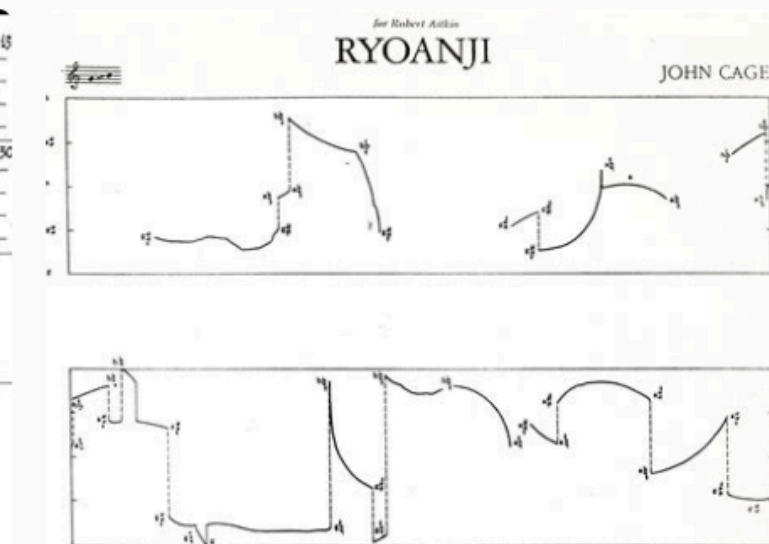
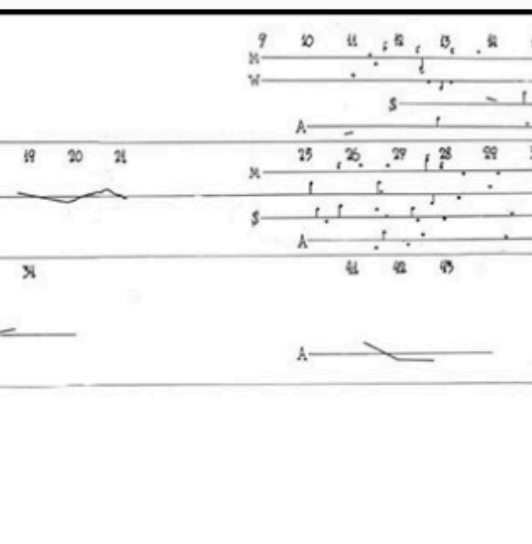
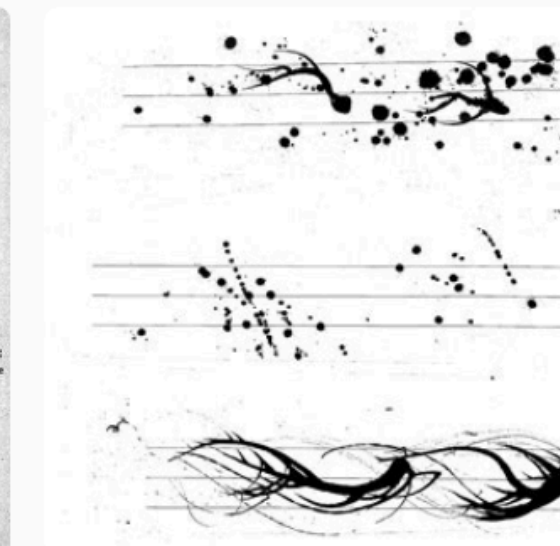
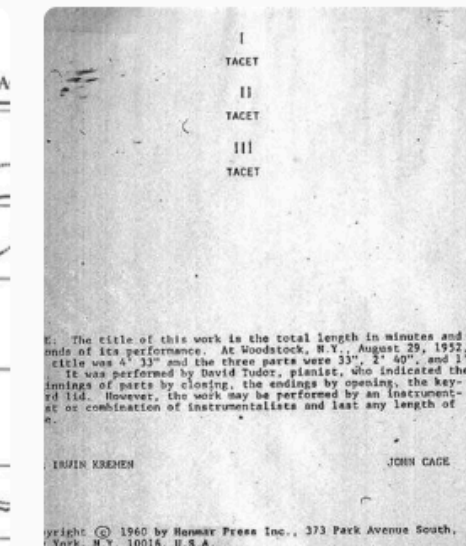
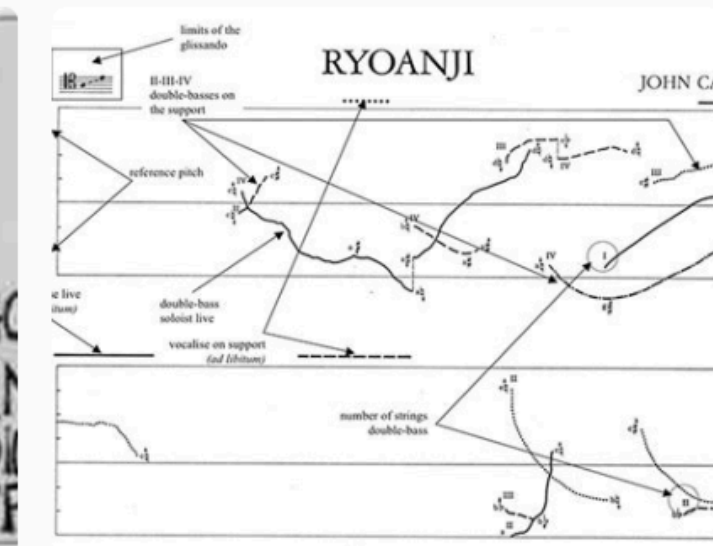
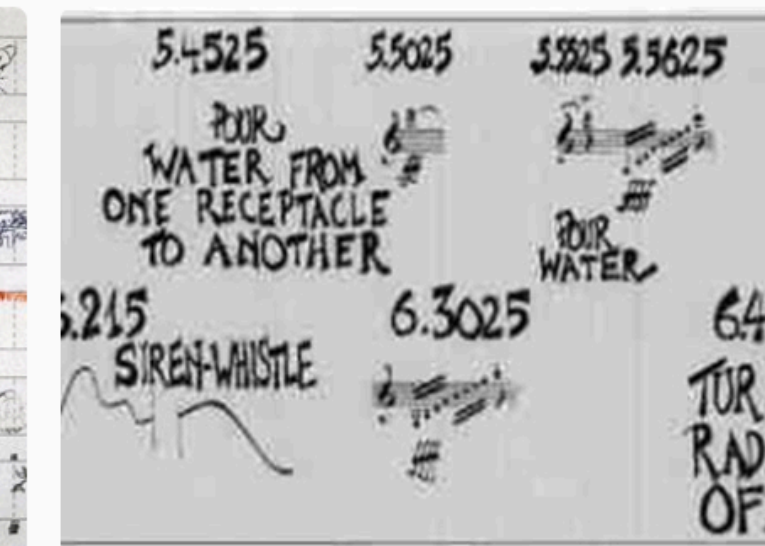
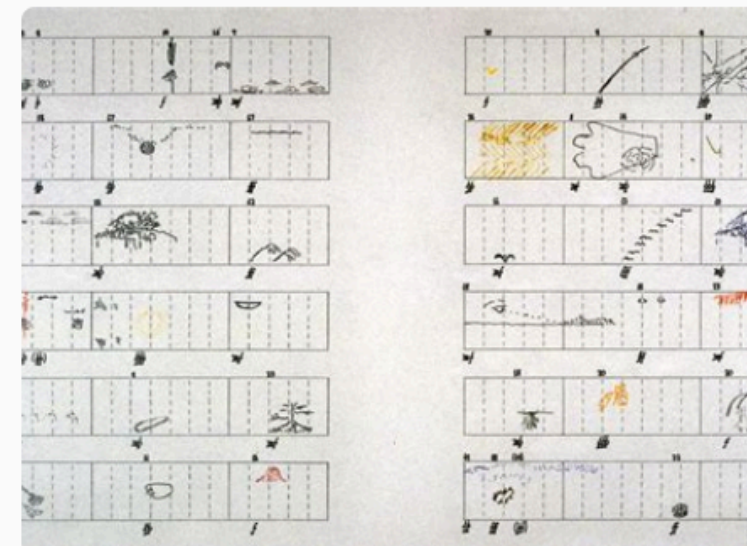
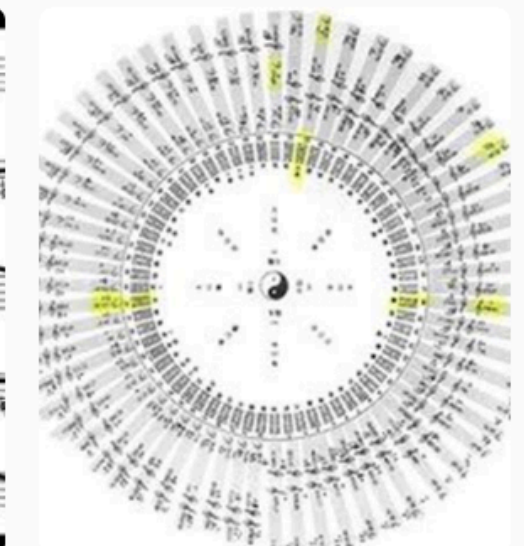
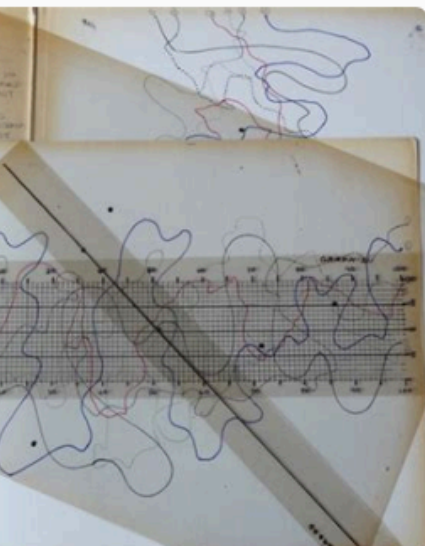
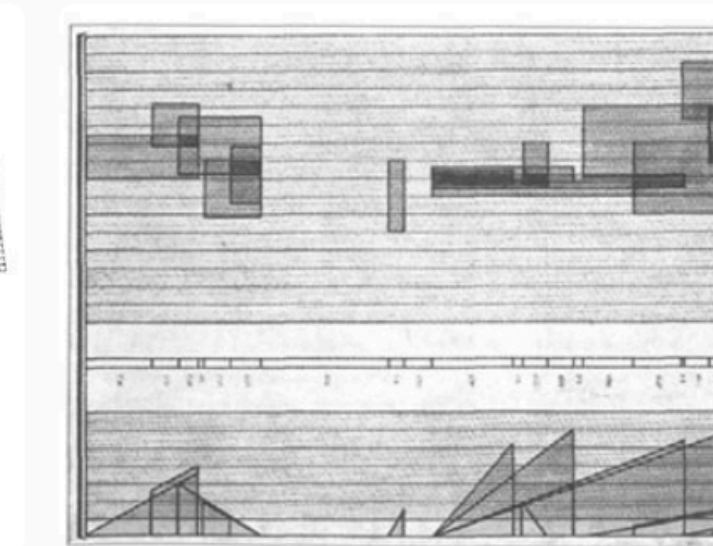
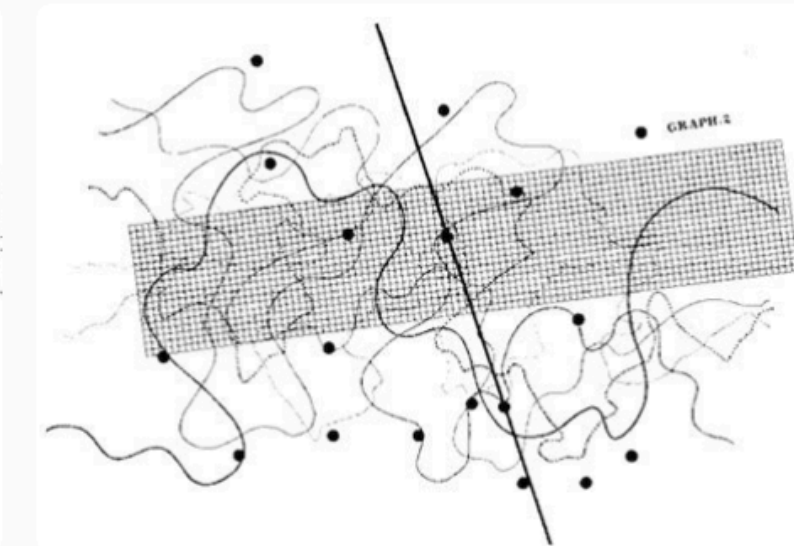
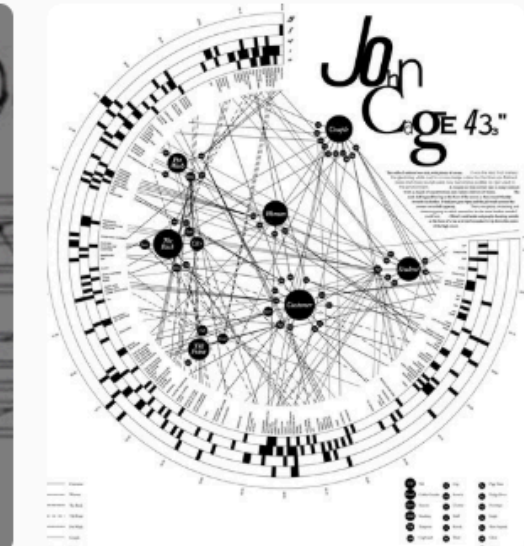
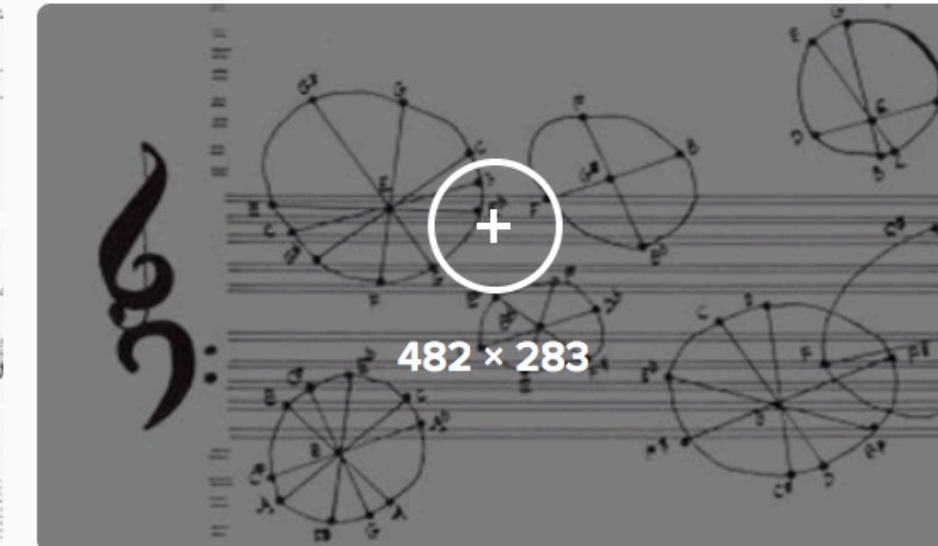
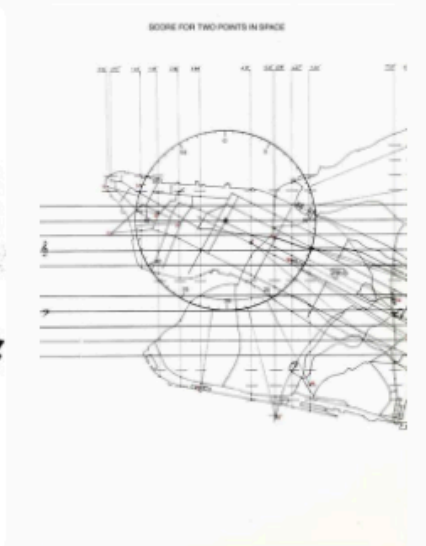
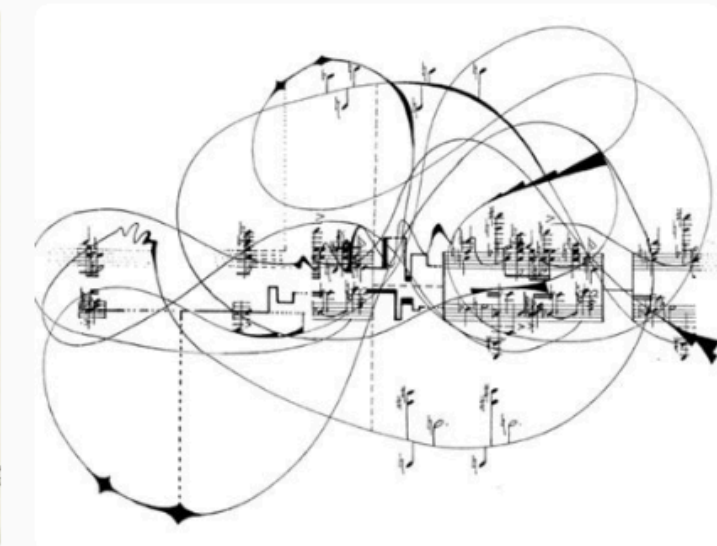
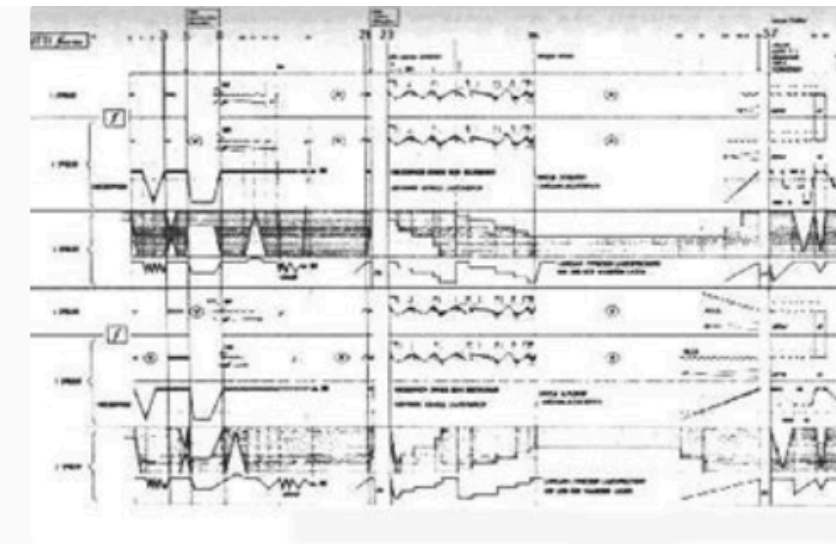
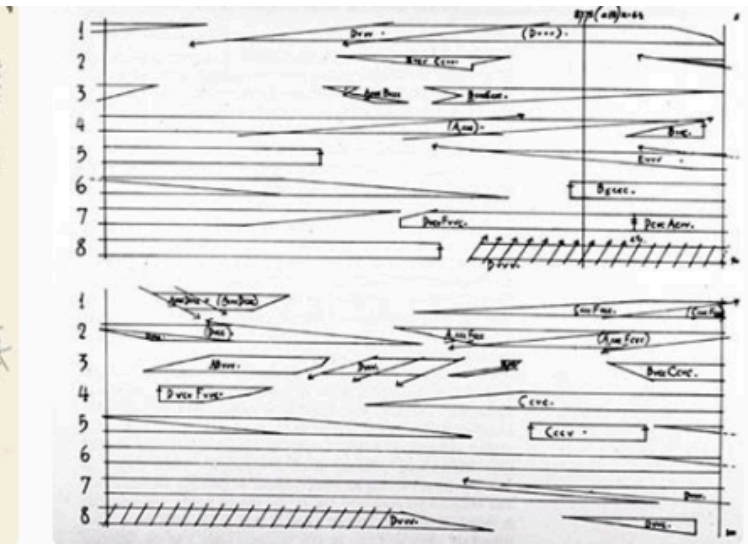
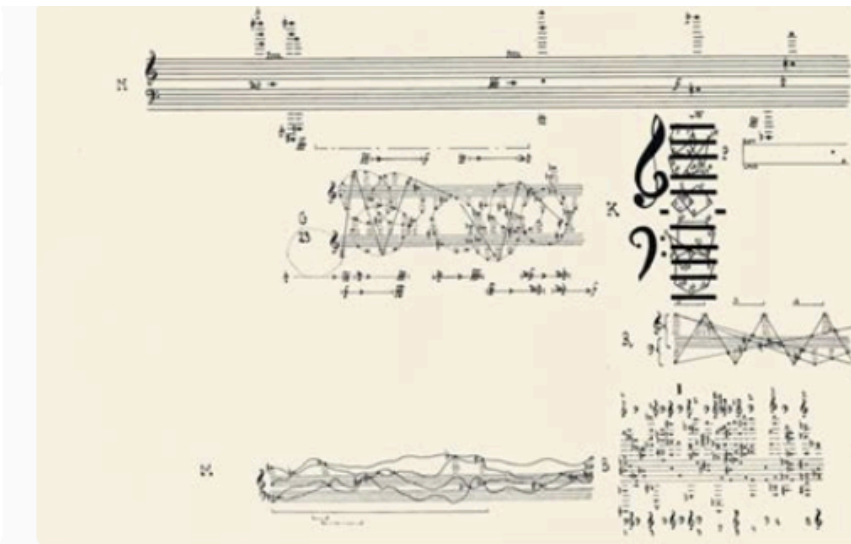
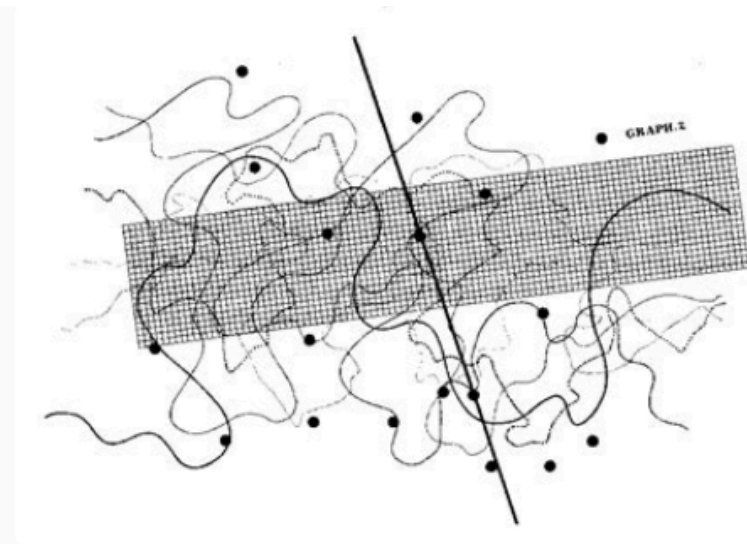
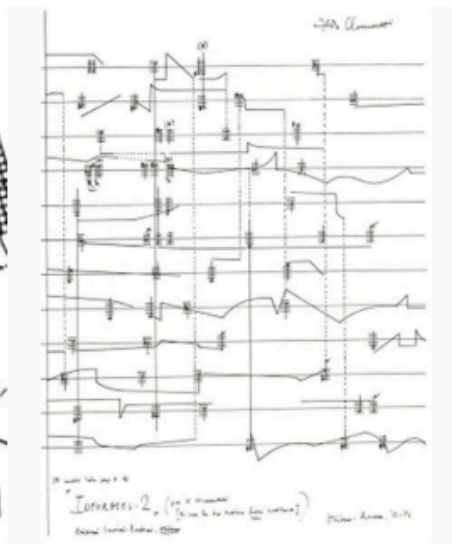
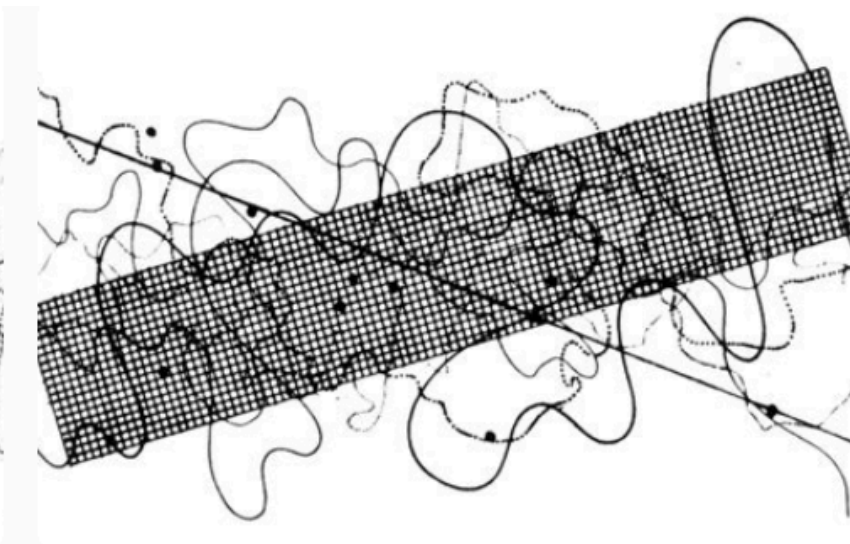
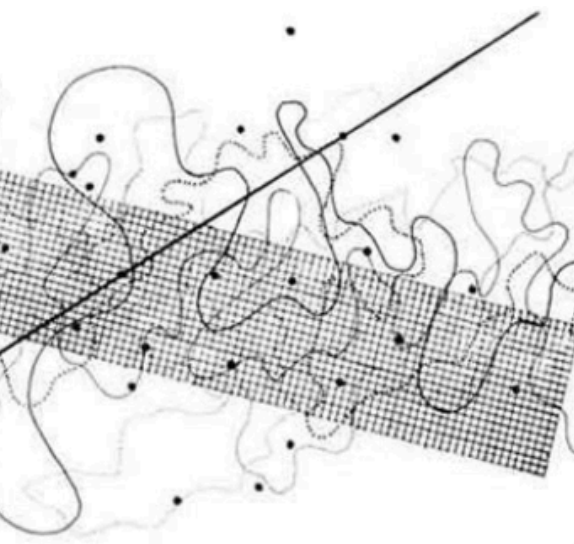
M.M. 60 = 



The image displays a musical score for castagnettes, consisting of a single melodic line on a treble clef staff in 3/4 time. The tempo is marked as M.M. 60. Below the staff are four figures, each labeled 'Fig. 1.' and numbered 1 through 4. Each figure includes a diagram of the castagnettes and a series of rhythmic patterns. Figure 1 shows a simple rhythmic pattern. Figure 2 includes a diamond-shaped diagram and a sequence of notes numbered 1 through 10. Figure 3 features a zigzag diagram and a sequence of notes with accents. Figure 4 has a zigzag diagram and a sequence of notes with accents. The rhythmic patterns are represented by vertical lines of varying heights and some circled notes, indicating specific rhythmic values and accents.



https://www.moma.org/explore/inside_out/2012/12/21/exhibiting-fluxus-keeping-score-in-tokyo-1955-1970-a-new-avant-garde/



Ice Spirits

4/8

35 C

Sop Sop Sop Sop

ice conversations 3x

ice conversations (2nd x)

ice conversations

2nd time 2nd ending

45

pwm

Fine Tune Keyframes

Speed min: 0.00, max: 5742.63 px/sec
Speed at 0:00:01:11: 739.14 px/sec

Align Paragraph
Align Layers to: Composition
Distribute Layers:

Effect Controls

MVI_9378 * MVI_9456.MOV

Video Effects

- fx Motion
- fx Opacity
- fx Time Remapping
- fx Brightness & Contrast
 - Brightness 98,6
 - Contrast 18,0
- fx Black & White

Velocity: 22,8 / second

00:00:41:20

Project: generiek2 *
generiek2.prproj

Media Browser

49 Items

MVI_9405.MOV 2:20 MVI_9406.MOV 3:09 MVI_9411.MOV 7:25 MVI_9413.MOV 3:37 MVI_9419.MOV 9:08

48 1 54 127 C min 4/4

Smart Drag: No Overlap

Synth Lead

Gain Reduction

EQ

Input

Audio FX

Sends

Output

Pan

VCA

dB

Main 111 21 133 Start 4 | 1 | 860 End 4 | 1 | 860 Length 0 | 0 | 000

Sub 0:34

Cursor 1 | 1 | 2

Grid 0 | 1 | 000

Nudge 0 | 0 | 480

Count Off Meter 1 bar 484 Tempo 71.0000

Barr/Beats

Min:Secs

Tempo

Chords

Markers

Collab

Inserts A-E

I/O

5 Bass DI.01

7 Ac GTR U87 1

8 AC GTR

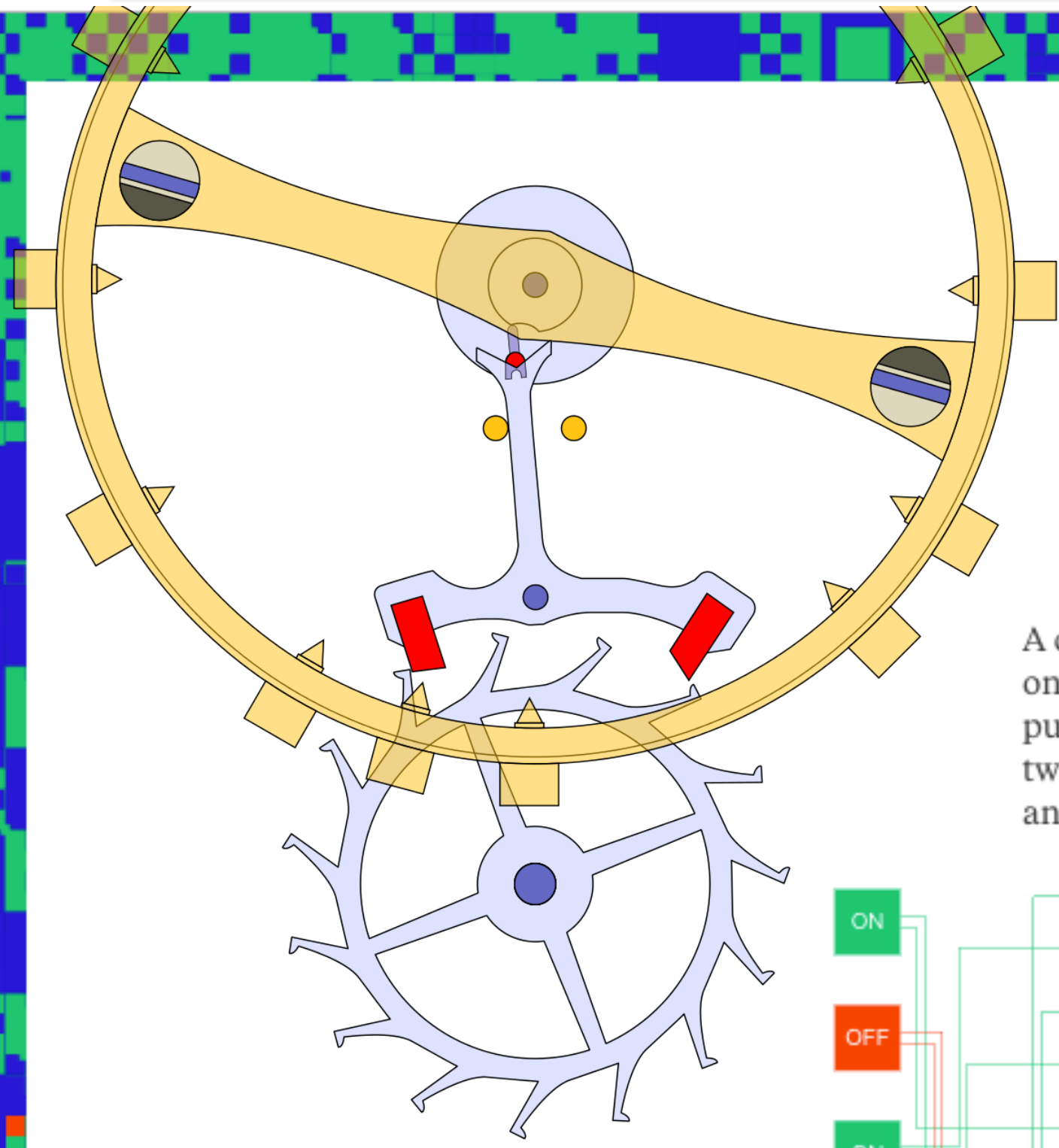
9 EG 11R.01

10 EG FX

11 Pad.01

12 LeadVoxENG

13 Back ENG.01

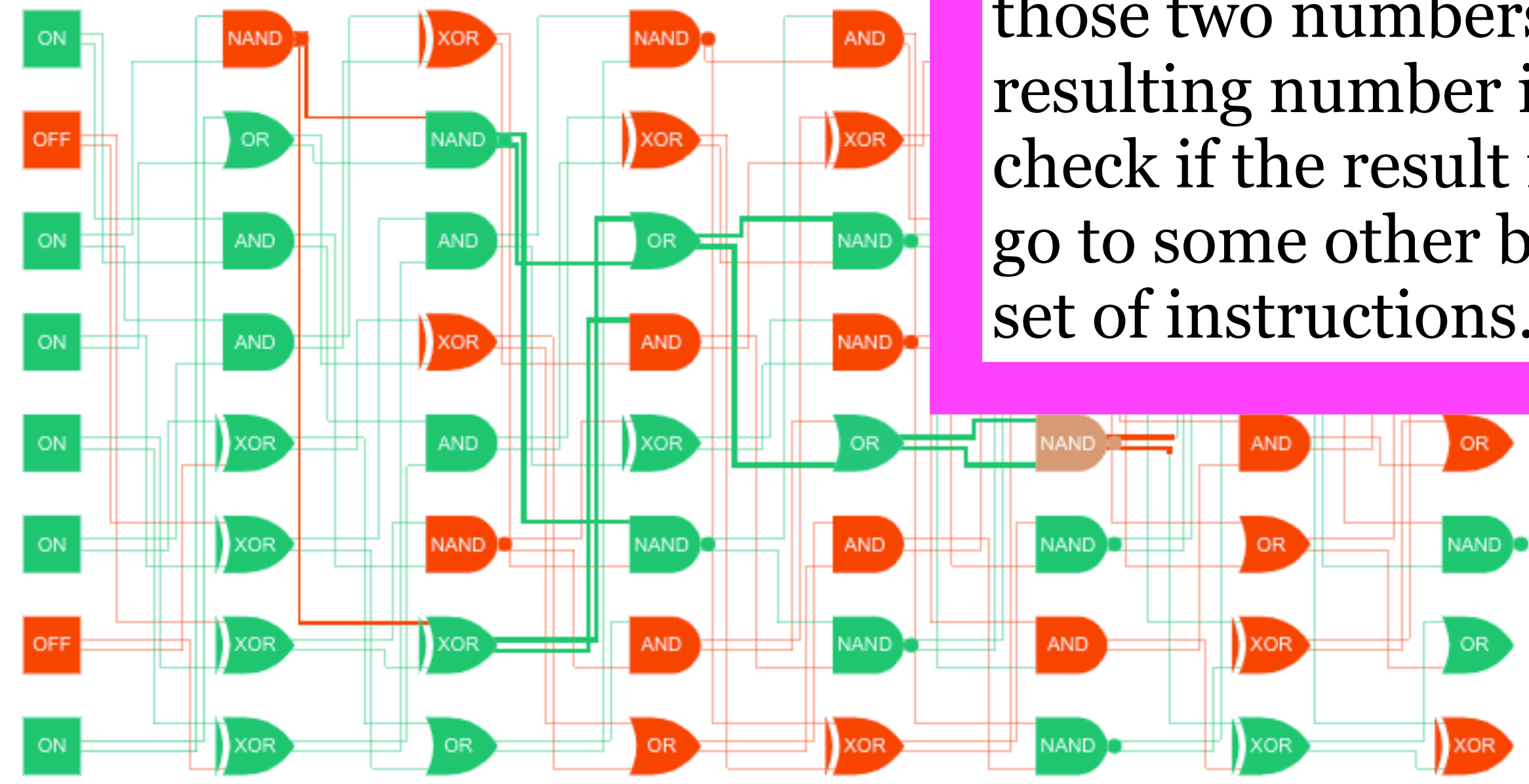


2

Let's Be

A computer is a clock with benefits. They all work the same, doing second-grade math, one step at a time: Tick, take a number and put it in box one. Tick, take another number, put it in box two. Tick, *operate* (an operation might be addition or subtraction) on those two numbers and put the resulting number in box one. Tick, check if the result is zero, and if it is, go to some other box and follow a new set of instructions.

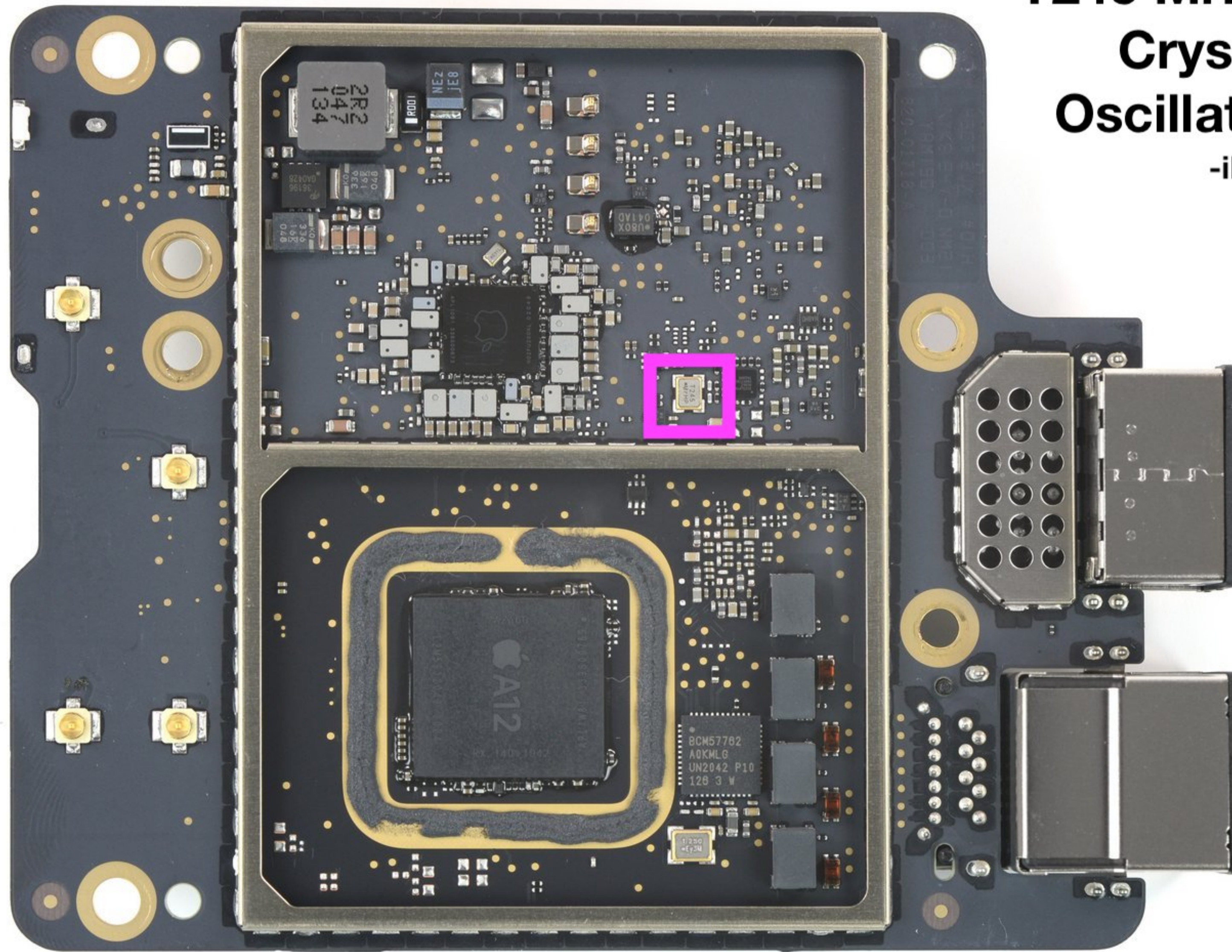
“A computer is a clock with benefits. They all work the same, doing second-grade math, one step at a time: Tick, take a number and put it in box one. Tick, take another number, put it in box two. Tick, *operate* (an operation might be addition or subtraction) on those two numbers and put the resulting number in box one. Tick, check if the result is zero, and if it is, go to some other box and follow a new set of instructions.”

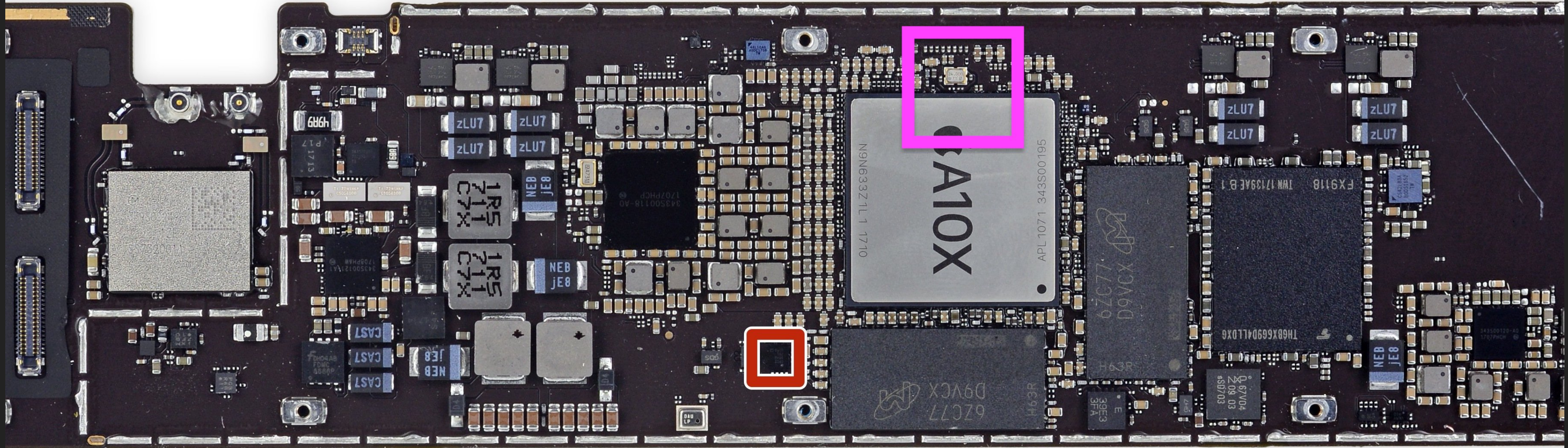


This is simulated circuitry that's computing as you watch. The switches on the left turn the current on and off at random, and the logic gates direct the flow of the current. Click the boxes to change the circuits. Enough of these can compute anything computable.

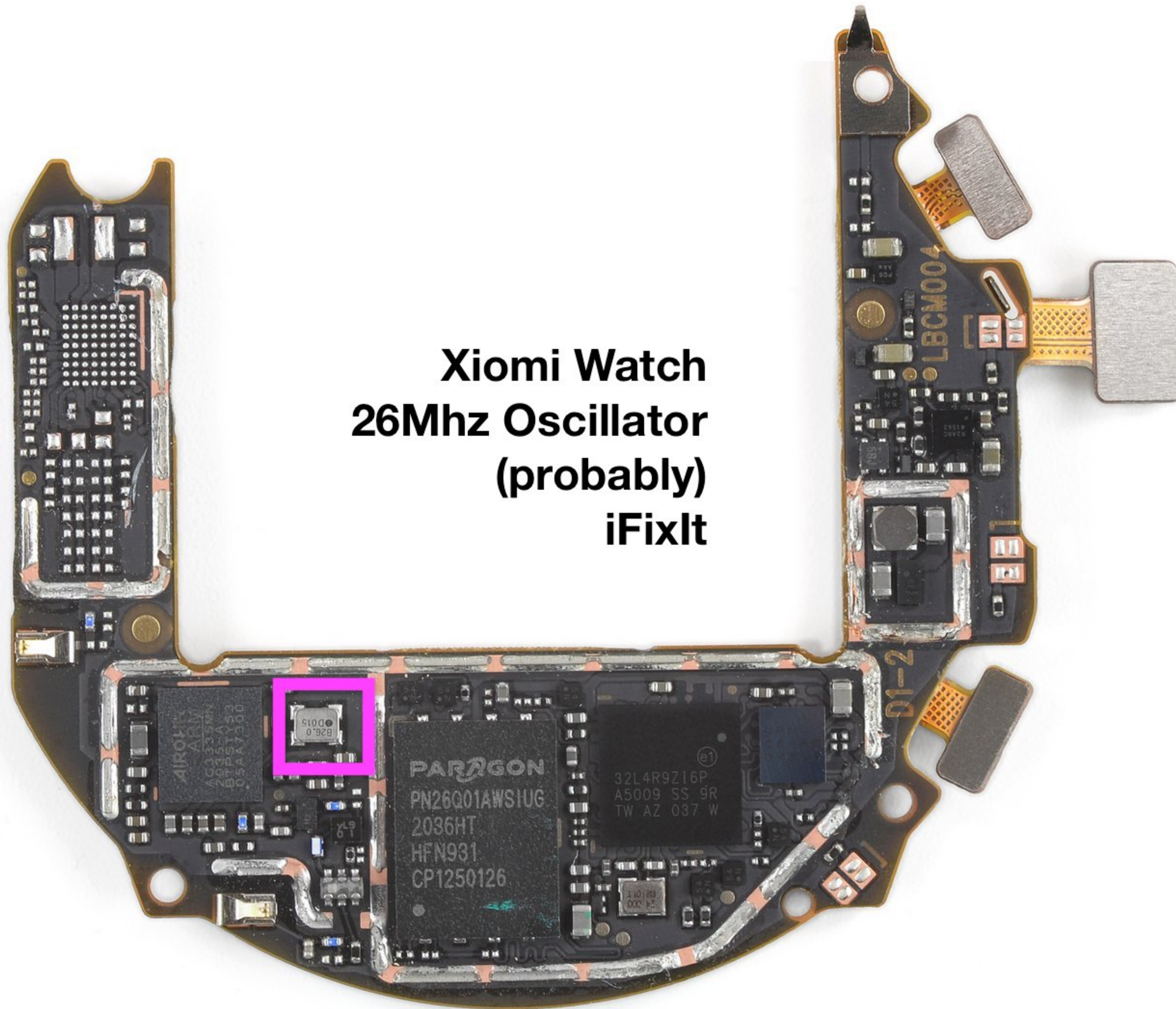
Crystal Oscillator

-iFixIt





**Xiomi Watch
26Mhz Oscillator
(probably)
iFixIt**



NBVSBAXXX Series

2.5 V/3.3 V, LVPECL Voltage-Controlled Crystal Oscillator (VCXO) PureEdge™ Product Series

The NBVSBAXXX series voltage-controlled crystal oscillator (VCXO) devices are designed to meet today's requirements for 2.5 V and 3.3 V LVPECL clock generation applications. These devices use a high Q fundamental mode crystal and Phase Locked Loop (PLL) multiplier to provide a wide range of frequencies from 60 MHz to 700 MHz (factory configurable per user specifications) with a pullable range of ± 100 ppm and a frequency stability of ± 50 ppm. The silicon-based PureEdge™ products design provides users with exceptional frequency stability and reliability. They produce an ultra low jitter and phase noise LVPECL differential output.

The NBVSBAXXX series are members of ON Semiconductor's PureEdge™ clock family that provides accurate and precision clock generation solutions.

Available in the industry standard 5.0 x 7.0 x 1.8 mm and in a new 3.2 x 5.0 x 1.2 mm SMD (CLCC) package on 16 mm tape and reel in quantities of 1,000.

Features

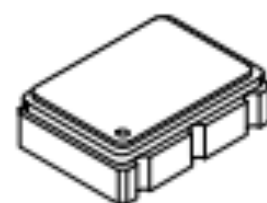
- LVPECL Differential Output
- Operating Range: 2.5 V $\pm 5\%$, 3.3 V $\pm 10\%$



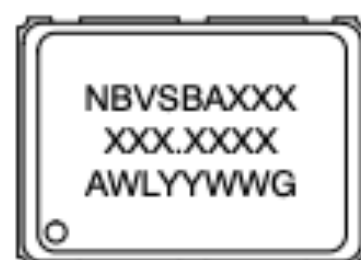
ON Semiconductor®

<http://onsemi.com>

MARKING DIAGRAM



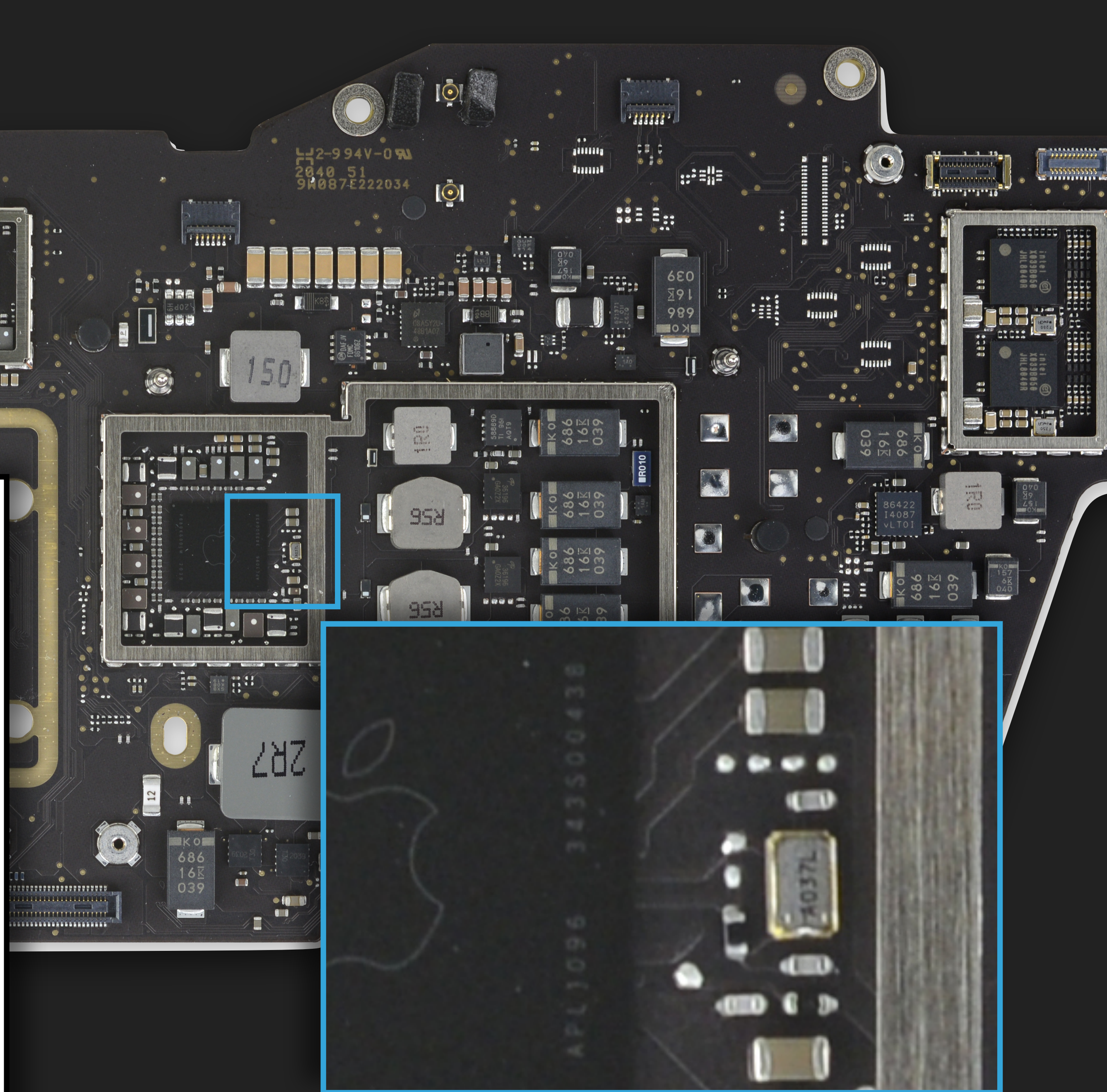
6 PIN CLCC
LN SUFFIX
CASE 848AB



6 PIN CLCC
LU SUFFIX
CASE 848AC



NBVSBAXXX = NBVSBAXXX (± 50 ppm)
XXX.XXXX = Output Frequency (MHz)
A = Assembly Location



ONSEMI A037L 707.35MHz VCXO

```
sketch_sep29a | Arduino 1.8.13
sketch_sep29a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

ARDUINO

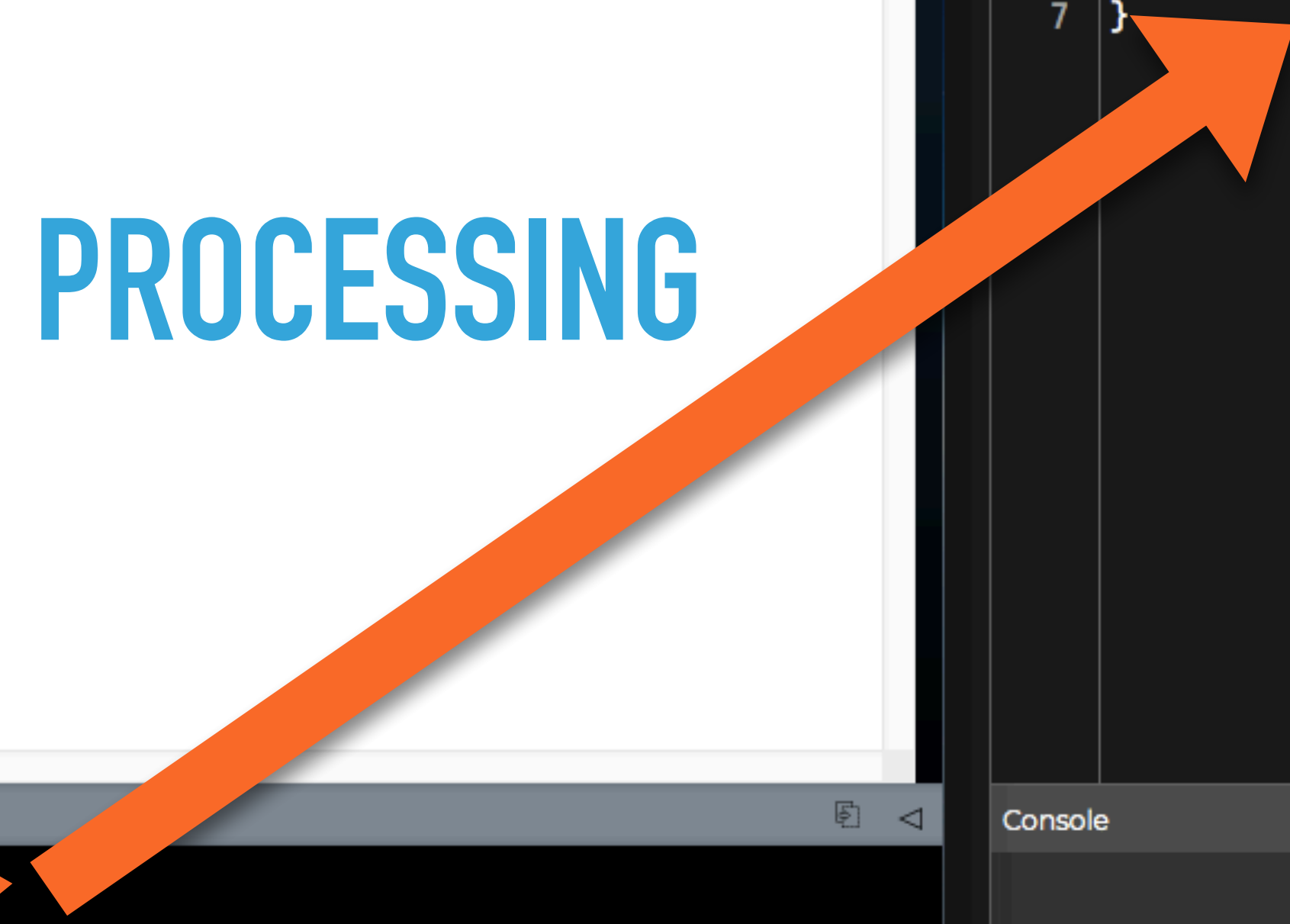
```
sketch_200929a | Processing 3.5.3
sketch_200929a
1 void setup() {
2
3 }
4
5 void draw() {
6
7 }
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
```

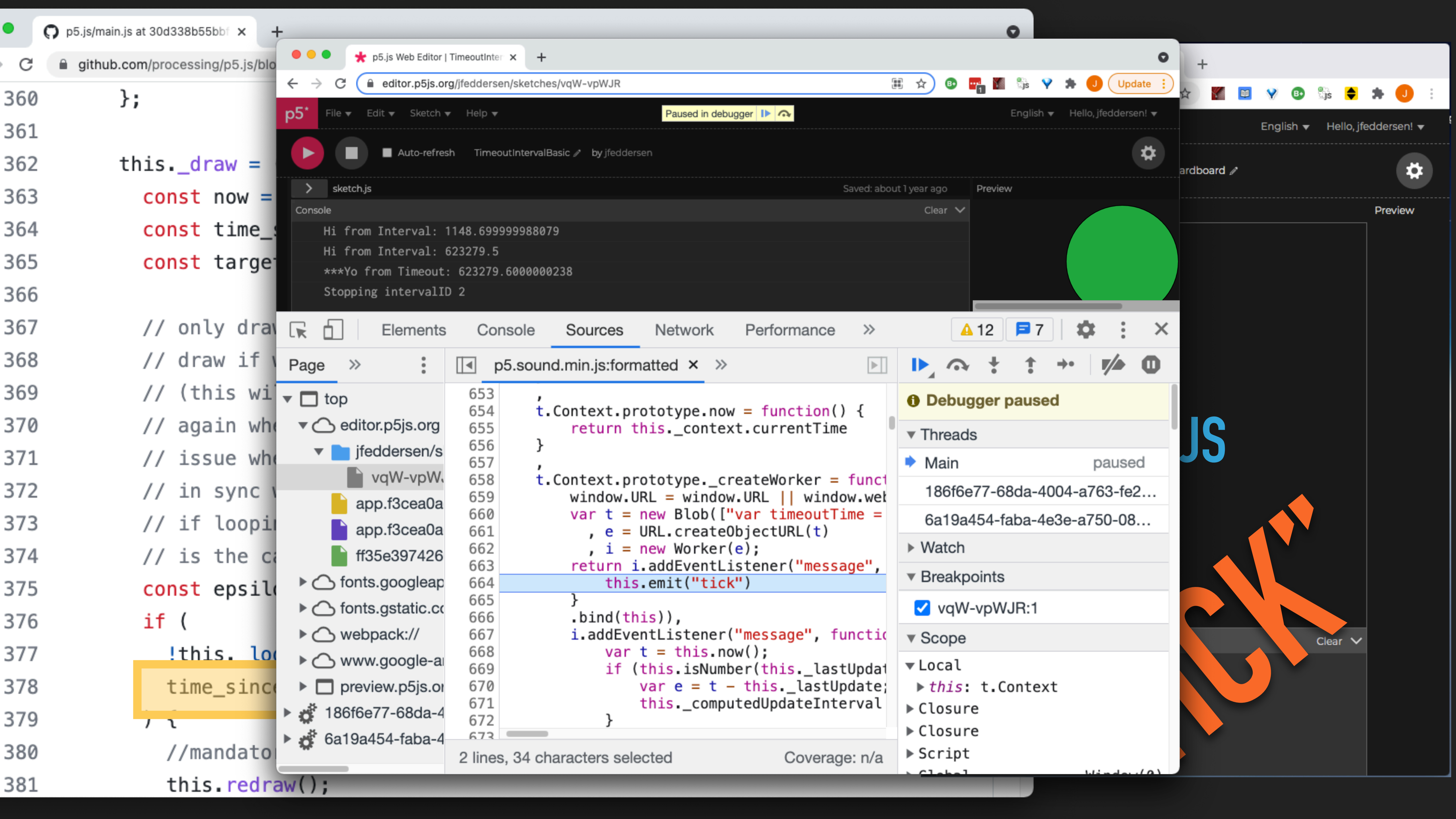
PROCESSING

```
p5.js Web Editor
editor.p5js.org
p5* File Edit Sketch Help English Hello, jfeddersen!
Auto-refresh Nettle cardboard
sketch.js Preview
1 function setup() {
2   createCanvas(400, 400);
3 }
4
5 function draw() {
6   background(220);
7 }
```

P5JS

TICK





```
360 };
361
362 this._draw =
363   const now =
364   const time_
365   const target
366
367 // only draw
368 // draw if v
369 // (this wi
370 // again wh
371 // issue wh
372 // in sync v
373 // if loopi
374 // is the c
375 const epsil
376 if (
377   !this. loc
378   time_sinc
379
380 //mandato
381 this.redraw();
```

p5.js Web Editor | TimeoutInter x

editor.p5js.org/jfeddersen/sketches/vqW-vpWJR

Paused in debugger

File Edit Sketch Help

Auto-refresh TimeoutIntervalBasic by jfeddersen

sketch.js Saved: about 1 year ago Preview

Console

```
Hi from Interval: 1148.699999988079
Hi from Interval: 623279.5
***Yo from Timeout: 623279.6000000238
Stopping intervalID 2
```

Elements Console Sources Network Performance

Page >> p5.sound.min.js:formatted x >>

top

- editor.p5js.org
 - jfeddersen/s
 - vqW-vpW,

```
653 ,
654 t.Context.prototype.now = function() {
655   return this._context.currentTime
656 }
657 ,
658 t.Context.prototype._createWorker = functi
659 window.URL = window.URL || window.web
660 var t = new Blob(["var timeoutTime =
661   , e = URL.createObjectURL(t)
662   , i = new Worker(e);
663   return i.addEventListener("message",
664     this.emit("tick")
665   )
666   .bind(this)),
667   i.addEventListener("message", functio
668     var t = this.now();
669     if (this.isNumber(this._lastUpdat
670       var e = t - this._lastUpdate;
671       this._computedUpdateInterval
672     }
673
```

Debugger paused

Threads

- Main paused

Watch

Breakpoints

- vqW-vpWJR:1

Scope

- Local
 - this: t.Context
- Closure
- Closure
- Script

JS

CRACK

EASING

Smoothly transition a variable from one value to another in a set time

SIMULATION

Use physics or other rules to determine next frame for one or more objects.

TIMELINES

Schedule code for execution in the future

EASING

Quadratic Easing

Flash's Timeline tweens use something called *quadratic easing*—which could actually be termed “normal” easing. The word *quadratic* refers to the fact that the equation for this motion is based on a squared variable, in this case, t^2 :

$$p(t) = t^2$$



NOTE: I always wondered why the term *quad-ratic* (the prefix means “four”) is used to describe equations with a degree of two (x^2). While writing this chapter, I finally looked it up in the dictionary (RTFD, you might say). I discovered that *quad* originally referred to the four sides of a square. Thus, a squared variable is *quadratic*.

I used the quadratic easing curve earlier in Figure 7-4. It's actually half a parabola. Here it is again, for reference purposes, in Figure 7-7.

Here's the quadratic ease-in ActionScript function:

```
Math.easeInQuad = function (t, b, c, d) {  
    return c*(t/=d)*t + b;  
};
```

Recall that t is time, b is beginning position, c is the total change in position, and d is the duration of the tween.

This equation is more complex than the linear tween, but it's the simplest of the equations that implement easing. Basically, I normalize t by dividing it by d . This forces t to fall between 0 and 1. I multiply t by itself to produce quadratic curvature in the values. Then I scale the value from a

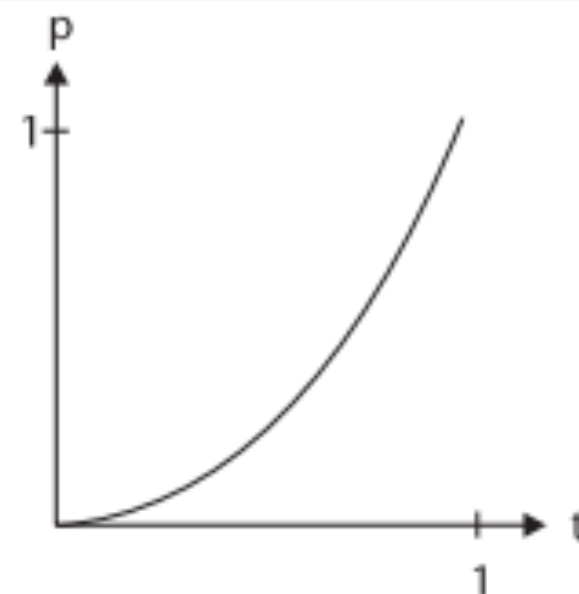


FIGURE 7-7

Graph of quadratic easing

ROBERT PENNER

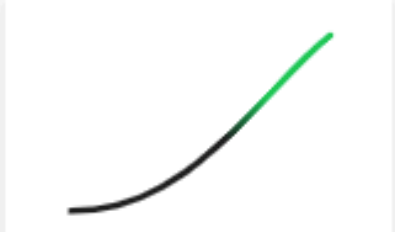
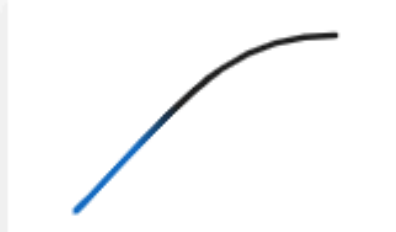
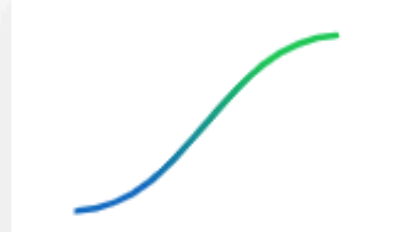
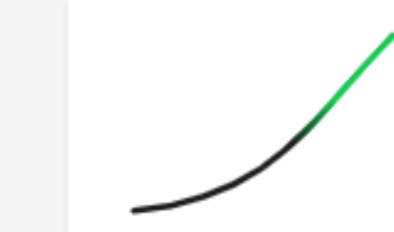

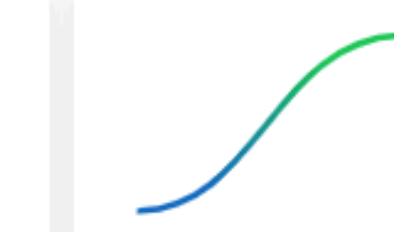
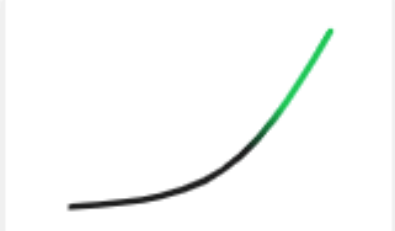
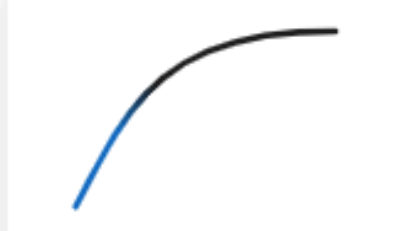
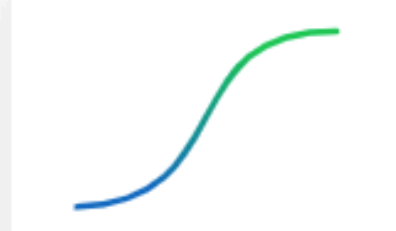


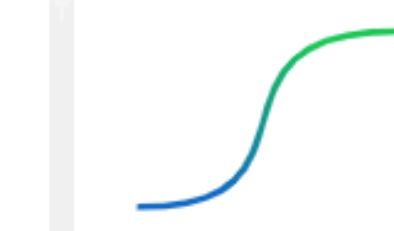
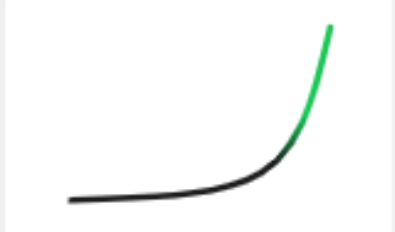
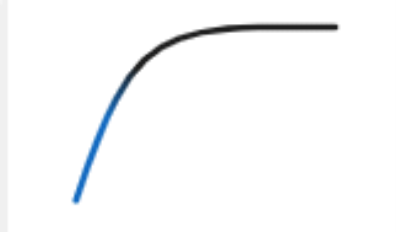
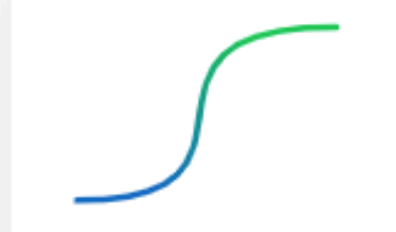


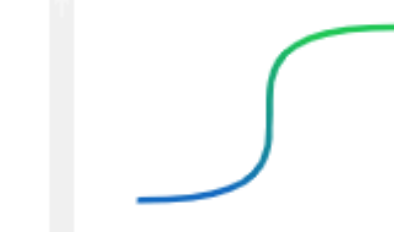
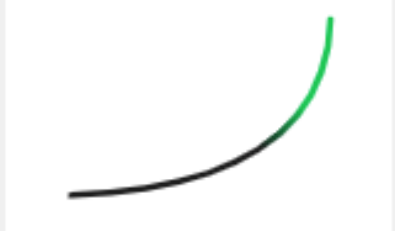

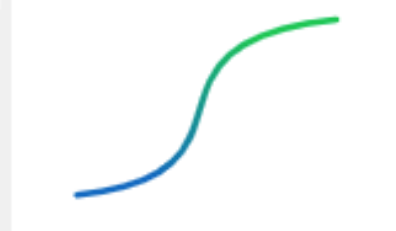


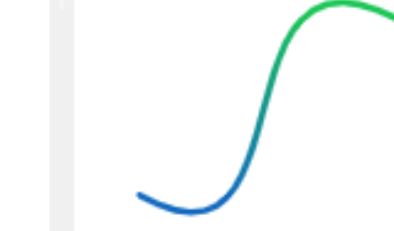

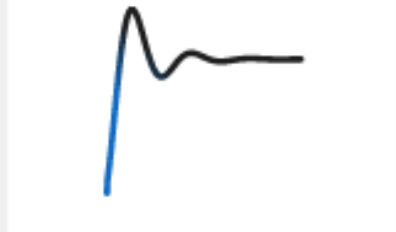


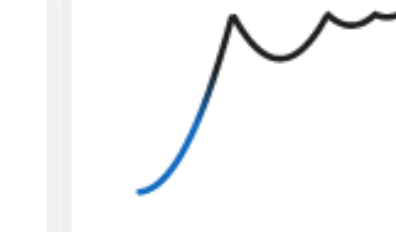

← → ↻ <https://easings.net/en> ★ ABP ... B+ js J +

Easing functions specify the rate of change of a parameter over time.

Objects in real life don't just start and stop instantly, and almost never move at a constant speed. When we open a drawer, we first move it quickly, and slow it down as it comes out. Drop something on the floor, and it will first accelerate downwards, and then bounce back up after hitting the floor.

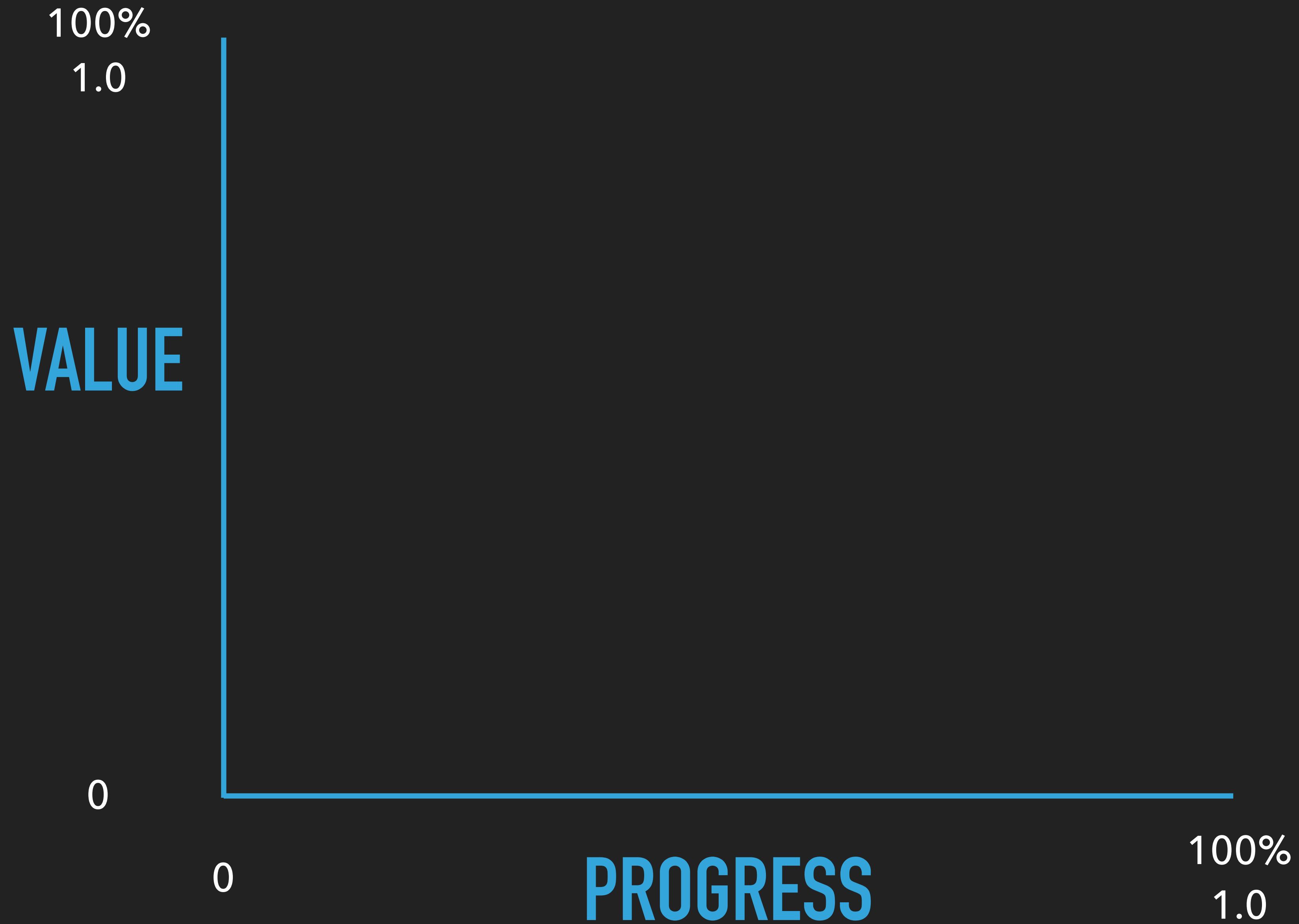
This page helps you choose the right easing function.

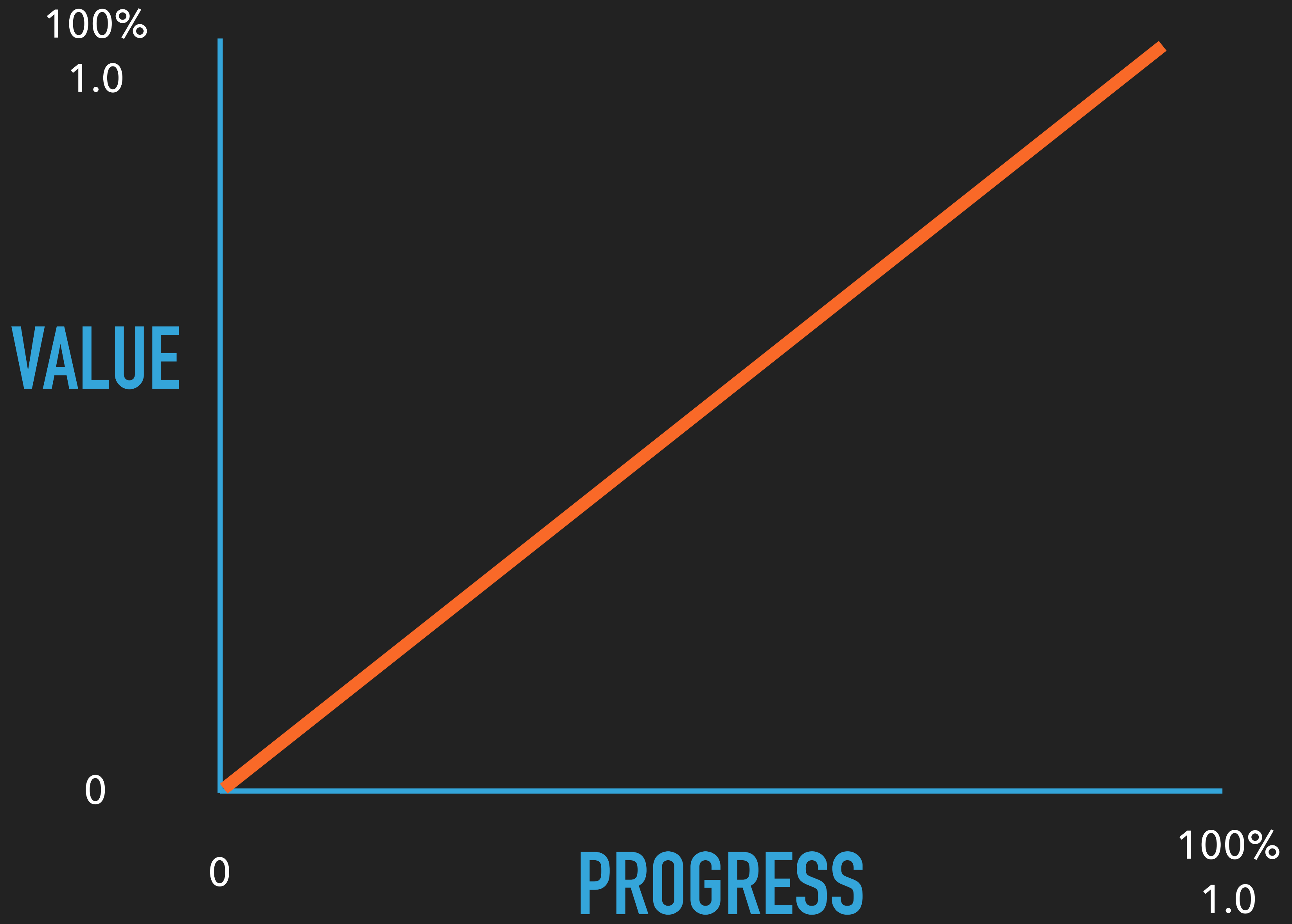
[Open Source](#)
Help translate site to your language

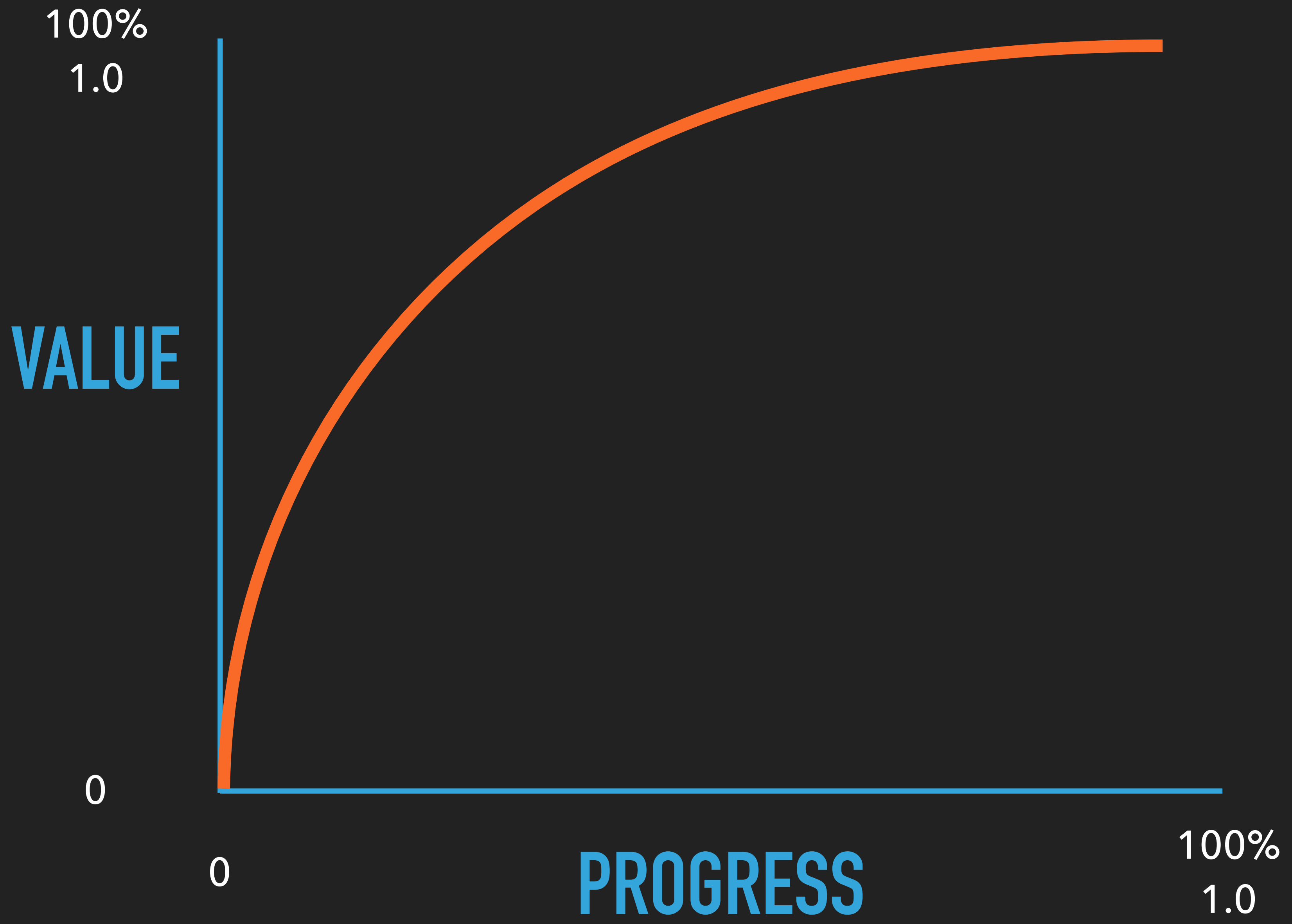
 easeInSine	 easeOutSine	 easeInOutSine	 easeInQuad	 easeOutQuad	 easeInOutQuad
 easeInCubic	 easeOutCubic	 easeInOutCubic	 easeInQuart	 easeOutQuart	 easeInOutQuart
 easeInQuint	 easeOutQuint	 easeInOutQuint	 easeInExpo	 easeOutExpo	 easeInOutExpo
 easeInCirc	 easeOutCirc	 easeInOutCirc	 easeInBack	 easeOutBack	 easeInOutBack
 easeInElastic	 easeOutElastic	 easeInOutElastic	 easeInBounce	 easeOutBounce	 easeInOutBounce

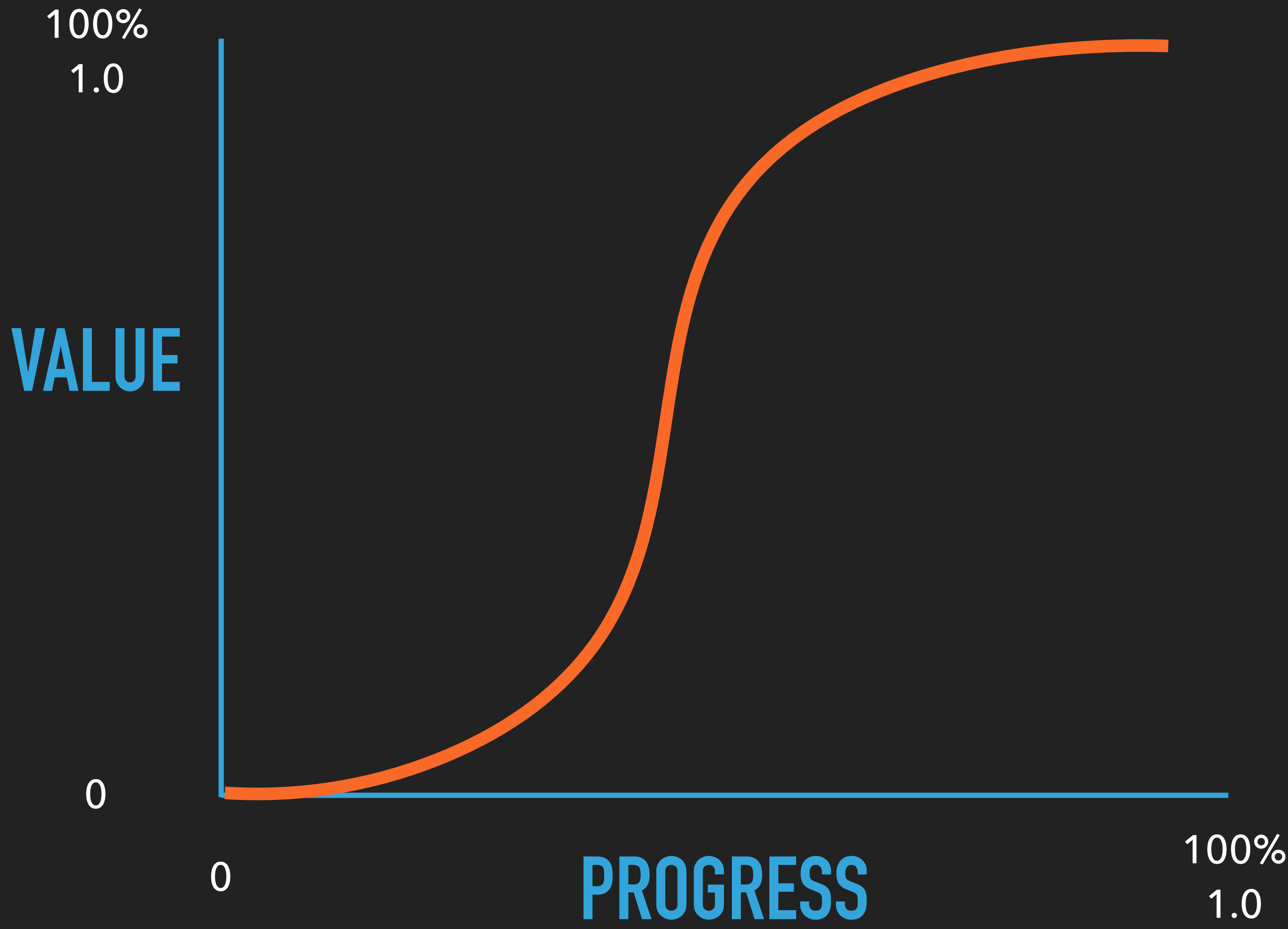
EASING

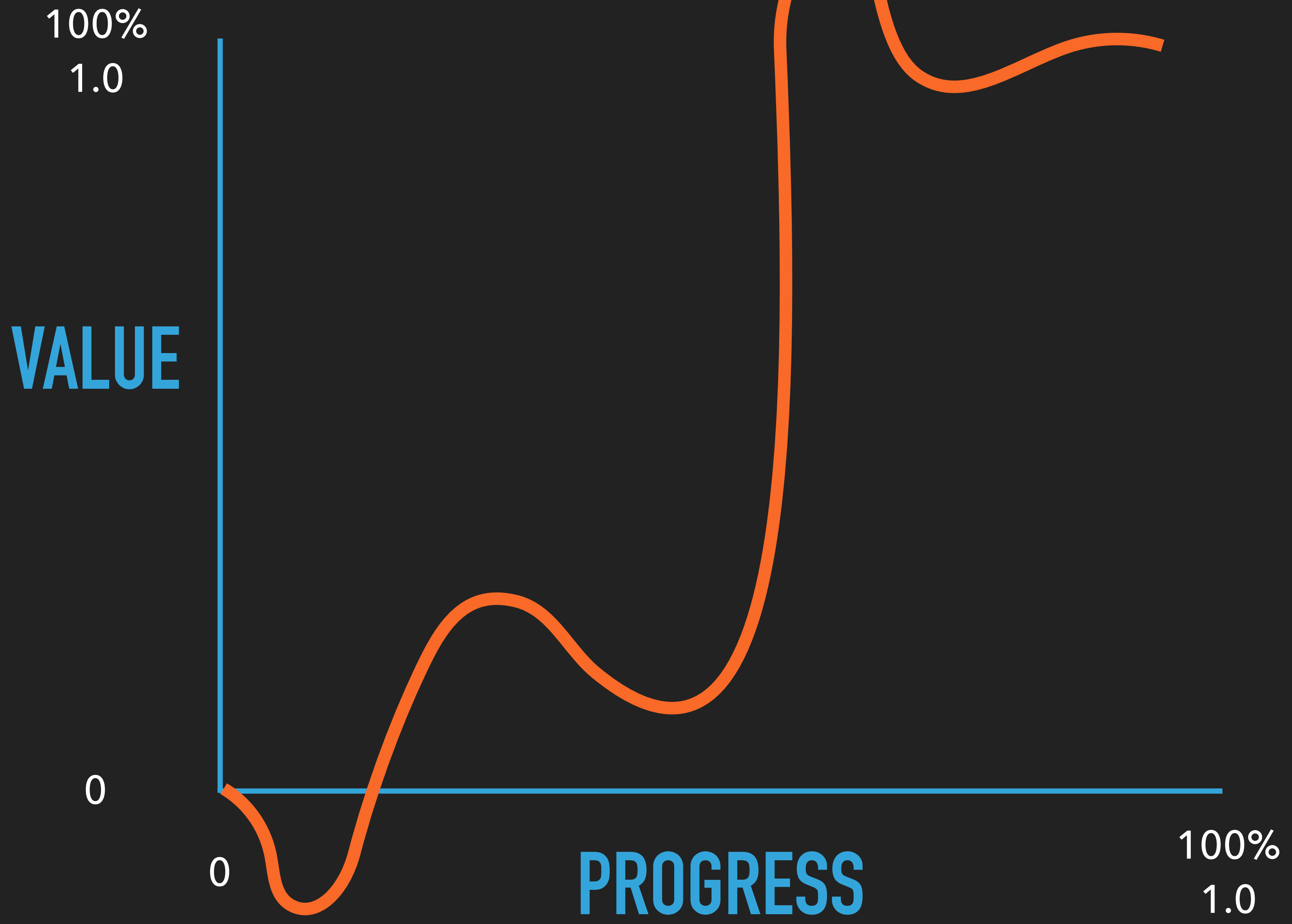
Smoothly transition a variable from one value to another in a set time











ROBERT PENNER







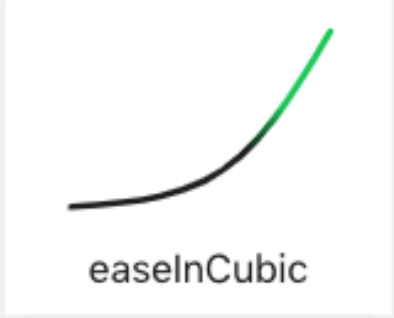




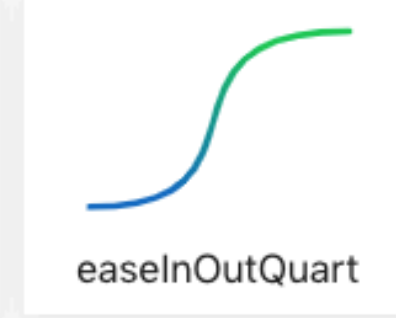



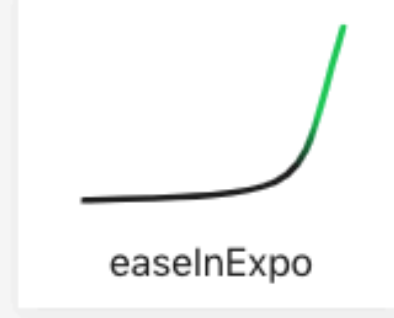
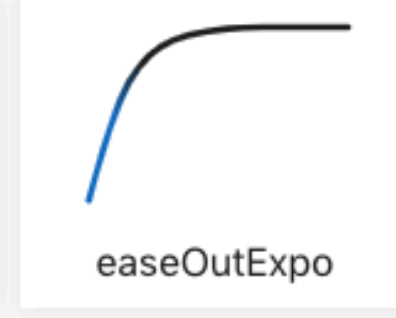
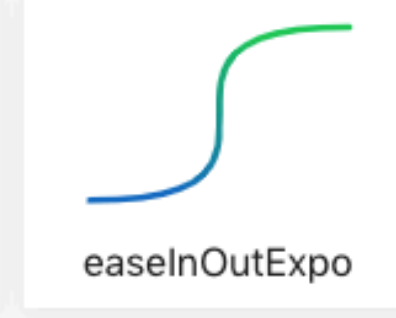








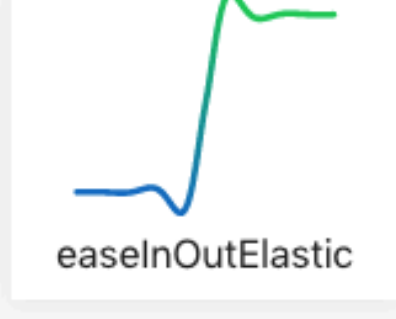



← → ↻ <https://easings.net/en> ★ ABP ... B+ js J +

Easing functions specify the rate of change of a parameter over time.

Objects in real life don't just start and stop instantly, and almost never move at a constant speed. When we open a drawer, we first move it quickly, and slow it down as it comes out. Drop something on the floor, and it will first accelerate downwards, and then bounce back up after hitting the floor.

This page helps you choose the right easing function.

[Open Source](#)
Help translate site to your language

 easeInSine	 easeOutSine	 easeInOutSine	 easeInQuad	 easeOutQuad	 easeInOutQuad
 easeInCubic	 easeOutCubic	 easeInOutCubic	 easeInQuart	 easeOutQuart	 easeInOutQuart
 easeInQuint	 easeOutQuint	 easeInOutQuint	 easeInExpo	 easeOutExpo	 easeInOutExpo
 easeInCirc	 easeOutCirc	 easeInOutCirc	 easeInBack	 easeOutBack	 easeInOutBack
 easeInElastic	 easeOutElastic	 easeInOutElastic	 easeInBounce	 easeOutBounce	 easeInOutBounce

Quadratic Easing

Flash's Timeline tweens use something called *quadratic easing*—which could actually be termed “normal” easing. The word *quadratic* refers to the fact that the equation for this motion is based on a squared variable, in this case, t^2 :

$$p(t) = t^2$$



NOTE: I always wondered why the term *quad-ratic* (the prefix means “four”) is used to describe equations with a degree of two (x^2). While writing this chapter, I finally looked it up in the dictionary (RTFD, you might say). I discovered that *quad* originally referred to the four sides of a square. Thus, a squared variable is *quadratic*.

I used the quadratic easing curve earlier in Figure 7-4. It's actually half a parabola. Here it is again, for reference purposes, in Figure 7-7.

Here's the quadratic ease-in ActionScript function:

```
Math.easeInQuad = function (t, b, c, d) {
    return c*(t/=d)*t + b;
};
```

Recall that t is time, b is beginning position, c is the total change in position, and d is the duration of the tween.

This equation is more complex than the linear tween, but it's the simplest of the equations that implement easing. Basically, I normalize t by dividing it by d . This forces t to fall between 0 and 1. I multiply t by itself to produce quadratic curvature in the values. Then I scale the value from a

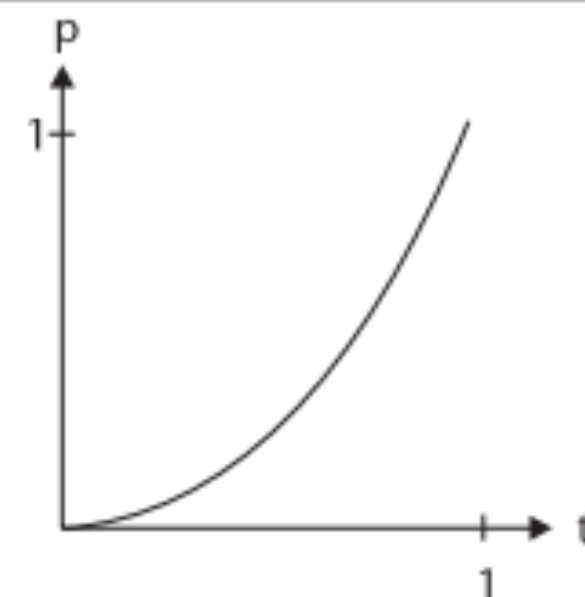
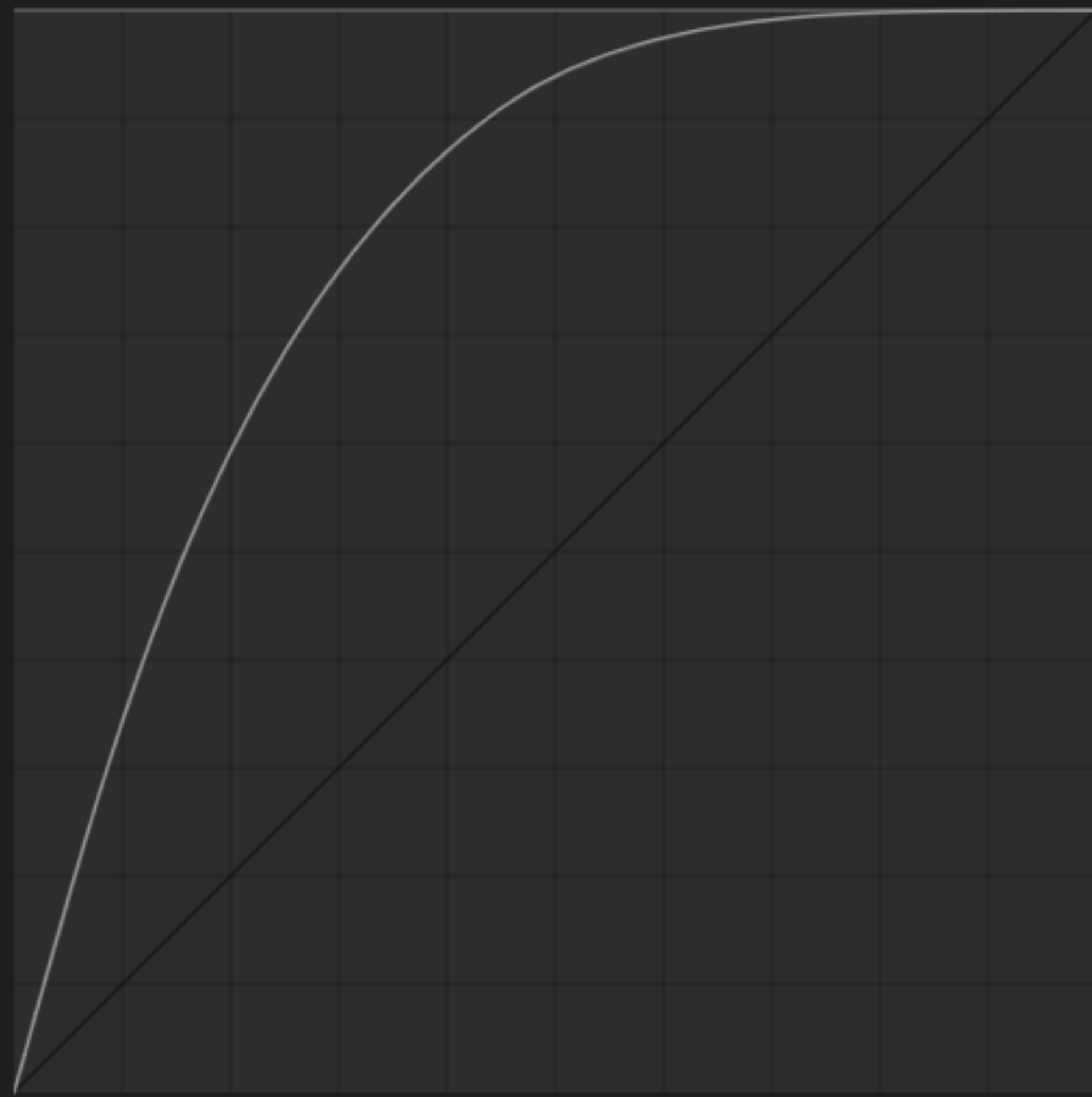


FIGURE 7-7

Graph of quadratic easing

GreenSock Ease Visualizer



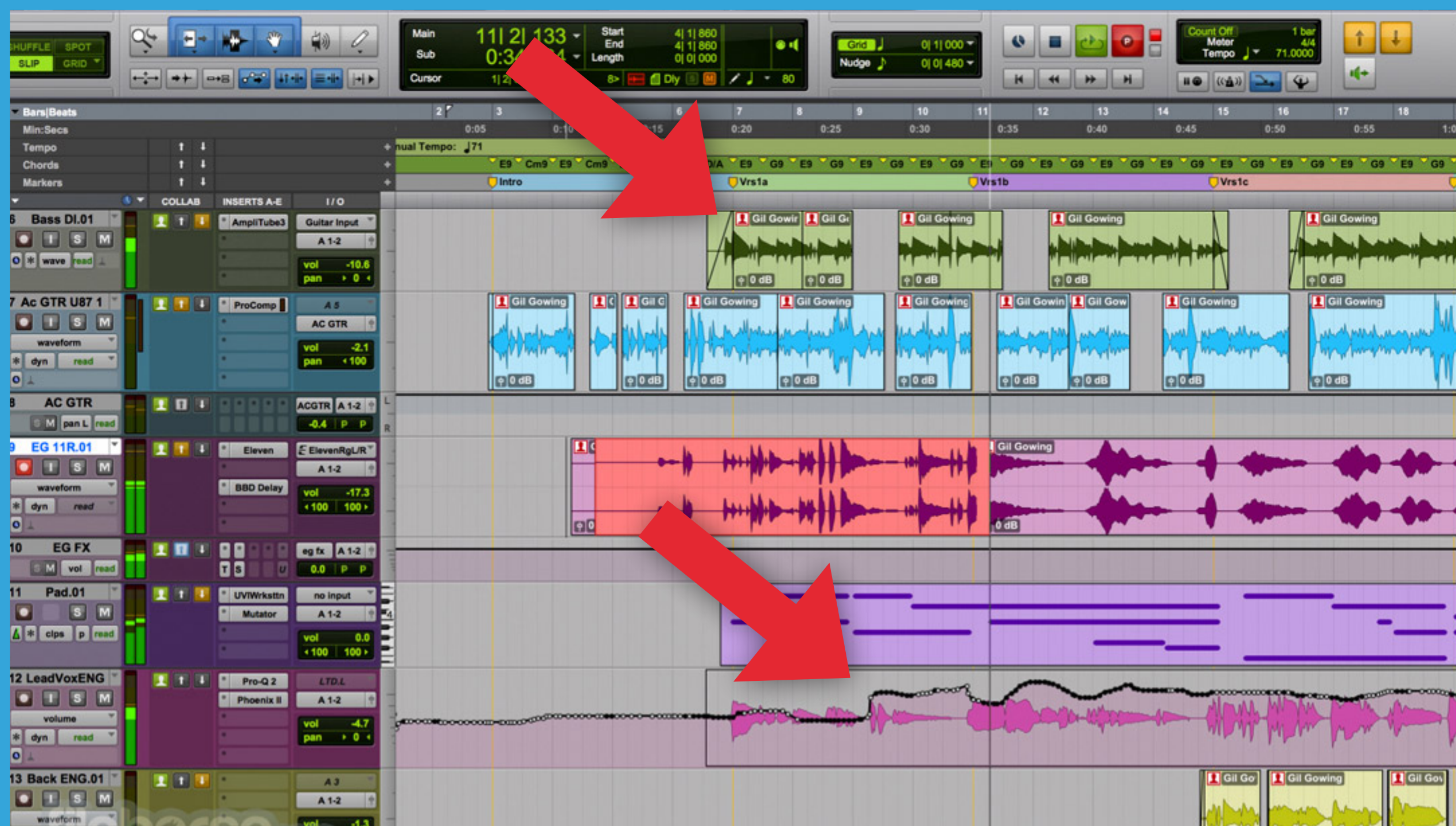
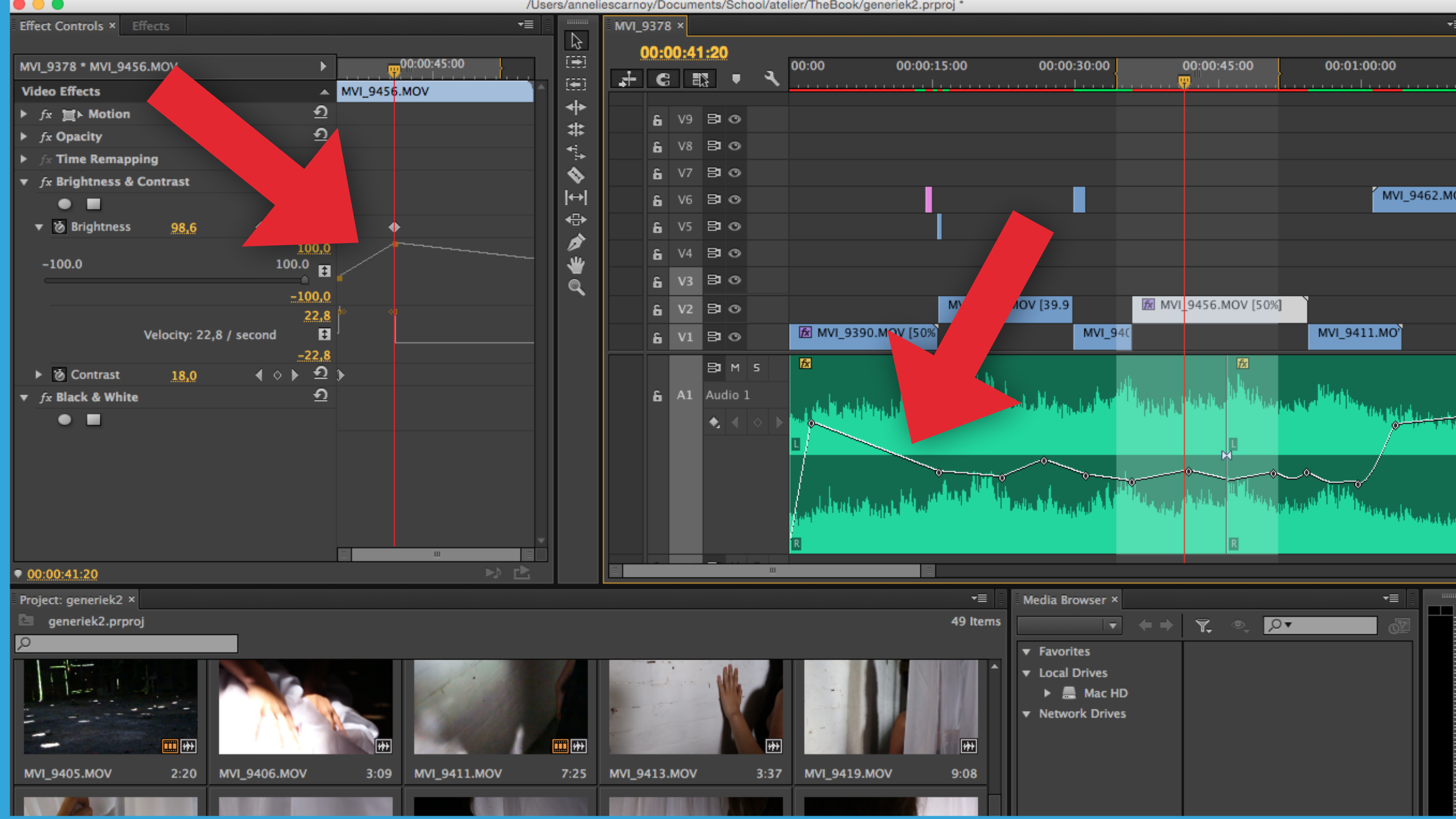
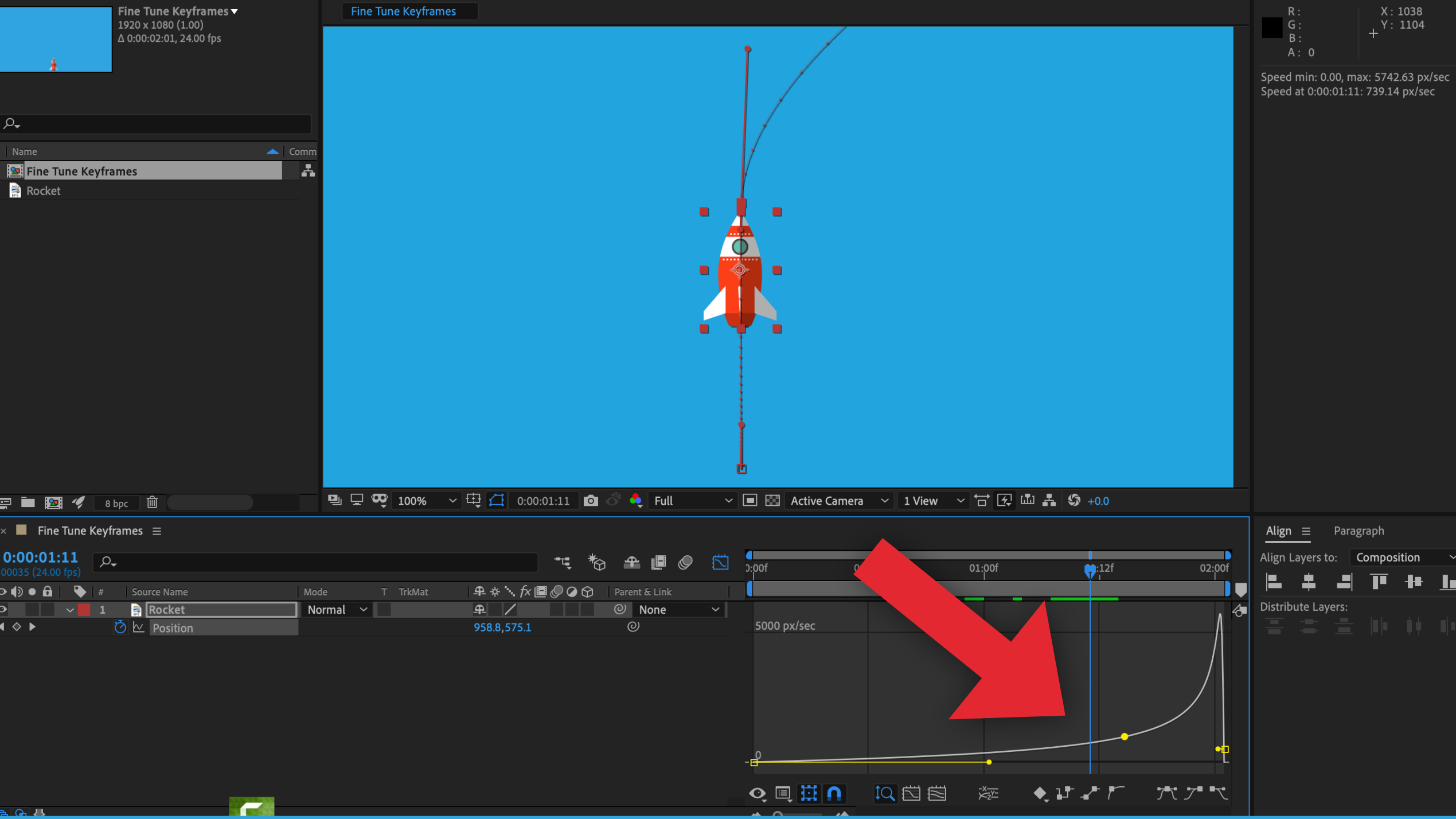
none
power1
power2
power3
power4
back
elastic
bounce
rough
slow
steps
circ
expo
sine
Custom

Type: out

RUN

1.00

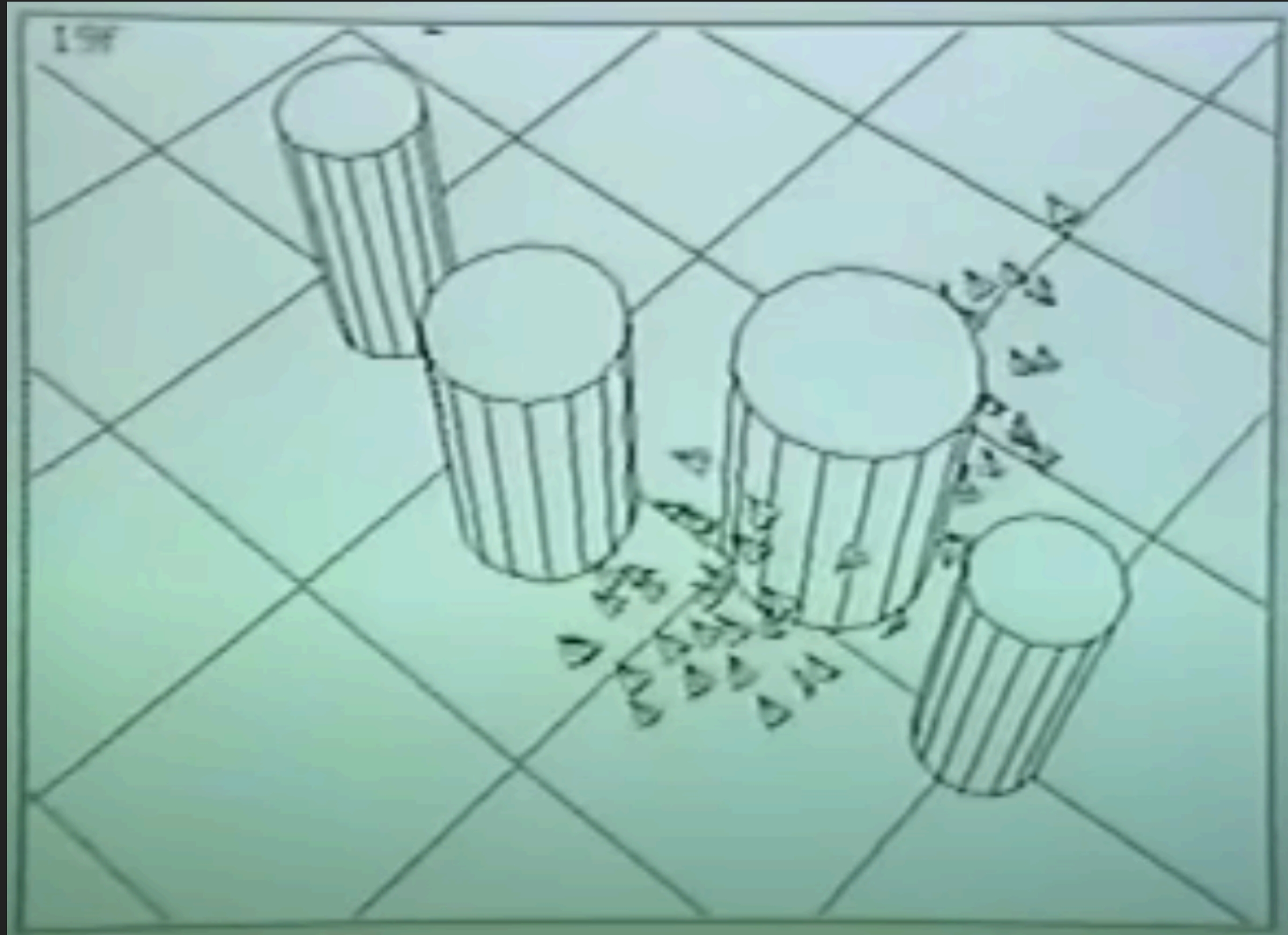
```
// click and modify the underlined values  
gsap.to(graph, { duration: 2.5, ease: "power3.out", y: -500 });
```



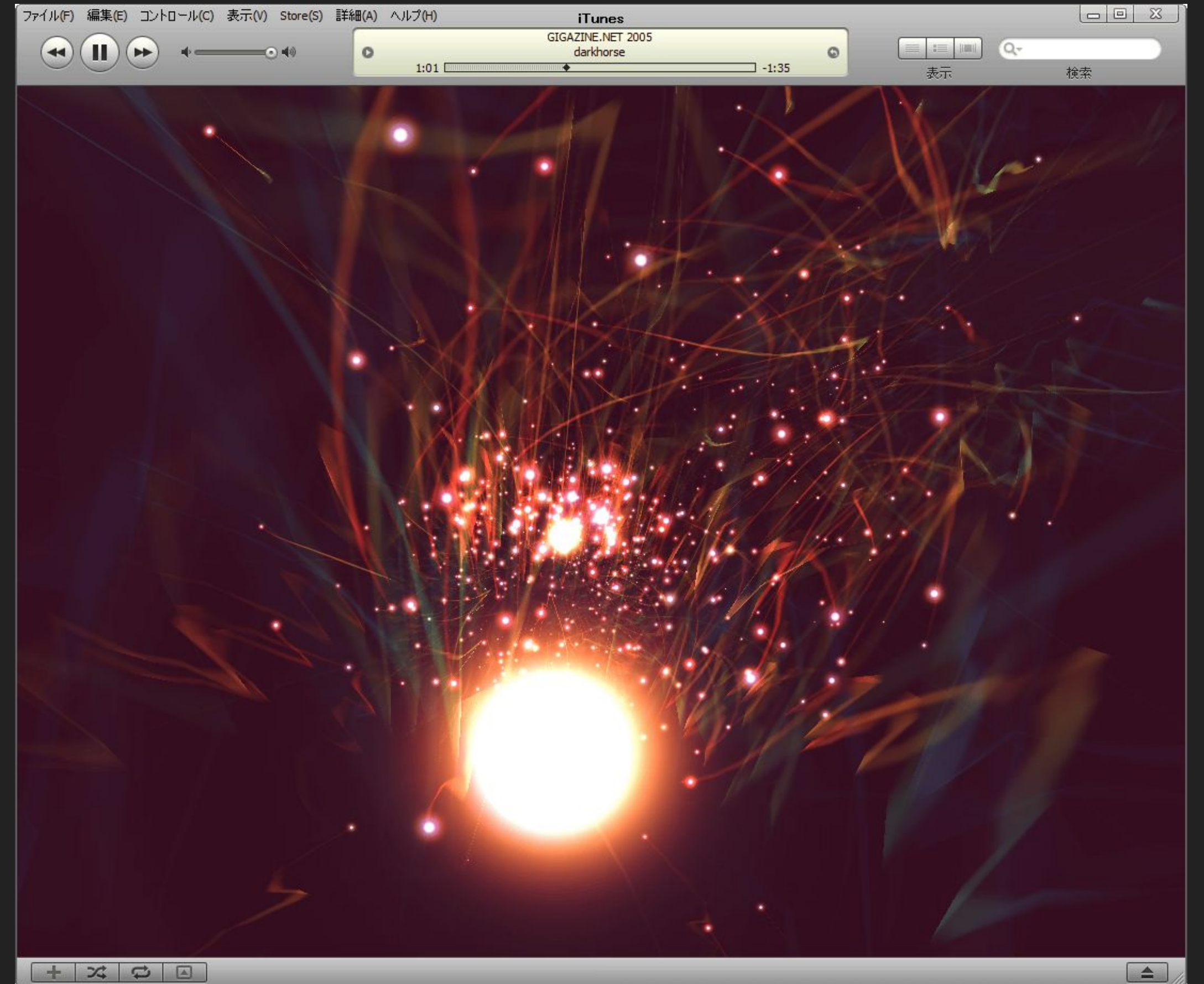
SIMULATION

SIMULATION

Use physics or other rules to determine next frame for one or more objects.

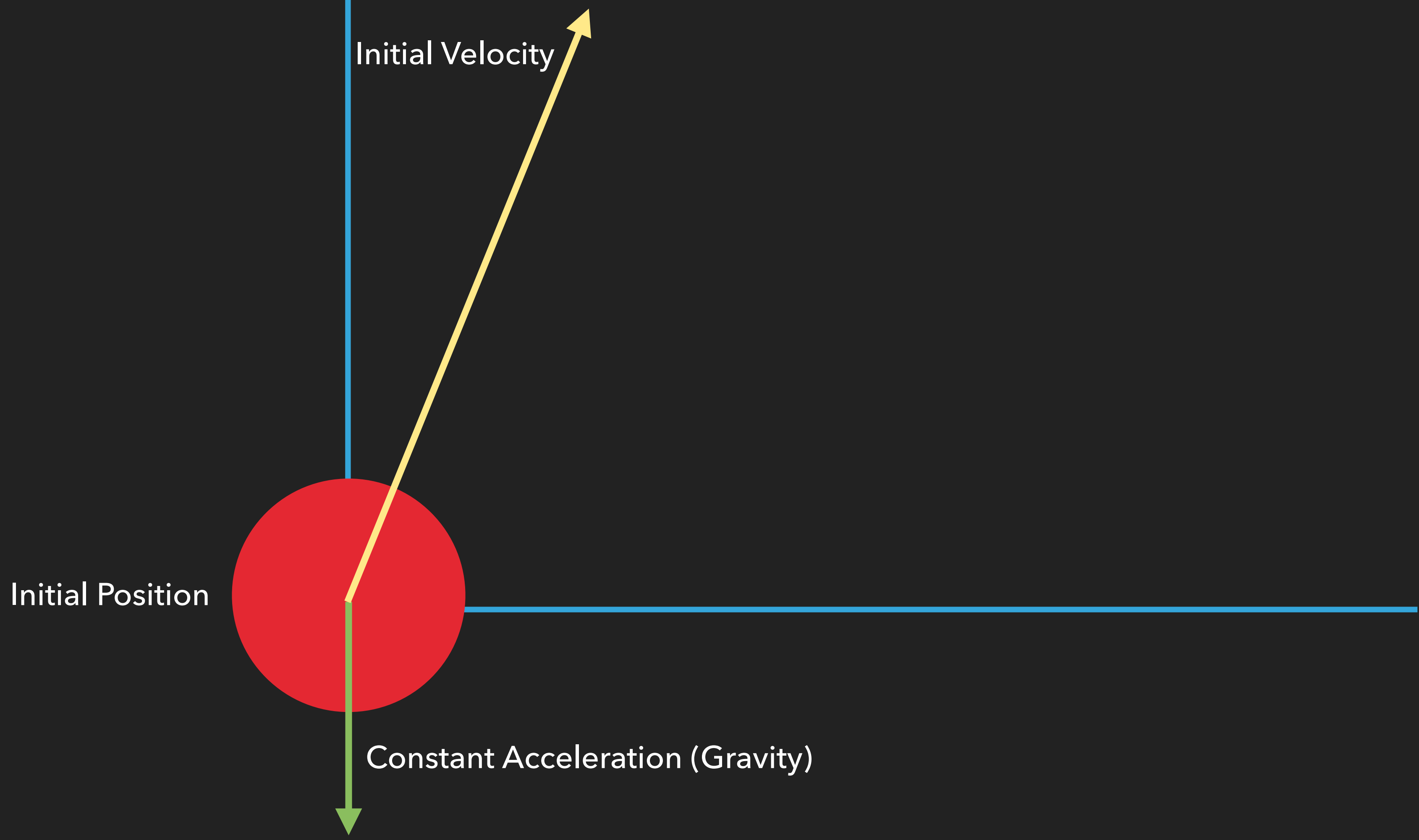


Craig Reynolds' Boids (1986)



Robert Hodgin's (Flight 404)
Magnetosphere, 2007

$T = 0$



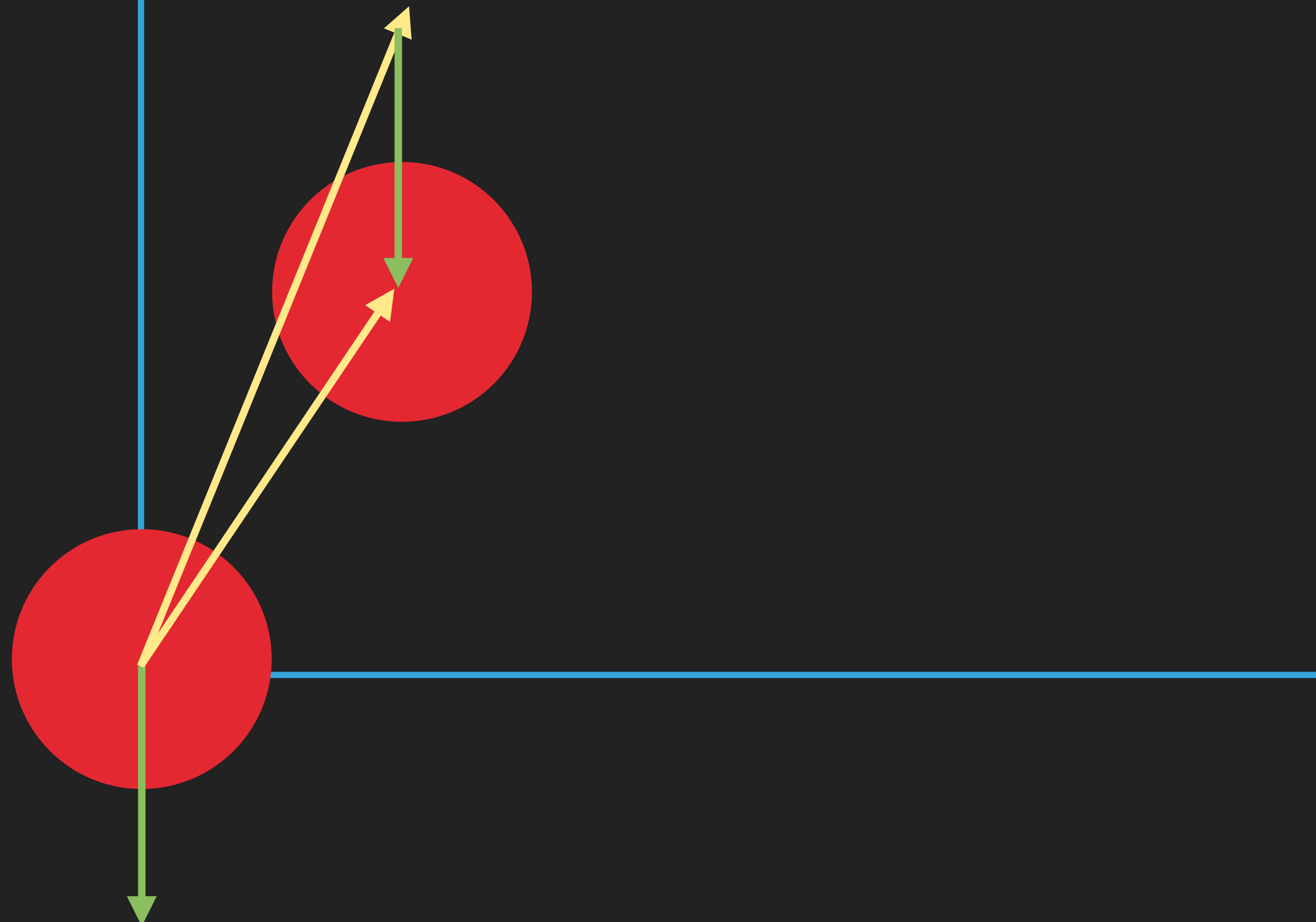
Initial Velocity

Initial Position

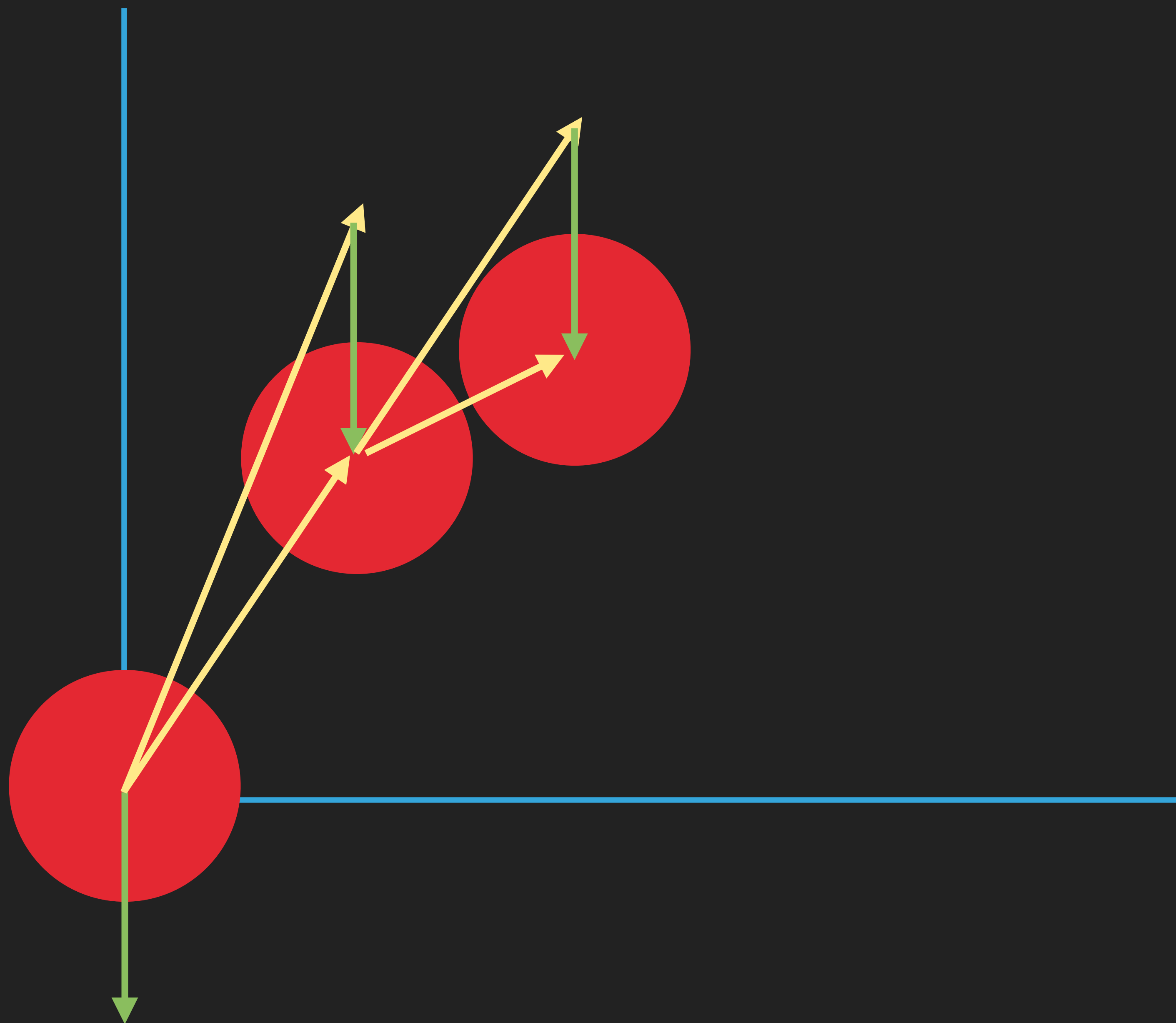
Constant Acceleration (Gravity)

T = 1

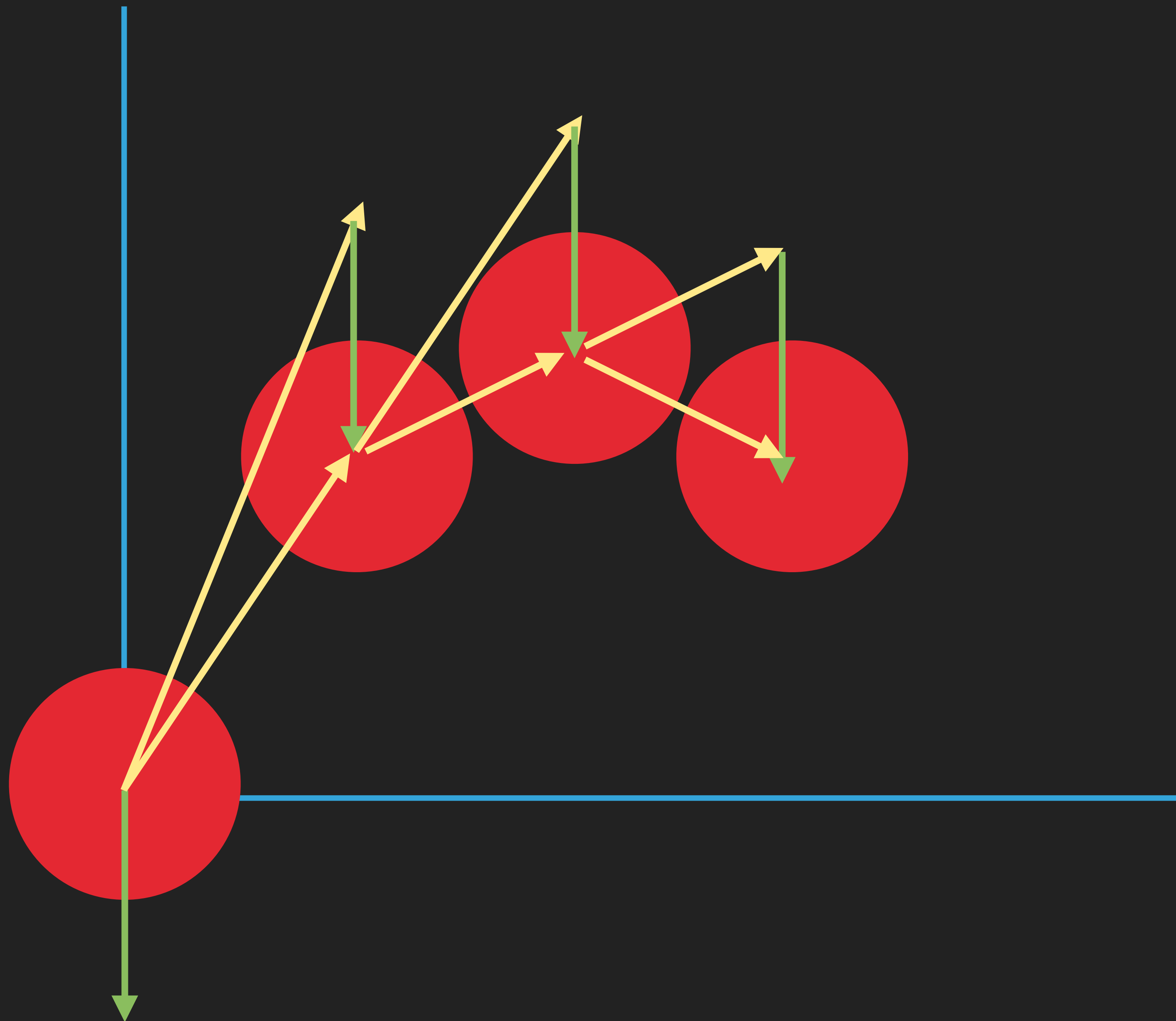
Acceleration is sum of forces acting on particle
Add acceleration to velocity
Add velocity to position



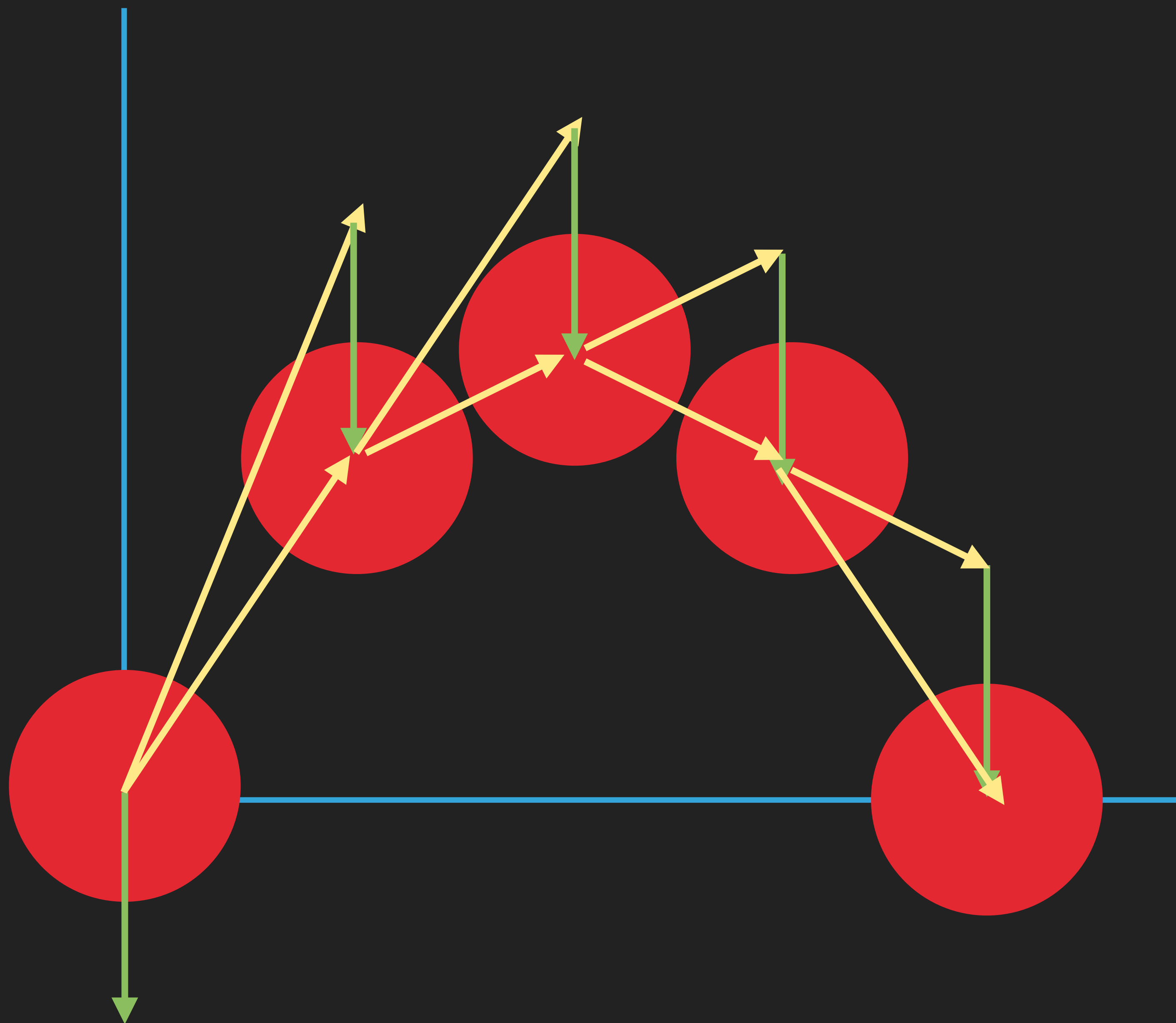
$T = 2$



$T = 3$



$T = 4 \dots$



Can be expanded to three dimensions, multiple particles, and attractive and repulsive forces. But the steps between frames will remain basic vector addition.



<http://roberthodgin.com/project/magnetosphere>

```

5 // Created by Robert Hodgins on 5/14/12.
6 // Copyright (c) 2012 __MyCompanyName__. All rights reserved.
7 //
8
9 #include "cinder/app/AppBasic.h"
10 #include "cinder/Rand.h"
11 #include "cinder/Sphere.h"
12 #include "Particle.h"
13
14 using namespace ci;
15
16 Particle::Particle(){}
17
18 Particle::Particle( const Vec3f &pos, float charge )
19     : mPos( pos ), mCharge( charge )
20 {
21     mVel          = Vec3f::zero();
22     mAcc          = Vec3f::zero();
23     mForce        = 0.0f;
24
25     mRadius       = 1.0f;
26     mShellRadius  = 12.0f;
27 }
28
29 void Particle::update( const Camera &cam, float dt )
30 {
31     Sphere s          = Sphere( mPos, mRadius * 10.0f );
32     mScreenPos        = cam.worldToScreen( mPos, app::getWindowWidth(), app::getWindowHeight() );
33     mScreenRadius     = cam.getScreenRadius( s, app::getWindowWidth(), app::getWindowHeight() );
34
35     mColor            = mCharge * 0.5f + 0.5f;
36
37     mVel += mAcc * dt;
38     mPos += mVel * dt;
39     mAcc = Vec3f::zero();
40
41     mShellRadius = mRadius + fabs( mForce ) * 5000.0f;
42
43     mMatrix.setToIdentity();
44     mMatrix.translate( mPos );
45 }
46
47 void Particle::draw()
48 {
49     gl::color( Color( mColor, mColor, mColor ) );
50     gl::drawSphere( mPos, mRadius );

```

Acceleration is sum of forces acting on particle
 Add acceleration to velocity
 Add velocity to position

```

35     mColor            = mCharge * 0.5f + 0.5f;
36
37     mVel += mAcc * dt;
38     mPos += mVel * dt;
39     mAcc = Vec3f::zero();
40
41     mShellRadius = mRadius + fabs( mForce ) * 5000.0f;

```

“The physics of the simple vehicle model is based on forward Euler integration. At each simulation step, behaviorally determined steering forces (as limited by *max_force*) are applied to the vehicle’s point mass. This produces an acceleration equal to the steering force divided by the vehicle’s mass. That acceleration is added to the old velocity to produce a new velocity, which is then truncated by *max_speed*. Finally, the velocity is added to the old position:

```
    steering_force = truncate (steering_direction,  
max_force)  
    acceleration = steering_force / mass  
    velocity = truncate (velocity + acceleration,  
max_speed)  
    position = position + velocity
```

Acceleration is sum of forces acting on particle
Add acceleration to velocity
Add velocity to position

The simple vehicle model maintains its velocity-aligned local space by *incremental adjustment* from the previous time step.”

The Coding Train - YouTube

youtube.com/c/TheCodingTrain/playlists?view=50&sort=dd&shelf_id=9

shiffman particles coding train

YouTube

Home

Trending

Subscriptions

Library

The Coding Train ✓
1.16M subscribers

JOIN SUBSCRIBED

HOME VIDEOS **PLAYLISTS** COMMUNITY CHANNELS ABOUT

The Nature of Code: Simulating Natural Systems with Processing ▾

- 1: Vectors - The Nature of Code**
The Coding Train ✓
VIEW FULL PLAYLIST
- 2: Forces - The Nature of Code**
The Coding Train ✓
VIEW FULL PLAYLIST
- 3: Oscillation - The Nature of Code**
The Coding Train ✓
VIEW FULL PLAYLIST
- 4: Particle Systems - The Nature of Code**
The Coding Train ✓
VIEW FULL PLAYLIST
- 5: Physics Libraries - The Nature of Code**
The Coding Train ✓
VIEW FULL PLAYLIST
- 6: Autonomous Agents - The Nature of Code**
The Coding Train ✓
VIEW FULL PLAYLIST
- 7: Cellular Automata - The Nature of Code**
The Coding Train ✓
VIEW FULL PLAYLIST
- 8: Fractals - The Nature of Code**
The Coding Train ✓
VIEW FULL PLAYLIST
- 9: Genetic Algorithms - The Nature of Code**
The Coding Train ✓
VIEW FULL PLAYLIST

https://www.youtube.com/watch?v=qMq-zd6hguc&list=PLRqWx-V7Uu6bR4BcLjHHTopXitsjRA7yG

Pretty good source in-house

