EXISTING IN TIME

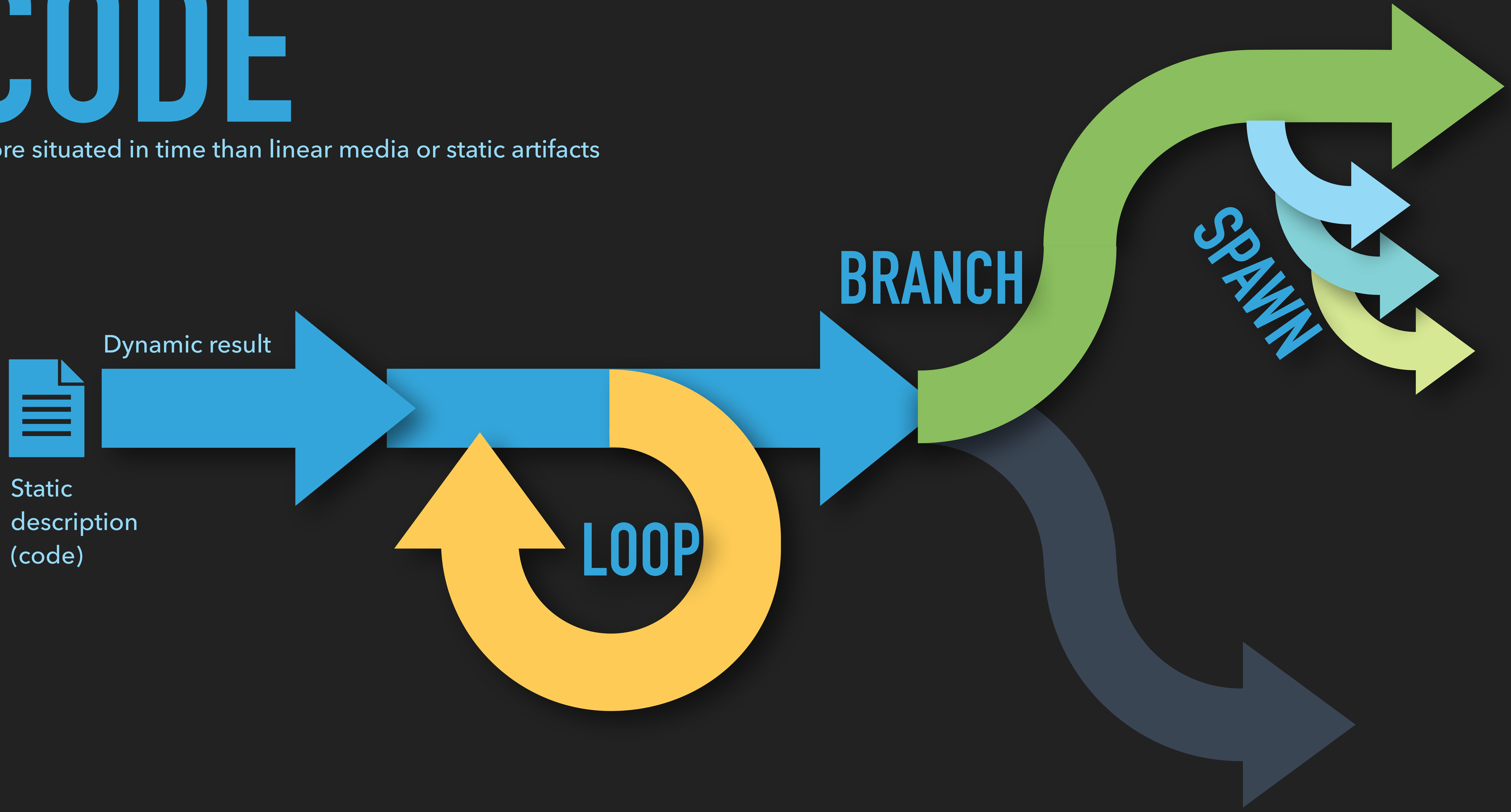

OFF


Soundcraft Vi6


SEASENSE
Marine / Sport
HORN
Recommended for boats up to 12 meters / 39.4 feet
Ideal for all sporting events
WARNING: Use only as directed. Do not spray near ignition sources, open flames, sparks or hot surfaces. Use only with adequate ventilation. Intentional misuse by deliberately concentrating and inhaling contents can be harmful or fatal.
KEEP OUT OF REACH OF CHILDREN.

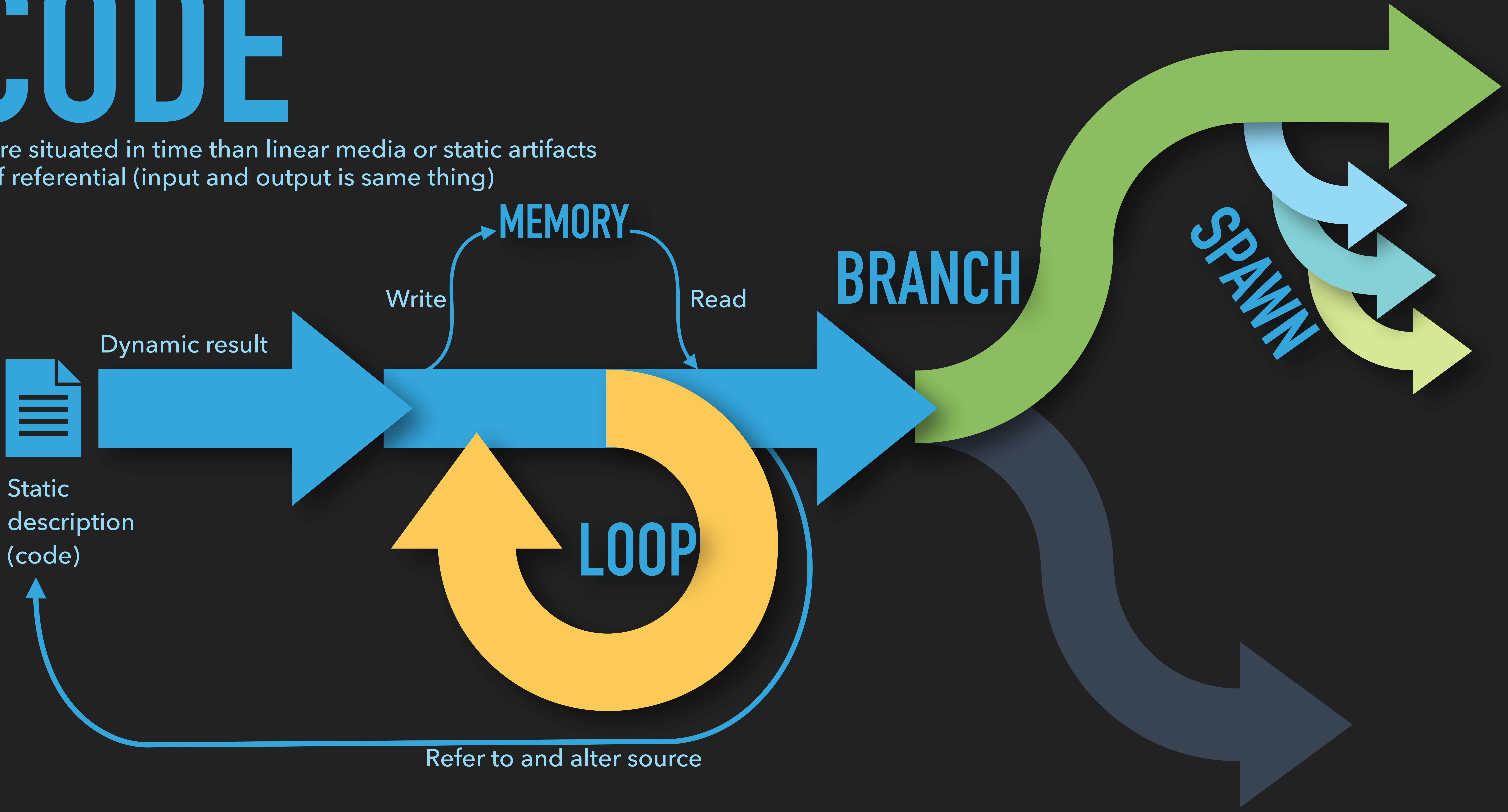
CODE

More situated in time than linear media or static artifacts



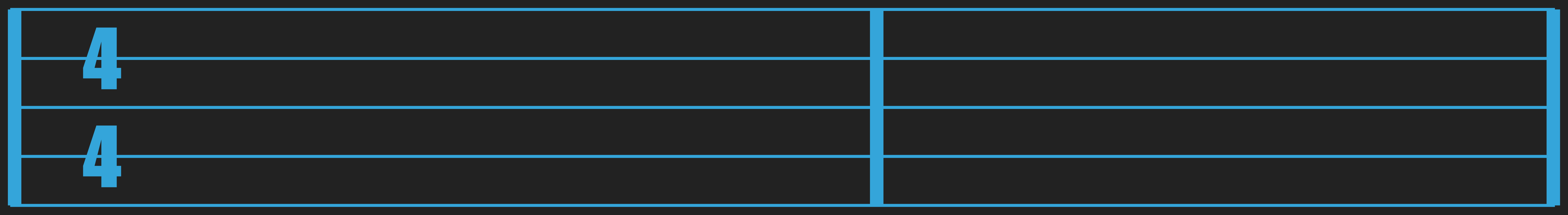
CODE

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



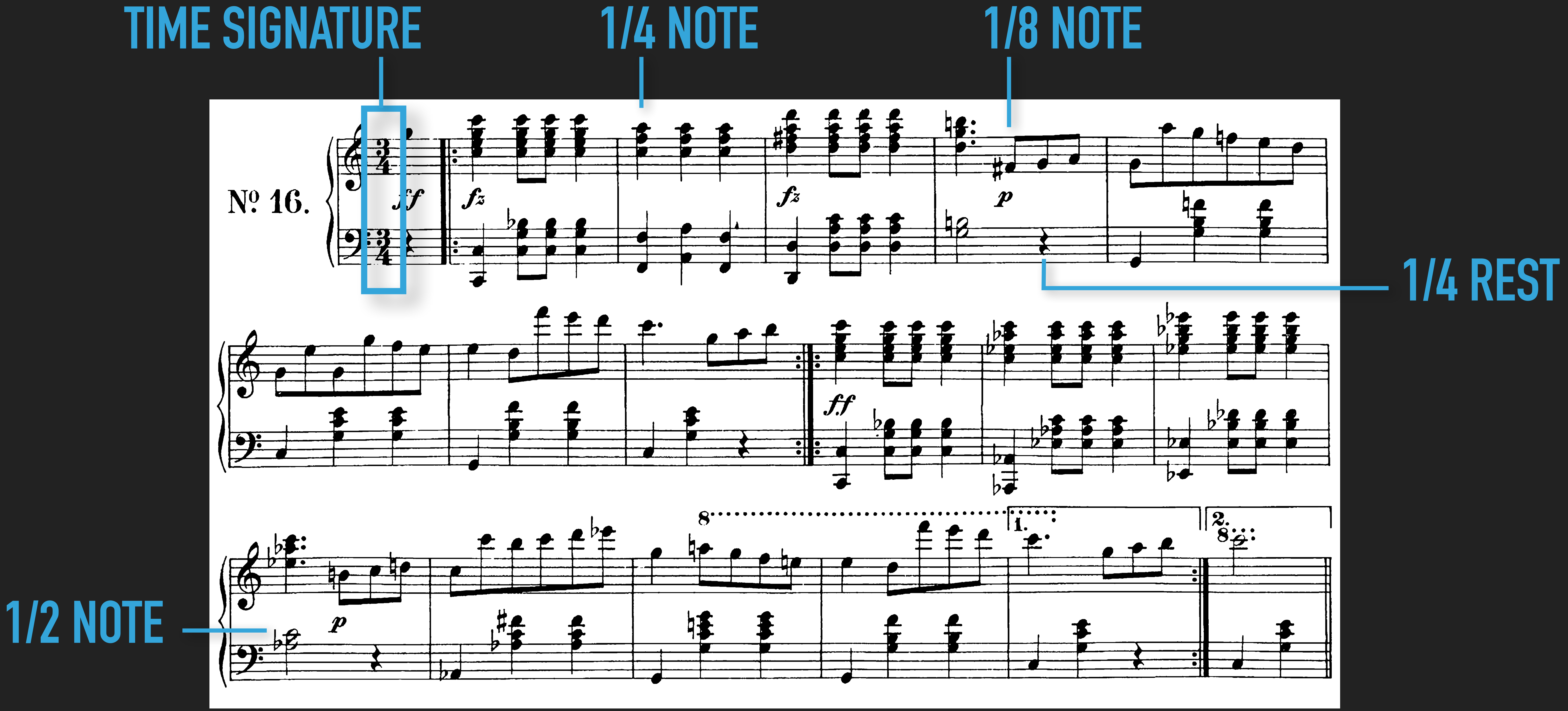
~~CODE~~ MUSIC

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



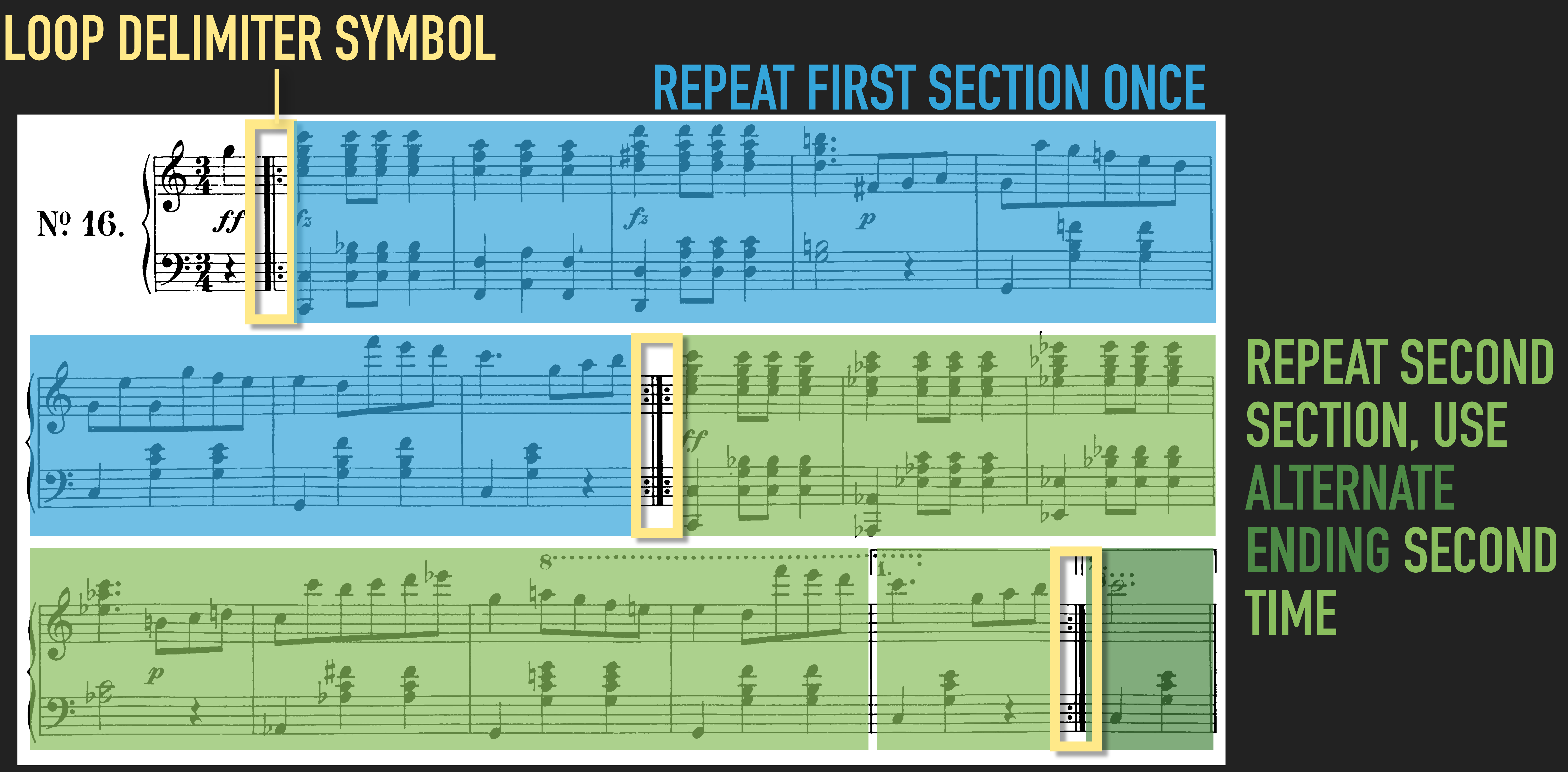
~~CODE~~ MUSIC

Symbols for representing time



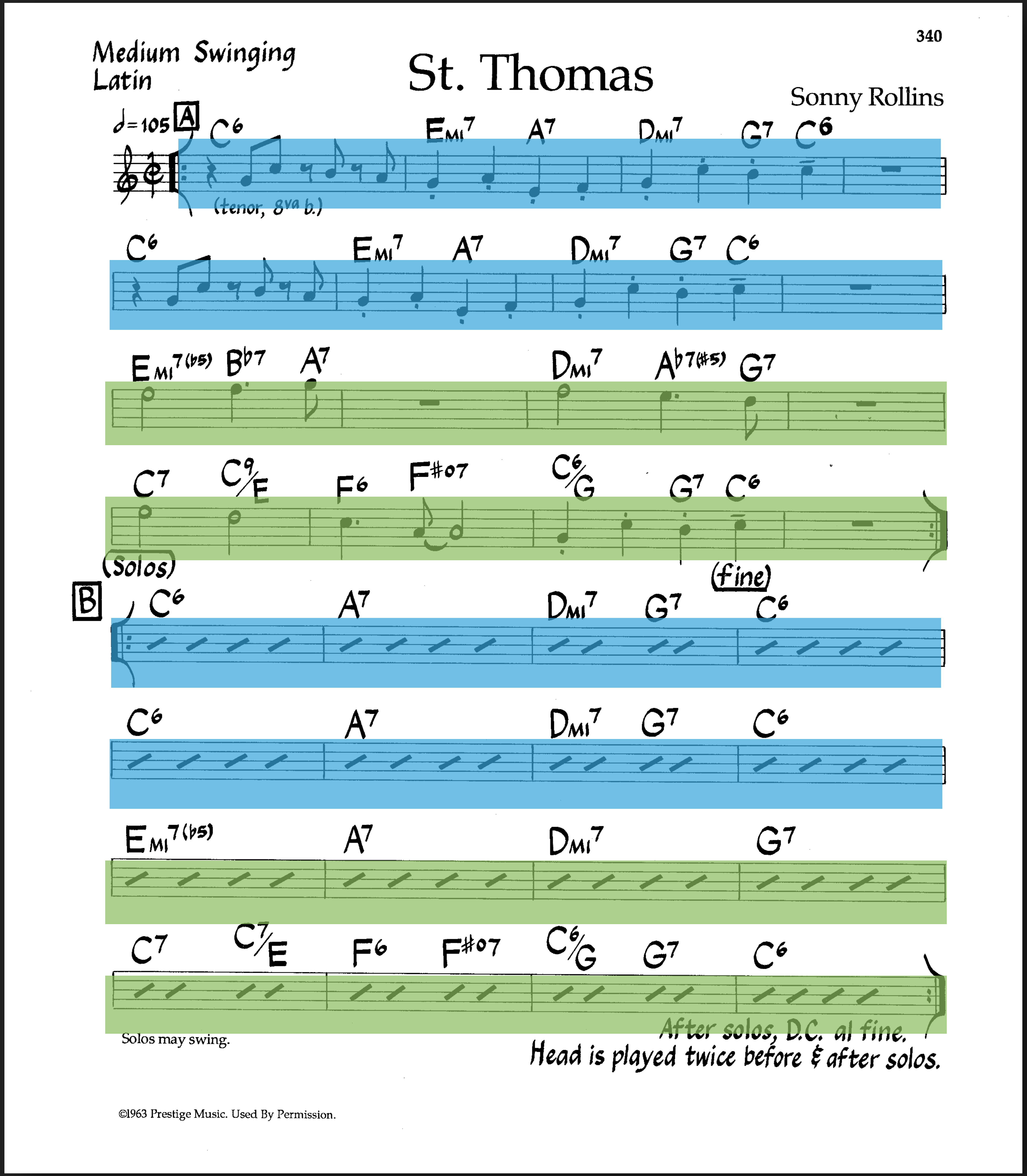
~~CODE~~ MUSIC

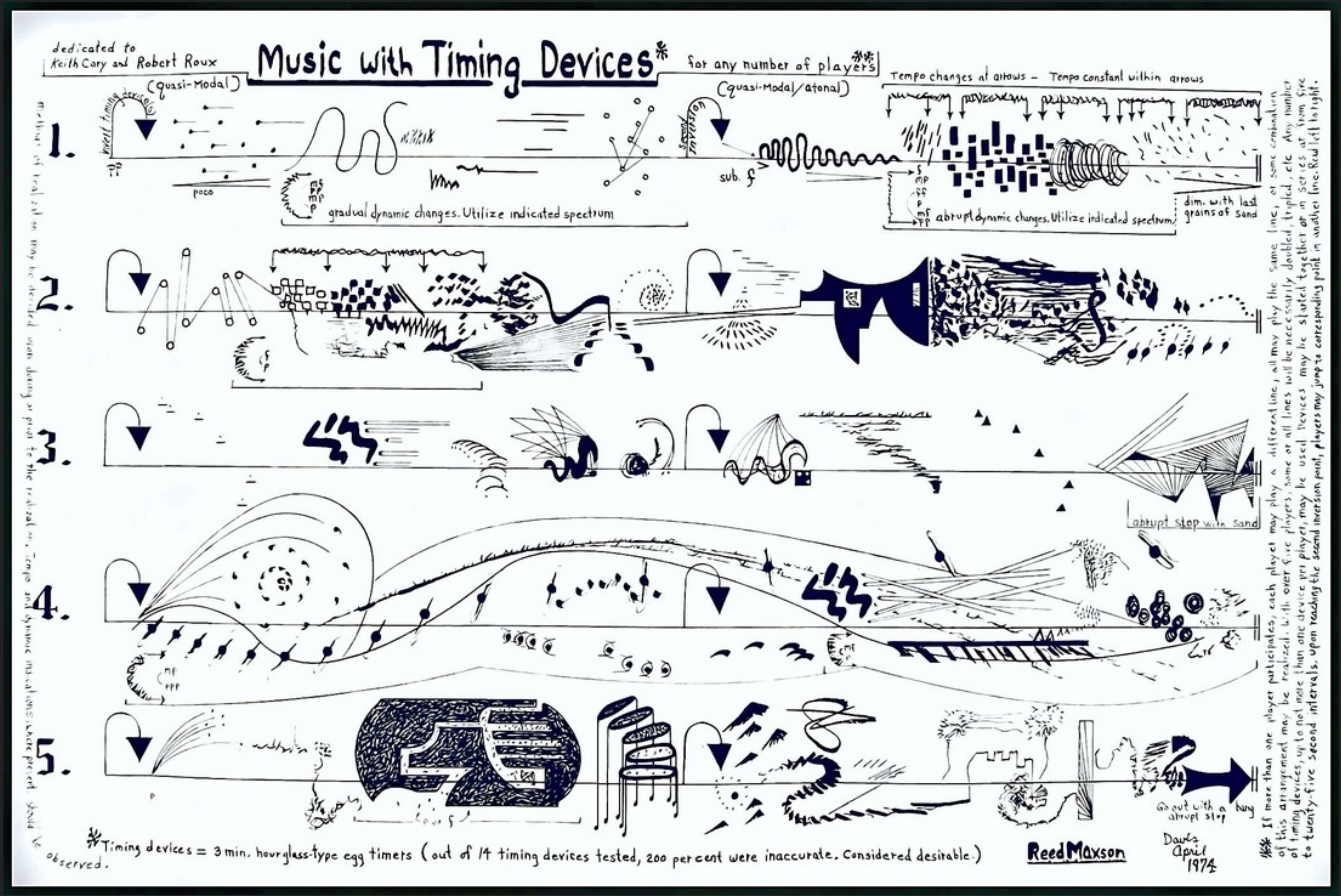
Symbols for structuring execution



~~CODE~~ MUSIC

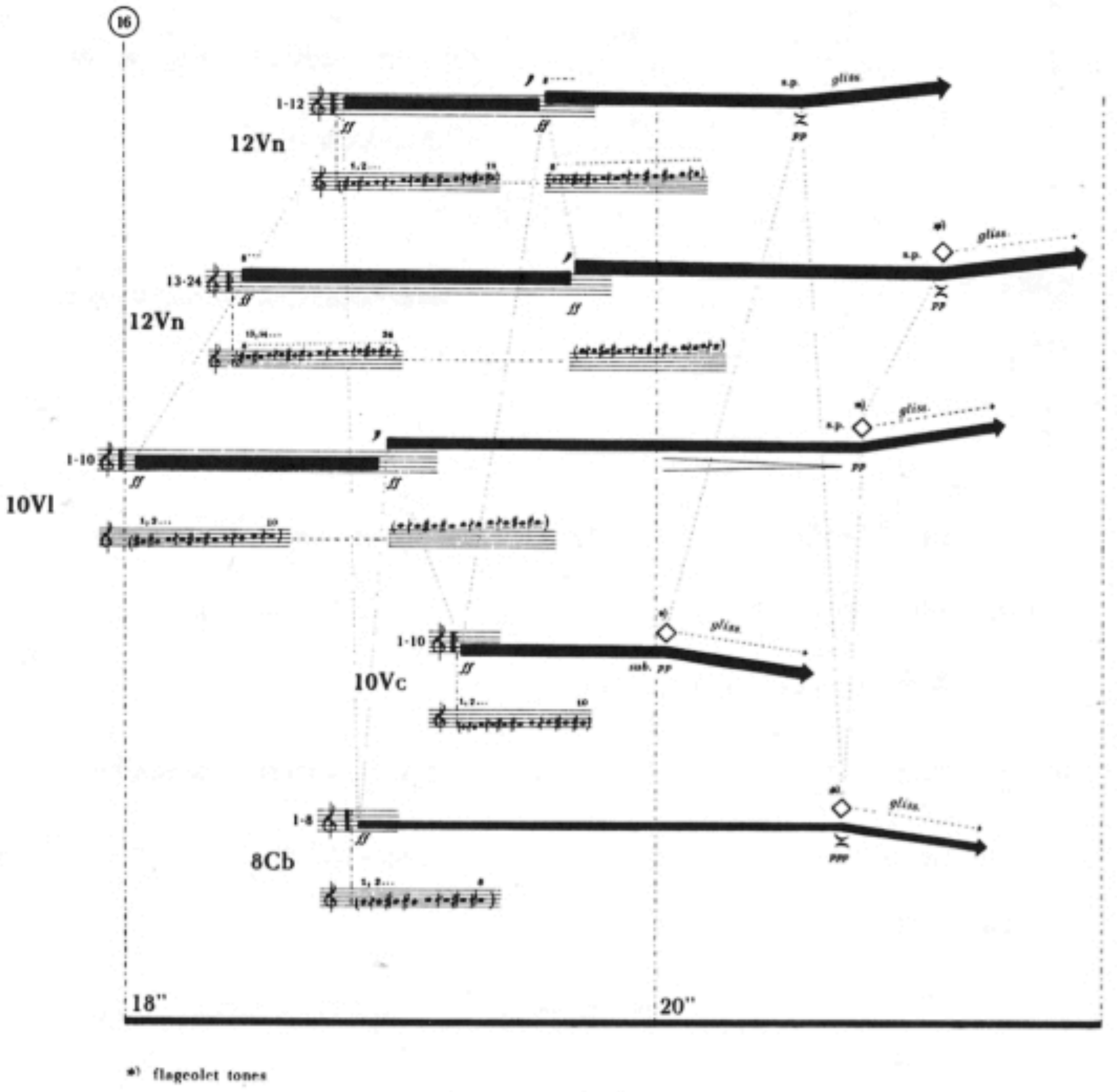
Simple outline can be decoded in sophisticated way by specially-trained agents aka 'musicians'

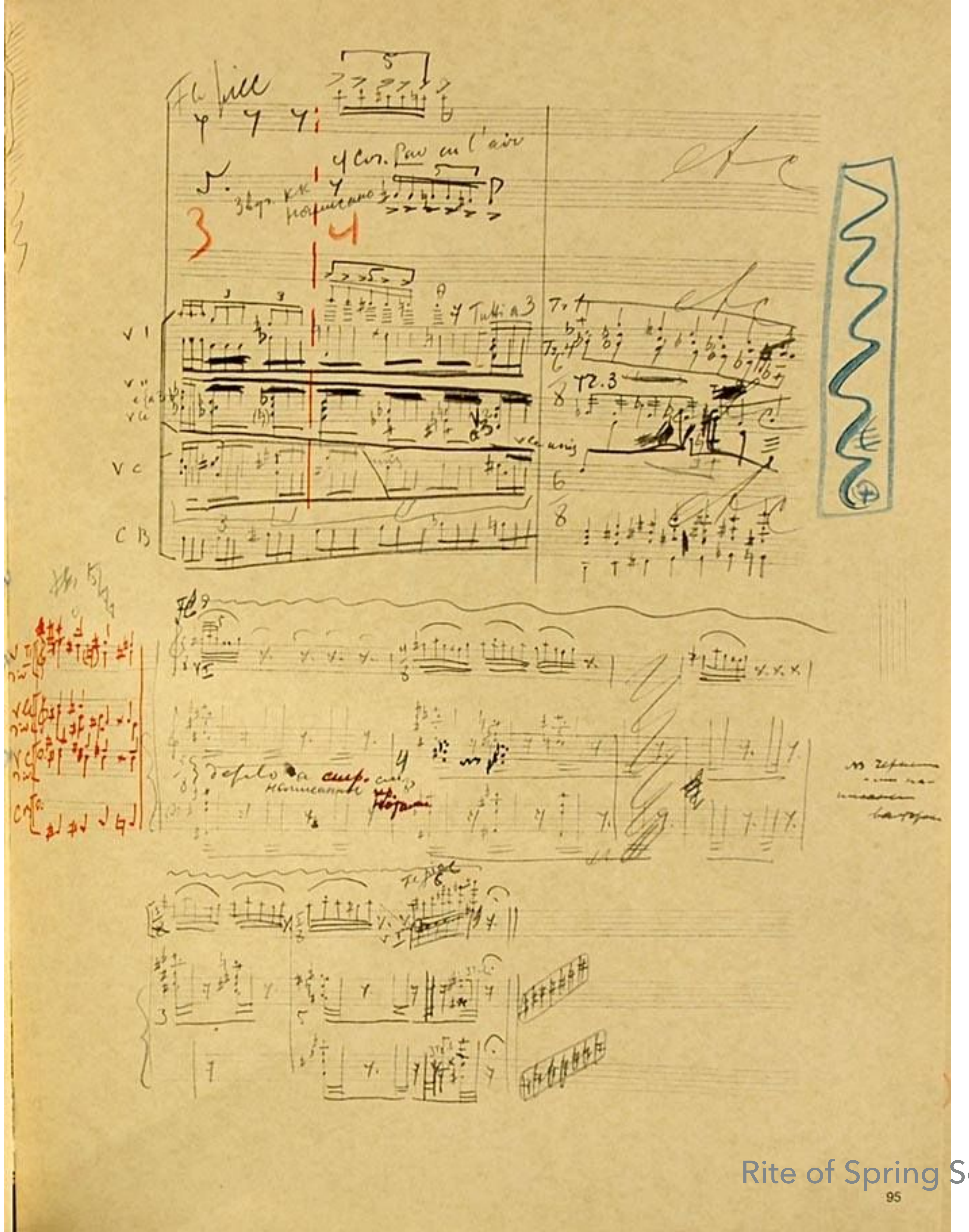




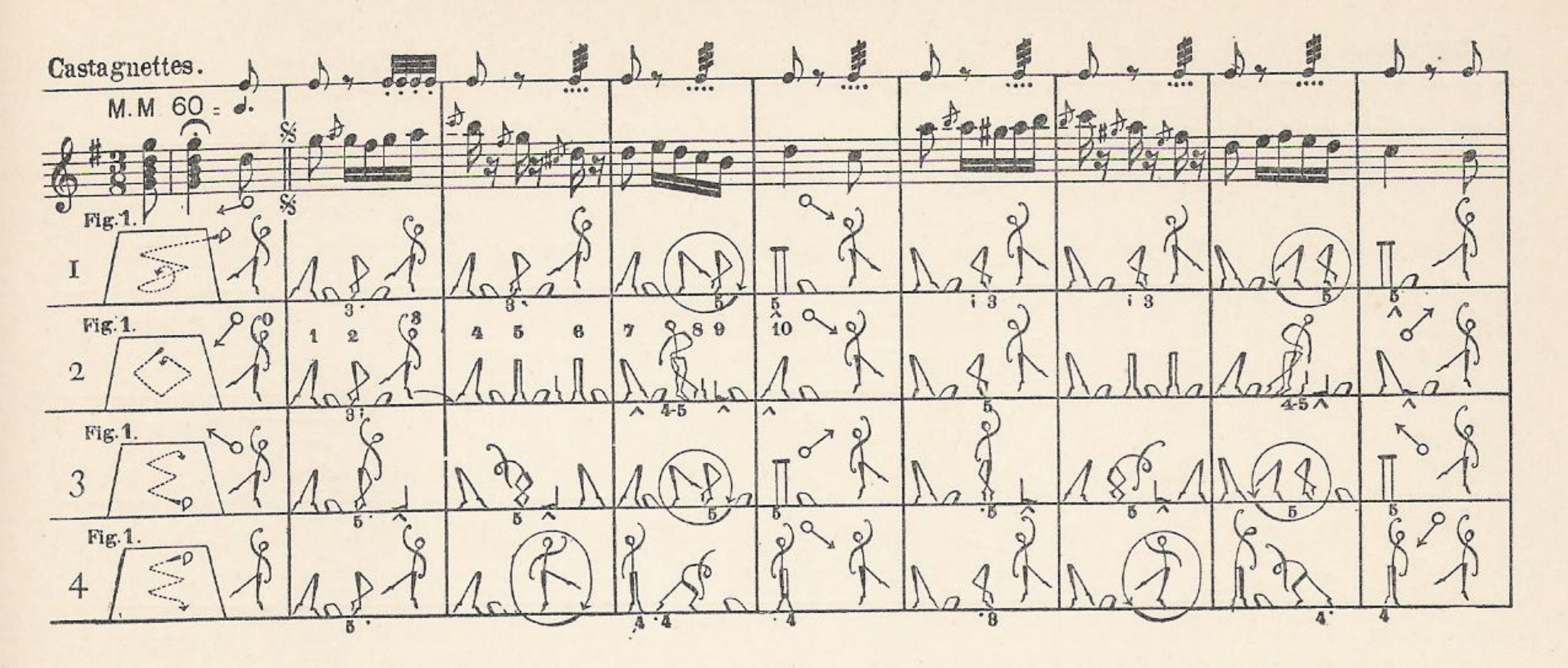
https://www.reedmaxson.com/graphic-scores.html

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

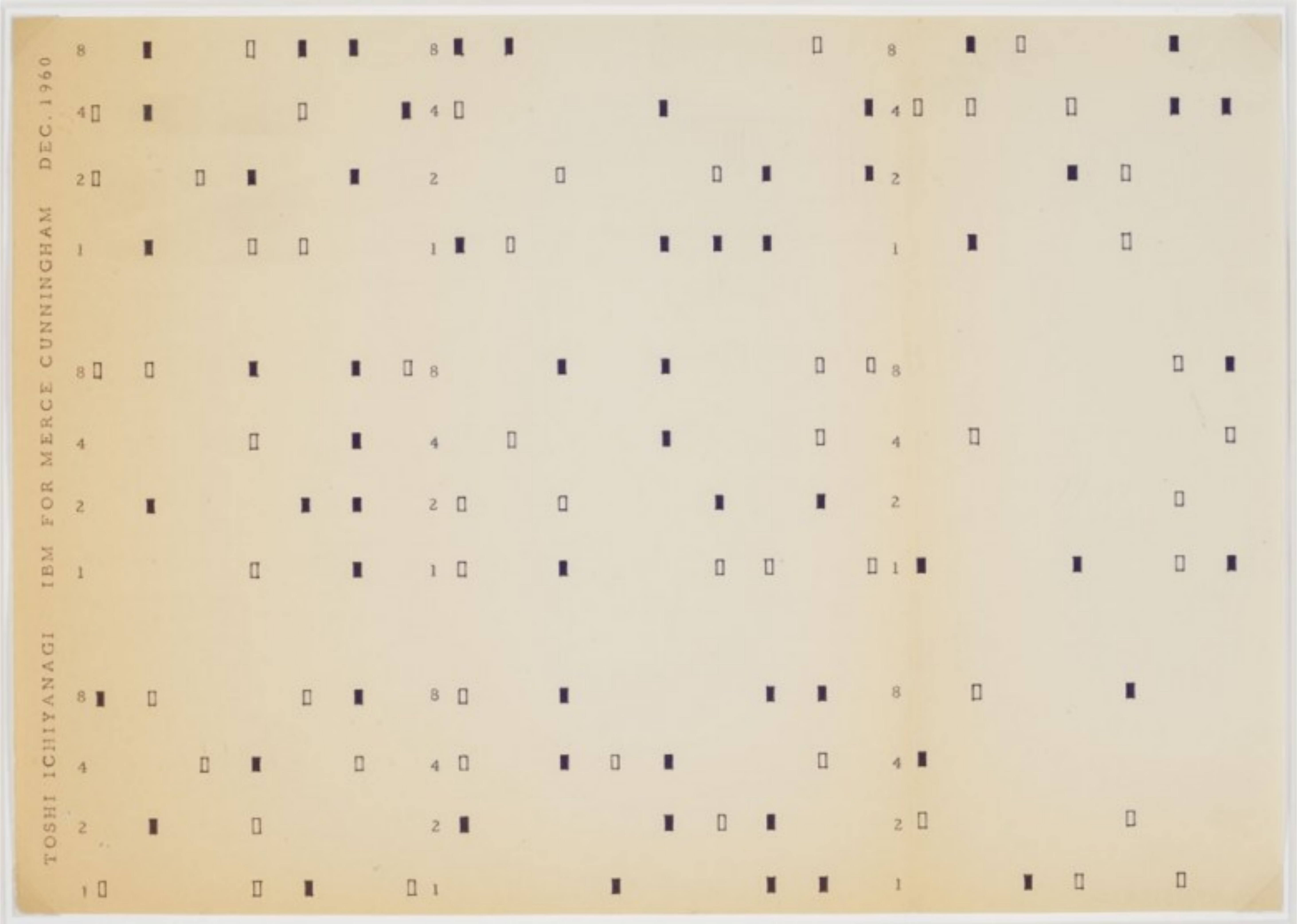




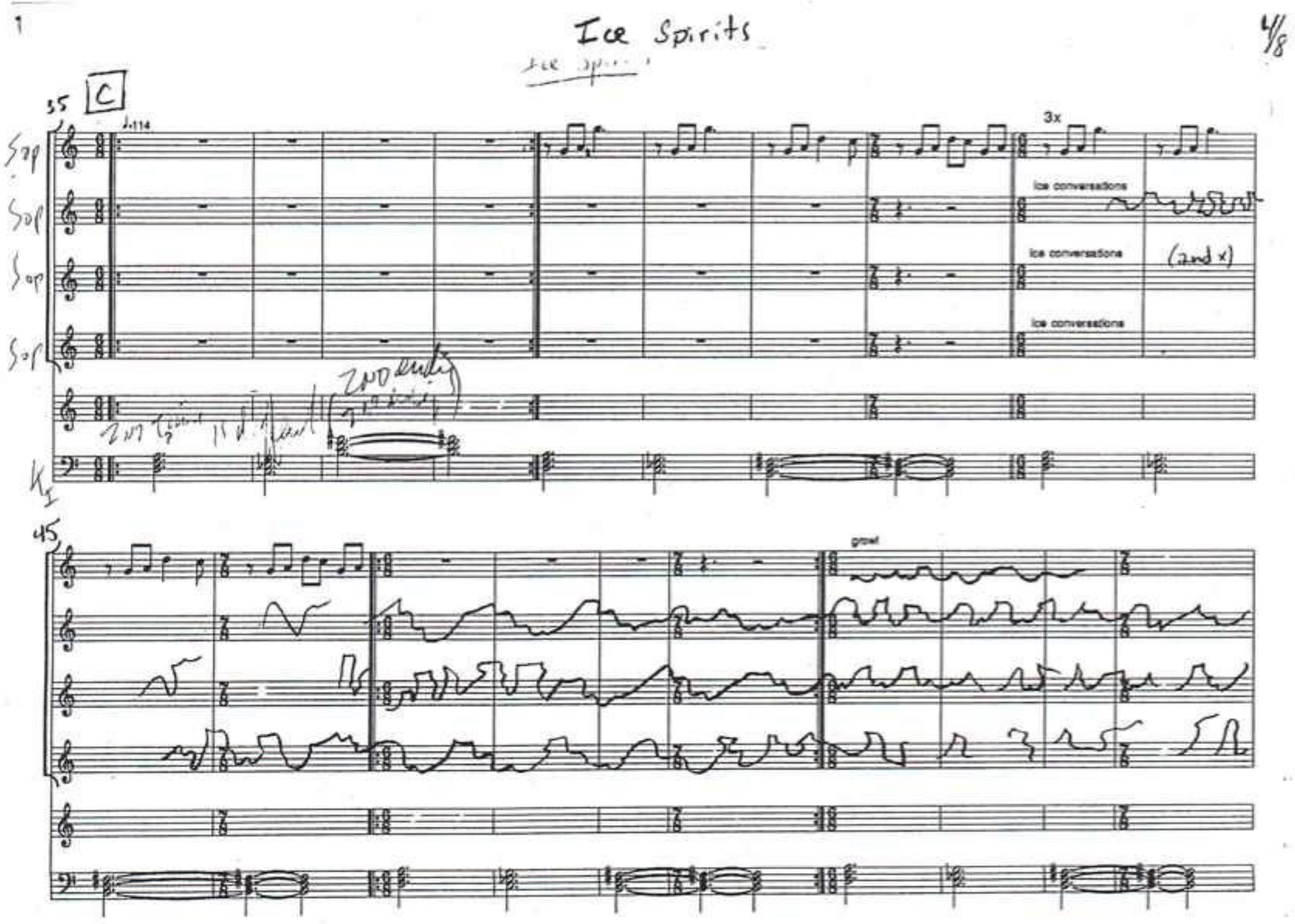
Rite of Spring Score, Igor Stravinsky



La Cachucha, by Friedrich Albert Zorn (wikipedia)



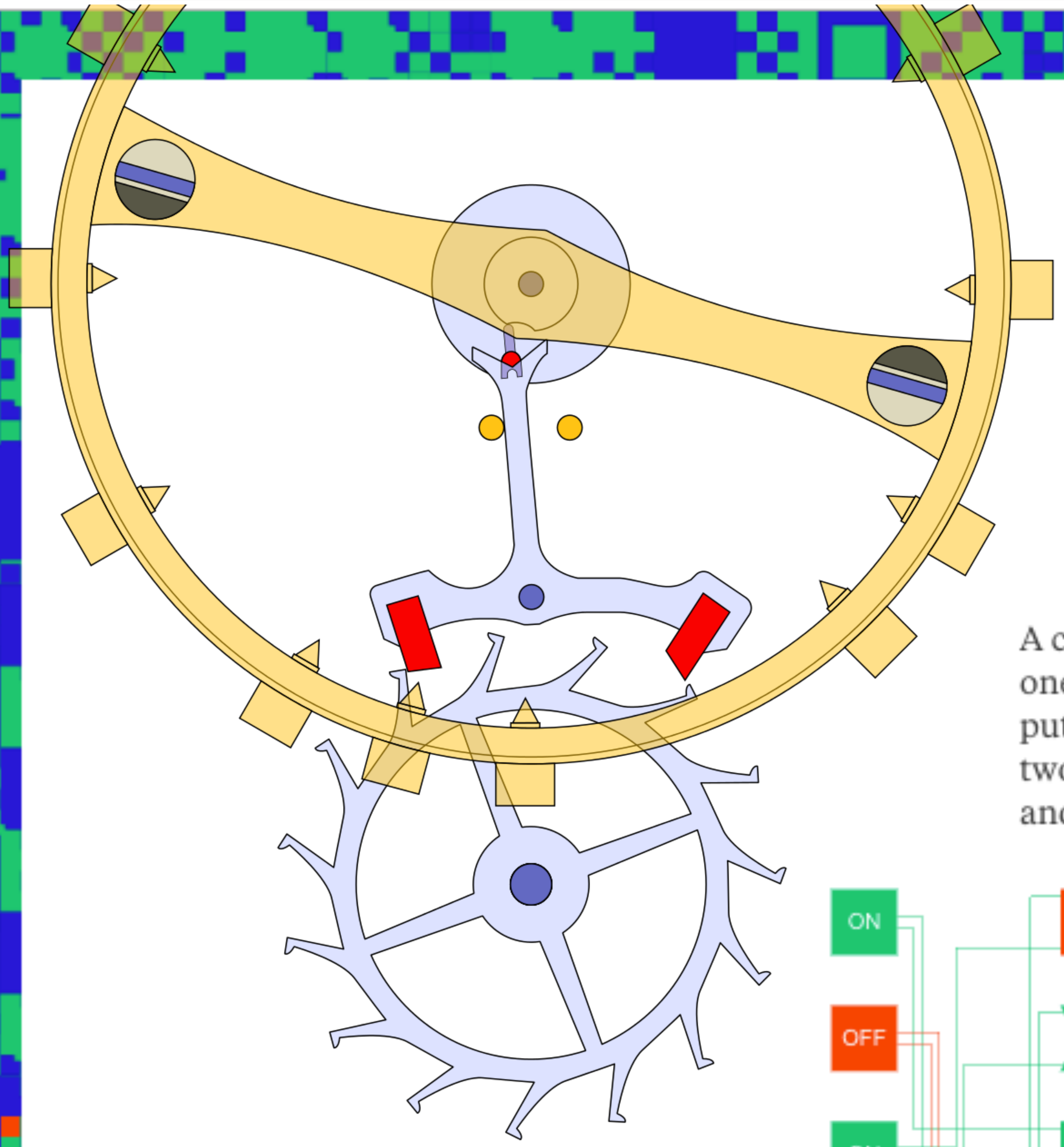
https://www.moma.org/explore/inside_out/2012/12/21/exhibiting-fluxus-keeping-score-in-tokyo-1955-1970-a-new-avant-garde/

GRAPH 2
482 × 283
John Cage 4'33"
5.4525
5.5025
5.5525 5.5625
POUR WATER FROM ONE RECEPTACLE TO ANOTHER
POUR WATER
6.3025
SIREN-WHISTLE
RYOANJI
JOHN CAGE
RYOANJI
JOHN CAGE
ORGAN²/ASLSP
4'33"
SOLO FOR VOICE 95
4'33"
FOR ANY INSTRUMENT OR COMBINATION

Score of Ice Spirits, Meredith Monk

Fine Tune Keyframes
1920 x 1080 (1.00)
Δ 0:00:02:01, 24.00 fps
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
Effects
MVI_9378 * MVI_9456.MOV
Video Effects
Motion
Opacity
Time Remapping
Brightness & Contrast
Brightness
98,6
Velocity: 22,8 / second
Contrast
18,0
Black & White
00:00:41:20
Project: generiek2
generiek2.prproj
49 Items
Media Browser
Favorites
Local Drives
Mac HD
Network Drives
MVI_9405.MOV
MVI_9406.MOV
MVI_9411.MOV
MVI_9413.MOV
MVI_9419.MOV
Audio 1
Synth Lead
Read
Volume
Snap:
Smart
Drag:
No Overlap
Edit
Options
View
Single
Tracks
All
Audio
Inst
Aux
Bus
Input
Output
Master
MIDI
Bass DI.01
Ac GTR U87 1
AC GTR
EG 11R.01
EG FX
Pad.01
12 LeadVoxENG
13 Back ENG.01
Intro
Vrs1a
Vrs1b
Vrs1c
Gil Gowing

Paul Ford: What Is Code? | Bloo

Bloomberg LP [US] | bloomberg.com/graphics/2015-paul-ford-what-is-code/

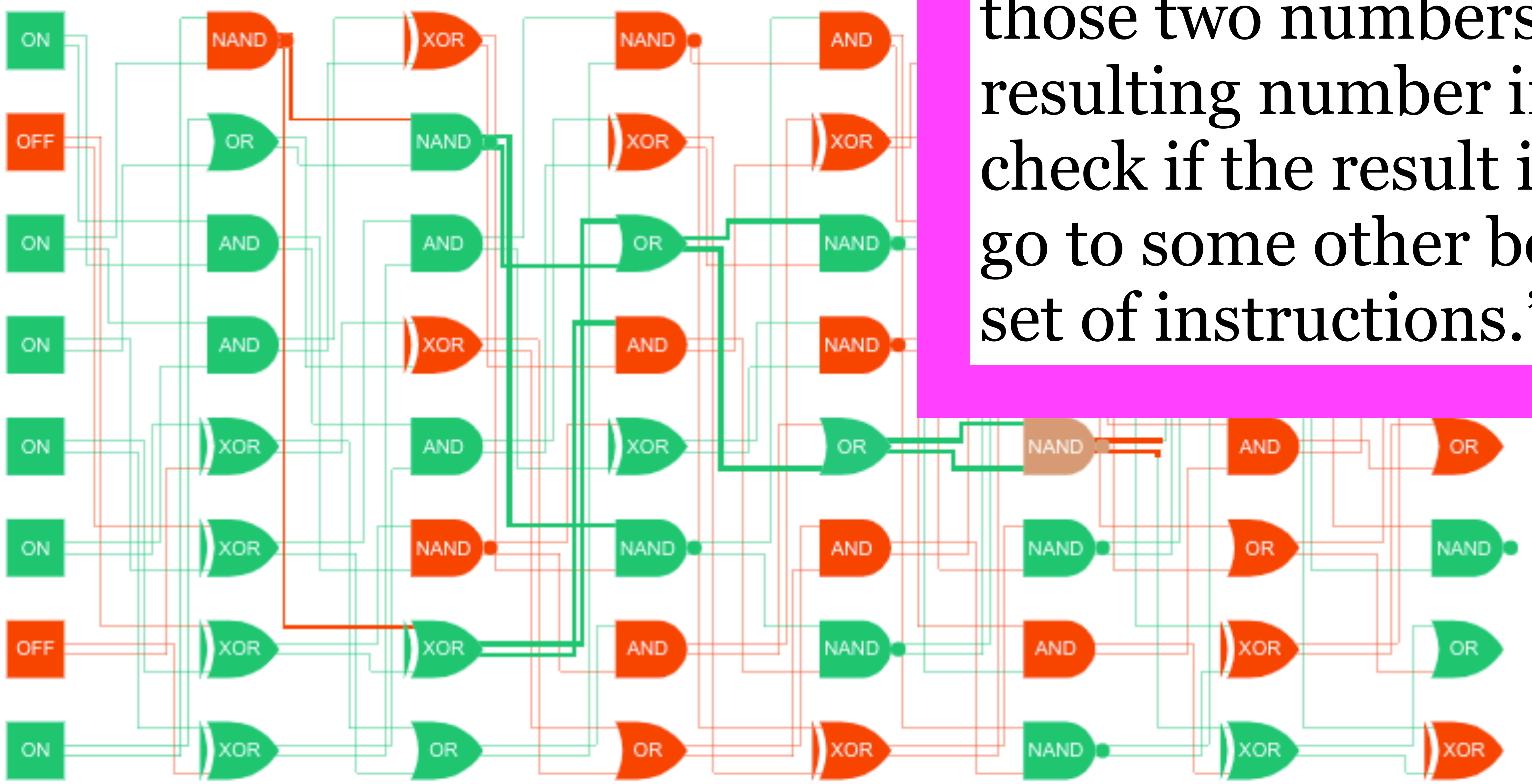


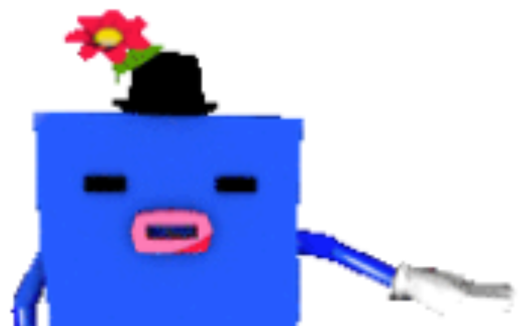
2

Let's Be

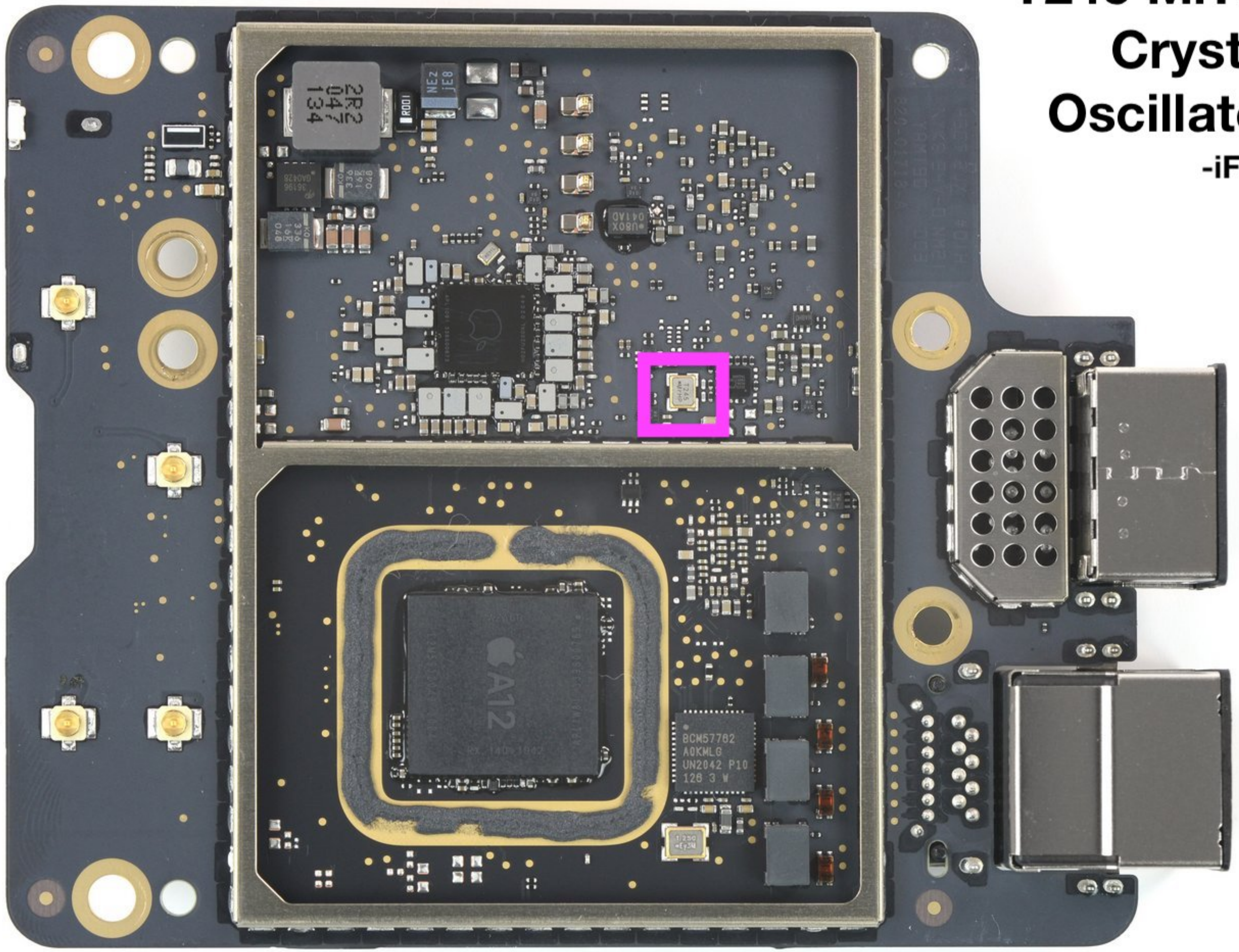
A computer is a clock with benefits. They all work the sa
one step at a time: Tick, take a number and put it in box
put it in box two. Tick, *operate* (an operation might be a
two numbers and put the resulting number in box one. T
and if it is, go to some other box and follow a new set of

"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."

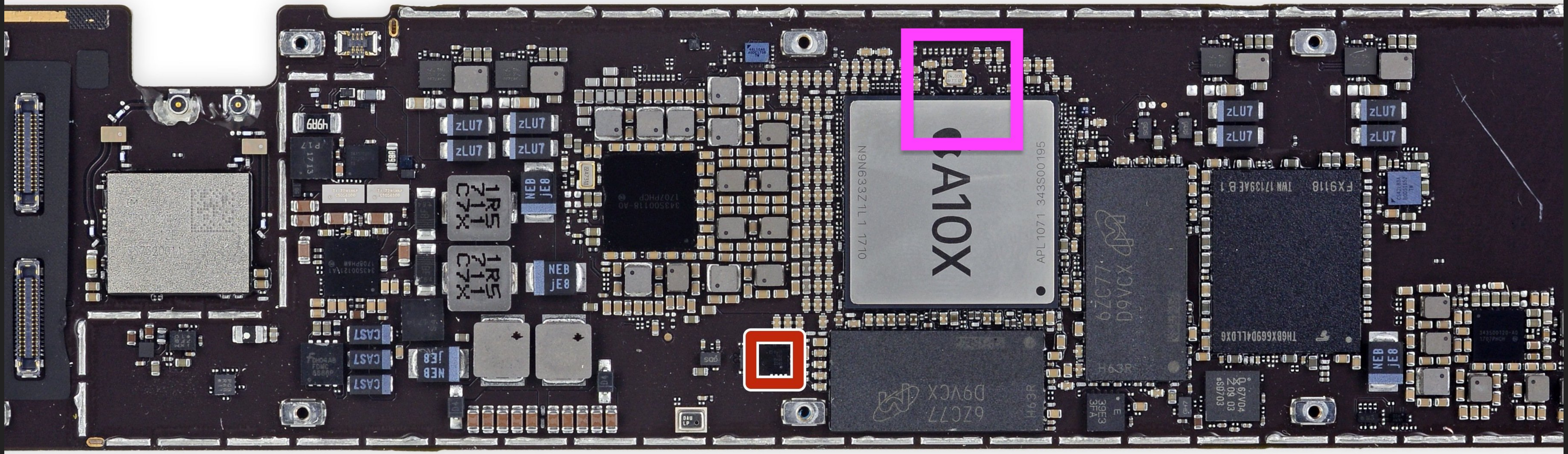




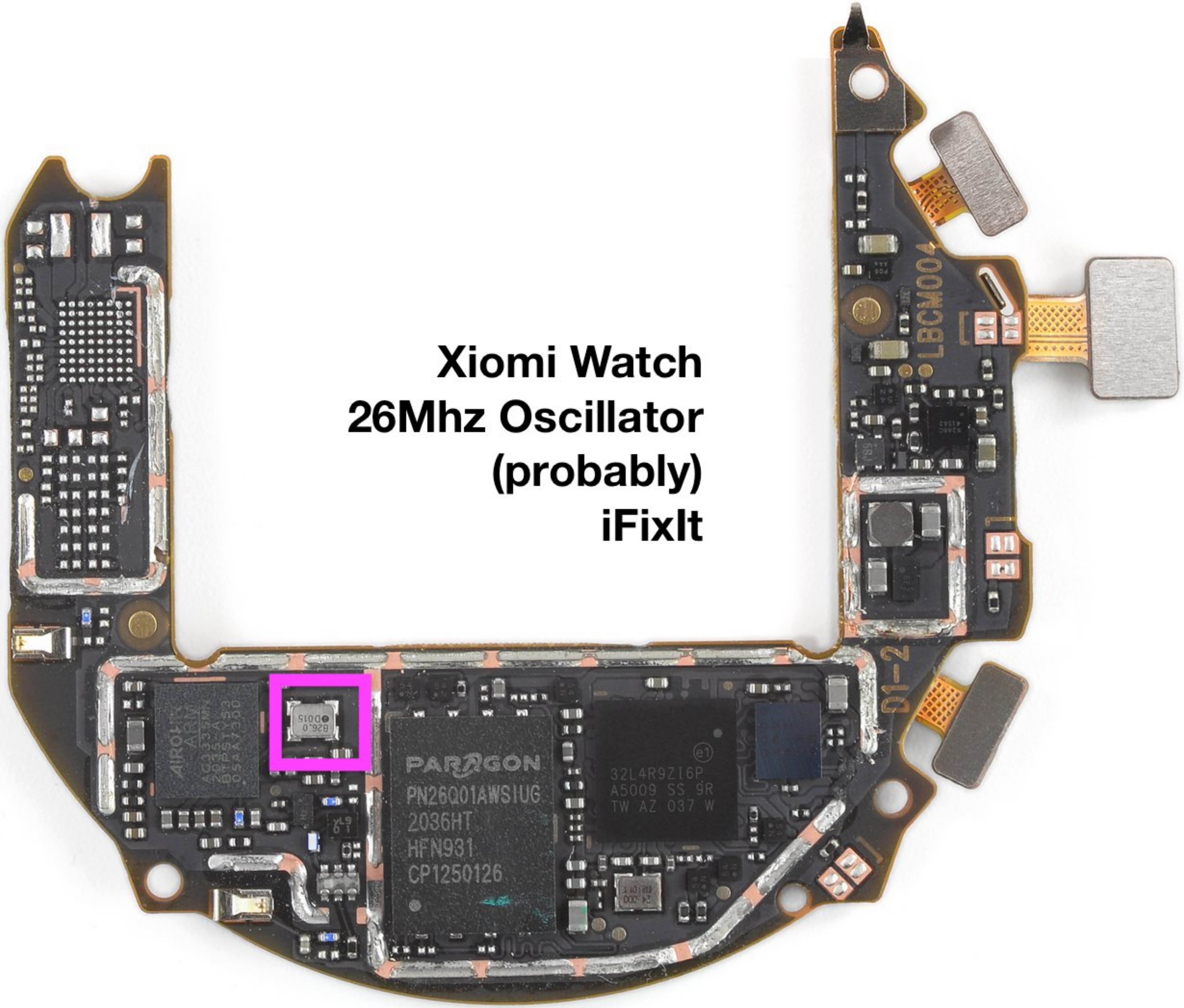
This is simulated circuitry that's computing as you watch. The switches on the left turn the current on and off at random, and the logic gates direct the flow of the current. Click the boxes to change the circuits. Enough of these can compute anything computable.



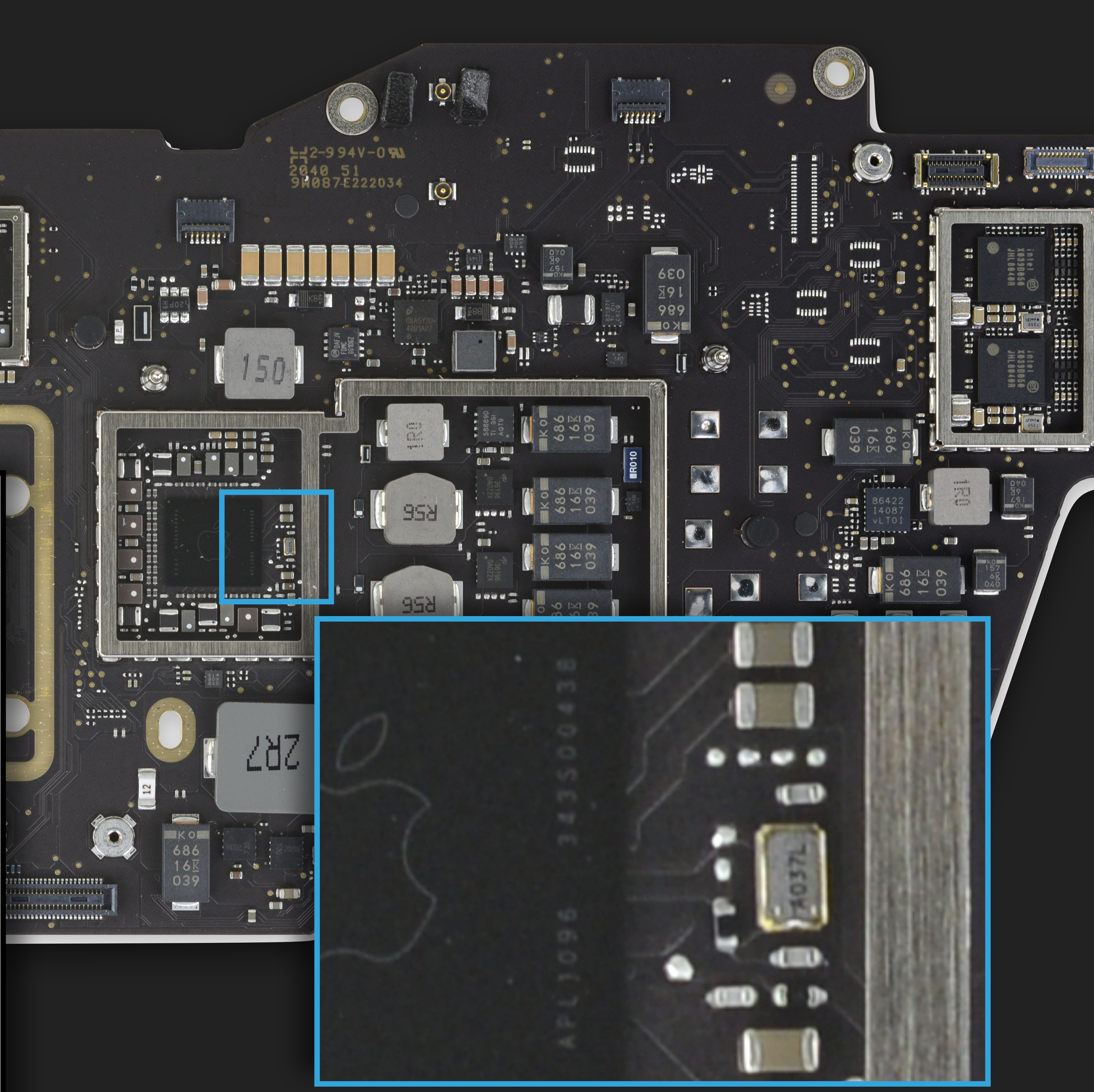
Crystal
Oscillator
-iFixIt


A10X


24.000



Xiomi Watch
26Mhz Oscillator
(probably)
iFixIt



NBVSBAXXX Series

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

ON Semiconductor®

http://onsemi.com

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.

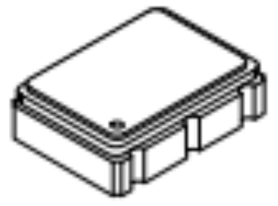
The NBVSBAXXX series are members of ON Semiconductor's PureEdge™ clock family that provides accurate and precision clock generation solutions.

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

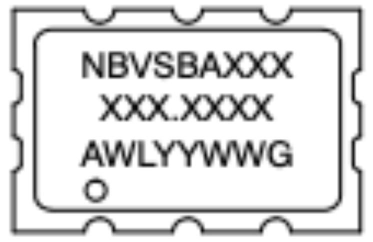
- LVPECL Differential Output
- Operating Range: 2.5 V ±5%, 3.3 V ±10%

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
void setup() {
// put your setup code here, to run once:
}
void loop() {
// put your main code here, to run repeatedly:
}
ARDUINO
sketch_200929a | Processing 3.5.3
Java
sketch_200929a
void setup() {
}
void draw() {
}
PROCESSING
Console
Errors
Updates
p5.js Web Editor
editor.p5js.org
File
Edit
Sketch
Help
English
Hello, jfeddersen!
Auto-refresh
Nettle cardboard
sketch.js
Preview
function setup() {
createCanvas(400, 400);
}
function draw() {
background(220);
}
Console
Clear
P5JS
"TICK"


p5.js/main.js at 30d338b55bb
github.com/processing/p5.js/blo
};
this._draw =
const now =
const time_
const target
// only draw
// draw if
// (this wi
// again whe
// issue whe
// in sync
// if loopin
// is the ca
const epsil
if (
!this._loo
time_since
) {
//mandator
this.redraw();
p5.js Web Editor | TimeoutInter
editor.p5js.org/jfeddersen/sketches/vqW-vpWJR
File Edit Sketch Help
Paused in debugger
English
Hello, jfeddersen!
Auto-refresh
TimeoutIntervalBasic
by jfeddersen
sketch.js
Saved: about 1 year ago
Preview
Console
Clear
Hi from Interval: 1148.699999988079
Hi from Interval: 623279.5
***Yo from Timeout: 623279.6000000238
Stopping intervalID 2
Elements
Console
Sources
Network
Performance
12
7
Page
p5.sound.min.js:formatted
top
editor.p5js.org
jfeddersen/s
vqW-vpW
app.f3cea0a
app.f3cea0a
ff35e397426
fonts.googleap
fonts.gstatic.c
webpack://
www.google-a
preview.p5js.o
186f6e77-68da-4
6a19a454-faba-4
t.Context.prototype.now = function() {
return this._context.currentTime
}
t.Context.prototype._createWorker = funct
window.URL = window.URL || window.web
var t = new Blob(["var timeoutTime =
, e = URL.createObjectURL(t)
, i = new Worker(e);
return i.addEventListener("message",
this.emit("tick")
}
.bind(this)),
i.addEventListener("message", functio
var t = this.now();
if (this.isNumber(this._lastUpdat
var e = t - this._lastUpdate;
this._computedUpdateInterval
}
2 lines, 34 characters selected
Coverage: n/a
Debugger paused
Threads
Main
paused
186f6e77-68da-4004-a763-fe2...
6a19a454-faba-4e3e-a750-08...
Watch
Breakpoints
vqW-vpWJR:1
Scope
Local
this: t.Context
Closure
Closure
Script
English
Hello, jfeddersen!
ardboard
Preview
Clear
JS
"tick"

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

ROBERT PENNER

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 *quadratic* (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

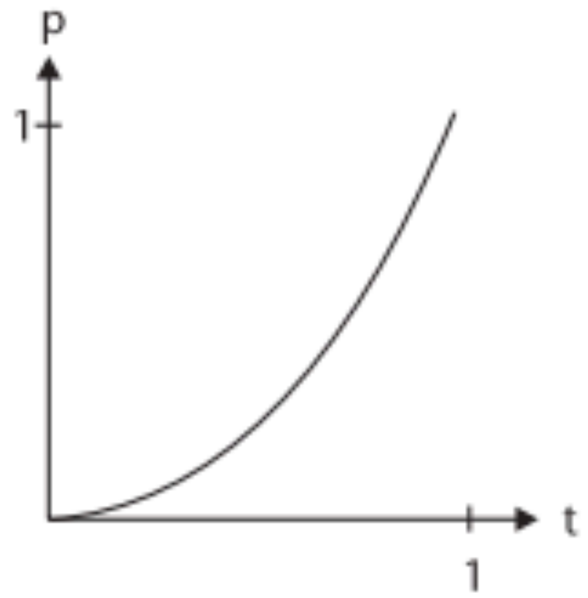
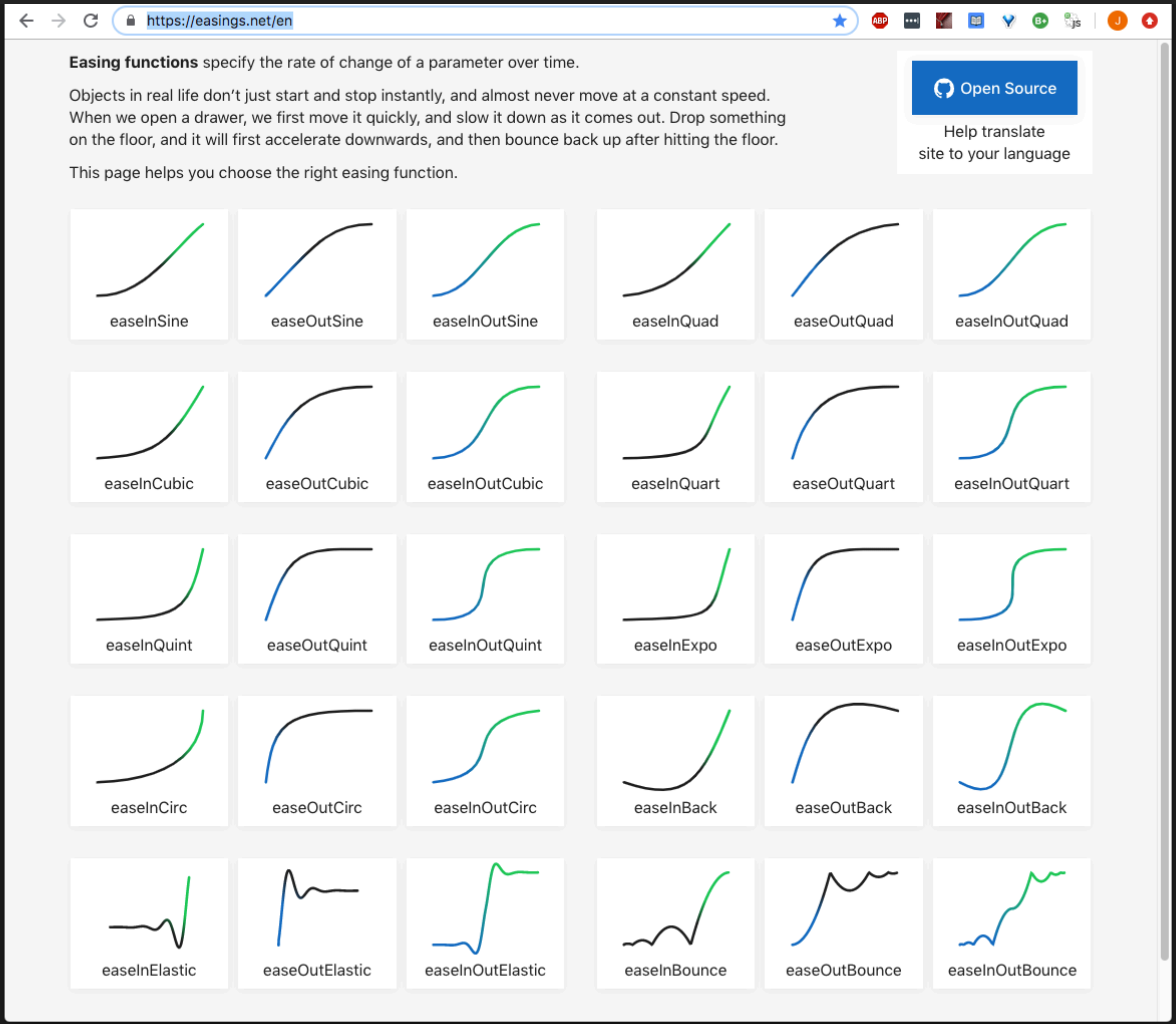


FIGURE 7-7

Graph of quadratic easing

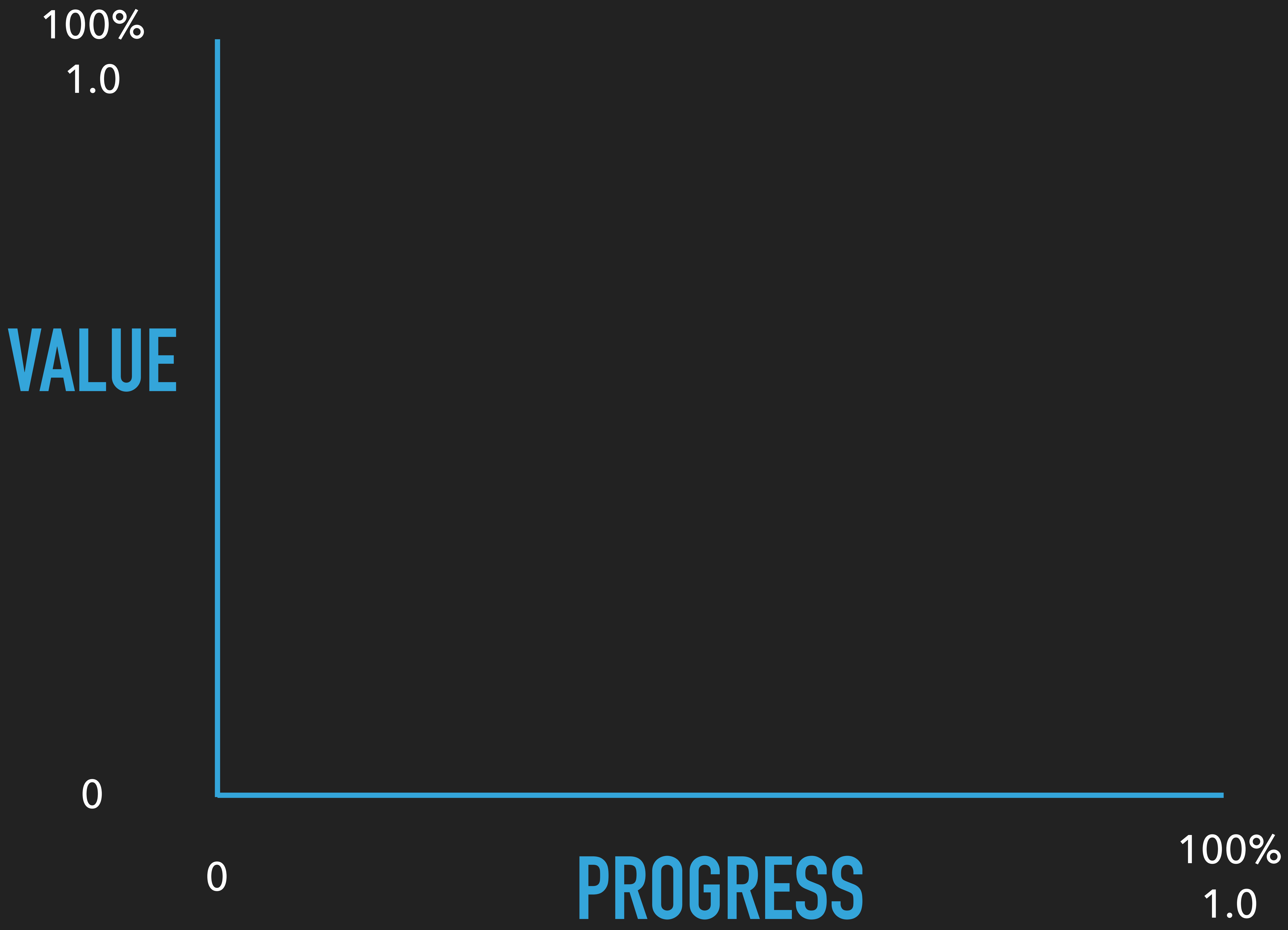
Robert Penner's Programming Macromedia Flash (2002)

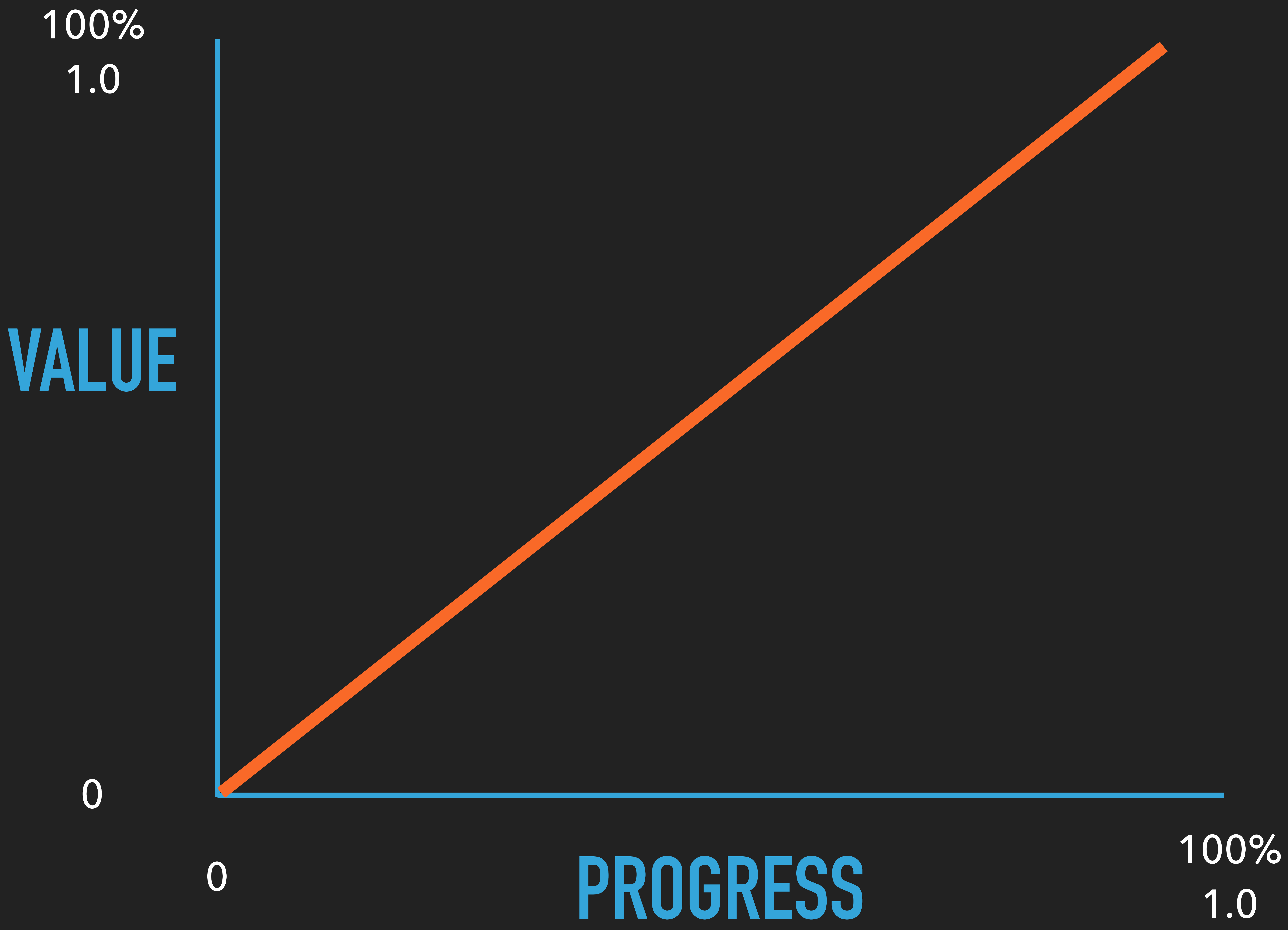
ROBERT PENNER

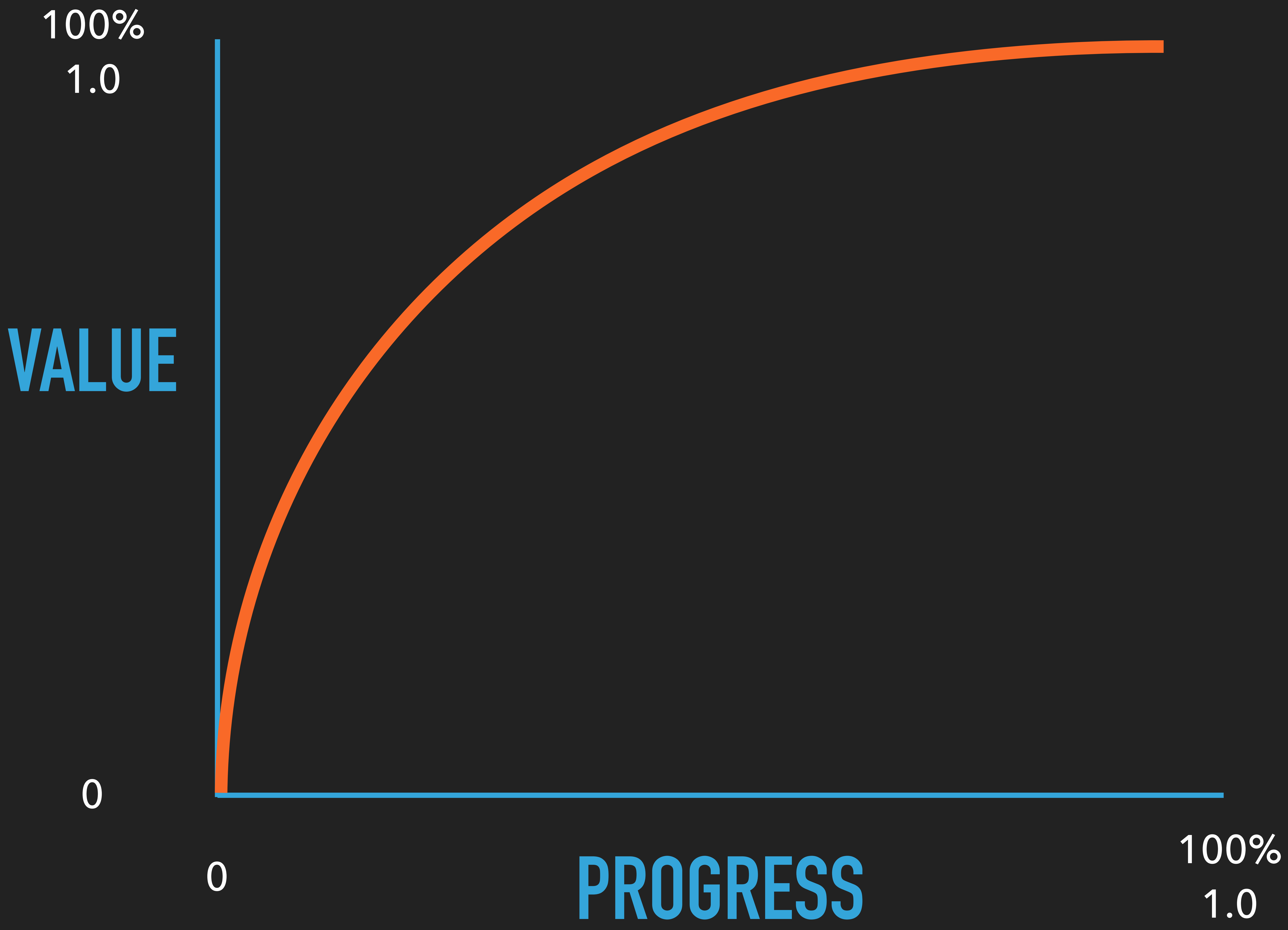


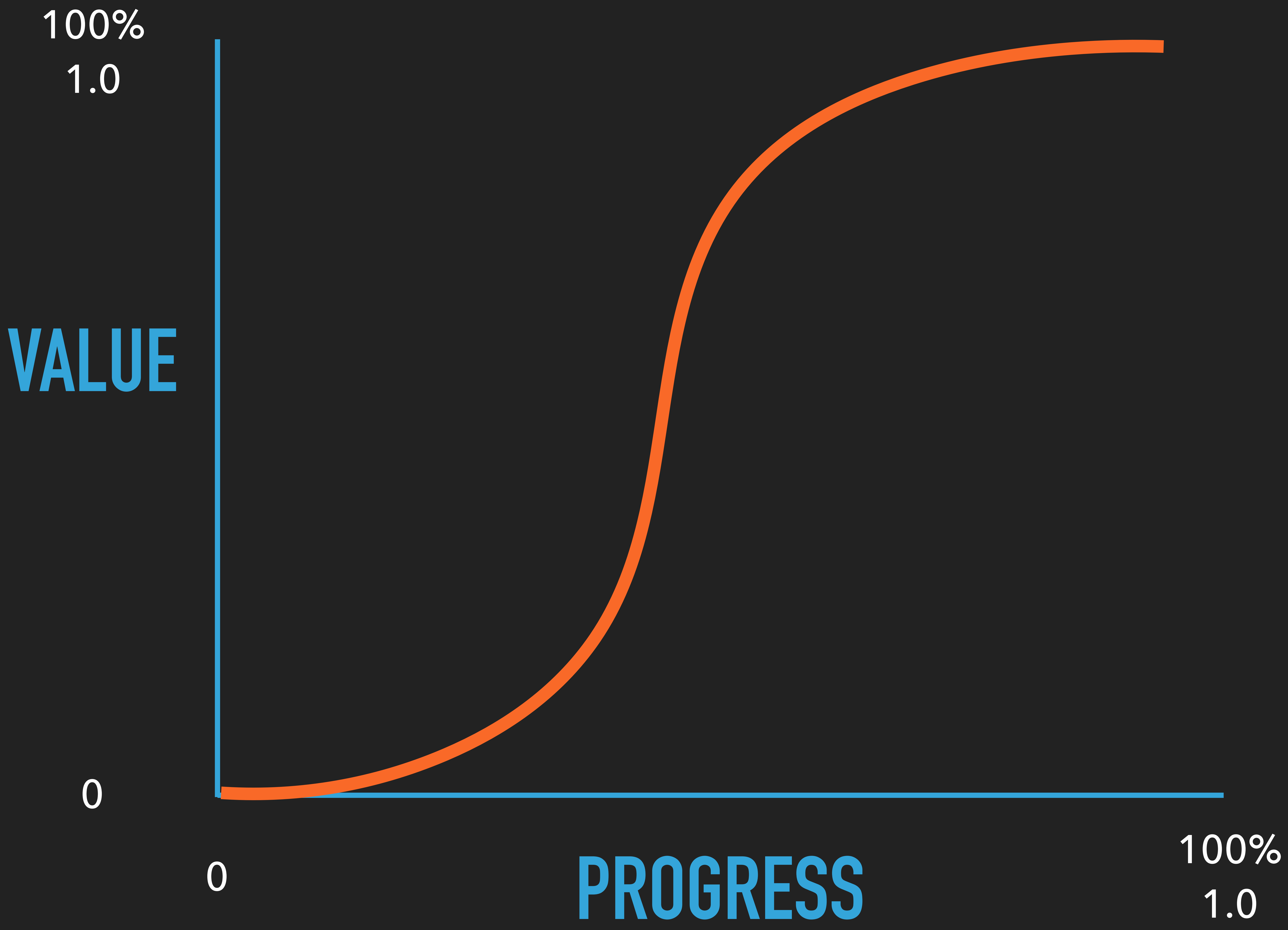
https://easings.net/en

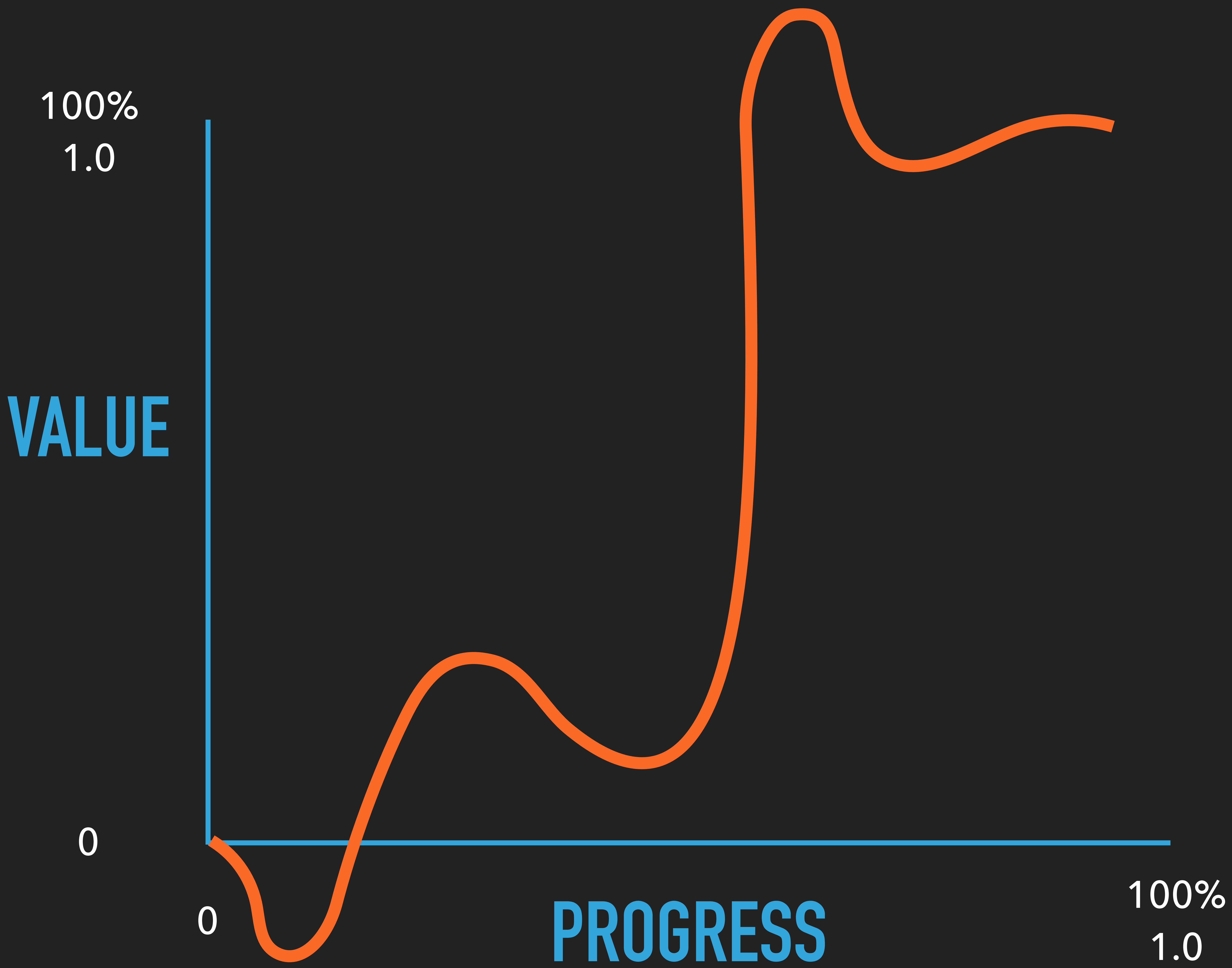
EASING Smoothly transition a variable from one value to another in a set time



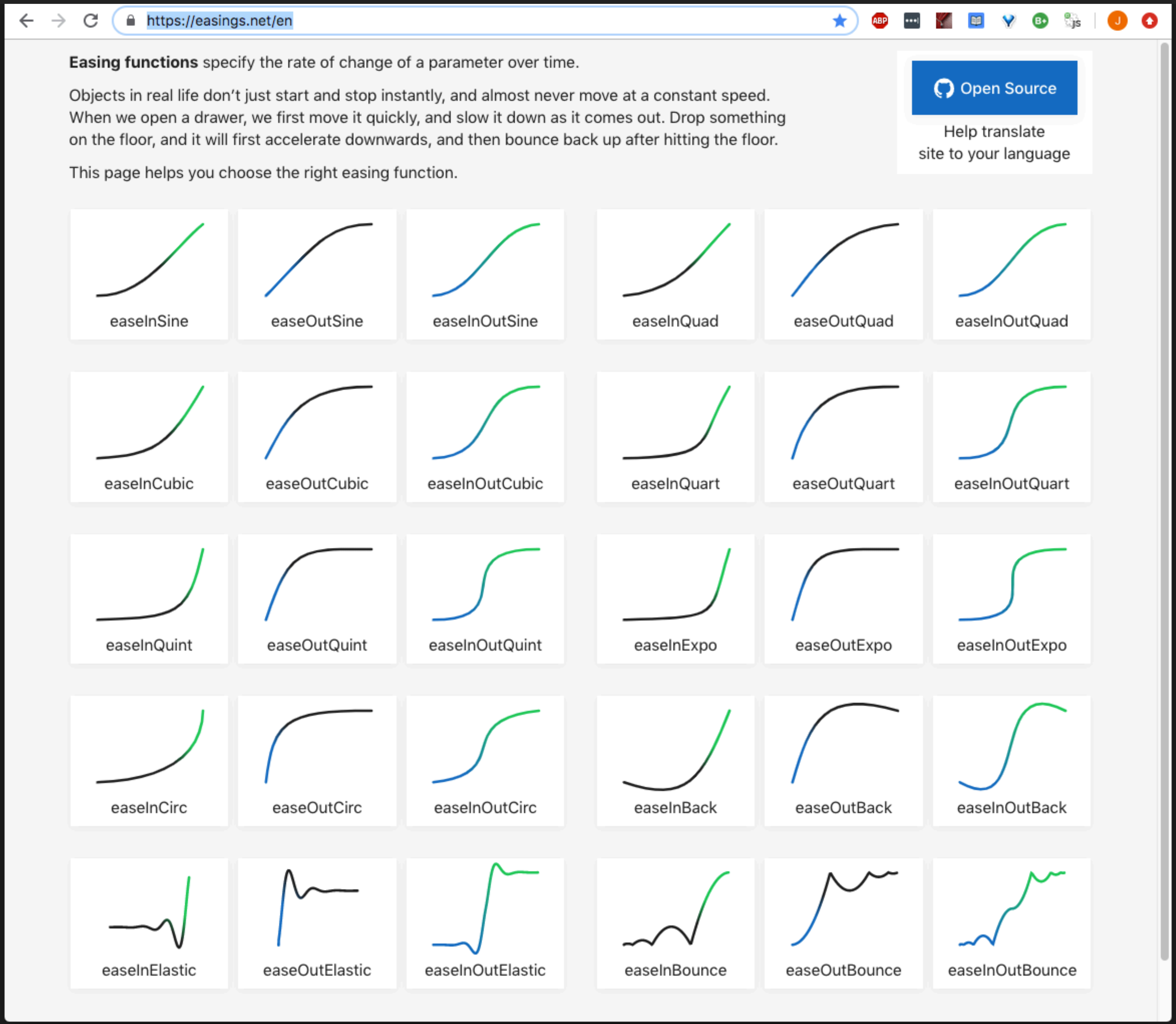

100%
1.0
VALUE
0
0
PROGRESS
100%
1.0


100%
1.0
VALUE
0
0
PROGRESS
100%
1.0


100%
1.0
VALUE
0
0
PROGRESS
100%
1.0


100%
1.0
VALUE
0
0
PROGRESS
100%
1.0

ROBERT PENNER



https://easings.net/en

ROBERT PENNER

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 *quadratic* (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

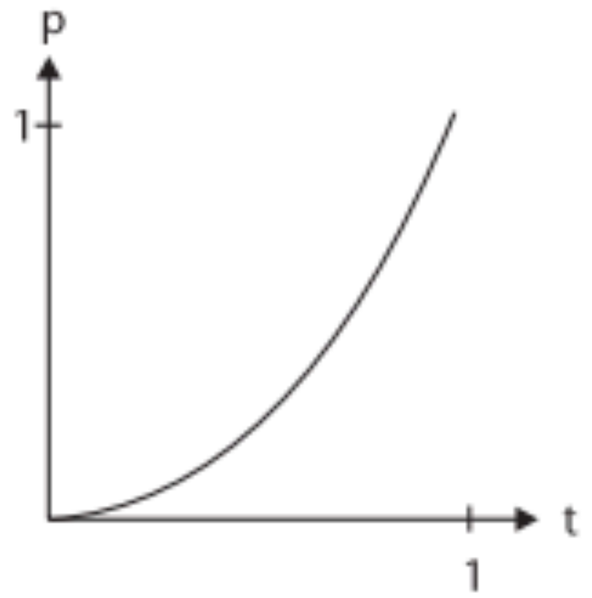
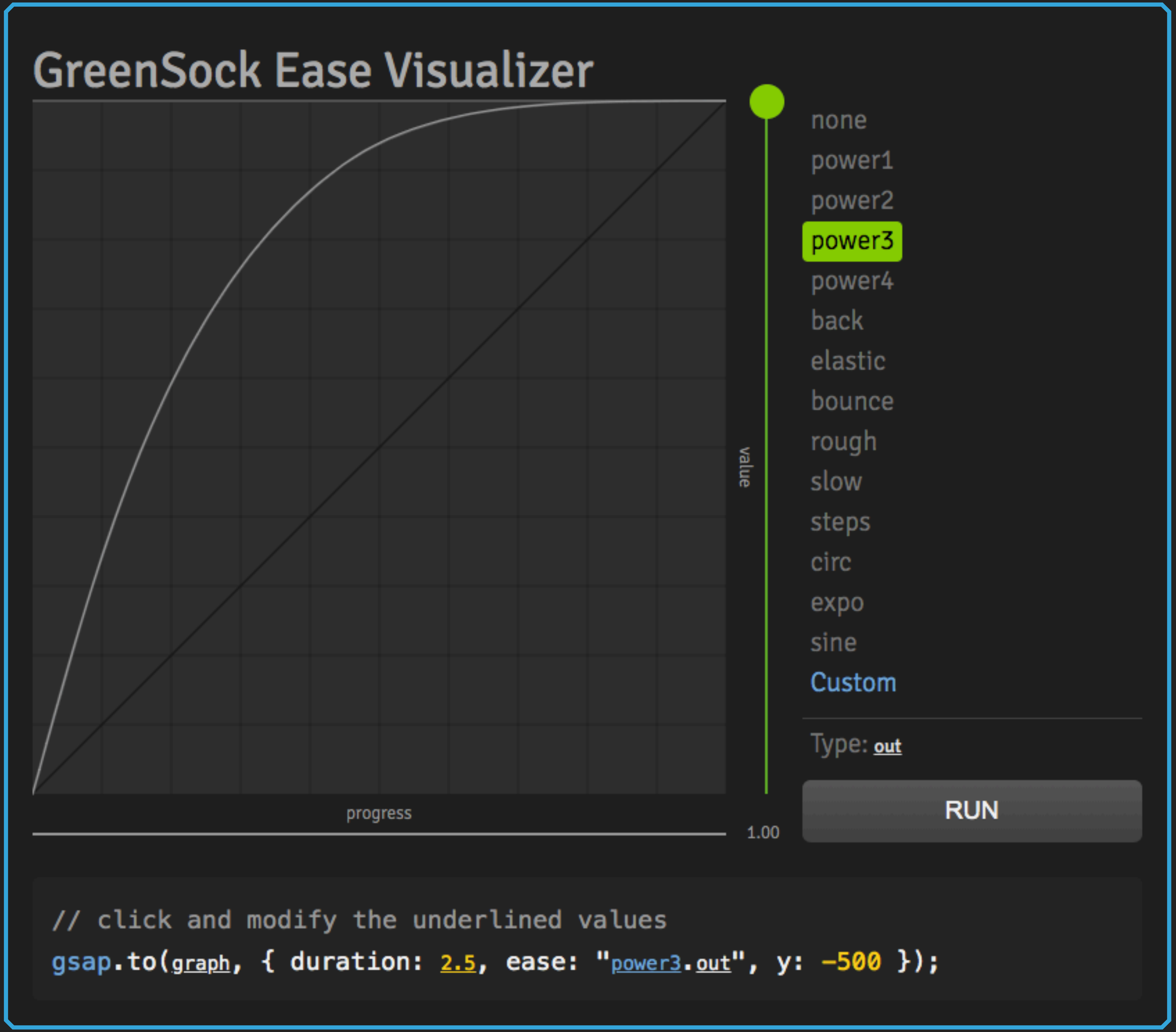


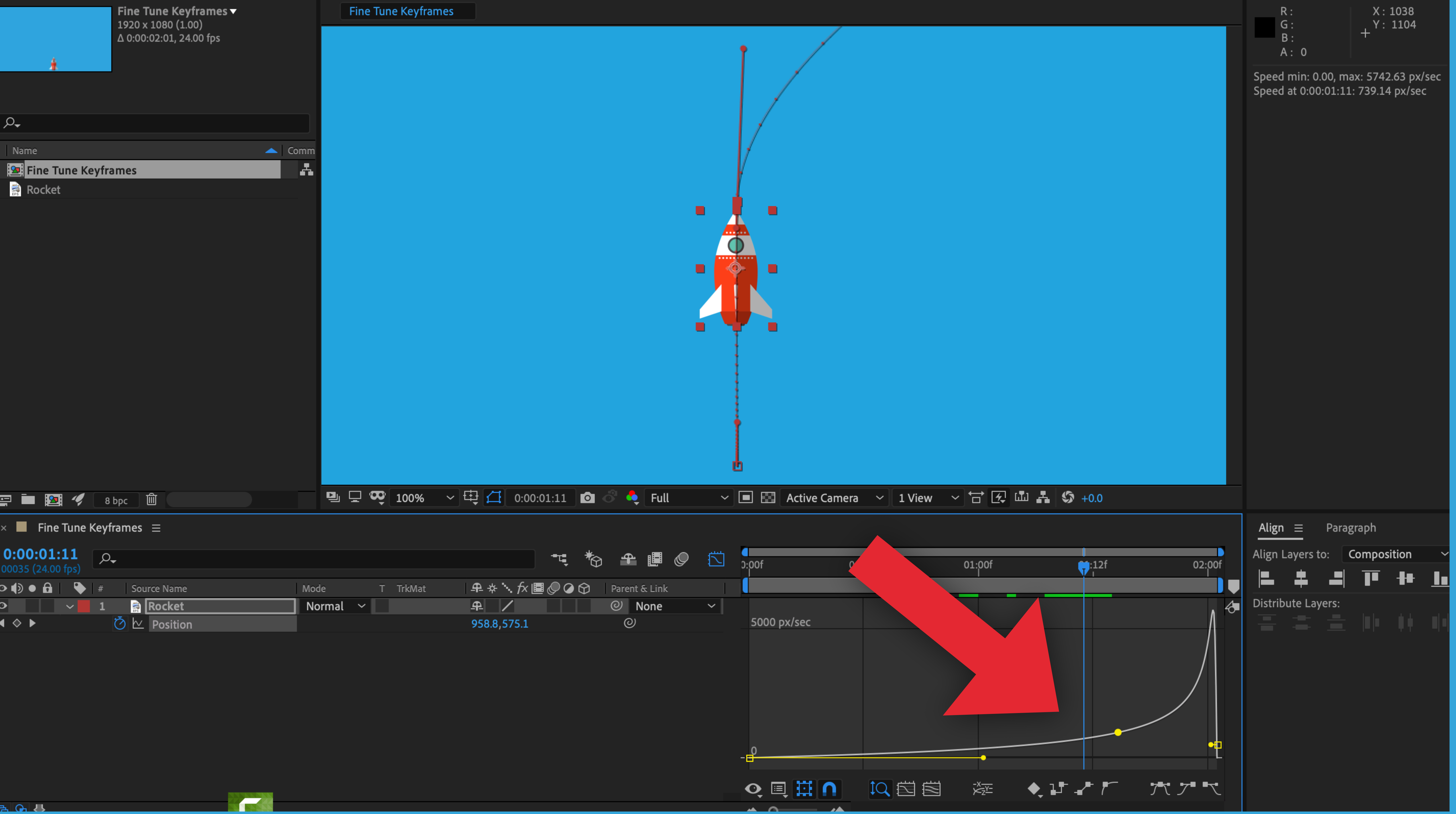
FIGURE 7-7

Graph of quadratic easing

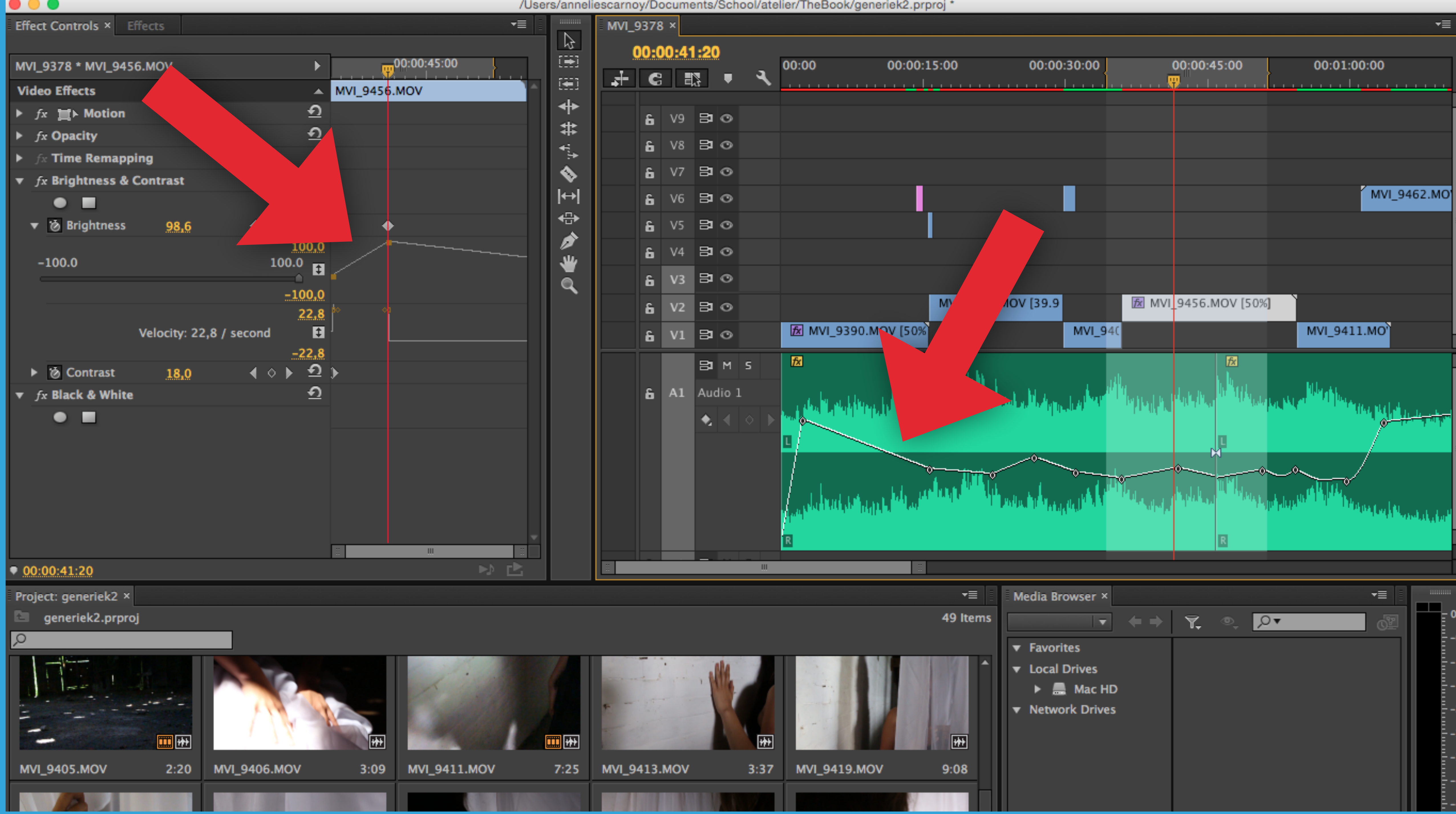
Robert Penner's Programming Macromedia Flash (2002)



https://greensock.com/ease-visualizer

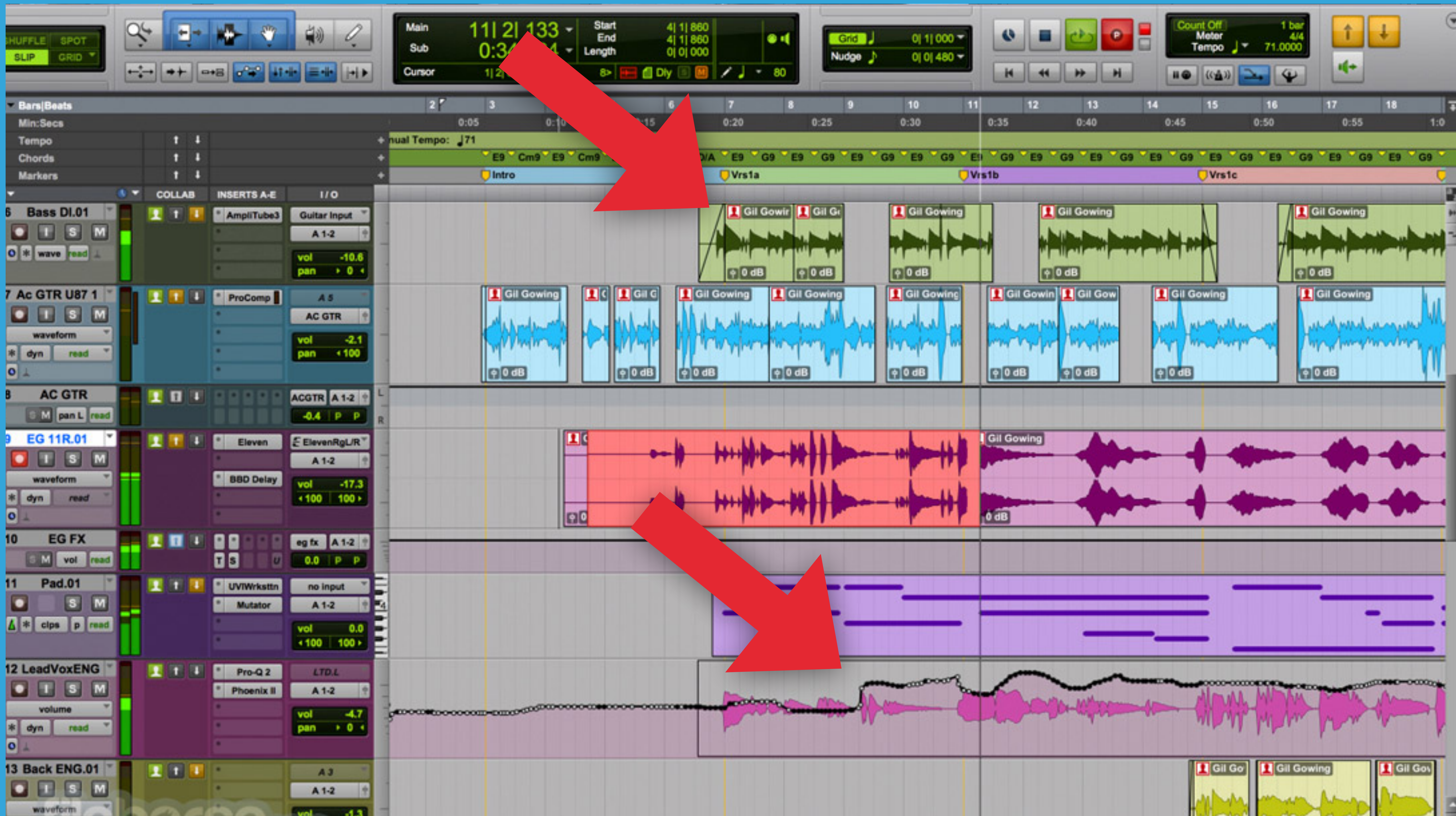


Fine Tune Keyframes
1920 x 1080 (1.00)
Rocket
Speed min: 0.00, max: 5742.63 px/sec
Speed at 0:00:01:11: 739.14 px/sec
0:00:01:11
Source Name
Mode
Position
958.8,575.1
5000 px/sec
Align
Paragraph
Align Layers to:
Composition
Distribute Layers:
Active Camera
1 View



Effect Controls
Effects
MVI_9378 * MVI_9456.MOV
Video Effects
Motion
Opacity
Time Remapping
Brightness & Contrast
Brightness
98,6
Velocity: 22,8 / second
Contrast
18,0
Black & White
00:00:41:20
MVI_9378
Audio 1
Project: generiek2
generiek2.prproj
49 Items
Media Browser
Favorites
Local Drives
Mac HD
Network Drives
MVI_9405.MOV
MVI_9406.MOV
MVI_9411.MOV
MVI_9413.MOV
MVI_9419.MOV

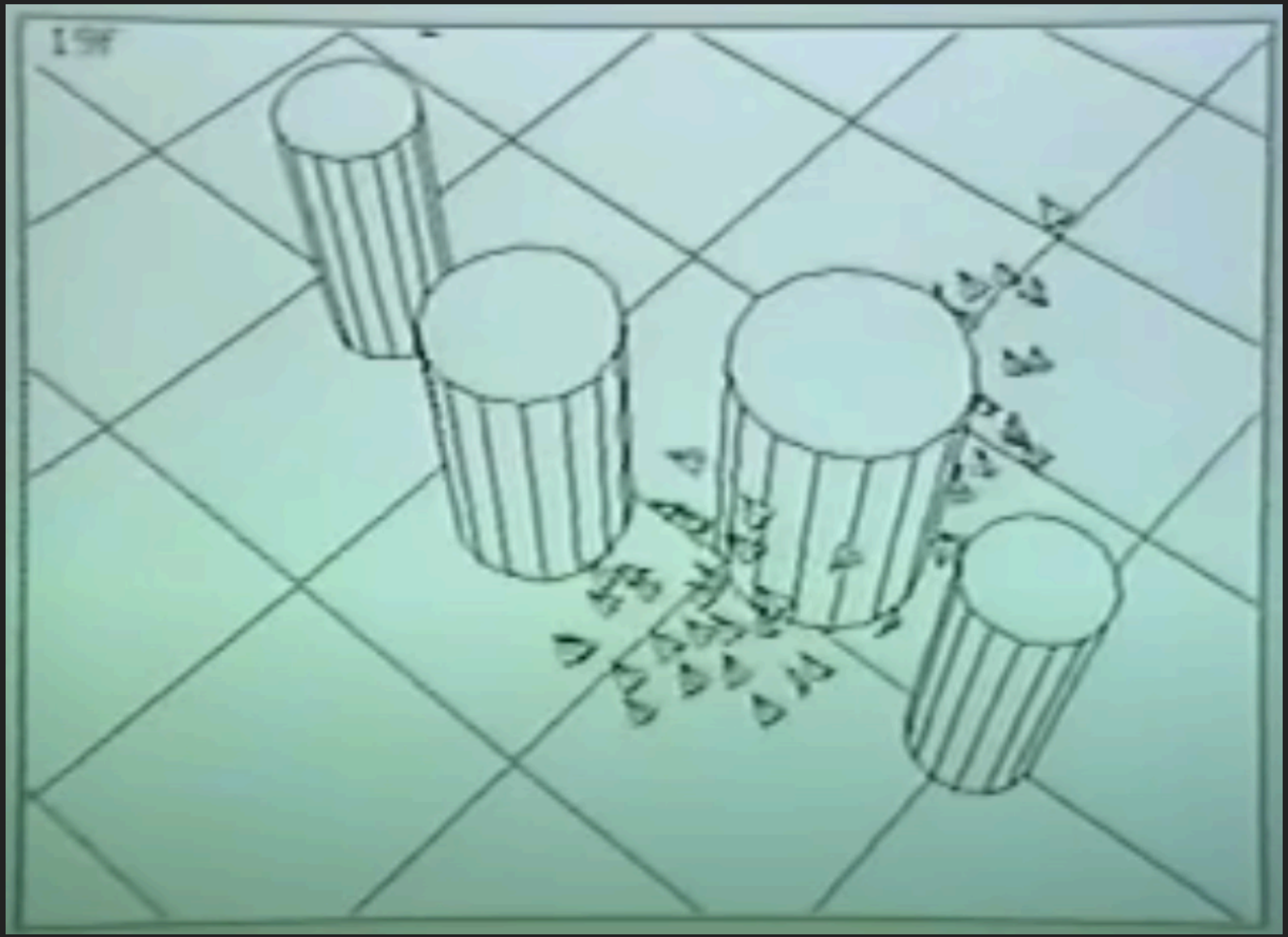
Synth Lead
Track
Read
Volume
Snap:
Smart
Drag:
No Overlap
Single
Tracks
All
Audio
Inst
Aux
Bus
Input
Output
Master
MIDI
Gain Reduction
EQ
Input
Audio FX
Sends
Output
Pan
VCA
dB



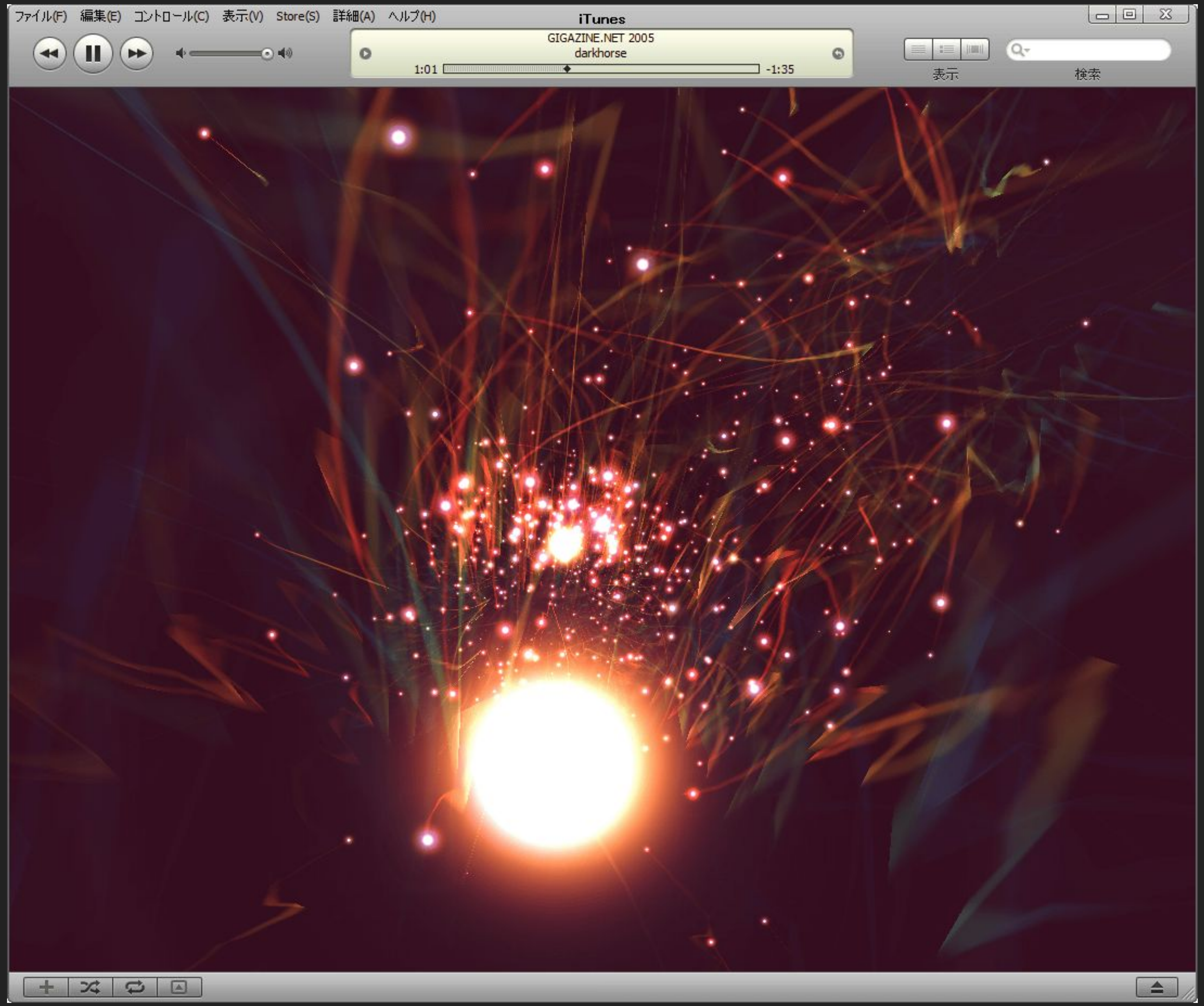
Bars|Beats
Min:Secs
Tempo
Chords
Markers
Intro
Vrs1a
Vrs1b
Vrs1c
Bass DI.01
Ac GTR U87 1
AC GTR
EG 11R.01
EG FX
Pad.01
12 LeadVoxENG
13 Back ENG.01
Gil Gowing

SIMULATION

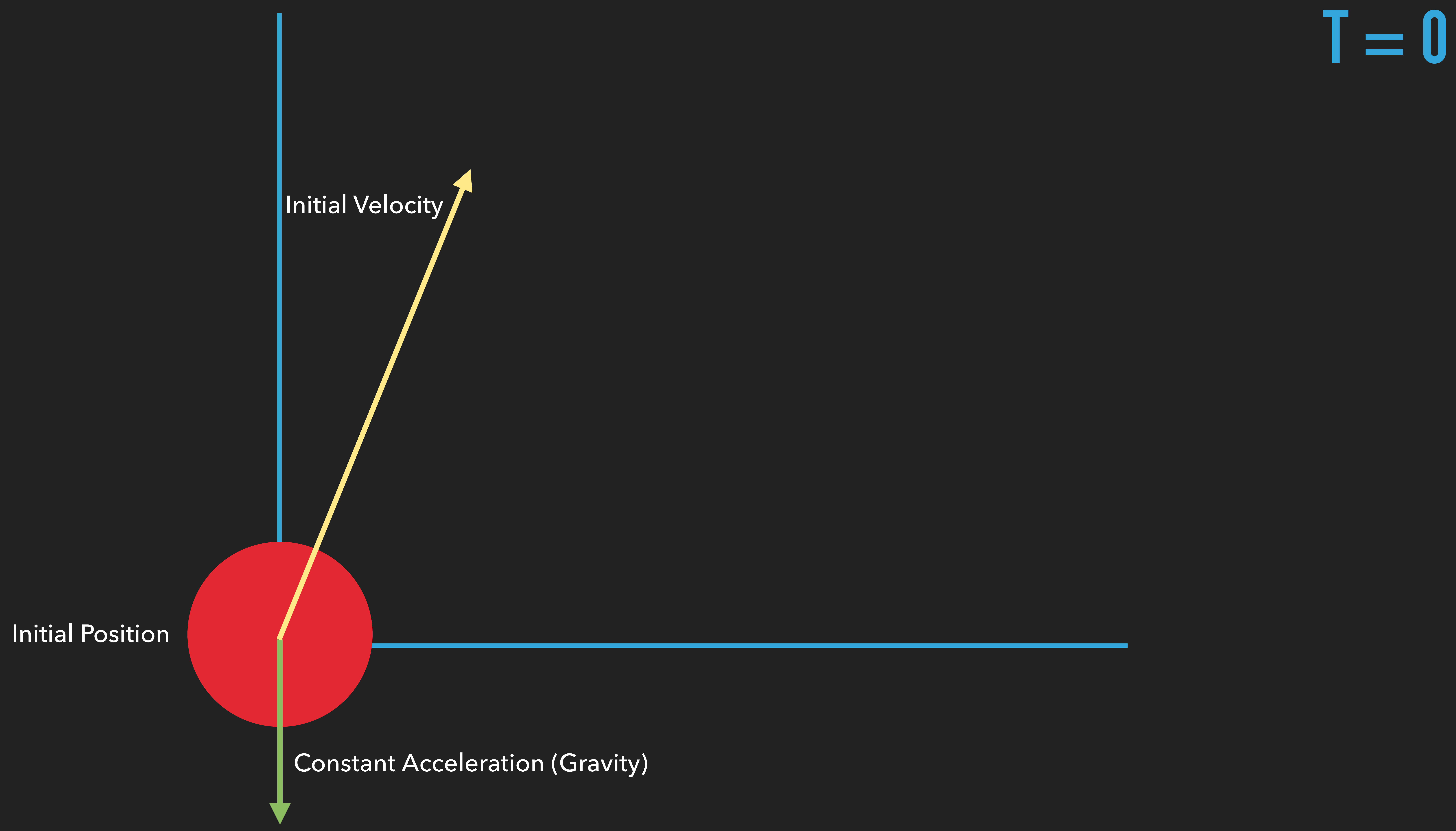
SIMULATION Use physics or other rules to determine next frame for one or more objects.



Craig Reynolds' Boids (1986)



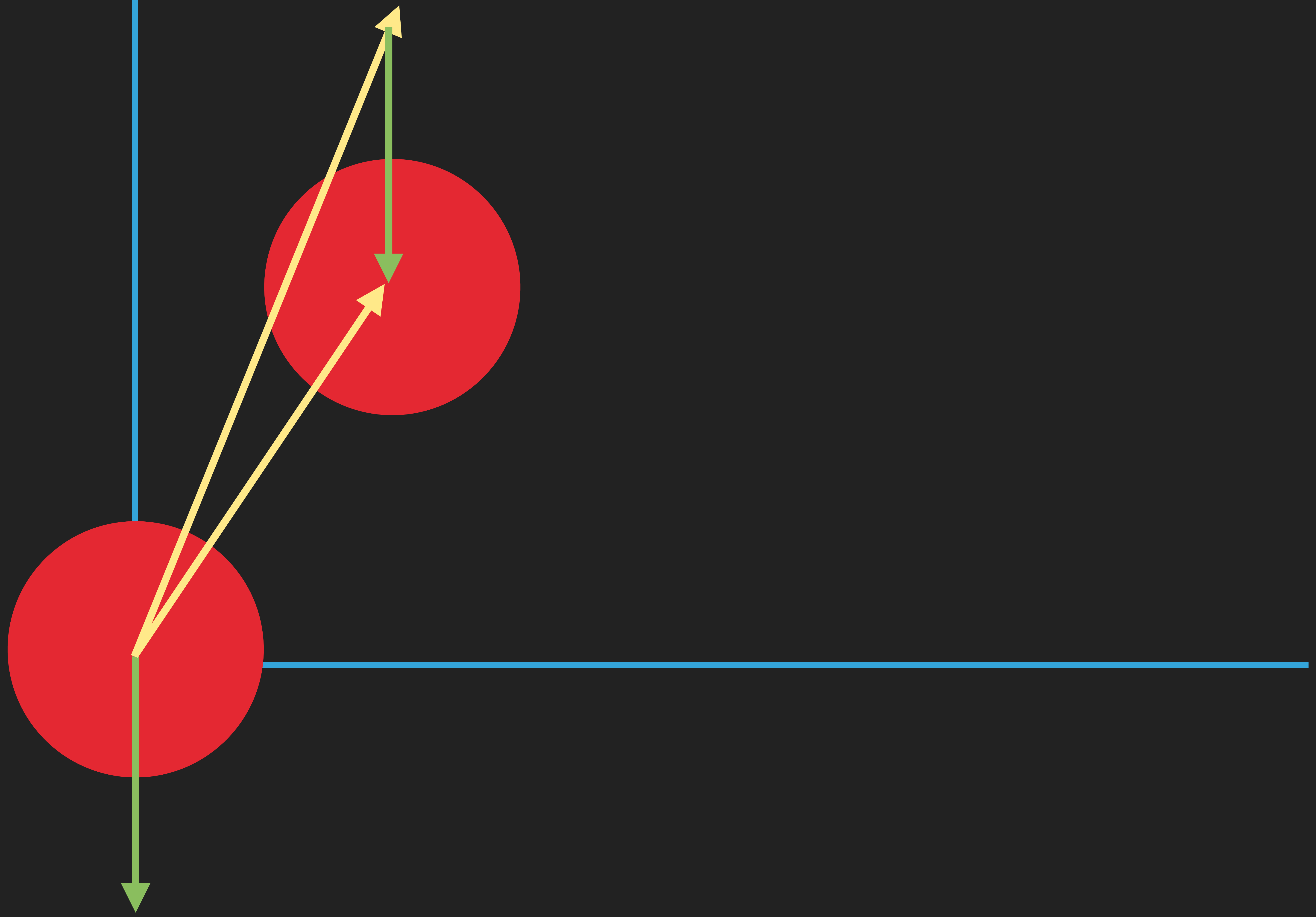
Robert Hodgin's (Flight 404) Magnetosphere, 2007

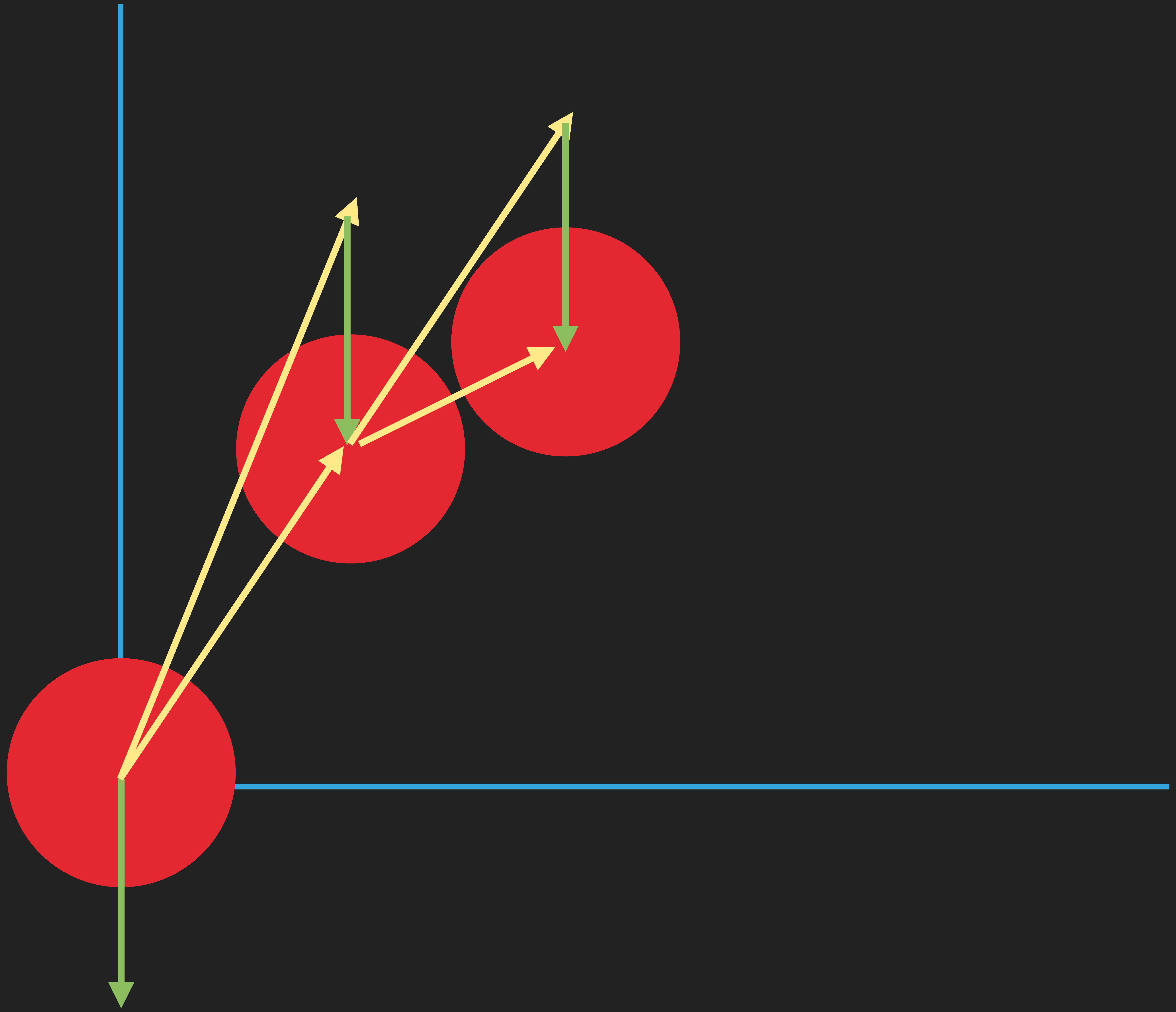


T = 0
Initial Velocity
Initial Position
Constant Acceleration (Gravity)

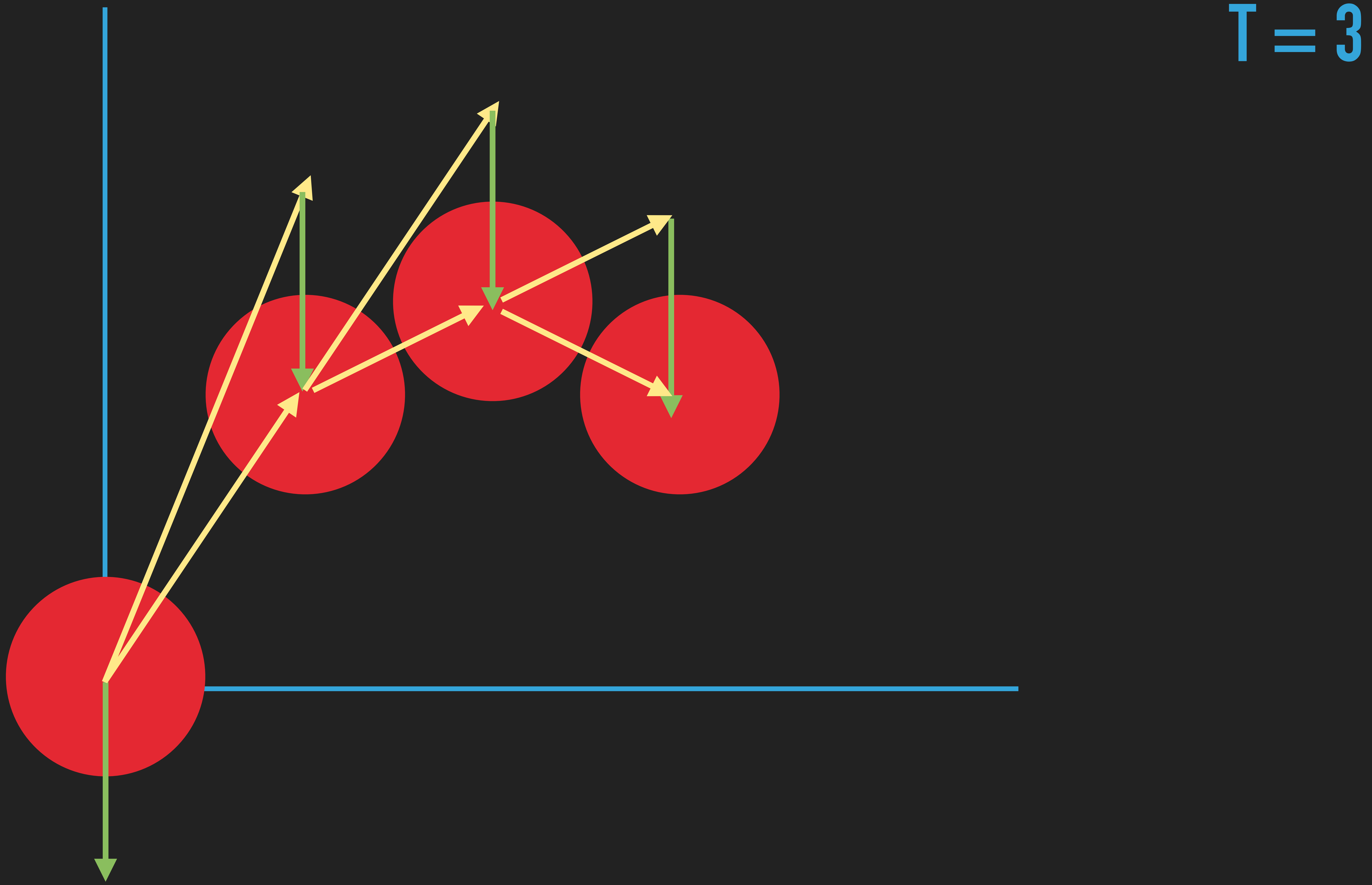
T = 1

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

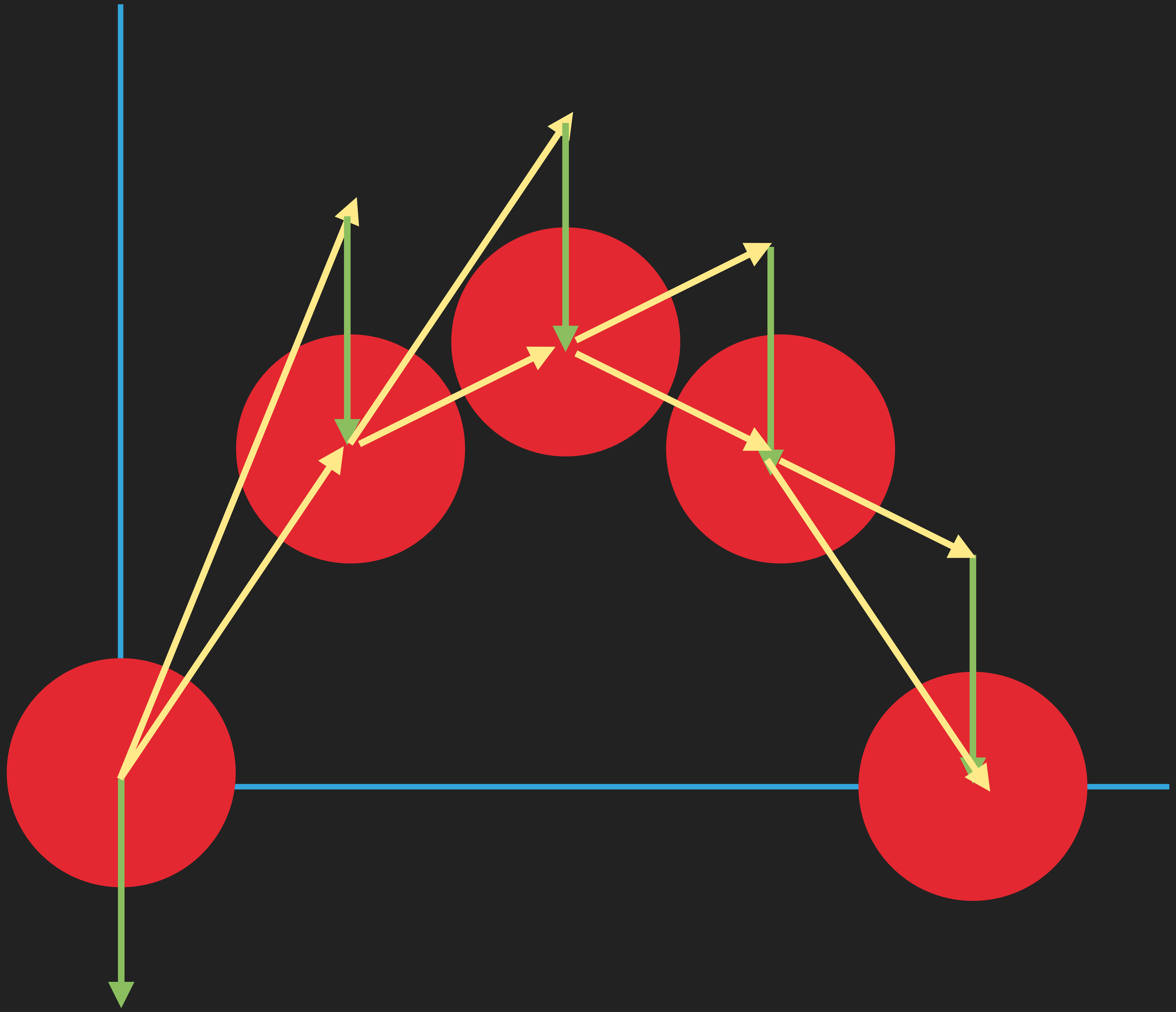




T = 2


T = 3

T = 4...



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

```cpp
//  Created by Robert Hodgin on 5/14/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#include "cinder/app/AppBasic.h"
#include "cinder/Rand.h"
#include "cinder/Sphere.h"
#include "Particle.h"

using namespace ci;

Particle::Particle(){}

Particle::Particle( const Vec3f &pos, float charge )
	: mPos( pos ), mCharge( charge )
{
	mVel			= Vec3f::zero();
	mAcc			= Vec3f::zero();
	mForce			= 0.0f;

	mRadius			= 1.0f;
	mShellRadius	= 12.0f;
}

void Particle::update( const Camera &cam, float dt )
{
	Sphere s			= Sphere( mPos, mRadius * 10.0f );
	mScreenPos			= cam.worldToScreen( mPos, app::getWindowWidth(), app::getWindowHeight() );
	mScreenRadius	= cam.getScreenRadius( s, app::getWindowWidth(), app::get

	mColor			= mCharge * 0.5f + 0.5f;

	mVel += mAcc * dt;
	mPos += mVel * dt;
	mAcc = Vec3f::zero();

	mShellRadius = mRadius + fabs( mForce ) * 50000.0f;

	mMatrix.setToIdentity();
	mMatrix.translate( mPos );
}

void Particle::draw()
{
	gl::color( Color( mColor, mColor, mColor ) );
	gl::drawSphere( mPos, mRadius );
```


```cpp
	mVel += mAcc * dt;
	mPos += mVel * dt;
	mAcc = Vec3f::zero();
```


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

https://github.com/flight404/Eyeo2012

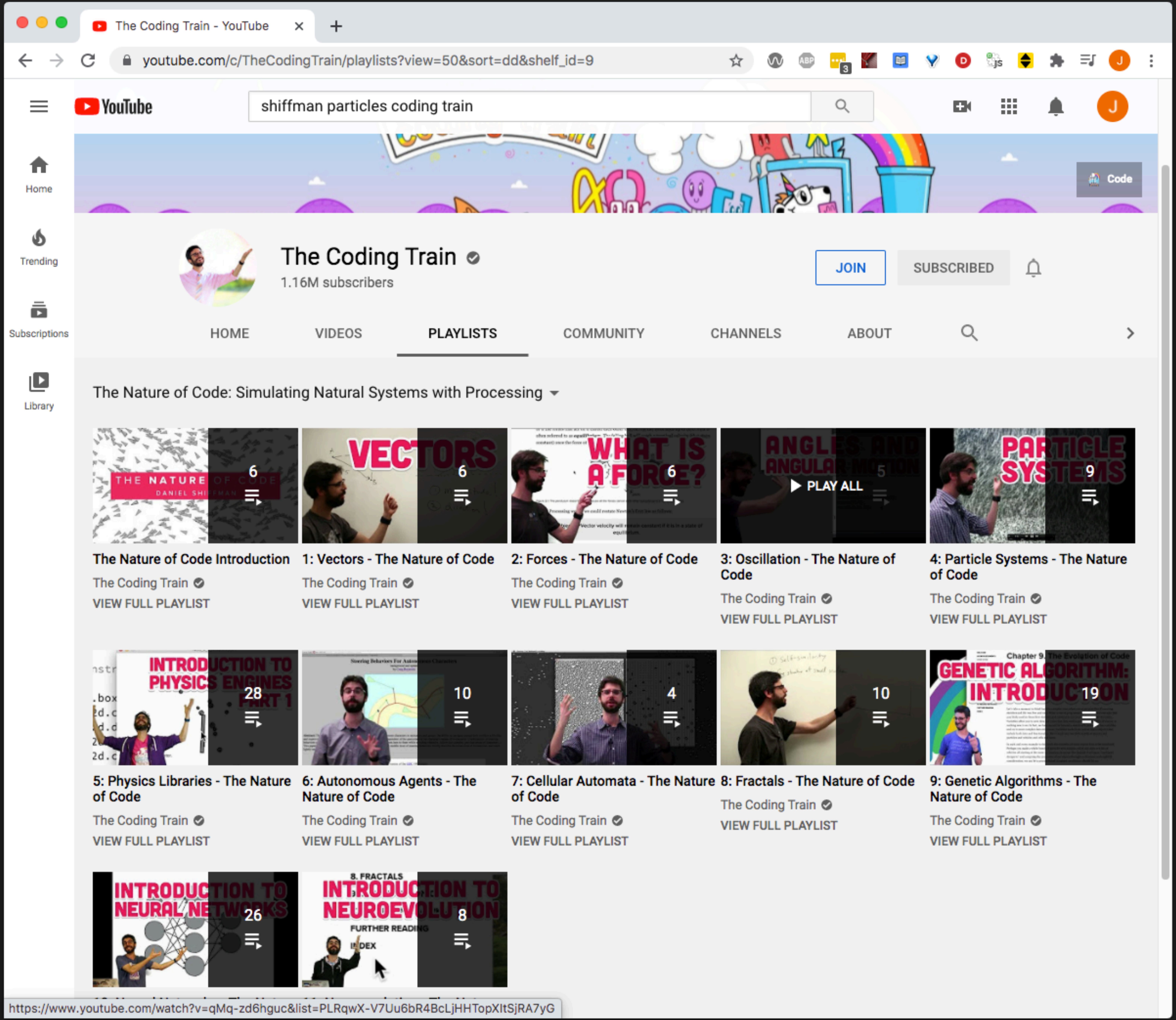
"The physics of the simple vehicle model is based on forward Euler integration. At each simulation step, behaviorally determined steering forces (as limited by *max_force*) are applied to the vehicle's point mass. This produces an acceleration equal to the steering force divided by the vehicle's mass. 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
```

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

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

Steering Behaviors For Autonomous Characters
Craig W. Reynolds
https://www.red3d.com/cwr/steer/gdc99/



Pretty good source in-house