

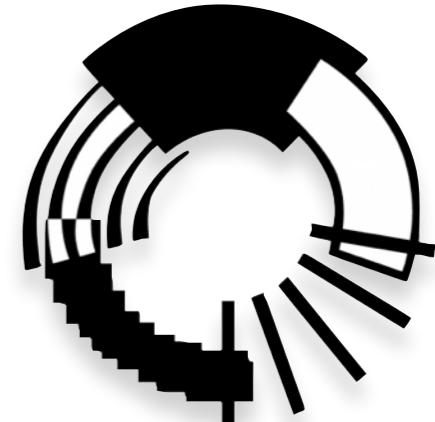
Hallo

Adriaan Wormgoor

<http://adriaanwormgoor.nl/>

info@adriaanwormgoor.nl

(@bombarie)



FourceLabs
Pioneers of Play

Adriaan Wormgoor

creative technology

Paar projecten waar ik
aan heb gewerkt

Wip'n'Kip



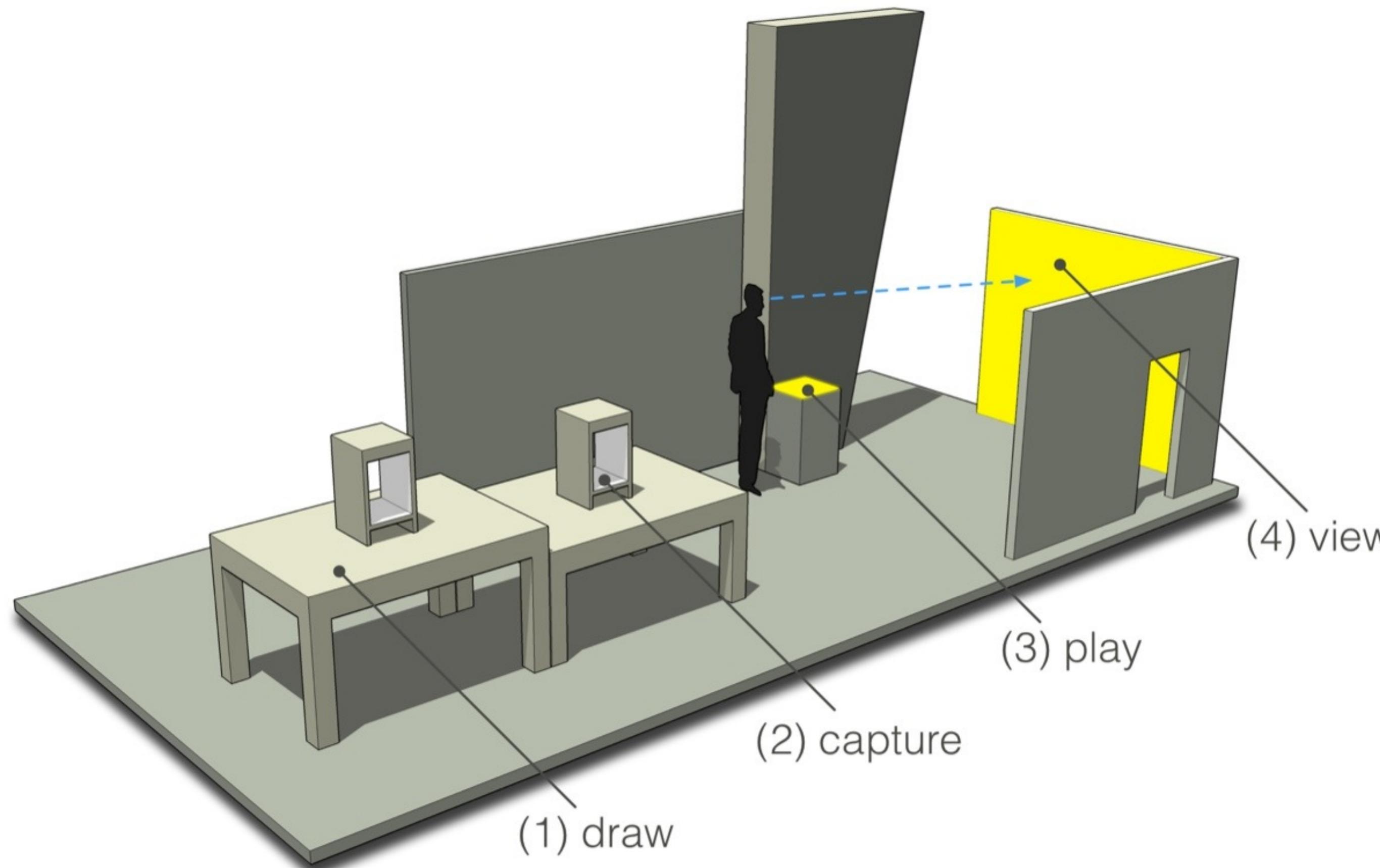




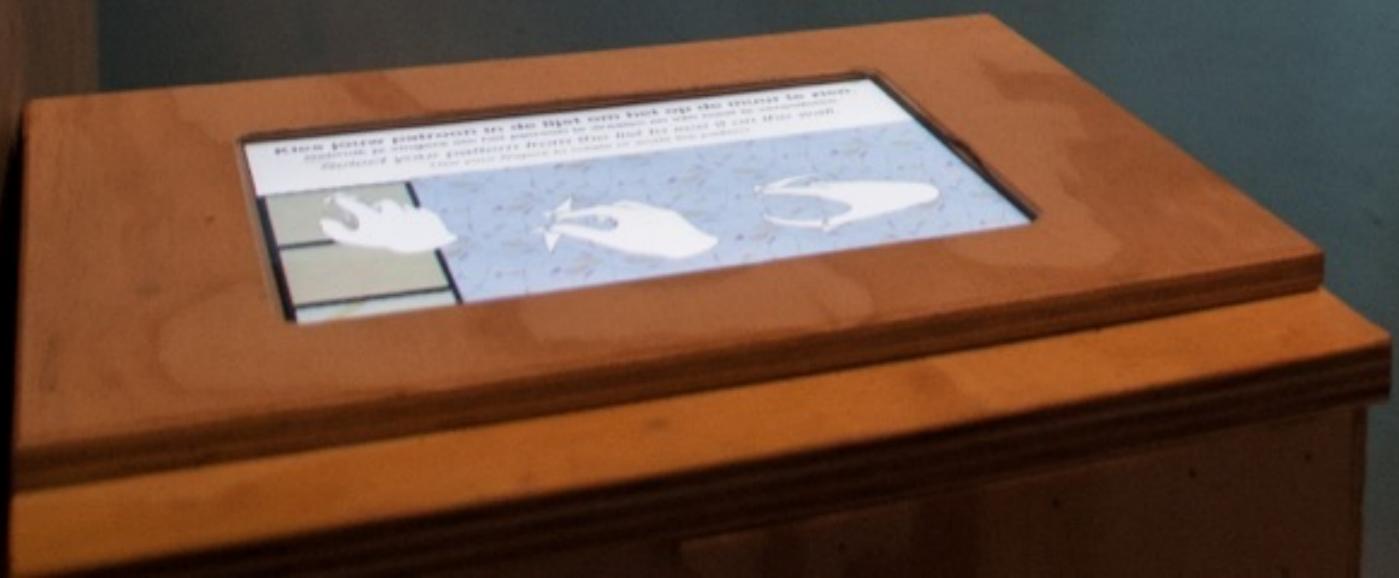


DecorLAB





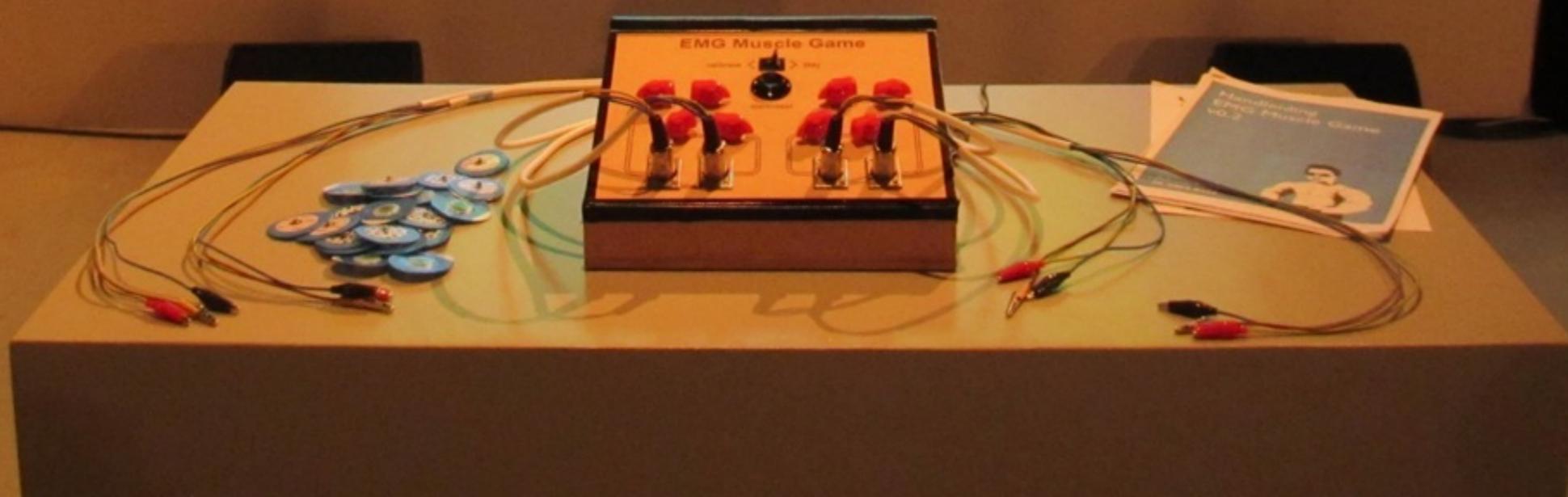
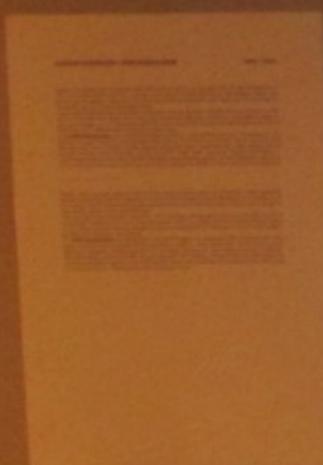




EMG Muscle Game

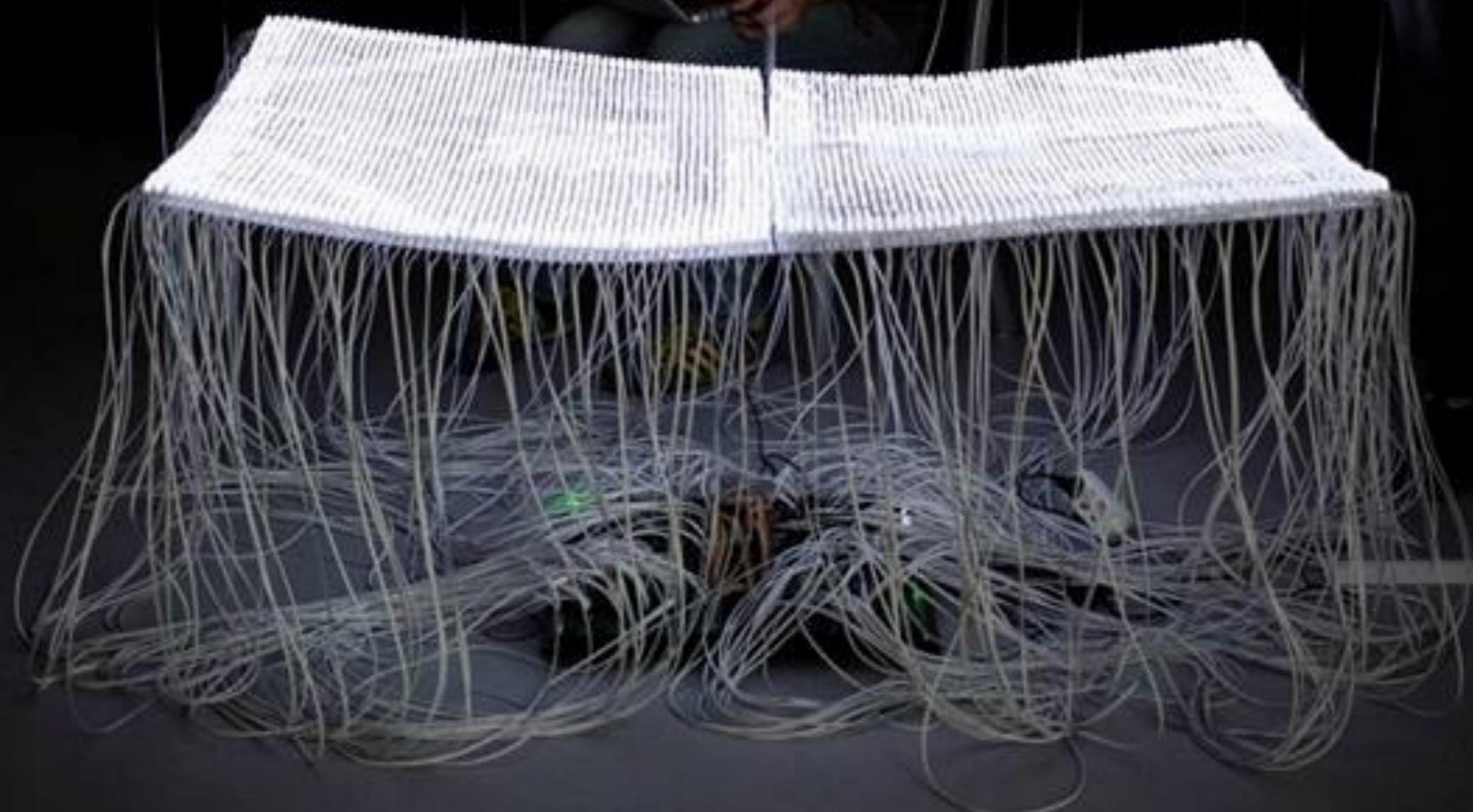


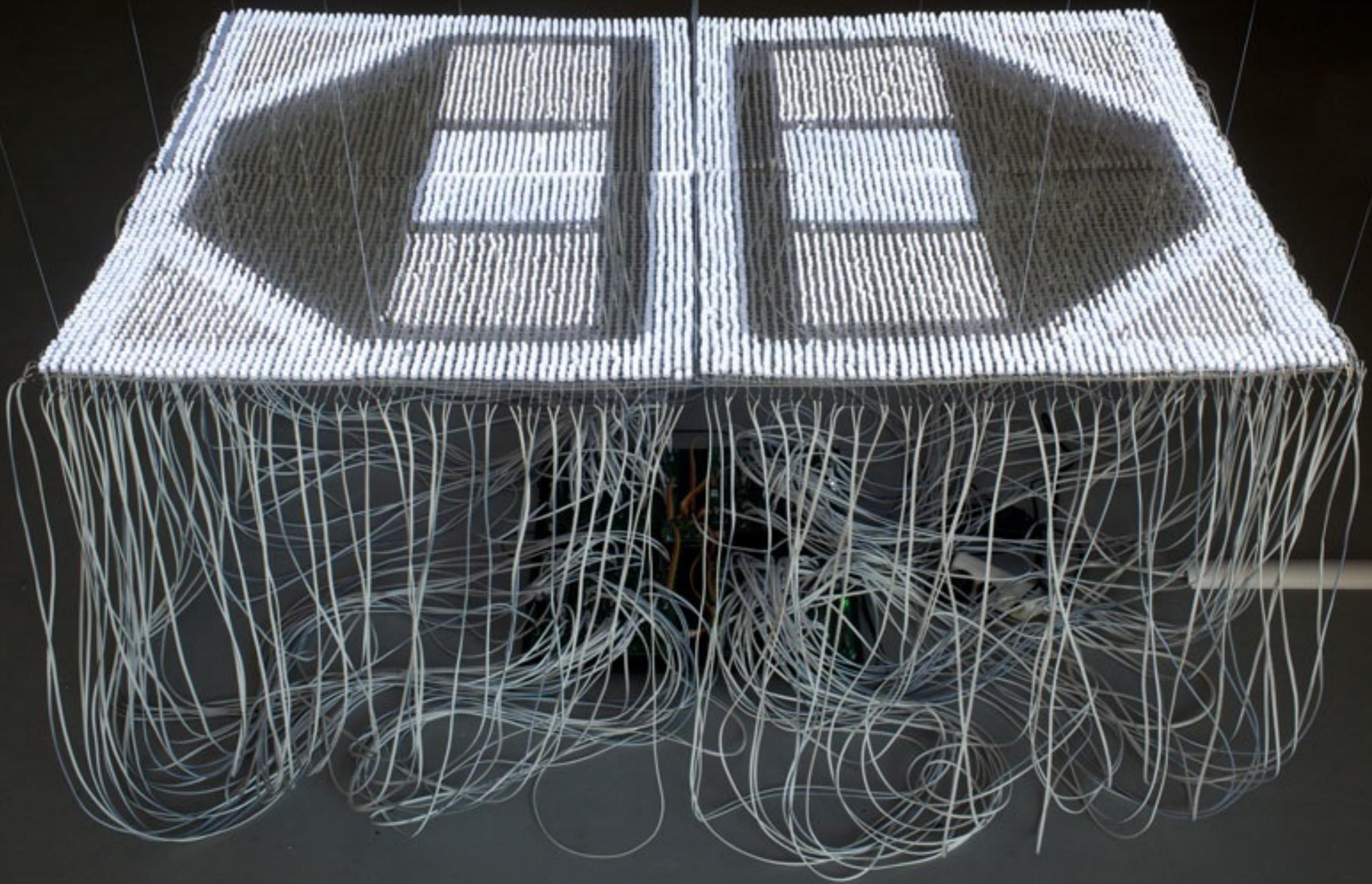
press start button
when ready

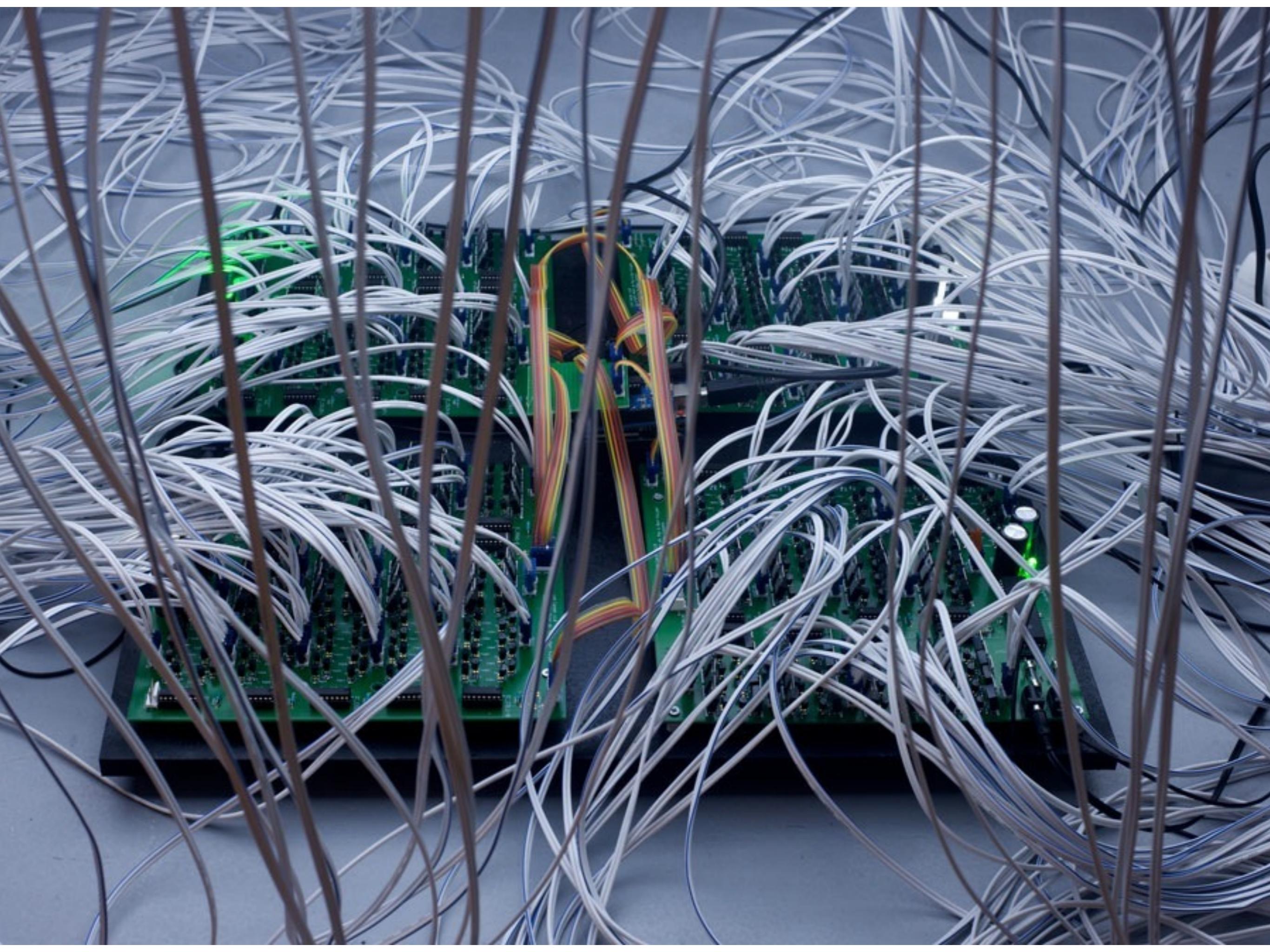




Led Carpet









Okee.. Processing
(fingers crossed!)

Processing

- Open source, multi-platform programmeeromgeving voor 2d / 3d graphics
- Gebaseerd op JAVA
- Ca. 15 jaar geleden ontstaan

Wie gebruikt het?

- Kunstenaars
- Installatie bouwers
- Data visualizers
- Game developers (mindere mate)
- Educators (prima instapniveau voor jonge programmeurs)

Waarom leuk

- Lekkere workflow: compiled snel, weinig code overhead (focus op resultaat!), multi-platform.
- Maakt het makkelijk om visueel te prototypen
- Toegankelijke interface naar onderliggend systeem / peripherals
- Levendige community

Generative visuals

het echte spelen!

Generative visuals

voorbeeld 1







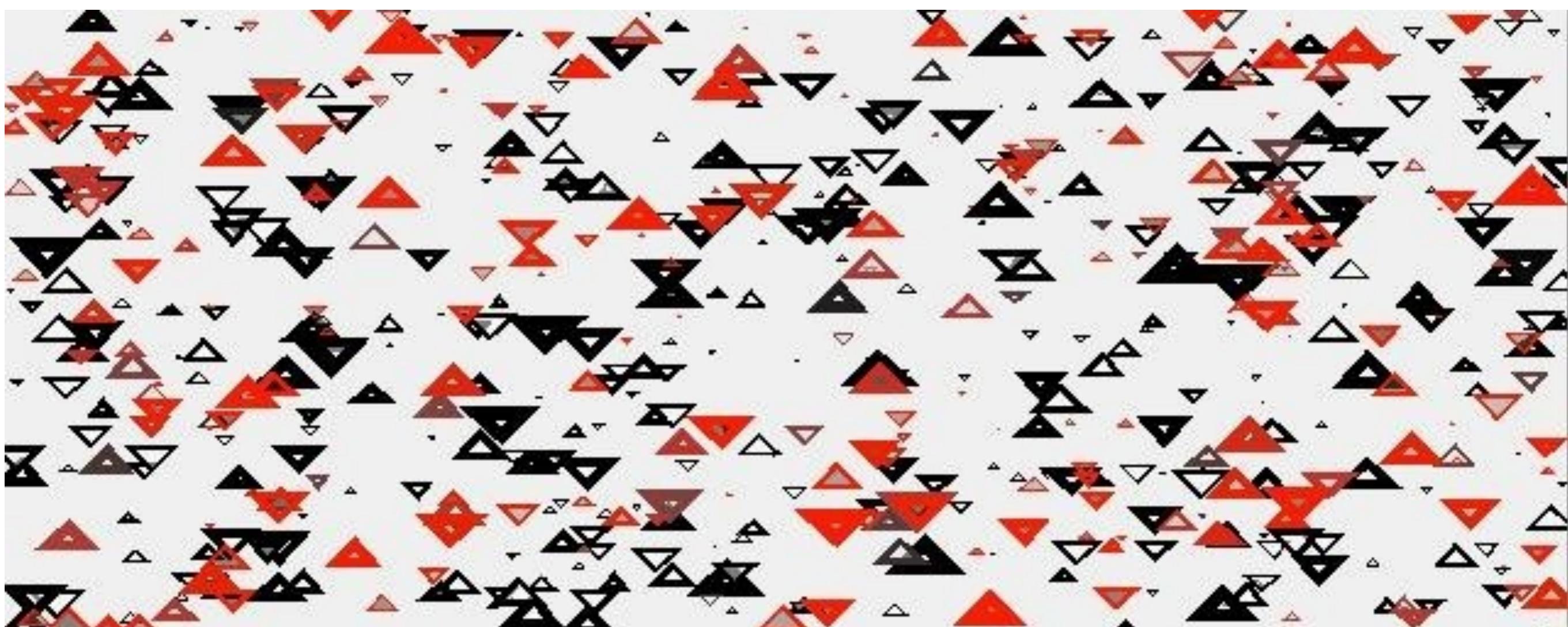


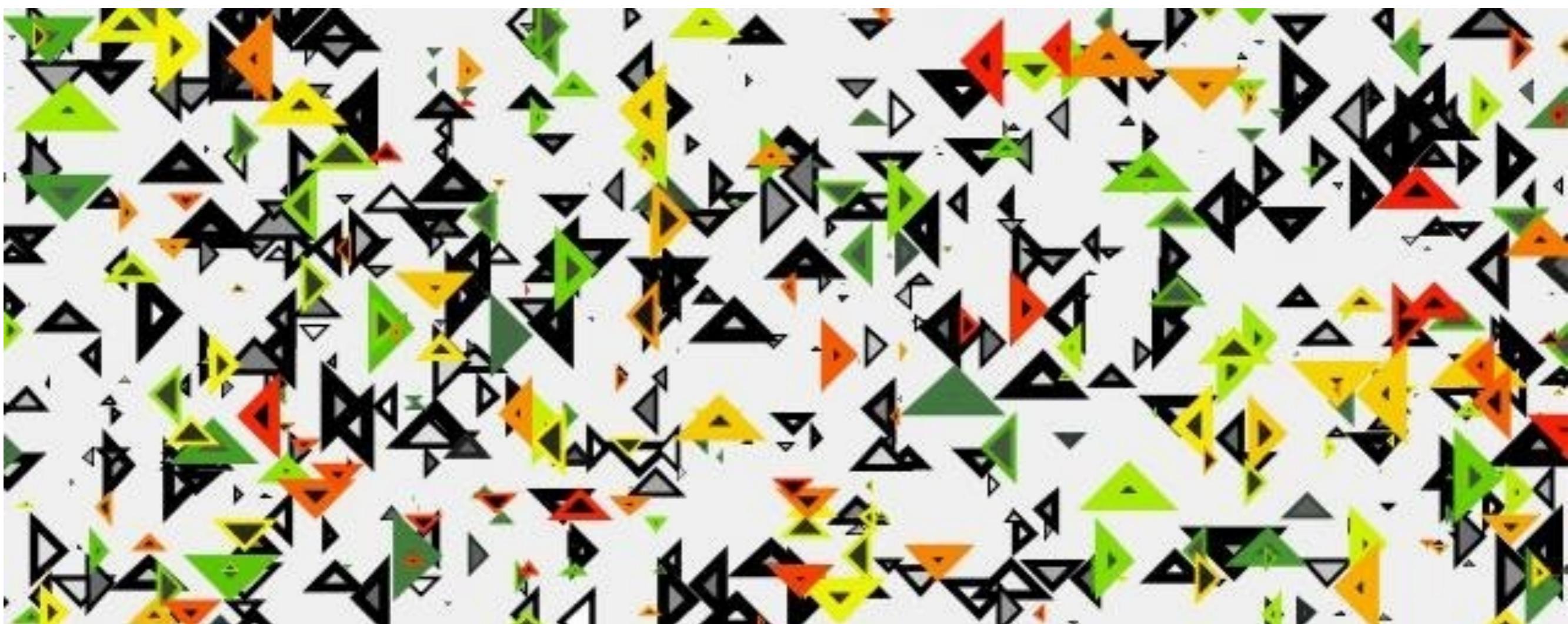


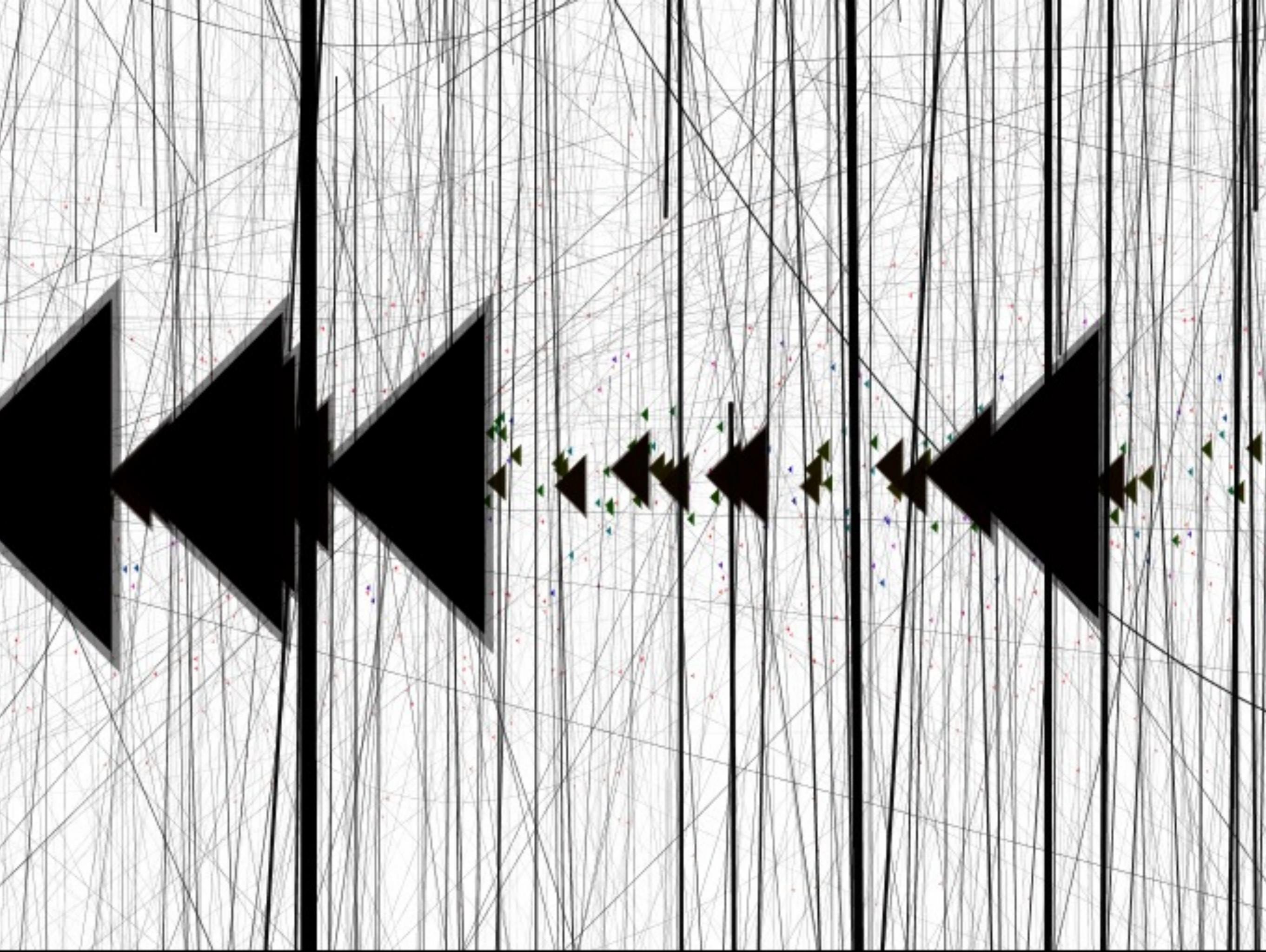


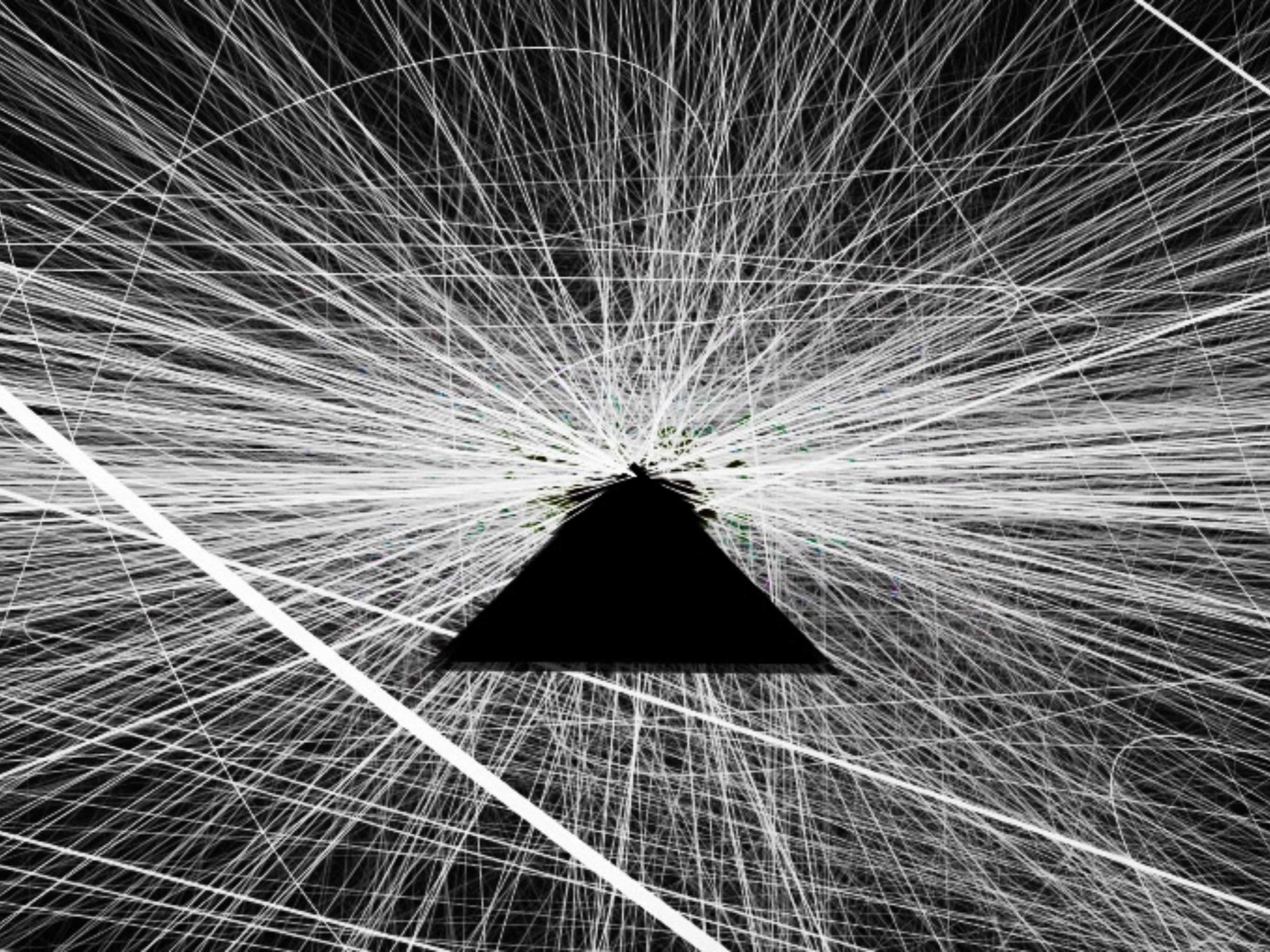
Generative visuals

voorbeeld 2





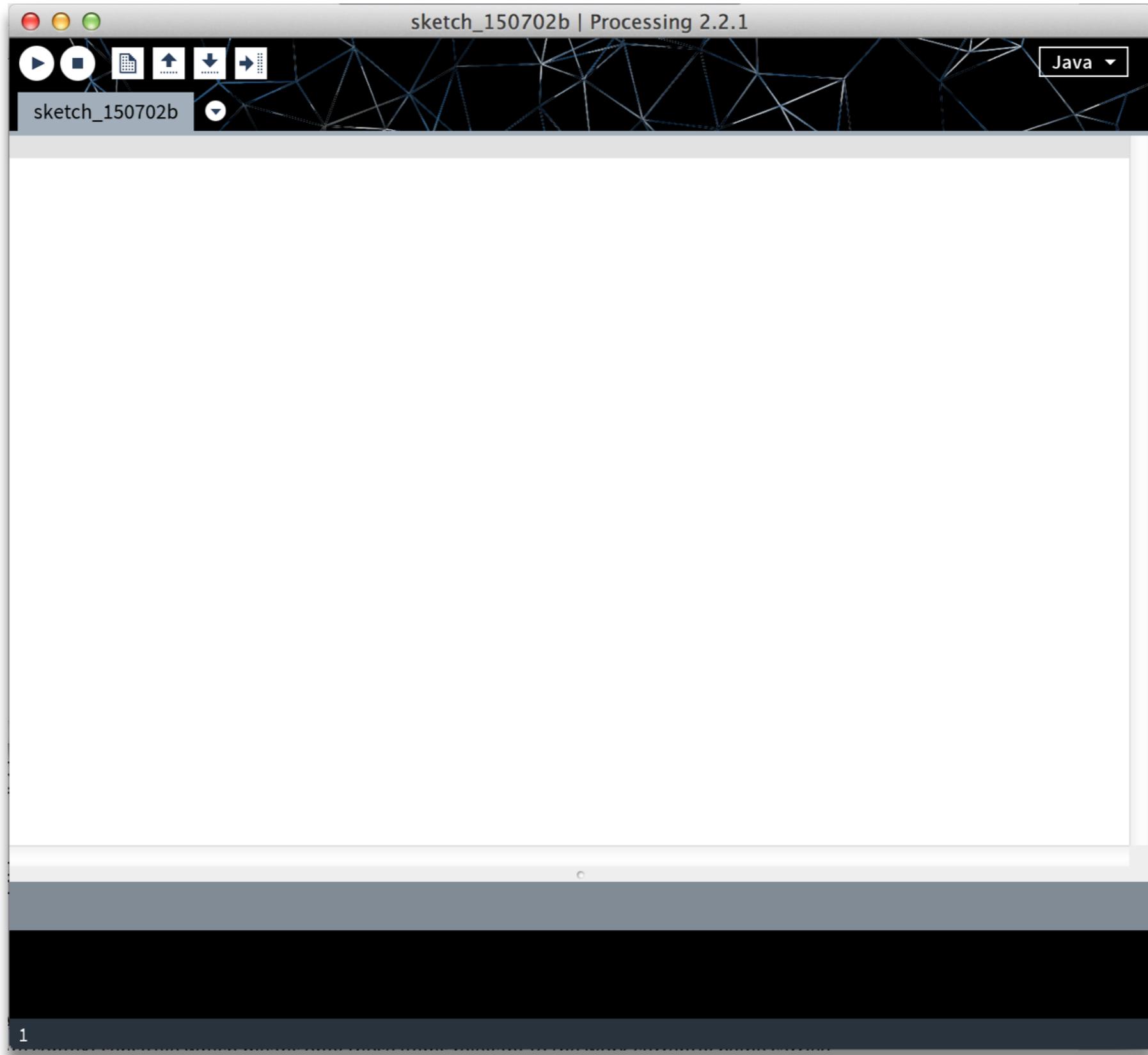




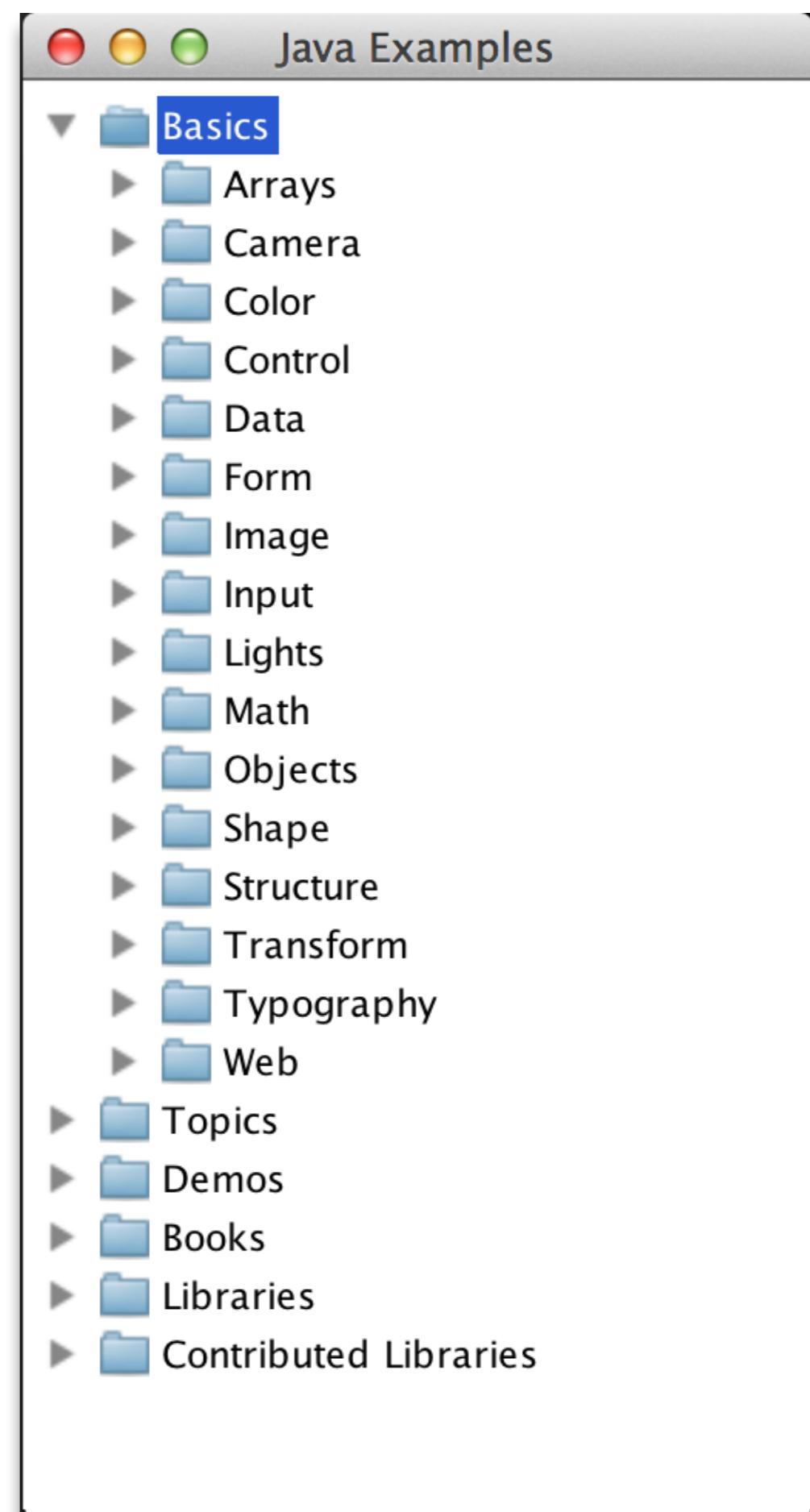
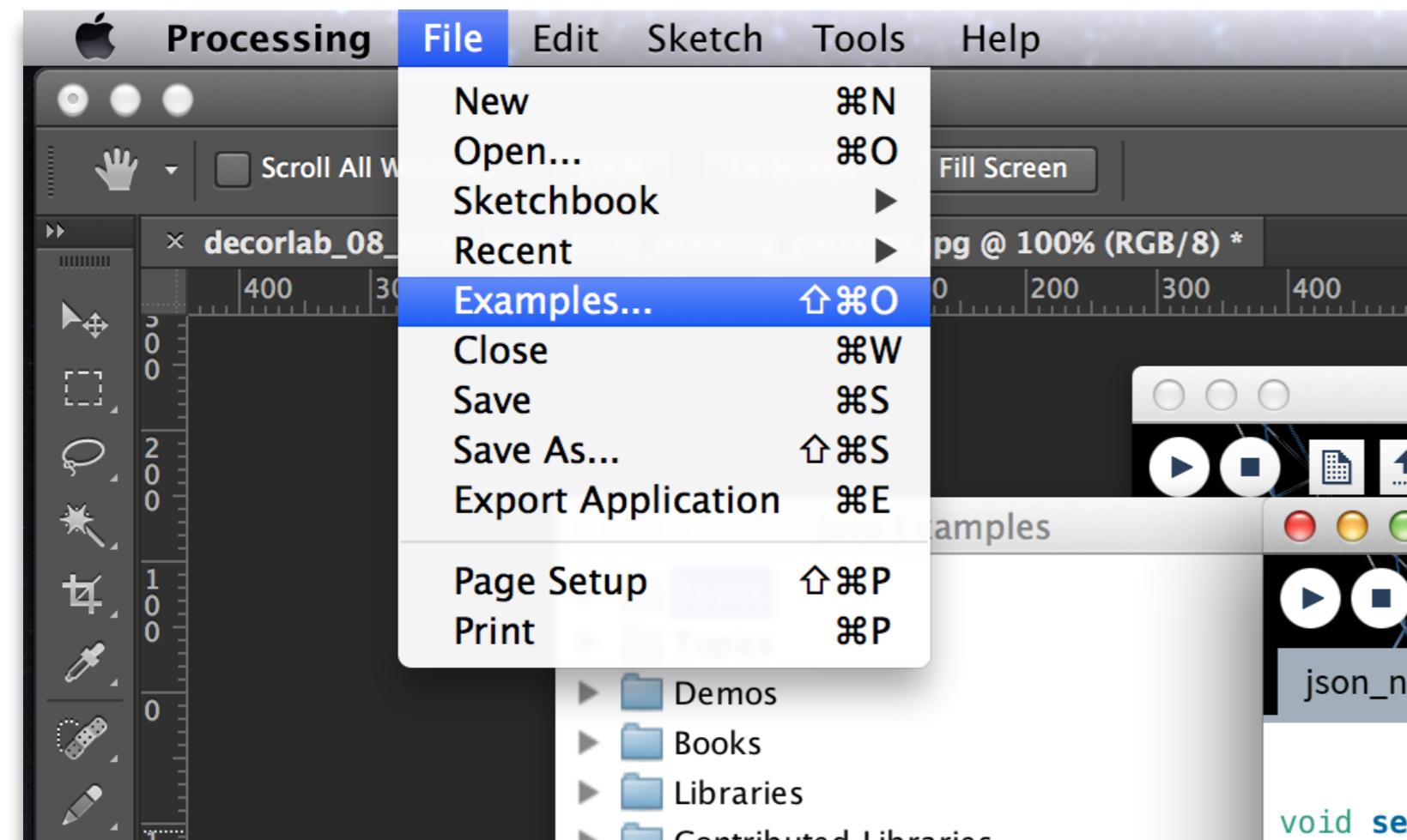
Haiku

<http://www.openprocessing.org/sketch/88203>

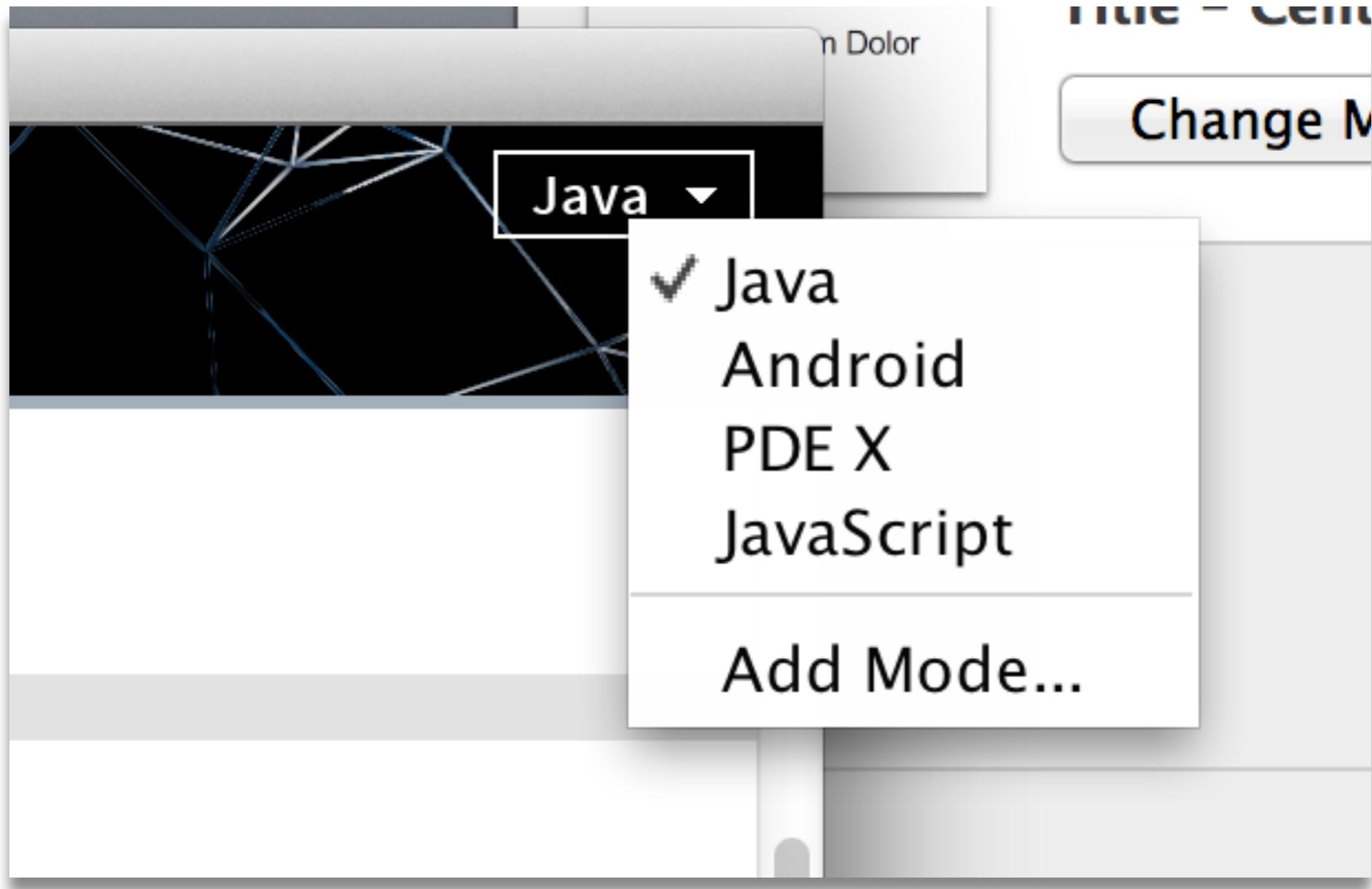
Introductie v/d IDE



Live demo-uitleg van de IDE
(zie ook: <https://processing.org/reference/environment/>)



Immens veel examples standaard bij
Processing.. heerlijk om jezelf een
avondje in kwijt te raken!



Enkele export modi

Naast een desktop app kan dezelfde sketch ook worden geëxporteerd naar bijvoorbeeld Android of het web (via HTML5 Canvas m.b.v. processing.js library).

Wat gaan we vanavond doen?

- Processing leren d.m.v. generatieve graphics!
- In teams van twee (pair-programming)

Hello World

... een paar eerste stapjes.

Baby steps

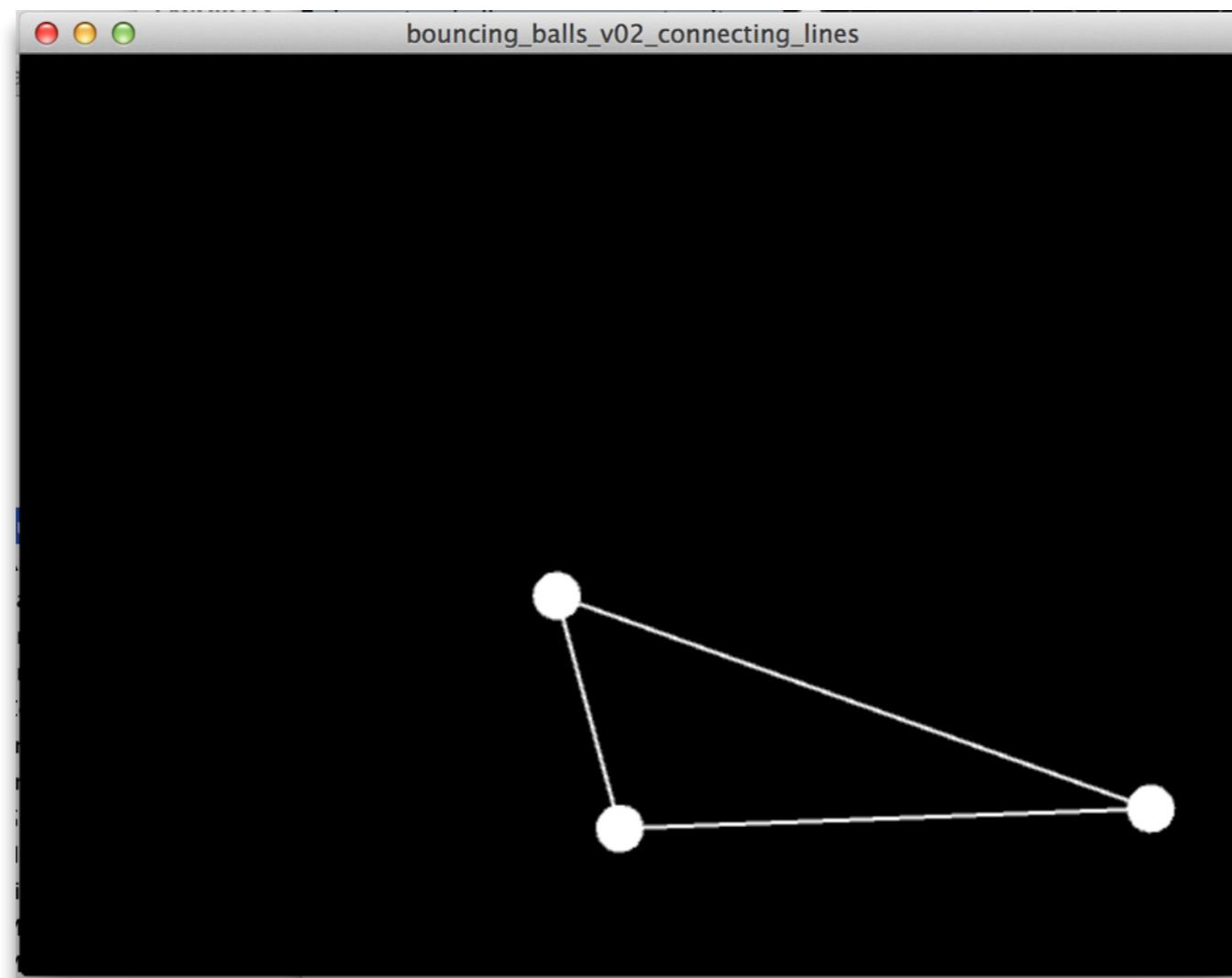
- Maak een nieuwe sketch
- Teken één van de 2D primitives (bv. ellipse, rect, triangle) (zie de Processing reference)
- Laat muis-positie de kleur van de primitive bepalen
- Laat de primitive je muis volgen

Bouncey bouncey

- Maak een systeem wat meerdere van deze primitives over het scherm laat vliegen (zonder friction)
- De randen van het scherm zijn solid.. laat de primitives stuiteren!
- Kan je met keyboard-events (bijv. pijltjes omhoog / omlaag) de hoeveelheid instances variëren?

Lijnenspel

- Je hebt nu de positie van die primitives... laten we nu lijnen gaan tekenen tussen die posities.



Nu gaan we
schilderen!



Dit, bijvoorbeeld, ontstaat door incrementeel tekenen

Tekenen

- Haal de background() in je draw () loop weg.. wat gebeurt er? Tip: teken de lijn heel genuanceerd, met een lage alpha (fill(r, g, b, alpha) of fill(rgb, alpha))
- Nu variëren... spelen met de expressie van die lijn. Verander enkele gedragingen.. maak specifieke conditionele logica zoals dikkere lijnen wanneer twee primitives dichterbij elkaar zijn, of de kleur expressie afhankelijk maken van snelheid, etc.

In dit stadium kunnen er al verassende emergente effecten optreden: visueel gedrag wat je weliswaar zelf hebt geprogrammeerd maar wat z'n eigen leven lijkt gaan leiden, soms met verassend mooie resultaten.

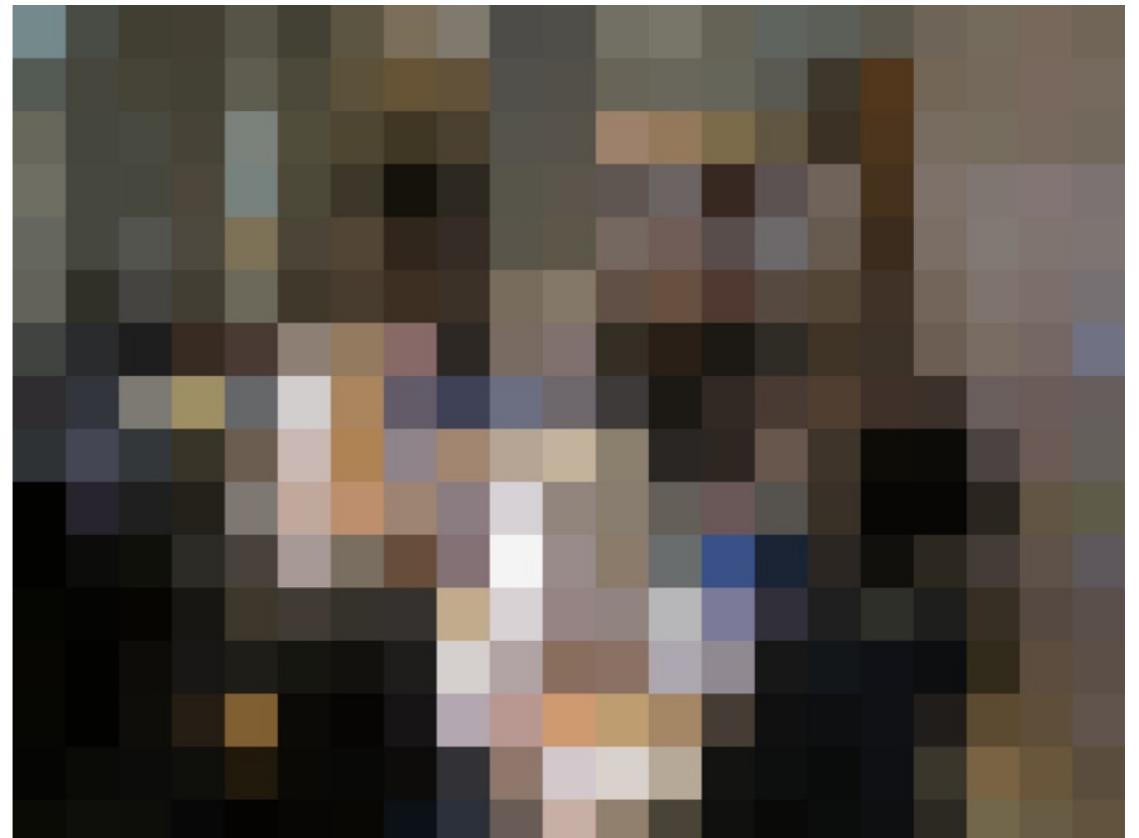
De echte wereld samplen!

- Gebruik de Minim audio library om de expressies van je teken tool (dat is het immers!) te sturen (/Libraries/minim/)
 - Bijv. AnalyzeSound example voor toonhoogte (FFT analysis)
 - Bijv. RecordAudioInput voor amplitude.

Nieuwe oefening:
video

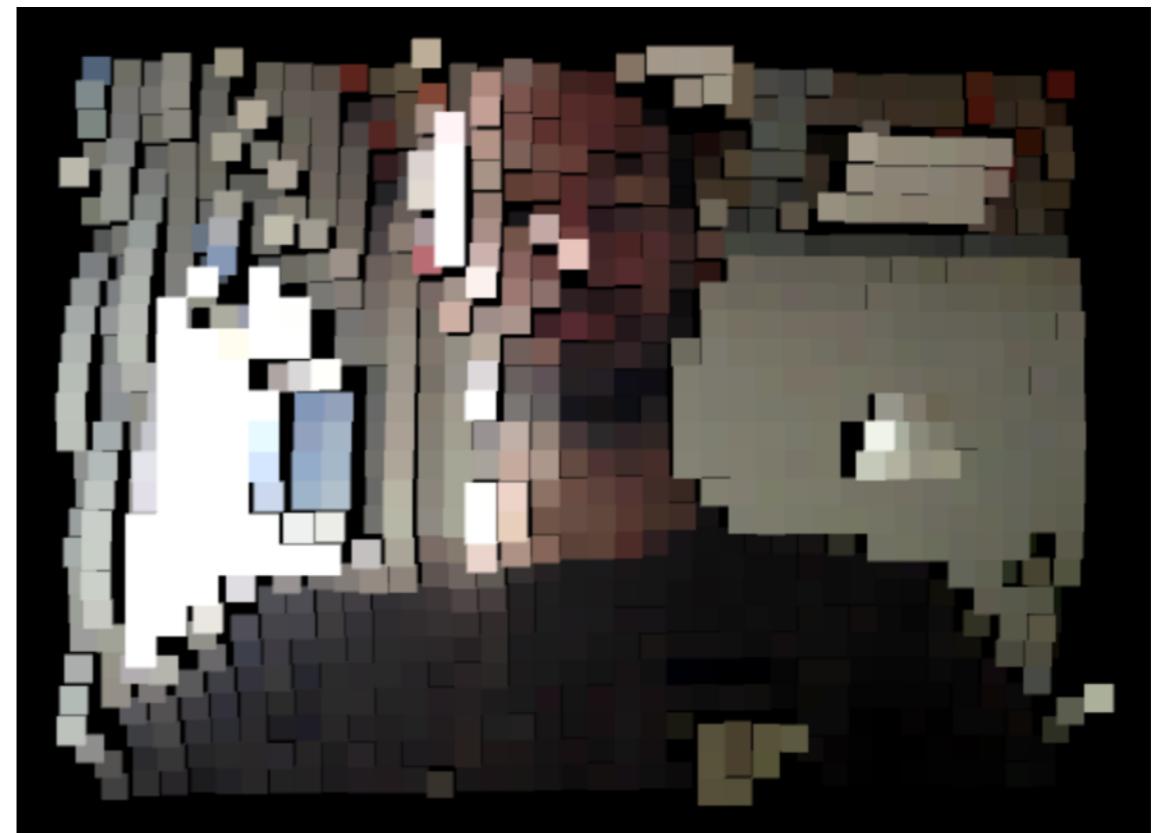
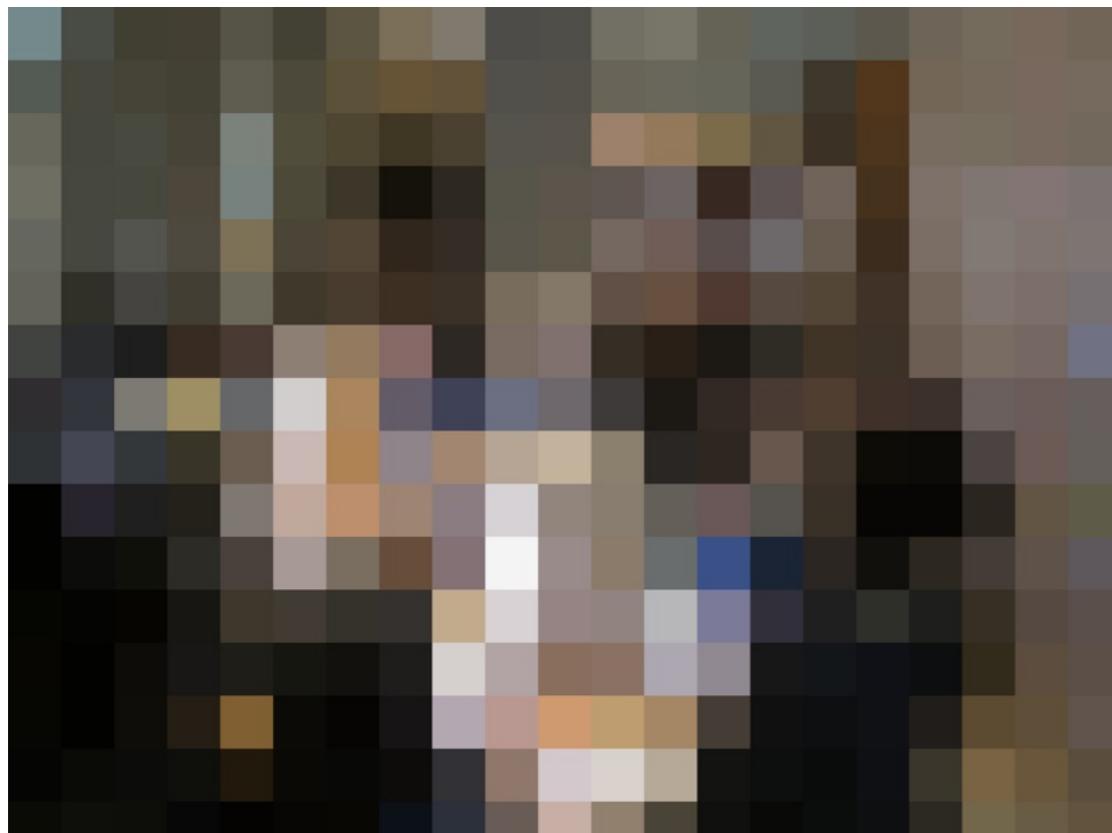
Video

- Leer je webcam gebruiken. Zie /Libraries/video/Capture/GettingStartedCapture om te beginnen.
- Maak een filter waardoor je het videobeeld kunt abstraheren zoals hieronder (hint: loadPixels(), of zie de example / Libraries/video/Capture/BackgroundSubtraction)



Video

- Gebruik brightness () om de kleurvlakken de diepte in de plaatsen. Zorg dat je de P3D renderer gebruikt -> size (width, height, P3D))



Live internet data visualiseren

API + JSON data-objects

regulation
innovation
1981-2008



Voorbeeld van een Processing visualisatie van API data

Eerst: API aanvragen

- Ga naar <http://developer.nytimes.com/>
- registreer -> vraag API key op voor de ‘Article Search API’ categorie (de eerste, als het goed is)
- Bedenk je wat je zou willen opvragen
 - Denk aan gebeurtenissen, thema’s, of patronen die je over tijd zou willen mappen
 - Je gaat het opvragen als keyword(s)
 - evt ingekaderd met begin- en einddatum

API call in Processing

```
String baseURL = "http://api.nytimes.com/svc/search/v2/articlesearch.json";
String apiKey = "je_api_key_van_de_NYT";
String query = "Homer+Simpson"; // url-compatible syntax (bijv. geen spaties)
String begindate = "19900101"; // YYYYMMDD
String enddate = "20150701"; // YYYYMMDD

String request = baseURL + "?query=";
request += query;
request += "&begin_date=" + begindate; // deze is optioneel
request += "&end_date=" + enddate; // deze is optioneel
request += "&api-key=" + apiKey;
String result = join( loadStrings( request ), "" );
JSONObject request_json = JSONObject.parse(result);

// Nu heb je een JSONObject waar je handig mee uit de voeten kan.
```

JSON visualiseren

- Net als in de eerste oefening: begin simpel. Eerst een recht-toe-recht-aan weergave van de data. Van daaruit uitbouwen, uitbouwen, uitbouwen.

Tot dusver...

Bedankt!

- Adriaan Wormgoor
- <http://adriaanwormgoor.nl/>
- info@adriaanwormgoor.nl
- @bombarie)

Resources

- Processing IDE (<https://processing.org>)
 - code reference: <https://processing.org/reference/>
 - voor nerds: werken vanuit Eclipse (meer JAVA!): <https://processing.org/tutorials/eclipse/>
- Rody Middelkoop (Devnology lid / HAN) -> Game API in Processing: <https://github.com/ddoa/gameapi-processing>