

# 情報・工学探究

## 第2回 メディアプログラミング授業資料

見て，感じて，描いてみよう

講義担当：情報学部 情報学科 情報メディ専攻 清水 哲也

## 第2回の流れ

- 解説：生成的アートの考え方
- インタラクティブコード紹介
- 課題制作：「感情」や「音楽」をテーマに

## 解説：生成的アートとは？

- 「コンピュータの偶然性」を活かしたアート
- 例：毎回違うパターンが生成される
- プログラムがルールを与え，結果は多様

## 有名な生成アートの例

- Generative Design
- Vera Molnár
- Tyler Hobbs (Fidenza)


# Generative Design

- 書籍『Generative Design』シリーズが有名
- JavaScript / p5.js で造形ルールを実装
- 「秩序＋偶然」で構成される視覚表現
- 公式サイト：[generative-gestaltung.de](http://generative-gestaltung.de)

## Generative Design：作例

- グリッド内でのシンプルな円の配置：[Link](#)
- マウス位置による円弧の回転と重ね描き：[Link](#)
- 画像をピクセル単位で分析し円で再描画：[Link](#)
- タイポグラフィの一部を動的に再構成：[Link](#)
- ランダムなラインの生成と重なり：[Link](#)


# Vera Molnár ( ヴェラ・モルナール )

- ハンガリー出身，コンピュータアートの先駆者
- 1960年代から生成的な図形に取り組む
- 「規則」と「わずかなズレ」を重視した作品群
-  公式サイト：<http://www.veramolnar.com/>

作品タイトル	内容	関連リンク
(Des)Ordres (1974)	完全な秩序（ordre）と、それを少しずつ壊していく（désordre）というテーマ。グリッドの構成要素が徐々にズレていく構成。	<a href="#">DAM.org</a>
Interruptions (1968)	グリッドに配置された線分を「意図的に欠けさせる」ことで、視覚的なリズムや緊張感を作り出す	<a href="#">DAM.org</a>
1% de désordre (1976)	99%の規則と1%のズレというテーマで、ほとんど規則通りの構造に“わずかな逸脱”を持ち込む	<a href="#">The Art of Vera Molnar 1947-1974</a>
Structure de Quadrilatères (1972)	正方形や四角形の回転・重なり・変形により、単純な幾何学構造から複雑な視覚効果を生む	<a href="#">The MFAH Collections</a>
Computer Drawings (1970年代)	MolnárがFORTRANを用いて制作した初期のプロッターローイングの総称。規則性と偶然性の探究	<a href="#">MoMA</a>



# Tyler Hobbs（タイラー・ホッブス）

- 現代の代表的ジェネラティブアーティスト
- 作品「Fidenza」でNFT界でも注目
- 有機的・抽象的な美しさが特徴
-  公式サイト：[tylerxhobbs.com](https://tylerxhobbs.com)

# Tyler Hobbs : 作例 ( Fidenza )



## Fidenza ( フィデンツァ )

- 概要：2021年に発表された，999点からなるジェネラティブアートのシリーズ
- 特徴：「フローフィールド」アルゴリズムを用いて，有機的で流れるような曲線と色彩のバリエーションを生成
- Link：[Fidenza 公式ページ](#)

# 比較まとめ

作家	特徴	技法例
Generative Design	デザイン寄り，実装例が豊富	グリッド・反復・配色制御
Vera Molnár	美術・理論の重視，初期世代	幾何学・ズレ・規則の操作
Tyler Hobbs	現代的・自然な造形	色・形・構造の融合

# 今日の課題

## 『感情』や『音楽』をテーマに

- 自分だけのデジタルビジュアル作品を制作
- テーマ例：好きな曲，気分，思い出
- 色，形，動きの組合せで気持ちを表現

## 制作のヒント

- `random()` で偶然性を加える
- `mouseDragged()` や `touchStarted()` で操作を加える
- カラーコードや円・線の組合せで自分らしさを
- 背景・透明度で“余白”を表現

## 参考リンク

- [p5.js公式サイト](#)
- [p5.js日本語公式サイト](#) ※少し情報が古いかも
- [p5.jsリファレンス](#)
- [p5.js 初めの一步 Creative Coding p5.js - HIM.CO ヒム・カンパニー](#)
- [文系大学生のためのp5.js入門](#)

## 【付録】 使用した関数まとめ

# 基本構成

- `function setup()` : 画面サイズ, 背景色, 枠線描画の有無などを設定
- `function draw()` : この関数内は無限ループ, 描画したい命令を書く

```
function setup() {  
  createCanvas(400, 400);  
}  
function draw() {  
  background(255);  
}
```



## 色・背景・線

- `r`, `g`, `b`, `gray` : 0～255までの値
- <https://www.colordic.org/> : このサイトを参考にすると値がわかりやすい
- `noStroke()`, `noFill()` : 枠線を描画しない, 塗りつぶしをしない設定になる
- 色を指定したらそれより下の描画はすべて指定した色になる

```
fill(r, g, b);  
stroke(r, g, b);  
background(gray);  
noStroke();
```

## 図形（楕円と円）

- `ellipse(x, y, w, h)` : 座標 `(x, y)` を中心に高さ `w` , 幅 `h` の楕円を描画
- `circle(x, y, d)` : 座標 `(x, y)` を中心に直径 `d` の円を描画

`ellipse`(中心のx座標, 中心のy座標, x方向の直径, y方向の直径)  
`circle`(中心のx座標, 中心のy座標, 直径)

サンプル : <https://editor.p5js.org/shimizu-sit/sketches/jtttc3wUeN>

## 図形（長方形と正方形）

- `rect(x, y, w, h)` : 左上の角が座標 `(x, y)` の幅 `w` , 幅 `h` の長方形を描画
- `square(x, y, s)` : 左上の角が座標 `(x, y)` の一辺 `s` の正方形を描画

`rect`(左上隅のx座標, 左上隅のy座標, x方向の長さ, y方向の長さ)  
`square`(左上隅のx座標, 左上隅のy座標, 一辺の長さ)

サンプル : <https://editor.p5js.org/shimizu-sit/sketches/Mp3X0mkik>

## 図形（三角形と四角形）

- `triangle()` : 三角形の3つの頂点 `(x1, y1)`, `(x2, y2)`, `(x3, y3)` を指定して描画
- `quad()` : 四角形の4つの頂点 `(x1, y1)`, `(x2, y2)`, `(x3, y3)`, `(x4, y4)` を指定して描画

```
triangle(x1, y1, x2, y2, x3, y3)  
quad(x1, y1, x2, y2, x3, y3, x4, y4)
```

サンプル : <https://editor.p5js.org/shimizu-sit/sketches/If8KWF05k>

## 図形（線と線の太さ）

- `line()` : 始点 (x1, y1) から終点 (x2, y2) へ線を描画
- `strokeWeight()` : 線の太さを指定

```
strokeWeight(線の太さ);  
line(始点のx座標, 始点のy座標, 終点のx座標, 終点のy座標);
```

サンプル : [https://editor.p5js.org/shimizu-sit/sketches/Kq1WjnNz\\_](https://editor.p5js.org/shimizu-sit/sketches/Kq1WjnNz_)

# マウス操作

```
mouseX, mouseY  
mousePressed()  
mouseDragged()
```

# ランダム

```
random(255);  
random(width);
```

# アルファ（透明度）

```
fill(r, g, b, alpha);
```



# テキスト

```
text("Hello", x, y);  
textSize(24);
```

# 変数

同じ値を何度も使う場合や、値を一次的に保存する場合に使います

```
let radius = 200;  
ellipse(200, 200, radius, radius);
```

おつかれさまでした！