PAT 204/504 (Fall 2026)

Creative Coding

Lecture 1: Introduction

Instructor: Hao-Wen (Herman) Dong

UNIVERSITY OF MICHIGAN

Creative **Coding**

Creative Coding

What is this course all about?

An introduction to principles and practices of **computer programming for musical applications**. Emphasis is on **creative and artistic uses of code**.

Processing

Max

Learning Objectives

- Gain an understanding of **programming for music and multimedia**
- Learn fundamentals **computer programming concepts**
- Gain hands-on experience on **implementing audiovisual systems**

About Me

- Hao-Wen (**Herman**) Dong
- Pronouns: he/him
- Email: **hwdong**@umich.edu
- Office: Stearns 131 (15 min walk to the north from Moore)
- Office hours: By appointments (link on the course website)
- Research areas: Generative AI for music, audio, and video

Welcome! Tell Us about Yourself!

- Name
- Pronouns
- Program/year
- What is your **most familiar instrument** (if any)?
- Have you ever coded? Which **programming language**?

Intro to Processing

What is Processing?

- A **free** and **open-source** programming language
- Built for electronic arts, new media art and visual design
- Easy to get started with!
- Based on **Java**
  - 3rd most popular programming language on GitHub
  - Can leverage the power of Java

Processing

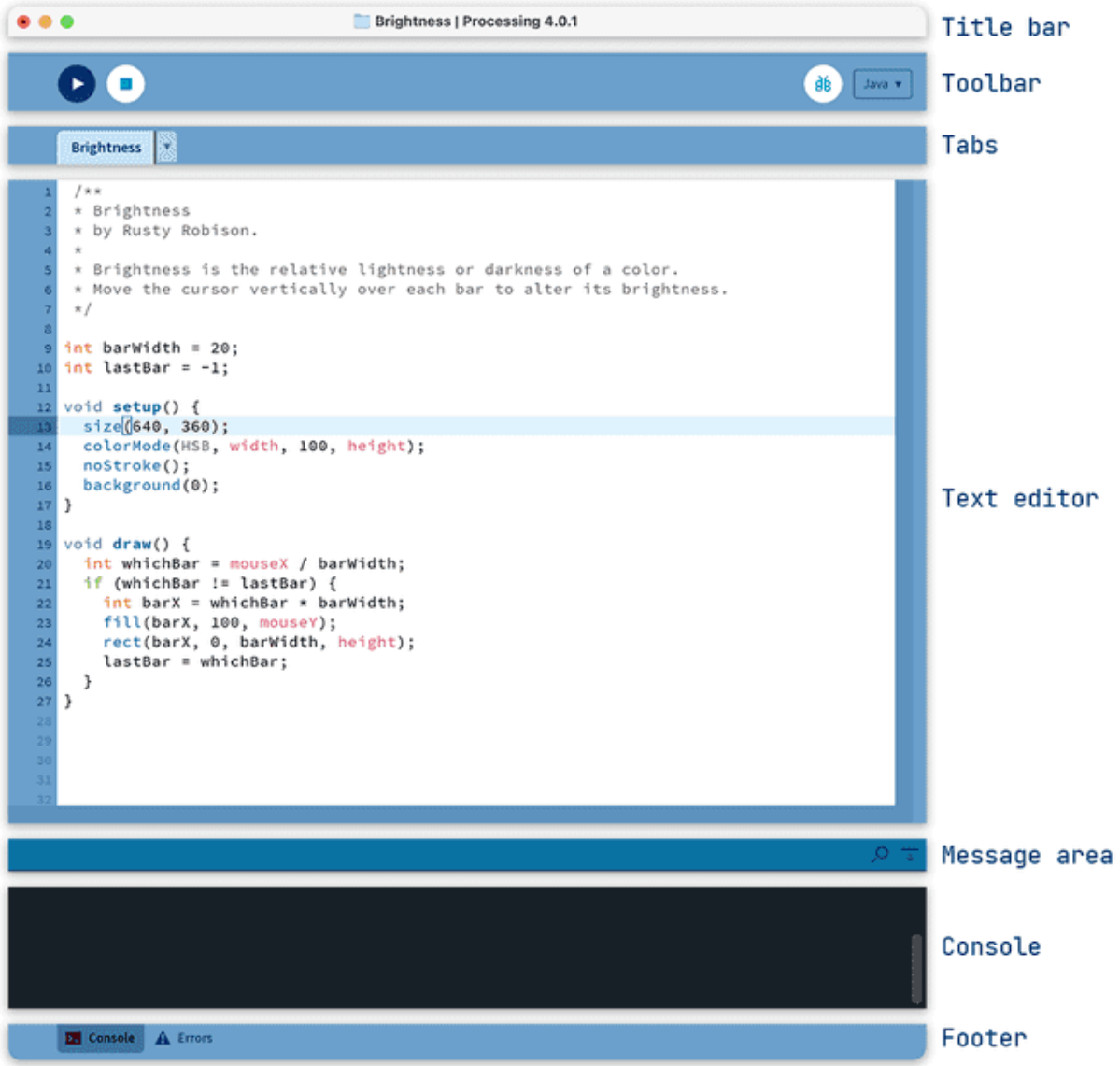
Radiohead: House of Cards (2007)

github.com/dataarts/radiohead

A Processing Sketch

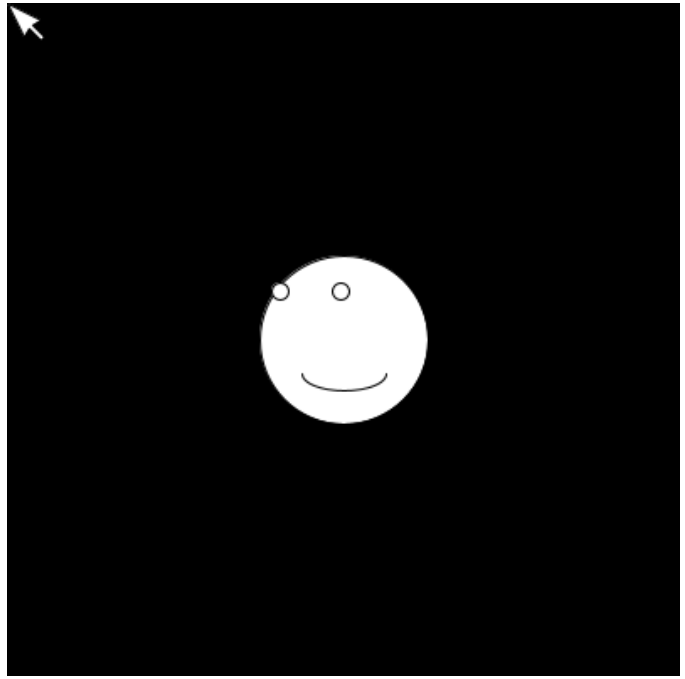
- Processing comes with an IDE
  (Integrated Development Environment)
  - A **text editor**
  - A **console**
  - A **display window** (when you click the *run* button)

Display window

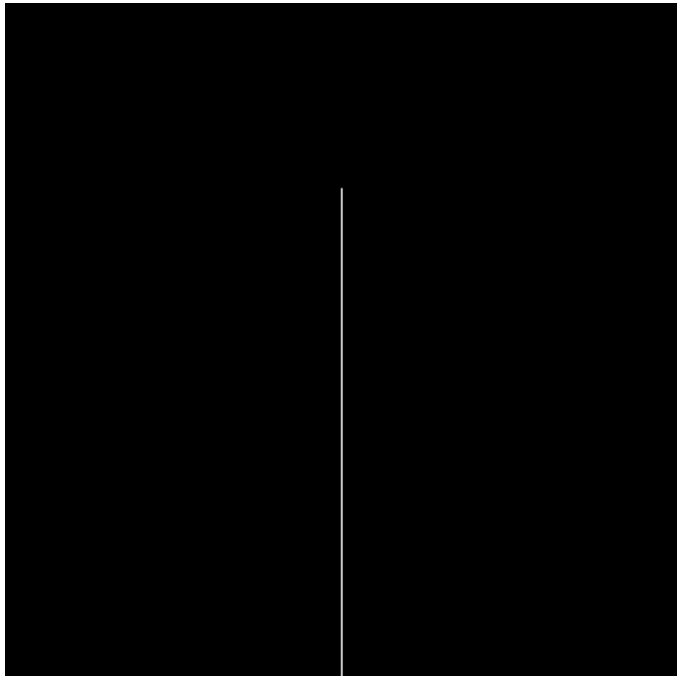


Example Processing Sketches

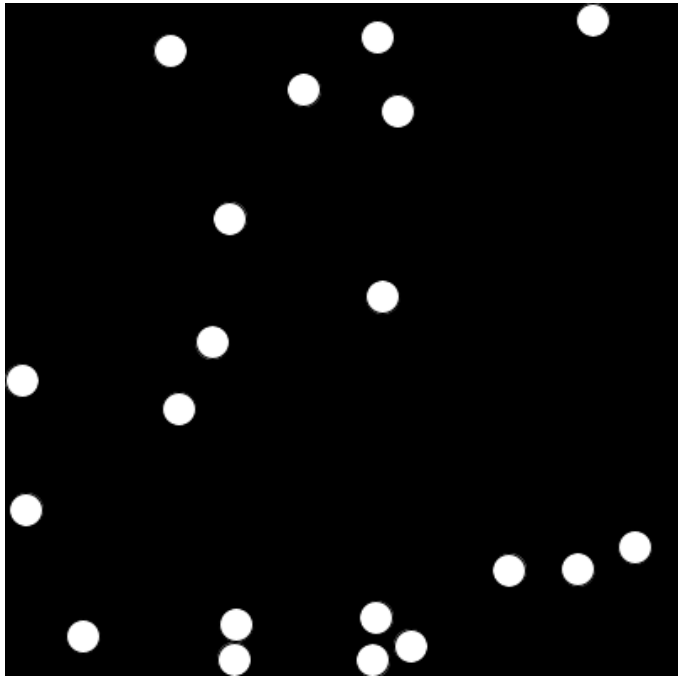
Creepy Eyes



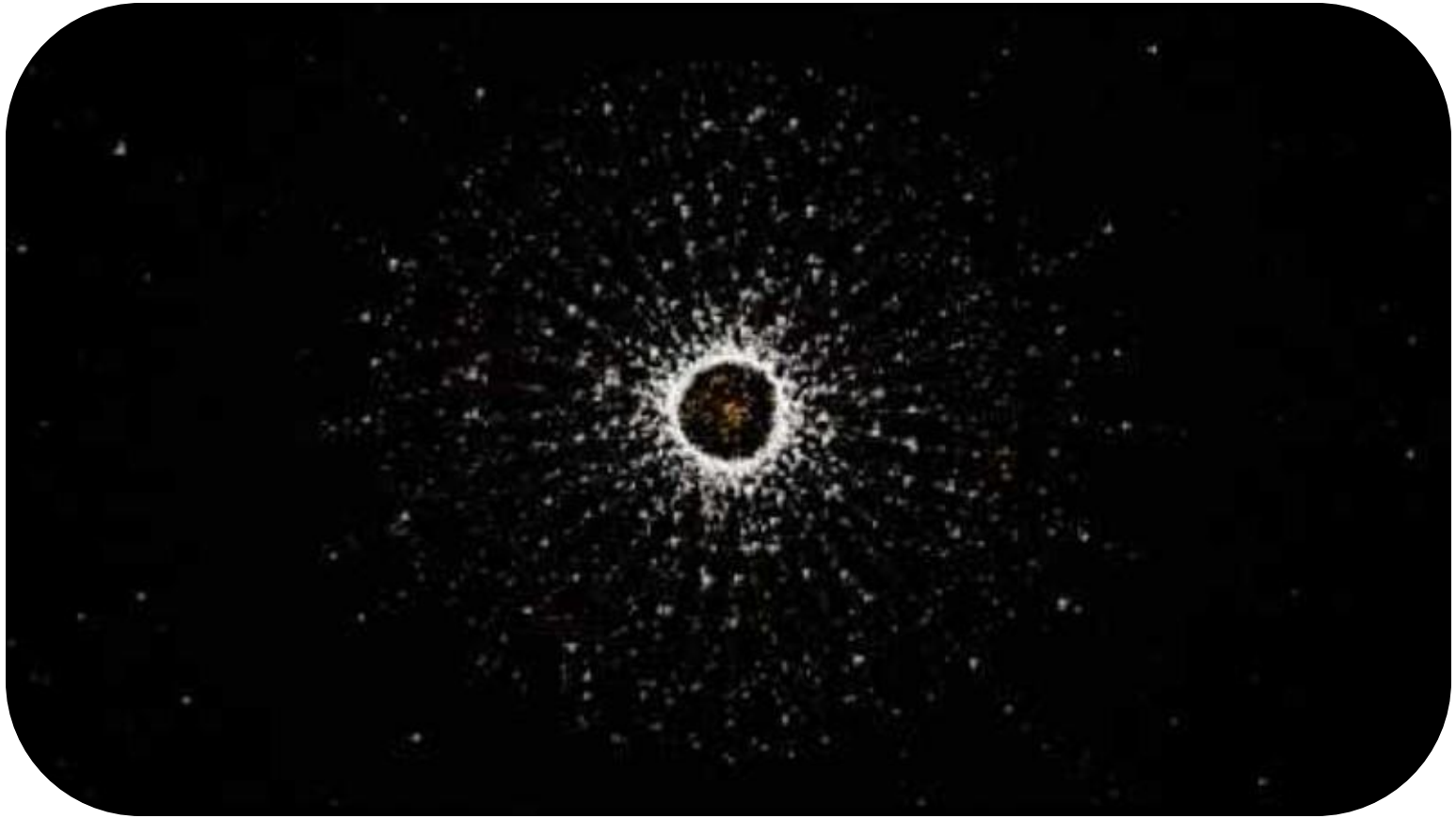
Recursive Tree



Bouncing Balls



Music Visualizer



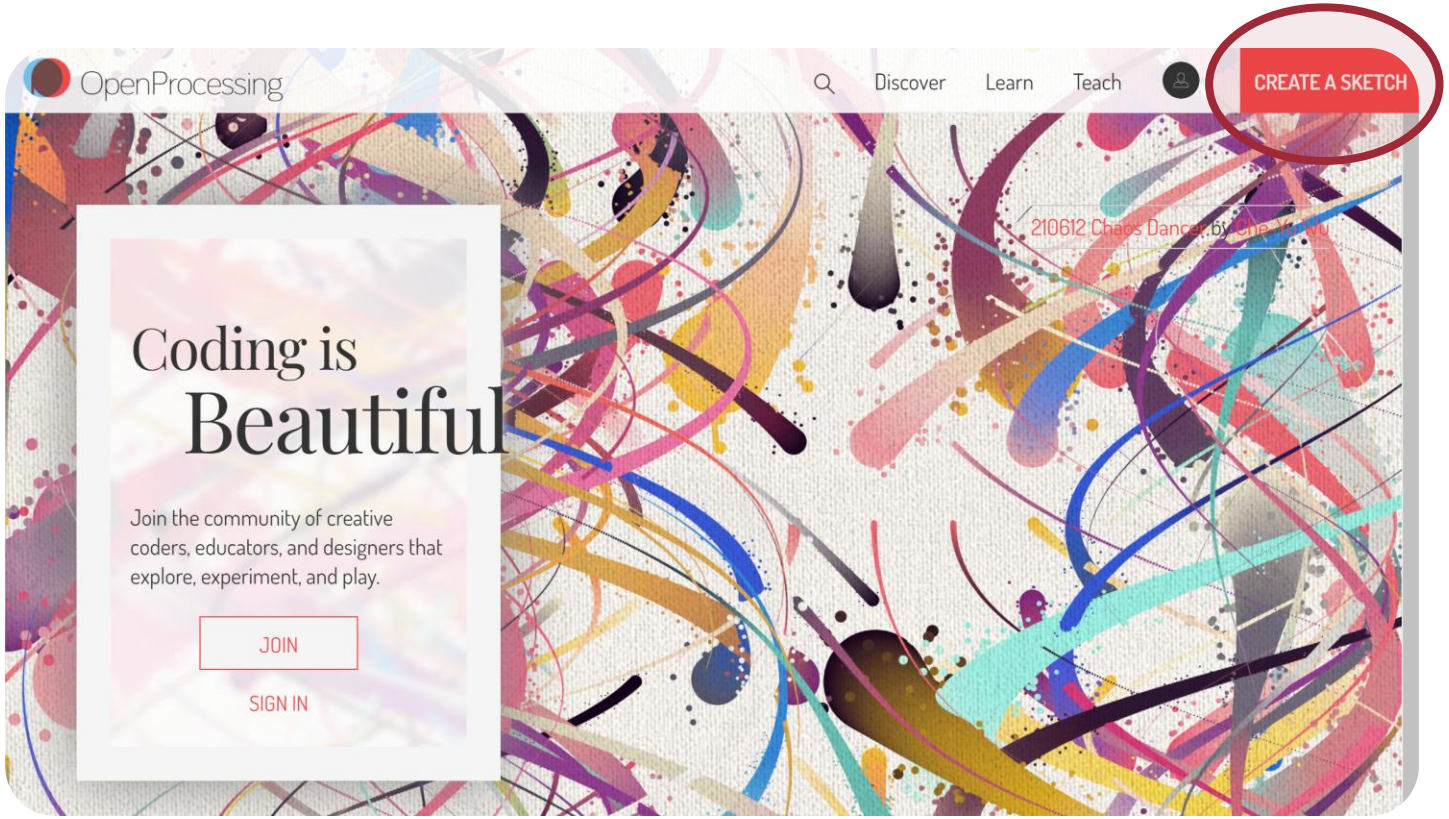
youtu.be/283rmgvFDE0 & pastebin.com/JtAn1mV5

Generative Portraits

youtu.be/ZzNO3FvkTJM

OpenProcessing

- Large community of creative coders, educators and designers!

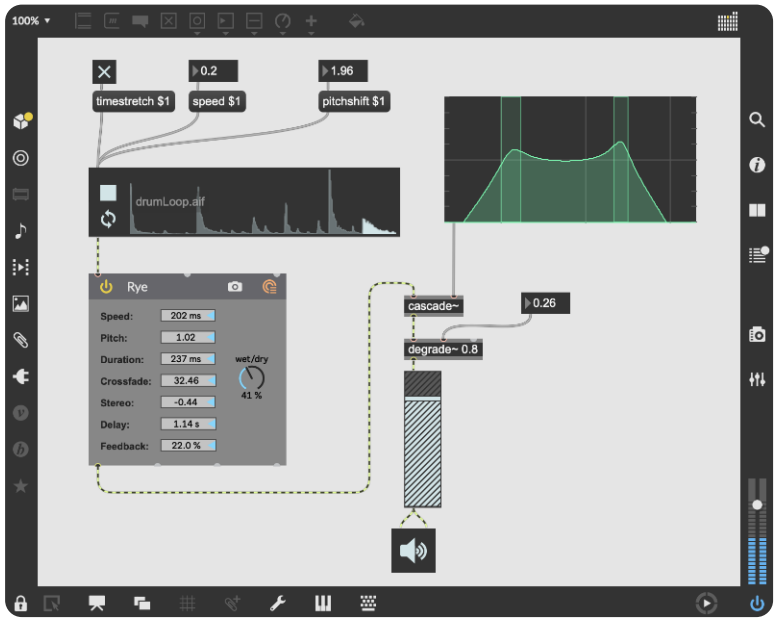


openprocessing.org

Intro to Max

What is Max?

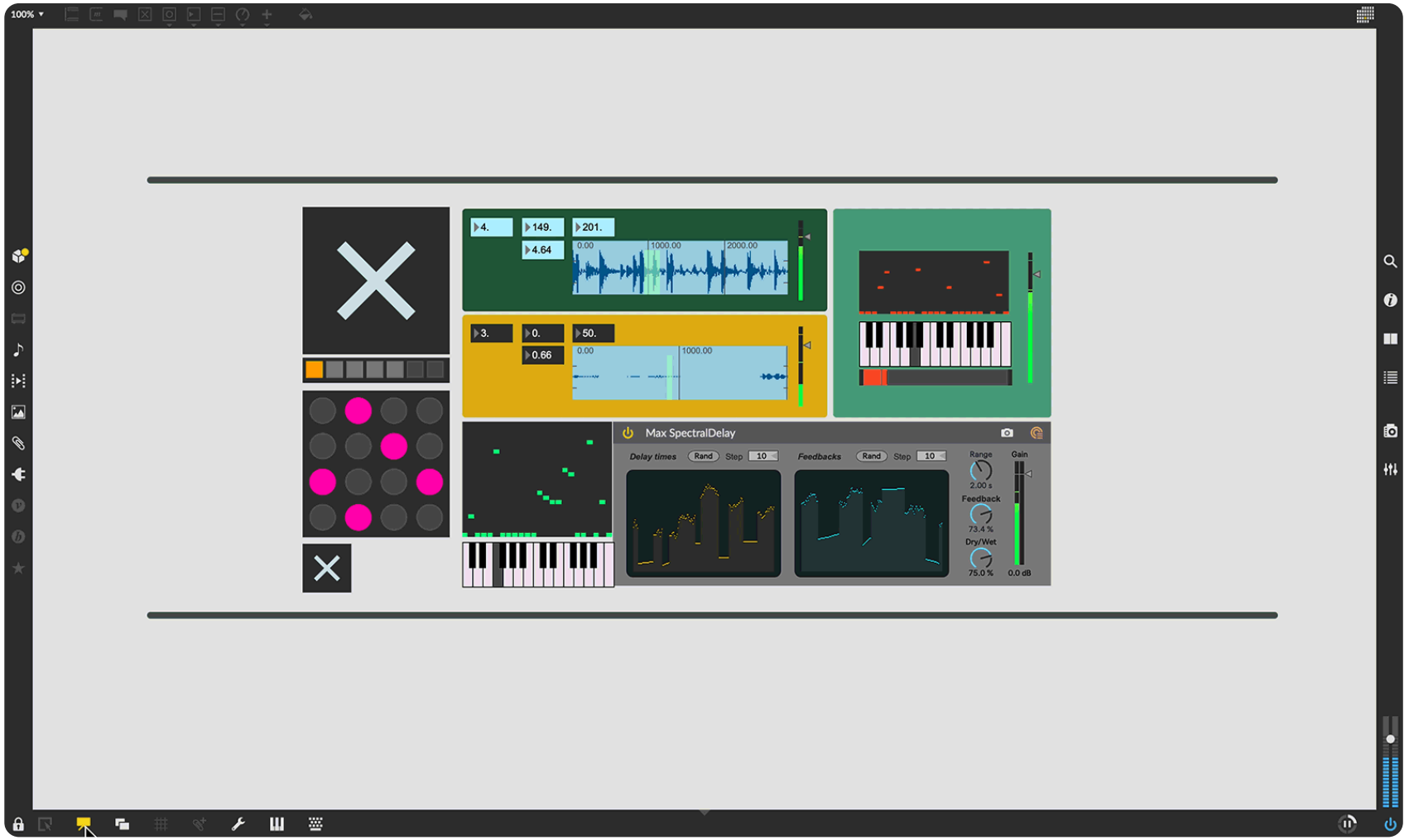
- A **visual programming language** for music and multimedia
- Generate, analyze, and manipulate **signals** (audio, video, etc.) in real time
- Steep learning curve… but we'll get through it!



(Source: Cycling '74)

cycling74.com/products/max

Max's Interface

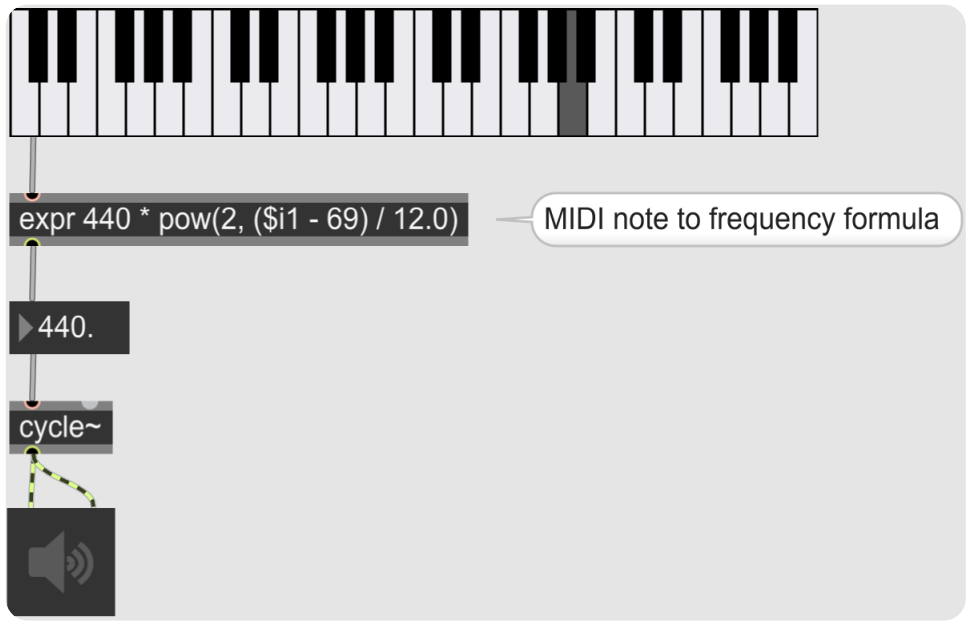


(Source: Cycling '74)

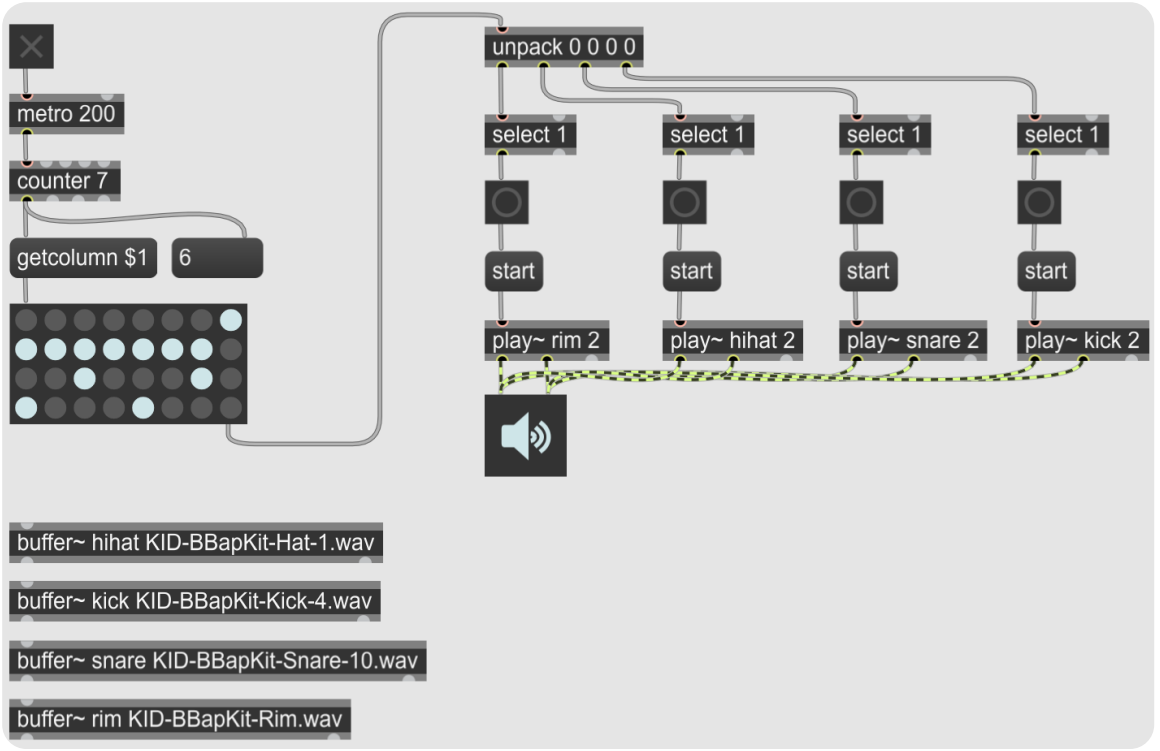
cycling74.com/products/max

MSP: Signal Processing for Max

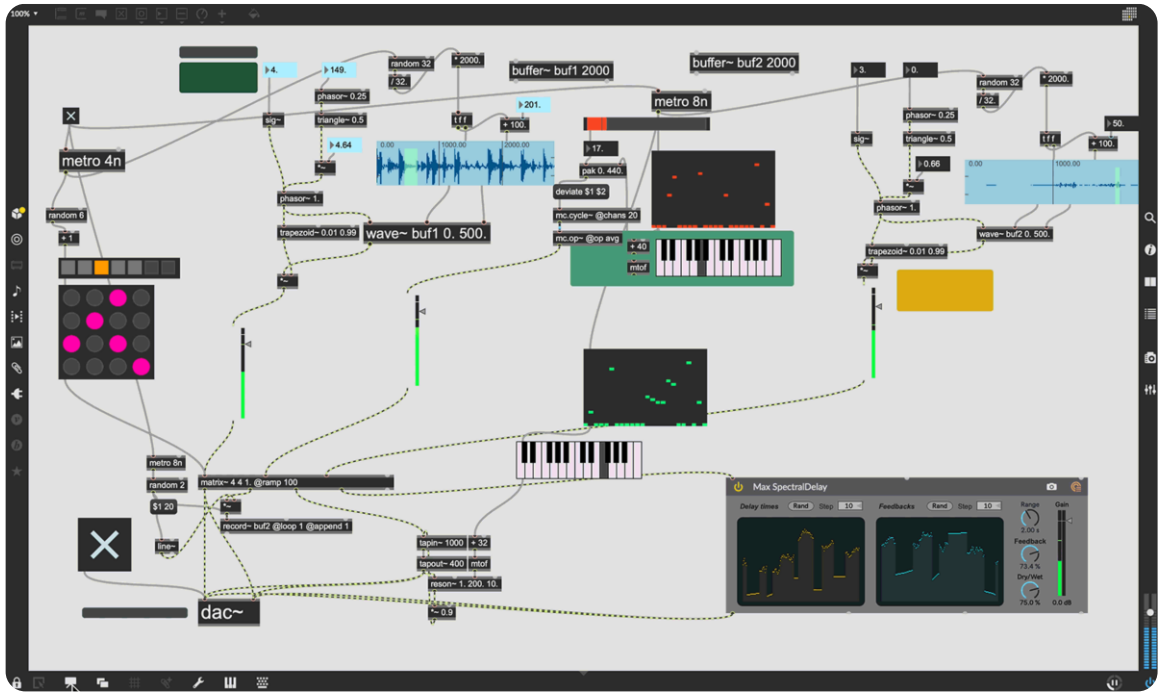
MIDI Keyboard



Drum Machine



Max/MSP vs. Analog Synthesizers



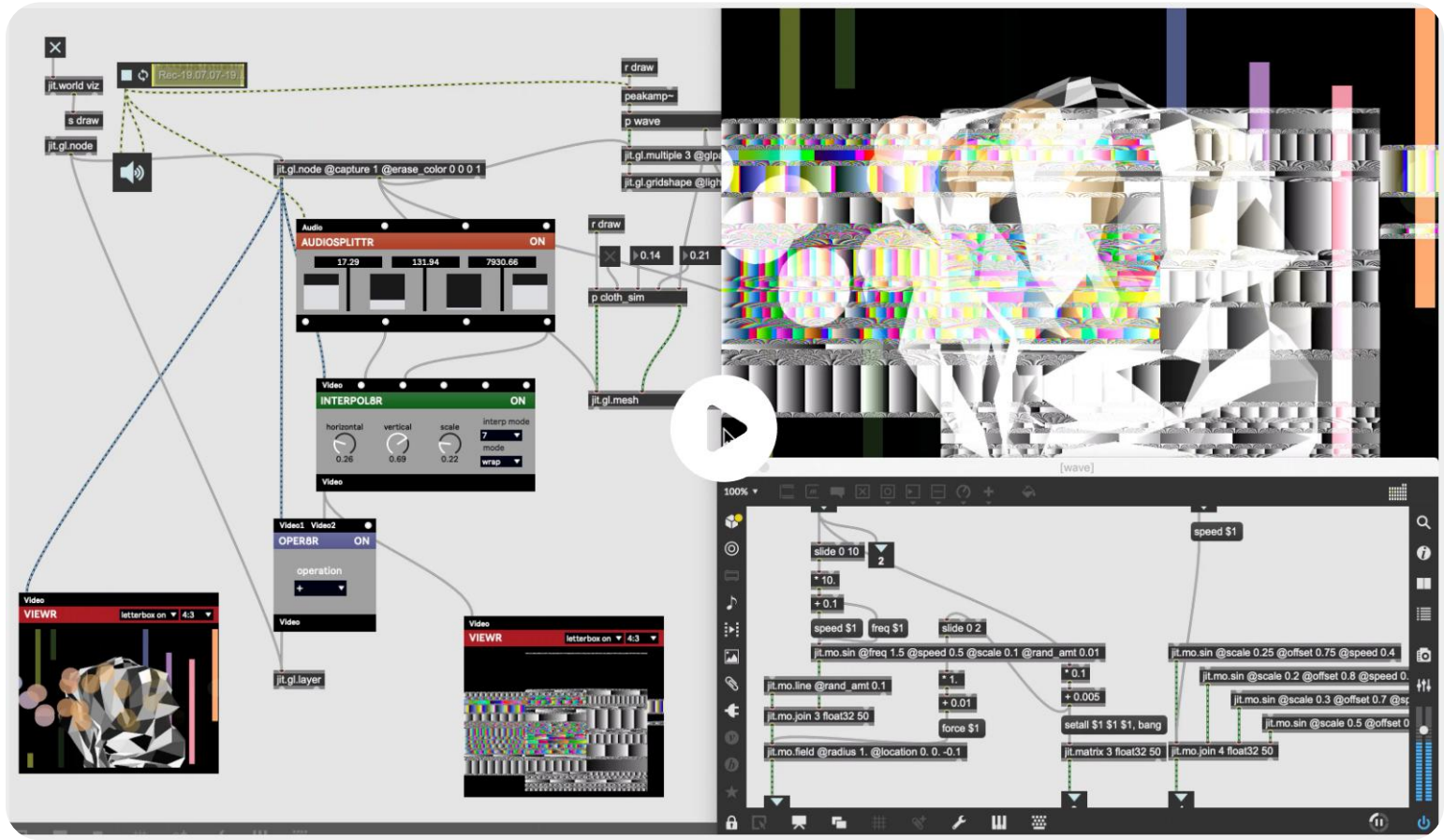
(Source: Cycling '74)

(Source: Wikimedia Commons)

cycling74.com/products/max

Pete Brown from Gambrills, MD, USA, CC BY 2.0, via Wikimedia Commons

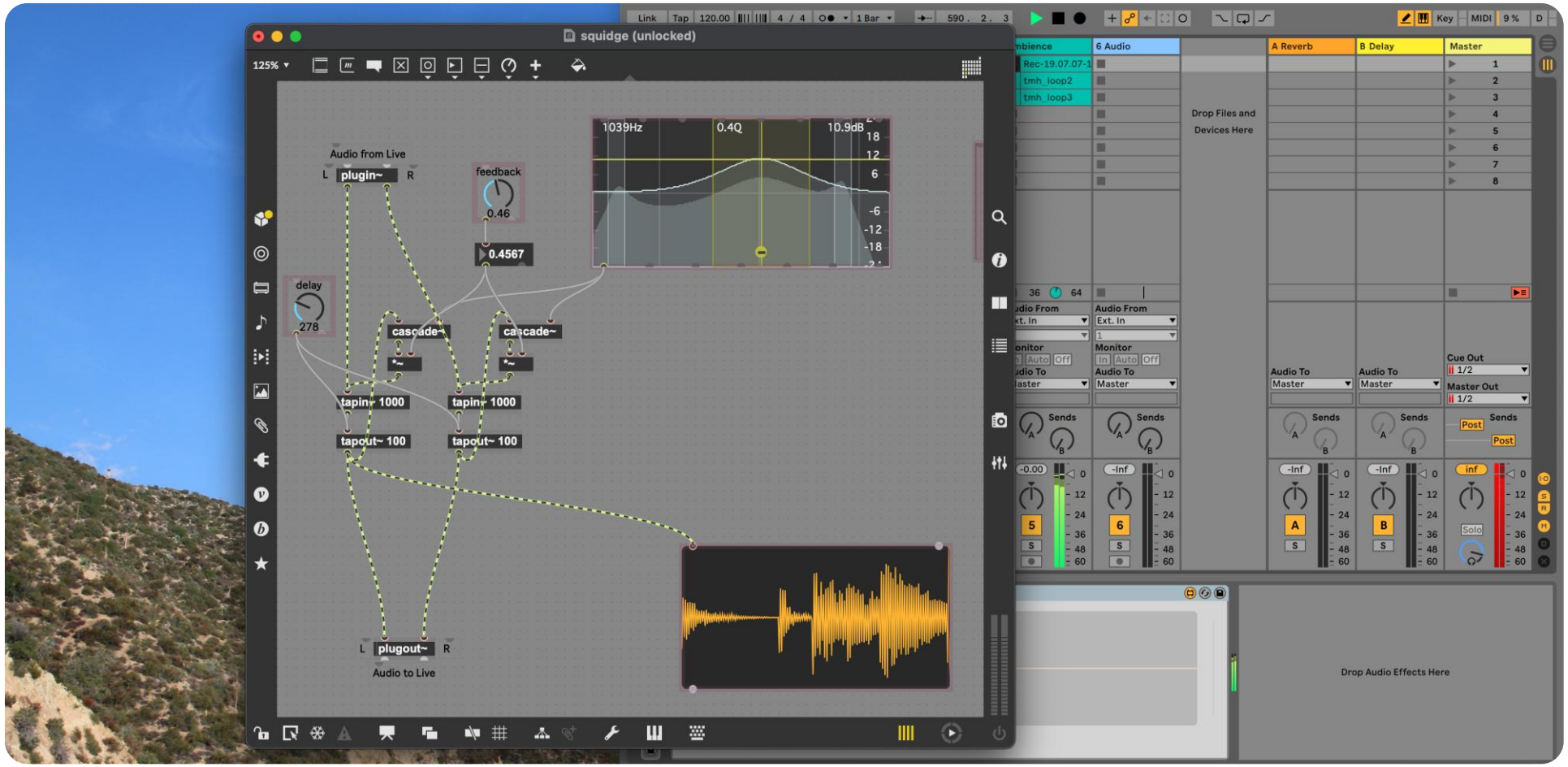
Jitter: Visuals for Max



(Source: Cycling '74)

cycling74.com/products/max

Max for Live: Running Max in Ableton Live



(Source: Cycling '74)

cycling74.com/products/maxforlive

What is this course all about?

An introduction to principles and practices of **computer programming for musical applications**. Emphasis is on **creative and artistic uses of code**.

Processing

Max

Course Logistics

Communications

- **Course website**: Syllabus, schedule, materials, recordings, etc.
- **Email**: Announcements
- **Google Chat**: Q&A

hermandong.com/teaching/pat204

Prerequisites

- **None!**
- We'll learn to code in Processing and Max **from scratch**!

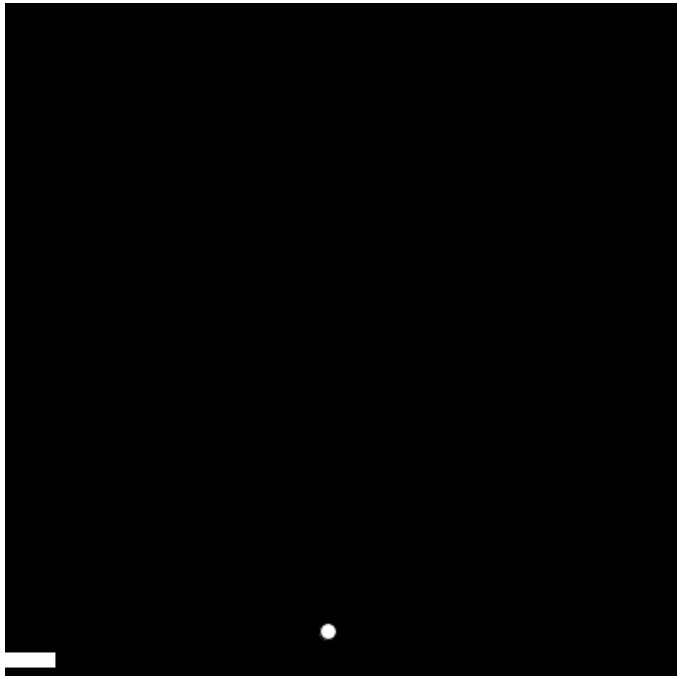
(Tentative) Assignments

- **Homework**
  - Bouncing “Hello, World!”
  - Paddle ball game
  - Solar system
  - Spectrum visualizer
  - Simple synth
  - MIDI keyboard
  - Polyphonic FM synthesizer
  - Singing balls
- **Midterm assignment**
  - Build your own music visualizer

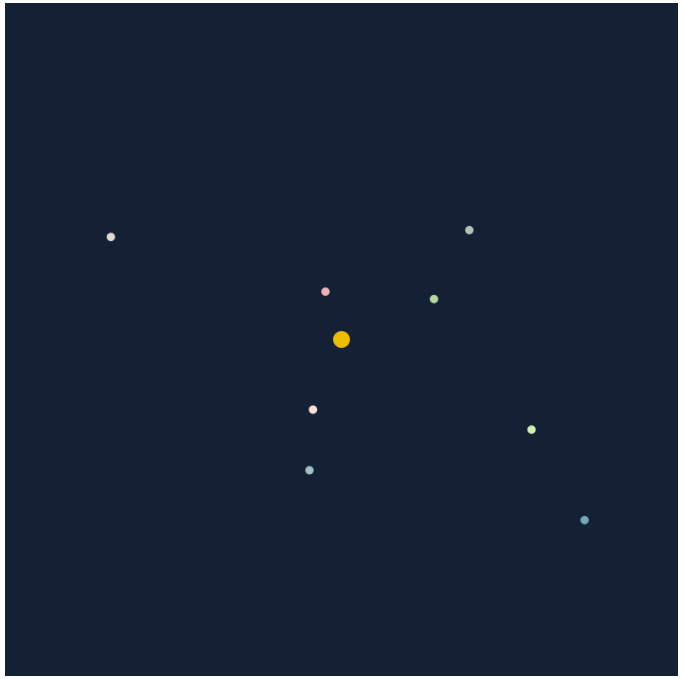
Homework Examples

Bouncing Hello World

Paddle Ball Game

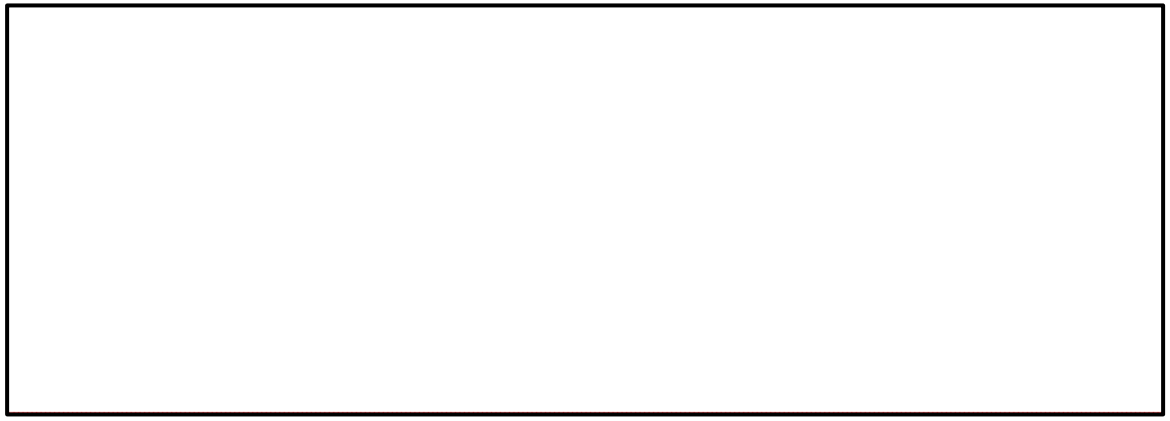


Solar System

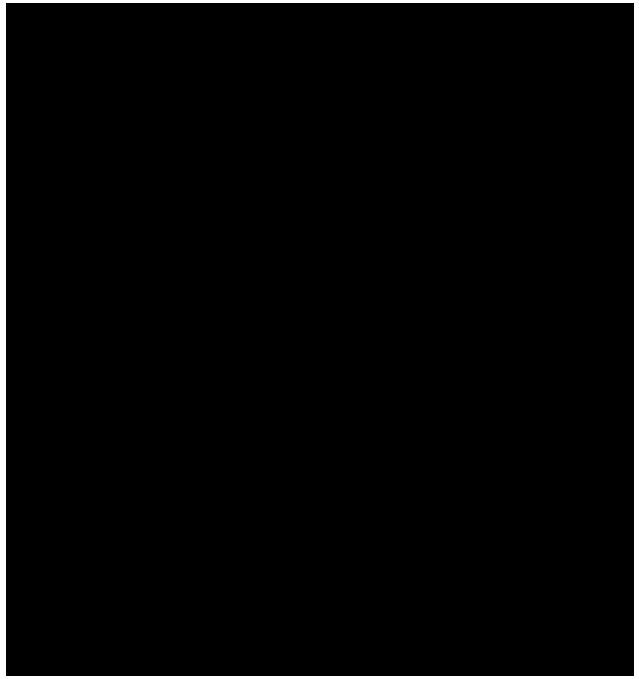


Homework Examples

Sonic Visualizer



Singing Ball



Assignment Policies

- All assignments must be **completed on your own**
  - You are welcome to exchange ideas with your peers, but this should be in the form of concepts and discussion, not in the form of writing and code.
- **Please turn in your work even if it's incomplete for partial credits!**
- Due at **11:59pm ET** on the date specified
- Late submissions: **up to a week**, **1 point deducted per day**

Project

- **Open-ended, individual project**
- Deliverables due at **11:59pm ET** on the date specified
- **No late submissions!** Submit your work early and update it later.
- Milestones
  - **Presentation** on **Dec 2** (tentative)
  - **Final report** due on **Dec 11** (tentative)

Project Formats

- Composition
- Live performance
- Playable instrument
- Synthesizer
- Audio effect
- Max for Live plugin
- Application
- Generative system
- Audiovisual installation

Grading

- All **grading** and **regrade requests** will be handled on Gradescope
- **Homework** (**50%**)
- **Midterm assignment** (**20%**)
- **Project** (**30%**)

A+	97+	B+	87–89	C+	77–79	D+	67–69	F	<60
A	93–96	B	83–86	C	73–76	D	63–66		
A-	90–92	B-	80–82	C-	70–72	D-	60–62		

Resources

- **Processing**
  - Free software with an LGPL license available at processing.org/download
  - **Documentation**: processing.org/reference
  - Official **tutorials**: processing.org/tutorials
  - Official **examples**: processing.org/examples
- **Max**
  - Licensed software with a **30-day trial** available at cycling74.com/shop
  - **Documentation**: docs.cycling74.com
- Both Processing and Max are available at the **Music Tech Lab**

Optional Reading

- "*Processing: A Programming Handbook for Visual Designers and Artists*" by Casey Raes and Ben Fry
- "*The Nature of Code*" by Daniel Shiffman
- "*Electronic Music and Sound Design: Theory and Practice with Max/MSP*" by Alessandro Cipriani and Maurizio Giri
- "*The Theory and Technique of Electronic Music*" by Miller Puckette

Policies: Attendance

- **On-time, in-person attendance is expected**
- Zoom link can be found on the course website
- Course lectures will be recorded and made available on the course website
- Please **attend in-person** for
  - **Midterm assignment showcase**
  - **Project presentation**

Policies: Generative AI Usage

- Feel free to use GenAI tools in your workflow. However, **you must disclose your usage of GenAI services in your write-ups**.
  - For example, U-M GPT, ChatGPT, Gemini, Claude, GitHub Copilot, DALLE, Midjourney, Stable Diffusion, Firefly, etc.
- **You take full responsibility for AI-generated materials as if you had produced them yourself**: ideas should be attributed and facts should be true.

Policies: Academic Integrity

- Plagiarism and cheating violate SMTD's Academic Code of Conduct. **All plagiarism, cheating and other academic misconduct cases will be reported to SMTD's Office of Academic and Student Affairs.**
- **All assignments must be completed on your own**. You are welcome to exchange ideas with your peers, but this should be in the form of concepts and discussion, not in the form of writing and code.
- You must **provide proper citations/references for any external resources** you use in your writing and code.

Performing Arts Technology (PAT)

Minor in Performing Arts Technology (PAT)

Intro (Required)

PAT 100: Music in Technology
PAT 200: Intro to Electronic Music

PAT Theory/Studies (6 credits)

PAT 150: Experiential Music Theory
PAT 205: Intermedia AI Music Practice
PAT 305: Video Game Music
PAT 315: Diversity in Music Technology
PAT 316: NOISE

PAT Practice (3 credits)

PAT 202: Computer Music
PAT 204: Creative Coding
PAT 220: Songwriting Workshop
PAT 280: Sound Reinforcement
PAT 412: Digital Music Ensemble
PAT 413: Electronic Chamber Music

Electives (6 credits)

PAT 421: Advanced Psychoacoustics
PAT 422: Technical Ear Training & Critical Listening
PAT 424: Dialog of the Senses
PAT 431/432: Contemporary Practice in Studio Production I/II
PAT 441: Sound for Film and Games
PAT 443: Immersive Media
PAT 451/452: Interactive Media Design I/II
PAT 454: Digital Fabrication for Acoustics
PAT 461: Performance Systems
PAT 462: Digital Sound Synthesis
PAT 463: Music & AI
PAT 464: Generative AI for Music & Audio Creation
PAT 472: Business of Music

PAT PERFORMING ARTS TECHNOLOGY
UNIVERSITY OF MICHIGAN

🤔 Questions on the Syllabus?

hermandong.com/teaching/pat204

Let’s Get Started with Processing!

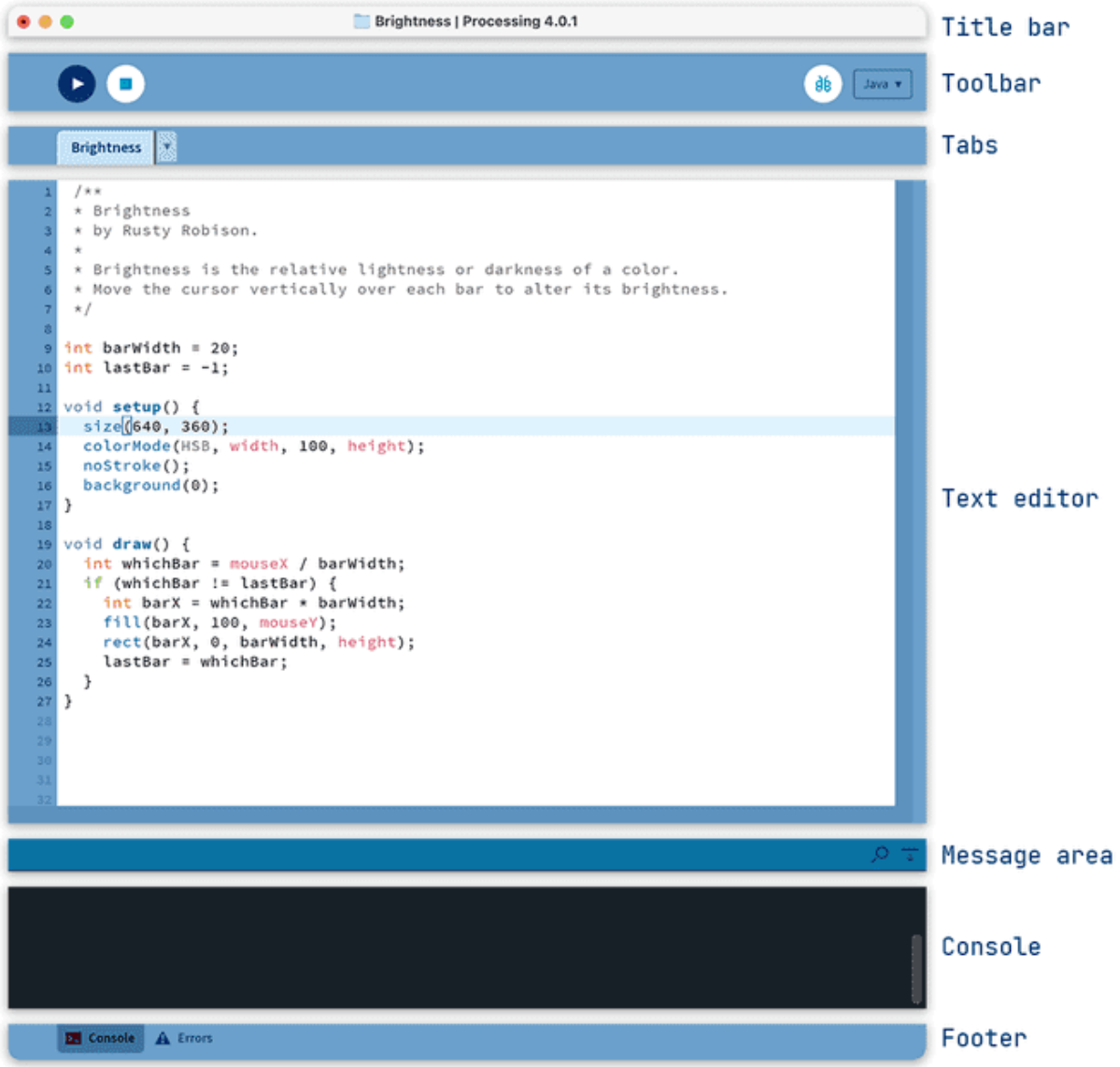
Accessing Processing

- Processing can be downloaded at **processing.org/download**
- Available at the **Music Tech Lab**

A Processing Sketch

- Processing comes with an IDE
  (Integrated Development Environment)
  - A **text editor**
  - A **console**
  - A **display window** (when you click the *run* button)

Display window



The Canvas

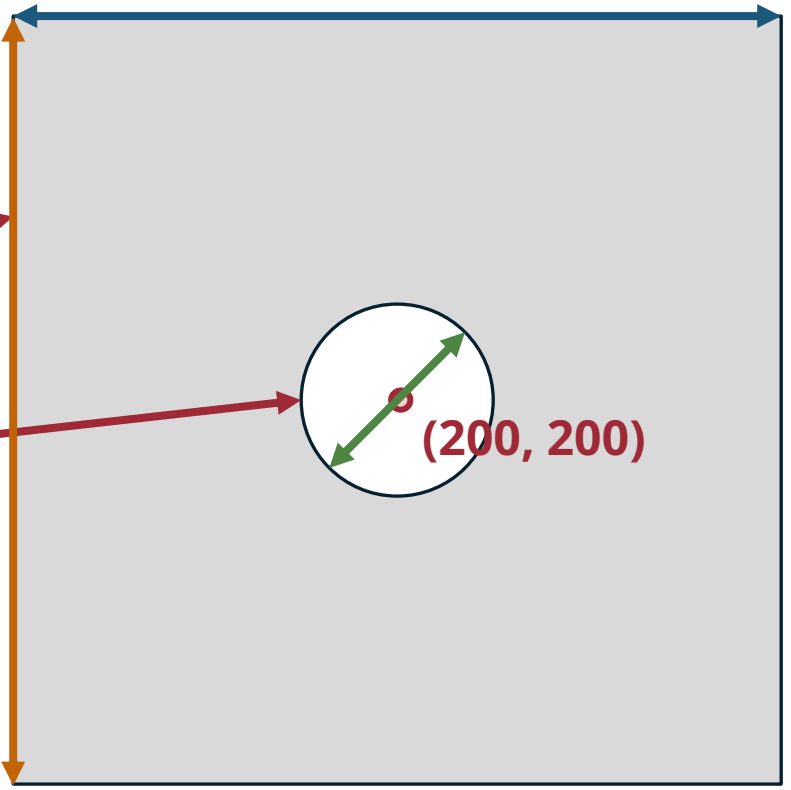
Coordinate System

Our First Processing Sketch

```
size(400, 400);
circle(200, 200, 100);
```

width height

x y diameter



Our First Processing Sketch

Parentheses

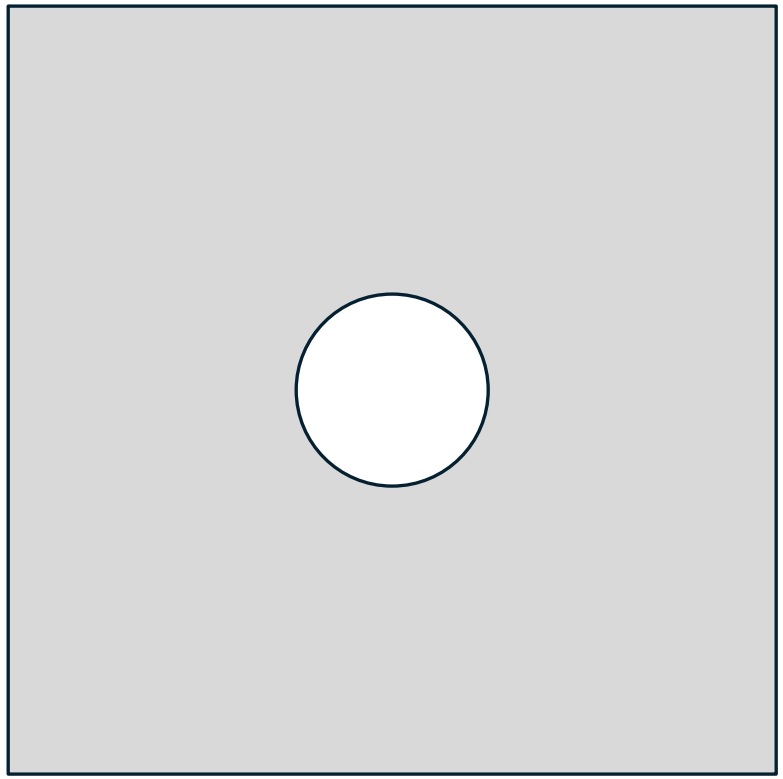
Function name

Arguments (parameters)

```
size(400, 400);
```

Semicolon!

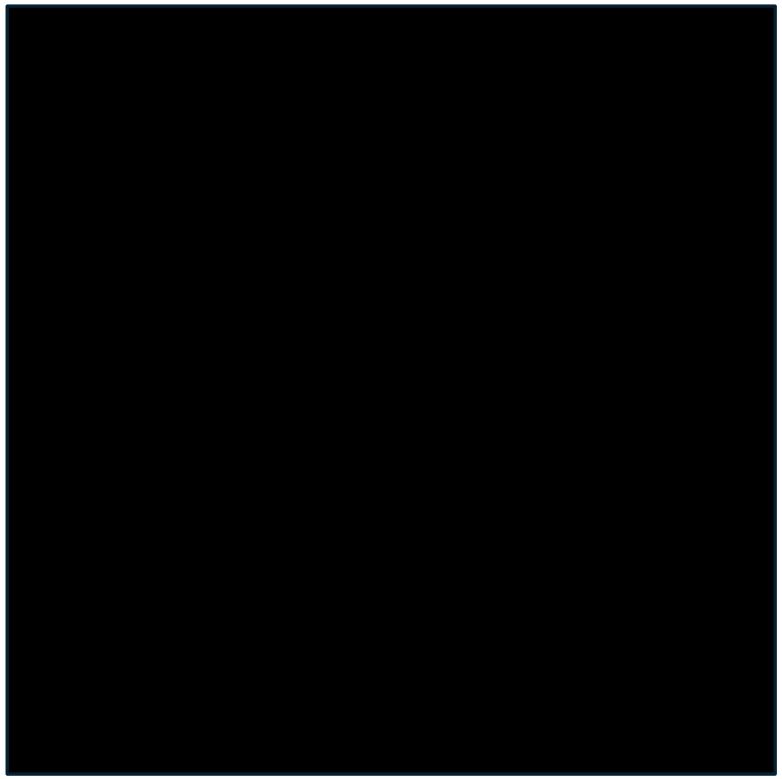
```
circle(200, 200, 100);
```



Colors

Background Color

```
size(400, 400);
background(0);
```

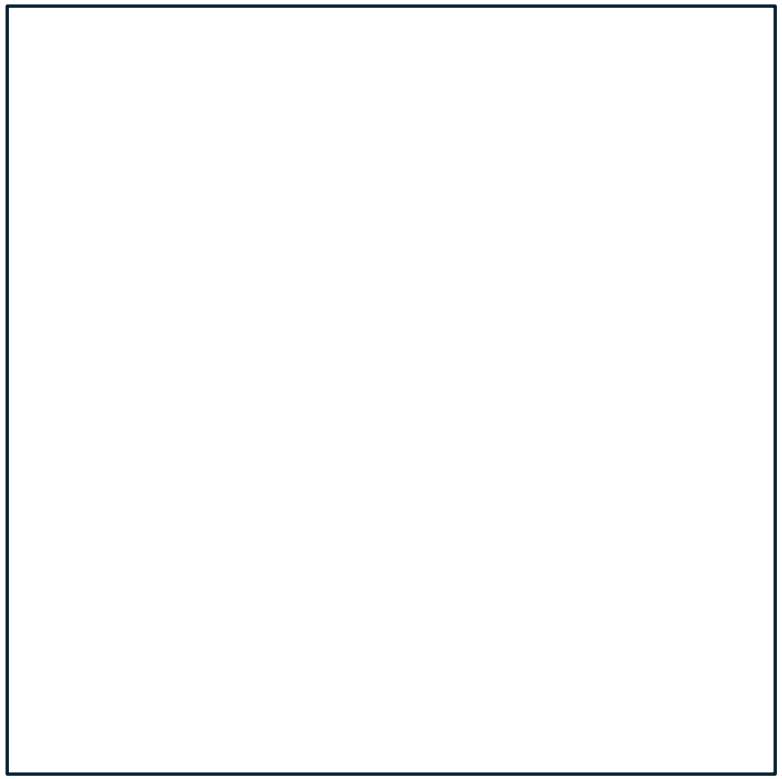


Background Color

```
size(400, 400);
background(255);
```

Range: **0–255**

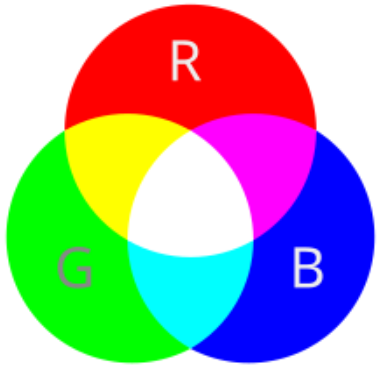
Why?



Background Color

```
size(400, 400);
background(0, 39, 76);
```

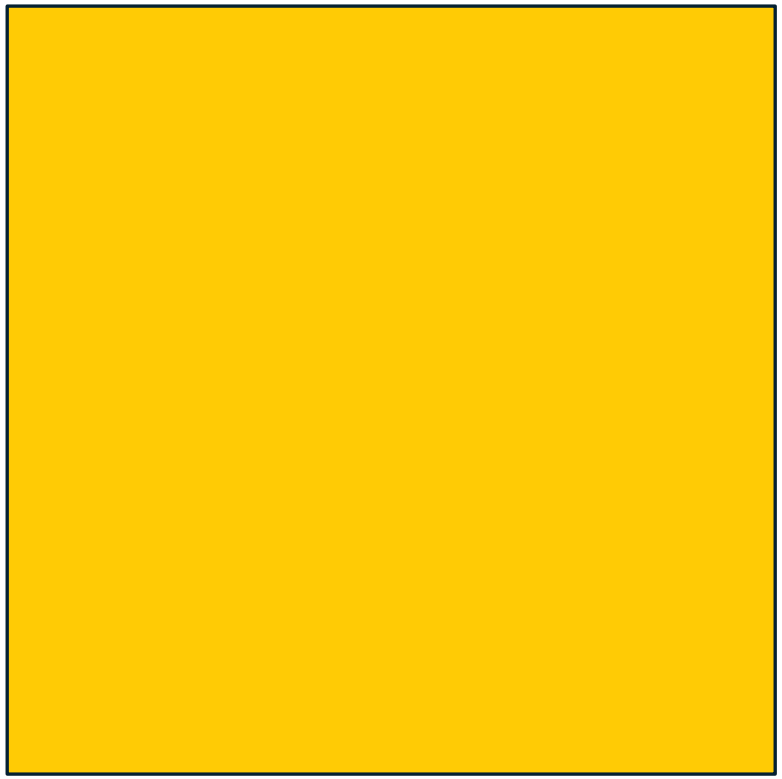
R G B



Background Color

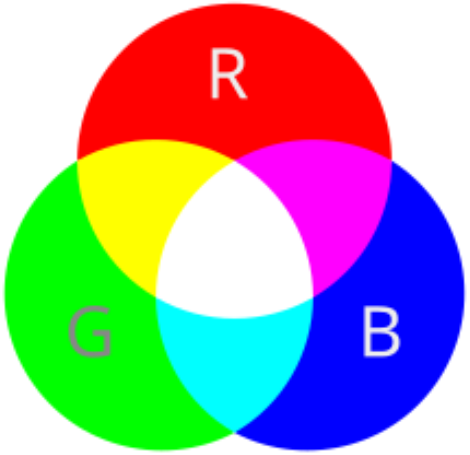
```
size(400, 400);
background(#FFCB05);
```

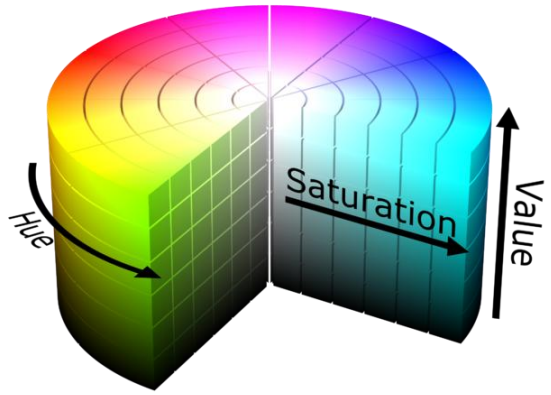
R G B



Colors in Processing

- **Grayscale**: `0–255`
- **RGB color**: `(255, 203, 5)`
- **Hex code**: `#00274C`
- **HSB color**
  - **H**: Hue
  - **S**: Saturation
  - **B**: Brightness
  - Enable with `colorMode(HSB)`





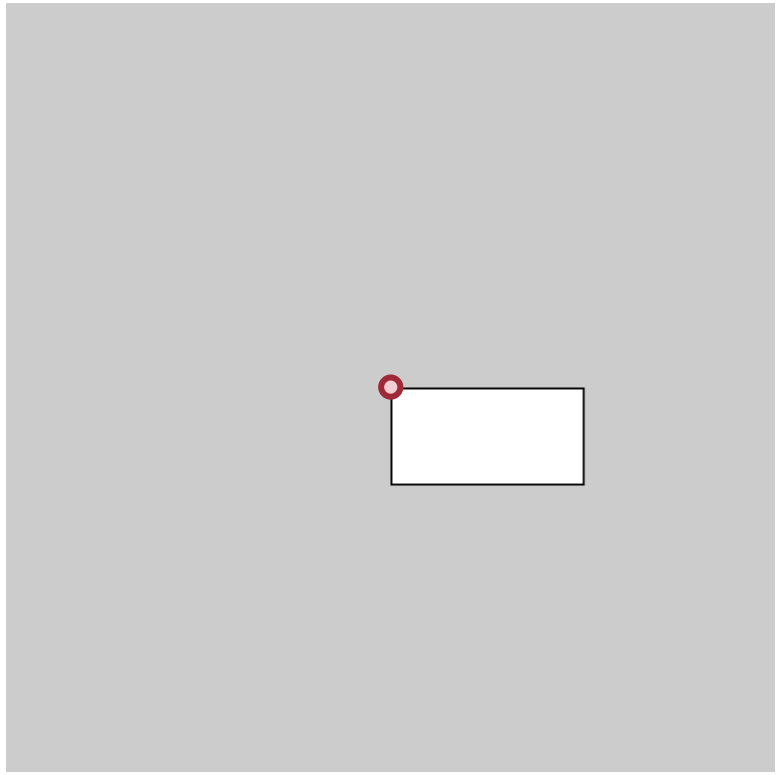
(Source: SharkD)

Shapes

Creating a Rectangle

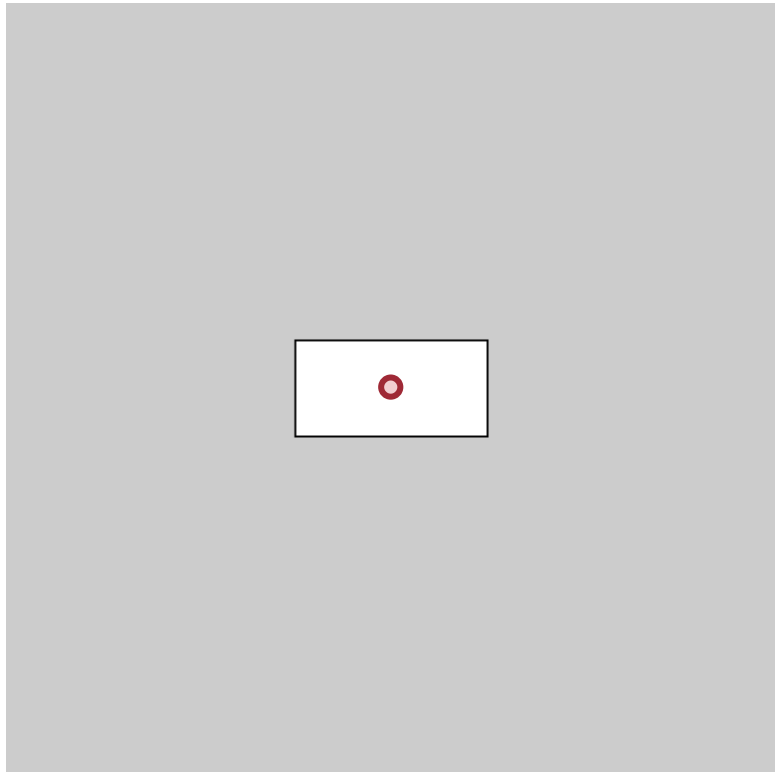
```
size(400, 400);
rect(200, 200, 100, 50);
```

x y width height



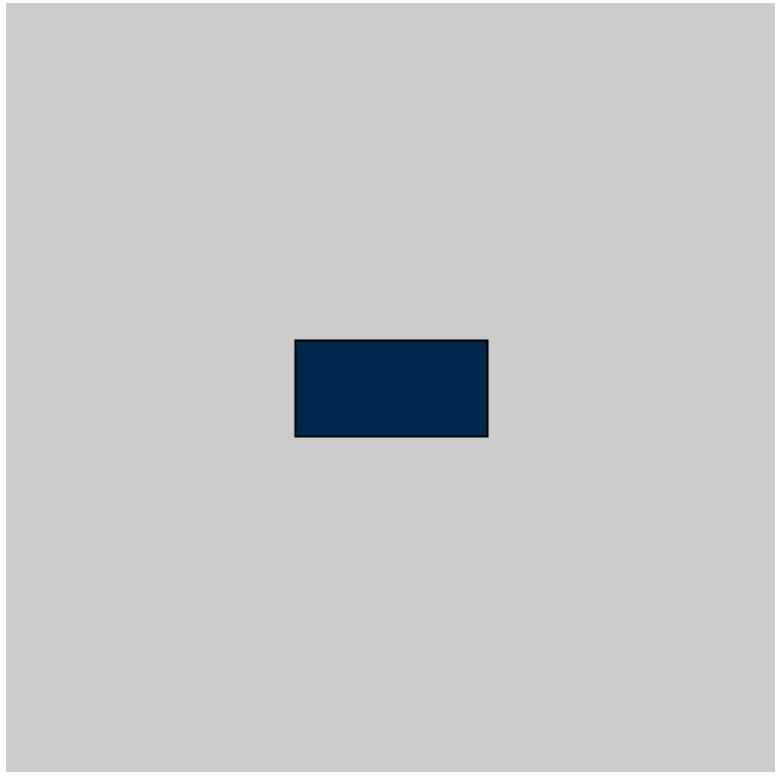
Setting the Anchor Points of Rectangles

```
size(400, 400);
rectMode(CENTER);
rect(200, 200, 100, 50);
```



Changing the Colors of Shapes

```
size(400, 400);
rectMode(CENTER);
fill(#00274C);
rect(200, 200, 100, 50);
```



More Shapes

- Circle `circle(x, y, diameter)`
- Ellipse `ellipse(x, y, width, height)`
- Square `square(x, y, width)`
- Rectangle `rect(x, y, width, height)`
- Point `point(x, y)`
- Line `line(x1, y1, x2, y2)`
- Triangle `triangle(x1, y1, x2, y2, x3, y3)`
- Quadrilateral `quad(x1, y1, x2, y2, x3, y3, x4, y4)`

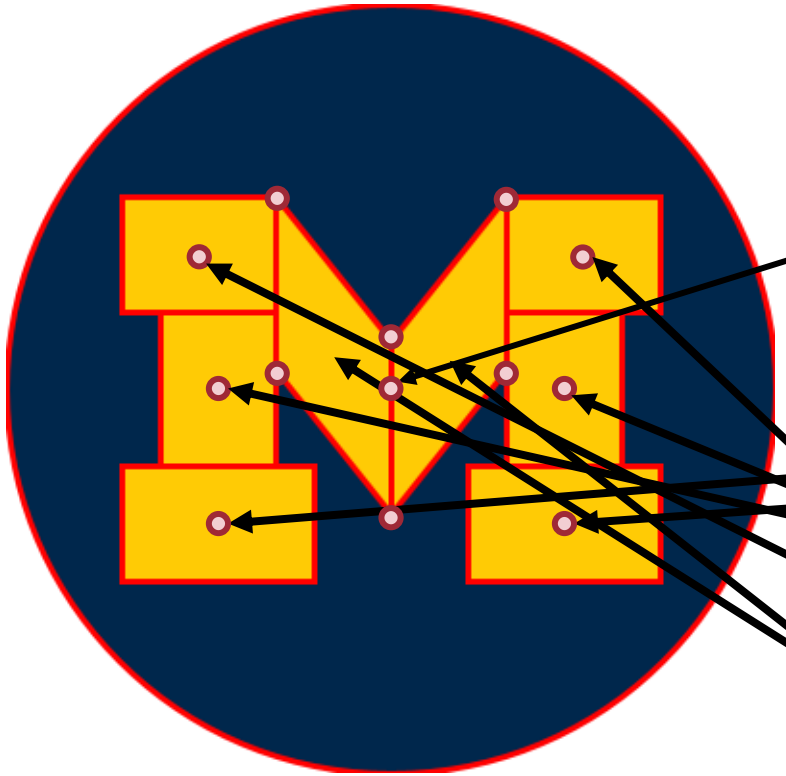
Essential Shortcuts

- **Cmd+R**: Run
- **Esc**: Stop
- **Cmd+Shift+F**: Search in the online reference
- **Cmd+/**: Comment/uncomment
- **Cmd+T**: Auto format

🧐 **Exercise**: Block M

- Michigan **Blue**: **#00274C**
- Michigan **Maize**: **#FFCB05**
- Functions you'll find useful:
  - `size(width, height)`
  - `background(color)`
  - `noStroke()`
  - `fill(color)`
  - `circle(x, y, diameter)`
  - `rect(x1, y1, x2, y2, x3, y3, x4, y4)`
  - `rectMode(mode)` → `rectMode(CENTER)`

🧐 Exercise: Block M



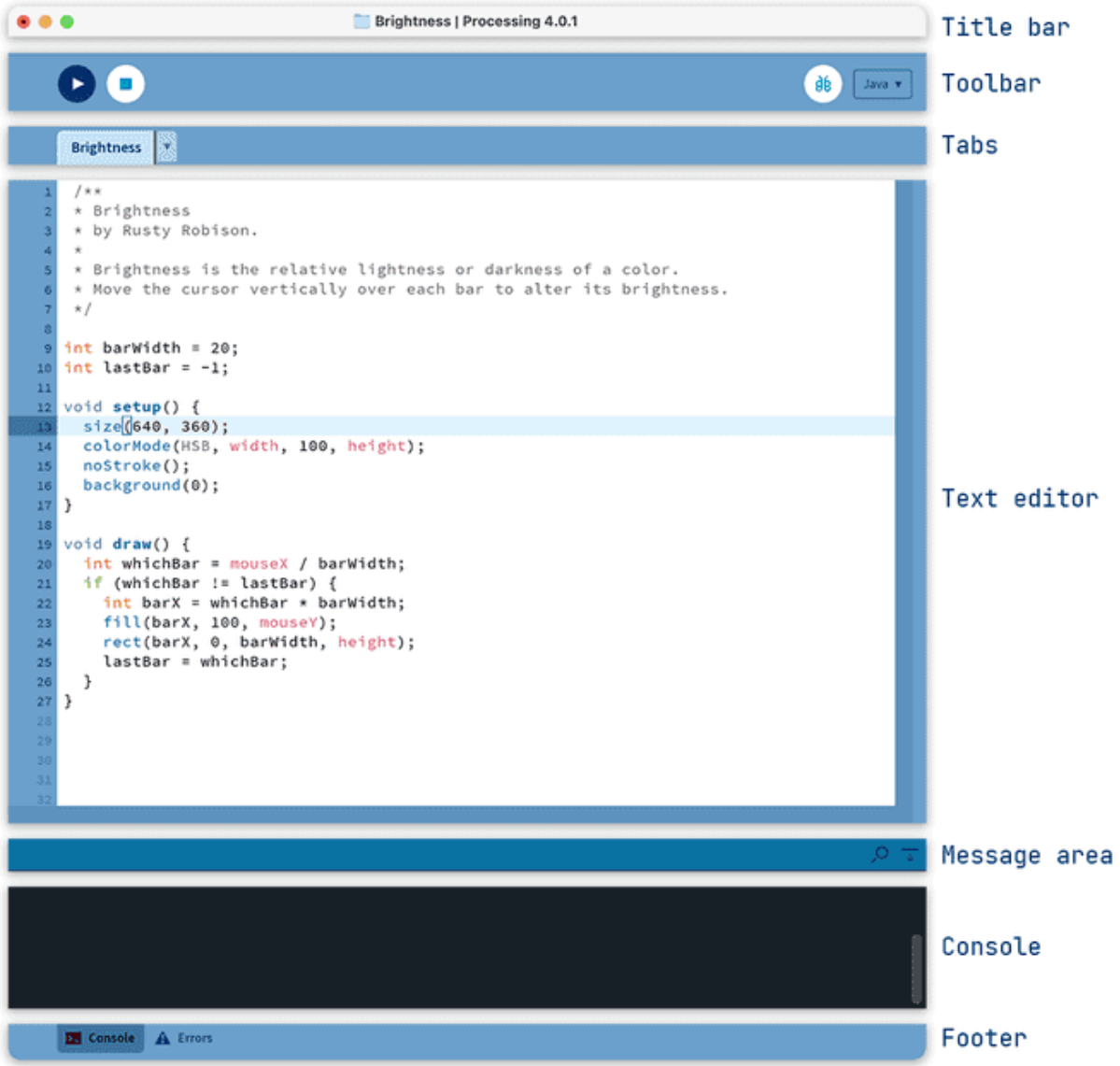
```
void setup() {
  // Create a 400x400 canvas
  size(400, 400);
}

void draw() {
  // Set the background color to white
  background(255);

  // Draw the shapes without outlines
  noStroke();

  // Draw the blue circle at the back
  fill(#00274C);
  circle(200, 200, 400);

  // Set the anchor point of rectangles to the center
  rectMode(CENTER);

  // Set up the yellow text color
  fill(#FFCB05);

  // Draw the feet
  rect(110, 270, 100, 60);
  rect(290, 270, 100, 60);

  // Draw the columns
  rect(110, 210, 60, 150);
  rect(290, 210, 60, 150);

  // Draw the caps
  rect(100, 130, 80, 60);
  rect(300, 130, 80, 60);

  // Draw the "V"
  quad(140, 100, 140, 190, 200, 265, 200, 175);
  quad(260, 100, 260, 190, 200, 265, 200, 175);
}
```

Recap

A Processing Sketch

- Processing comes with an IDE
  (Integrated Development Environment)
  - A **text editor**
  - A **console**
  - A **display window** (when you click the *run* button)

Display window

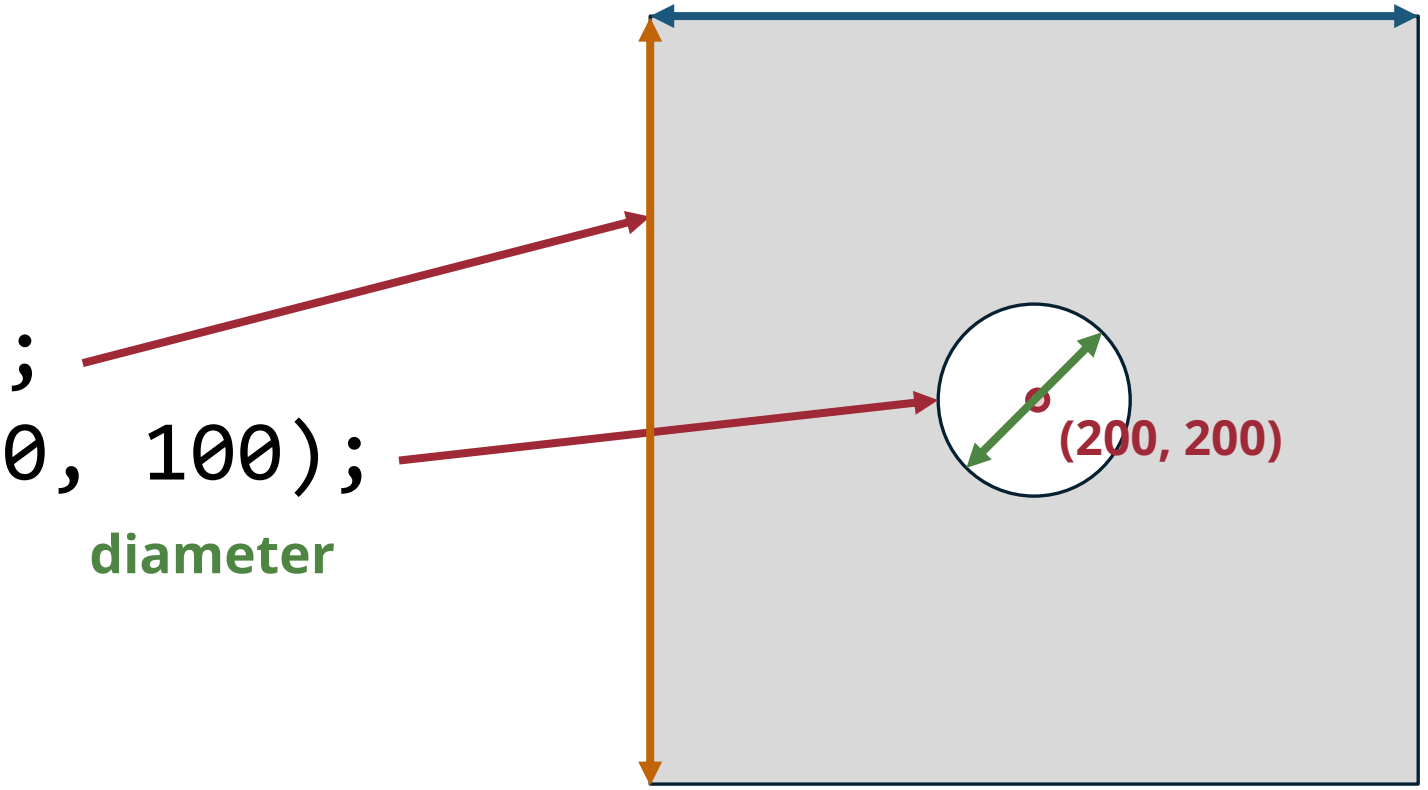


Our First Processing Sketch

```
size(400, 400);
circle(200, 200, 100);
```

width height

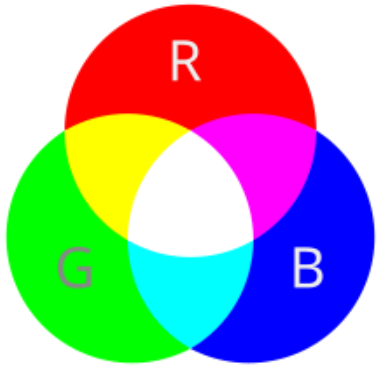
x y diameter



Background Color

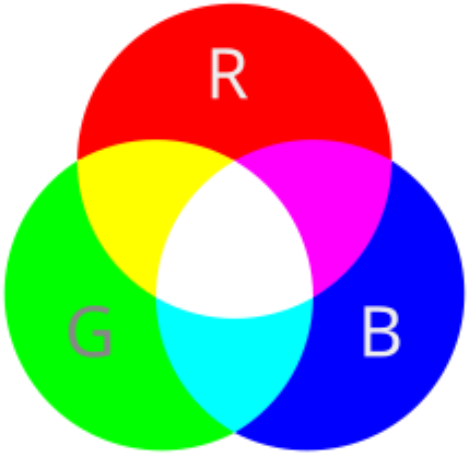
```
size(400, 400);
background(0, 39, 76);
```

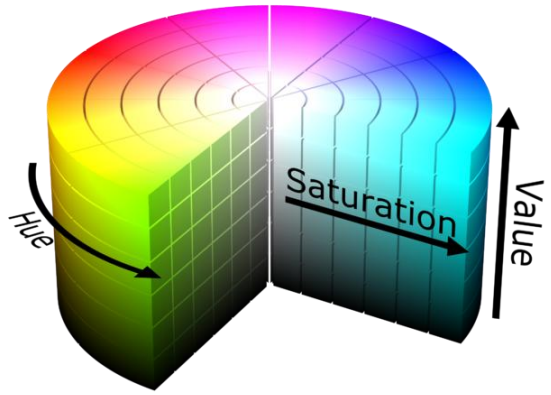
R G B



Colors in Processing

- **Grayscale**: `0–255`
- **RGB color**: `(255, 203, 5)`
- **Hex code**: `#00274C`
- **HSB color**
  - **H**: Hue
  - **S**: Saturation
  - **B**: Brightness
  - Enable with `colorMode(HSB)`

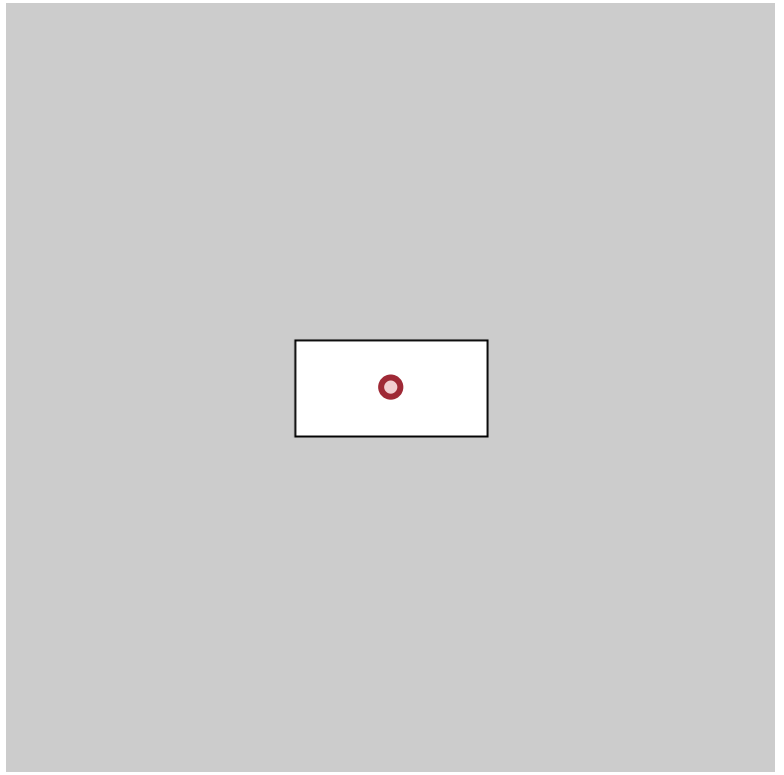




(Source: SharkD)

Setting the Anchor Points of Rectangles

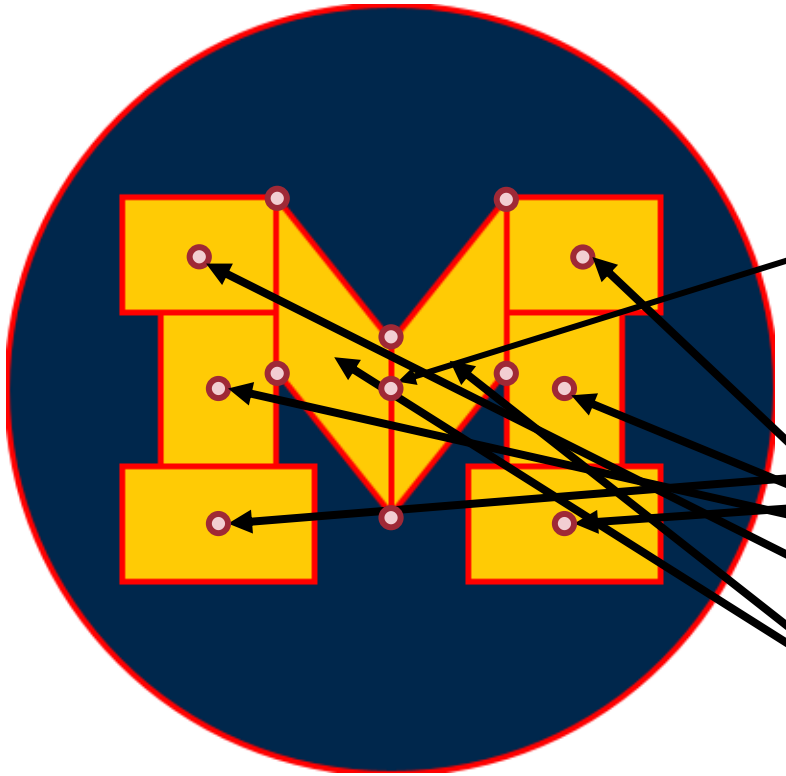
```
size(400, 400);
rectMode(CENTER);
rect(200, 200, 100, 50);
```



More Shapes

- Circle `circle(x, y, diameter)`
- Ellipse `ellipse(x, y, width, height)`
- Square `square(x, y, width)`
- Rectangle `rect(x, y, width, height)`
- Point `point(x, y)`
- Line `line(x1, y1, x2, y2)`
- Triangle `triangle(x1, y1, x2, y2, x3, y3)`
- Quadrilateral `quad(x1, y1, x2, y2, x3, y3, x4, y4)`

🧐 Exercise: Block M



```
void setup() {
  // Create a 400x400 canvas
  size(400, 400);
}

void draw() {
  // Set the background color to white
  background(255);

  // Draw the shapes without outlines
  noStroke();

  // Draw the blue circle at the back
  fill(#00274C);
  circle(200, 200, 400);

  // Set the anchor point of rectangles to the center
  rectMode(CENTER);

  // Set up the yellow text color
  fill(#FFCB05);

  // Draw the feet
  rect(110, 270, 100, 60);
  rect(290, 270, 100, 60);

  // Draw the columns
  rect(110, 210, 60, 150);
  rect(290, 210, 60, 150);

  // Draw the caps
  rect(100, 130, 80, 60);
  rect(300, 130, 80, 60);

  // Draw the "V"
  quad(140, 100, 140, 190, 200, 265, 200, 175);
  quad(260, 100, 260, 190, 200, 265, 200, 175);
}
```

Next Lecture

Processing Basics