PAT 204/504 (Fall 2024)

Creative Coding

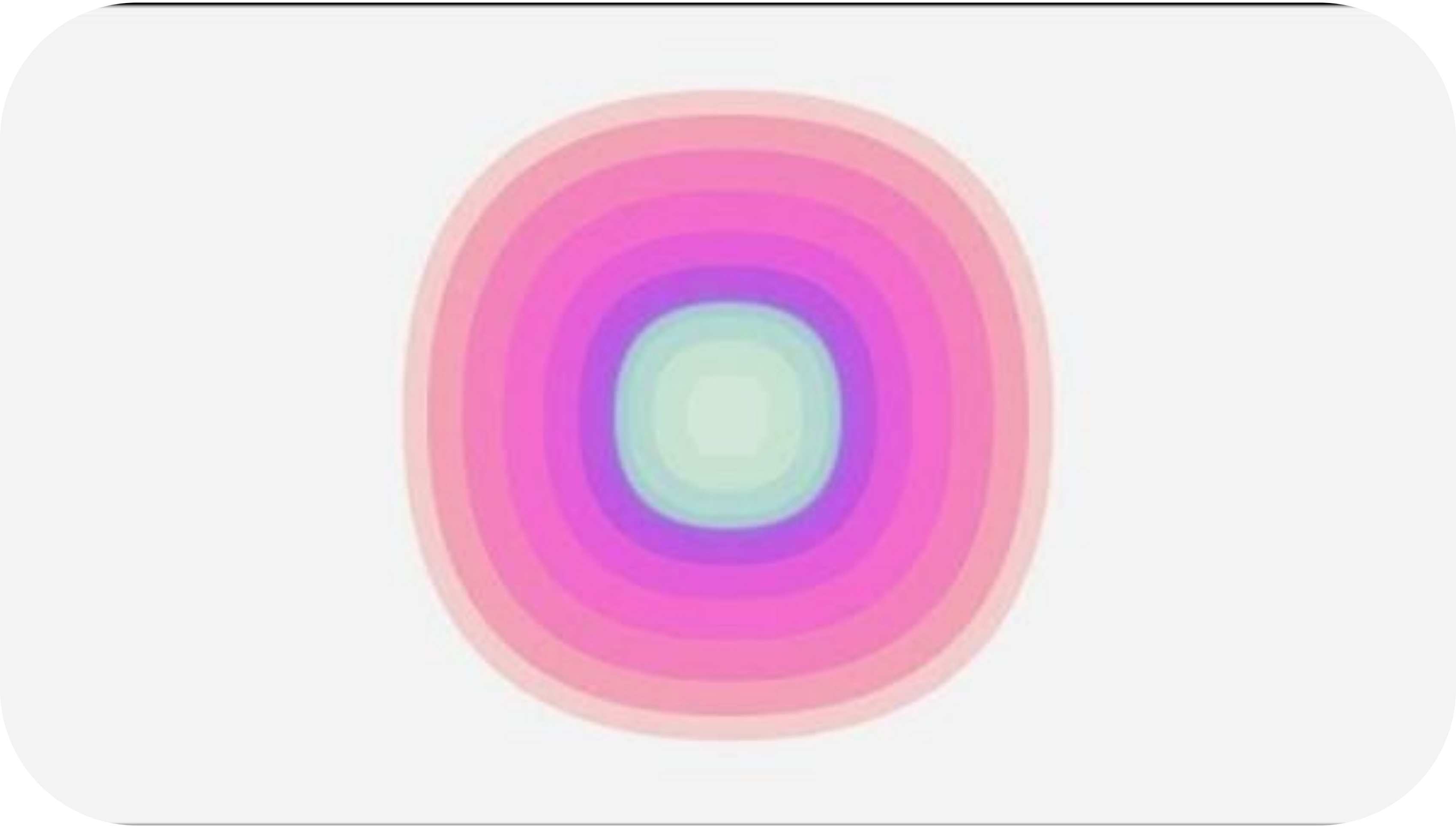
Lecture 10: Motion & Physics

Instructor: Hao-Wen Dong

Midterm Assignment: Build Your Own Music Visualizer

- **Open-ended** assignment
- Use everything you’ve learned from the class (and beyond!)
- Instructions will be released on Gradescope
- Due at **11:59pm ET** on **October 7**
- Late submissions: **NOT Accepted** **(Submit early and update later!)**

Example

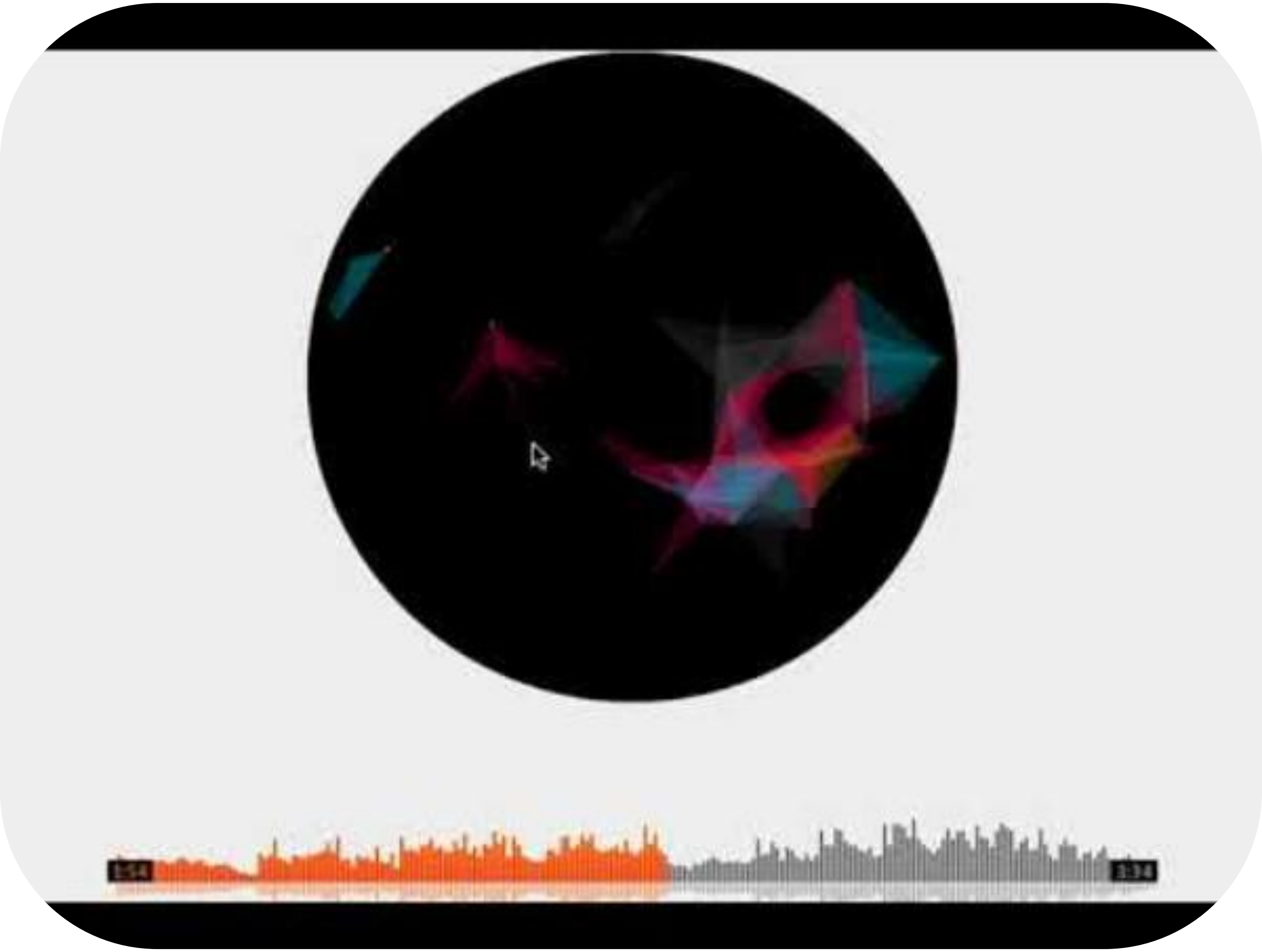


youtu.be/5-ttqEsf518

Example

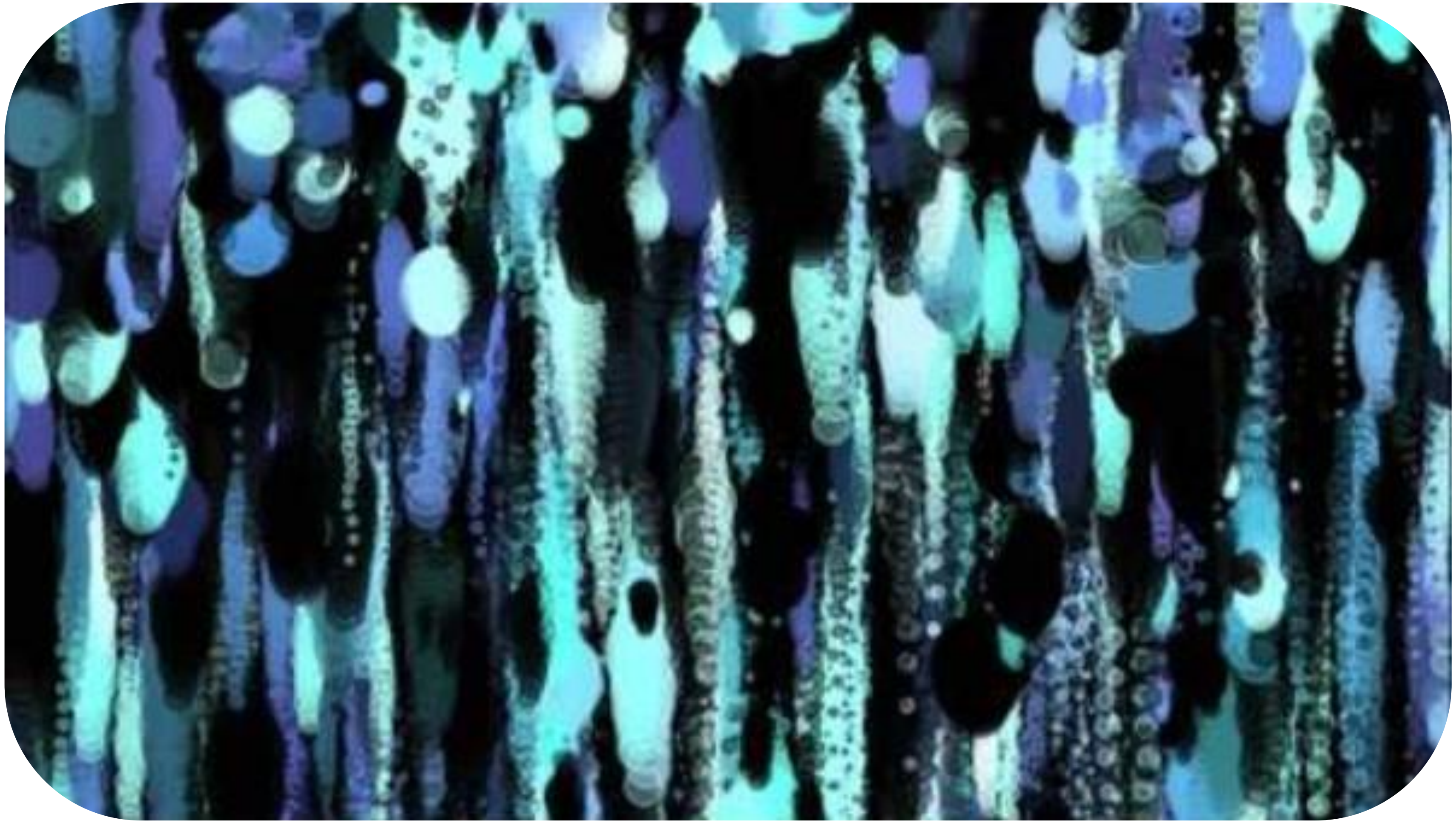
youtu.be/MoM6AxN7TK0

Example



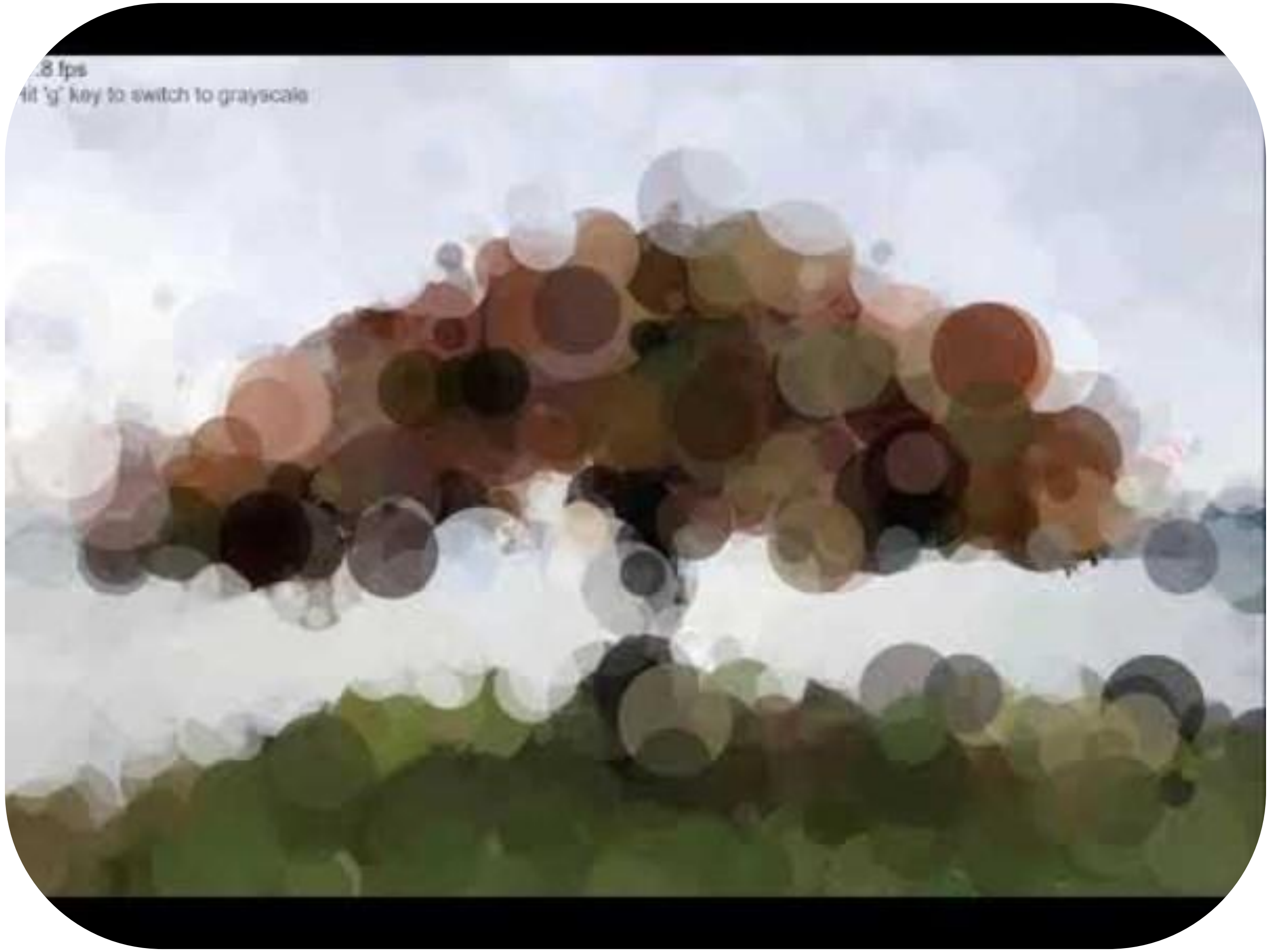
youtu.be/00kQX4m28IU

Example



youtu.be/LMBx6AYaRXc

Example

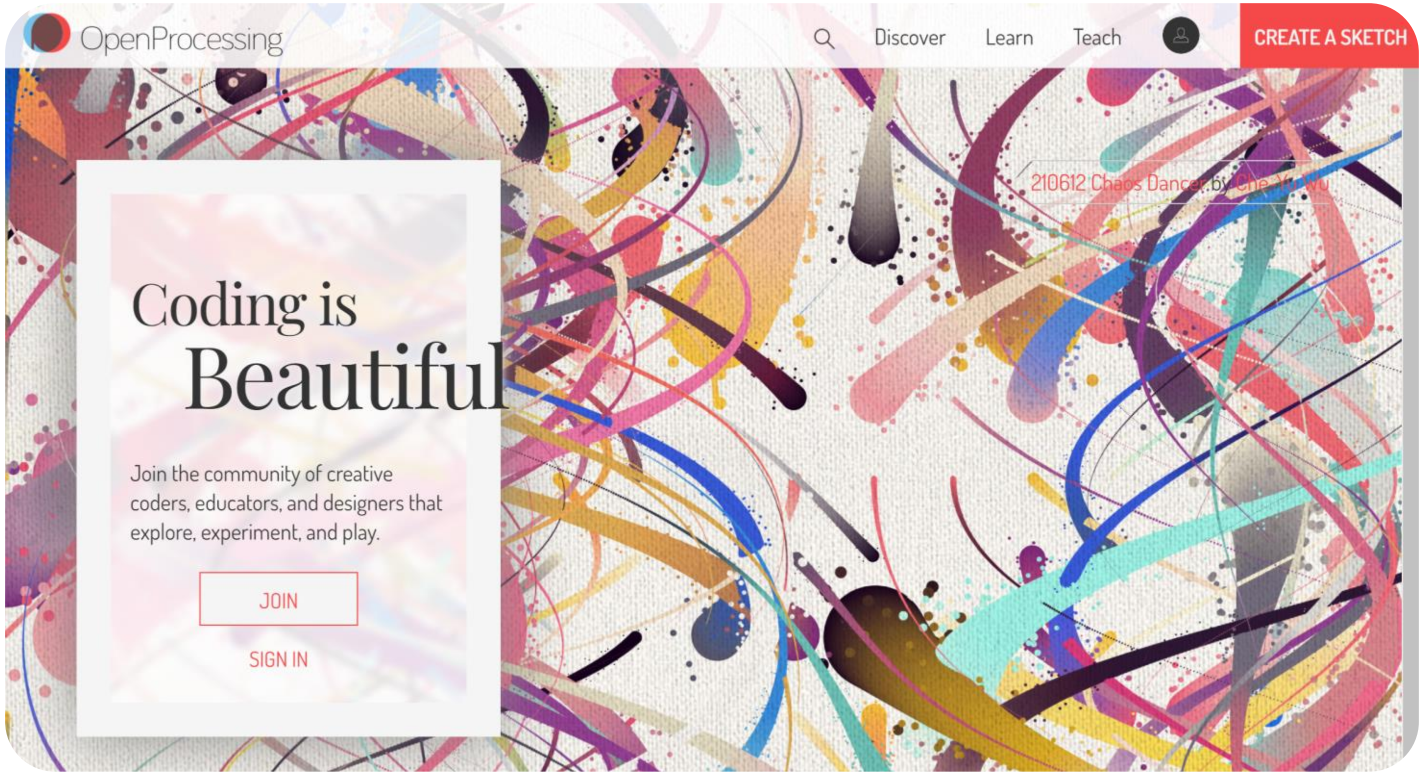


.8 fps
Hit 'g' key to switch to grayscale

youtu.be/UvoABmr4Fdo

OpenProcessing

- Large community of creative coders, educators and designers!

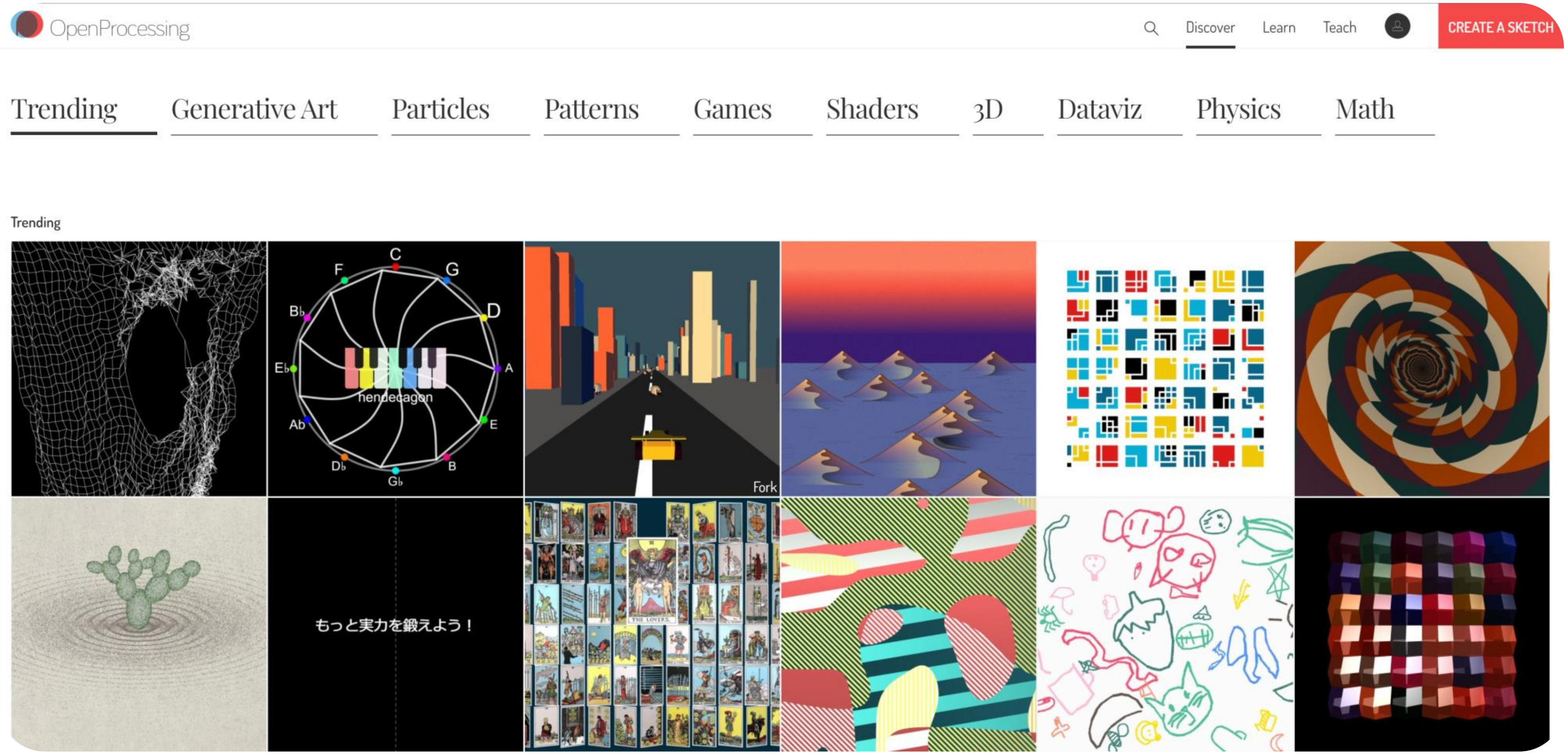


Coding is Beautiful

Join the community of creative coders, educators, and designers that explore, experiment, and play.

openprocessing.org

OpenProcessing Gallery



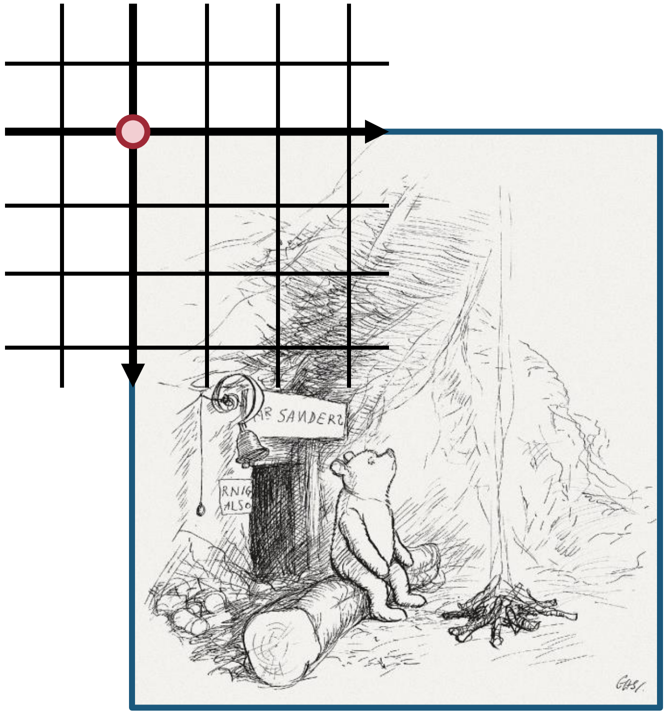
openprocessing.org/discover

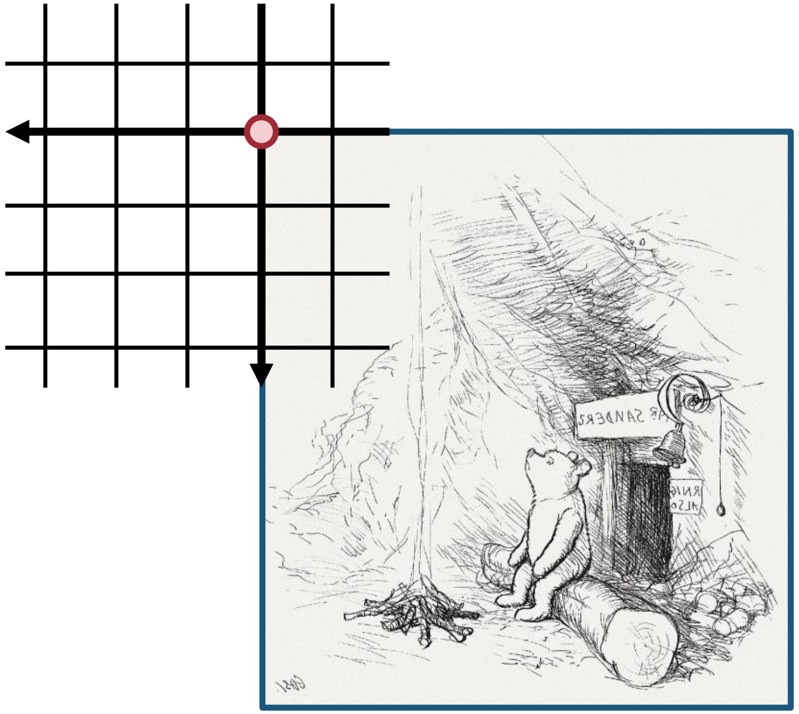
(Recap) Example: Mirroring Capture

```
void draw() {
  image(video, 0, 0);
}
```

→

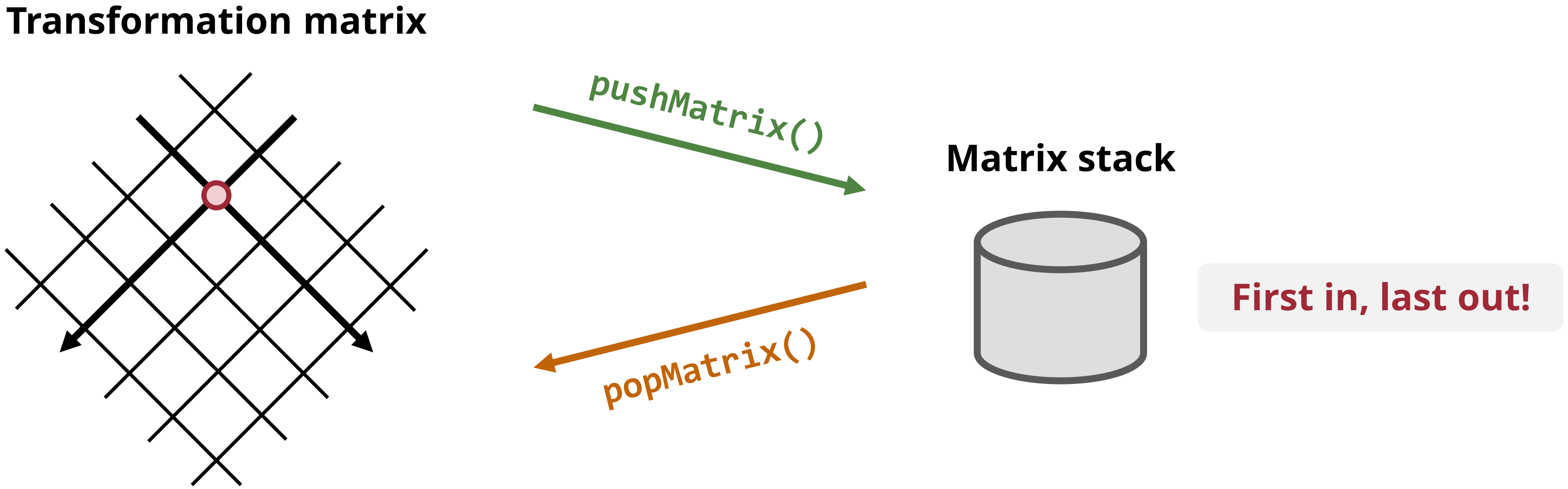
```
void draw() {
  scale(-1, 1);
  image(video, -video.width, 0);
}
```





(Recap) Matrix Transforms

`resetMatrix()`	Reset to identity matrix
`pushMatrix()`	Push the current transformation matrix to the stack
`popMatrix()`	Pop the latest transformation matrix off the stack



First in, last out!

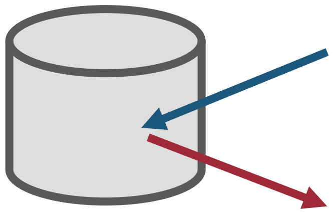
(Recap) Example: Spinning Objects

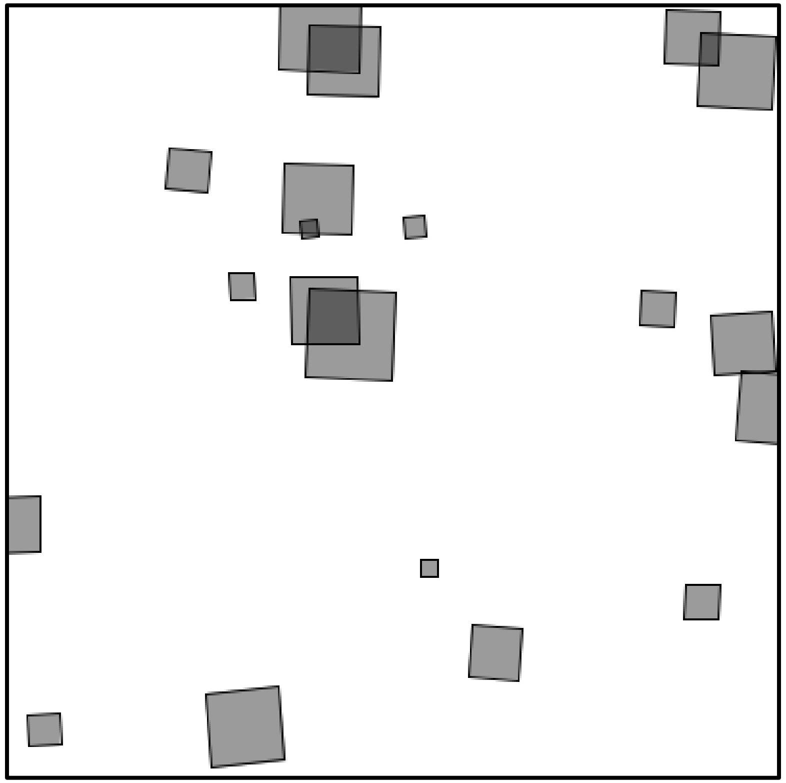
```
class Rotater {
  ...

  void spin() {
    theta += speed;
  }

  void display() {
    rectMode(CENTER);
    stroke(0);
    fill(0, 100);

    pushMatrix();
    translate(x, y);
    rotate(theta);
    rect(0, 0, w, w);
    popMatrix();
  }
}
```

Store the current matrix

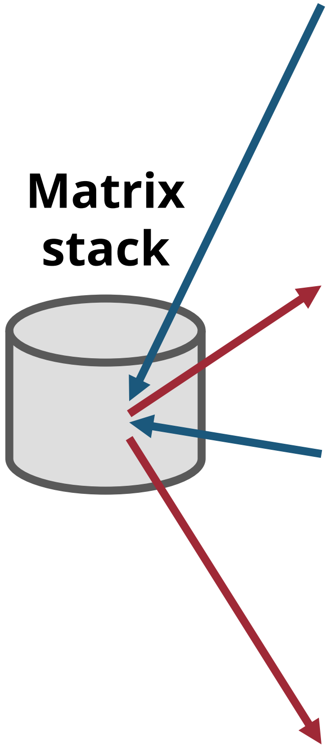
Restore the stored matrix

Matrix stack

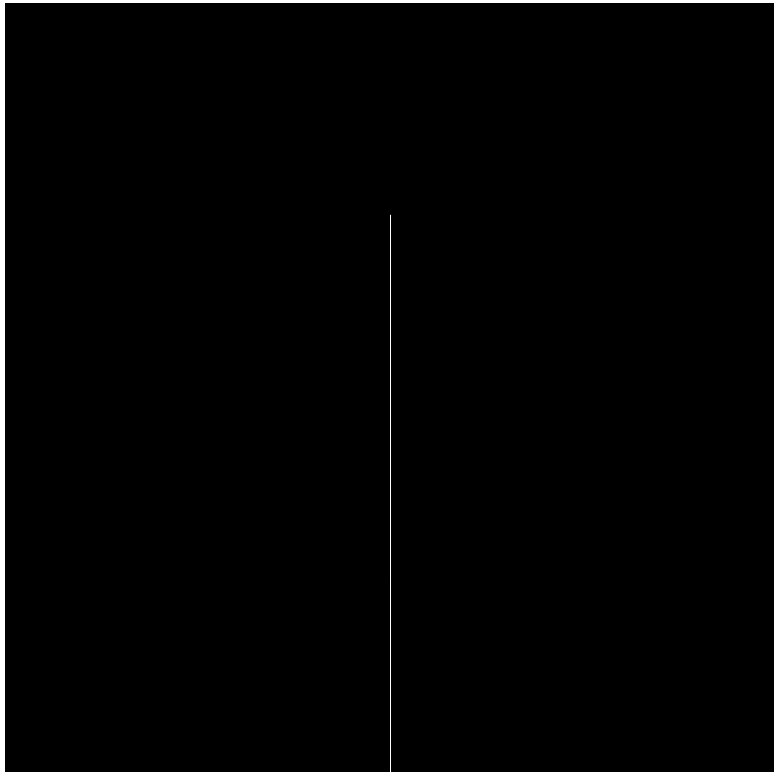




(Recap) Example: Recursive Tree

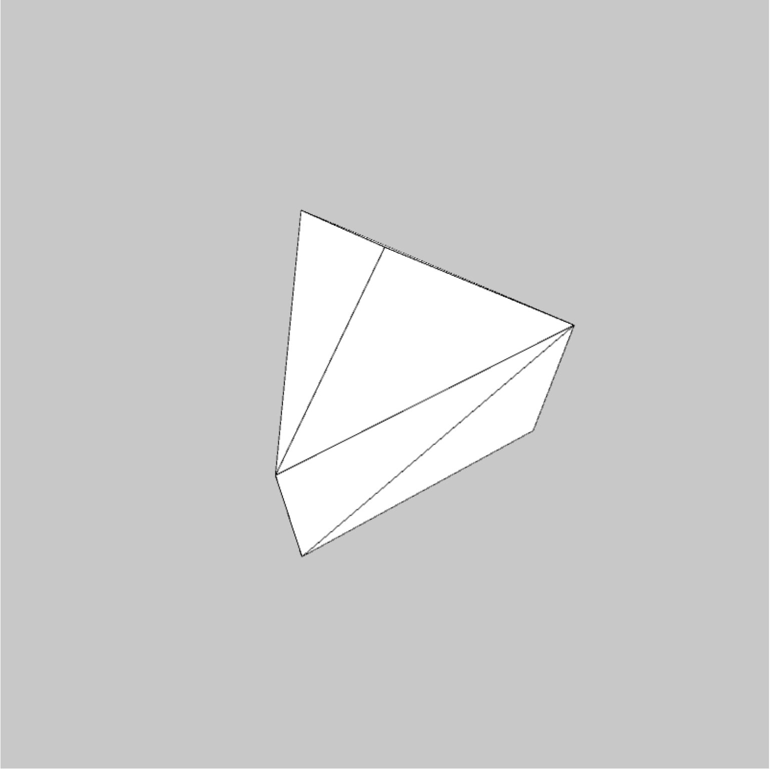


```
void branch(float h) {
  if (h < 2) break;

  // Right branch
 🟩pushMatrix();
 🟪rotate(theta);
  line(0, 0, 0, -h * scale);
  translate(0, -h * scale);
  branch(h * scale);
  popMatrix();

  // Left branch
  pushMatrix();
 🟫rotate(-theta);
  line(0, 0, 0, -h * scale);
  translate(0, -h * scale);
  branch(h * scale);
  popMatrix();
}
```



(Recap) Sphere Details

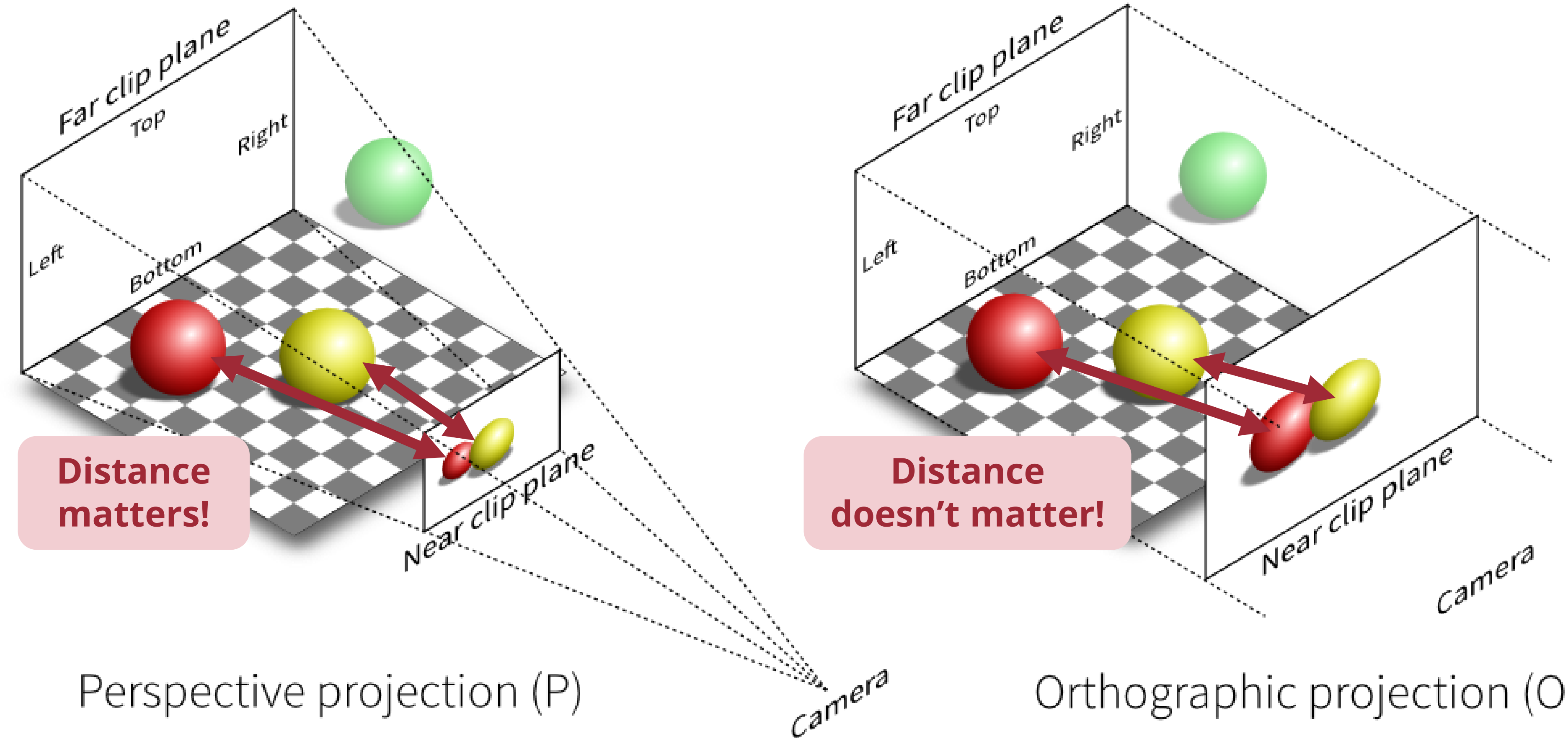
```
int res = 3;

void setup() {
  size(800, 800, P3D);
}

void draw() {
  background(200);
  fill(255);
  stroke(0);

  translate(400, 400, 0);
  rotateX(-1);

  sphereDetail(res);
  sphere(200);

  res += 1;
  if (res > 200) exit();
}
```



(Recap) Perspective vs Orthographic Projections

perspective()

ortho()



Distance matters!

Distance doesn't matter!

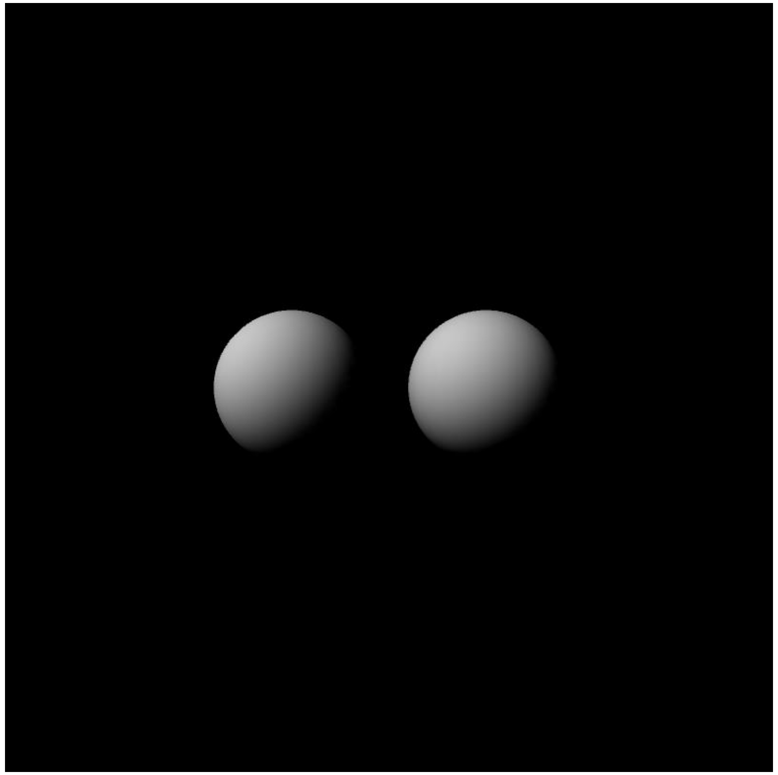
Perspective projection (P)

Orthographic projection (O)

glumpy.github.io/modern-gl.html

(Recap) Example: Creepy Eyes 3D

```
void setup() {
  size(800, 800, P3D);
}

void draw() {
  background(0);

  float dirX = (mouseX - width / 2) / (width / 2.0);
  float dirY = (mouseY - height / 2) / (height / 2.0);
  directionalLight(200, 200, 200, -dirX, -dirY, -1);

  fill(255);
  noStroke();
  translate(300, 400, 0);
  sphere(80);
  translate(200, 0, 0);
  sphere(80);
}
```



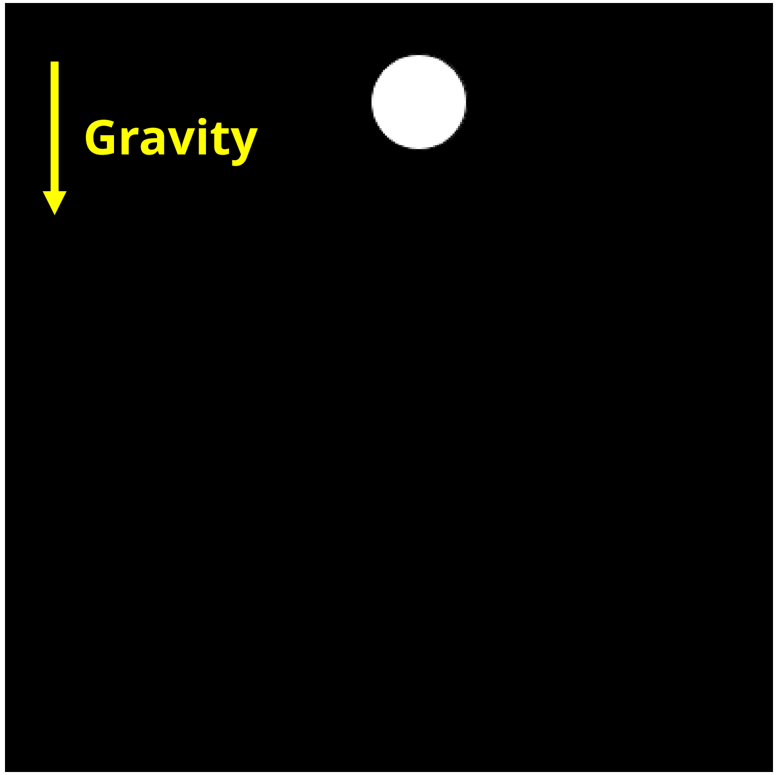
Acceleration

Example: Gravity

```
// Apply gravity to the ball
void applyGravity() {
  speedY += gravity;
}

// Check if the ball hit the walls
void checkWalls() {
  ...

  // Check if the ball hit the top and bottom walls
  if (y > height - radius) {
    speedY = -abs(speedY) * decay;
    y = height - radius;
  } else if (y < radius) {
    speedY = abs(speedY);
    y = radius;
  }
}
```

Apply gravity as y-acceleration

Reduce the speed a little bit when it hits the bottom wall

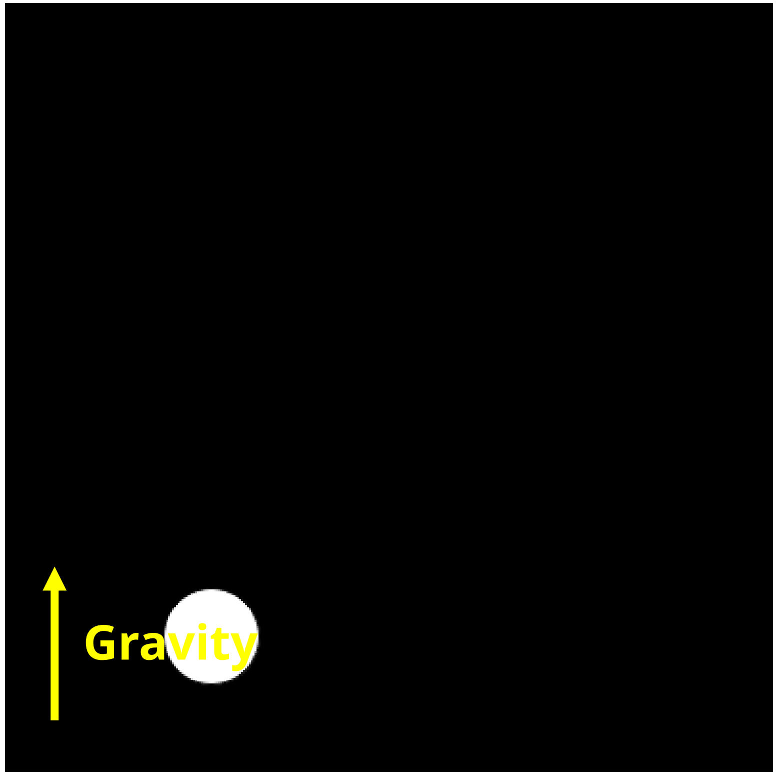


Example: Upside-down Gravity

```
// Apply gravity to the ball
void applyGravity() {
  speedY -= gravity;
}

// Check if the ball hit the walls
void checkWalls() {
  ...

  // Check if the ball hit the top and bottom walls
  if (y > height - radius) {
    speedY = -abs(speedY);
    y = height - radius;
  } else if (y < radius) {
    speedY = abs(speedY) * decay;
    y = radius;
  }
}
```

Apply gravity as y-acceleration

Reduce the speed a little bit when it hits the top wall

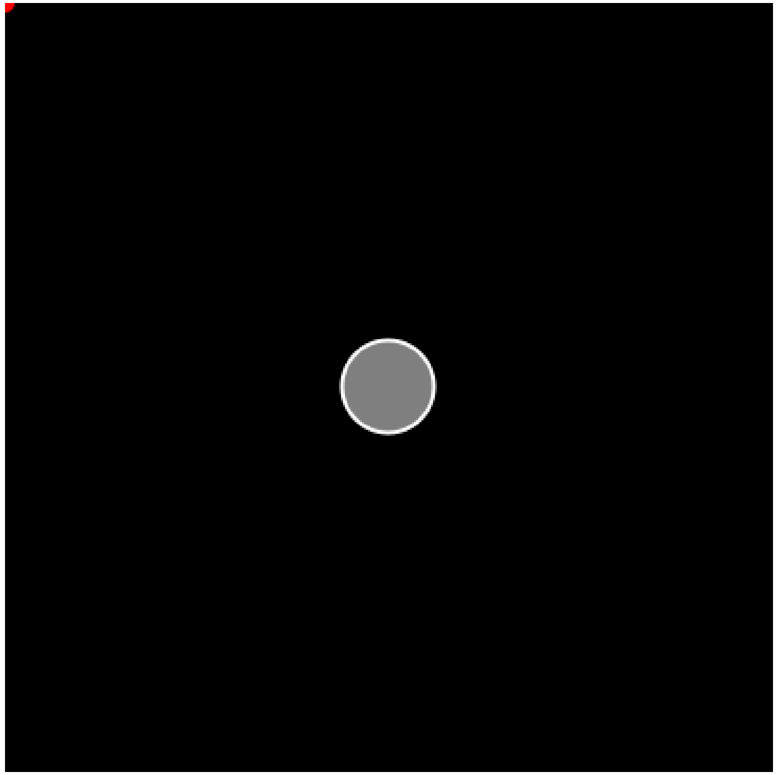


Example: Acceleration

```
class Mover {
  PVector location;
  PVector velocity;
  PVector acceleration;
  float topspeed = 5;

  Mover() {
    location = new PVector(width/2, height/2);
    velocity = new PVector(0, 0);
  }

  void display() {
    stroke(255);
    strokeWeight(2);
    fill(127);
    ellipse(location.x,location.y,48,48);
  }

  ...
}
```

Declare the location, velocity and acceleration vectors

Initialize the ball to be at the center with zero initial speed



processing.org/examples/accelerationwithvectors.html

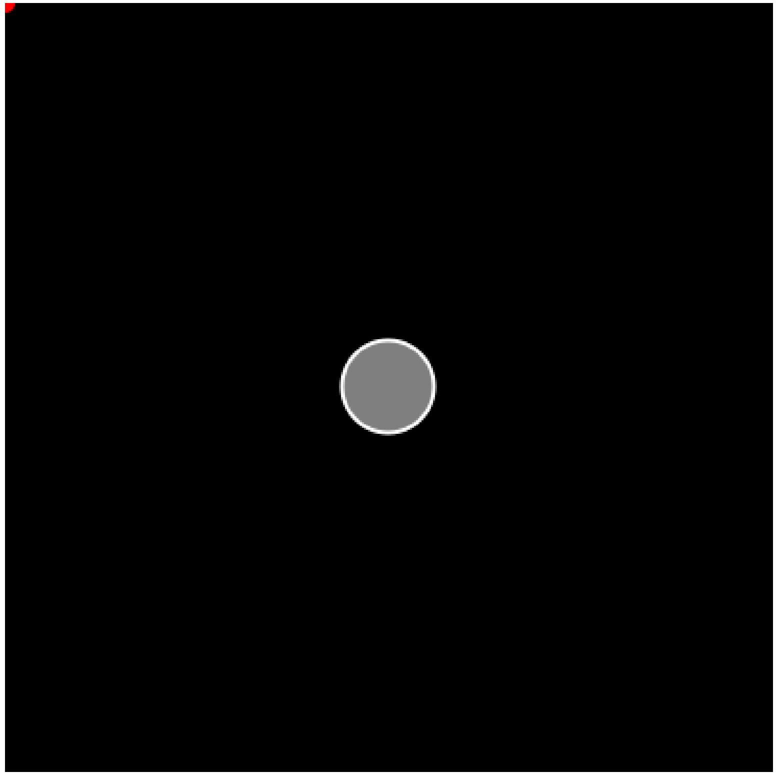
Example: Acceleration

```
class Mover {
  PVector location;
  PVector velocity;
  PVector acceleration;
  float topspeed = 5;

  ...

  void update() {
    PVector mouse = new PVector(mouseX, mouseY);
    PVector acceleration = PVector.sub(mouse, location);
    acceleration.setMag(0.2);

    velocity.add(acceleration);
    velocity.limit(topspeed);

    location.add(velocity);
  }
}
```

Calculate acceleration

Apply the acceleration

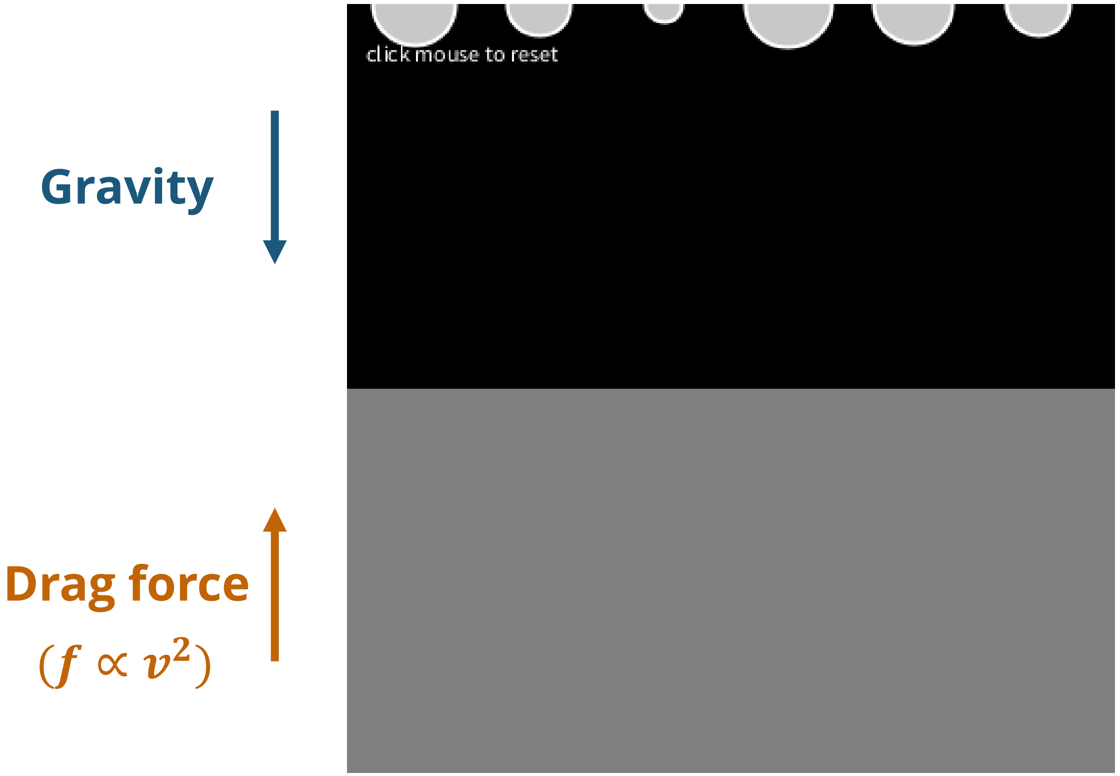
Move the ball



processing.org/examples/accelerationwithvectors.html

Example: Gravity and Fluid Resistance

- Simulating multiple forces
  - Gravity
  - Fluid resistance (drag force)



click mouse to reset

Gravity

Drag force

$(f \propto v^2)$

processing.org/examples/forceswithvectors.html

Example: Purple Rain

- Maintaining an array of **`Drop`** objects
- Each **`Drop`** object
  - Falls with gravity
  - Random initial velocity
  - Random stroke weight (illusion of distance)

Collision

(Recap) Example: Bouncing Balls

```
Ball[] balls = new Ball[20];
```

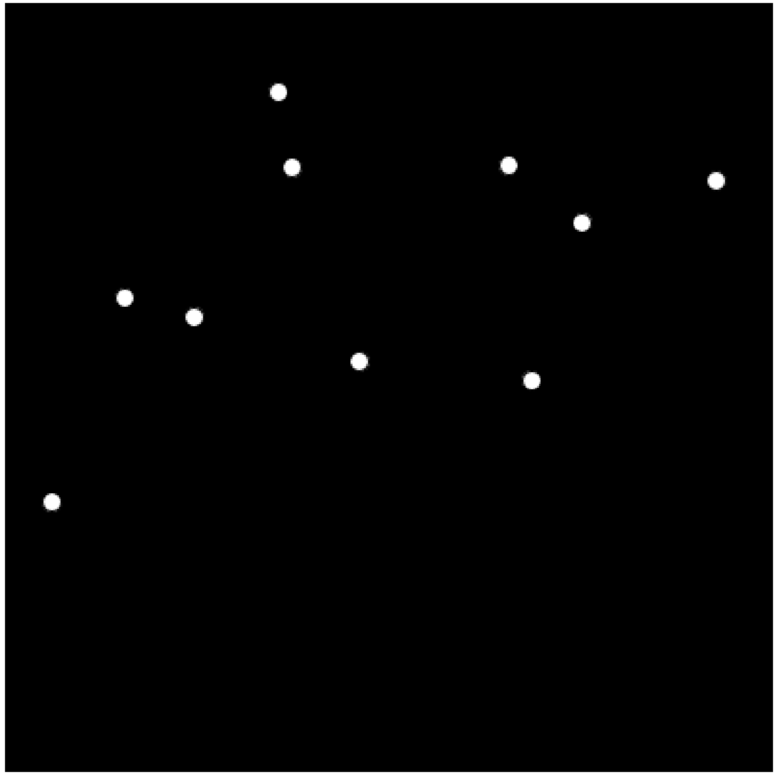
An array of objects

```
void setup() {
  size(400, 400);

  for (int i = 0; i < balls.length; i++) {
    balls[i] = new Ball();
  }
}
```

Initialization

```
void draw() {
  background(0);

  for (int i = 0; i < balls.length; i++) {
    balls[i].move();
    balls[i].checkWalls();
    balls[i].show();
  }
}
```

Call the methods!



Example: Bouncing Balls with Collision Detection

- What else do we need?
- How to check if the ball collides with another?
- What kind of function do we need?

```
Ball[] others;
```

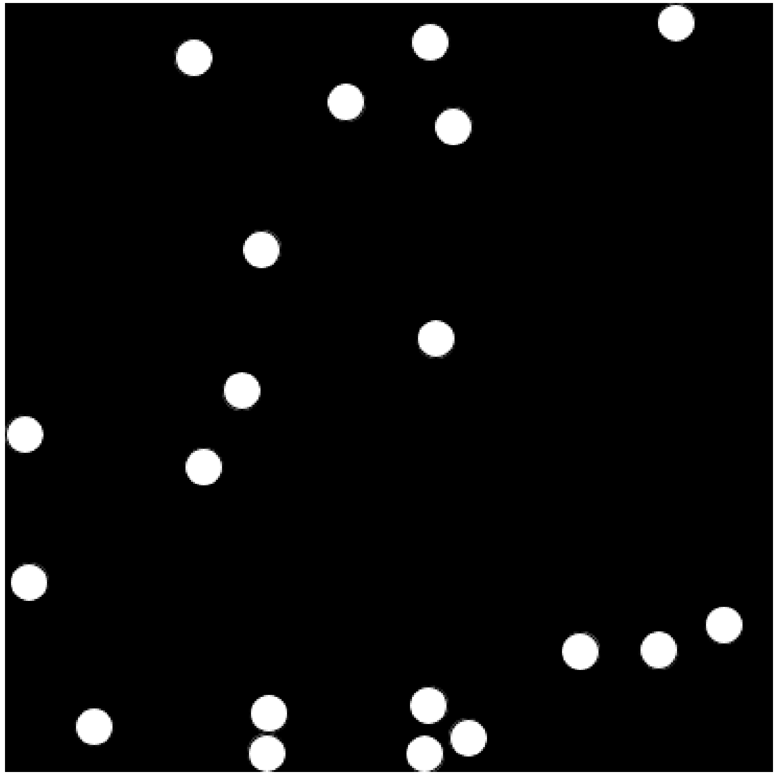
An array to store other balls

```
void checkCollisions() {
  for (Ball other: others) {
    collide(other);
  }
}
```

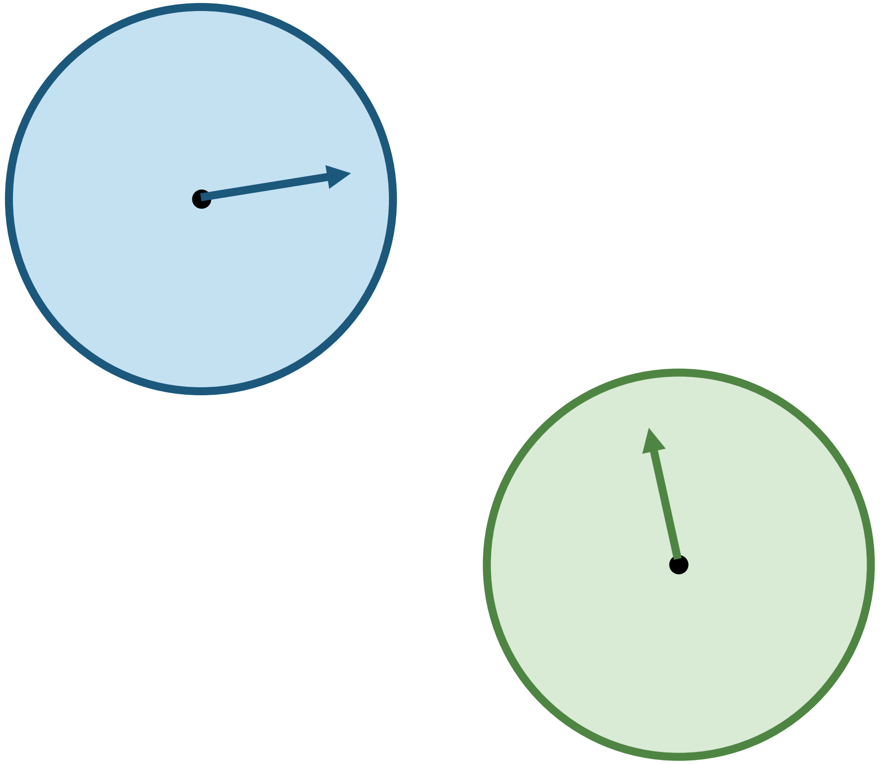
Iterate over other balls to check if there’s a collision

```
void collide(Ball other) {
  ???
}
```

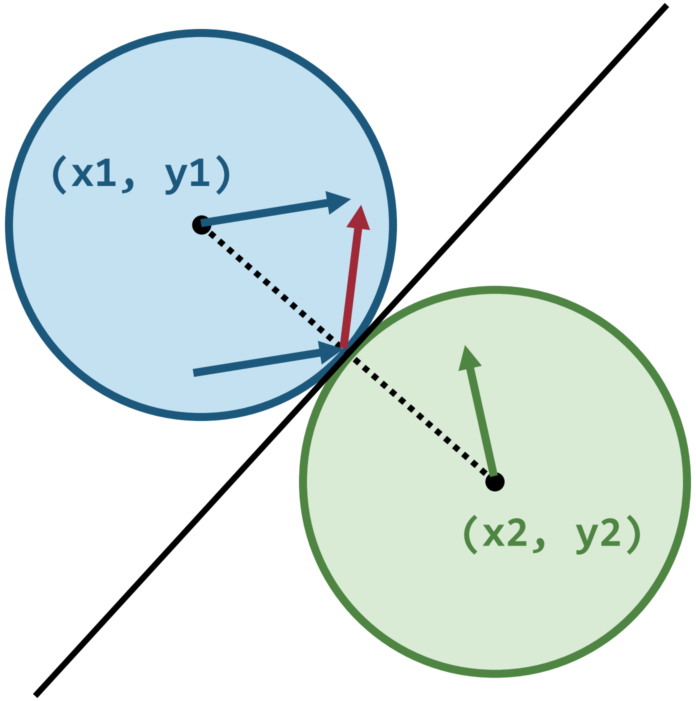
Check if there’s a collision



Handling Collisions

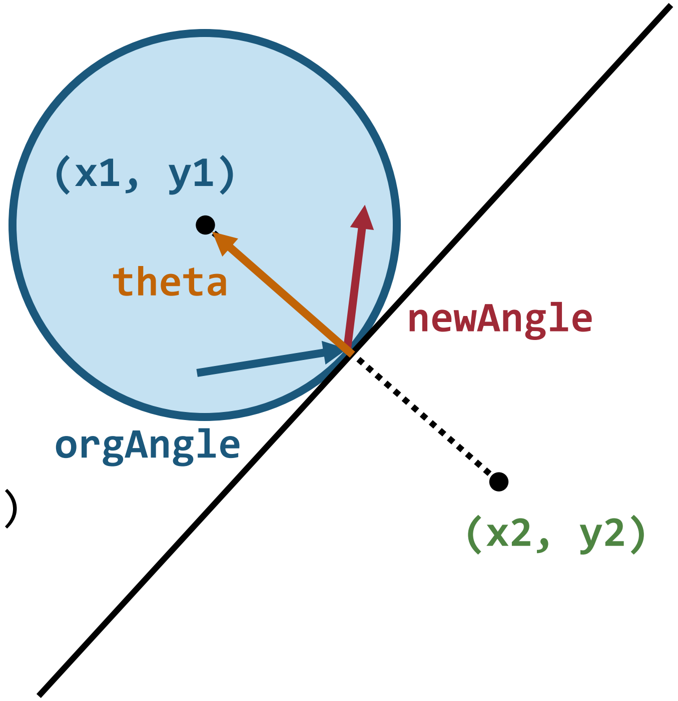


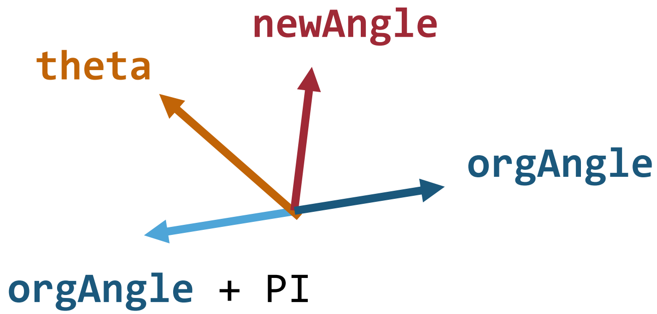
Handling Collisions



Handling Collisions

```
theta = atan2(y2 - y1, x2 - x1)

orgAngle = atan2(speedY, speedX)
```





```
newAngle = (orgAngle + PI) – 2 (orgAngle + PI – theta)
         = 2 theta - PI - orgAngle
```

Example: Bouncing Balls with Collision Detection

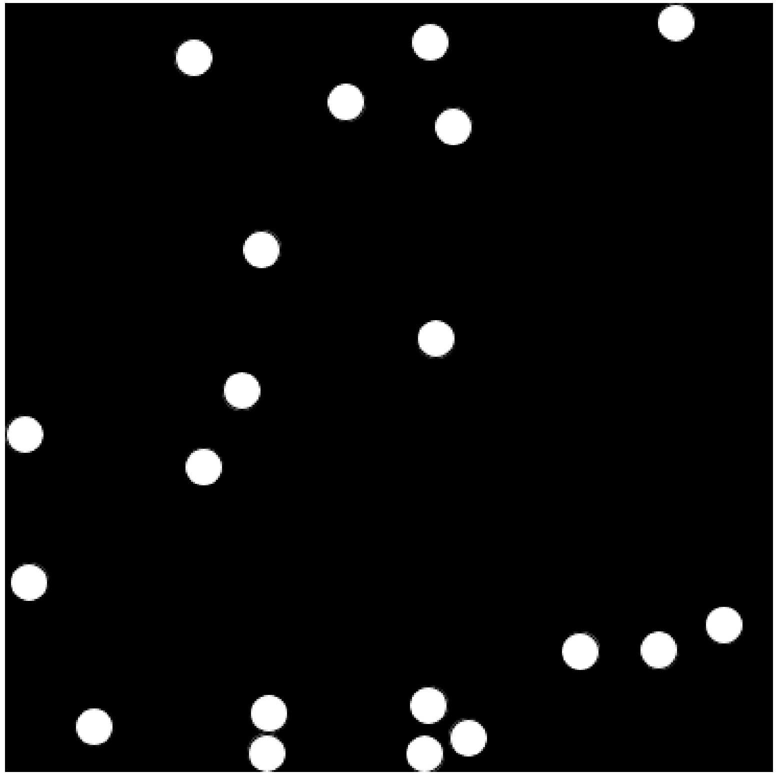
```
void collide(Ball other) {
  if (other == this) return;

  float dist = dist(x, y, other.x, other.y);

  if (dist >= size) return;

  x -= speedX;
  y -= speedY;

  float theta = atan2(other.y - y, other.x - x);
  float orgAngle = atan2(speedY, speedX);
  float newAngle = (theta - PI + theta - orgAngle);
  speedX = speed * cos(newAngle);
  speedY = speed * sin(newAngle);
}
```

Do nothing if it’s the same ball

Do nothing if they do not collide

Revert the ball back to where it was before the collision

Find the velocity after the collision



Elastic Collisions

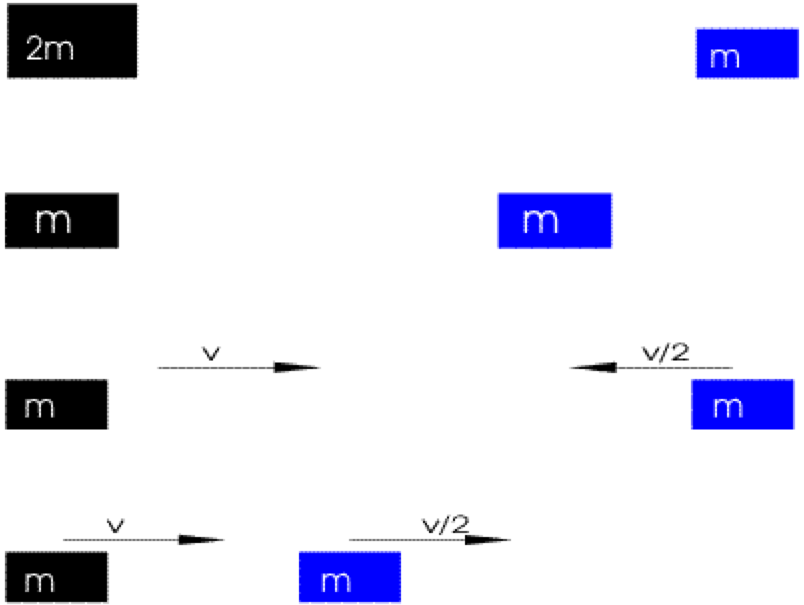
Conservation of momentum

$$m_1 v_1 + m_2 v_2 = m_1 v_1' + m_2 v_2'$$

Conservation of kinetic energy

$$m_1 v_1 + m_2 v_2 = \frac{1}{2} m_1 {v_1'}^2 + \frac{1}{2} m_2 {v_2'}^2$$

(Does not hold for inelastic collision)



(Source: Raul Roque, via Wikimedia Commons)

Raul Roque, CC BY-SA 2.5, via Wikimedia Commons

2D Elastic Collisions

Conservation of momentum

$$m_1\vec{v}_1 + m_2\vec{v}_2 = m_1\vec{v}_1' + m_2\vec{v}_2'$$

Conservation of kinetic energy

$$m_1\vec{v}_1 + m_2\vec{v}_2 = \frac{1}{2}m_1{\vec{v}_1'}^2 + \frac{1}{2}m_2{\vec{v}_2'}^2$$

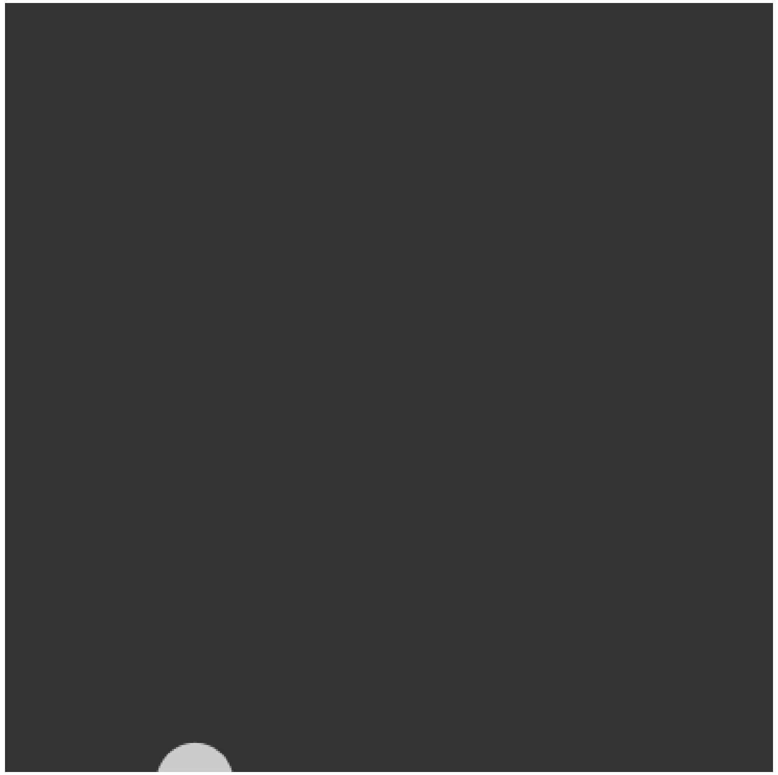
Conservation of angular momentum

(Source: Simon Steinmann, via Wikimedia Commons)

Simon Steinmann, CC BY-SA 2.5, via Wikimedia Commons

Example: Collision with Weights

- Different **weights** and different **speeds**
- Solve the equations to find the new velocities
  - Conservation of momentum
  - Conservation of kinetic energy
  - Conservation of angular momentum



processing.org/examples/circlecollision.html

Example: Reflection

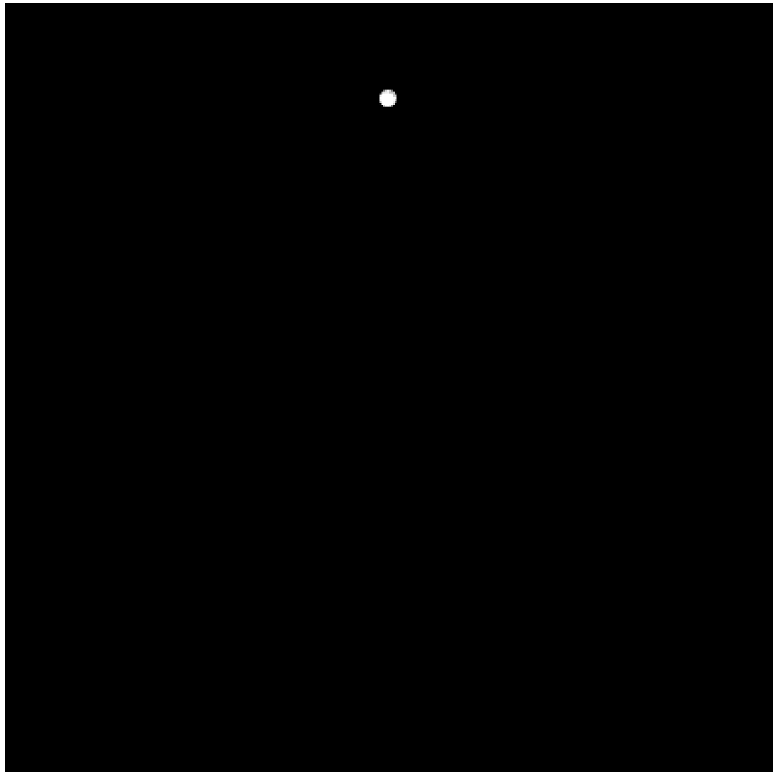
- Several `Ground` objects
  - Initialized with random widths and slopes
  - Together compose the ground surface
- An `Orb` object
  - Initialized with a small x-velocity
  - Influenced by gravity
  - May collide with the walls and the grounds

processing.org/examples/reflection2.html

Particle Systems

Example: Simple Particle System

- A **`Particle`** object
  - Falls with gravity
  - Fades out over time (die after a predefined lifespan)
- A **`ParticleSystem`** object is an ArrayList of **`Particle`** objects (i.e., **`ArrayList<Particle>`**)
  - Allows storing a variable number of particles
- Show all the **`Particle`** objects at each call of the `draw()` function



Example: Fireworks

- A **`Firework`** object
  - Starts as one single **`Particle`** object
    - Initialized with random force up
    - Flies up with gravity slowing it down
    - Explodes when speed reaches zero
  - Becomes many **`Particle`** objects after explosion
    - Initialized with random forces towards random directions
    - Fall with gravity
    - Die after invisible on the canvas