PAT 204/504 (Fall 2024)

Creative Coding

Lecture 13: Midterm Review

Instructor: Hao-Wen Dong

SCHOOL OF MUSIC, THEATRE & DANCE
PERFORMING ARTS TECHNOLOGY
UNIVERSITY OF MICHIGAN

Some More Things Worth Knowing

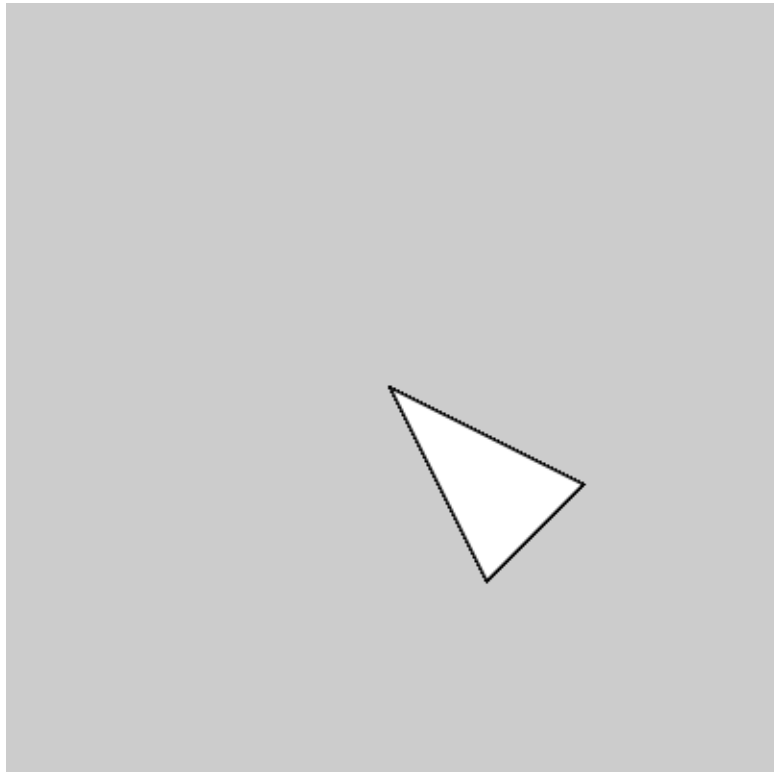
PShape – Object-oriented Interface for Shapes

```
PShape s;
```
Declare the shape

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

  s = createShape();
  s.beginShape();
  s.vertex(0, 0);
  s.vertex(50, 100);
  s.vertex(100, 50);
  s.endShape(CLOSE);
}

void draw() {
  shape(s, 200, 200);
}
```
Create the shape

Define the shape

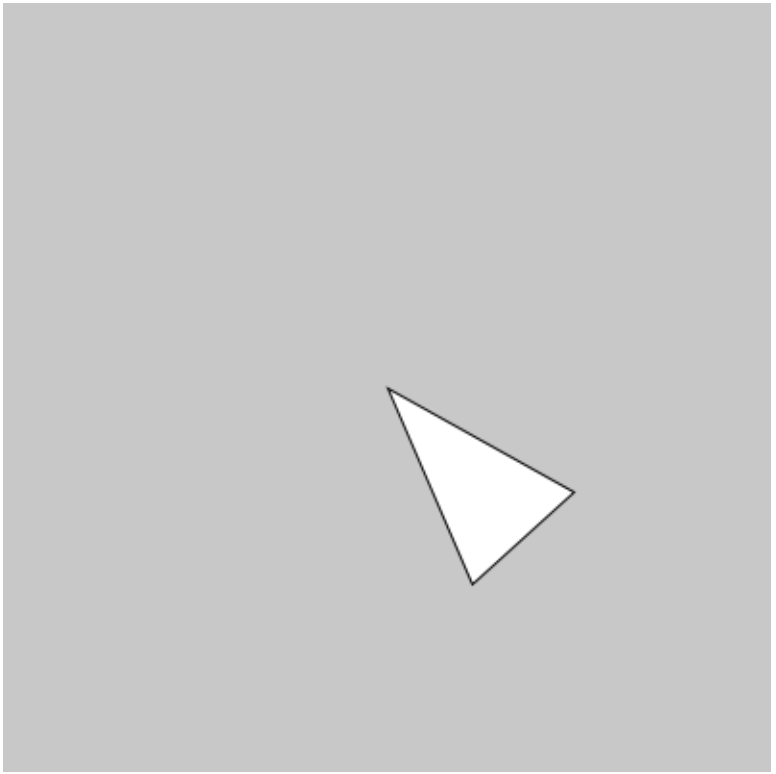
Show the shape



PShape – Object-oriented Interface for Shapes

```
PShape s;

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

  s = createShape();
  s.beginShape();
  s.vertex(0, 0);
  s.vertex(50, 100);
  s.vertex(100, 50);
  s.endShape(CLOSE);
}

void draw() {
  background(200);
  s.rotate(0.05);
  shape(s, 200, 200);
}
```

Rotate the shape



PShape – Object-oriented Interface for Shapes

- **`loadShape()`** Load a shape from an SVG or OBJ file

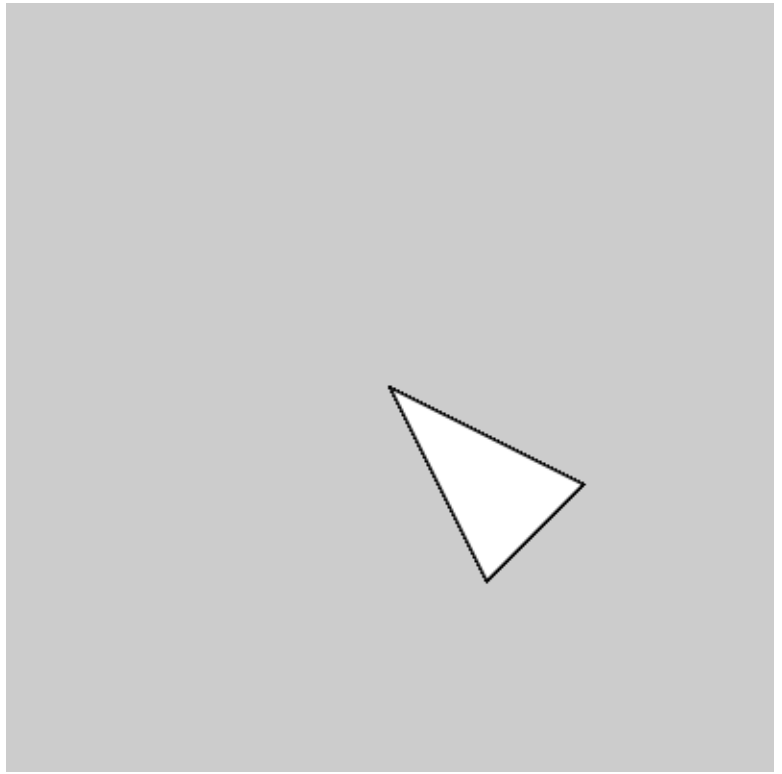
- **`translate()`** Translate the shape
- **`rotate()`** Rotate the shape
- **`scale()`** Scale the shape
- **`resetMatrix()`** Reset the transformation matrix of the shape

PShape – Object-oriented Interface for Shapes

```
PShape s;

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

  s = createShape();
  s.beginShape();
  s.vertex(0, 0);
  s.vertex(50, 100);
  s.vertex(100, 50);
  s.endShape(CLOSE);
}

void draw() {
  stroke(#FF0000);
  shape(s, 200, 200);
}
```

Does not apply to PShape

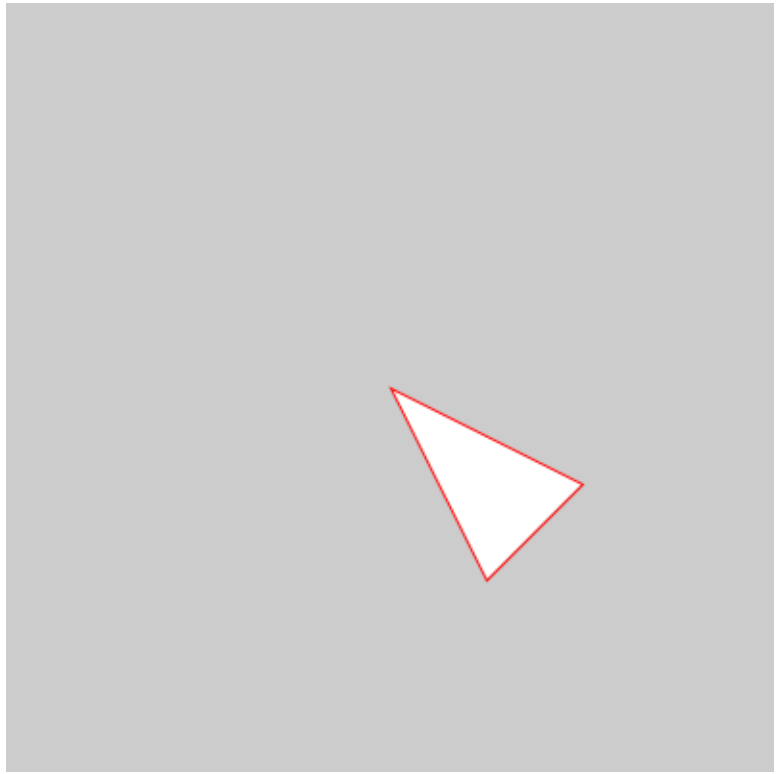
~~Set the stroke color~~



PShape – Object-oriented Interface for Shapes

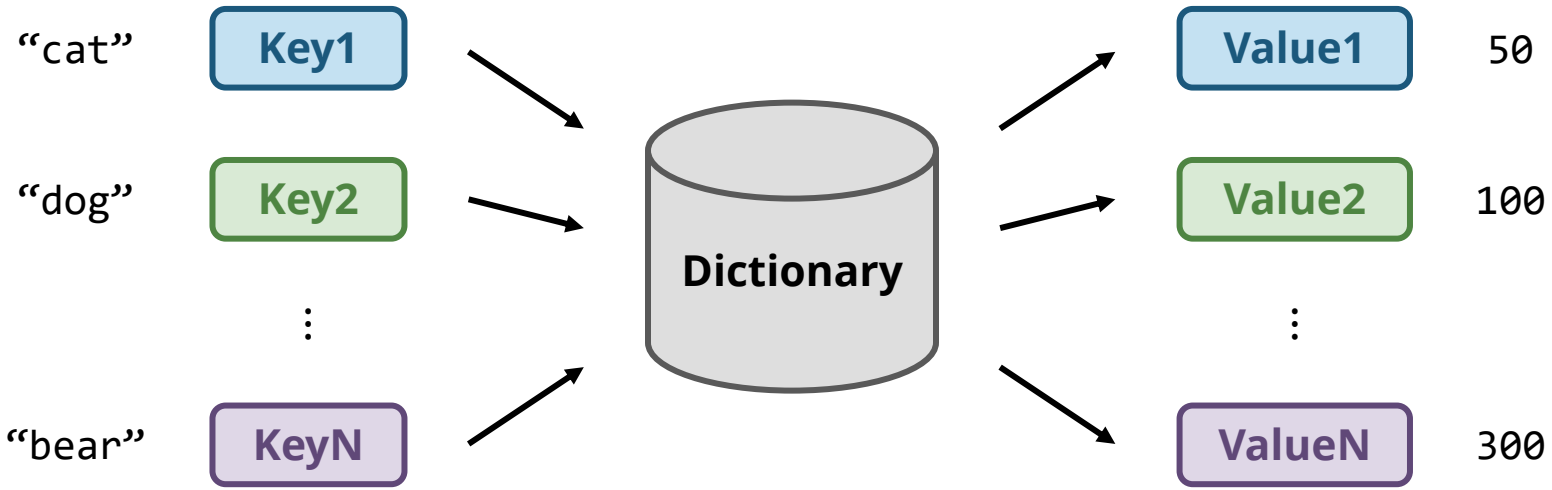
```
PShape s;

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

  s = createShape();
  s.beginShape();
  s.vertex(0, 0);
  s.vertex(50, 100);
  s.vertex(100, 50);
  s.endShape(CLOSE);
}

void draw() {
  s.setStroke(#FF0000);
  shape(s, 200, 200);
}
```

Object-oriented!



Dictionary

- A dictionary supports **fast lookup** of the corresponding "**value**" for a "**key**"
- **`IntDict`** String-to-integer mapping
- **`FloatDict`** String-to-float mapping
- **`StringDict`** String-to-string mapping



IntDict – String-to-Integer Mapping

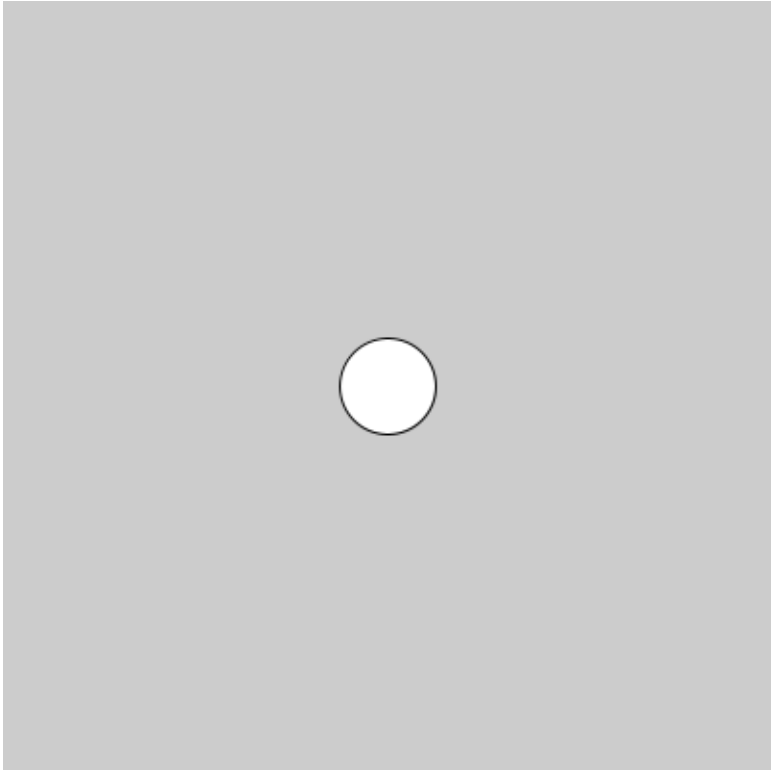
```
IntDict weightMap = new IntDict();
```

Declare the shape

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

  weightMap.set("cat", 50);
  weightMap.set("dog", 100);
  weightMap.set("bear", 300);
}

void draw() {
  int weight = weightMap.get("cat");
  circle(200, 200, weight);
}
```

Set up the dictionary

Query the dictionary



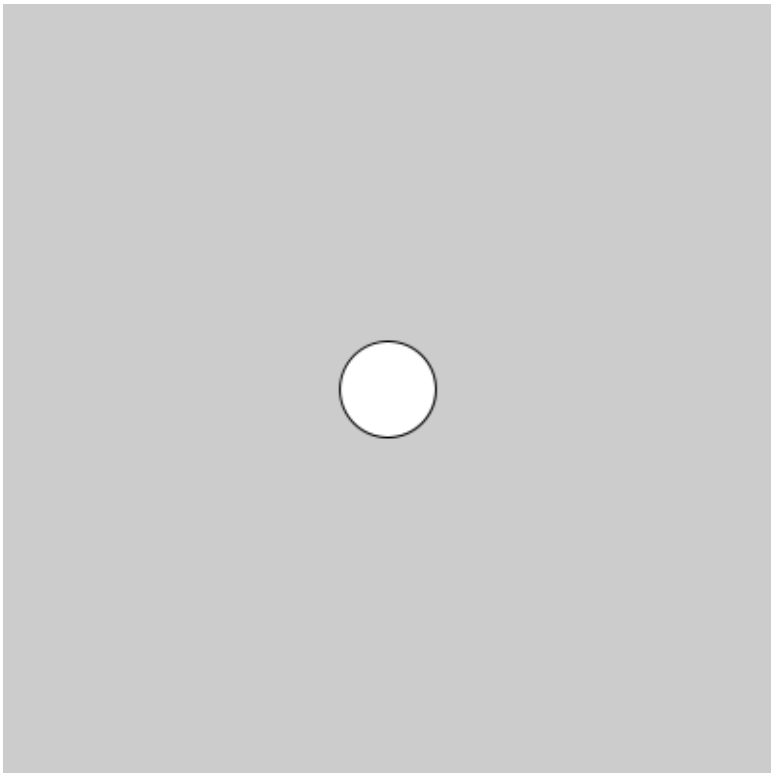
IntDict vs "string[] & int[]"

- Can we use **a string array** and **an integer array** to do something similar?

```
String[] animals = {"cat", "dog", "bear"};
int[] weights = {50, 100, 300};

void setup() {
  size(400, 400);
  noLoop();
}

void draw() {
  for (int i = 0; i < animals.length; i++) {
    if (animals[i] == "cat") {
      circle(200, 200, weights[i]);
    }
  }
}
```

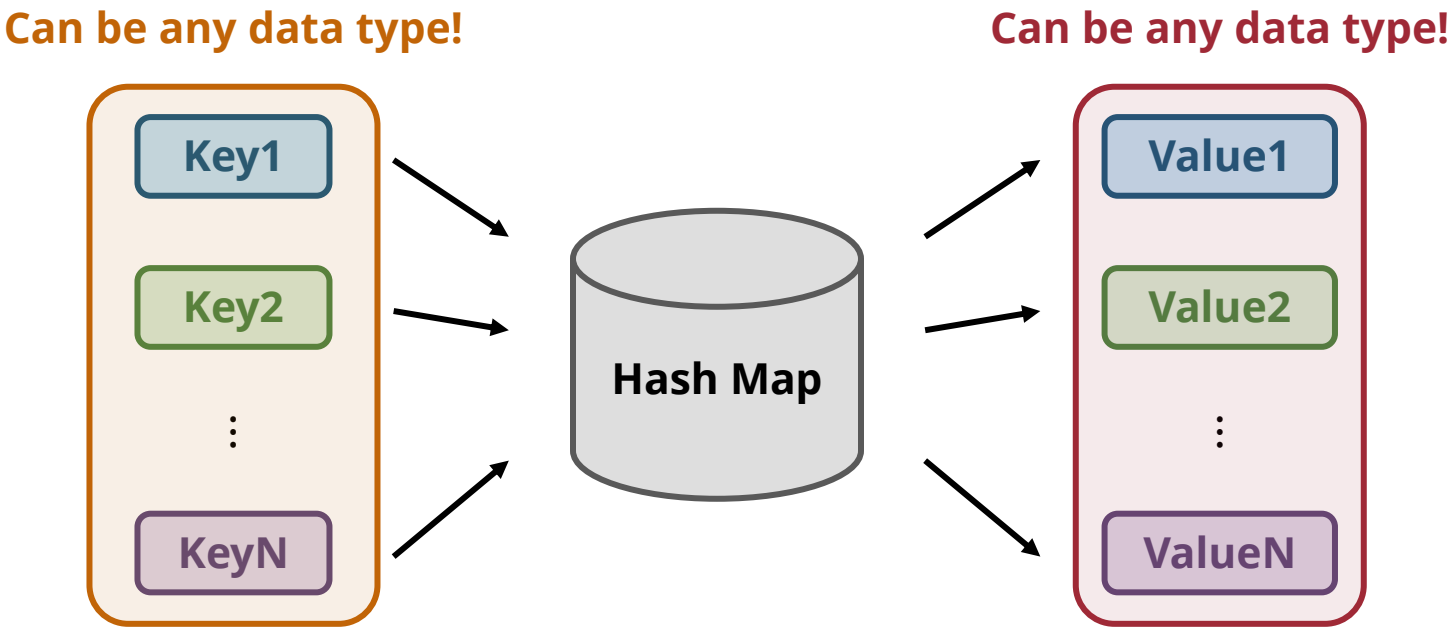
This is slow because we may need to go through the whole array to find the match!

Use a for loop to look up the "key"



Hash Map

- A **hash map** is the magic behind dictionaries
- **`HashMap`** Key-to-value mapping with desired key and value data types



A String-to-Array Hash Map

```
HashMap<String, int[]> m;
```

Declare a hash map that maps strings to integer arrays

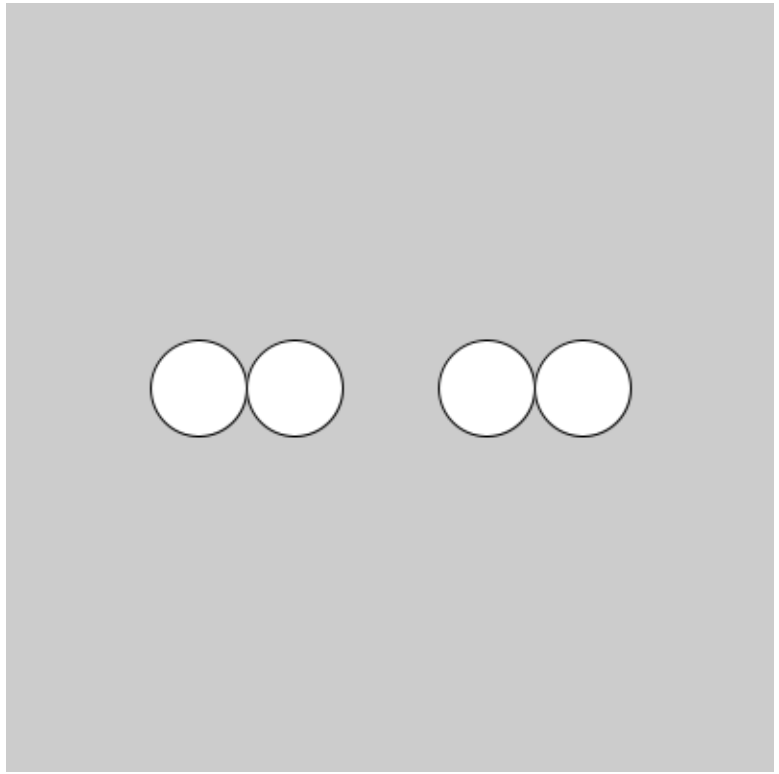
```
void setup() {
  size(400, 400);
  noLoop();

  m = new HashMap<String, int[]>();
  m.put("cat", new int[]{100, 150, 200});
  m.put("dog", new int[]{100, 300});
  m.put("bear", new int[]{100, 150, 250, 300});
}
```

Set up the dictionary

```
void draw() {
  int[] val = m.get("bear");
  for (int x: val) {
    circle(x, 200, 50);
  }
  println(m);
}
```

Query the hash map

```
{cat=[I@76cc6ab0, bear=[I@1d88e68, dog=[I@4e0d571f}
```



Review – Objects

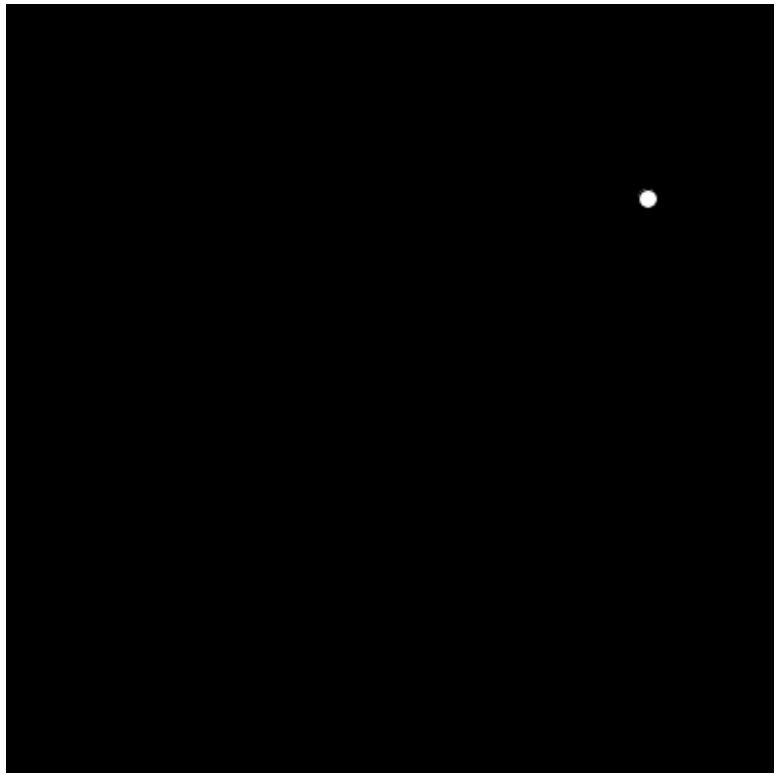
Example: Bouncing Ball

```
class Ball {
  float size = 10;
  float speed = 5;
  float x, y, speedX, speedY;

  Ball() {
    // Constructor
  }

  void show() {
    // Show the ball
  }

  void move() {
    // Move the ball
  }

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

Fields

Constructor

Methods

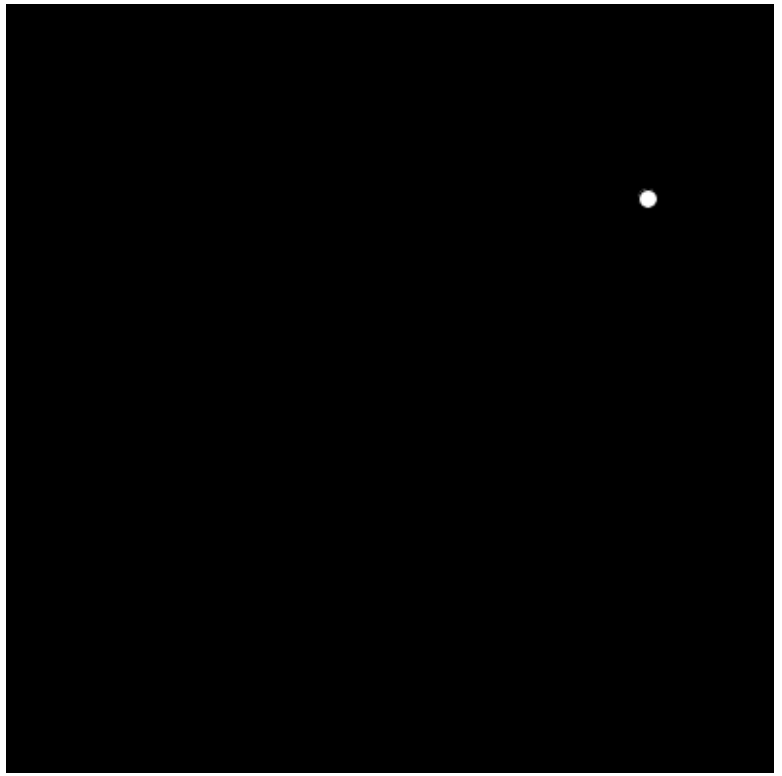


Example: Bouncing Ball

```
class Ball {
  ...

  void checkWalls() {
    float radius = size / 2;

    if (x > width - radius) {
      speedX = -abs(speedX);
    } else if (x < radius) {
      speedX = abs(speedX);
    }

    if (y > height - radius) {
      speedY = -abs(speedY);
    } else if (y < radius) {
      speedY = abs(speedY);
    }
  }

  ...
}
```

Check if the ball hit the left and right walls

Check if the ball hit the left and right walls



Example: Bouncing Ball

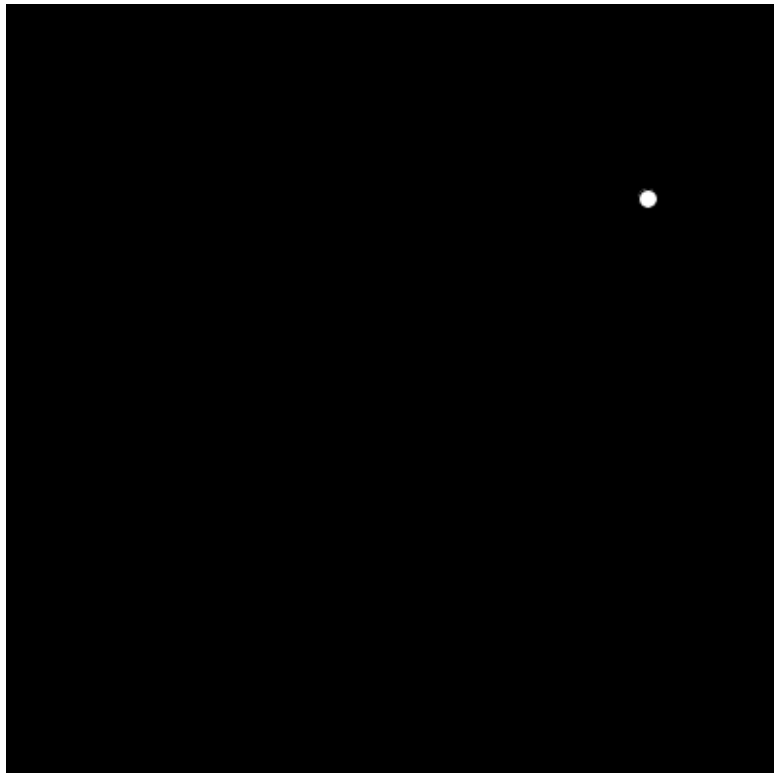
```
Ball ball;
```

Declaration

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

  ball = new Ball();
}

void draw() {
  background(0);

  ball.move();
  ball.checkWalls();
  ball.show();
}
```

Initialization

Call the methods!



Example: Bouncing Balls

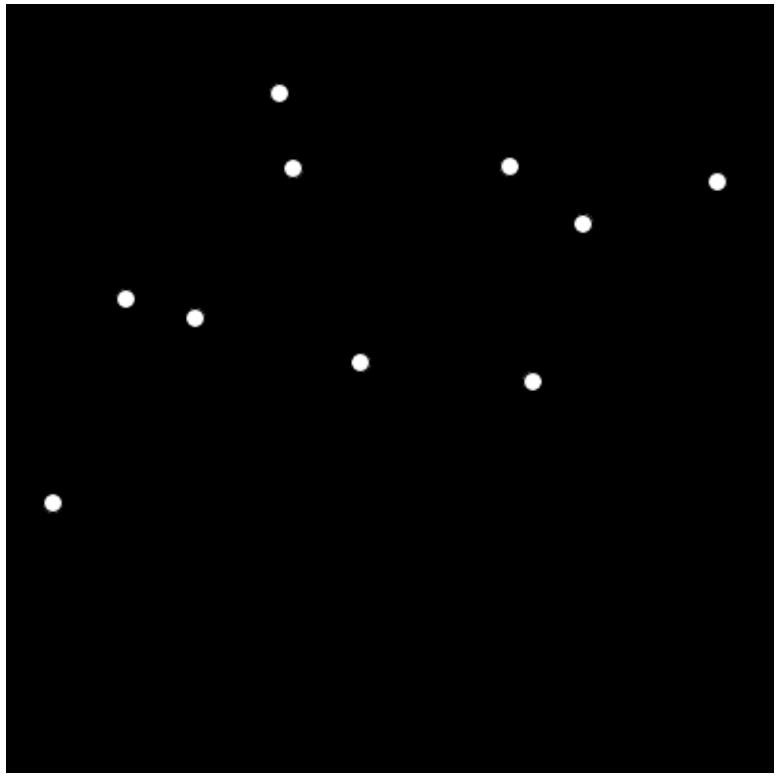
```
Ball[] balls = new Ball[20];
```

An array of objects

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

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

void draw() {
  background(0);

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

Initialization

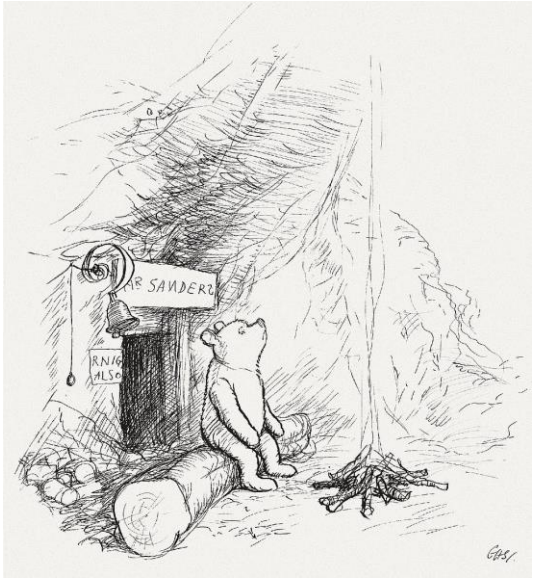
Call the methods!

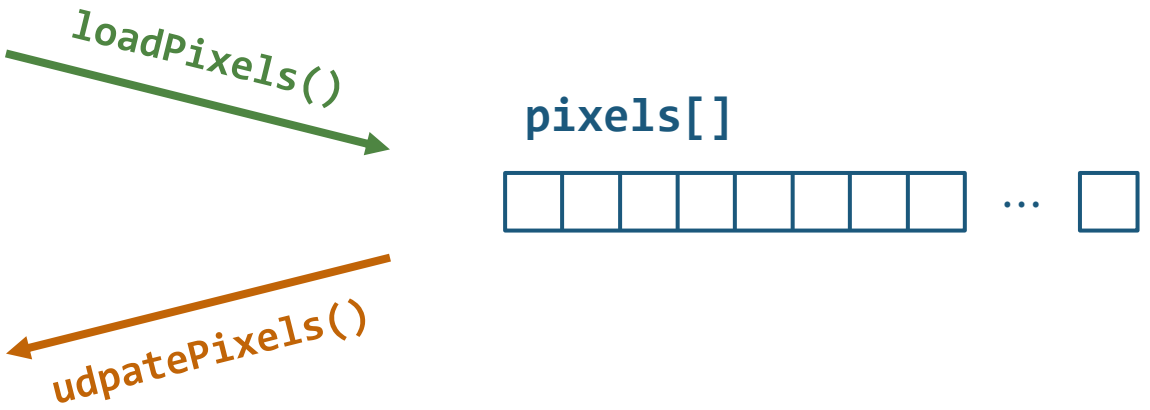


Review – Images

Loading the Pixels

- We can directly interact with the pixels of an image
  - `Image.pixels[]` Array of all the pixels in the image
  - `Image.loadPixels()` Load the image content to `Image.pixels[]`
  - `Image.updatePixels()` Update the image content with `Image.pixels[]`





Example: Manipulating Pixels

- Modify **`image.pixels`** directly
- Add a small red box at the bottom right

```
void setup() {
  size(400, 400);
  img = loadImage("pooh.jpg");
  img.loadPixels();
}

void draw() {
  for (int i = img.width - 50; i < img.width; i++) {
    for (int j = img.height - 50; j < img.height; j++) {
      img.pixels[j * img.width + i] = #ff0000;
    }
  }
  img.updatePixels();
  image(img, 0, 0, 400, 400);
}
```

Load the image

Load the image content to `pixels[]`

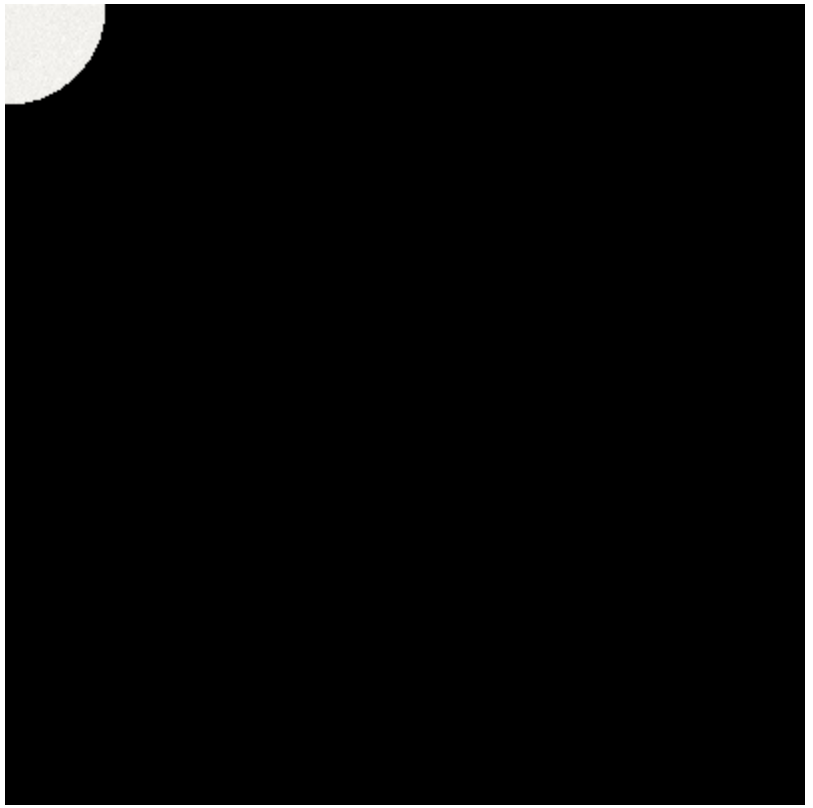
Update the pixel values

Update the image content with `pixels[]`

Exercise: The Reveal Effect

```
void setup() {
  size(400, 400);
  img = loadImage("pooh.jpg");
  image(img, 0, 0, 400, 400);
  loadPixels();
  org = pixels.clone();
  background(0);
  loadPixels();
}

void draw() {
  for (int x = 0; x < width; x++) {
    for (int y = 0; y < height; y++) {
      int loc = x + y * width;
      float d = dist(x, y, mouseX, mouseY);
      if (d < 50) {
        pixels[loc] = org[loc];
      }
    }
  }
  updatePixels();
}
```

Update the pixel values



Example: Pointillism

```
PImage img;

void setup() {
  size(400, 400);
  img = loadImage("sakura.jpg");
  background(255);
  noLoop();
}

void draw() {
  for (int i = 0; i < 10000; i++) {
    int x = int(random(img.width));
    int y = int(random(img.height));
    int loc = x + y * img.width;

    img.loadPixels();
    float r = red(img.pixels[loc]);
    float g = green(img.pixels[loc]);
    float b = blue(img.pixels[loc]);

    noStroke();
    fill(r, g, b, 100);
    circle(x, y, 20);
  }
}
```

Pick a random pixel

Find the color of the pixel

Set the color of the circle

Draw the circle

Example: Loading a Movie

```
import processing.video.*;

Movie myMovie;

void setup() {
  size(640, 360);
  myMovie = new Movie(this, "movie.mov");
  myMovie.loop();
}

void movieEvent(Movie m) {
  m.read();
}

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

Import video library

Initialize the `movie` object

Called whenever a new frame is available to read

Example: Webcam Capture

```
import processing.video.*;

Capture cam;

void setup() {
  size(640, 480);

  println(Capture.list());
  cam = new Capture(this, 640, 480);
  cam.start();
}

void draw() {
  if (cam.available() == true) {
    cam.read();
  }
  image(cam, 0, 0);
}
```

Import video library

Print the webcam list

Initialize the `Capture` object

Read the frame whenever it's available

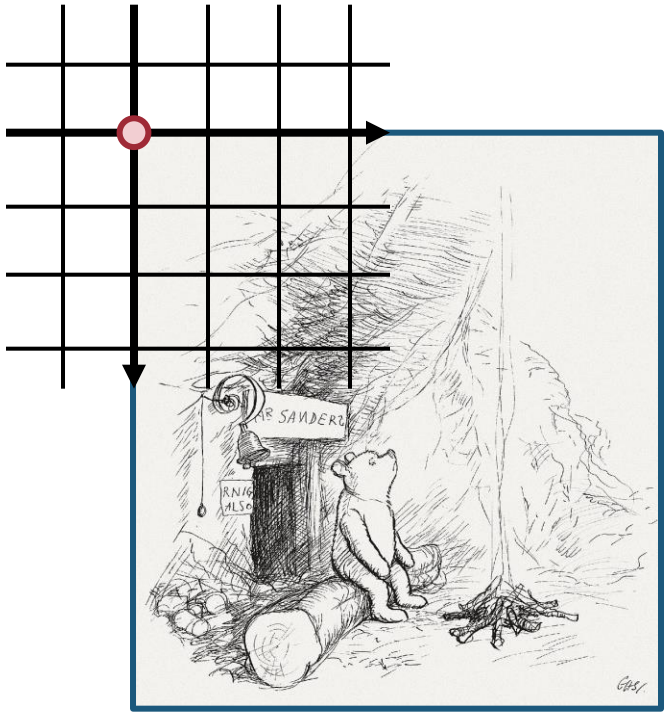
Review – Transformation

Transformations

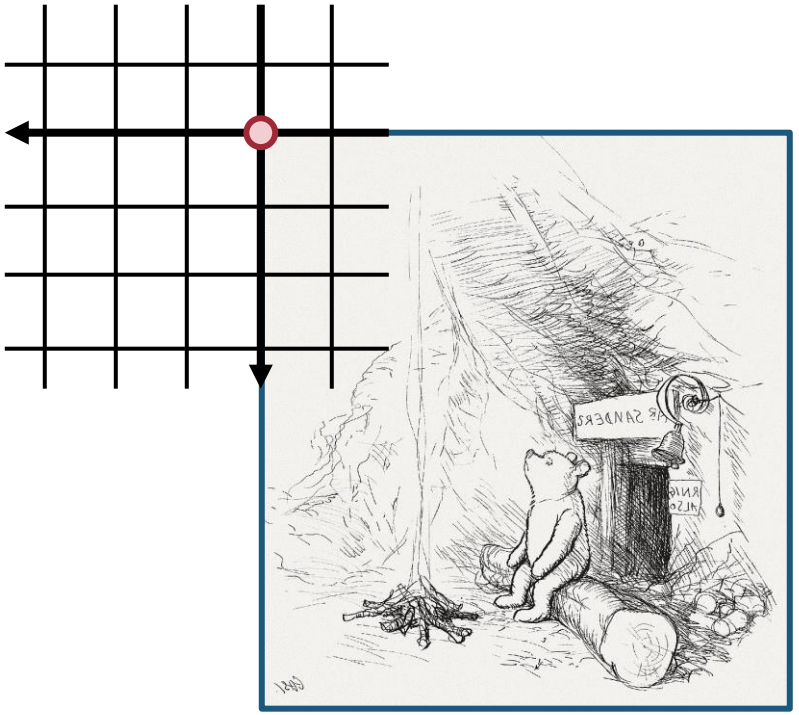
- `translate(x, y)` Translate the object
- `rotate(angle)` Rotate the object
- `scale(s)` Scale the object
- `scale(x, y)` Scale the object

Example: Mirroring Capture

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

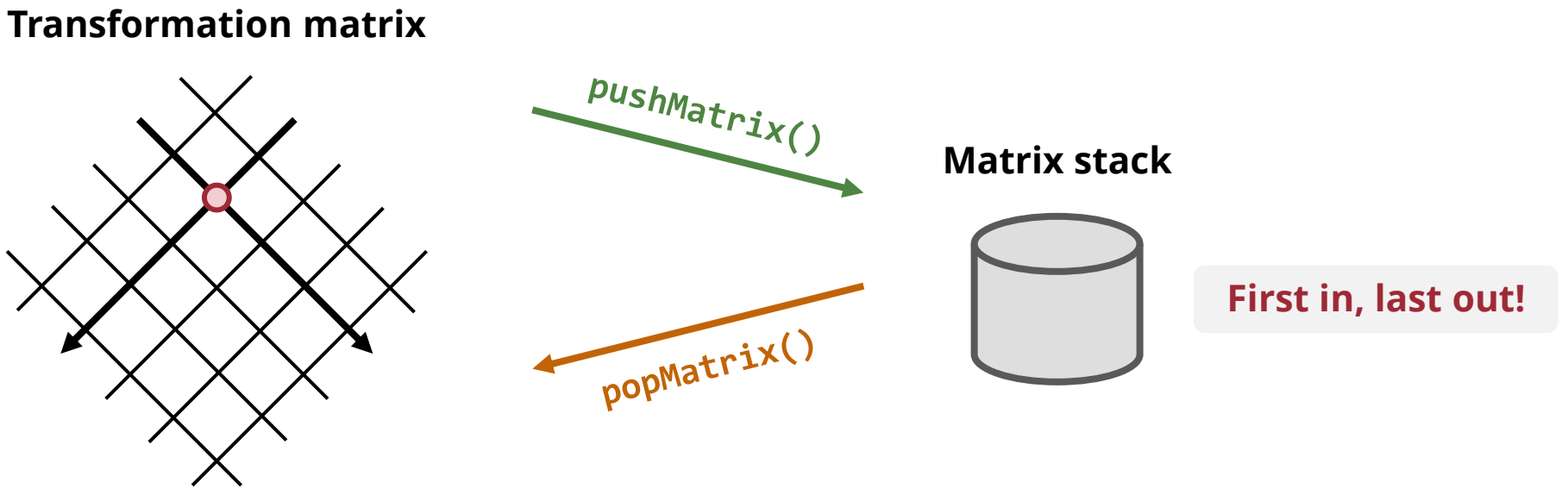


```
void draw() {
  scale(-1, 1);
  image(video, -video.width, 0);
}
```



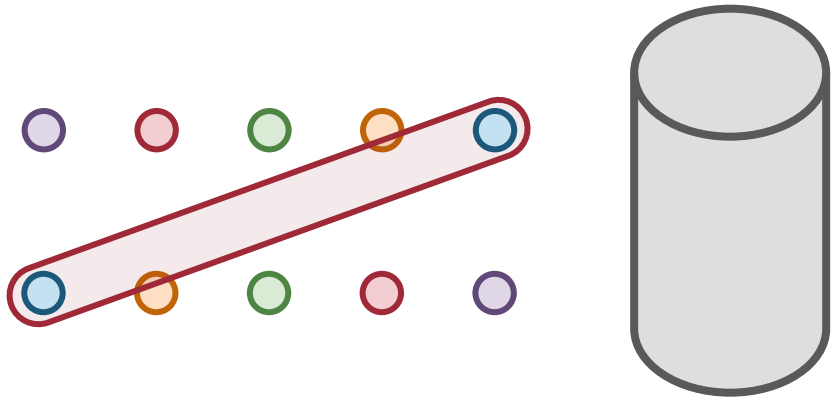
Matrix Transforms

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



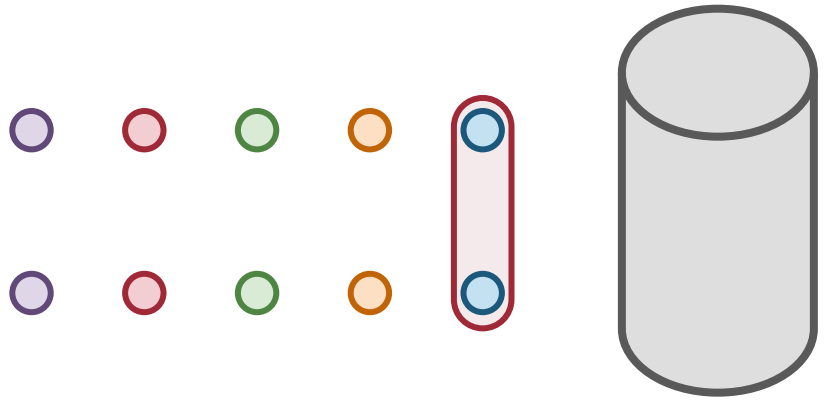
Stack vs Queue

Stack



First in last out

Queue



First in first out

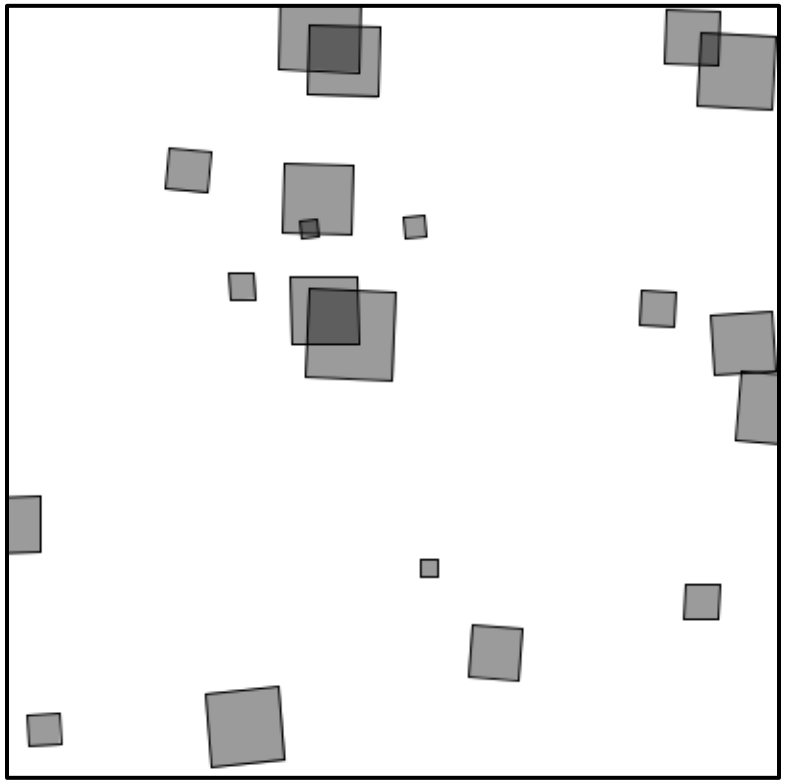
Example: Spinning Objects

```
Rotater[] rotaters = new Rotater[20];
float x, y, speed, w;

void setup() {
  size(400, 400);
  for (int i = 0; i < rotaters.length; i++) {
    x = random(width);
    y = random(height);
    speed = random(-0.1, 0.1);
    w = random(5, 50);
    rotaters[i] = new Rotater(x, y, speed, w);
  }
}

void draw() {
  background(255);
  for (Rotater rotater: rotaters) {
    rotater.spin();
    rotater.display();
  }
}
```

Declare an array of rotater objects

Initialize each rotater with a random position, size and rotation speed

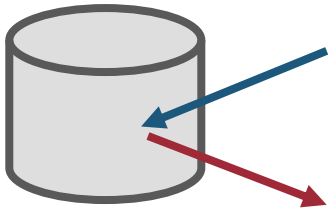
Spin and show the rotaters!



Example: Spinning Objects

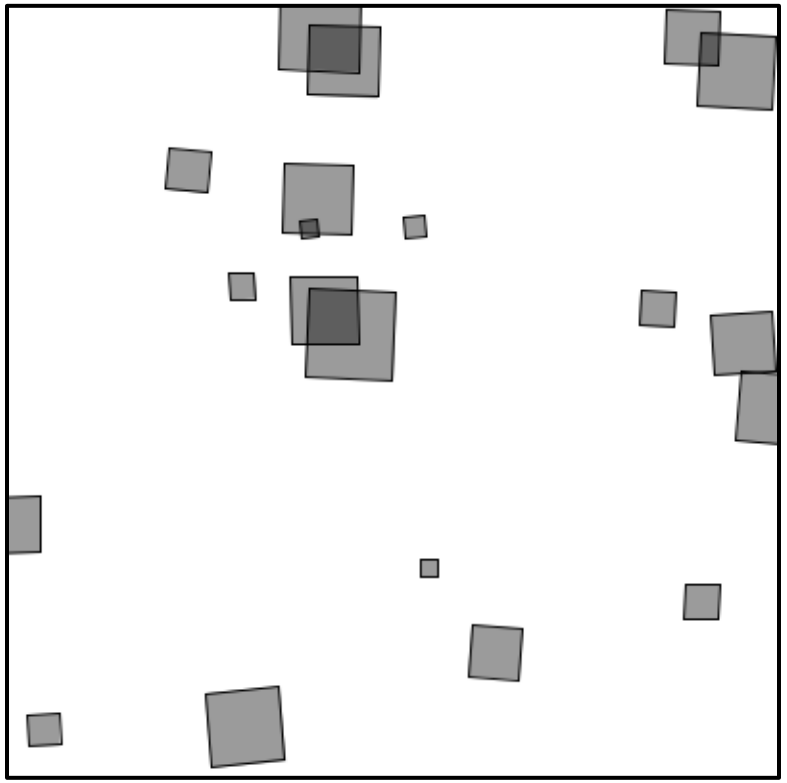
```
class Rotater {
  ...

  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();
  }
}
```

Matrix stack



Store the current matrix

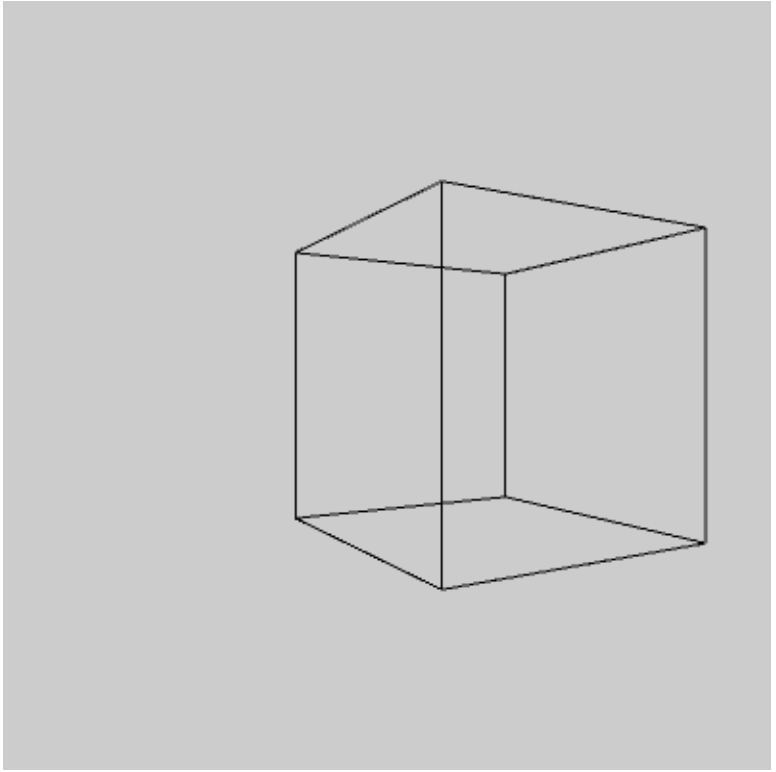
Restore the stored matrix



Review – 3D Graphics

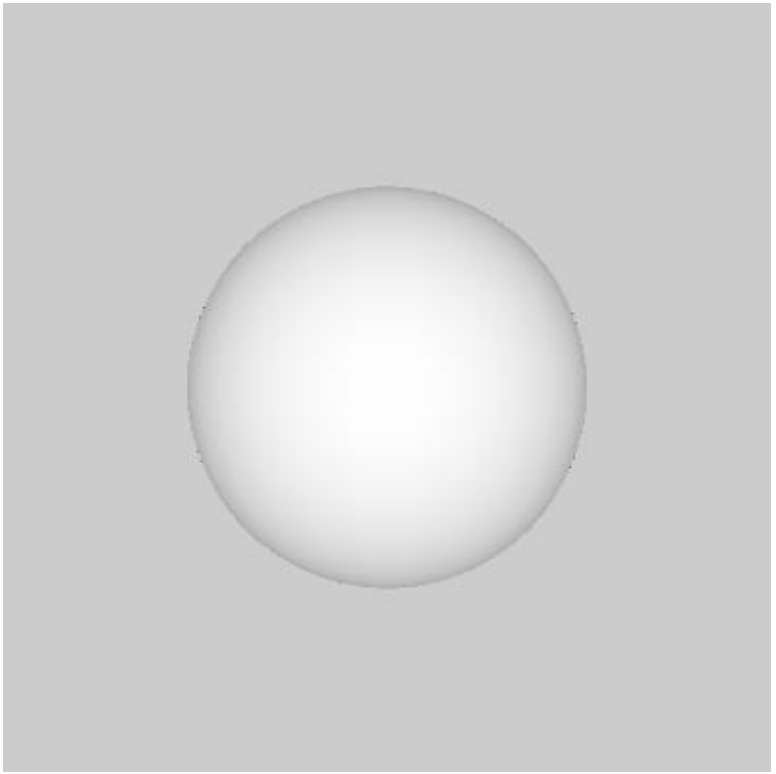
Example: Box

```
size(400, 400, P3D);
translate(250, 200, 0);
rotateY(0.5);
fill(0, 10);
box(150);
```



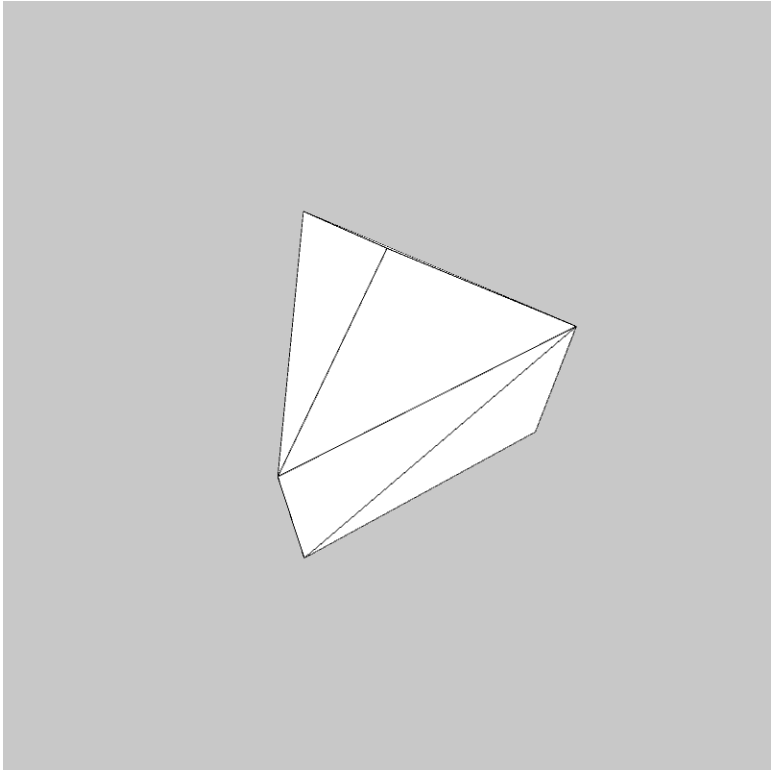
Example: Sphere

```
size(400, 400, P3D);
noStroke();
lights();
translate(200, 200, 0);
sphere(100);
```



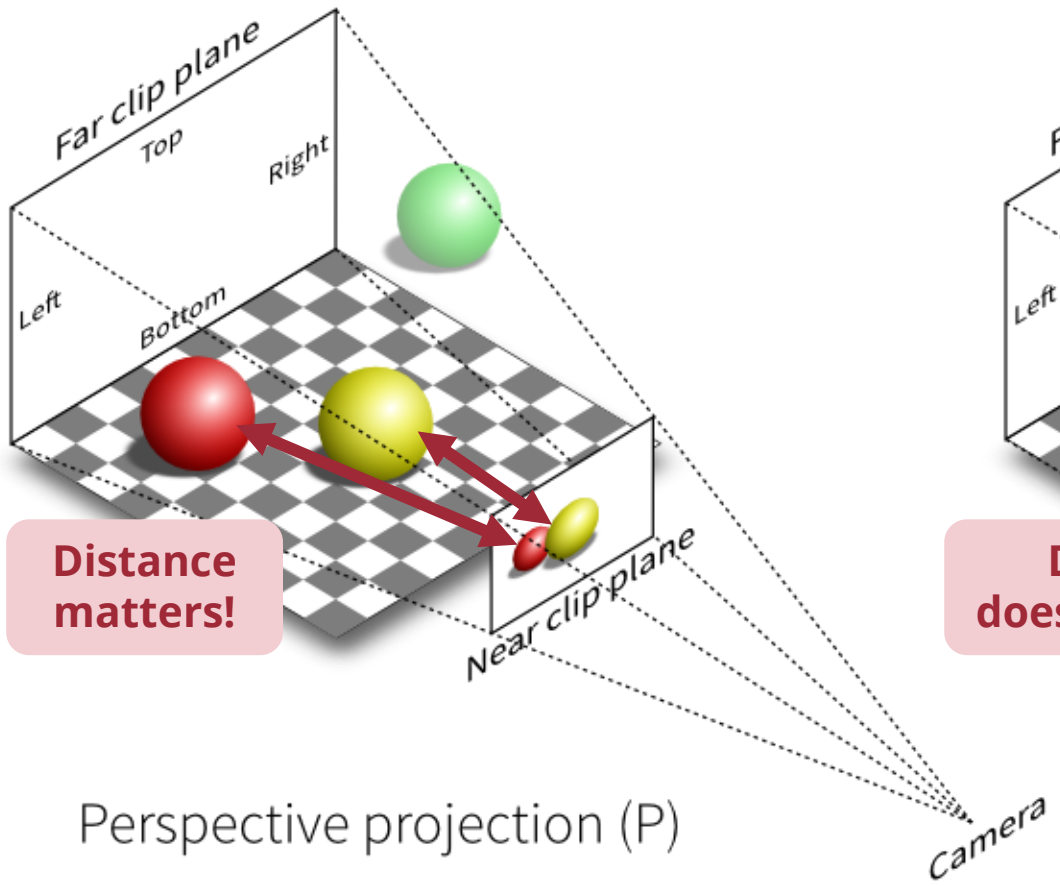
Sphere Details

```
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();
}
```



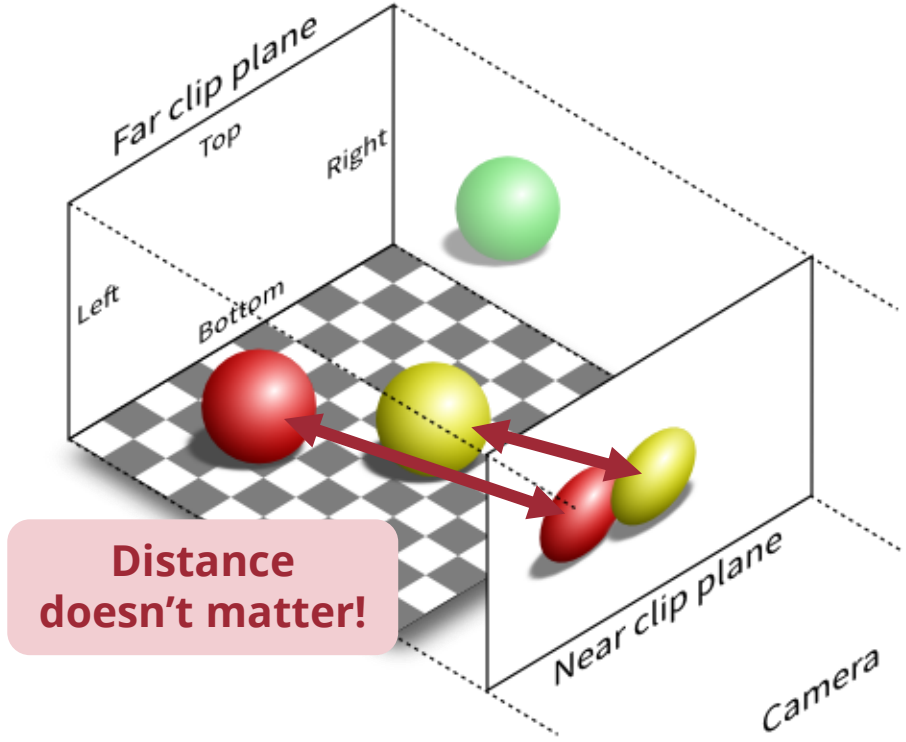
Perspective vs Orthographic Projections

perspective()



Perspective projection (P)

ortho()

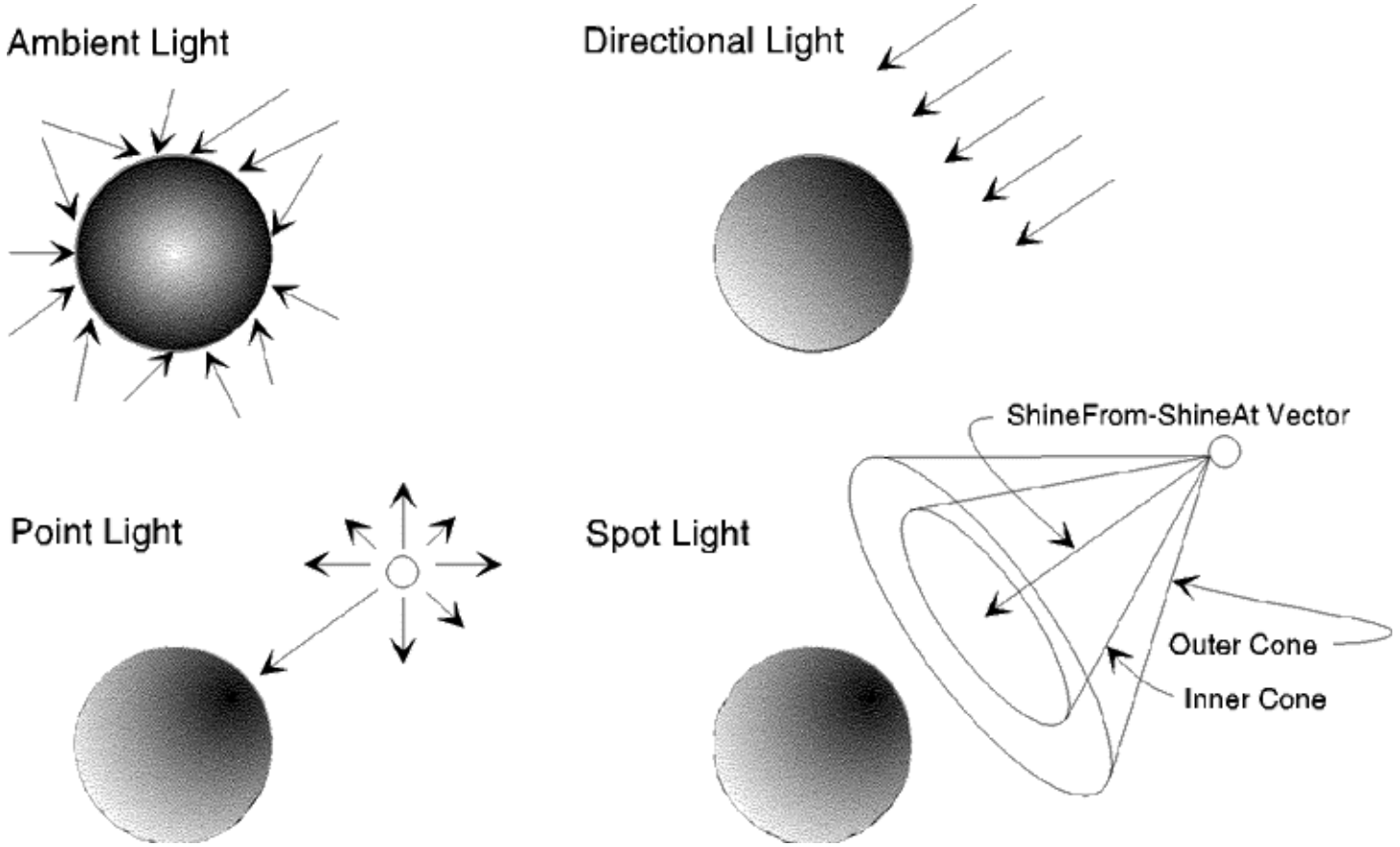


Orthographic projection (O)

glumpy.github.io/modern-gl.html

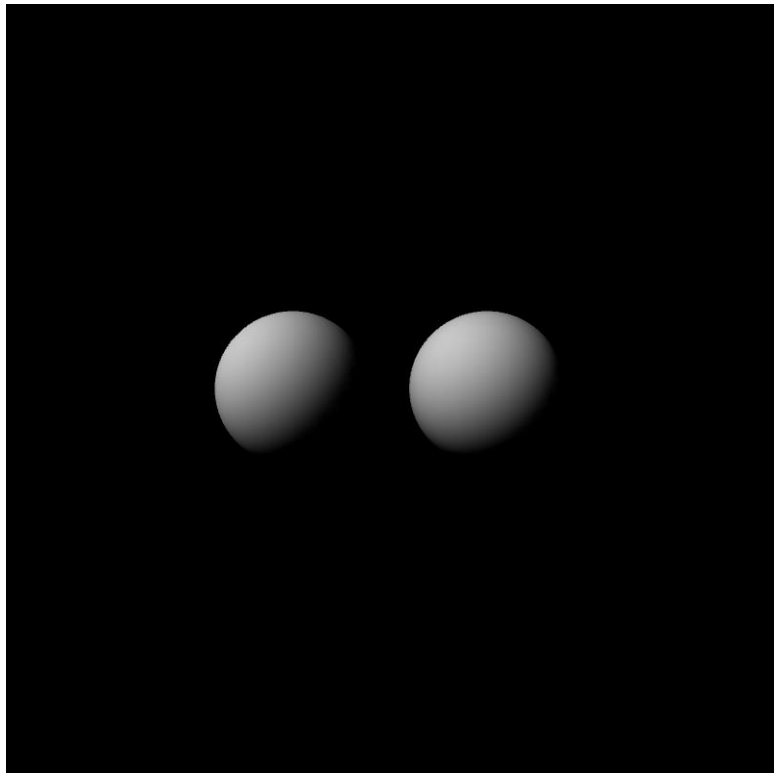
Lights

- `ambientLight()`
- `directionalLight()`
- `spotlight()`
- `pointLight()`



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);
}
```



Review – Motion & Physics

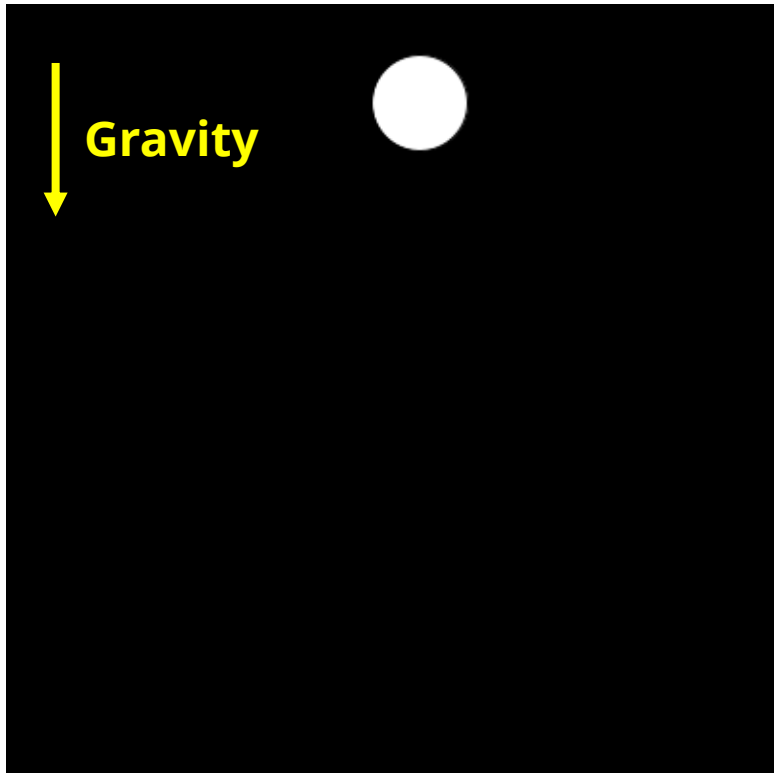
Example: Gravity

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

Apply gravity as y-acceleration

```
// 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;
  }
}
```

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



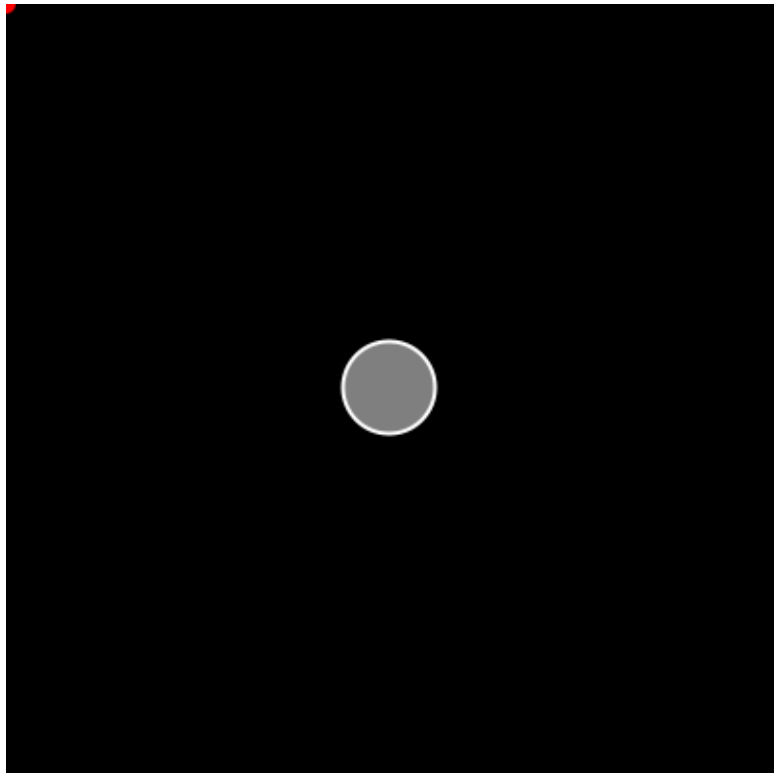
Example: Acceleration

```
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

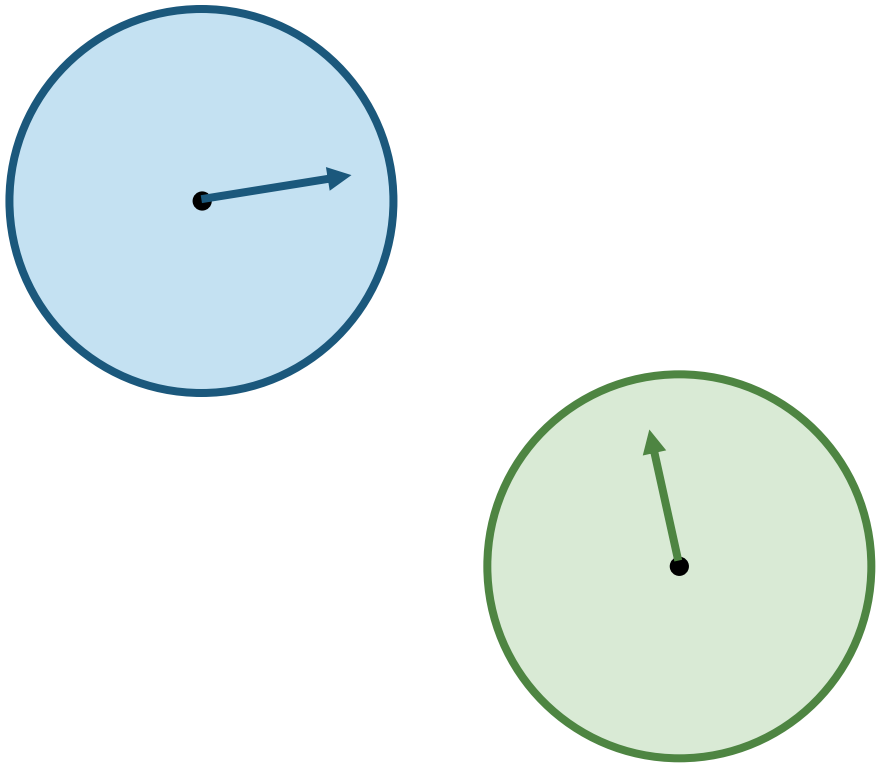
Apply the acceleration

Move the ball

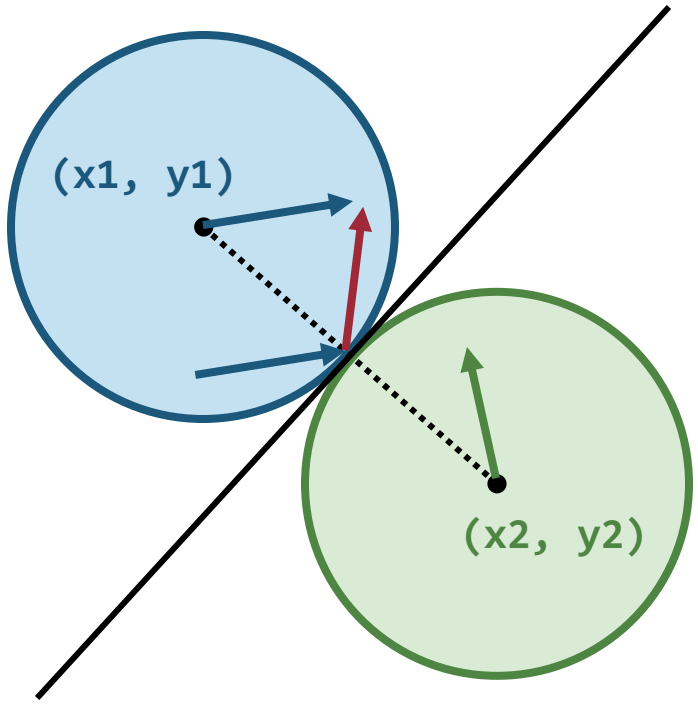


processing.org/examples/accelerationwithvectors.html

Handling Collisions



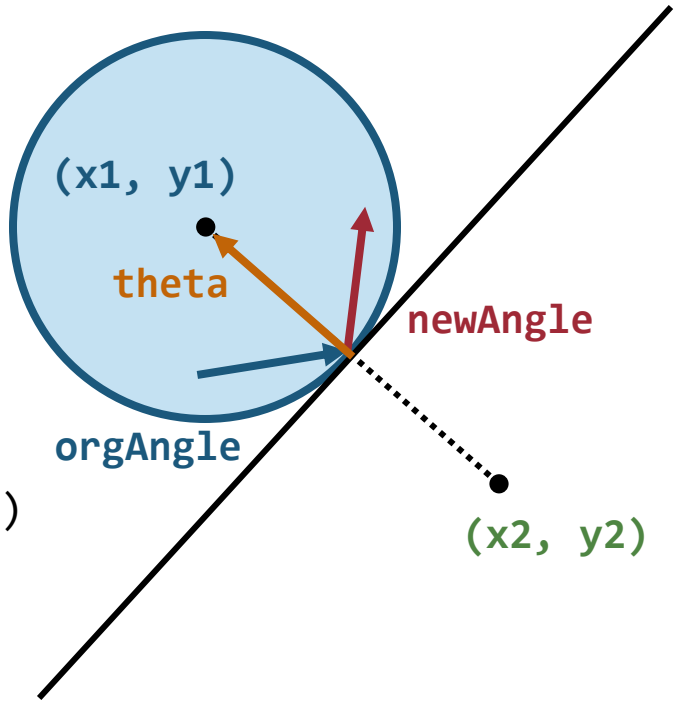
Handling Collisions

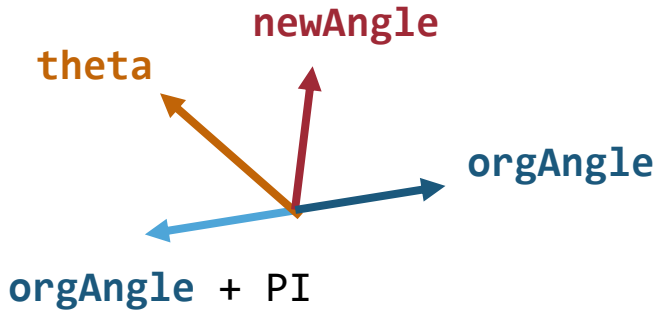


Handling Collisions

```
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

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

```
Ball[] others;
```

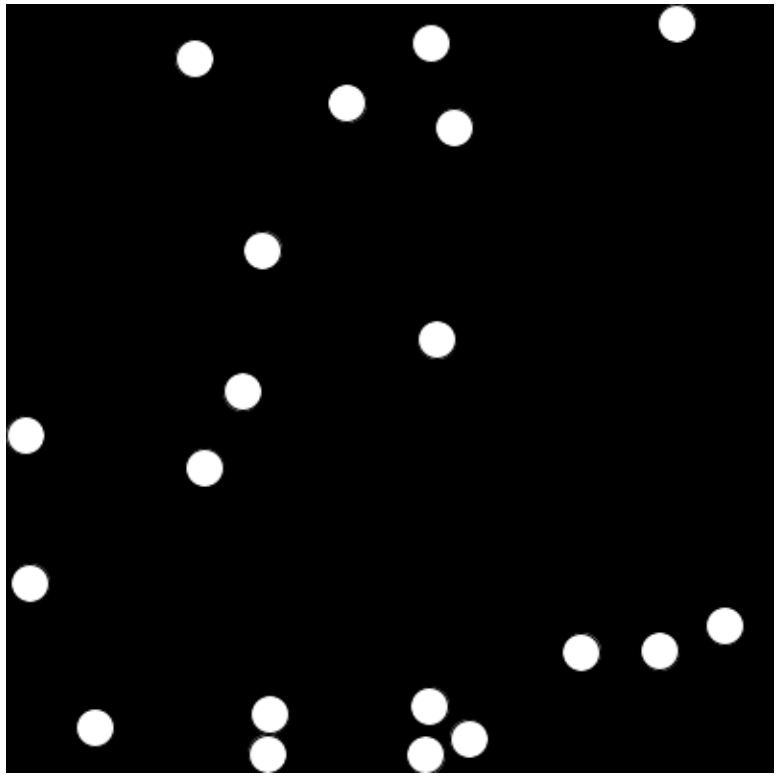
An array to store other balls

```
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



Example: Bouncing Balls with Collision Detection

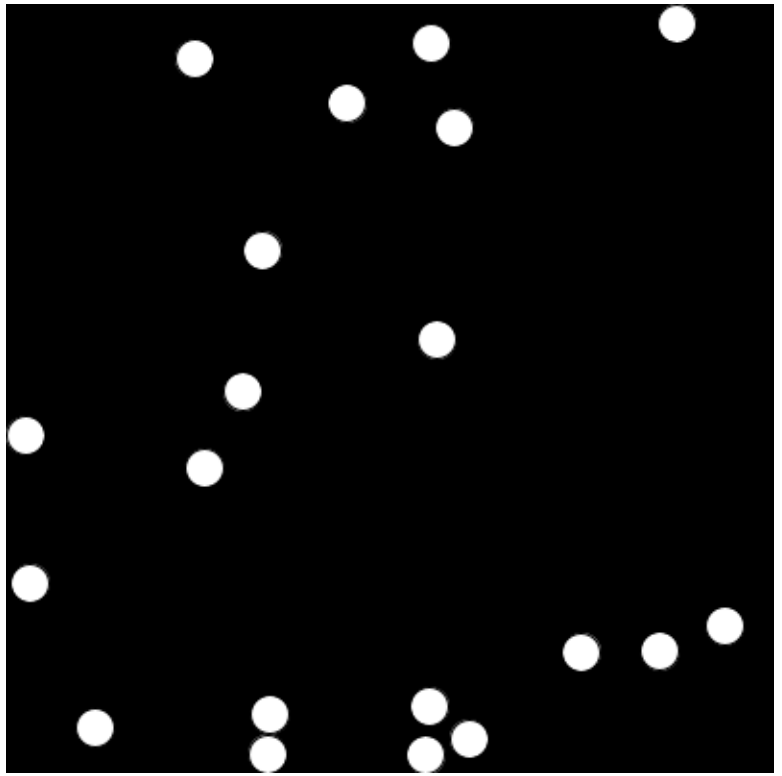
```
void collide(Ball other) {
  if (other == this) return;
  float dist = dist(x, y, other.x, other.y);
  if (dist >= size) return;
  x -= speedX;
  y -= speedY;
  float theta = atan2(other.y - y, other.x - x);
  float orgAngle = atan2(speedY, speedX);
  float newAngle = (theta - PI + theta - orgAngle);
  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



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

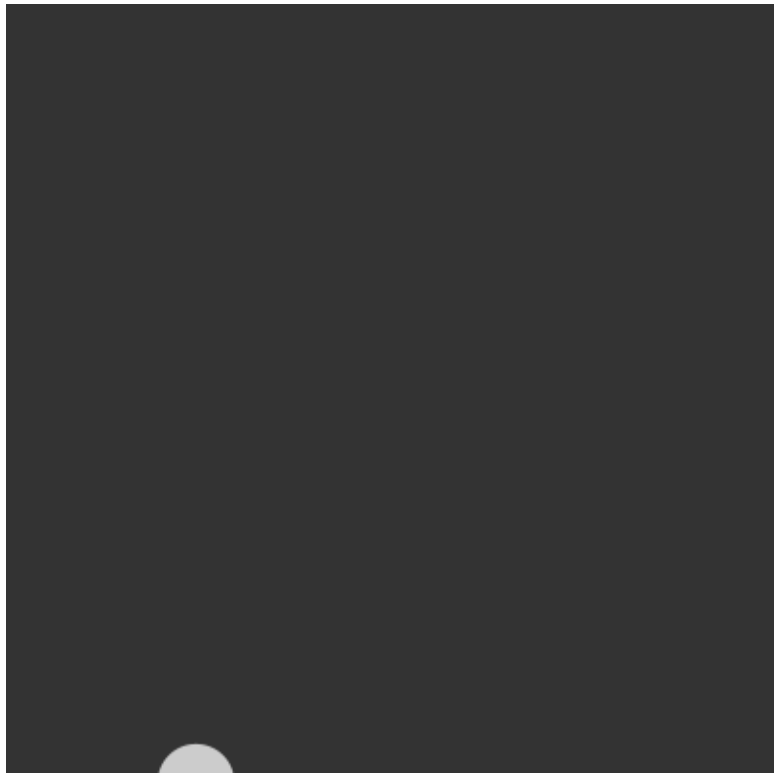
$$m_1\vec{v}_1 + m_2\vec{v}_2 = \frac{1}{2}m_1\vec{v}_1'^{\,2} + \frac{1}{2}m_2\vec{v}_2'^{\,2}$$

Conservation of angular momentum

(Source: Simon Steinmann, via Wikimedia Commons)

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

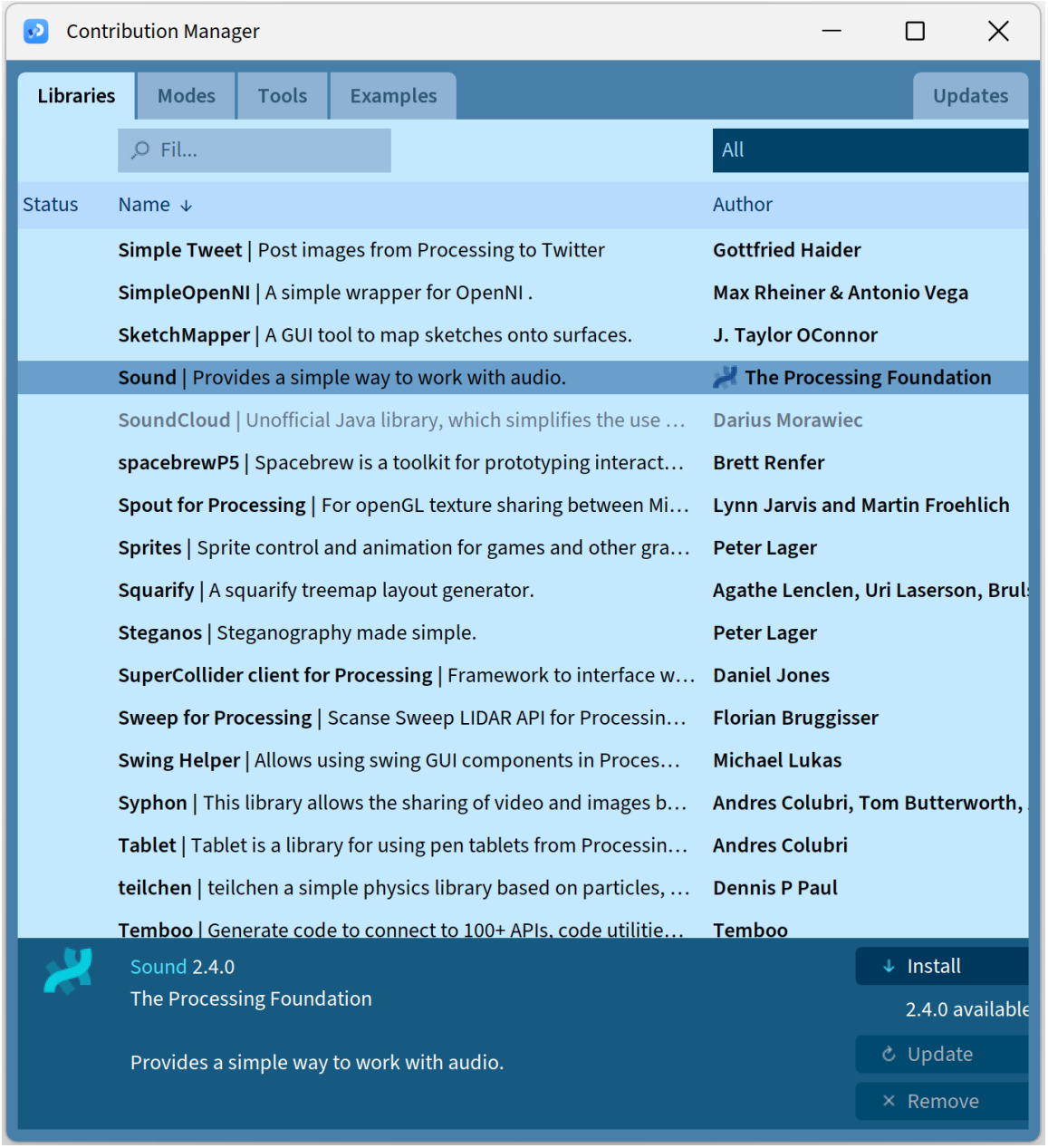
Example: Fireworks

- 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

Review – Libraries

Library Manager

- Official Libraries maintained by the **Processing Foundation**
  - Sound
  - Video
  - Hardware I/O
  - JavaFX
- Many other libraries
  - Networking
  - GUI
  - Animation



(Recap) Amplitude Class

```
import processing.sound.*;

Amplitude amp = new Amplitude(this);
AudioIn in = new AudioIn(this, 0);
float a;

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

  in.start();
  amp.input(in);
}

void draw() {
  background(0);
  a = amp.analyze();
  circle(200, 200, a * 400);
}
```

Initialize an Amplitude object

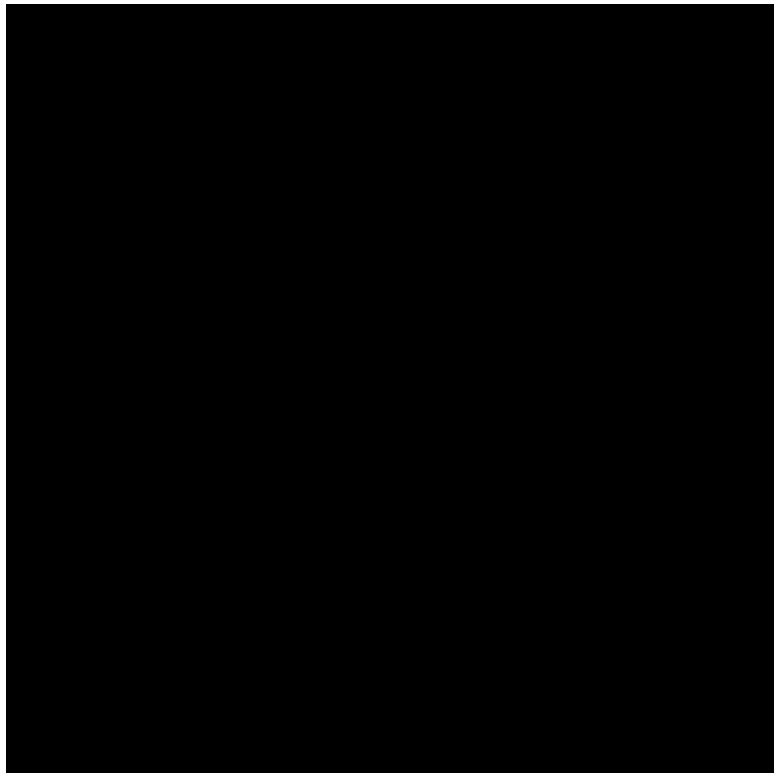
Initialize an AudioIn object

Start taking audio input

Route the audio input to the amplitude meter

Measure the amplitude

Normalized to [0, 1]



FFT Class

```
import processing.sound.*;

int bands = 512;
FFT fft = new FFT(this, bands);
AudioIn in = new AudioIn(this, 0);
float[] spectrum = new float[bands];

void setup() {
  size(512, 360);

  in.start();
  fft.input(in);
}

void draw() {
  background(255);

  fft.analyze(spectrum);

  for(int i = 0; i < bands; i++){
    line(i, height, i, height - spectrum[i] * height * 5);
  }
}
```

Import the Sound library

Initialize an FFT object

Initialize an AudioIn object

Initialize an array to store the spectrum

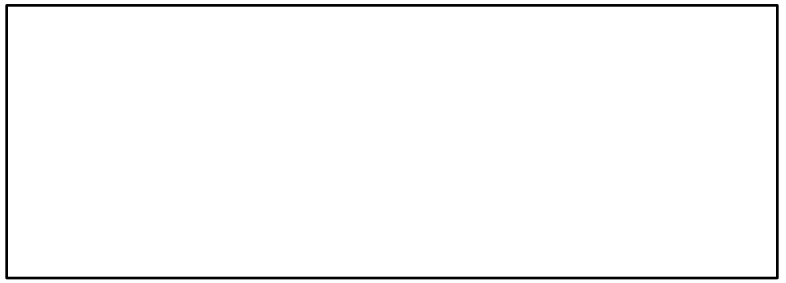
Start taking audio input

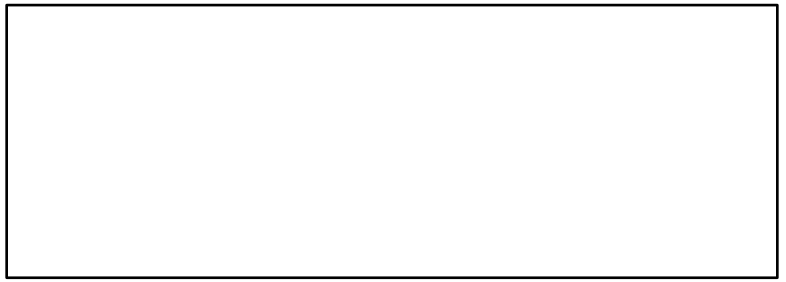
Route the audio input to the FFT analyzer

Specify the array to store the outputs

Run Fast Fourier Transform

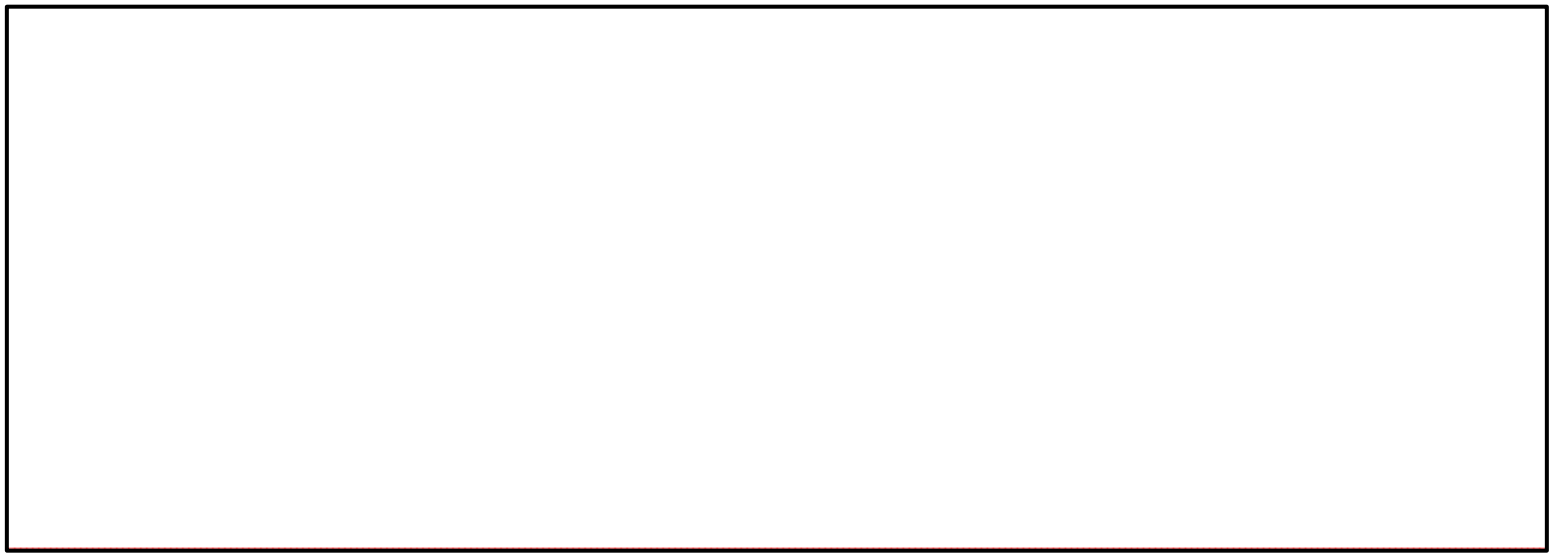
Normalized to [0, 1]





Homework 3: Spectrum Visualizer

- Modify the template code to implement a spectrum visualizer
- Instructions will be released on Gradescope
- Due at **11:59pm ET** on **September 23**
- Late submissions: **1 point deducted per day**



Deep Vision Library

- Deep learning-powered computer vision library that supports
  - Object **detection**
  - Object **recognition**
  - Object **segmentation**
  - **Keypoint detection**
  - **Depth estimation**
  - **Style transfer**
  - **Super-resolution**

github.com/cansik/deep-vision-processing

Review – Extensions

Processing on Different Platforms

• JavaScript	p5.js	p5js.org
• Python	processing.py	py.processing.org
• Android		android.processing.org
• Raspberry Pi		pi.processing.org

p5.js – Processing for Java Script

- Ideal for web programming
- Can be embedded on websites
- Slightly different syntax but same design philosophy
- Online editor: editor.p5js.org

The Communities

- p5.js community at p5js.org/community
- OpenProessing community at openprocessing.org

Discussions

- Why do we need computer art or music?
- Why do we want AI art or music?
- What’s the difference between a program and an AI algorithm?

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

That's It for the First Half of This Course!