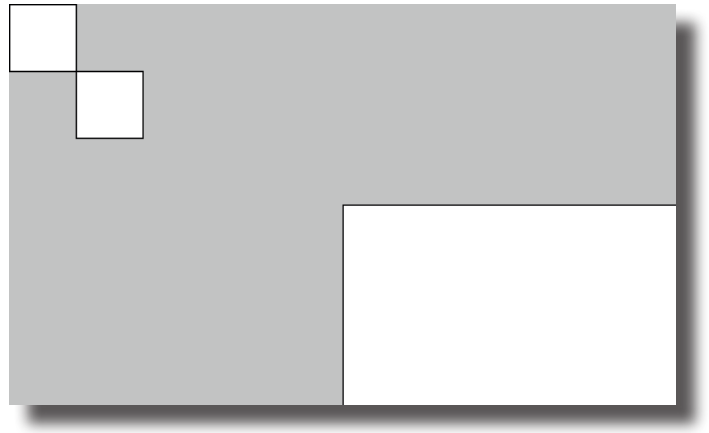


Processing - Beispiel 1

Basic1



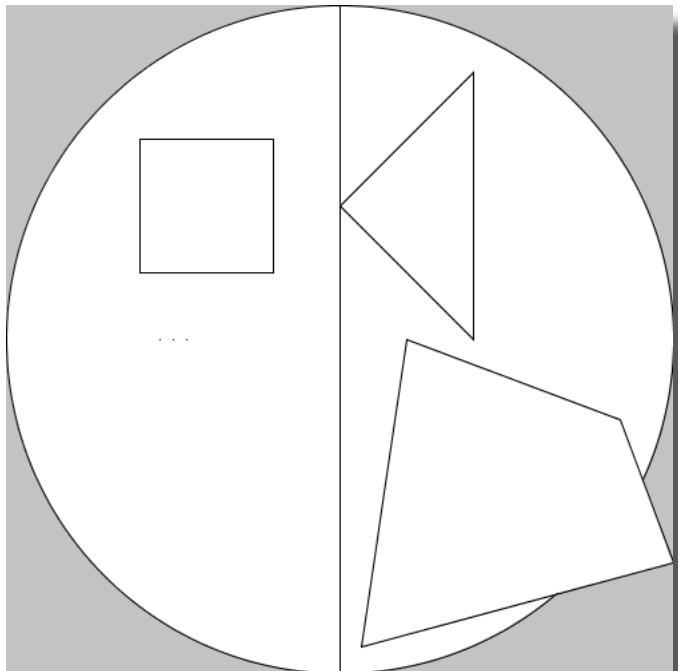
Code:

```
//My first Sketch
```

```
void setup() {  
  size(500, 300);  
  rect(0, 0, 50, 50);  
  rect(50, 50, 50, 50);  
  rect(width/2, height/2, width/2, height/2);  
}
```

Processing - Beispiel 2

Basic2

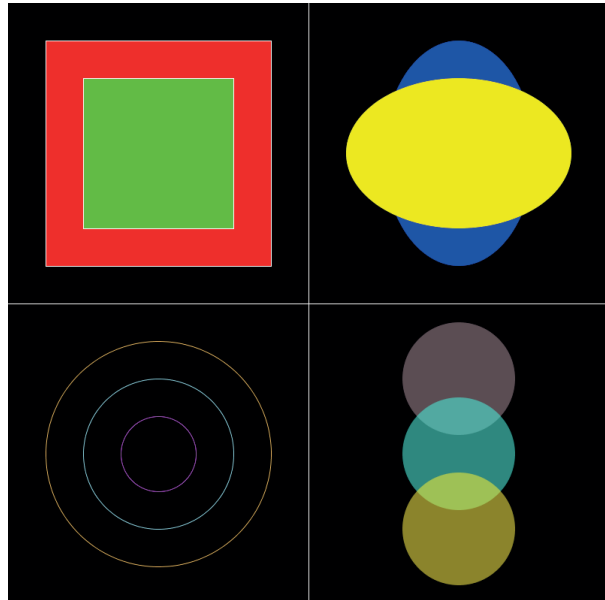


Code:

```
void setup() {  
  size(500, 500);  
  
  ellipse(width/2, height/2, 500, 500);  
  
  rect(100, 100, 100, 100);  
  
  line(250, 0, 250, 500);  
  
  point(115, 250);  
  point(125, 250);  
  point(135, 250);  
  
  triangle(250, 150, 350, 50, 350, 250);  
  
  quad(300, 250, 460, 310, 500, 417, 266, 480);  
}
```

Processing - Beispiel 3

Basic3



Code:

```
void setup() {

  size(800, 800);
  background(0, 0, 0);

  stroke(255, 255, 255);
  line(400, 0, 400, 800);
  line(0, 400, 800, 400);

  //oben links
  stroke(255, 255, 255);
  fill(255, 0, 0);
  rect(50, 50, 300, 300);
  fill(0, 255, 0);
  rect(100, 100, 200, 200);

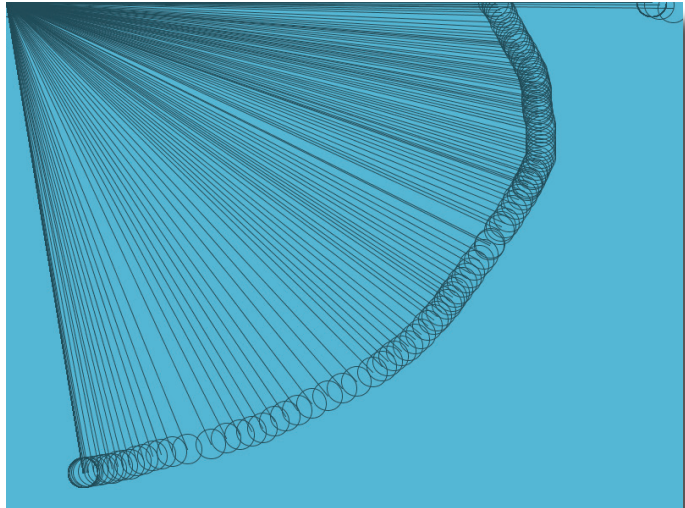
  //oben rechts
  noStroke();
  fill(0, 0, 255);
  ellipse(600, 200, 200, 300);
  fill(255, 255, 0);
  ellipse(600, 200, 300, 200);

  //unten links
  noFill();
  stroke(#E744FA);
  ellipse(200, 600, 100, 100);
  stroke(#9FDBEA);
  ellipse(200, 600, 200, 200);
  stroke(#F2BF5F);
  ellipse(200, 600, 300, 300);

  //unten rechts
  noStroke();
  fill(#FFD1DF, 100);
  ellipse(600, 500, 150, 150);
  fill(#50E3D5, 150);
  ellipse(600, 600, 150, 150);
  fill(#FCE724, 150);
  ellipse(600, 700, 150, 150);
}
```

Processing - Beispiel 4

Input1

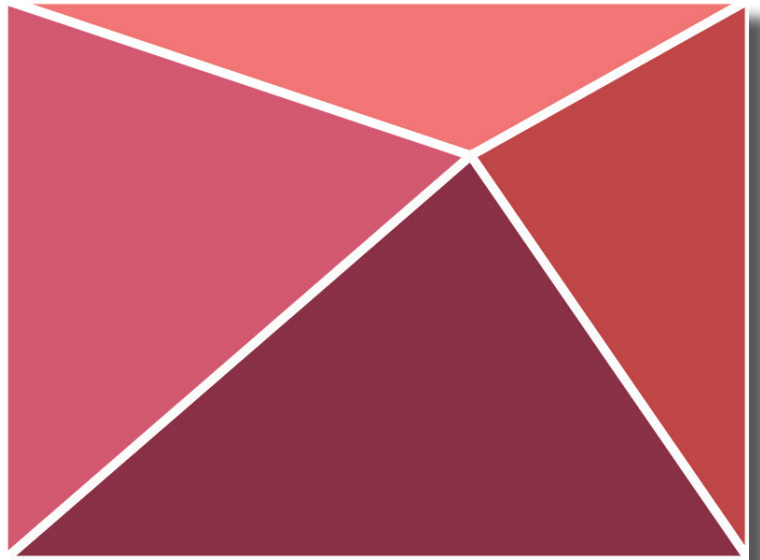


Code:

```
void setup() {  
  size (800, 600);  
  background(#62B8D8);  
}  
  
void draw() {  
  //background(#62B8D8);  
  
  stroke(#294D5A);  
  
  line(0, 0, mouseX, mouseY);  
  
  noFill();  
  
  ellipse(mouseX, mouseY, 35, 35);  
  
  println(mouseX+" "+mouseY);  
}
```

Processing - Beispiel 5

Input2



Code:

```
void setup() {  
  size (800, 600);  
  strokeWeight(10);  
  stroke(255, 255, 255);  
}  
  
void draw() {  
  background(255, 255, 255);  
  
  fill(#FA7272);  
  triangle(0, 0, width, 0, mouseX, mouseY);  
  
  fill(#C13E3E);  
  triangle(width, 0, width, height, mouseX, mouseY);  
  
  fill(#903043);  
  triangle(width, height, 0, height, mouseX, mouseY);  
  
  fill(#D3536D);  
  triangle(0, 0, 0, height, mouseX, mouseY);  
}
```

Input3

Code:

```
void setup() {
  size(700, 450);

  noStroke();
  strokeWeight(4);
  fill(0, 0, 0);
  rect(0, 0, width/2, height);
  fill(255, 255, 255);
  rect(width/2, 0, width/2, height);
}

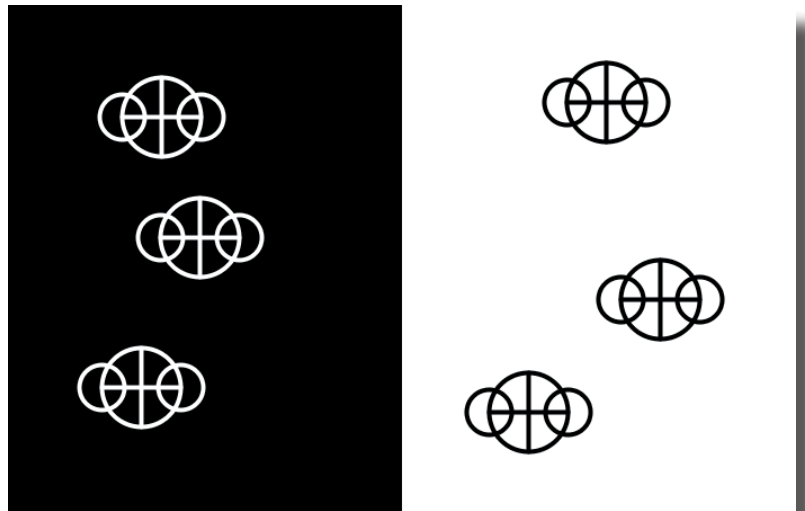
void draw() {
}

void mousePressed() {

  println(mouseX);
  noFill();

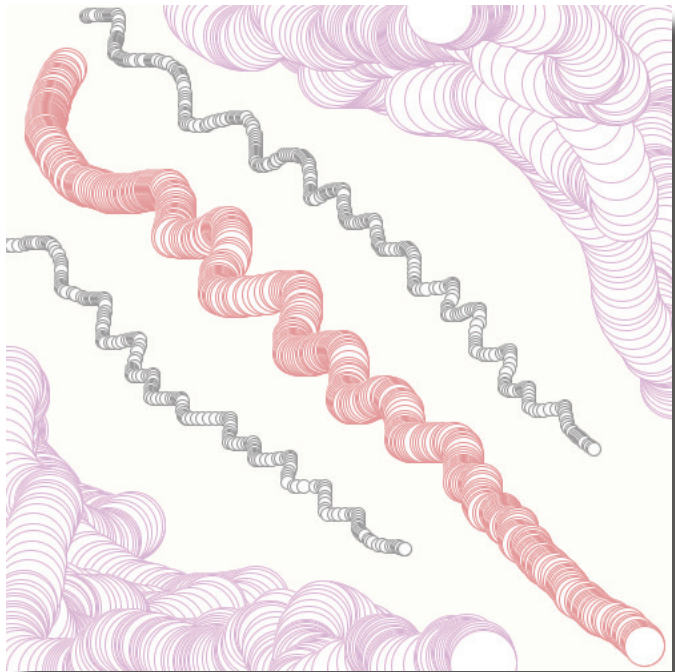
  if ( mouseX > width/2 ) {
    stroke(0, 0, 0);
  } else {
    stroke(255, 255, 255);
  }

  ellipse(mouseX, mouseY, 70, 70);
  line(mouseX-35, mouseY, mouseX+35, mouseY);
  line(mouseX, mouseY+35, mouseX, mouseY-35);
  ellipse(mouseX-35, mouseY, 40, 40);
  ellipse(mouseX+35, mouseY, 40, 40);
}
```



Processing - Beispiel 7

Input4



Code:

```
int mode;

void setup() {
  size(500, 500);
  background(#F7F5E6);

  mode = 1;
}

void draw() {
}

void mouseDragged() {
  if (mode == 1) {
    stroke(#A5A0A0);
    ellipse(mouseX, mouseY, 10, 10);
  }
  if (mode == 2) {
    stroke(#E59B9C);
    ellipse(mouseX, mouseY, 30, 30);
  }
  if (mode == 3) {
    stroke(#E8BFED);
    ellipse(mouseX, mouseY, 50, 50);
  }
}
```

```
void keyPressed() {
  if (key == '1') {
    mode = 1;
  }

  if (key == '2') {
    mode = 2;
  }

  if (key == '3') {
    mode = 3;
  }

  if (key == '4') {
    background(#F7F5E6);
  }

  if (key == '5') {
    saveFrame();
  }
}
```

Processing - Beispiel 8

Random1



Code:

```
float start;
float breite;
void setup() {
  size(800, 200);
  background(255, 255, 255);
  start = 0;

  noStroke();
}

void draw() {
}

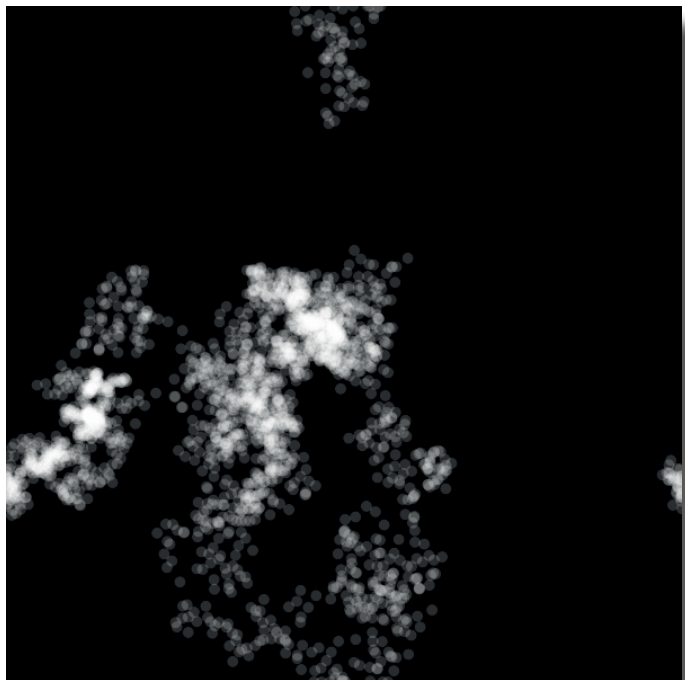
void keyPressed() {
  breite = random(150);
  fill(random(0, 255), random(100, 255), random(200, 255));
  rect(start, 0, breite, height);

  start += breite;

  if (start > width) {
    start = 0;
  }
}
```


Processing - Beispiel 9

Random2A



Code:

```
float positionX;
float positionY;
float punkt;
float tempo;

void setup() {
  size(600, 600);
  background(0, 0, 0);
  positionX = width/2;
  positionY = height/2;
  punkt = 10;
  tempo = 4.0;
  noStroke();
}

void draw() {

  fill(255, 255, 255, 50);
  ellipse(positionX, positionY, punkt, punkt);

  positionX += tempo*random(-1, 1) ;
  positionY += tempo*random(-1, 1) ;

  //rechts
  if (positionX > width) {
    positionX = width;
  }
}
```

```

//links
if (positionX < 0) {
    positionX = 0;
}

//unten
if (positionY > height) {
    positionY = height;
}

//oben
if (positionY < 0) {
    positionY = 0;
}
}

void keyPressed() {

    if (key == '1') {
        tempo = 4.0;
    }

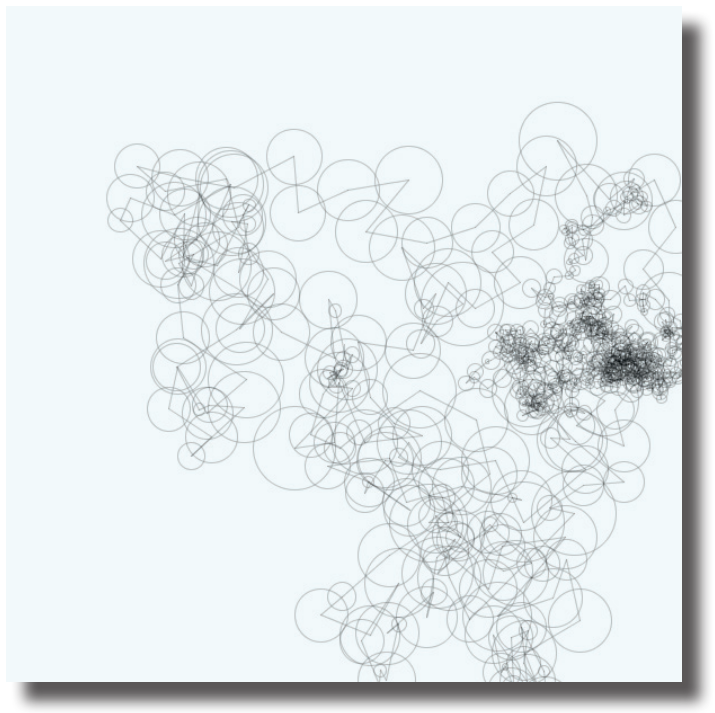
    if (key == '2') {
        tempo = 8.0;
    }

    if (key == '3') {
        tempo = 16.0;
    }
}

```

Processing - Beispiel 10

Random2B



Code:

```
float posXold;  
float posYold;  
float posXnew;  
float posYnew;  
float punkt;  
float tempo;  
float distance;  
  
void setup() {  
  size(600, 600);  
  background(#F5FAFC);  
  posXnew = width/2;  
  posYnew = height/2;  
  punkt = 10;  
  tempo = 38.0;  
  frameRate(10);  
  noFill();  
}  
  
void draw() {  
  
  stroke(0, 0, 0, 50);  
  
  posXold = posXnew;  
  posYold = posYnew;  
  
  posXnew += tempo*random(-1, 1) ;  
  posYnew += tempo*random(-1, 1) ;
```

```

distance = dist(posXold, posYold, posXnew, posYnew);
println(distance);

ellipse(posXnew, posYnew, distance, distance);
line(posXold, posYold, posXnew, posYnew);

//rechts
if (posXnew > width) {
    posXnew = width;
}

//links
if (posXnew < 0) {
    posXnew = 0;
}

//unten
if (posYnew > height) {
    posYnew = height;
}

//oben
if (posYnew < 0) {
    posYnew = 0;
}
}

void keyPressed() {

    if (key == '1') {
        tempo = 14.0;
    }

    if (key == '2') {
        tempo = 38.0;
    }

    if (key == '3') {
        tempo = 56.0;
    }
}

```

Processing - Beispiel 11

Variable1



Code:

```
int lineA;
float lineB;

void setup() {
  size(800, 200);
  background(0, 0, 0);

  lineA = 0;
  lineB = 0;
}

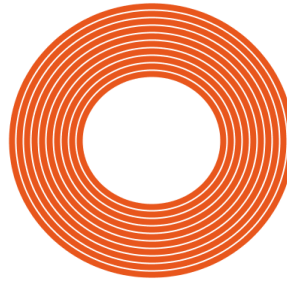
void draw() {
  background(0, 0, 0);
  rect(0, 0, lineA, height/2);
  rect(0, height/2, lineB, height/2);
  lineA = lineA + 1;
  lineB = lineB + 0.3;

  if (lineA > width) {
    lineA = 0;
  }

  if (lineB > width) {
    lineB = 0;
  }
}
```

Processing - Beispiel 12

ForLoop1



Code:

```
float kreis;

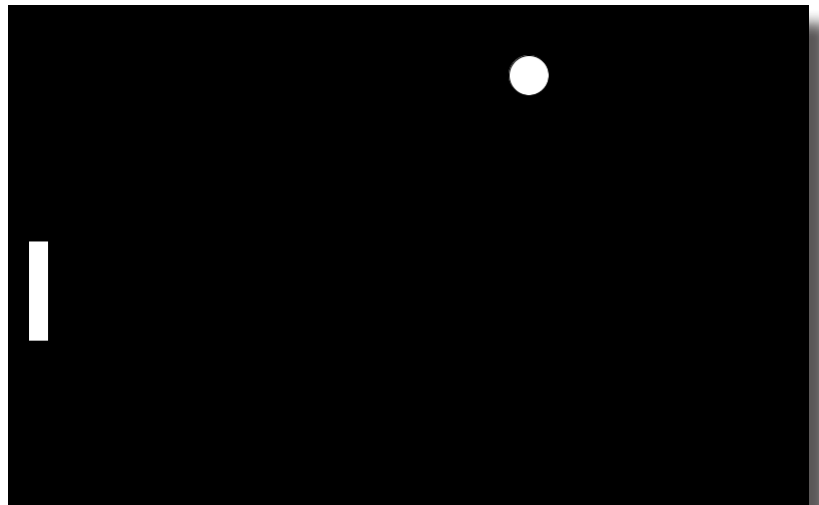
void setup() {
  size(600, 600);
  noFill();
  stroke(#E8561C);
  strokeWeight(8);

  kreis = 20;
}

void draw() {
  background(255, 255, 255);
  for (int i = 0; i < 10; i++) {
    ellipse(width/2, height/2, kreis*i+mouseX, kreis*i+mouseY );
  }
}
```

Processing - Pong Spiel

Pong1



Code:

```
//Pong Balken width und height
float balkenH;
float balkenW;
//pong Balken Position
float balkenPosX;
float balkenPosY;
//Ball Position
float ballX;
float ballY;
//Ball Geschwindigkeit
float ballSpeedX;
float ballSpeedY;
//Ball Size
float ballSize;
//Der Punkt an dem der Ball den Balken berührt
float contact;

void setup() {
  size(800, 500);
  noStroke();
  fill(255, 255, 255);

  balkenH = 100;
  balkenW = 20;

  balkenPosX = 20;

  ballX = width/2;
  ballY = height/2;
```

```

    ballSpeedX = 4.0;
    ballSpeedY = 4.0;

    ballSize = 40;
}

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

    rect(balkenPosX, balkenPosY,    balkenW, balkenH);

    ellipse(ballX, ballY, ballSize, ballSize);

    ballX += ballSpeedX;
    ballY += ballSpeedY;

    balkenPosY = mouseY-balkenH/2;

    //linke Seite
    if (ballX < balkenPosX) {
        ballX = width/2;
        ballY = height/2;
        ballSpeedX *=-1;
    }

    //rechte Seite
    if (ballX > width-ballSize/2) {
        ballSpeedX *= -1;
    }

    //obere Seite
    if (ballY < 0+ballSize/2) {
        ballSpeedY *= -1;
    }

    //untere Seite
    if (ballY > height-ballSize/2) {
        ballSpeedY *= -1;
    }

    //Pong Balken
    contact = balkenPosX+balkenW+(ballSize/2);
    if ( (ballY > balkenPosY) && (ballY < balkenPosY+balkenH) && (ballX < contact) ) {
        ballSpeedX *= -1;
        ballX = contact;
    }
}

```


Pong2

Code:

```
//Pong Balken width und height
float balkenH;
float balkenW;
//Pong Balken Tempo
float balkenSpeed;
//pong Balken Position
float balkenPosX;
float balkenPosY;
//Ball Position
float ballX;
float ballY;
//Ball Geschwindigkeit
float ballSpeedX;
float ballSpeedY;
//Ball Size
float ballSize;
//Der Punkt an dem der Ball den Balken berührt
float contact;

void setup() {
  size(800, 500);
  noStroke();
  fill(255, 255, 255);

  balkenH = 100;
  balkenW = 20;

  balkenSpeed = 6.5;
  balkenPosX = 20;

  ballX = width/2;
  ballY = height/2;

  ballSpeedX = 5.0;
  ballSpeedY = 5.0;

  ballSize = 40;
}

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

  rect(balkenPosX, balkenPosY, balkenW, balkenH);

  ellipse(ballX, ballY, ballSize, ballSize);

  ballX += ballSpeedX;
  ballY += ballSpeedY;

  // balkenPosY = mouseY-balkenH/2;

  if (ballSize > 8) {

    if (balkenPosY < mouseY-balkenH/2) {
      balkenPosY += balkenSpeed;
    }
  }
```

```

    if (balkenPosY > mouseY-balkenH/2) {
        balkenPosY -= balkenSpeed;
    }

    //linke Seite
    if (ballX < balkenPosX) {
        ballX = width/2;
        ballY = height/2;
        ballSpeedX *=-1;

        ballSpeedX *= 1.1;
        ballSpeedY *= 1.1;
        balkenH *= 0.9;
        ballSize *= 0.9;
    }

    //rechte Seite
    if (ballX > width-ballSize/2) {
        ballSpeedX *= -1;
        ballSpeedY += random(-0.25, 0.25);
    }

    //obere Seite
    if (ballY < 0+ballSize/2) {
        ballSpeedY *= -1;
        ballSpeedX += random(-0.25, 0.25);
    }

    //untere Seite
    if (ballY > height-ballSize/2) {
        ballSpeedY *= -1;
        ballSpeedX += random(-0.25, 0.25);
    }

    //Pong Balken
    contact = balkenPosX+balkenW+(ballSize/2);
    if ( (ballY > balkenPosY) && (ballY < balkenPosY+balkenH) && (ballX < contact) ) {
        ballSpeedX *= -1;
        ballSpeedY += random(-0.25, 0.25);
        ballX = contact;
    }
} else {
    background(255, 255, 255);
}
}

```

Processing

Pong3

```
import ddf.minim.*;
Minim minim;
AudioSample kick;
AudioSample snare;

//Pong Balken width und height
float balkenH;
float balkenW;
//Pong Balken Tempo
float balkenSpeed;
//pong Balken Position
float balkenPosX;
float balkenPosY;
//Ball Position
float ballX;
float ballY;
//Ball Geschwindigkeit
float ballSpeedX;
float ballSpeedY;
//Ball Size
float ballSize;
//Der Punkt an dem der Ball den Balken berührt
float contact;

void setup() {
  size(800, 500);

  //sound
  minim = new Minim(this);
  kick = minim.loadSample( "BD.mp3", 512);
  snare = minim.loadSample("SD.wav", 512);

  noStroke();
  fill(255, 255, 255);

  balkenH = 100;
  balkenW = 20;

  balkenSpeed = 6.5;
  balkenPosX = 20;

  ballX = width/2;
  ballY = height/2;

  ballSpeedX = 5.0;
  ballSpeedY = 5.0;

  ballSize = 40;
}

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

  rect(balkenPosX, balkenPosY, balkenW, balkenH);
  ellipse(ballX, ballY, ballSize, ballSize);
```

```

ballX += ballSpeedX;
ballY += ballSpeedY;

// balkenPosY = mouseY-balkenH/2;

if (ballSize > 8) {

    if (balkenPosY < mouseY-balkenH/2) {
        balkenPosY += balkenSpeed;
    }

    if (balkenPosY > mouseY-balkenH/2) {
        balkenPosY -= balkenSpeed;
    }

    //linke Seite
    if (ballX < balkenPosX) {
        ballX = width/2;
        ballY = height/2;
        ballSpeedX *=-1;

        ballSpeedX *= 1.1;
        ballSpeedY *= 1.1;
        balkenH *= 0.9;
        ballSize *= 0.9;
        kick.trigger();
    }
    //rechte Seite
    if (ballX > width-ballSize/2) {
        ballSpeedX *= -1;
        ballSpeedY += random(-0.25, 0.25);
        snare.trigger();
    }
    //obere Seite
    if (ballY < 0+ballSize/2) {
        ballSpeedY *= -1;
        ballSpeedX += random(-0.25, 0.25);
        snare.trigger();
    }
    //untere Seite
    if (ballY > height-ballSize/2) {
        ballSpeedY *= -1;
        ballSpeedX += random(-0.25, 0.25);
        snare.trigger();
    }

    //Pong Balken
    contact = balkenPosX+balkenW+(ballSize/2);
    if ( (ballY > balkenPosY) && (ballY < balkenPosY+balkenH) && (ballX < contact) ) {
        ballSpeedX *= -1;
        ballSpeedY += random(-0.25, 0.25);
        ballX = contact;
        snare.trigger();
    }
} else {
    background(255, 255, 255);
}
}

```

Processing - Pong Spiel

Pong4



Code:

```
//Global

PFont font;
int score;
float fillColor;

//in void setup

font = loadFont("font.vlw");
textFont(font, 32);
score = 0;
fillColor = 0;

// ganz oben in void draw

// background(0, 0, 0);

if (fillColor > 0) {
  fillColor-=5;
}

fill(fillColor, 50);
rect(0, 0, width, height);

//in if linke Seite
fillColor = 255;

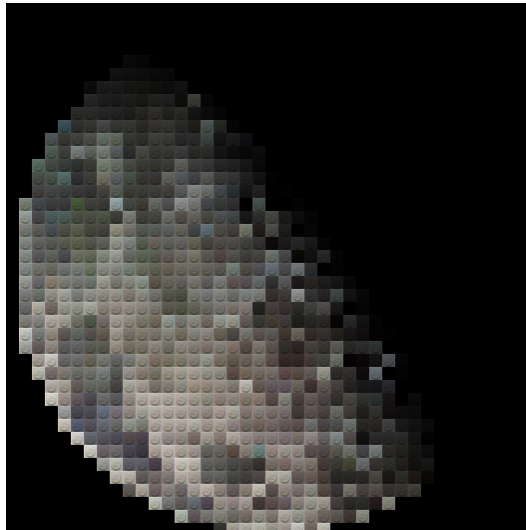
//in if pong balken
score++;

//nach, if (ballSize > 8)
  text("Score: "+score, width-200, 40);

//am Ende in else
fill(0, 0, 0);
textAlign(CENTER);
text("Game Over", width/2, height/2);
```

Processing - Beispiel 13

NestedLoop

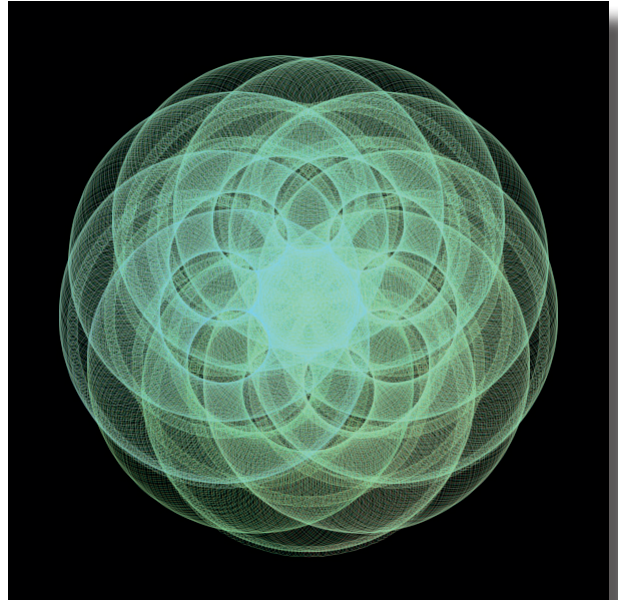


Code:

```
PImage lego;  
PImage foto;  
  
void setup() {  
  lego = loadImage("lego.png");  
  foto = loadImage("image.jpg");  
  
  size(foto.width, foto.height);  
  
  for (int x = 0; x < width; x+=16) {  
    for (int y = 0; y < height; y+=16) {  
      color c = foto.get(x,y);  
      tint(c);  
      image(lego, x, y, 16, 16);  
    }  
  }  
}  
  
void draw() {  
}  
  
void keyPressed(){  
  saveFrame("bild.png");  
}
```

Processing - Bonus

sine_cosine



Code:

```
float x;  
float y;  
  
float x2;  
float y2;  
  
float counter;  
  
void setup() {  
  size(600, 600);  
  colorMode(HSB, 360, 100, 100);  
  noStroke();  
  noFill();  
  background(0, 0, 0);  
}  
  
void draw() {  
  //background(0, 0, 100);  
  //fill(counter,100,100,50);  
  stroke(counter, 44, 100, 40);  
  
  x = (sin(counter)*100)+ 300 + (sin(counter/10)*100);  
  y = (cos(counter)*100)+ 300 + (cos(counter/10)*100);  
  
  ellipse(x, y, 10, 10);
```

```
x2 = sin(counter/2)*50;
y2 = cos(counter/2)*50;

x2 += x;
y2 += y;

stroke(counter*1.2, 44, 100, 40);
ellipse(x, y, 100, 100);
ellipse(x2, y2, 10, 10);

counter+=0.05;
}

void keyPressed() {

  if (key == 's') {
    saveFrame();
  }
}
```