

2017年度情報メディア基盤ユニット
6月20日講義内で作ったサンプルプログラム

その1

//2次元配列の例、オセロ板を描く

```
int[][] board;
final int empty = 0; // constant,定数
final int black = 1;
final int white = 2;
final int side = 50;

void setup(){
  size(400,400);
  board = new int[8][8];
  for(int y = 0;y < board.length;y++){
    for(int x = 0;x < 8;x++){
      board[y][x] = empty;
    }
  }
  board[3][3] = black;
  board[4][4] = black;
  board[3][4] = white;
  board[4][3] = white;
}

void draw(){
  stroke(0);
  fill(10,230,10);
  for(int y=0;y < 8;y++){
    for(int x=0;x < 8;x++){
      rect(x*side,y*side,side,side);
    }
  }

  for(int y=0;y < 8;y++){
    for(int x=0;x < 8;x++){
      if(board[y][x] == black){
        fill(0);
        ellipse(x*side+side/2,y*side+side/2,0.8*side,0.8*side);
      }else if(board[y][x] == white){
        fill(255);
        ellipse(x*side+side/2,y*side+side/2,0.8*side,0.8*side);
      }
    }
  }
}

void mouseClicked(){
  int x = mouseX / side;
  int y = mouseY / side;
  if(mouseButton == LEFT){
    board[y][x] = black;
  }else if(mouseButton == RIGHT){
```

```
    board[y][x] = white;
  }
}
```

その2

//最大値を求める関数の例

```
float[] v;

float myMax(float[] a) {
  float maxSoFar;
  maxSoFar = a[0];
  for (int i=0; i < a.length; i++) {
    if (a[i] > maxSoFar) {
      maxSoFar = a[i];
    }
  }
  return maxSoFar;
}

void setup() {
  size(400, 400);
  v = new float[5];
  v[0] = 100;
  v[1] = 150;
  v[2] = 50;
  v[3] = 120;
  v[4] = 90;
}

void draw() {
  background(255);
  fill(0, 32, 64);
  float r = myMax(v);
  ellipse(width/2, height/2, 2*r, 2*r);
}
```

その3

//跳ね返り処理のサンプル

```
float xPos;
float yPos;
float vx;
float vy;
int radius;

void bounce(float nx, float ny) {
  float a = vx*nx+vy*ny;
  float rx = vx - 2 * a * nx;
  float ry = vy - 2 * a * ny;
  vx = rx;
  vy = ry;
}

void move(){
```

```

    xPos += vx;
    yPos += vy;
}

void setup(){
    size(400,400);
    float theta = random(2*PI);
    vx = cos(theta);
    vy = sin(theta);
    xPos = width/2;
    yPos = height/2;
    radius = 10;
}

void draw(){
    background(255);
    move();
    if(yPos < radius){
        bounce(0,1);
        yPos = radius;
    }else if(yPos + radius >= height){
        bounce(0,-1);
        yPos = height - radius;
    }else if(xPos < radius){
        bounce(1,0);
        xPos = radius;
    }else if(xPos+radius > width){
        bounce(-1,0);
        xPos = width-radius;
    }
    fill(0,32,64);
    ellipse(xPos,yPos,2*radius,2*radius);
}

```

その4

```

//
final int empty = 0;
final int black = 1;
final int white = 2;
final int side = 50;

int[][] board;

void setup() {
    size(400, 400);
    board = new int[8][8];
    for(int y=0;y<8;y++){
        for(int x=0;x<8;x++){
            board[y][x] = empty;
        }
    }
    board[3][3] = black;
    board[4][4] = black;
    board[3][4] = white;
}

```

```

    board[4][3] = white;
}

void draw() {
    stroke(0);
    fill(10, 230, 10);
    for (int y=0; y < 8; y++) {
        for (int x=0; x < 8; x++) {
            rect(x*side, y*side, side, side);
        }
    }
    for (int y=0; y < 8; y++) {
        for (int x=0; x < 8; x++) {
            if (board[y][x] == black) {
                fill(0);
                ellipse(x*side+side/2, y*side+side/2, 0.8*side, 0.8*side);
            } else if (board[y][x] == white) {
                fill(255);
                ellipse(x*side+side/2, y*side+side/2, 0.8*side, 0.8*side);
            }
        }
    }
}
}

```

その5

```

//
float a = random(10);
float b = random(10);

if(a > b){
    println("a:"+a);
}else{
    println("b:"+b);
}

float max2(float a,float b){
    if(a > b){
        return a;
    }else{
        return b;
    }
}

```

その6

```

//
float A = random(10);
float B = random(10);
float C = random(10); //<>// //<>//
if(A>=B && B >= C){
    println(A);
}else if(A>=C && C >= B){
    println(A);
} else if(B>=A && A >= C){

```

```
    println(B);
} else if(B>=C && C >= A){
    println(B);
} else if(C>=A && A >= B){
    println(C);
} else if(C>=B && B >= A){
    println(C);
}
```

その7

```
//
float a = random(10); //<>//
float b = random(10);
float c = random(10);
float maxSoFar;

if(a > b){
    maxSoFar = a;
}else{
    maxSoFar = b;
}
if(c > maxSoFar){
    maxSoFar = c;
}

println(maxSoFar);float a = random(10);
```

その8

```
//
float b = random(10);
float c = random(10);
float maxSoFar;

maxSoFar = a;
if(b > maxSoFar){
    maxSoFar = b;
}
if(c > maxSoFar){
    maxSoFar = c;
}

println(maxSoFar);
```

その9

```
//
float a = random(10);
float b = random(10);
float c = random(10);
float d = random(10);
float maxSoFar; //<>//

maxSoFar = a;
```

```

if(b > maxSoFar){
    maxSoFar = b;
}
if(c > maxSoFar){
    maxSoFar = c;
}
if(d > maxSoFar){
    maxSoFar = d;
}
println(maxSoFar);

```

その 10

```

//
float[] a;
a = new float[5];
for(int i=0;i<5;i++){
    a[i] = random(10);
}

float maxSoFar; //<>//
maxSoFar = a[0];
for(int i=0;i<5;i++){
    if(a[i] > maxSoFar){
        maxSoFar = a[i];
    }
}
println(maxSoFar);

```

その 11

```

//
float[] a;
a = new float[5];
for(int i=0;i<5;i++){
    a[i] = random(10);
}
println(max(a));

```

その 12

```

//
final int empty = 0;
final int black = 1;
final int white = 2;
final int side = 50;

int[][] board = {{empty, empty, empty, empty, empty, empty, empty,
empty},
    {empty, empty, empty, empty, empty, empty, empty, empty},
    {empty, empty, empty, black, white, empty, empty, empty},
    {empty, empty, empty, white, black, empty, empty, empty},
    {empty, empty, empty, empty, empty, empty, empty, empty},
    {empty, empty, empty, empty, empty, empty, empty, empty},

```

```

    {empty, empty, empty, empty, empty, empty, empty, empty}}};

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

void draw() {
    stroke(0);
    fill(10, 230, 10);
    for (int y=0; y < 8; y++) {
        for (int x=0; x < 8; x++) {
            rect(x*side, y*side, side, side);
        }
    }
    for (int y=0; y < 8; y++) {
        for (int x=0; x < 8; x++) {
            if (board[y][x] == black) {
                fill(0);
                ellipse(x*side+side/2, y*side+side/2, 0.8*side, 0.8*side);
            } else if (board[y][x] == white) {
                fill(255);
                ellipse(x*side+side/2, y*side+side/2, 0.8*side, 0.8*side);
            }
        }
    }
}

void mouseClicked(){
    int x = mouseX / side;
    int y = mouseY / side;
    if(mouseButton == LEFT){
        board[y][x] = black;
    }else if(mouseButton == RIGHT){
        board[y][x] = white;
    }
}
}

```

その 13

```

//
float[][] x;
x = new float[100][100];
float[][][] y;
y = new float[100][100][100];
float[][][][] z;
z = new float[100][100][100][100];float xPos;

```

その 14

```

// 跳ね返りのサンプル
float yPos;
float vx;
float vy;
float radius = 10;

```

```

void setup() {
  size(400, 400);
  xPos = width/2;
  yPos = height/2;
  float theta = random(2*PI);
  vx = cos(theta);
  vy = sin(theta);
}

// the vector (nx,ny) must be an unit vector.
void bounce(float nx, float ny) {
  float a = vx*nx+vy*ny;
  float rx = vx - 2 * a * nx;
  float ry = vy - 2 * a * ny;
  vx = rx;
  vy = ry;
}

void move() {
  xPos += vx;
  yPos += vy;
}

void draw() {
  background(255);
  fill(0, 32, 64);
  move();
  if (yPos < radius) {
    bounce(0, 1);
    yPos = radius;
  } else if (yPos + radius >= height) {
    bounce(0, -1);
    yPos = height - radius;
  } else if (xPos < radius) {
    bounce(1, 0);
    xPos = radius;
  } else if (xPos+radius > width) {
    bounce(-1, 0);
    xPos = width-radius;
  }
  ellipse(xPos, yPos, 2*radius, 2*radius);
}

void mouseClicked() {
  xPos = width/2;
  yPos = height/2;
  float theta = random(2*PI);
  vx = cos(theta);
  vy = sin(theta);
}

```