

미디어애펀테크 튜터링

4회차

미디어경영학과 2022-01

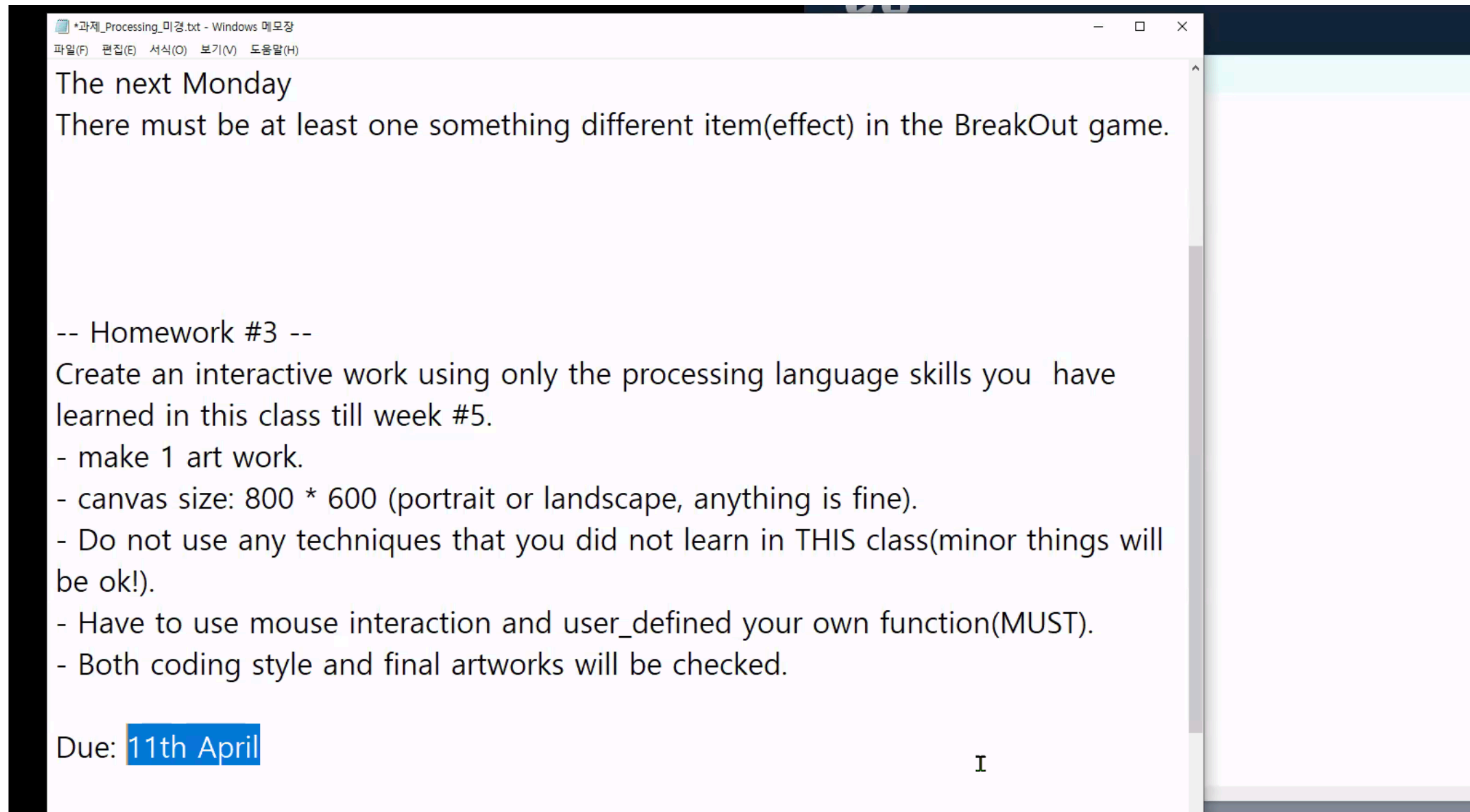
튜터링 일정

4회차

일정	내용
1회차	OT, Processing 소개, 선,도형 다루기, 변수, 연산자,반복문
2회차	동적 프로그래밍, 논리 연산자, if문 표정그리기, 공 움직이기 , x축 충돌처리
3회차	Y축 충돌처리,pad 만들기, 2중 for 문, 배열 및 2차원 배열, 벽돌깨기 완성
4회차	“create an interactive work”
5회차	함수, 벽돌깨기 리팩토링, while문, 격자 무늬 만들기, bar chart 만들기, list 활용 dist 활용
6회차	???
7회차	???
8회차	???
9회차	???
10회차	???

과제 설명

create an interactive work



18일 까지

5주차 까지 배운것들

create an interactive work

- `size(w,h)`, `stroke(rgb)`, `strokeWeight(a)`, `noStroke()`, `fill(rgb)`
- `line(x1,y1,x2,y2)`, `ellipse(x,y,w,h)`, `triangle(x1,y1,x2,y2,x3,y3)`
- 변수,관계 및 논리 연산자, 증감 연산자, for문, while문
- `setup()`, `draw()`, if else if else, 1차원배열, 2차원배열
- 함수, `mousePressed`, `mouseX`, `mouseY`, `random`, `frameRate(a)`

함수란?

create an interactive work

- 기능들의 집합입니다.
- 필요할 때 호출해서 사용합니다.
- 변수는 값을 담아두고 필요할때 가져다 쓴다면
- 함수는 기능들을 묶어두고 필요할 때 호출해서 사용합니다.

함수란?

create an interactive work

- void setup()
- void draw()
- void drawRect()
- 가 함수입니다.
- {} 안에 기능들이 묶여져있기 때문입니다.
- drawRect 함수 같은 경우 필요할 때
 - 호출을 하고 있습니다.

```
sketch_220413a ▼
1 void setup() {
2   size(800,600);
3 }
4
5 void draw() {
6   drawRect();
7 }
8
9 void drawRect() {
10   rect(100,100,150,150);
11 }
12
```

함수란?

create an interactive work

- 둘이 같은 코드지만, 오른쪽은 함수를 사용하고 왼쪽은 사용하지 않았습니다.
- draw 입장에서 봤을때 왼쪽 코드가 더 명확 해 보입니다.

```
sketch_220413a
1 void setup() {
2   size(800,600);
3 }
4
5 void draw() {
6   drawRect();
7 }
8
9 void drawRect() {
10   rect(100,100,150,150);
11 }
```

```
sketch_220413a | Processing
1 void setup() {
2   size(800,600);
3 }
4
5 void draw() {
6   rect(100,100,150,150);
7 }
8
```

함수란?

create an interactive work

```
4
5 void draw() {
6     drawYelloRect();
7
8     drawRedRect();
9
10    drawBlueRect();
11 }
12
13 void drawYelloRect() {
14     fill(#FFFF00);
15     rect(100,100,100,100 );
16 }
17
18 void drawRedRect() {
19     fill(#FF0000);
20     rect(200,100,100,100 );
21 }
22
23 void drawBlueRect() {
24     fill(#0000FF);
25     rect(300,100,100,100 );
26 }
```

```
4
5 void draw() {
6
7     fill(#FFFF00);
8     rect(100,100,100,100 );
9
10    fill(#FF0000);
11    rect(200,100,100,100 );
12
13    fill(#0000FF);
14    rect(300,100,100,100 );
15 }
16
17
18
19
20
21
22
```


함수(고급)

create an interactive work

- 함수에 값을 전달할 수 있습니다.
- 중복된 내용이 사라지고, 재 사용성이 높은 함수를 작성하게 됩니다.

```
4
5 void draw() {
6     drawYelloRect();
7
8     drawRedRect();
9
10    drawBlueRect();
11 }
12
13 void drawYelloRect() {
14     fill(#FFFF00);
15     rect(100,100,100,100 );
16 }
17
18 void drawRedRect() {
19     fill(#FF0000);
20     rect(200,100,100,100 );
21 }
22
23 void drawBlueRect() {
24     fill(#0000FF);
25     rect(300,100,100,100 );
26 }
```

```
4
5 void draw() {
6     drawRect(100,#FFFF00);
7     drawRect(200,#FF0000);
8     drawRect(300,#0000FF);
9 }
10
11 void drawRect(int x, color rectColor) {
12     fill(rectColor);
13     rect(x,100,100,100);
14 }
15
```

함수란?

create an interactive work

- `void 내가만든함수() {`
- `ellipse(100,100,100,100)`
- `}`
- `내가만든함수();` \leq 동그라미가 그려짐
- `void 두번째함수(int x, int y,) {`
- `ellipse(x,y,100,100)`
- `}`
- `두번째함수(200,200);` \leq 전달한 200,200 좌표에 동그라미가 그려짐

mouse interaction

create an interactive work

```
5 |  
6 void draw() {  
7  
8     background(120);  
9  
10    if(mousePressed) {  
11        fill(random(255), random(255), random(0255));  
12        stroke(random(255));  
13        strokeWeight(random(20));  
14        ellipse(random(800), random(600), random(800), random(600));  
15    }  
16 }  
17
```

mouse interaction

create an interactive work

```
5  
6 void draw() {  
7  
8     if(mousePressed) {  
9         line(mouseX,mouseY,pmouseX, pmouseY);  
10    }  
11 }
```

frameRate(rate)

create an interactive work

- draw 함수 안에서 그리는 순 간 동영상을 그리는 것 입니다.
 - 즉 프레임이라는 개념이 들어가게 됩니다.
- 여기서 프레임은 움직이는 화면에서 1장 1장을 뜻합니다. 단위라고 볼 수 있습니다.
- frameRate 함수는 1초에 몇장을 그릴 건지 결정합니다.
 - frameRate(1)
 - 1초에 1장만 그림
 - frameRate(10)
 - 1초에 10장만 그림
- 프레임이 높을 수록 자연스러운 영상이되고, 낮을수록 끊기는 영상이 됩니다.

VerticalLines

create an interactive work

```
virtical_lines ▼
1 void setup() {
2   size(800,600);
3   frameRate(1);
4 }
5
6 void draw(){
7   background(255);
8
9   /* 정의 된 verticalLines 함수를 호출함 */
10  /* 0부터 511 까지의 랜덤 숫자를 x1,y1,x2,y2에게 전달. */
11  verticalLines(random(512),random(512),random(512),random(512));
12 }
13
14 /* */
15 void verticalLines(float x1,float y1,float x2,float y2) {
16
17   float x = x1;
18
19   while(x < x2) { // x1의 값을 받은 x가 x2보다 작아지는 동안에,
20
21     line(x,y1,x,y2); // 일직선의 선을 그린다.
22                     // 선이 시작하는 x축과 끝나는 x축이 같으니깐
23                     // 일직선이 나온다.
24
25     x += 10.0;      // x를 10씩 증가시킵니다.
26                     // 그래야지 x2보다 작아질 때 까지 선을 그릴 수 있습니다.
27   }
28 }
```

Random

create an interactive work

```
random ▾
1 void setup(){
2   size(600, 600);
3   stroke(128);
4 }
5
6 void draw(){
7   int lines = 0;
8
9   background(0);
10
11   /* 최대 10개의 선을 만듭니다. */
12   while(lines < 10){
13     /* 시작지점이 mouseX 와 mouseY 니깐, 마우스포인터로부터 선을 그립니다 */
14     /* 선이 어디까지 그려질지는 모릅니다.왜냐하면 선의 끝점은 random으로 만들어지기 때문입니다.*/
15     line(mouseX, mouseY, random(width), random(height));
16     lines++;
17   }
18 }
```

Nested-for-loop

create an interactive work

```
random ▼
1 void setup() {
2   size(600, 600);
3
4   /* 1초에 1장씩 영상을 그립니다. */
5   frameRate(1);
6 }
7
8 void draw() {
9
10  /* 아래는 2중 for문 입니다. */
11
12  /* 첫번째 for문은 20번 반복 할 것이고 */
13  /* 두번째 for문도 20번 반복 할 것입니다. */
14
15  /* 첫번째 for문이 1번 실행될때마다 두번째 for문은 20번씩 반복하게 됩니다. */
16  /* 즉, 20x20. 총 400번 반복하게 됩니다.*/
17
18  for(int x=0; x<width; x+=30){
19    for (int y=0; y<height; y+=30){
20
21      /* 색을 랜덤으로 채우면서 */
22      fill(random(255), random(255), random(255));
23      /* 사각형을 하나씩 그립니다. */
24      /* for 문에서 x 와 y 좌표를 30씩 증가시키고 있고 */
25      /* 그 좌표로 사각형을 그립니다. */
26
27      /* 너비와 높이는 30으로 그리고 있는데, 좌표를 30씩 움직였으니깐 겹칠일 이 없습니다. */
28      /* 사각형은 400개가 그려지게 됩니다.*/
29      rect(x, y, 30, 30);
30    }
31  }
```


Nested-for-loop_2

create an interactive work

```
random
5
6  /* 원 테두리 없애기 */
7  noStroke();
8
9  /* 0,0 ~ width,height 사이의 거리를 구하고 2로 나눕니다. */
10 /* 이것은 절대적인 거리이기 때문에 2를 나뉘도 되고 3으로 나뉘도 됩니다. */
11 /* 하지만 이 값이 커지는 만큼 상대적으로 원이 작을것이고 */
12 /* 작아지는 만큼 원이 커질 것 입니다. */
13 max_distance = dist(0, 0, width, height) /3;
14 }
15
16 void draw() {
17
18     background(0);
19
20     int inc = 20;
21
22     /* 첫번째 for문 31번 반복합니다. */
23     for(int i = 0; i <= width; i += inc) {
24         /* 첫번째 for문이 1번 반복할때마다 2번째 for문은 31번씩 반복합니다. */
25         for(int j = 0; j <= height; j += inc) {
26
27             /* 마우스포인터로부터 i,j 까지의 거리를 계산합니다. */
28             /* 원이 그려질때 마우스 포인터의 위치에 따라 원이 다르게 변한다는 것 입니다.*/
29             float d = dist(mouseX, mouseY, i, j);
30
31             /* 위에서 구했던 마우스 포인터에 대한 상대적인 거리로 원의 지름을 결정합니다. */
32             float radius = d / max_distance * inc;
33             ellipse(i, j, radius, radius);
34         }
35     }
}
```

Array-bar-chart

create an interactive work

```
float[] chart = new float[20],

void setup() {
  size(800, 800);
  /* 1초에 5장씩 그립니다. */
  frameRate(5);
}

void draw() {
  background(255);

  println(chart.length);
  /* 벽돌깨기에서 했던것처럼 1차원배열에 그냥 랜덤인 값을 넣습니다. */
  /* 0부터 799 까지의 랜덤 값이 들어갑니다.*/

  /* chart.length 는 20입니다. 선언할때 20개로 선언했기 때문입니다. */
  for(int i = 0; i < chart.length; i++){

    /* 0번부터 19번까지 랜덤 값을 넣습니다. */
    chart[i] = random(width);
  }

  /* 20번 반복합니다. */
  for(int i = 0; i < chart.length; i++){
    /* 색을 랜덤으로 채우고 */
    fill(random(255), random(255), random(255));

    /* 사각형의 너비를 아까 채웠던 랜덤값으로 넣어주기때문에 계속 바뀝니다. */
    rect(0, i*height/chart.length, chart[i], height/chart.length);
  }
}
```

Circle_chainging

create an interactive work

```
void setup() {
  size(800, 600);
  diam = 10;

  /* 너비와 높이를 2로 나누었으니깐 정중앙입니다. */
  centerX = width / 2;
  centerY = height / 2;

  stroke(0);
  strokeWeight(5);
  fill(255, 30);
}

void draw() {

  /* 정중앙에 원을 만들고, 너비와 높이는 diam로 정하는데,*/
  ellipse(centerX, centerY, diam, diam);

  /* 너비를 점점 커지게 합니다. */
  diam += 10;

  /* 이 부분을 나머지연산이라고 합니다. */
  /* diam와, width를 나머지연산을 했을때, */
  /* diam 는 width의 값보다 커질 수 없고, 최대 width-1 의 값이 나오게 됩니다.*/
  /* 그렇기 때문에 width보다 커지지 않고 원을 만들게 됩니다. */
  /* 10 % 800은 10이 나오고, */
  /* 30 % 800은 30이 나옵니다.*/
  /* 799 % 800은 799가 나오고, */
  /* 800 % 800은 0이 나옵니다.*/
  diam %= width;
}
```

box_drawing

create an interactive work

```
array_drawing_rectangles
1 int length;
2 boolean [][] map;
3
4 void setup(){
5     size(600, 600);
6     background(0);
7     length = 30;
8
9     /* 30x30 값을 넣을 수 있는 2차원 배열입니다. */
10    map = new boolean[length][length];
11
12    /* 벽돌깨기처럼 배열에 값만 넣습니다. */
13    /* 지금은 false 만 넣는데, 클릭했을때 true로 바뀌게 할 것입니다.*/
14    for(int x = 0; x < length; x = x+1){
15        for(int y = 0; y < length; y = y+1){
16            map[y][x] = false;
17        }
18    }
19 }
```

```
void mousePressed(){
    /* 마우스포인터의 좌표 / 길이(30) 하면 해당 위치의 절대적인 사각형의 좌표가 나옵니다.*/
    /* 왜냐하면 사각형은 length의 배수만큼 그렸기 때문입니다.*/
    if (map[mouseY/length][mouseX/length] == true){
        map[mouseY/length][mouseX/length] = false;
    }
    else{
        map[mouseY/length][mouseX/length] = true;
    }
}

void draw(){
    // 모든 2차원 배열을 반복하면서
    for(int x = 0; x < length; x = x+1){
        for(int y = 0; y < length; y = y+1){
            /* 값이 true인 것만 색을 칠합니다. */
            if (map[y][x] == true){
                fill(0);
            }
            else{
                fill(255);
            }
            /* x 랑 y를 0부터 곱하면서 차례대로 사각형을 그립니다. */
            /* 바둑판 형식으로 그려지게 됩니다. */
            rect(length*x, length*y, length, length);
        }
    }
}
```

예제 찾는 법

create an interactive work

- github.com에 자료가 많습니다.
- processing examples site:github.com
 - 위와 같이 검색하면 github.com 으로 검색 결과가 좁혀집니다.
- <https://github.com/donlnz/processing-examples>
- <https://github.com/danomatika/CodingExamples>
- <https://openprocessing.org/>

coding styles.

create an interactive work

- 들여쓰기
- 변수이름 (의미있는 변수 이름)
 - $a, b, c \Rightarrow x$
 - $xPos, yPos \Rightarrow ok$
- 함수이름 (의미있는 함수 이름)
-

comments

create an interactive work

- 한줄주석
 - //
- 여러줄 주석은
 - /* */

End.

4회차

미디어경영학과 2022-01