

미디어애펀테크 튜터링

3회차

미디어경영학과 2022-01

튜터링 일정

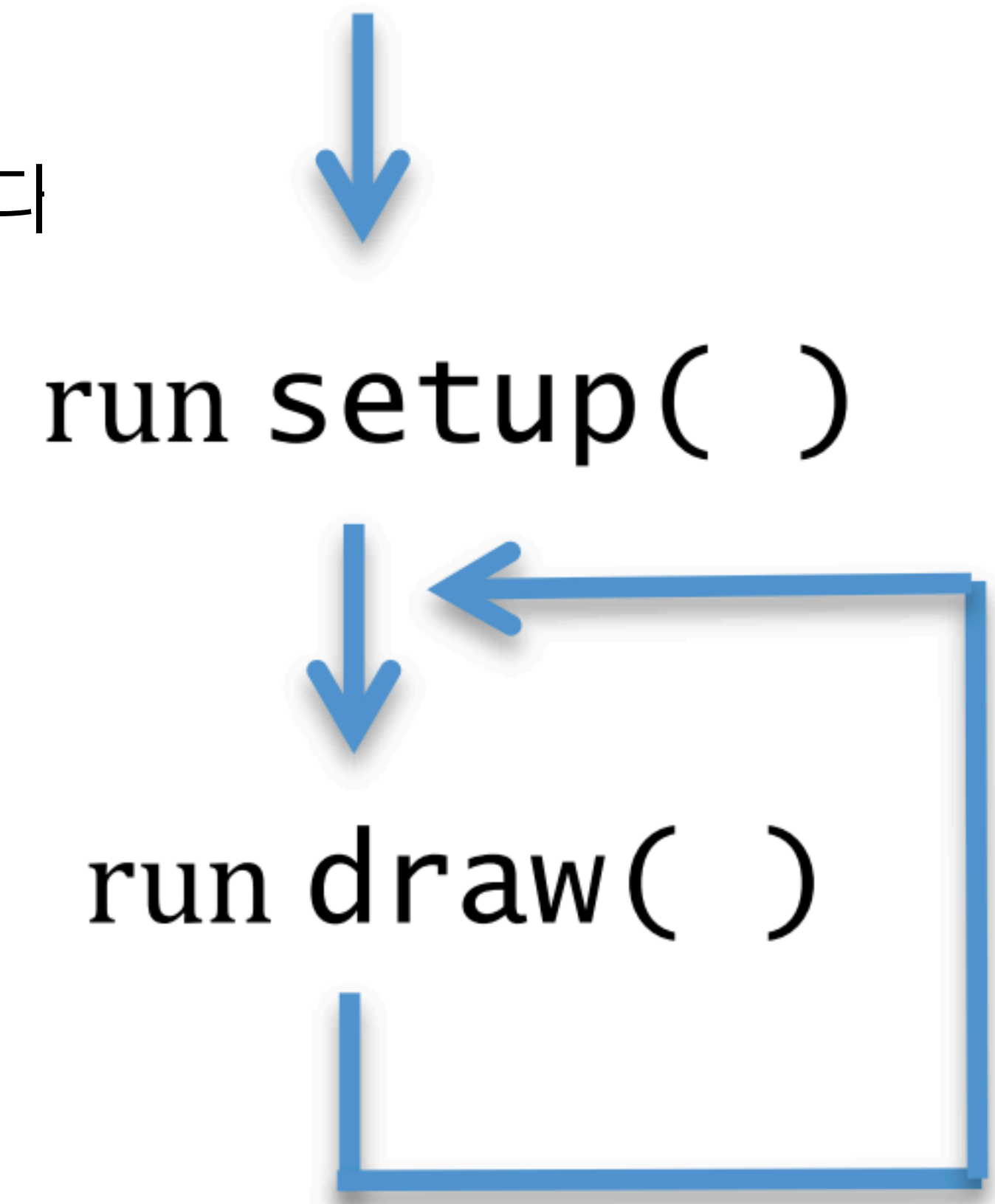
3회차

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	벽돌깨기에 점수, 게임오버 기능 추가하기
5회차	함수, 벽돌깨기 리팩토링, while문, 격자 무늬 만들기, bar chart 만들기, list 활용 dist 활용
6회차	???
7회차	???
8회차	???
9회차	???
10회차	???

2회차 리뷰

리뷰

- tweak 모드
 - draw 안에서 사용된 값들을 프로그램 실행중에 조정할 수 있습니다
- 동적 프로그래밍
 - 동영상을 그릴때 사용하는 기법입니다.
- draw() 함수
 - setup() 함수가 최초 실행되고 계속 반복하면서 실행됩니다.
- setup() 함수
 - 프로그램 실행 이후 맨 처음 최초 1번 실행됩니다.



2회차 리뷰

리뷰

- if, else if , else분기문
 - 만약 ~와 같은 일이 생길때를 구분합니다.
 - if(공이 부딪힐때) {
 - 공이 부딪히면 여기가 실행됨!
 - } else if (공의 색이 빨간색 일때) {
 - 공이 부딪히지 않으면서, 공의 색이 빨간색 일때 여기가 실행됨!
 - } else {
 - 공이 안부딪히고 빨간색도 아닐때 여기가 실행됨!
 - }

2회차 리뷰

리뷰

- $\&\&$ (AND)
 - 모두 참일때 참
- \parallel (OR)
 - 한개만 참이어도 참
- \wedge (XOR)
 - 모두 다르면 참
- $!$ (NOT)
 - 논리값 반전

2회차 리뷰

리뷰

- x 에 diam/2 를 더하거나 빼는 이유
 - x는 원의 중심이기 때문에,
 - 반지름을 더하거나 뺀 기준으로 충돌처리
- xdir 에 -1을 곱하는 이유
 - 양수에 -1을 곱하면 => 음수
 - 음수에 -1을 곱하면 => 양수
 - x 좌표의 부호가 달라져서 양 옆으로 왔다 갔다

```
12
13 void draw(){
14     background(200);
15     ellipse(x, y, diam, diam);
16     x = x + xdir; // ball movement
17
18     if (x + diam/2 > 600){
19         //change the direction.
20         xdir = xdir * -1;
21     }
22
23     if (x - diam/2 < 0){
24         xdir = xdir * -1;
25     }
26 }
27
```

Y축 충돌처리

벽돌깨기

- X 축 충돌과 마찬가지로 공의 Y축을 충돌처리

```
if(posY + diam/2 > 800) {  
    yDir = yDir * -1;  
}  
|  
if(posY - diam/2 < 0) {  
    yDir = yDir * -1;  
}
```

- Y 축의 좌표는 위 부터 아래로 좌표값이 증가함

Pad 만들기

벽돌깨기

- 사각형을 그리는 rect 함수의 x,y 좌표는 사각형 맨 처음 꼭짓점
- mouseX
 - 마우스 포인트의 x좌표
- 시작점 으로부터 패드 너비의 절반 만큼 뒤에서 패드를 그린다.

```
fill(padColor);  
padX = mouseX - padWidth/2;  
rect(padX, padY, padWidth, 20, 15);
```


Pad 충돌처리

벽돌깨기

- $\text{posX} > \text{padX}$
 - 공이 패드의 시작점 부터 충돌
- $\text{posX} < \text{padX} + \text{padWidth}$
 - 공이 패드의 끝점까지 충돌
- $\text{posY} > \text{padY} - \text{diam}/2$
 - 패드보다 낮게 떨어지면 안됨!
- 모두 $\&\&$ 연산 !

```
if ( posX > padX && posX < padX + padWidth &&  
    posY > padY - diam/2){  
    yDir *= -1;  
}
```

2중 for 문

반복문

- 반복을 반복합니다.
- for 문 안에서 for 문이 동작합니다.
- `for(int n=0; n<5; n++) {`
 - `for(int m=0; m<3; m++) {`
 - `println("반복!");` // 15번 (5x3) 반복
 - `}`
 - `}`
- `}`

2중 for 문

반복문

- 2중 for 문을 이해하기 가장 쉬운 예제는 구구단 출력하기
- 보여드릴게요! 느낌만 알고계시면 됩니다!

2중 for 문

반복문

- 벽돌깨기에서 2중 for 문이 쓰인 경우
 - 벽돌 그리기
 - 모든 벽돌에 랜덤 색을 표현하기

배열

배열

- 연속된 데이터를 나란히 보관합니다.
- 많은 개수의 데이터를 변수에 보관하고 싶은데...
- 그만큼 변수를 너무 많이 쓰면 손이 아프고 귀찮으니까!
- 배열이라는 걸 사용합니다.

1차원 배열

배열

- `int[] list = new int[10];`
 - 10개의 데이터를 보관합니다.
 - 0번 ~ 9개의 방이 생성됩니다 (0번부터 9번까지니깐 총 10개의 데이터를 보관합니다.)
- `list[0] = 1`
 - 0번 방에 1을 넣는다는 뜻!
- `list[8] = 323`
 - 8번 방에 323을 넣는다는 뜻!

2차원 배열

배열

- 1차원 배열을 배열로 가지는게 2차원 배열입니다.
- `int[][] list = new int[2][3];`
 - 6(2x3)개의 데이터를 보관합니다.
 - 0,0 | 0,1 | 0,2 | 1,0 | 1,1 | 1,2 방에 데이터를 넣을 수 있습니다.

0,0	0,1	0,2
1,0	1,1	1,2

벽돌깨기

- 총 몇 줄을 그릴건지에 대한 반복
 - 1줄에 몇개를 그릴건지에 대한 반복
- bRowNo x bColNo 개의 블록이 그려집니당
 - 예제 기준 (4x10) 총 40개의 블록이 그려집니당

```
int i, j;
for(i=0; i<bRowNo; i++){
    for(j=0; j<bColNo; j++){
        rect(j*bWidth, i*bHeight, bWidth, bHeight);
    }
}
```


벽돌깨기 - 1

벽돌깨기

- 벽돌을 깨는 원리
 - 1. 벽돌을 그리는 2차원 배열에 값을 넣어둡니다.
 - 2. draw() 함수 안에서 벽돌을 그릴때 값이 있을때만 벽돌을 그립니다.
 - 3. 공이 충돌하면 충돌한 벽돌의 값을 없앱니다.

벽돌깨기 - 2

벽돌깨기

- 벽돌을 그릴 2차원에 배열에 모든 값을 1로 넣어줍니다.
- 그래야 어떤 벽돌은 깨졌는지 깨지지 않았는지 알 수 있습니다.

```
9  
0  int i, j;  
1  for(i=0; i<bRowNo; i++){  
2      for(j=0; j<bColNo; j++){  
3          bricks[i][j] = 1;  
4      }  
5  }
```

벽돌깨기 - 4

벽돌깨기

- 공이 벽돌에
- 벽돌을 그릴때, 각 벽돌마다 1이라는 값이 있을때만 벽돌을 그립니다.

```
stroke(0);
strokeWeight(2);

int i, j;
for(i=0; i<bRowNo; i++){
  for(j=0; j<bColNo; j++){
    if ( bricks[i][j] == 1) {
      fill(#7E8AF0);
      rect(j*bWidth, i*bHeight, bWidth, bHeight);
    }
  }
}
```

벽돌깨기 - 4

벽돌깨기

- 공이 벽돌에 충돌하는 것을 구현합니다.
- $\text{posY} < \text{bHeight} * \text{bRowNo}$
 - 공의 벽돌들이 생성된 높이까지 올라간다면.

```
if ( posY < bHeight*bRowNo ) {  
    if ( bricks[posY/bHeight][posX/bWidth] > 0 ) {  
        yDir *= -1;  
        bricks[posY/bHeight][posX/bWidth]--;  
    }  
    else if ( posY < 0 ) yDir *= -1;  
}
```

벽돌깨기 - 4

벽돌깨기

- $\text{posY} < \text{bHeight} * \text{bRowNo}$
 - 공의 벽돌들이 생성된 높이까지 올라간다면.
- $\text{posY} / \text{bHeight}$ 를 하면 몇 층에 부딪혔는지.
- $\text{posX} / \text{bWidth}$ 를 하면 몇 번째 블록 (몇호수)에 부딪혔는지.

```
if ( posY < bHeight*bRowNo ) {  
    if ( bricks[posY/bHeight][posX/bWidth] > 0 ) {  
        yDir *= -1;  
        bricks[posY/bHeight][posX/bWidth]--;  
    }  
    else if ( posY < 0 ) yDir *= -1;  
}
```

End.

3회차

미디어경영학과 2022-01