

미디어애펀테크 튜터링

2회차

미디어경영학과 2022-01

튜터링 일정

2회차

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	???

1회차 리뷰

리뷰

- Processing language
 - 그림 그리는 비주얼 아트 프로그래밍 언어
- `line(x시작,y시작,x끝,y끝)`
 - 선을 그립니다.
- `ellipse(x좌표,y좌표,너비,높이)`
 - 원을 그립니다.
 - x좌표와 y좌표는 원의 중심입니다.

1회차 리뷰

리뷰

- `rect(x좌표,y좌표,너비,높이)`
 - 사각형을 그립니다.
- `stroke(rgb)`
 - 선(테두리)의 색을 정합니다.
- `noStroke()`
 - 테두리를 없앱니다.
- `fill(rgb)`
 - 모양의 색상을 정합니다.

1회차 리뷰

리뷰

- 변수
 - 변할 수 있는 수
 - 값을 담아놓고 필요할 때 가져다가 씁니당
- 관계 연산자
 - $<$, $>$, $<=$, $>=$, $==$, $!=$
- 증감 연산자
 - $a++$ $a--$

1회차 리뷰

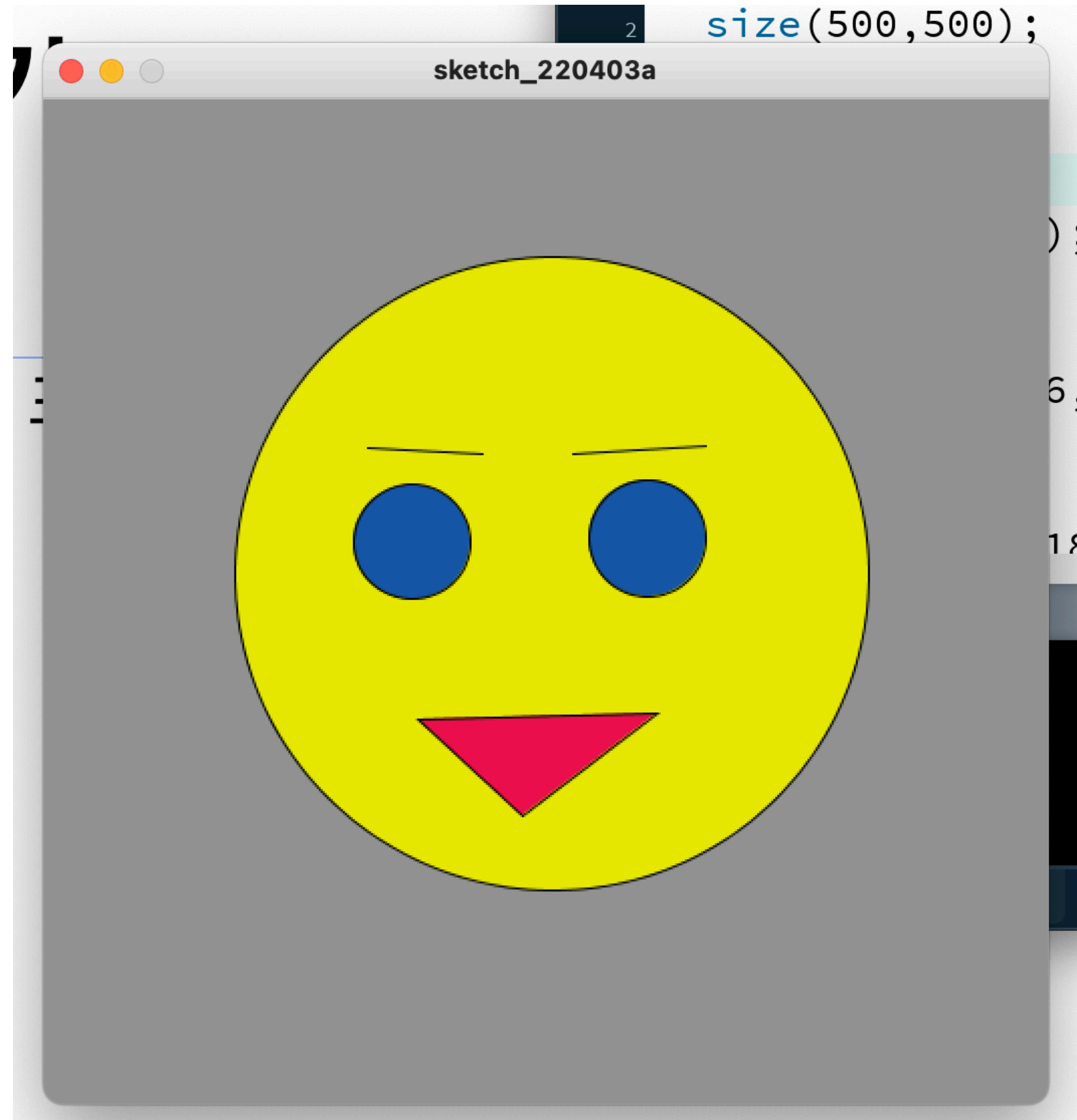
리뷰

- 반복문
 - 같은 동작을 반복해서 할 수 있게 하는 문법
 - 내가 예를 들어서 선을 10개 그리고 싶은데 10번 다
 - 치면 손이 아프니까 반복문으로 그린다.
- `for (int i=0; i<10; i++)` // 총 10번 반복한다.
 - 1부터 시작해서
 - 10이 같거나 커질때 까지
 - 1씩 증가 (`i++`) 를 한다.

[실습] 표정 그리기

도형

- 2주차 강의에서 진행했던 표정그리기를 한번 더 해볼게용
- tweak 모드를 사용



동적프로그래밍

processing 에서 말하는 동적프로그래밍

- 동영상을 그릴때 사용하는 기법입니다.
- 강의에서 사용했던 `void draw()` 가 동적 프로그래밍 기법입니다.
- 프로그램 실행 이후 에도 계속 변경점을 주고 싶을때 사용합니다.
 - 움직이는 공, 벽돌깨기
- 동영상은 프레임 단위로 움직입니다.
- (우리는 이전까지 이미지를 그리고 있었습니다.)

void draw()

processing 에서 말하는 동적프로그래밍

- 안에 있는 것들을 계속해서 실행합니다.
 - void draw () {
 - ellipse(150,150,200,200)
 - }
- 위 코드는 프로그래밍이 실행 되는 동안 원을 **계~~속** 그립니다.

void draw()

processing 에서 말하는 동적프로그래밍

- 안에 있는 것들을 계속해서 실행합니다.
- void draw () {
 - x = x + 1;
 - ellipse(x,150,200,200);
- }
- 위 코드는 프로그램이 실행 되는 동안 **계~~~속** 원이 오른쪽으로 움직입니다.

void setup()

processing 에서 말하는 동적프로그래밍

- 최초 한 번 실행합니다.
- 초기값들을 설정하는데 유용합니다.
 - void setup () {
 - size(500,500)
 - int posX = 50;
 - }

실행 주기

processing 에서 말하는 동적프로그래밍

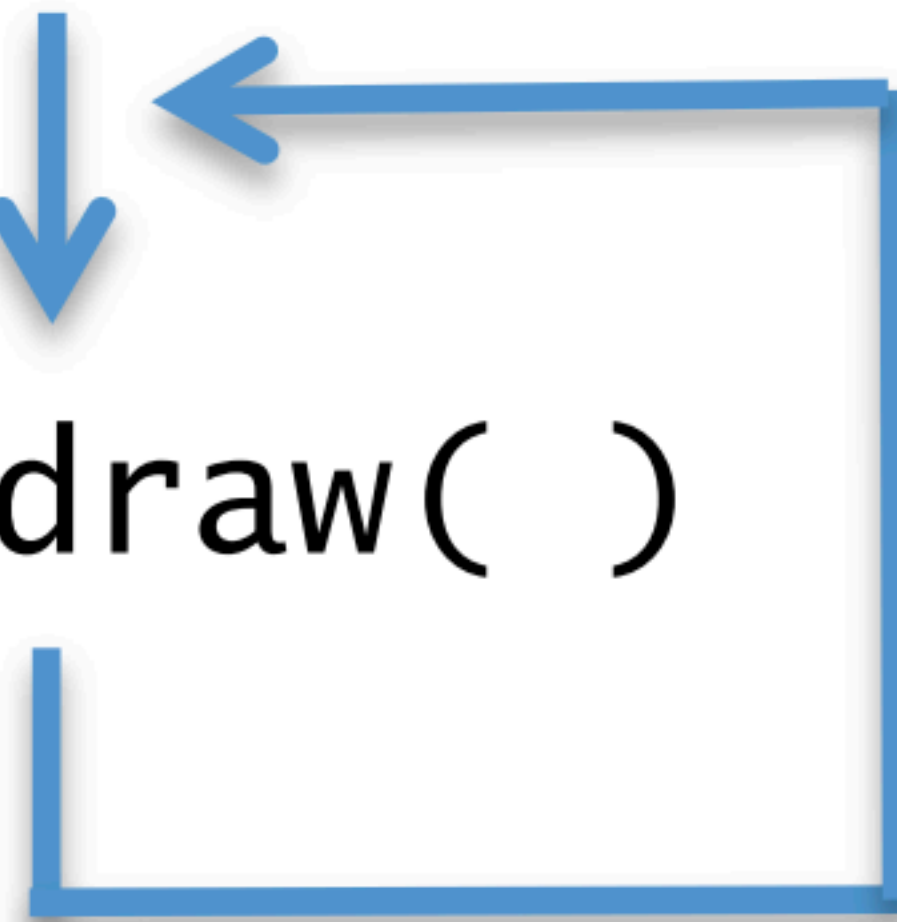
프로그램 실행



run setup()



run draw()



if()

분기문

- if 문은 만약 ~와 같은 일이 생길때를 구분합니다.
- 만약 공이 부딪힐때 방향전환을 하고싶으면
 - if (공이 부딪힌다면) {
 - 방향전환();
 - }

else()

분기문

- else 문은 만약에~~~가 통하지 않을때를 구분합니다.
- 만약 공이 부딪히기 전까지 공의 색깔을 랜덤으로 바꾸고 싶다면
 - if (공이 부딪힌다면) {
 - 방향전환();
 - } else {
 - 랜덤으로 공의 색깔 바꾸기();
 - }

else if()

분기문

- else if 문은 만약에~~~가 통하지 않을때, 또 다른 조건을 제시할 때 사용합니다.
- 만약 공이 부딪히기 전까지 공의 색깔을 랜덤으로 바꿀건데, 빨간색일때 공의 크기를 늘리려면
 - if (공이 부딪힌다면) {
 - 방향전환();
 - } else if (공이 빨간색이라면) {
 - 공의 크기를 늘리기();
 - } else {
 - 랜덤으로 공의 색깔 바꾸기();
 - }

논리연산자

연산자

- 논리식을 판단합니다.
- 관계 연산자랑 비슷합니다.
 - (다만 다른점은, 피 연산자가 true 또는 false 의 값만 가지는 변수여야 합니다.)
- and , or , xor , not 의 연산을 수행합니다.
- 논리곱 , 논리합 , 베타 논리합, 부정

논리연산자

연산자

- and 는 &&
 - if (true && true) => 결과는 ?
 - 두 개가 모두 참일때 참
- or 는 ||
 - if (true || false) => 결과는 ?
 - 둘 중에 하나가 참이면 참
- xor 는 ^
 - if (true ^ false) => 결과는 ?
 - 두개가 다르면 참
- not 은 !
 - if (!true) => 결과는 ?
 - 결과를 부정 ? 반전

if(), else if(), else() 분기문

- 굉장히 많이 쓰입니다.
- if , else if , else 순서대로 써야합니다.
- if , else , else if 이런 순서는 안됩니다.
 - if (조건) {
 - } else { <<<<< 이걸 틀렸음!
 - } else if (조건) {
 - }

논리연산자

연산자

- 벽돌깨기에서 실제로 적용했던 논리연산자 해설 (공튀기기 벽 충돌)
- if (공의 x 좌표 < 공의 너비 || 공의 x 좌표 > 화면 너비)
 - 공의 x 좌표가 공의 너비 보다 작다는것은, 화면 왼쪽 끝에 부딪혔다는 것.
 - 공의 x 좌표가 화면 너비보다 커진다는것은, 화면 오른쪽 끝에 부딪혔다는 것.
 - 둘 중에 하나 라도 해당 되면 $xDir = xDir * -1;$ 로 방향 전환

논리연산자

연산자

- 벽돌깨기에서 실제로 적용했던 논리연산자 해설 (벽돌깨기 패드 충돌)
- if (공의 $x >$ 패드의 x && $x <$ 패드의 $x +$ 패드의 너비 && $y >$ 패드의 y)
 - 공의 x 가 패드의 x 보다 커야한다는것은 공이 패드의 시작점 보다 커야한다는 것.
 - 공의 x 가 (패드의 $x +$ 패드의 너비) 보다 작아야 한다는것은 패드의 끝점보다 작아야 한다는것.
 - 공의 y 가 패드의 y 보다 커야 한다는것은 패드보다 낮게 떨어지면 안되기 때문에

[실습] 공 움직이기

공 움직이기

- draw() 안에서 구현합니다.
- 공을 계~속 한 방향으로 움직이게 하다가
 - 방향전환!

```
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
```

End.

2회차

미디어경영학과 2022-01