today

Due: Component 1, 2, 3 of Midterm project

CHANGE THE WORLD

Push Pop

Lab to work on midterm project

Student Presentations

Reading: Ch 4-6 on p.60, optional Ch 14

Monday, Feb 8

Due: Component 4 from Midterm project

Lab for component 5

Functions

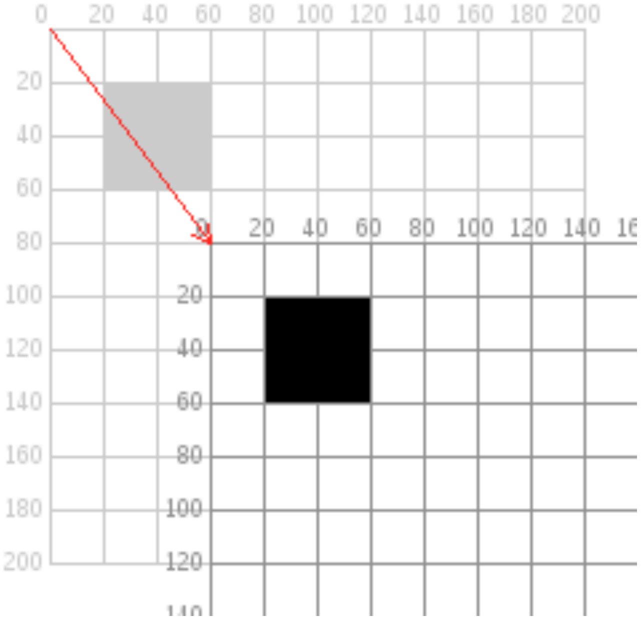
Student Presentations

a quick review from last class

Transform

translate
rotate
scale

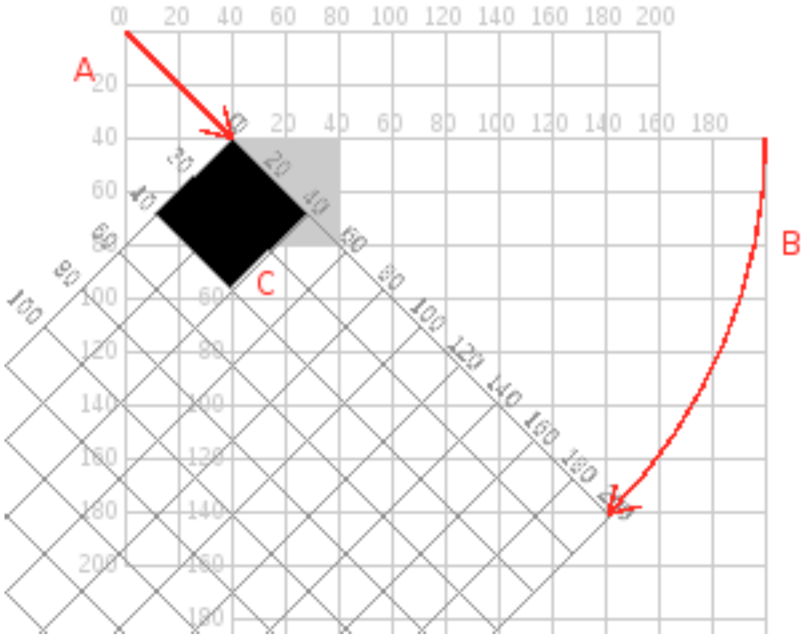
translate



```
translate(60,80);
rect(20,20,40,40);
```

translate

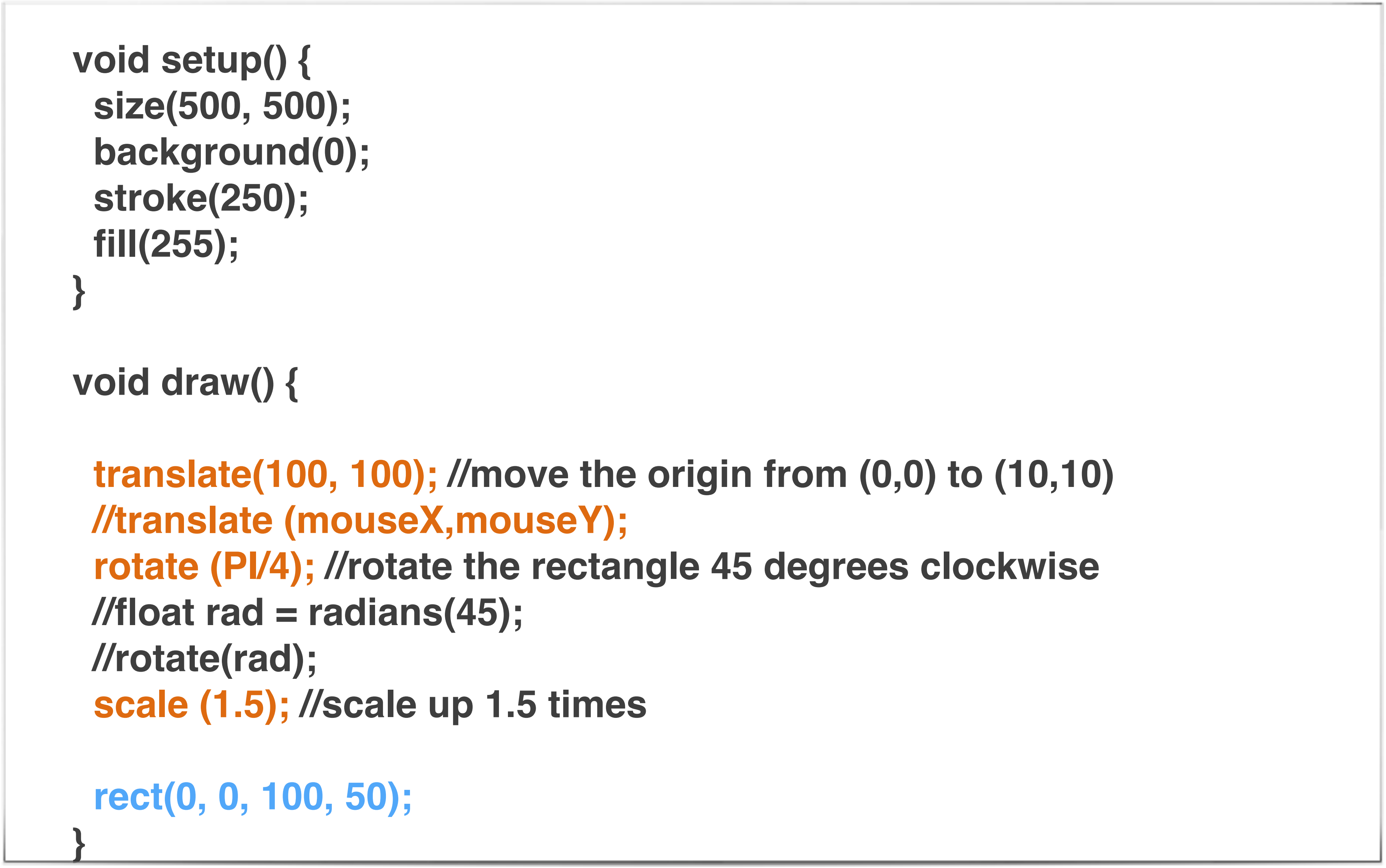
then rotate



```
translate(40,40);
rotate(radians(45));
rect(0,0,40,40);
```

Where is the center of rotation?

sketch_14_1_basic_translate_rotate_scale.pde



```
void setup() {
  size(500, 500);
  background(0);
  stroke(250);
  fill(255);
}

void draw() {

  translate(100, 100); //move the origin from (0,0) to (10,10)
  //translate (mouseX,mouseY);
  rotate (PI/4); //rotate the rectangle 45 degrees clockwise
  //float rad = radians(45);
  //rotate(rad);
  scale (1.5); //scale up 1.5 times

  rect(0, 0, 100, 50);
}
```

The order of operation matters!

What happens when you draw another shape after the coordinates have been

translated
rotated
scaled

?

The new transformed coordinates sticks with ya!

Houston, we have a problem!

This makes my math WAY too complex!

Solution:

Push/Pop matrix!

Solution:

Push/Pop matrix to isolate transformations!

pushMatrix() and popMatrix()

```
pushMatrix(); //saves current matrix

popMatrix(); // restores last saved matrix

draw(); // restores to default matrix
```

sketch_14_PushPop.pde

multiple translations

```
//Multiple Translations

void setup() {
  size (200, 200);
}

void draw() {
  translate (mouseX, mouseY);
  /*everything below this line of code is stuck to a new
coordinate system, which in this case, it is set by the
positions of the mouse.*/

  rect(0, 0, 30, 30);
  translate (35, 10);
  rect (0, 0, 15, 15);
}
//each time it draws, the coordinates are set to default.
```

sketch_14_PushPop.pde

multiple translations

isolating transformations

```
//Multiple Translations
void setup() {
  size (200, 200);
}

void draw() {
  translate (mouseX, mouseY);
  rect(0, 0, 30, 30);

  translate (35, 10);
  rect (0, 0, 15, 15);
}
```

```
//Isolating Transformations
void setup() {
  size (200, 200);
}

void draw() {
  pushMatrix(); //isolating the big rectangle
  translate (mouseX, mouseY);
  rect(0, 0, 30, 30);
  popMatrix();

  translate (35, 10);
  rect (0, 0, 15, 15);
}
```

if

you want the transformation to propagate - for each transformation to build on the last.

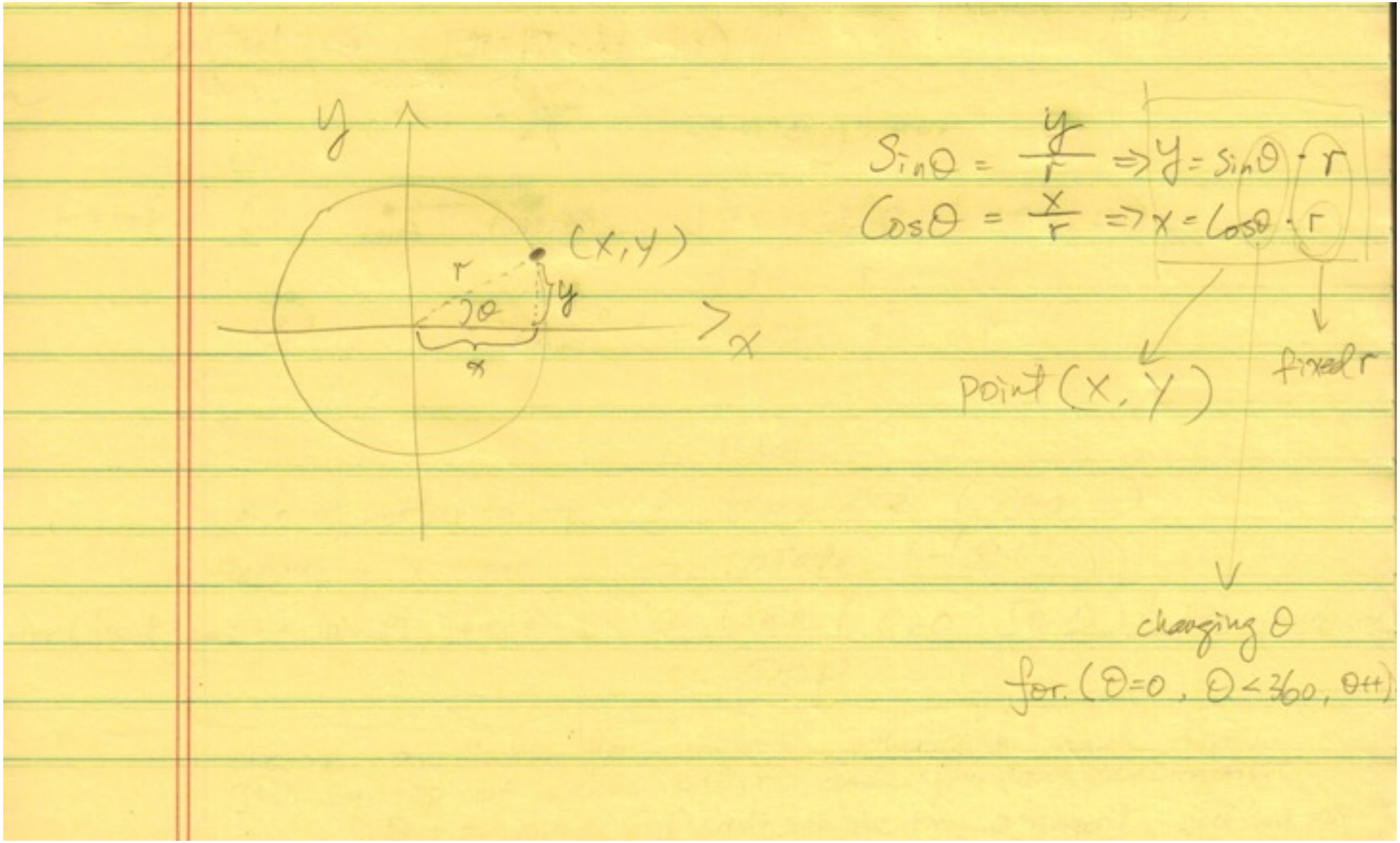
don’t use Push/Pop matrix!

a fun example on push pop

sketch_14_17_nested_pushpop.pde

p.292

first: a little math

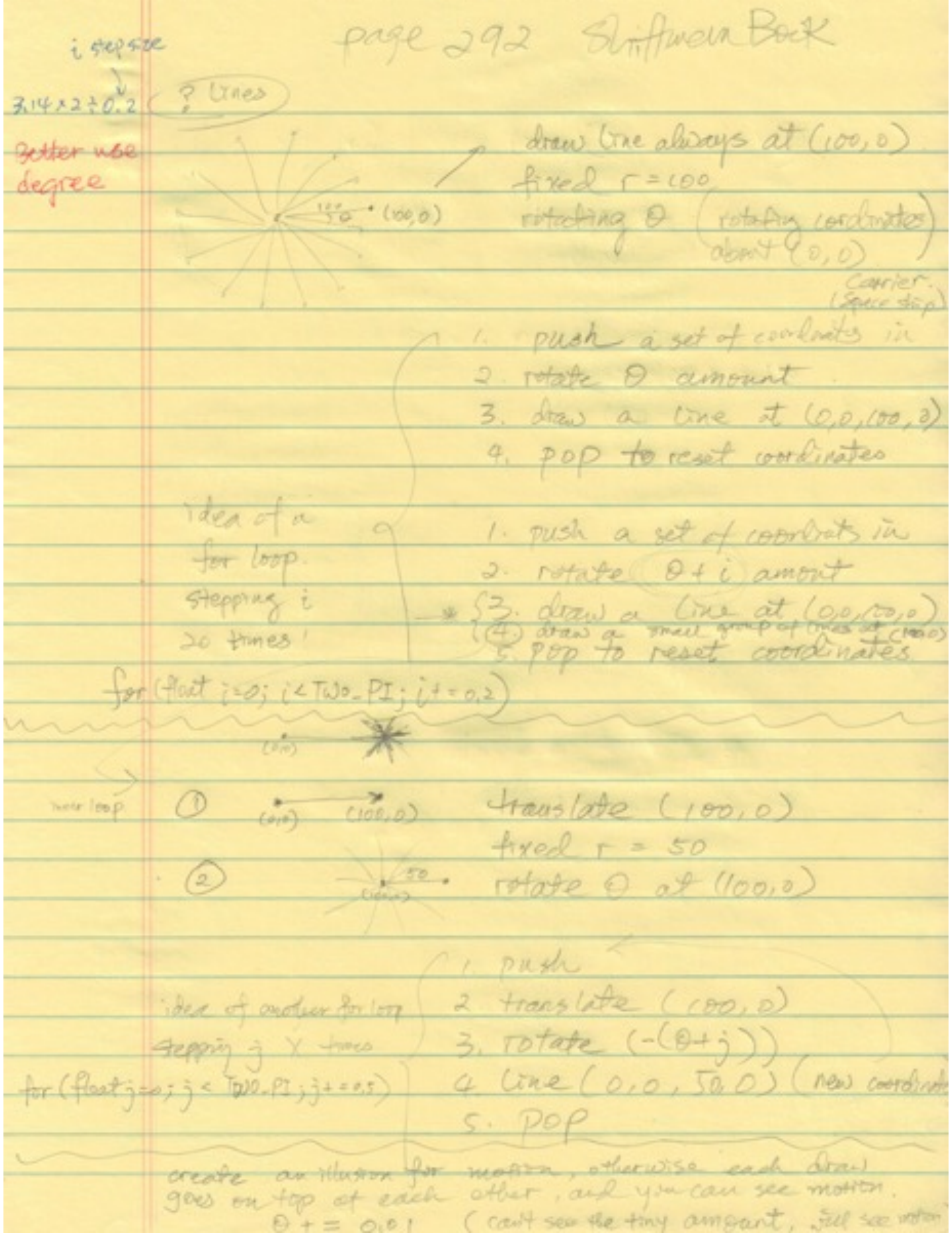


$$\sin\theta = \frac{y}{r} \Rightarrow y = \sin\theta \cdot r$$

$$\cos\theta = \frac{x}{r} \Rightarrow x = \cos\theta \cdot r$$

```
for. (θ=0 , θ<360, θ++)
```

logic for sketch_14_17_nested _pushpop.pde



page 292 Shiffman Book

i step size

3.14×2÷0.2 ? lines

better use degree

draw line always at (100,0)
fixed r = 100
rotating θ (rotating coordinates about (0,0))

Carrier.
(Space ship)

1. push a set of coordinats in
2. rotate θ amount
3. draw a line at (0,0,100,0)
4. pop to reset coordinates

idea of a for loop. stepping i 20 times!

1. push a set of coordinats in
2. rotate θ + i amount
3. draw a line at (0,0,100,0)
(4) draw a small group of lines at (100,0)
5. pop to reset coordinates

for (float i=0; i<TWO_PI; i+=0.2)

inner loop

① translate (100,0)

fixed r = 50

② rotate θ at (100,0)

idea of another for loop
stepping j X times
for (float j=0; j< TWO_PI; j+=0.5)

1. push
2 translate (100,0)
3. rotate (-(θ+j))
4 line (0,0,50,0) (new coordinat

5. pop

create an illusion for motion, otherwise each draw goes on top of each other, and you can see motion.
θ += 0.01 (can't see the tiny amount, still see motion)

today

Due: Component 1, 2, 3 of Midterm project

CHANGE THE WORLD

Push Pop

Lab to work on midterm project

Student Presentations

Reading: Ch 4-6 on p.60, optional Ch 14

Monday, Feb 8

Due: Component 4 from Midterm project

Lab for component 5

Functions

Student Presentations