

15-104 Introduction to Computing for Creative Practice

Fall 2021

06 Transformations

Instructor: Tom Cortina, tcortina@cs.cmu.edu, GHC 4117, 412-268-3514

1



Transformations

- In p5.js, you can perform a transformation on the canvas to create interesting effects.
- Types of transformations:
 - Translation (shift horizontally and/or vertically)
 - Rotation (rotate a certain number of degrees around a point)
 - Scaling (expand or contract)
- Transformations occur by moving the coordinate system of the canvas, not the object itself.

2

Translation

```
function setup() {
  createCanvas(500, 500);
  background(220);
}
function draw() {
  fill(255, 0, 0);
  rect(50, 50, 75, 75);
  translate(200, 200);
  fill(0, 255, 0);
  rect(50, 50, 75, 75);
}
```

Now the origin is at (100, 300)!

Now the origin is at (200, 200)!

3

Rotation

```
function setup() {
  createCanvas(500, 500);
  background(220);
}
function draw() {
  fill(255, 0, 0);
  rect(150, 150, 75, 75);
  fill(0, 255, 0);
  rotate(radians(30));
  rect(150, 150, 75, 75);
}
```

Rotate the coordinate system 15 degrees counter-clockwise around the origin!

Rotate the coordinate system 30 degrees clockwise around the origin!

4

rotate

- `rotate()` requires a parameter in **radians**
- `radians()` takes a value in degrees and converts it to radians
- Examples:
Rotating 90 degrees ($\pi/2$ radians)
`rotate(radians(90))`;
`rotate(PI/2)`;

5

push and pop

- `push` saves the current coordinate system and drawing properties.
- Then you can perform a transformation and draw with different colors, strokes, etc.
- `pop` returns you back to your previously saved coordinate system and drawing properties.
- You can nest `push` and `pop` inside each other to create a stack of coordinate system snapshots.

6

push and pop

```
function setup() {
  createCanvas(500, 500);
  background(220);
}
function draw() {
  fill(255, 0, 0);
  rect(150, 150, 75, 75);
  push();
  fill(0, 255, 0);
  rotate(radians(30));
  rect(150, 150, 75, 75);
  pop();
  rotate(radians(-15));
  rect(150, 150, 75, 75);
}
```

Fill resorts back to red
Rotation resorts back to 0 degrees.

7

Rotating around the origin

```
var angle = 0;
function setup() {
  createCanvas(200, 200);
}
function draw() {
  background(220);
  push();
  rotate(radians(angle));
  rect(5, 5, 50, 50);
  pop();
  angle = angle + 5;
}
```


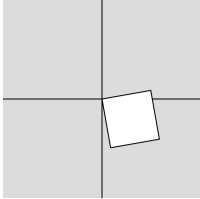
8

Rotating Around the Center

```

var angle = 0;
function setup() {
  createCanvas(200, 200);
}
function draw() {
  background(220);
  line(width/2, 0,
        width/2, height);
  line(0, height/2,
        width, height/2);
  push();
  translate(100, 100);
  rotate(radians(angle));
  // note: now we draw
  // the rectangle at 0,0:
  rect(0, 0, 50, 50);
  pop();
  angle += 5;
}

```

9


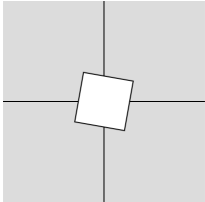
Spin on Center

```


var angle = 0;
function setup() {
  createCanvas(200, 200);
}
function draw() {
  background(220);
  line(width/2, 0,
        width/2, height);
  line(0, height/2,
        width, height/2);
  fill(255);
  stroke(0);
  push();
  translate(100, 100);
  rotate(radians(angle));
  rectMode(CENTER);
  rect(0, 0, 50, 50);
  pop();
  angle += 5;
}

```

`rectMode(CENTER)` interprets the first two parameters as the shape's center point, while the third and fourth parameters are its width and height.

10




Scaling

```
function setup() {
  createCanvas(500, 500);
  background(220);
}
```

```
function draw() {
  stroke(0);
  line(0, 100, 500, 100);
  line(0, 200, 500, 200);
  line(0, 300, 500, 300);
  line(0, 400, 500, 400);
  line(100, 0, 100, 500);
  line(200, 0, 200, 500);
  line(300, 0, 300, 500);
  line(400, 0, 400, 500);
  // cont'd on next page
}
```

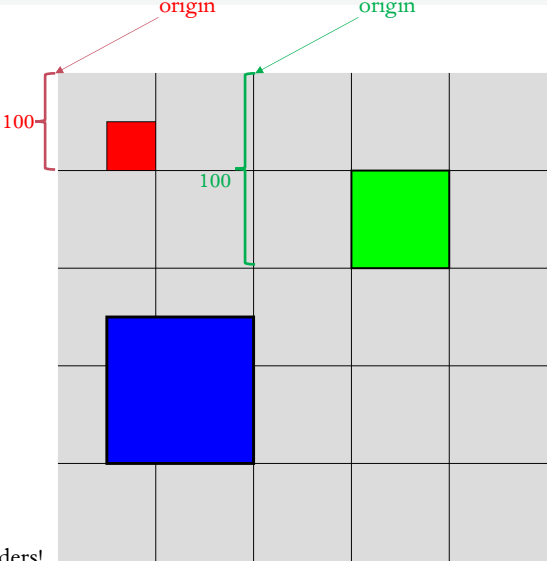
11



Scaling (cont'd)

```
fill(255, 0, 0);
rect(50, 50, 50, 50);
translate(200, 0);
scale(2.0); // now 2X
fill(0, 255, 0);
rect(50, 50, 50, 50);
translate(-150, 50);
scale(1.5); // now 3X
fill(0, 0, 255);
rect(50, 50, 50, 50);
```

Notice the width of the borders!



12



Try this:

- How would you spin the square on the location of the mouse?
- How can you create a spiral?

HINTS:

- The image is created over time by repeatedly drawing a small black circle.
- There are two global variables: angle as in the examples above, and another variable to represent where to draw the circle.
- Both variables increase by a small increment each time `draw()` is called.

