Custom Shapes,
Sine Waves &
Keyboard Interactions

Day 4 - Aug 6, 2015

Custom shapes

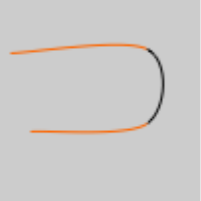
Basic shapes:

`line(), rect(), ellipse(), quad(), triangle()`

More shapes:

Arcs, Polygons, Bezier curves, Vertex, etc.

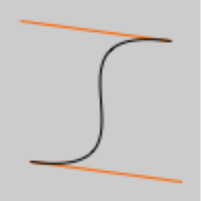
Curve & Bezier Curve



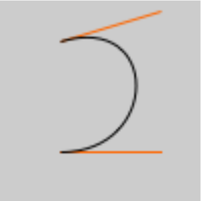
```
curve(5, 26, 5, 26, 73, 24, 73, 61);
curve(5, 26, 73, 24, 73, 61, 15, 65);
curve(73, 24, 73, 61, 15, 65, 15, 65);
```

curve(x1, y1, x2, y2, x3, y3, x4, y4)

curve(x1, y1, z1, x2, y2, z2, x3, y3, z3, x4, y4, z4)



```
bezier(85, 20, 10, 10, 90, 90, 15, 80);
```



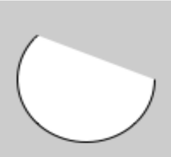
```
bezier(30, 20, 80, 5, 80, 75,  30, 75);
```

bezier(x1, y1, x2, y2, x3, y3, x4, y4)

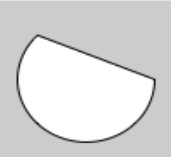
bezier(x1, y1, z1, x2, y2, z2, x3, y3, z3, x4, y4, z4)

Arc

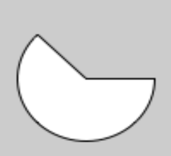
arc(a, b, c, d, start, stop)
arc(a, b, c, d, start, stop, mode)



```
arc(50, 50, 80, 80, 0, PI+QUARTER_PI, OPEN);
```



```
arc(50, 50, 80, 80, 0, PI+QUARTER_PI, CHORD);
```



```
arc(50, 50, 80, 80, 0, PI+QUARTER_PI, PIE);
```

Complex Shapes

`beginShape(MODE);`

Begin the shape with the specified mode

- available modes are `POINTS, LINES, TRIANGLES, TRIANGLE_STRIP, TRIANGLE_FAN, QUADS, QUAD_STRIP`

`vertex(x, y);`

Add a point to your shape

`endShape();`

End/close the shape

Reference

We'll live-code a few of these today.

BUT FOR MORE, go to...

File > Examples > Basics > Form

...and try using/customizing whatever shapes interest you.

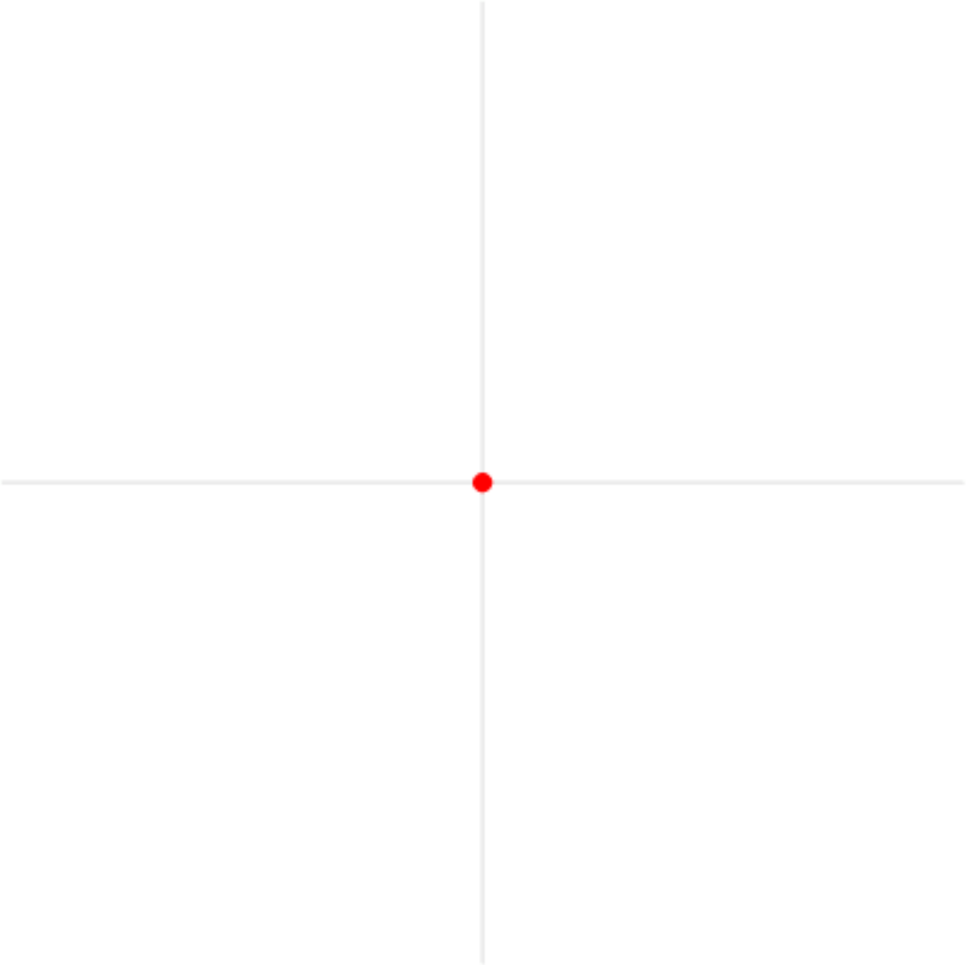
Now, the things we love the most:

MATH! Trigonometry! Yay.

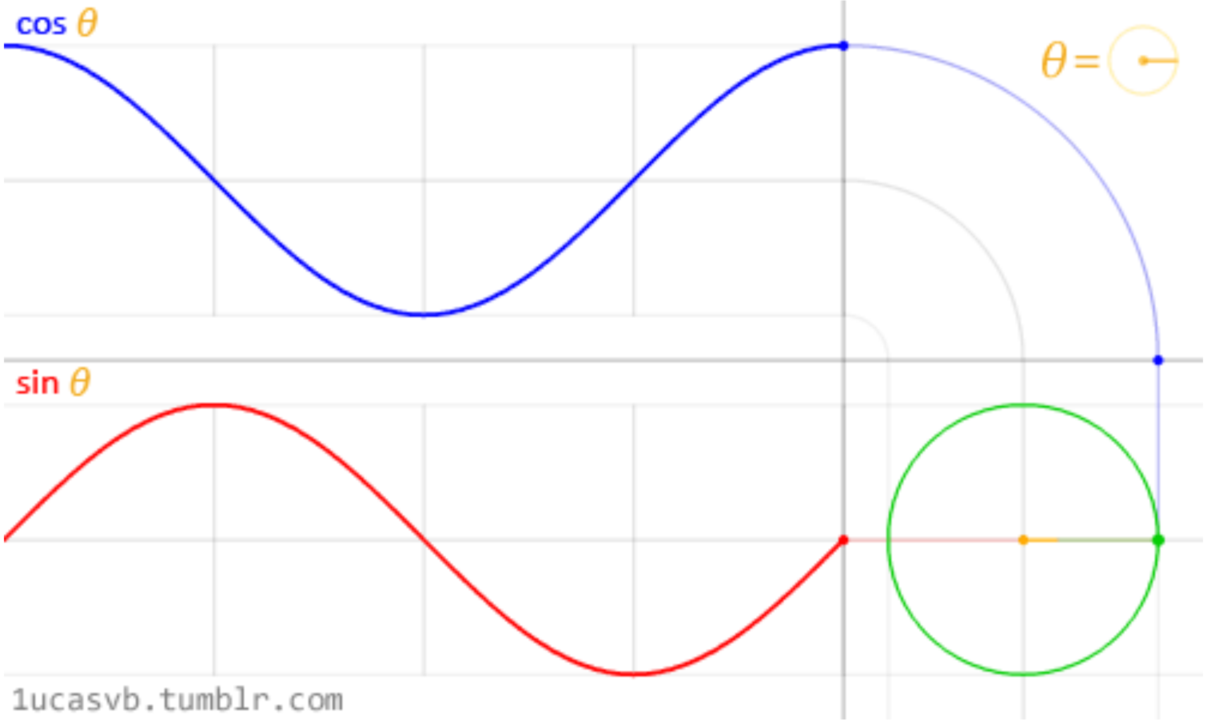
Know Your Radians

And a few constants will come in handy, because...
Processing uses **radians** rather than degrees.

- PI
- QUARTER_PI
- HALF_PI
- TWO_PI



Know Your Waves



Both oscillate between -1 and 1, which is why they make waves!

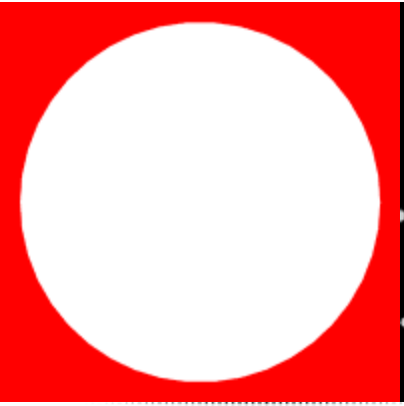
When sine = 0,
cosine = -1 or 1.
As sine increases,
cosine decreases
(and vice-versa).

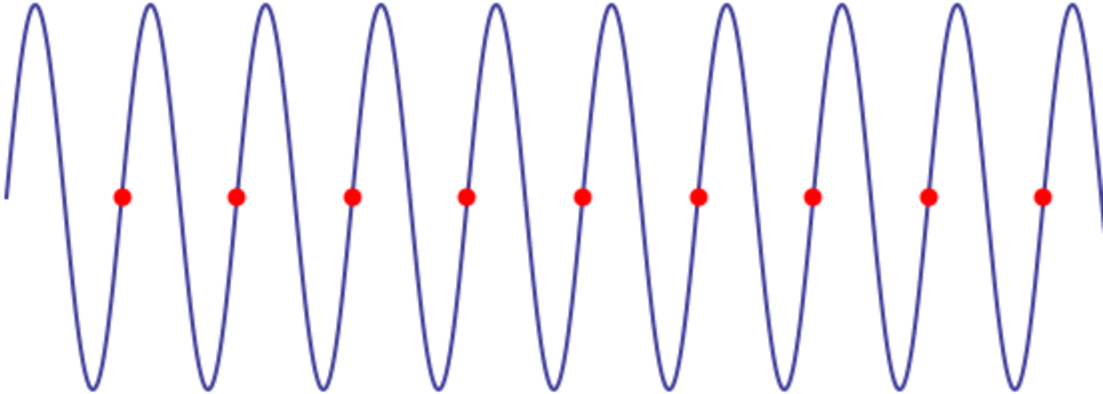
Why are we learning this?

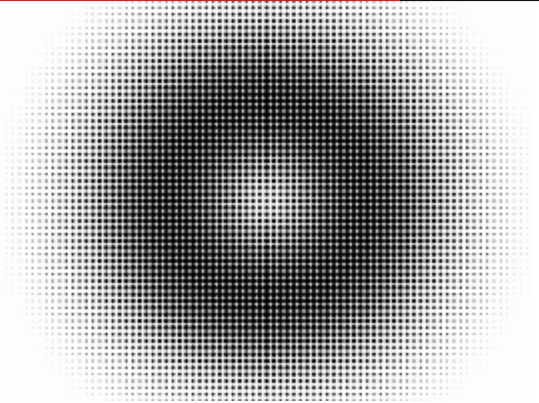
Sine waves are extremely useful for visual simulation and animation.

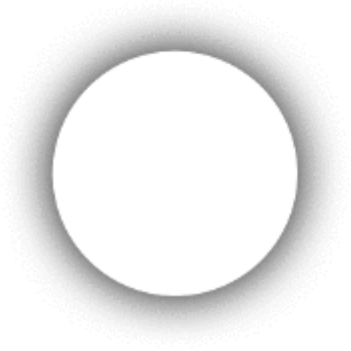
Static images: smooth circles, curves, waveforms, spirals

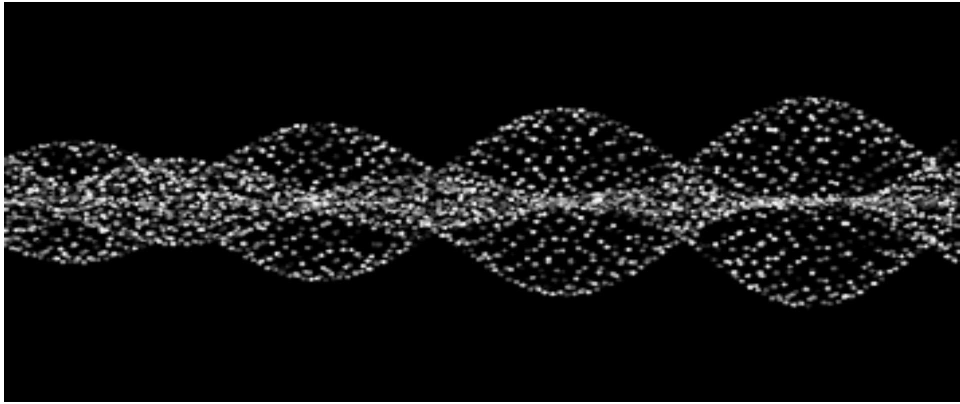
Animation: pulses, orbits, waves, oscillations











HOW? By incrementing values.

What changing values can we use for sin() and cos()?

Changing over space: *0 to 2π (360°), x position, y position*

> These create **static (frozen)** waves.

Changing over time: *frameCount, millis, int counter ++*

> These creates **animated** oscillations.

Cheatsheet: Combining Space and Time

Moving waves:

```
y = sin(x + frameCount)
```

Orbits:

```
x = sin(frameCount); y = cos(frameCount)
```

Pulses:

```
radius = abs(sin(frameCount))
```

Controlling your waves

You can control a wave's **length/speed** and its **height/size.** (*Nerd terms:* frequency and amplitude)

multiply INSIDE the function to change length/speed:

```
x = sin(frameCount*0.1) // this makes a slower pulse
```

multiply OUTSIDE to change height/size:

```
x = sin(frameCount)*10 //this makes a pulse with radius 10
```

Keyboard Interactions

Keypressed : ways to do it

1. Make an 'if' statement in the draw() loop:

```
if (keyPressed){
   //Do something when ANY key is pressed
}
if (keyReleased){
   //Do something when ANY key is released
}
```

Keypressed : ways to do it

2. As a separate function:

```
void keyPressed(){
   //Do something when any key is pressed
}
```

Keypressed : specific letters

```
if (keyPressed) {  // or void keyPressed(){
    if (key == 's') {
        saveFrame("line-######.jpg");
    }
    if (key == ' ') {  // this means the Spacebar key
        restart();
    }
}
```

Keypressed : special keys

```
if (keyPressed) { // or void keyPressed(){
    if (key == CODED) {
        if (key == UP) {
            yPosition -= 1;
        }
        if (key == DOWN) {
            yPosition += 1;
        }
    }
}
```

Letters vs. special keys

Letters/numbers are "ASCII" characters.

ASCII translates characters into computer code.

If you need a specific ASCII code, look it up!

http://www.ascii-code.com/

Special keys:

UP, DOWN, LEFT, RIGHT, ALT, CTRL, SHIFT, etc.

Use the "if (key == CODED)" syntax for these:

https://processing.org/reference/keyCode.html

Homework!

Make something alive! Pulses, spin, oscillates, animates, using the custom shapes you've learned.

BONUS: Add GUI to control it (like, buttons)