

## 15-104 Introduction to Computing for Creative Practice

*Fall 2020*

### 12 Polar Coordinates

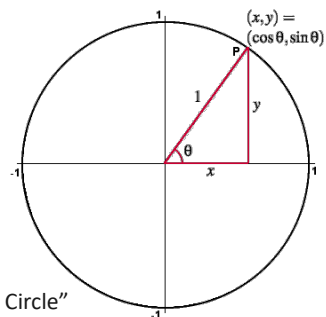
Instructor: Tom Cortina, [tcortina@cs.cmu.edu](mailto:tcortina@cs.cmu.edu), GHC 4117, 412-268-3514

1



## Polar Coordinates


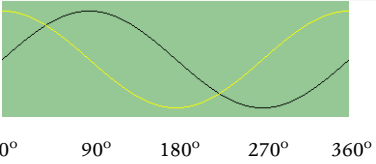
- So far, we have talked about expressing locations in terms of X and Y, but you can also describe locations in terms of angle and radius. These are called polar coordinates.
- $\cos$  and  $\sin$  functions tell you X and Y coordinates of a point on a circle of radius 1. The input parameter for  $\cos$  and  $\sin$  is the *angle*: how far to rotate around the circle. The output is where you land in terms of X (using  $\cos$ ) and Y (using  $\sin$ ).



"The Amazing Unit Circle"

2

## Sin and Cos


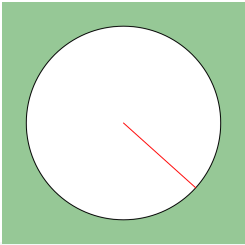



```
function setup() {
  createCanvas(360, 120);
}

function draw() {
  background(150, 200, 150);
  for (var x = 0; x < width; x = x + 1) { // x is degree value
    stroke(0);
    point(x, 60 - 50 * sin(radians(x))); // black sine wave
    stroke(255, 255, 0);
    point(x, 60 - 50 * cos(radians(x))); // yellow cosine wave
  }
}
```

3

## Polar Coordinates

```
var theta = 0; // rotation angle
var r = 200; // radius

function setup() {
  createCanvas(500, 500);
  frameRate(15);
}

function draw() {
  background(150, 200, 150);
  fill(255);
  strokeWeight(2);
  push();
  translate(250, 250);
  stroke(0);
  circle(0, 0, 2*r);
  var x = r * cos(radians(theta));
  var y = r * sin(radians(theta));
  stroke(255, 0, 0);
  line(0, 0, x, y);
  pop();
  theta += 1;
}
```

Remember that the unit circle here is upside-down (y increases downward!)

4



## Tracing in the Console

```
// add to the end of the draw function
var unit_x = map(x, -r, r, -1.0, 1.0);
unit_x = round(unit_x * 1000) / 1000; // round to nearest 1000th
var unit_y = map(y, -r, r, -1.0, 1.0);
unit_y = round(unit_y * 1000) / 1000; // round to nearest 1000th
print("unit_x = " + unit_x.toString()
      + " unit_y = " + unit_y.toString());
```

Rounding to nearest thousandth:

$\text{round}(0.123456789 * 1000) / 1000 \rightarrow \text{round}(123.456789) / 1000 \rightarrow 123 / 1000 \rightarrow 0.123$   
 $\text{round}(0.987654321 * 1000) / 1000 \rightarrow \text{round}(987.654321) / 1000 \rightarrow 988 / 1000 \rightarrow 0.988$

5



## Spinning Brightening Circles

Draw three circles that spin around the center of the canvas in a clockwise fashion.

One circle will cycle through the shades of red, one through the shades of green and one through the shades of blue.

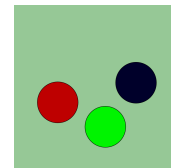
The red circle will start at 0 degrees.

The green circle will start at 120 degrees.


The blue circle will start at 240 degrees.

The green circle will rotate at twice the speed of the red circle.

The blue circle will rotate at thrice the speed of the red circle.



6



## Spinning Brightening Circles

```


var rAngle = 0;
var gAngle = 120;
var bAngle = 240;

function setup() {
  createCanvas(400, 400);
  frameRate(60);
}

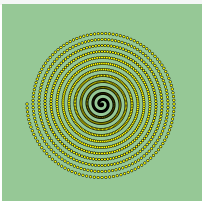
function draw() {
  background(150, 200, 150);
  var center_x = width / 2;
  var center_y = height / 2;
  var r = 100;
  push();
  translate(center_x, center_y);
  var xr = r * cos(radians(rAngle));
  var yr = r * sin(radians(rAngle));
  var xg = r * cos(radians(gAngle));
  var yg = r * sin(radians(gAngle));
  var xb = r * cos(radians(bAngle));
  var yb = r * sin(radians(bAngle));
  fill(rAngle % 256, 0, 0);
  circle(xr, yr, 100);
  fill(0, gAngle % 256, 0);
  circle(xg, yg, 100);
  fill(0, 0, bAngle % 256);
  circle(xb, yb, 100);
  rAngle += 1;
  gAngle += 2;
  bAngle += 3;
  pop();
}

```

7



## Spiraling



```


var angle = 0;
var radius = 0;
var frameCount = 0;

function setup() {
  createCanvas(400, 400);
  background(150, 200, 150);
  frameRate(60);
}

function draw() {
  var center_x = width / 2;
  var center_y = height / 2;
  push();
  translate(center_x, center_y);
  var x = radius *
    cos(radians(angle));
  var y = radius *
    sin(radians(angle));
  fill(255, 255, 0);
  circle(x, y, 5);
  radius += 0.1;
  angle += 3;
  frameCount += 1;
  pop();
  if (frameCount > 3000) {
    noLoop();
  }
}

```

8



## How would you restart the spiral?


Replace:

```
if (frameCount > 3000) {
  noLoop();
}
```


with:

```
if (frameCount > 3000) {
  frameCount = 0;
  radius = 0;
  angle = 0; // optional
  background(150, 200, 150);
}
```


9



## How would you make this?



10



## Dizzy Spiral

```

var angle = 0;
var radius = 0;
var frameCount = 0;
var circleSize = 1;


function setup() {
  createCanvas(400, 400);
  background(0);
  frameRate(60);
  noStroke();
}

function draw() {
  var center_x = width / 2;
  var center_y = height / 2;
  push();
  translate(center_x, center_y);

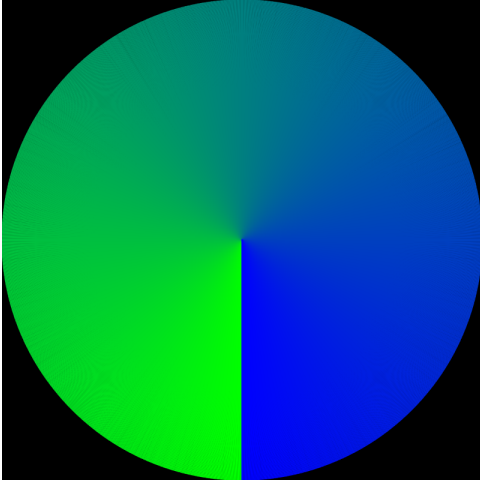
  var x = radius *
    cos(radians(angle));
  var y = radius *
    sin(radians(angle));
  fill(255, 255, 0);
  circle(x, y, circleSize);
  radius += frameCount / 1500;
  angle += 3;
  frameCount += 1;
  circleSize += 0.1;
  pop();
  if (frameCount > 1500) {
    noLoop();
  }
}

```

11



## Try this!



12