Nik Bhattacharya

co-founder

Hit Wicket Apps

What do I name this presentation?

Idea # 1

My **Grandfather** Invented Trigonometry

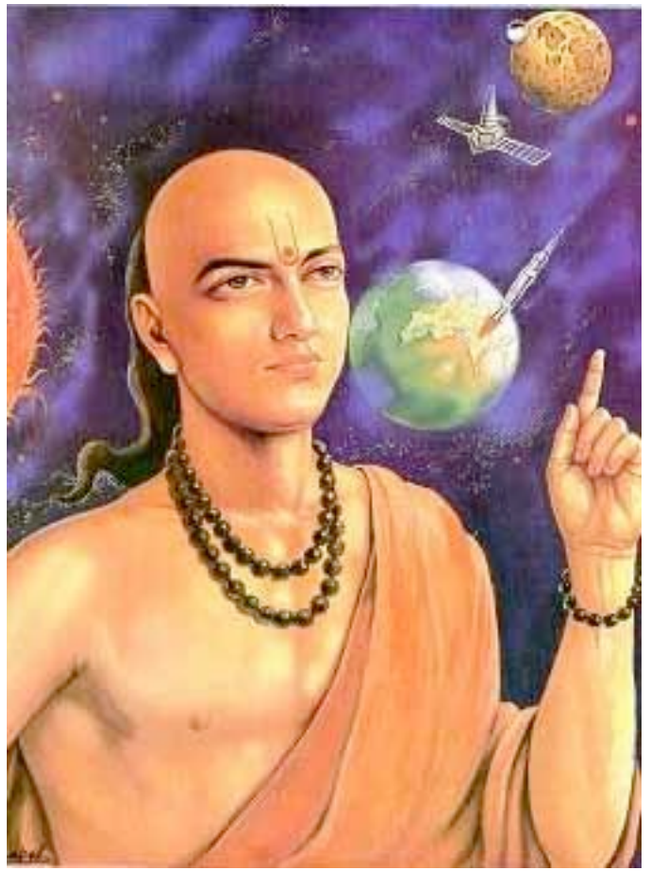
1. My Grandfather Invented Trigonometry

500 BC

Arya Bhatta

Bhattacharya

Me



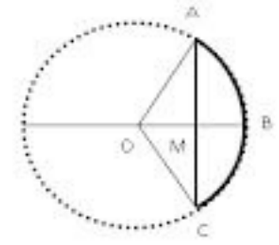
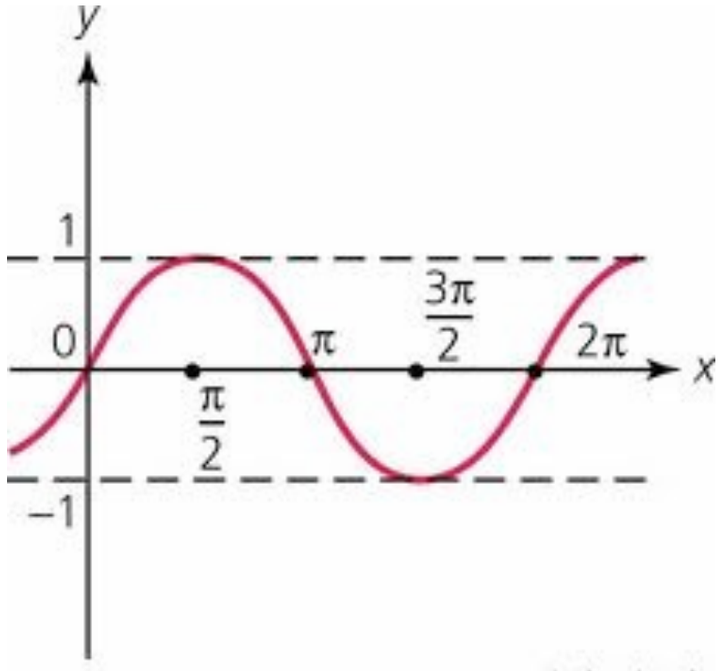


Figure-1

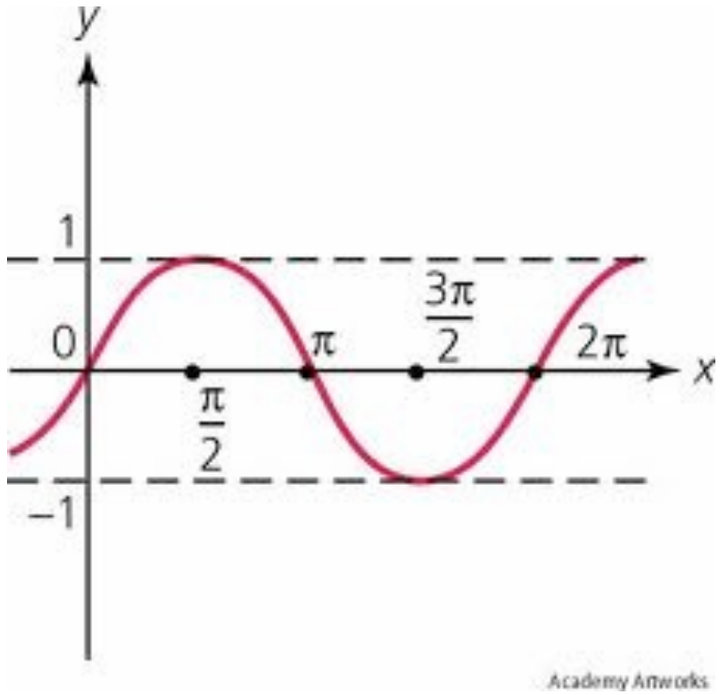
Idea # 2

sine\\



Idea # 2

sine\\ Curves are *SEXY*



Academy Artworks

Polished

User Interfaces

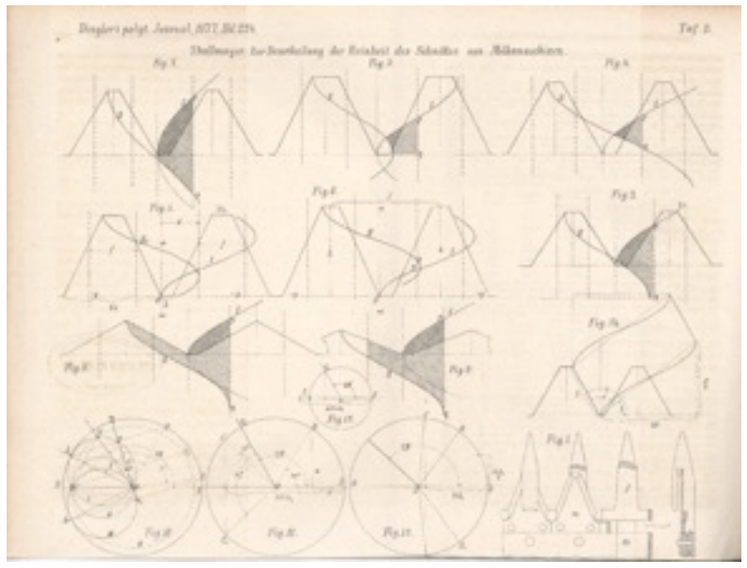
2 Main Topic Categories To Discuss

The “Problem”

Human Computer Interaction is getting richer as are **Consumer Expectations**

2 Main Topic Categories To Discuss

The “Problem”	Problem Solving Tools
Human Computer Interaction is getting richer as are **Consumer Expectations**	**Math** can unlock tools that help break down problems * **Algebra** * **Trigonometry**



2 Main Topic Categories To Discuss

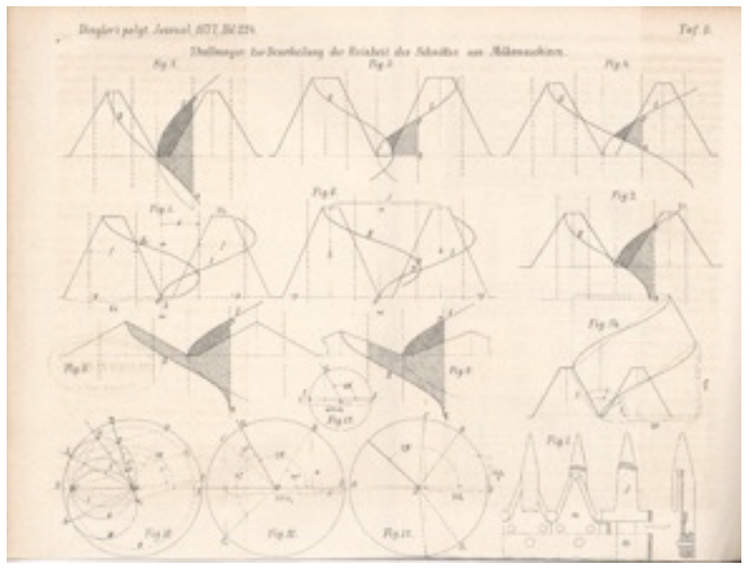
The “Problem”

Problem Solving Tools

Human Computer Interaction is getting richer as are **Consumer Expectations**

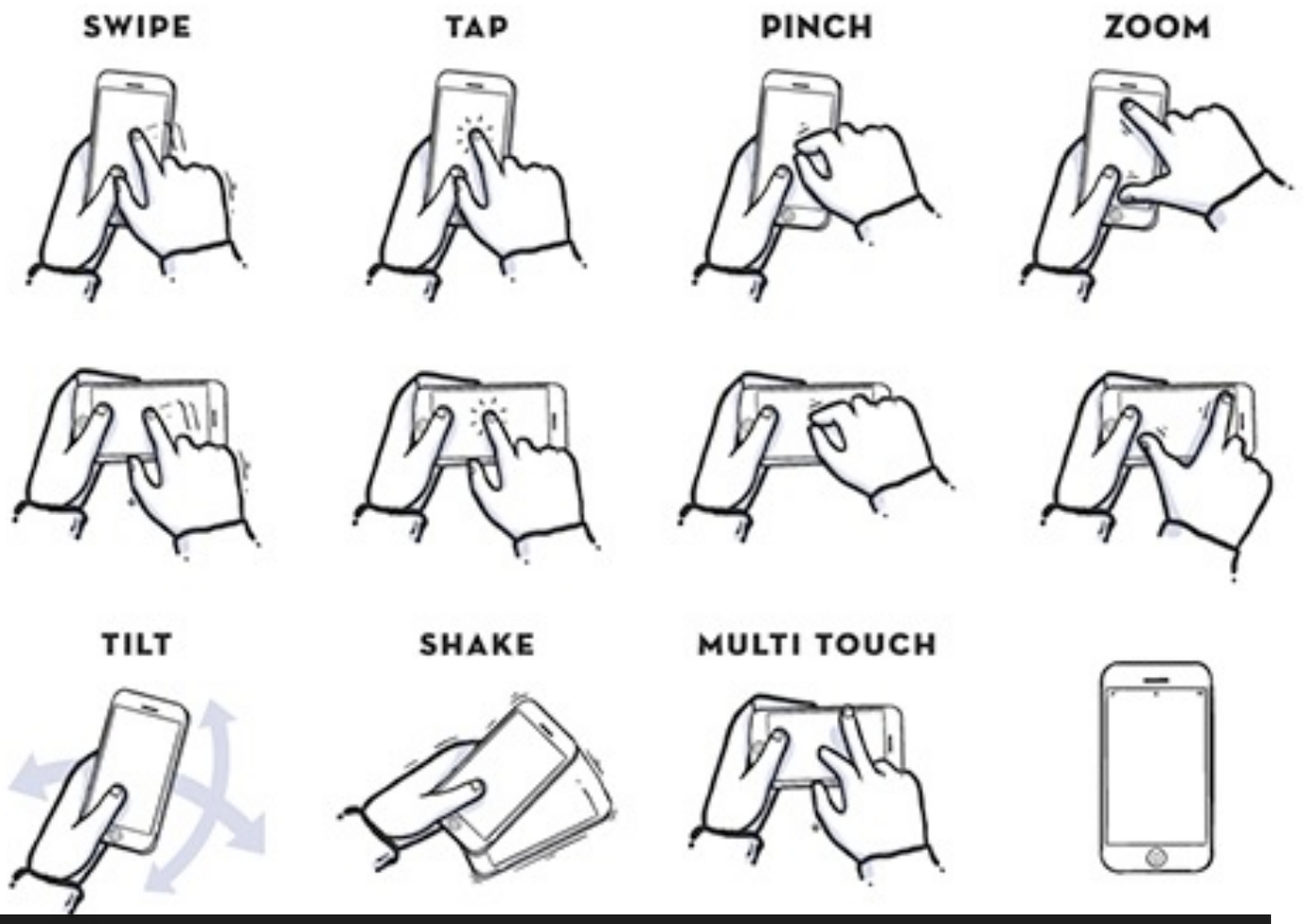
Math can unlock tools that help break down problems
* **Algebra**
* **Trigonometry**

Prototyping and Sketching is equally important for Computer Scientists as they are for artists and designers

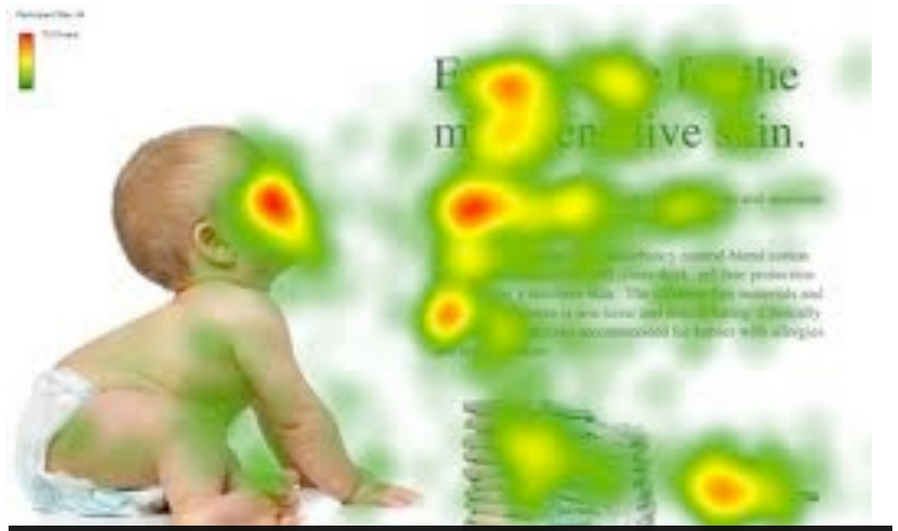


The Complexity “Problem”

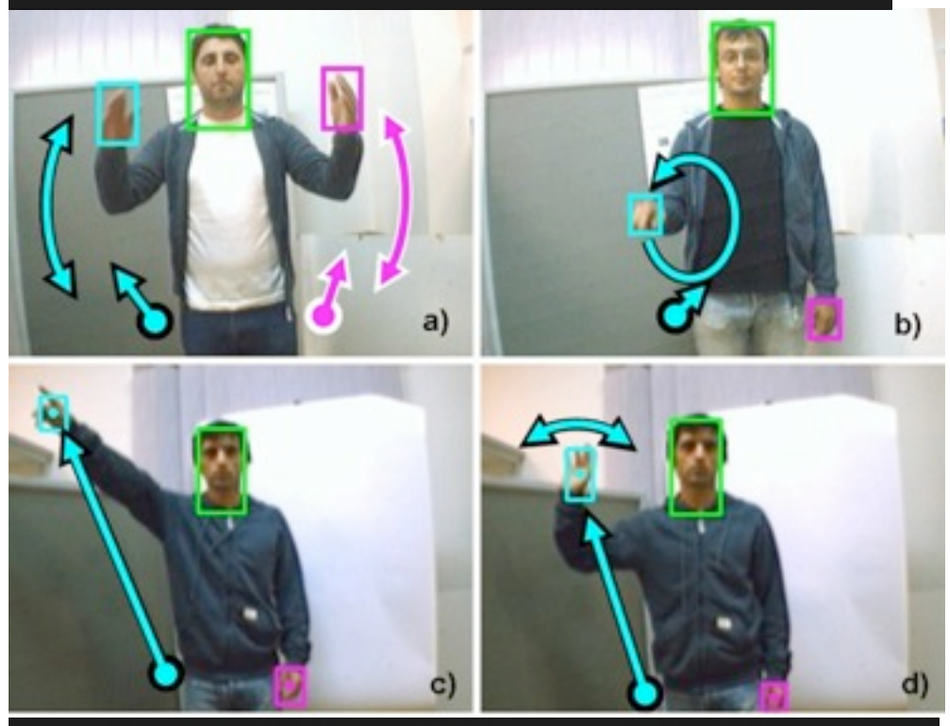
Human Computer Interaction is getting richer



Touch Events and Motion Events: *Phone & Tablet*

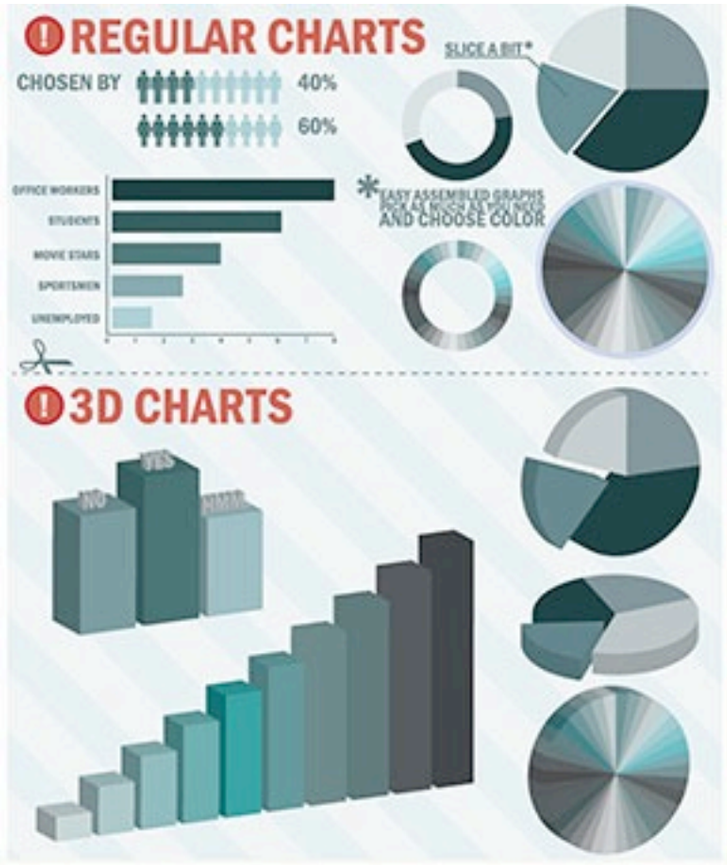


Eye Tracking: *Google Glass*

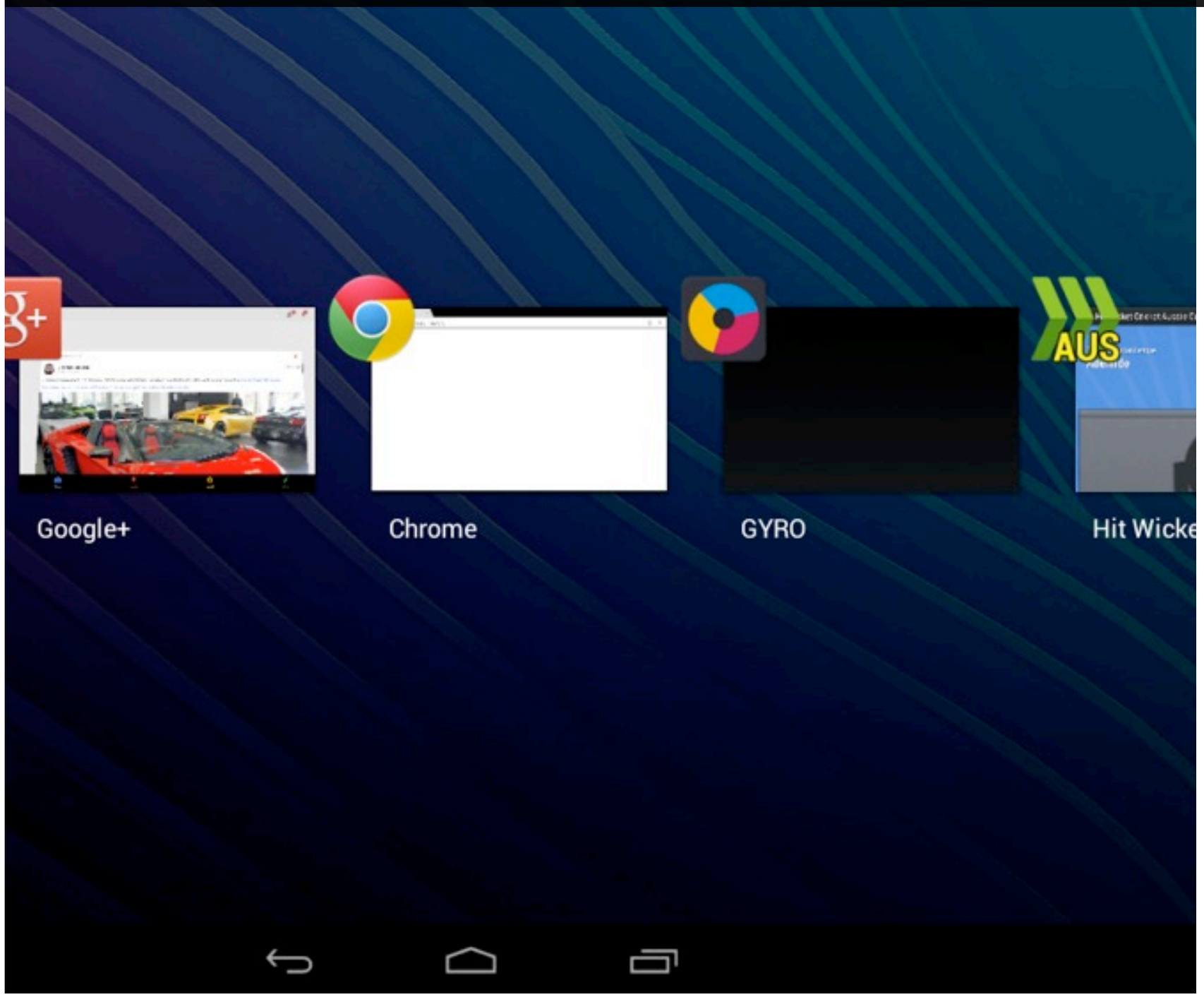


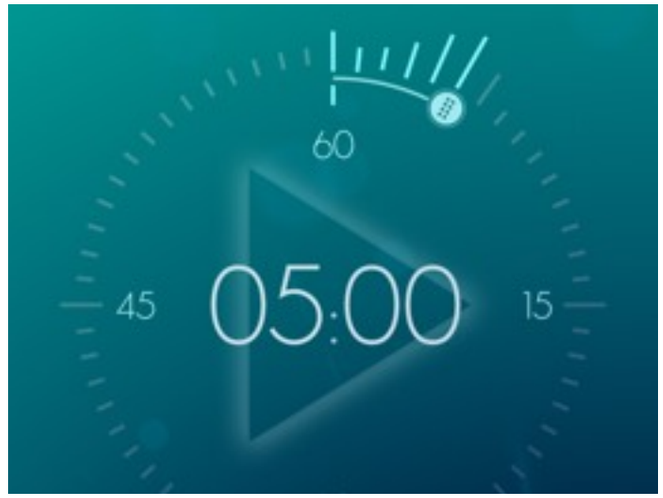
Gesture Tracking: *Kinect*

Consumers expect beautiful, interactive Infographics



Demos of Rich Interactions & Infographics





Topic 1: Touch Interfaces call for Rich Interactions

What tools as developers can we use to help make our lives easier?

Algebra and Trigonometry are Interactive Gold

Algebraic functions that change the way you break down problems

normalize

lerp

clamp

map

4

1. normalize

Converts a number in a specified range into a value between **0 and 1**

1. normalize

Converts a number in a specified range into a value between **0 and 1**

1. normalize

Converts a number in a specified range into a value between **0 and 1**

1. normalize

Converts a number in a specified range into a value between **0 and 1**

```
float norm(float value, float start, float stop) {
   return (value - start) / (stop - start);
}
```

2. lerp - Linear Interpolation

Returns a number in a specified range based on a percentage

2. lerp

Returns a number in a specified range based on a percentage

2. lerp

Returns a number in a specified range based on a percentage

```
float lerp(float start, float stop, float amt) {
   return start + (stop - start) * amt;
}
```

3. clamp

Restricts a value between an upper and lower bound

3. clamp

Restricts a value between an upper and lower bound

3. clamp

Restricts a value between an upper and lower bound

```
float clamp(float value, float lowerBound, float upperBound) {
    return Math.max(lowerBound, Math.min(value, upperBound));
}
```

4. map

Converts a value in a specified range into a value in another range

4. map

Converts a value in a specified range into a value in another range

4. map

Converts a value in a specified range into a value in another range

```
float map(  float value,
            float srcStart, float srcStop,
            float dstStart, float dstStop) {

            float normVal = norm(value, srcStart, srcStop);
            float dstRange = dstStop - dstStart;

            return dstStart + (dstRange * normVal);
}
```

4. map

Converts a value in a specified range into a value in another range

```
float map(  float value,
            float srcStart, float srcStop,
            float dstStart, float dstStop) {

            float normVal = norm(value, srcStart, srcStop);
            float dstRange = dstStop - dstStart;

            return dstStart + (dstRange * normVal);
}
```

4. map

Converts a value in a specified range into a value in another range

```
float map(  float value,
            float srcStart, float srcStop,
            float dstStart, float dstStop) {

            float normVal = norm(value, srcStart, srcStop);
            float dstRange = dstStop - dstStart;

            return dstStart + (dstRange * normVal);
}
```

4. map

Converts a value in a specified range into a value in another range

```
float map(  float value,
            float srcStart, float srcStop,
            float dstStart, float dstStop) {

            float normVal = norm(value, srcStart, srcStop);
            float dstRange = dstStop - dstStart;

            return dstStart + (dstRange * normVal);
}
```

4. map

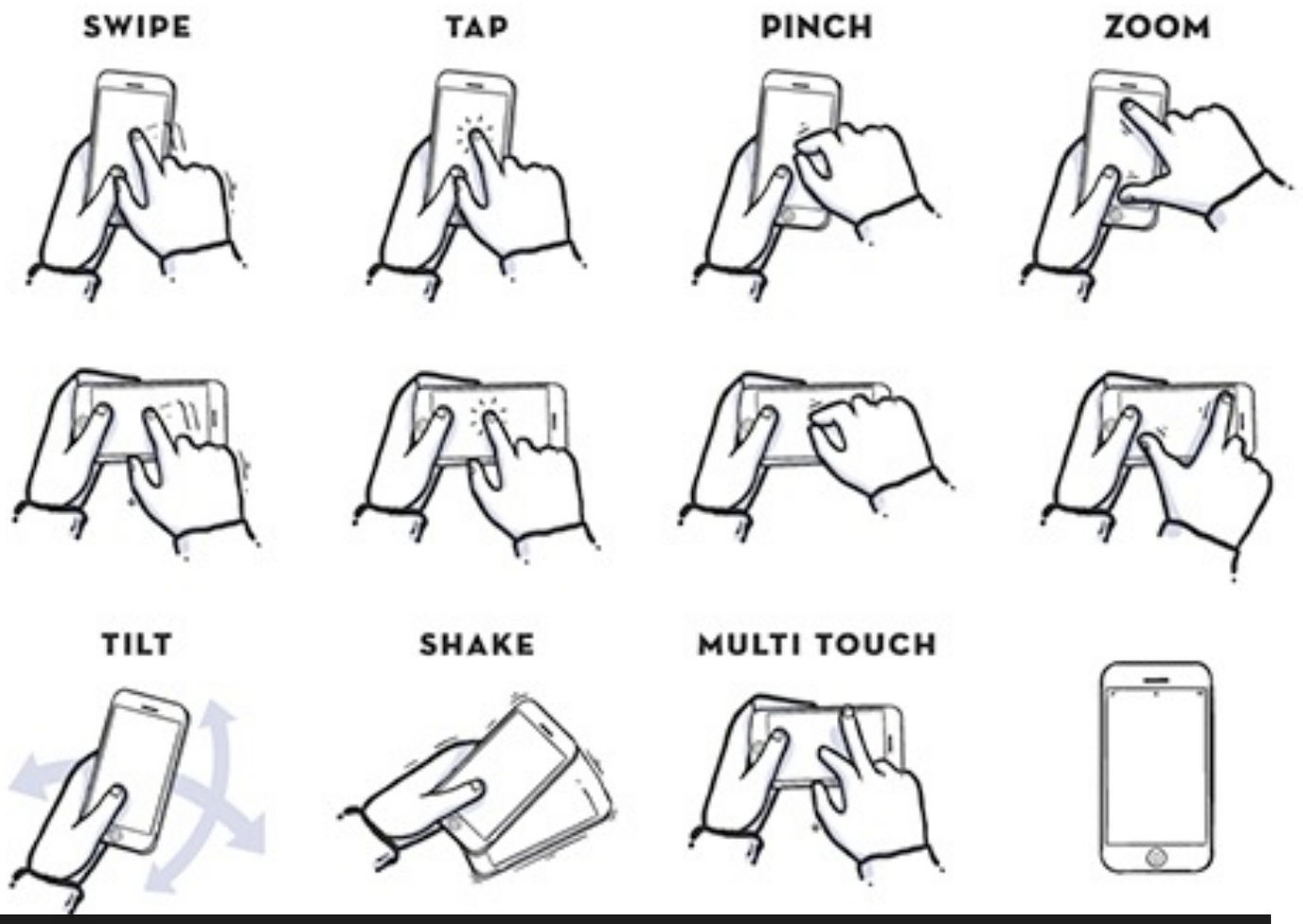
Converts a value in a specified range into a value in another range

4. map

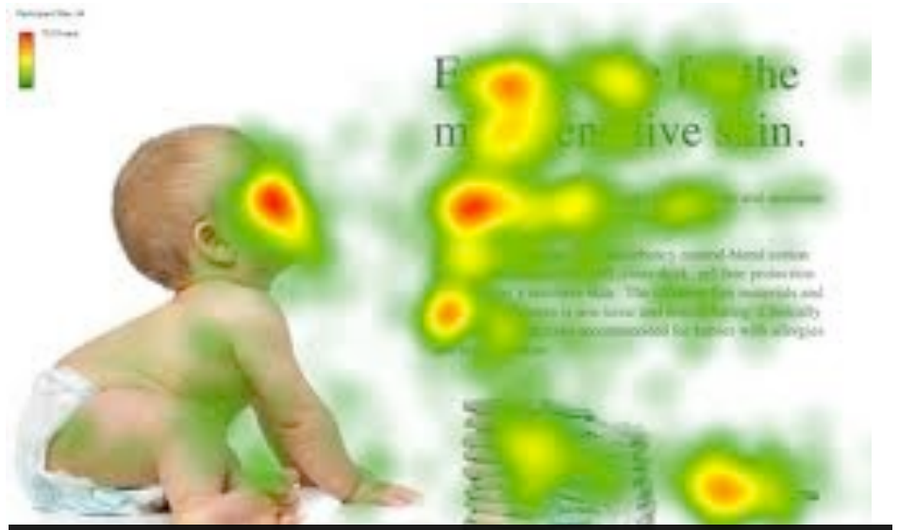
Converts a value in a specified range into a value in another range

```
float map(  float value,
            float srcStart, float srcStop,
            float dstStart, float dstStop) {

            return dstStart + (dstStop - dstStart) * norm(value,
        srcStart, srcStop) ;
}
```

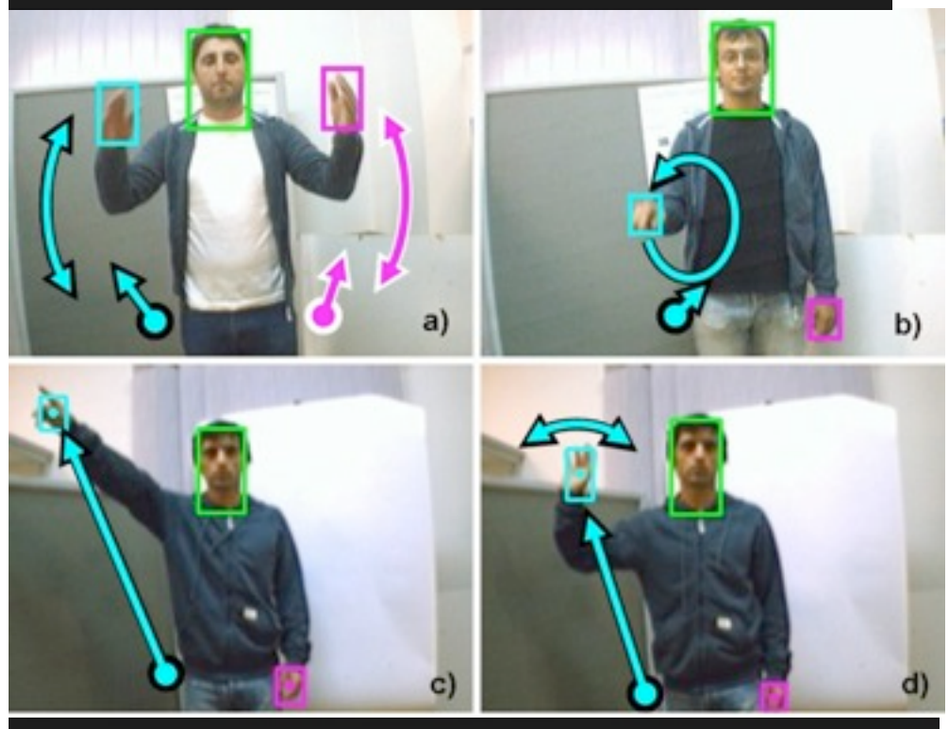
Mapping is a conceptual GIANT



Touch Events and Motion Events: *Phone & Tablet*



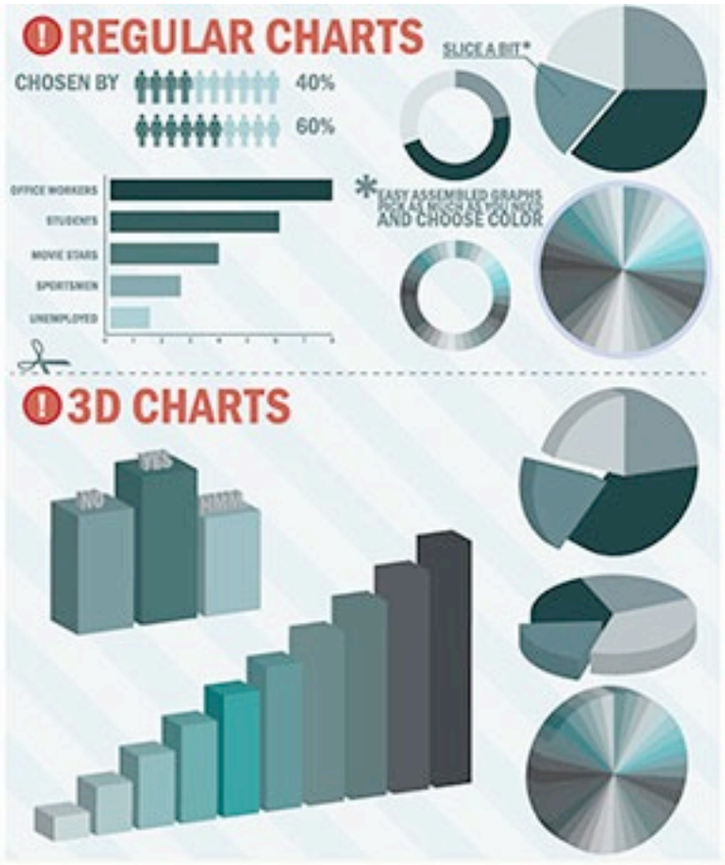
Eye Tracking: *Google Glass*



Gesture Tracking: *Kinect*

Dynamic Mapping

Mapping is a conceptual GIANT



Infographics

norm, lerp, clamp, map In Action

Android Task-switcher and the drag to dismiss interaction

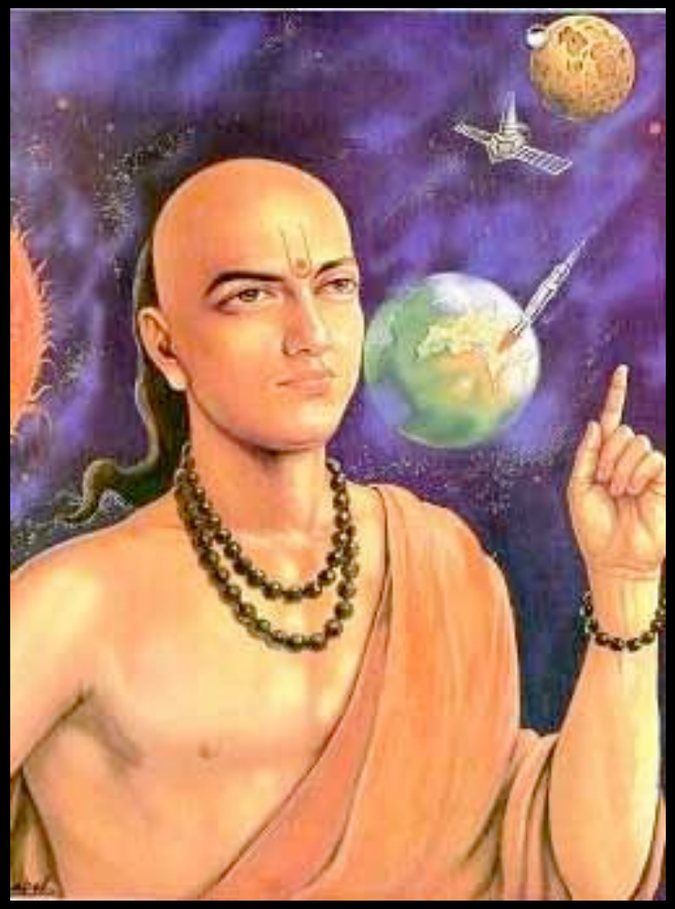
Drag

```
class DragActivity extends Activity {

	private View dragView;
	private int threshold;
	private float originY;
	private float movementY;


	@Override
	public boolean onTouchEvent(MotionEvent event) {
		int action = event.getActionMasked();

		switch (action) {
		case MotionEvent.ACTION_DOWN:
			originY = event.getY();
			break;
		case MotionEvent.ACTION_MOVE:
			movementY = event.getY() - originY;
			moveView(movementY);
			break;
		case MotionEvent.ACTION_UP:
		case MotionEvent.ACTION_CANCEL:
			movementEnded();
			break;
		default:
			break;
		}

		return super.onTouchEvent(event);
	}
```

getAlphaForMovement

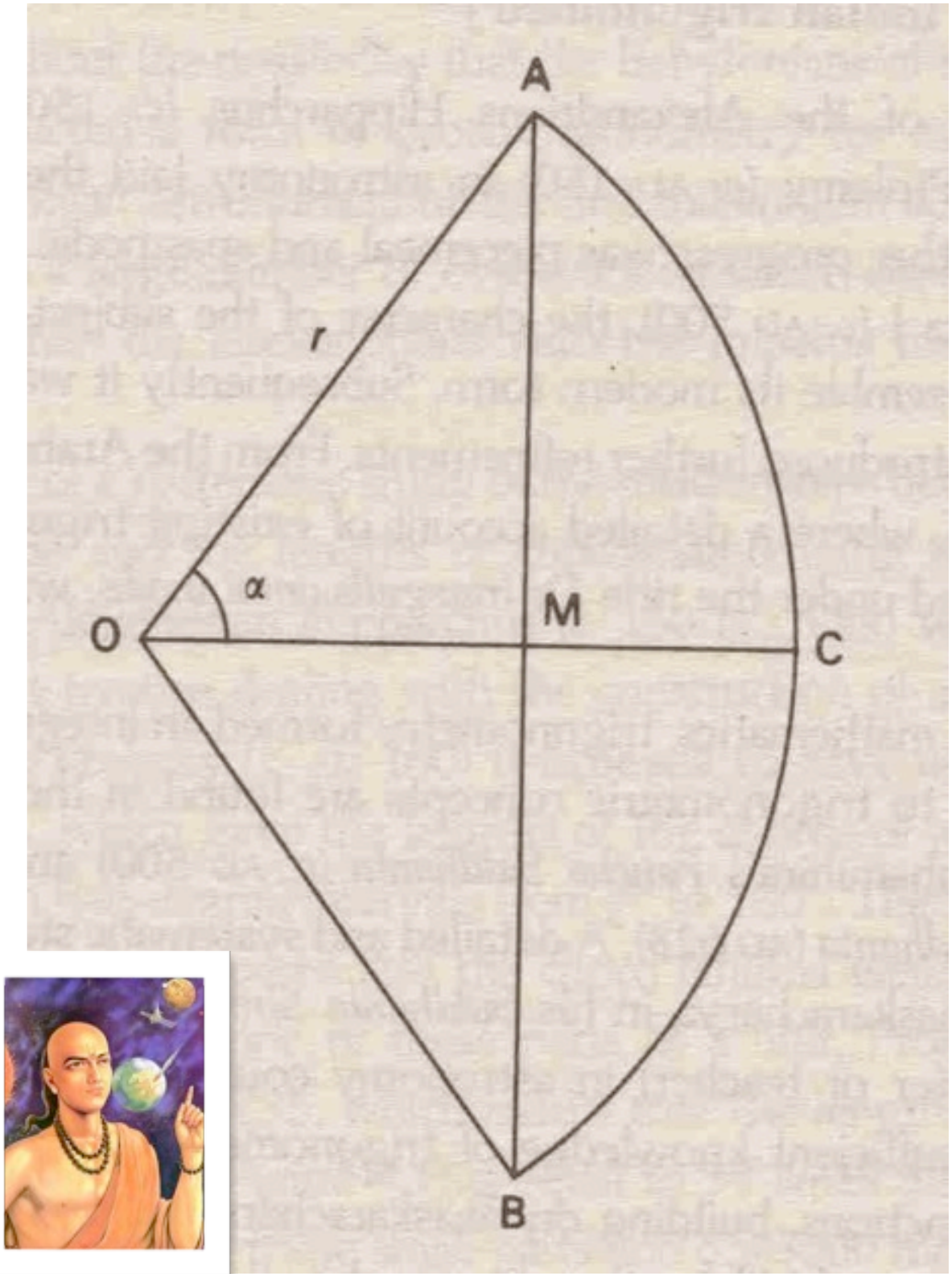
```
private void moveView(float movementY) {
        dragView.setTranslationY(movementY);
        dragView.setAlpha(getAlphaForMovement(movementY));
}
```

```
    /**
     * Returns an alpha transparency value corresponding to the view displacement.
     *
     * @param movementY
     * @return
     */
    private float getAlphaForMovement(float movementY) {
        threshold = getHeight() / 4;
        float movementYAbsValue = Math.abs(movementY);
        return 1.0f - Range.map(movementYAbsValue, 0.0f, threshold, 0.0f, 1.0f,
true /*clamp*/);

    }
```



Remember “my grandfather” who invented trigonometry?

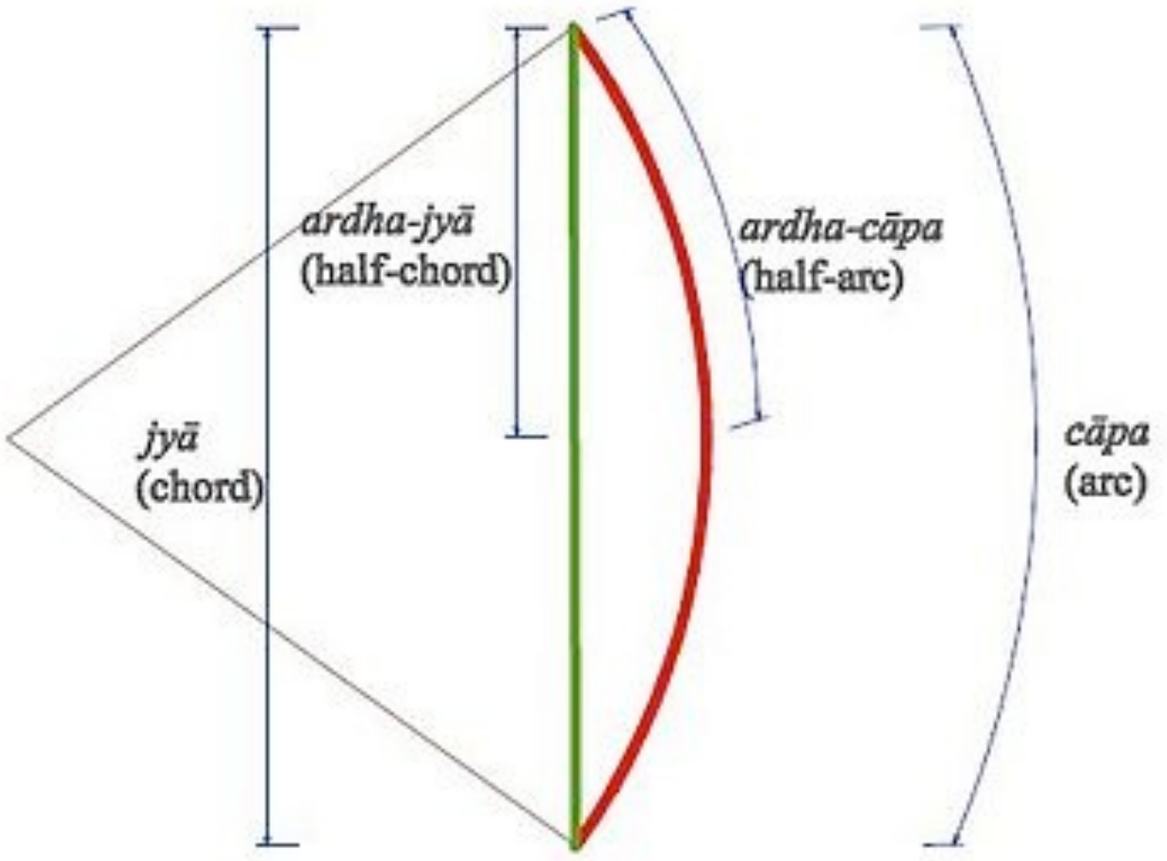
The story behind Sine Curves are SEXY



- **jya** = chord (sanskrit)
- **ardh-jya** = half chord (sanskrit)
- **jiva** (sanskrit)
- **jiyba** written as **jyb** (arabic)
- Interpreted by Europeans as **jAyb** = **bosom**
- **Sinus** = bosom (latin)
- **Sine**

If you don’t get distracted by the latin connotations of sine and stick to the original in sanskrit...

If you don’t get distracted by the latin connotations of sine and stick to the original in sanskrit...



Sine = **Chord** = Y axis

Cosine = **Arrow** = X axis

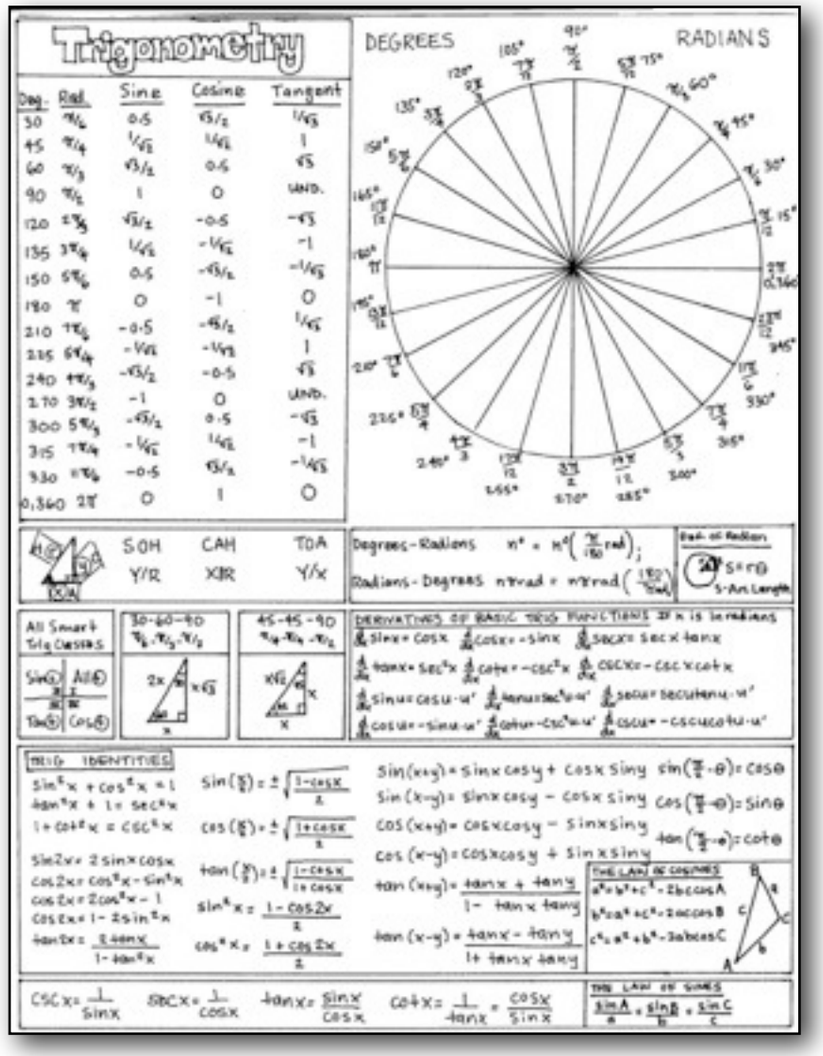
Trigonometry is great at answering 2 questions:

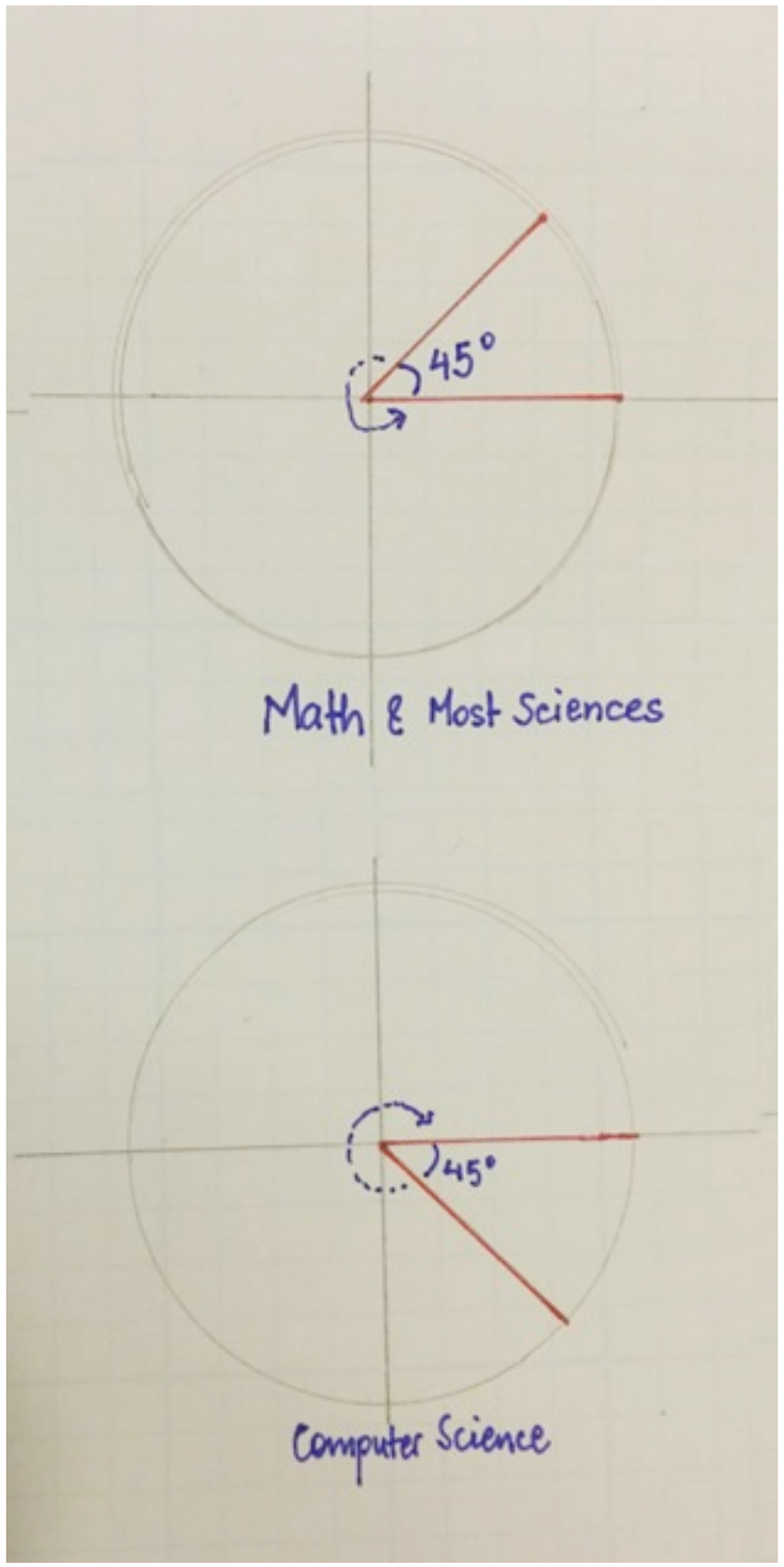
1.

Given an **angle**, what is the X and Y **coordinate**?

2.

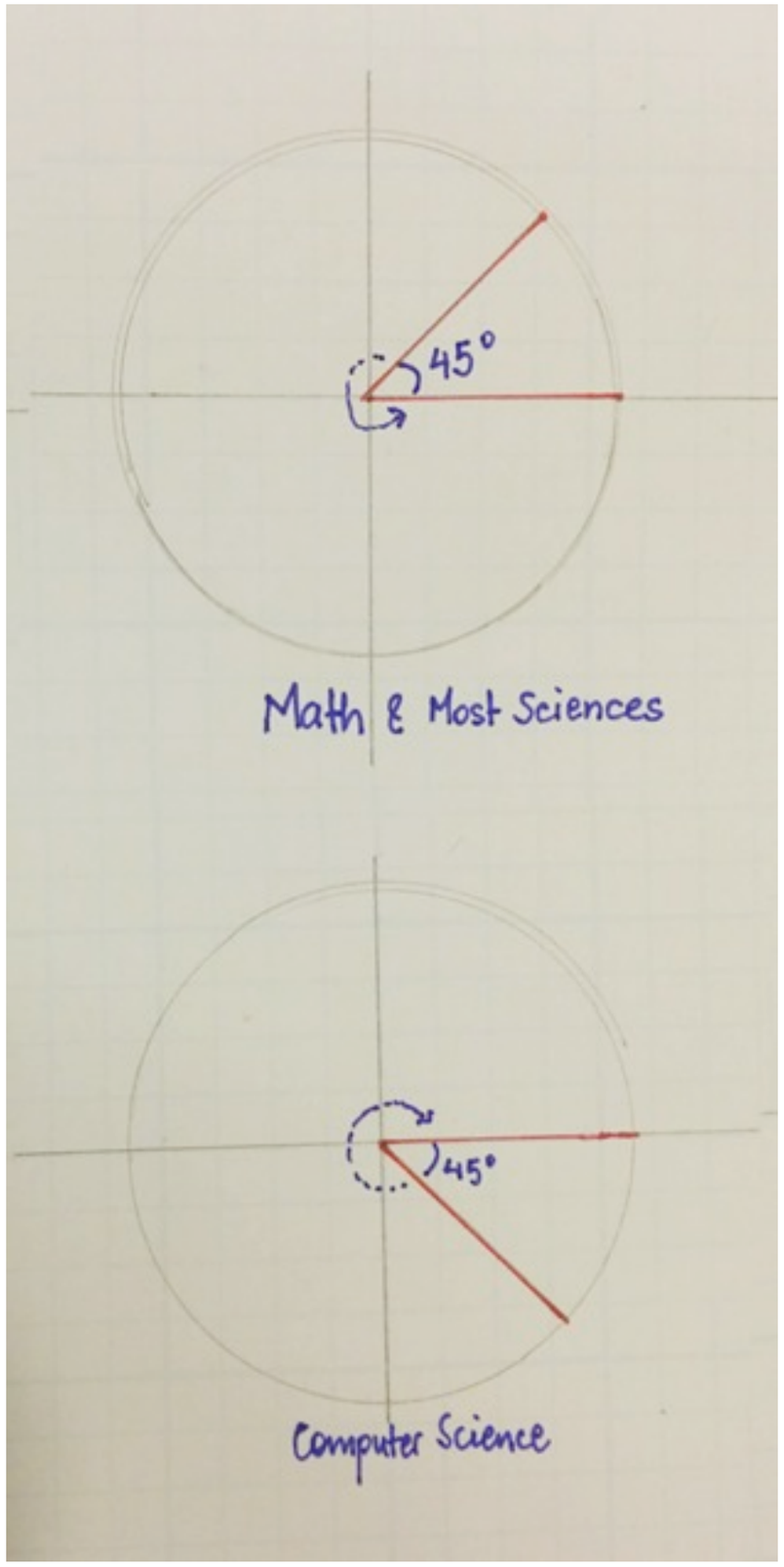
Given a **coordinate**, what is the **angle** it makes to the horizontal?



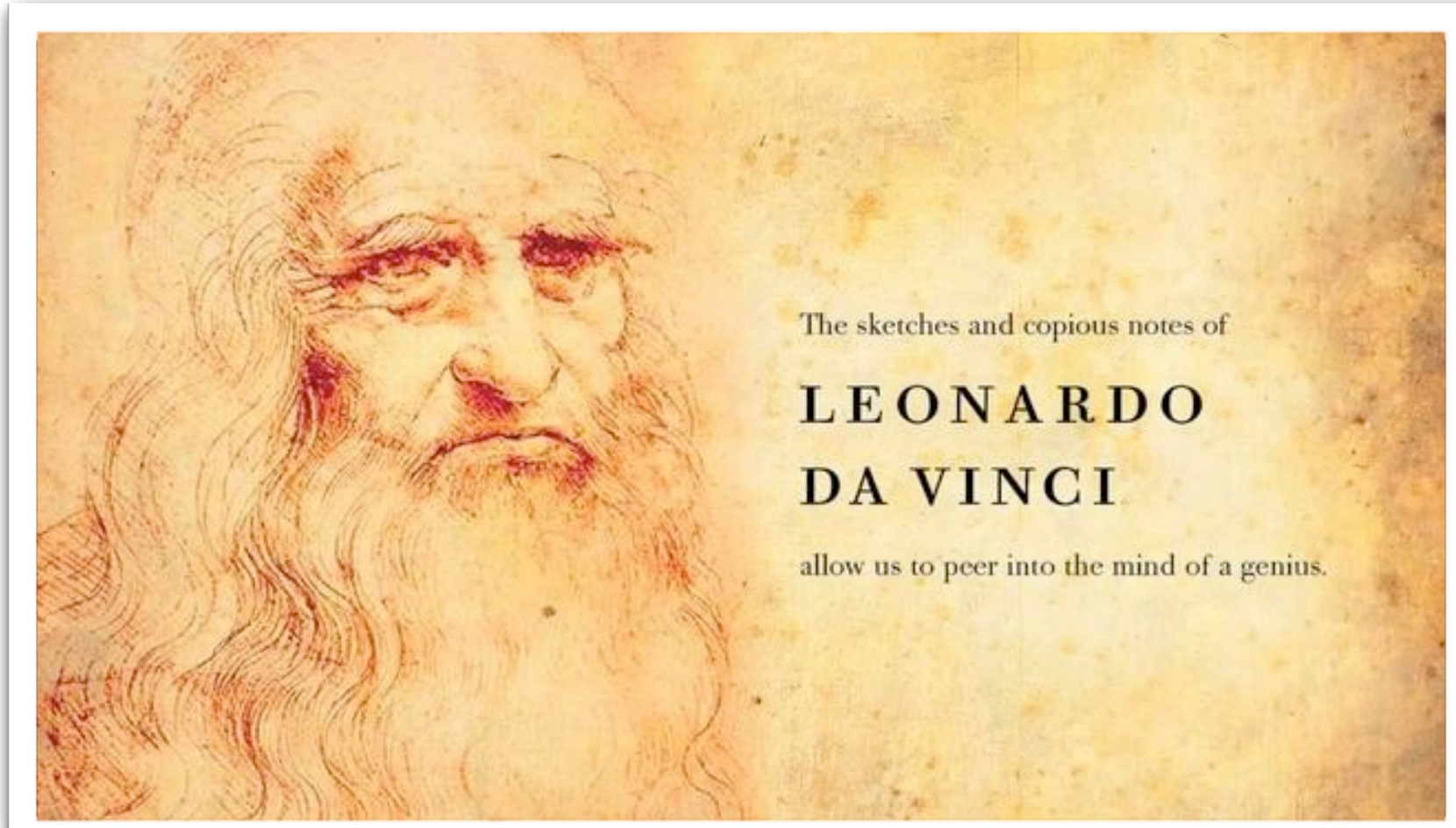


Everything in UI Programming is Upside Down

Everything in UI Programming is Upside Down



Everything in UI Programming is Upside Down

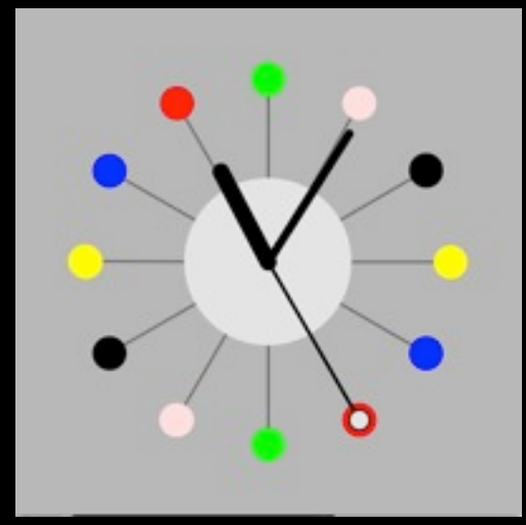


Given an **angle**, what are the **x** and **y** coords?

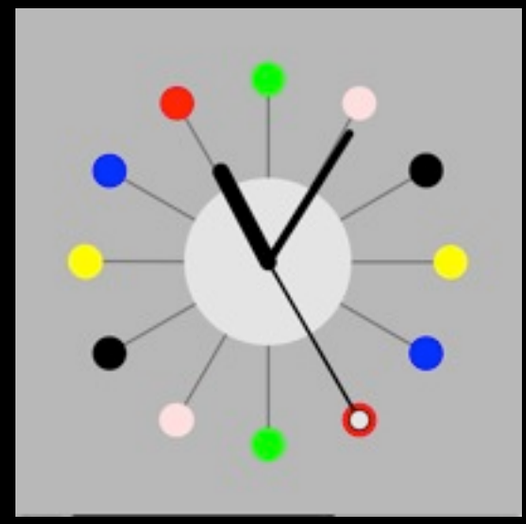
An analog clock
Demo

Inspiration: Dallas Museum of Art

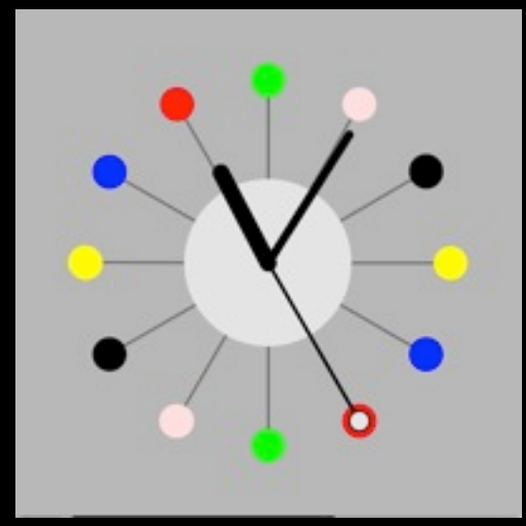
Given an **angle**, what are the **x** and **y** coords?



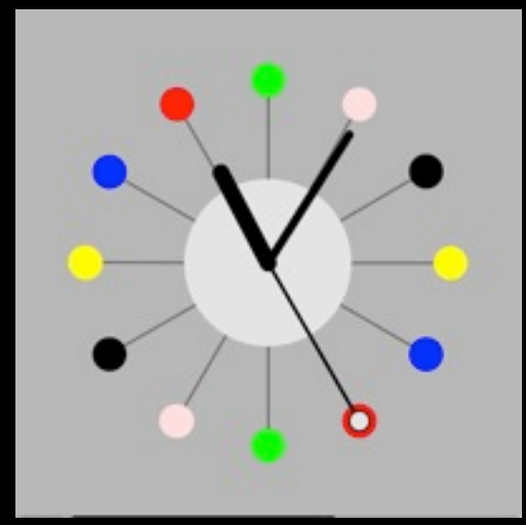
```
 1 ...
 9
10 void setup() {
11   size(600, 600);
12
13   int radius = min(width, height) / 2;
14   secondsRadius = radius * 0.72;
15   minutesRadius = radius * 0.60;
16   hoursRadius = radius * 0.4;
17   diskRadius = radius * 0.33;
18
19   cx = width / 2;
20   cy = height / 2;
21 }
22
```



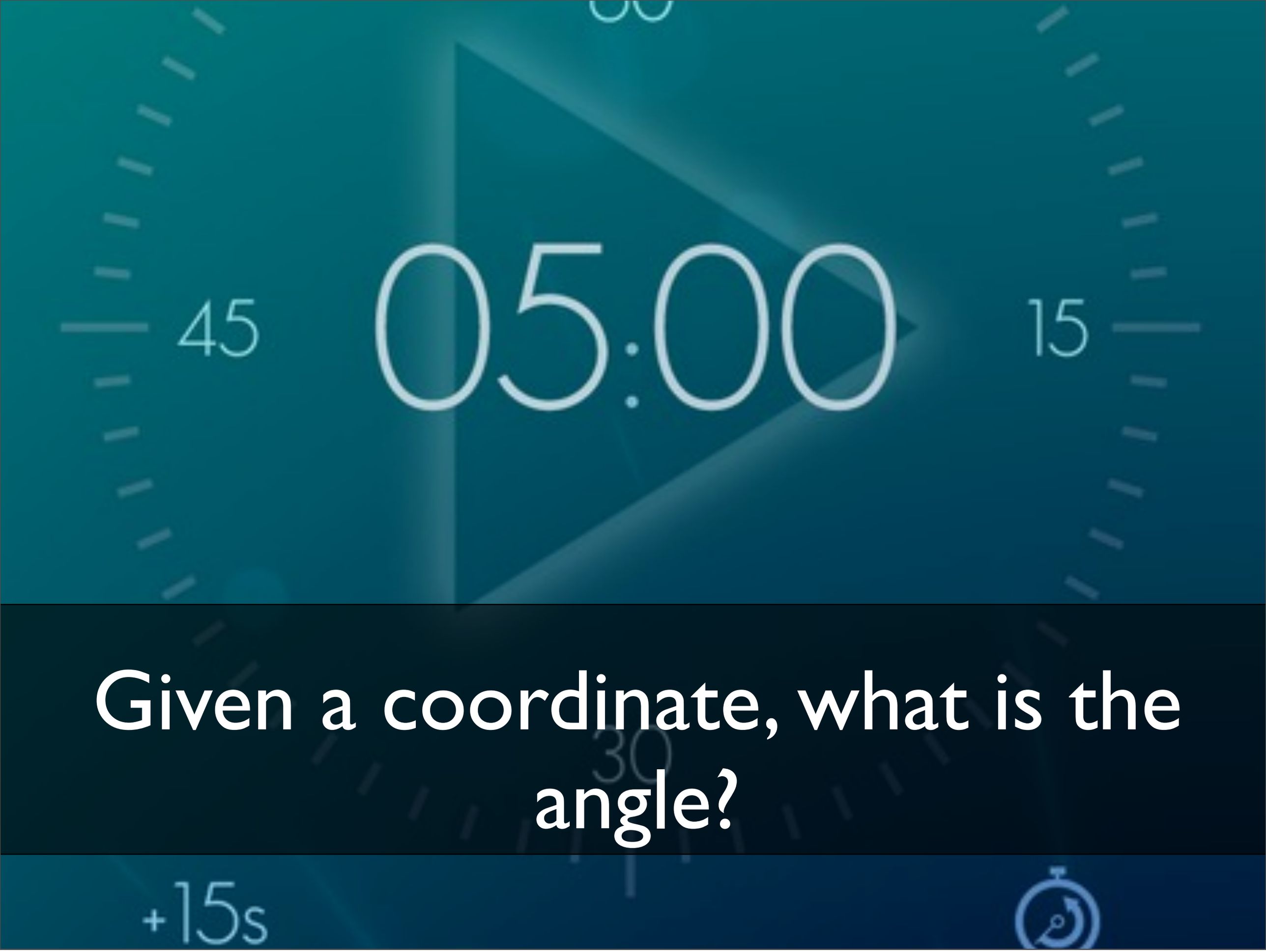
```
 1 ...
23 void draw() {
24   background(175, 175, 175);
25
26   drawTicks();
27
56   ...
57 }
58
59 void drawTicks() {
60   //draw the 12 ticks as colorful circles
61   beginShape(POINTS);
62   for (int a = 0; a < 12; a+=1) {
63     float angle = radians(a*30);
64
65     stroke(100);
66     strokeWeight(3);
67     line(cx, cy, cx + cos(angle) * secondsRadius, cy + sin(angle) * secondsRadius);
68
69     stroke(colors[a % 6]);
70     strokeWeight(40);
71
72     float x = cx + cos(angle) * secondsRadius;
73     float y = cy + sin(angle) * secondsRadius;
74
75     vertex(x, y);
76
77   }
78   endShape();
79 }
80
```



```
 1 ...
22
23 void draw() {
24   background(175, 175, 175);
25
26   drawTicks();
27
28   ...
33
34   // Angles start at 3 o'clock so subtract HALF_PI to make them start at the top
35   float s = map(second(), 0, 60, 0, TWO_PI) - HALF_PI;
36   float m = map(minute() + norm(second(), 0, 60), 0, 60, 0, TWO_PI) - HALF_PI;
37   float h = map(hour() + norm(minute(), 0, 60), 0, 24, 0, TWO_PI * 2) - HALF_PI;
38
39   ...

57 }
```



```
 1 ...
22
23 void draw() {
24   ...
40
41   //seconds
42   strokeWeight(4);
43   line(cx, cy, cx + cos(s) * secondsRadius, cy + sin(s) * secondsRadius);
44
45   //circle st the end of the seconds hand
46   strokeWeight(2);
47   ellipse(cx + cos(s) * secondsRadius, cy + sin(s) * secondsRadius, 12, 12);
48
49   //minutes
50   strokeWeight(10);
51   line(cx, cy, cx + cos(m) * minutesRadius, cy + sin(m) * minutesRadius);
52
53   //hours
54   strokeWeight(20);
55   line(cx, cy, cx + cos(h) * hoursRadius, cy + sin(h) * hoursRadius);
56
57 }
```



Given a coordinate, what is the angle?

Inspiration: Timely Timer

Given a coordinate, what is the angle?

soh cah toa

ArcTangent

is the answer to the question!

Inspiration: Timely Timer

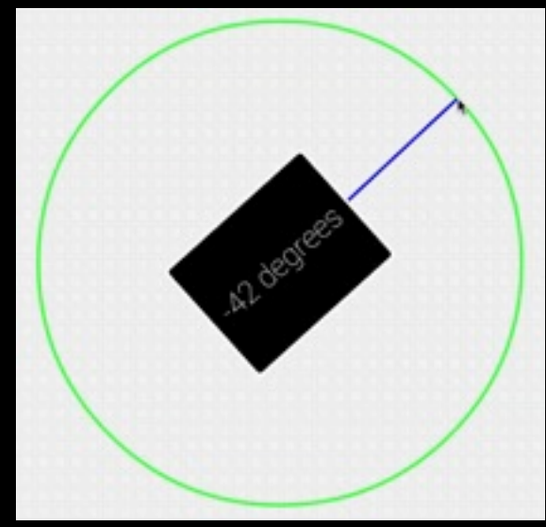
Given a coordinate, what is the angle?

double angle = Math.atan2(dy, dx)

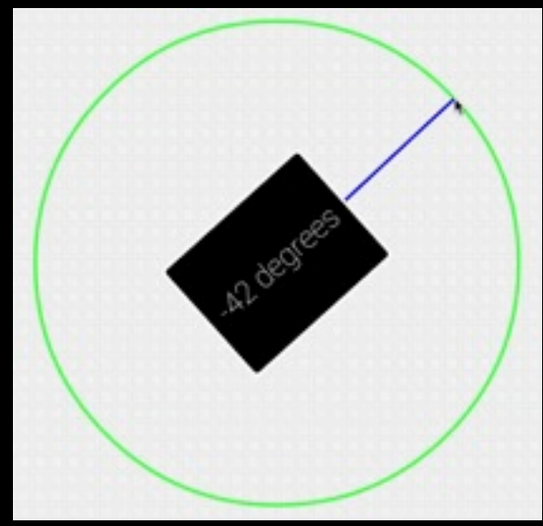
Inspiration: Timely Timer

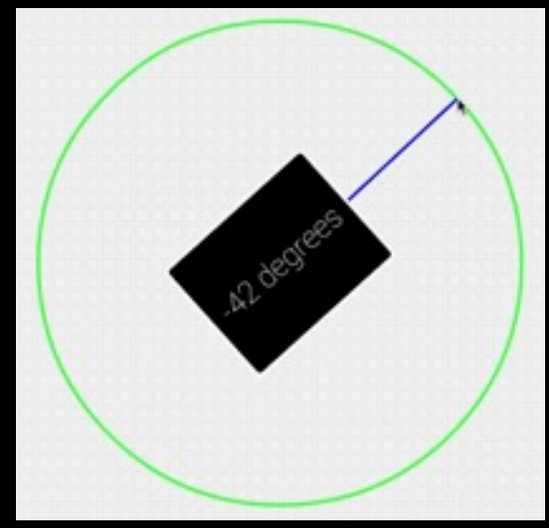
Given a coordinate, what is the angle?

```
13 public class FollowTheFingerActivity extends Activity {
14
15    private TextView rotateView;
16    private float currentFingerX;
17    private float currentFingerY;
18    private int locOnScreen[];
19    private FollowTheFingerView followFingerView;
20
21    @Override
22    protected void onCreate(Bundle savedInstanceState) {
23        super.onCreate(savedInstanceState);
          ...
37        init();
          ...
41    }
42
43    private void init() {
44        rotateView.getLocationOnScreen(locOnScreen);
45        locOnScreen[0] += rotateView.getWidth()/2;
46        locOnScreen[1] += rotateView.getHeight()/2;
47        setRotationText(0);
48    }
```



```
51      @Override
52      public boolean onTouchEvent(MotionEvent event) {
53          int action = event.getActionMasked();
54
55          switch (action) {
56          case MotionEvent.ACTION_DOWN:
57          case MotionEvent.ACTION_MOVE:
58              currentFingerX = event.getX();
59              currentFingerY = event.getY();
60              rotateView(currentFingerX, currentFingerY);
61              break;
62          case MotionEvent.ACTION_UP:
63          case MotionEvent.ACTION_CANCEL:
64              movementEnded();
65              break;
66          default:
67              break;
68          }
69
70          if (followFingerView != null) {
71              followFingerView.onTouchEvent(event);
72          }
73          return true;
74      }
```



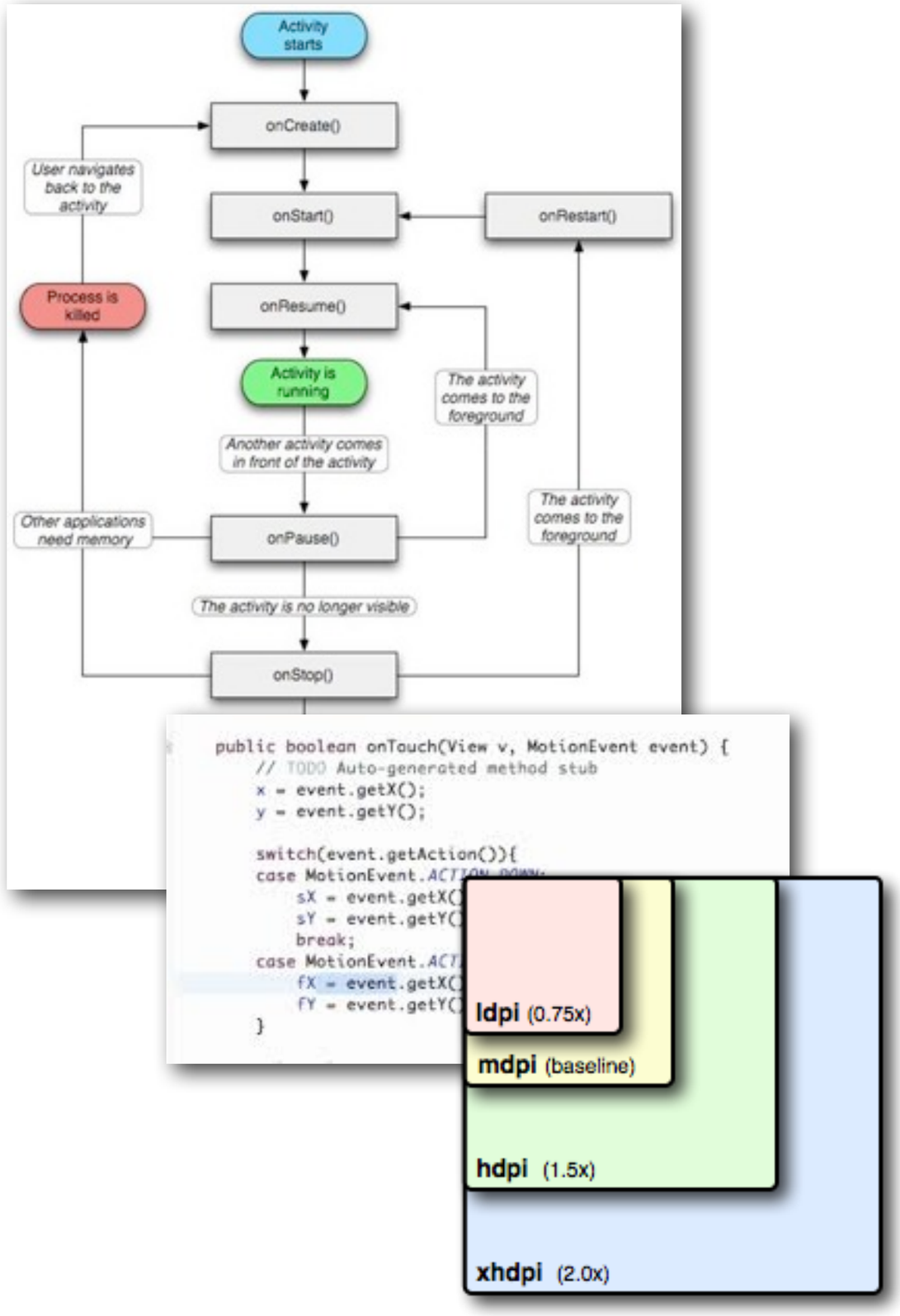



```
76    private void rotateView(float fingerX, float fingerY) {
77        int originX = locOnScreen[0];
78        int originY = locOnScreen[1];
79
80        int dx =  (int) currentFingerX - originX;
81        int dy  = (int) currentFingerY - originY;
82
83        double angleInRadians = 0;
84        double degrees = 0;
85
86        angleInRadians = Math.atan2(dy, dx);
87        degrees = Math.toDegrees(angleInRadians);
88
89        rotateView.setRotation((int) degrees);
90        setRotationText((int)degrees);
91    }
```

Prototyping and Sketching

Prototyping and Sketching

Android is powerful and capable platform



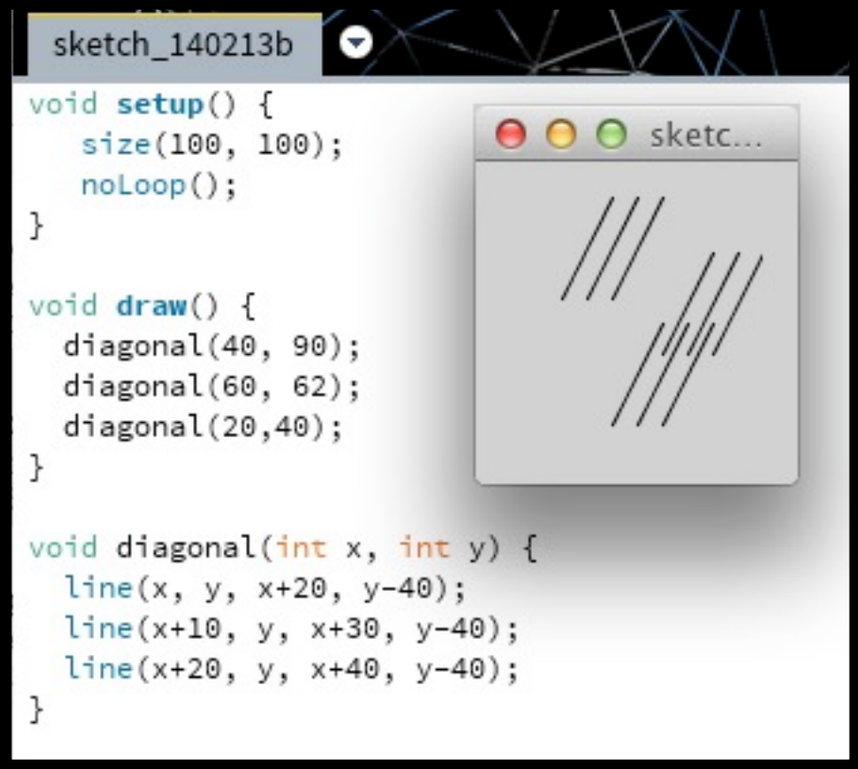
However, it is also steeped in complexity

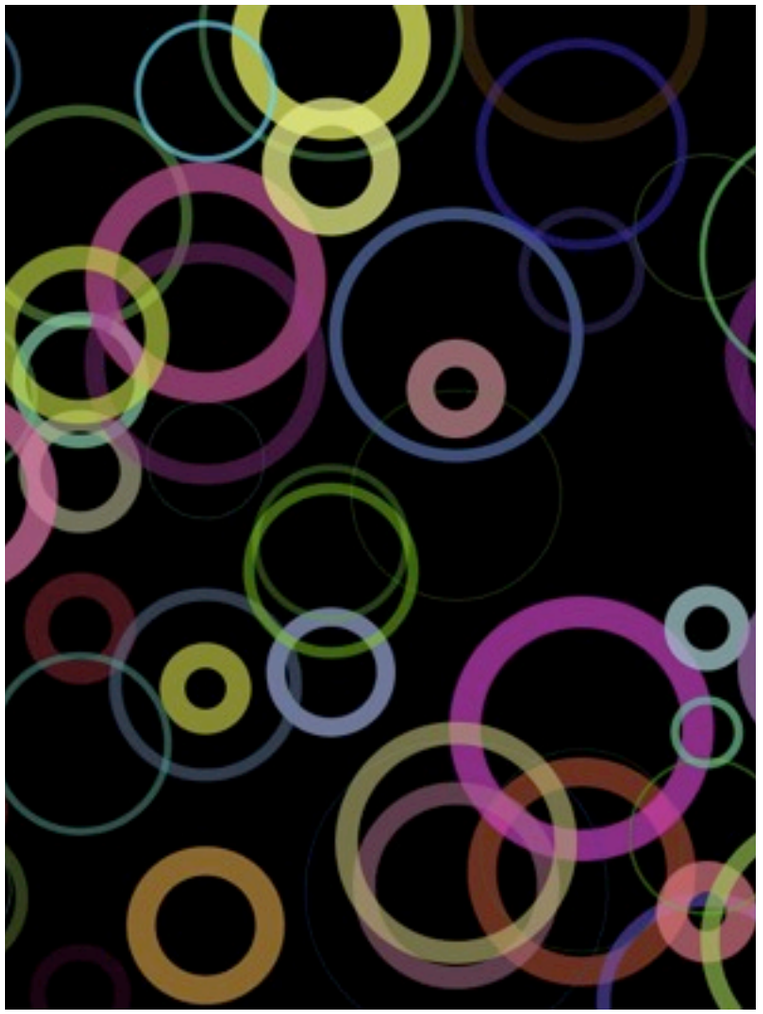
Simplicity is the essence of happiness.

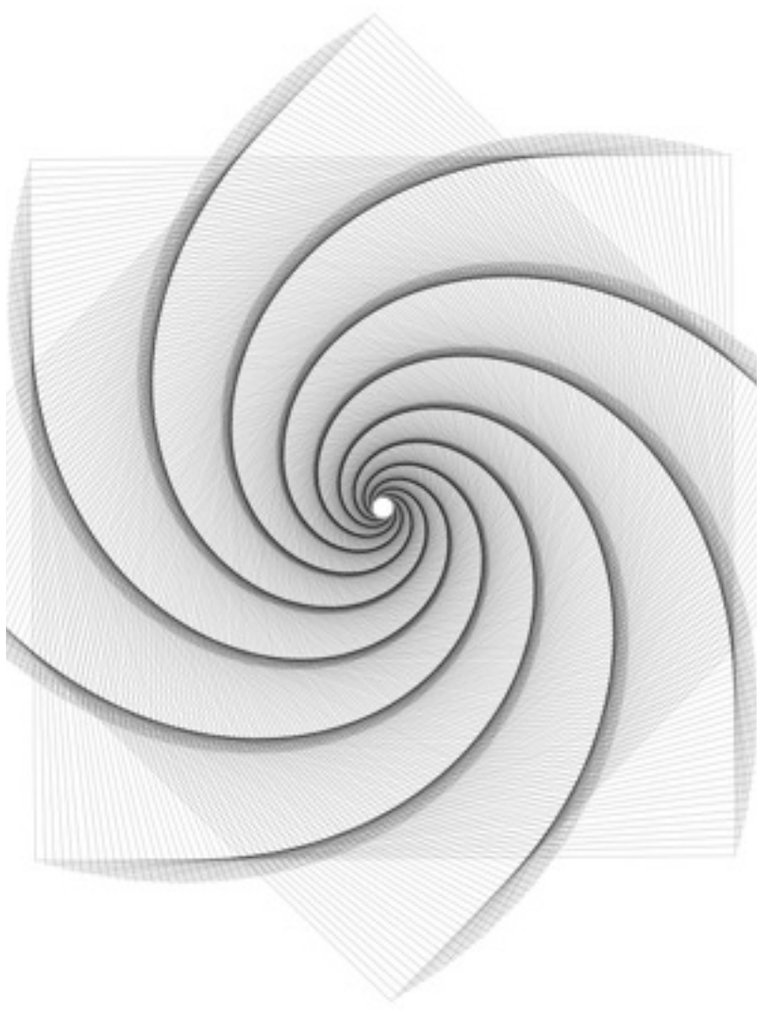
Cedric Bledsoe

Processing

- A open source language written at M.I.T.
- Written in Java but hides all boilerplate
- For artists and serious programmers
- Large community that loves to share







Simple to Sublime

Processing is as much a playground as it is a toy

Play in it while it interests and informs you, put it away when it's purpose has been served

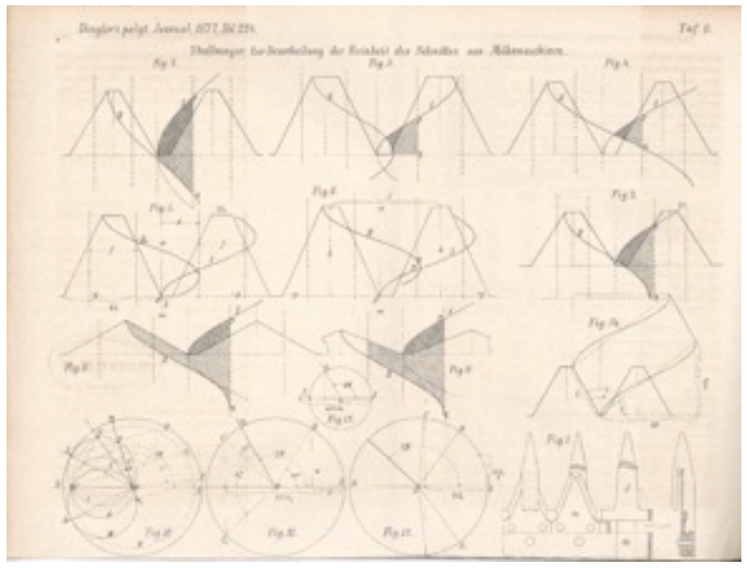
Summary

The “Problem”

Human Computer Interaction is getting richer as are **Consumer Expectations**

Summary

The “Problem”	Problem Solving Tools
Human Computer Interaction is getting richer as are **Consumer Expectations**	**Math** can unlock tools that help break down problems * **Algebra** * **Trigonometry**



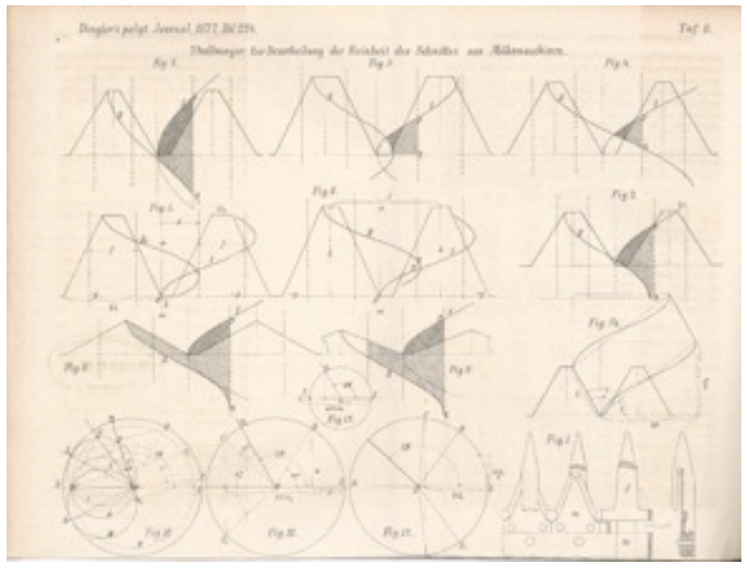
Summary

The “Problem”

Human Computer Interaction is getting richer as are **Consumer Expectations**

Problem Solving Tools

Math can unlock tools that help break down problems
* **Algebra**
* **Trigonometry**



Prototyping and Sketching is equally important for Computer Scientists as they are for artists and designers

Thanks

Nik Bhattacharya
co-founder
Hit Wicket Apps