openFrameworks!

about download documentation tutorials gallery community development

> forum > addons > wiki > github > mailing list > IRC

english / japanese

download

0.7.4 **is the most recent release. It has a lot of new features, new interfaces, and probably some new bugs too. 0.7.4 is not 100% compatible with older projects. Please see the changelog to get an overview of the differences between versions.**

To use openFrameworks you will need an IDE, and the setup guide for your platform can walk you through this. Please post any bugs on the issues page, and post to the forum if you have any other questions.
openFrameworks is distributed under the MIT License.

What is openFrameworks?

download
openFrameworks for
xcode

IDE setup guide
xcode

download
openFrameworks for
code::blocks
code::blocks (64 bit)

IDE setup guide
code::blocks
eclipse

download
openFrameworks for
code::blocks
visual studio 2010

IDE setup guides
code::blocks
visual studio 2010

ios

osx only

download
openFrameworks for
xcode

IDE setup guide
xcode

android

linux + osx only

download
openFrameworks for
eclipse

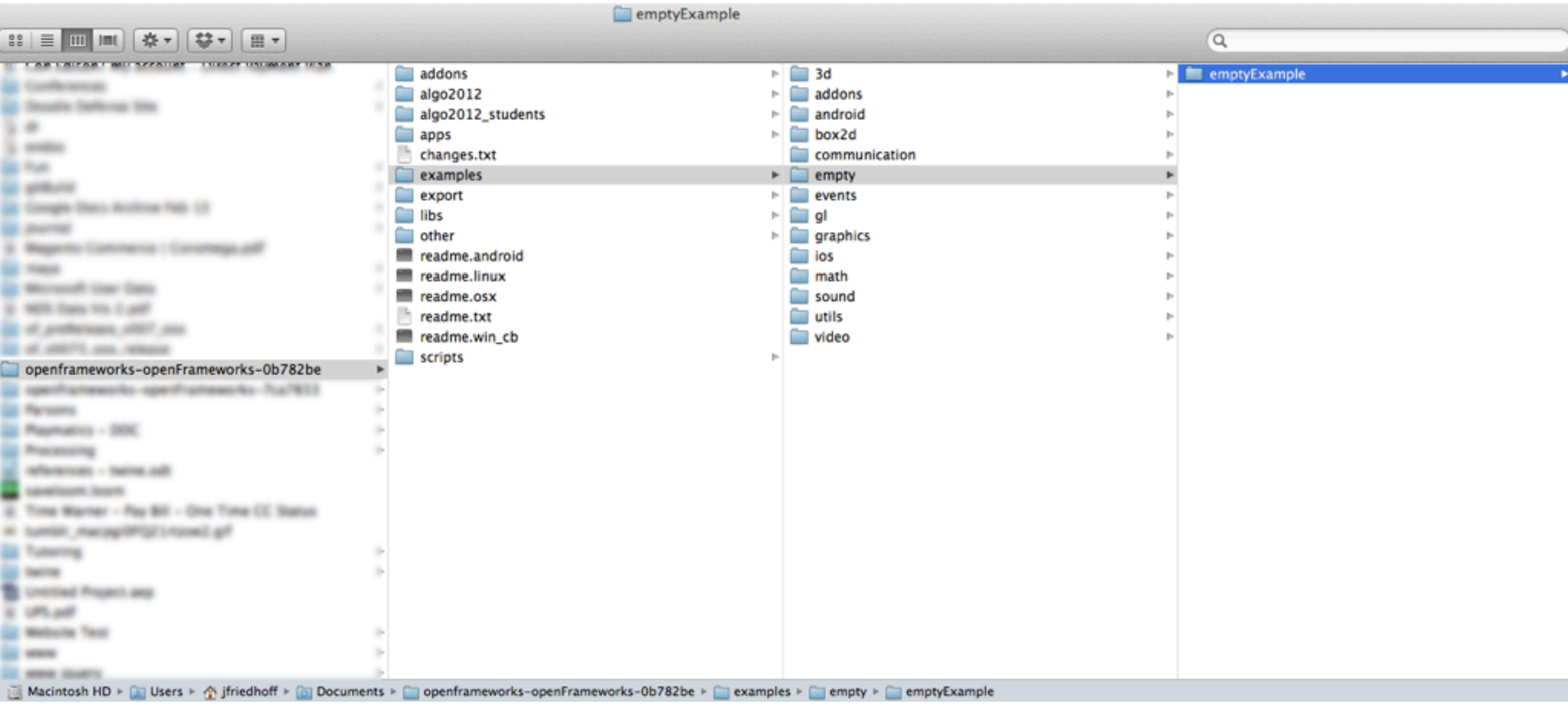
IDE setup guide
eclipse

What is openFrameworks?

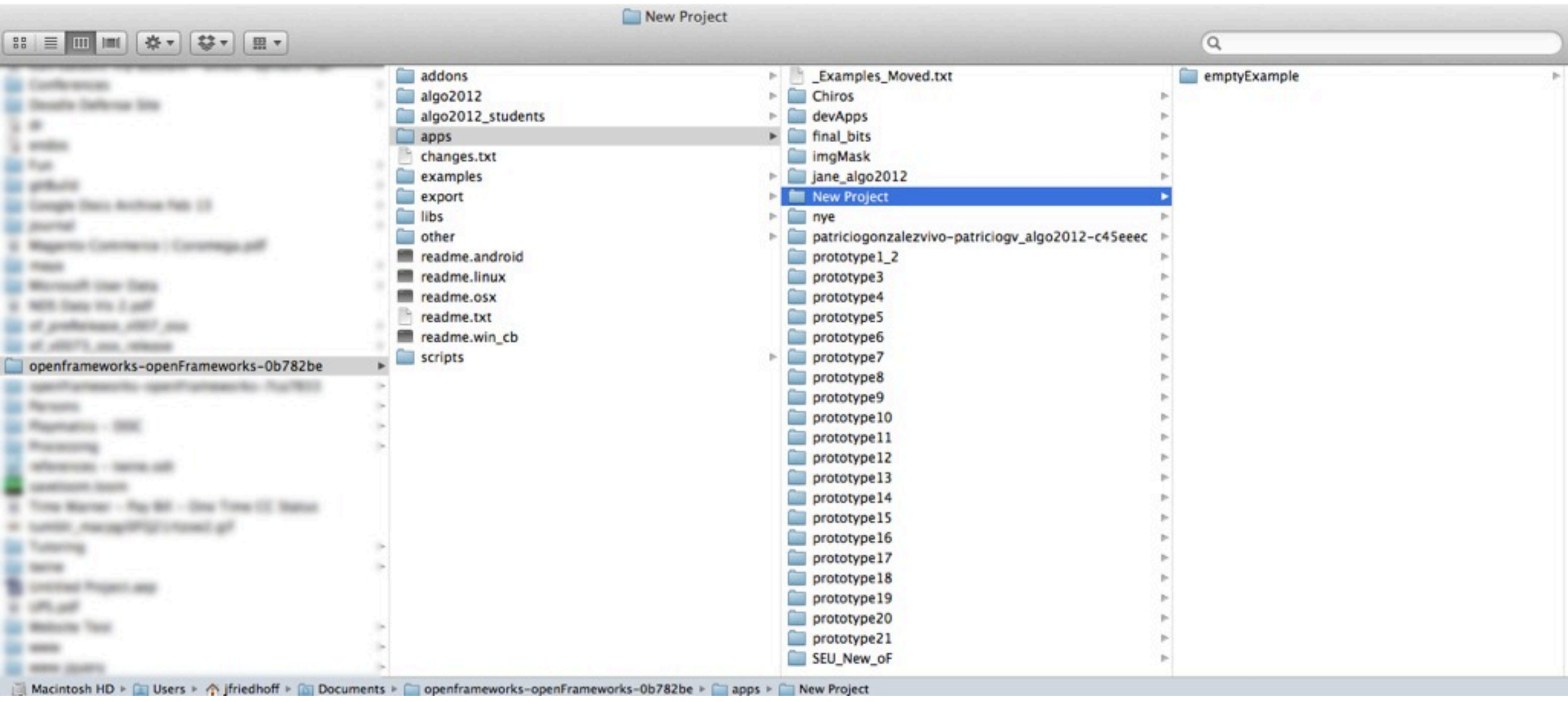
- oF is a software framework for C++
- Software framework: a prefab software infrastructure designed to provide low-level functionality

How do I make a new project?

Part 1: Copy emptyExample



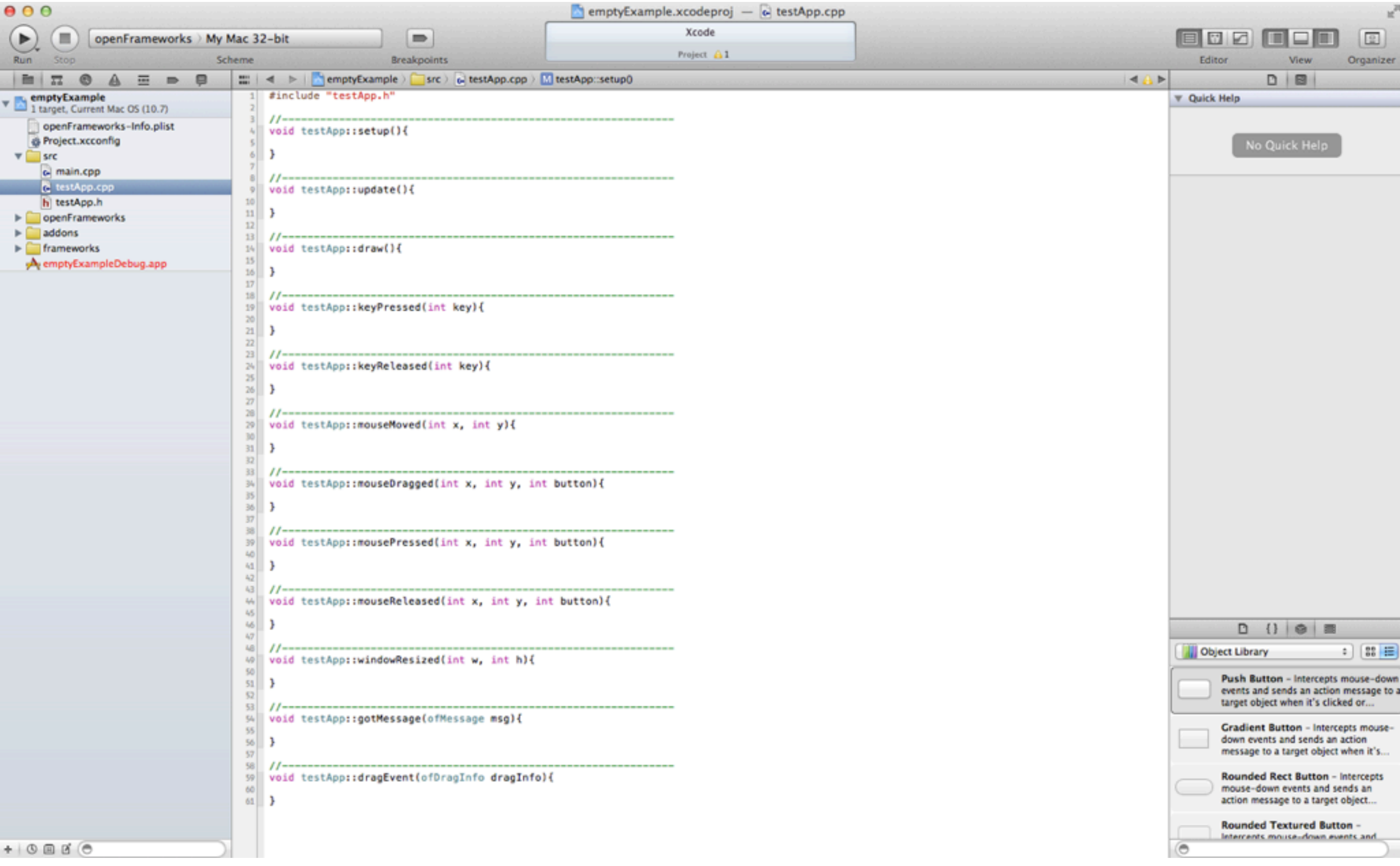
Part 2: Make a new folder in “apps” and paste it there



base folder | level 1 | level 2 | level 3

(Your project must be exactly three levels below your base oF folder!)

Part 3: Open the project



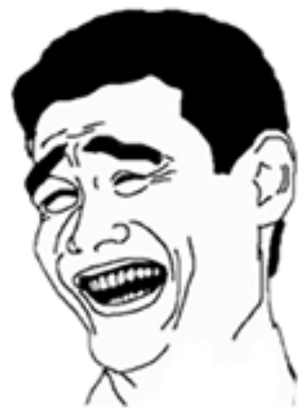
Can I take notes in my code?

le coding

```
Imports System

Public Class Employee
    Private _name As St
    Private _salary As I

    Public ReadOnly Pro
        Get
            Return _nam
        End Get
    End Property

    Public ReadOnly Pro
        Get
            Return _sal
        End Get
    End Property

    Public Sub Increase
```

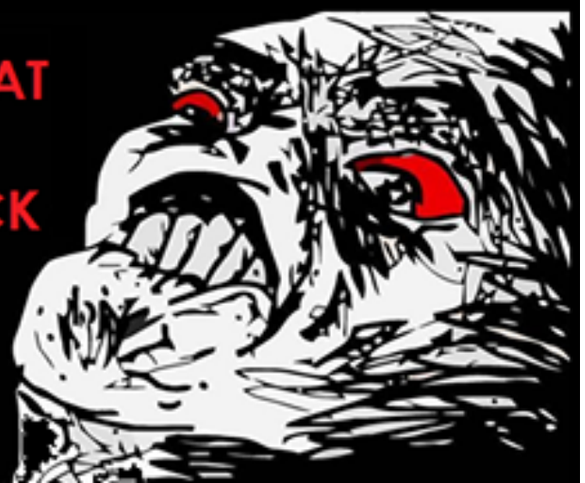
Enough for today, saving time!

Always put enough comments in your code!



Opening file 6 weeks later...

WHAT THE FUCK



```
// This is a one-line comment.

/*
    This comment can span multiple lines.
    Check me out, taking up all the space.
    Echo...
        echo...
            echo...
                echo...
                    echo...
                        echo...
*/
```

Commenting Code

- DO IT.
- Seriously, do it.
- Use two forward slashes (//) for a one-line comment.
- You can also do multi-line comments:
  - Preface your comment with slash-asterisk (/*)
  - End it with asterisk-slash (*/)


thousands of lines and zero comments
merry christmas and a happy bus accident
theprofoundprogrammer.tumblr

Scheme Breakpoints Project ⚠1

emptyExample › src › testApp.cpp › M testApp::setup()

```
#include "testApp.h"

//--------------------------------------------------------------
void testApp::setup(){

}

//--------------------------------------------------------------
void testApp::update(){

}

//--------------------------------------------------------------
void testApp::draw(){

}

//--------------------------------------------------------------
void testApp::keyPressed(int key){

}

//--------------------------------------------------------------
void testApp::keyReleased(int key){

}

//--------------------------------------------------------------
void testApp::mouseDragged(int x, int y, int button){

}

//--------------------------------------------------------------
void testApp::mousePressed(int x, int y, int button){

}

//--------------------------------------------------------------
void testApp::mouseReleased(int x, int y, int button){

}

//--------------------------------------------------------------
void testApp::windowResized(int w, int h){

}

//--------------------------------------------------------------
void testApp::gotMessage(ofMessage msg){

}

//--------------------------------------------------------------
void testApp::dragEvent(ofDragInfo dragInfo){

}
```

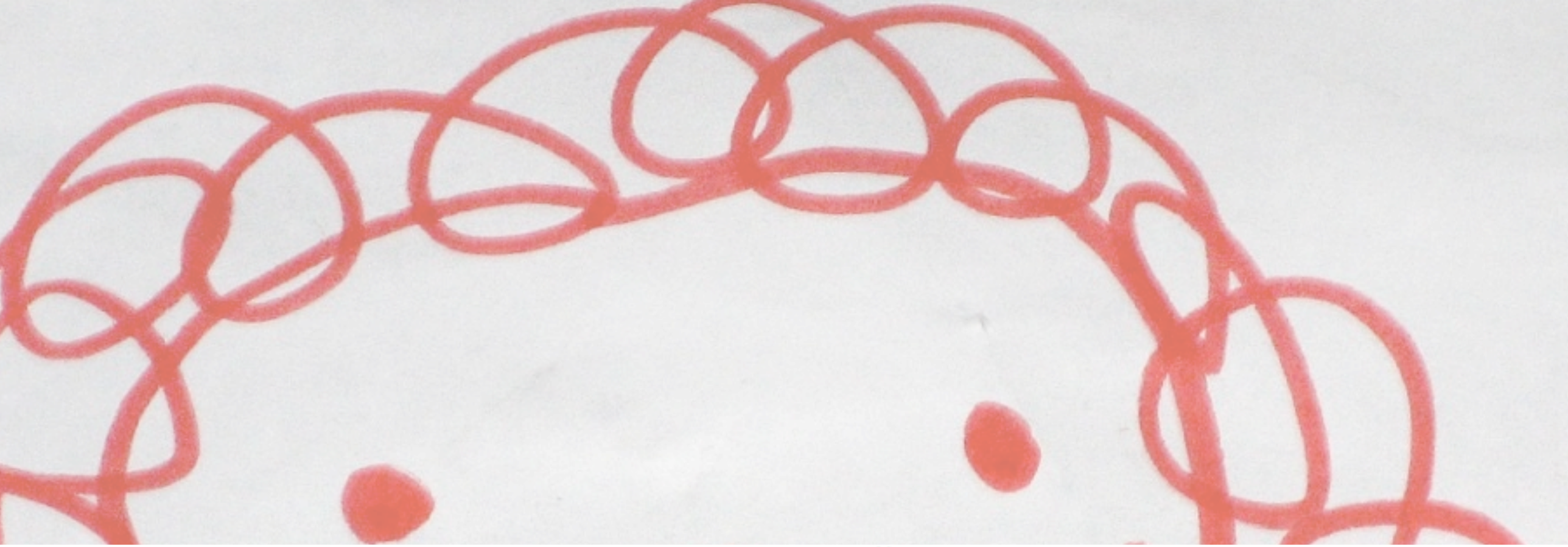
What's going on in testApp.cpp?

void testApp::setup() {}

```
//--------------------------------------------------------------
void testApp::setup(){

}
```

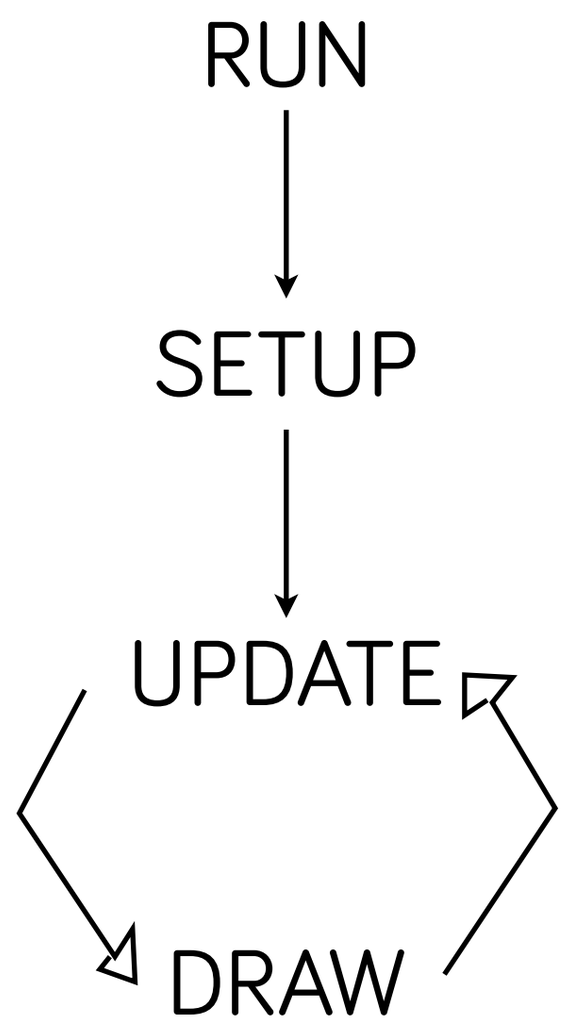
void testApp::update() {}

```
//--------------------------------------------------------------
void testApp::update(){

}
```



void testApp::draw() {}

```
//--------------------------------------------------------------
void testApp::draw(){

}
```


RUN
SETUP
UPDATE
DRAW

Main oF Functions

- setup(): runs just once when the app starts
  - Good place to set initial values for variables, e.g. playerHealth = 100;
- update(): runs once per frame, before draw()
  - Good place for number-crunching, e.g. playerHealth -= 1;
- draw(): runs once per frame, after update()
  - Where you should put all your drawing, e.g. player.draw();

Listeners

Other oF Functions

- oF will listen for certain events, and when they happen, it will run whatever code is in the corresponding event

- E.g. keyPressed() runs at the moment that a key is pressed (not held!)

How do I draw stuff?

Functions

- Function: a named section of a program that does a specific task
- Wraps up code in an easy-to-reference way
- Parameter: additional information you can give the function to change the output

Bake me a chocolate cake!

Baking a cake:
the action

Chocolate:
additional info
that affects
the action

Function Structure

```
bake_me_a_cake(chocolate);
```

- Name of the function
- Parentheses: delinate that it's a function, hold arguments
- Semicolon: end of line, move onto the next thing

Wait, that doesn't answer my question about drawing stuff!

Drawing Shapes

Function	Notes
ofCircle(x, y, radius);	x and y are at center by default
ofRect(x, y, width, height);	x and y are at upper-left corner by default
ofLine(x1, y1, x2, y2);	-
ofTriangle(x1, y1, x2, y2, x3, y3);	-

Compare the function with what's inside it.
Which would you rather type?

```
ofCircle(enemyX, enemyY, enemyRadius);
```

```
//----------------------------------------------------------
void ofGLRenderer::drawCircle(float x, float y, float z,  float radius){
    vector<ofPoint> & circleCache = circlePolyline.getVertices();
    for(int i=0;i<(int)circleCache.size();i++){
        circlePoints[i].set(radius*circleCache[i].x+x,radius*circleCache[i].y+y,z);
    }

    // use smoothness, if requested:
    if (bSmoothHinted && bFilled == OF_OUTLINE) startSmoothing();

    glEnableClientState(GL_VERTEX_ARRAY);
    glVertexPointer(3, GL_FLOAT, sizeof(ofVec3f), &circlePoints[0].x);
    glDrawArrays((bFilled == OF_FILLED) ? GL_TRIANGLE_FAN : GL_LINE_STRIP, 0, circlePoints.size());

    // use smoothness, if requested:
    if (bSmoothHinted && bFilled == OF_OUTLINE) endSmoothing();

}
```

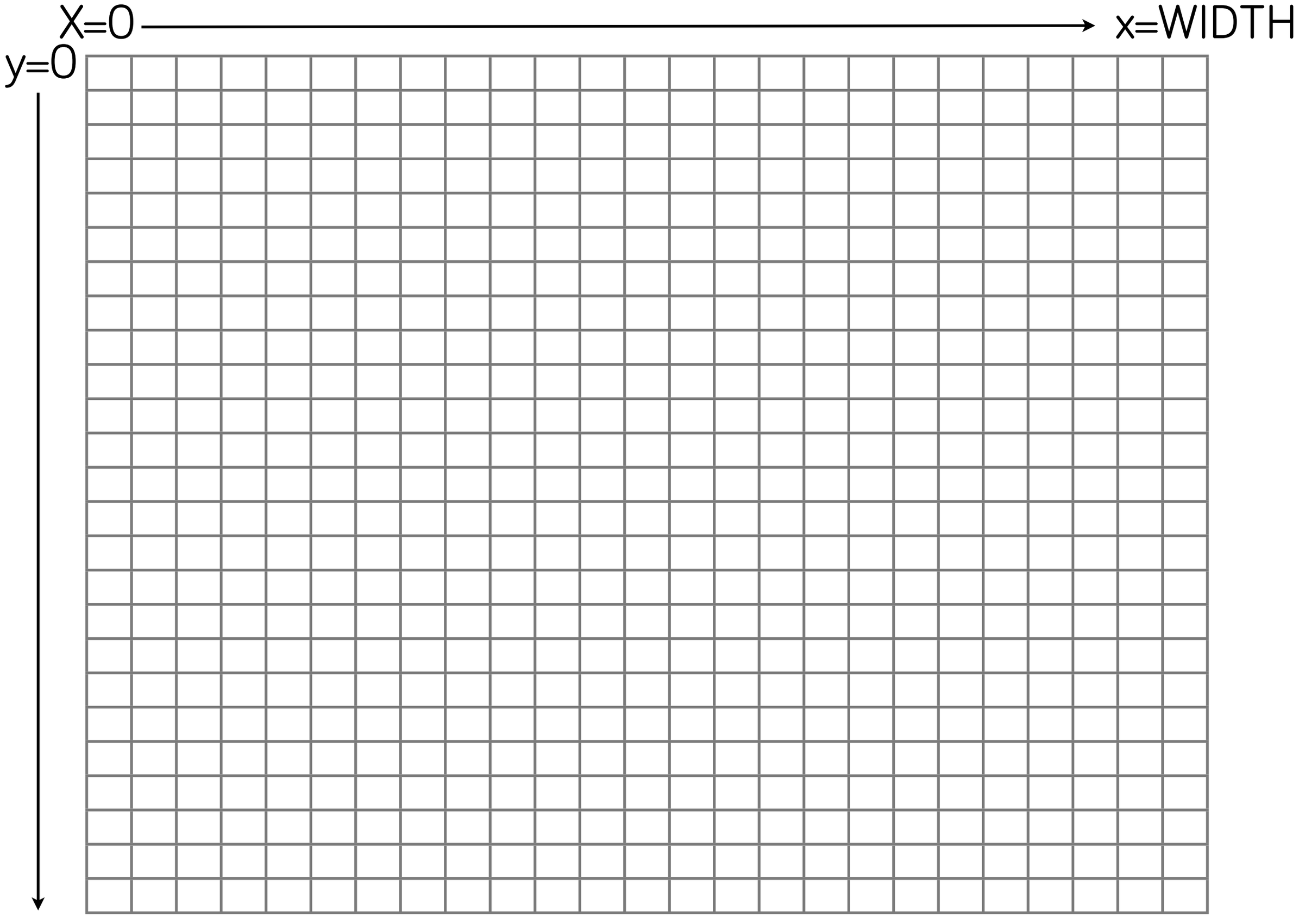
Coloring Shapes

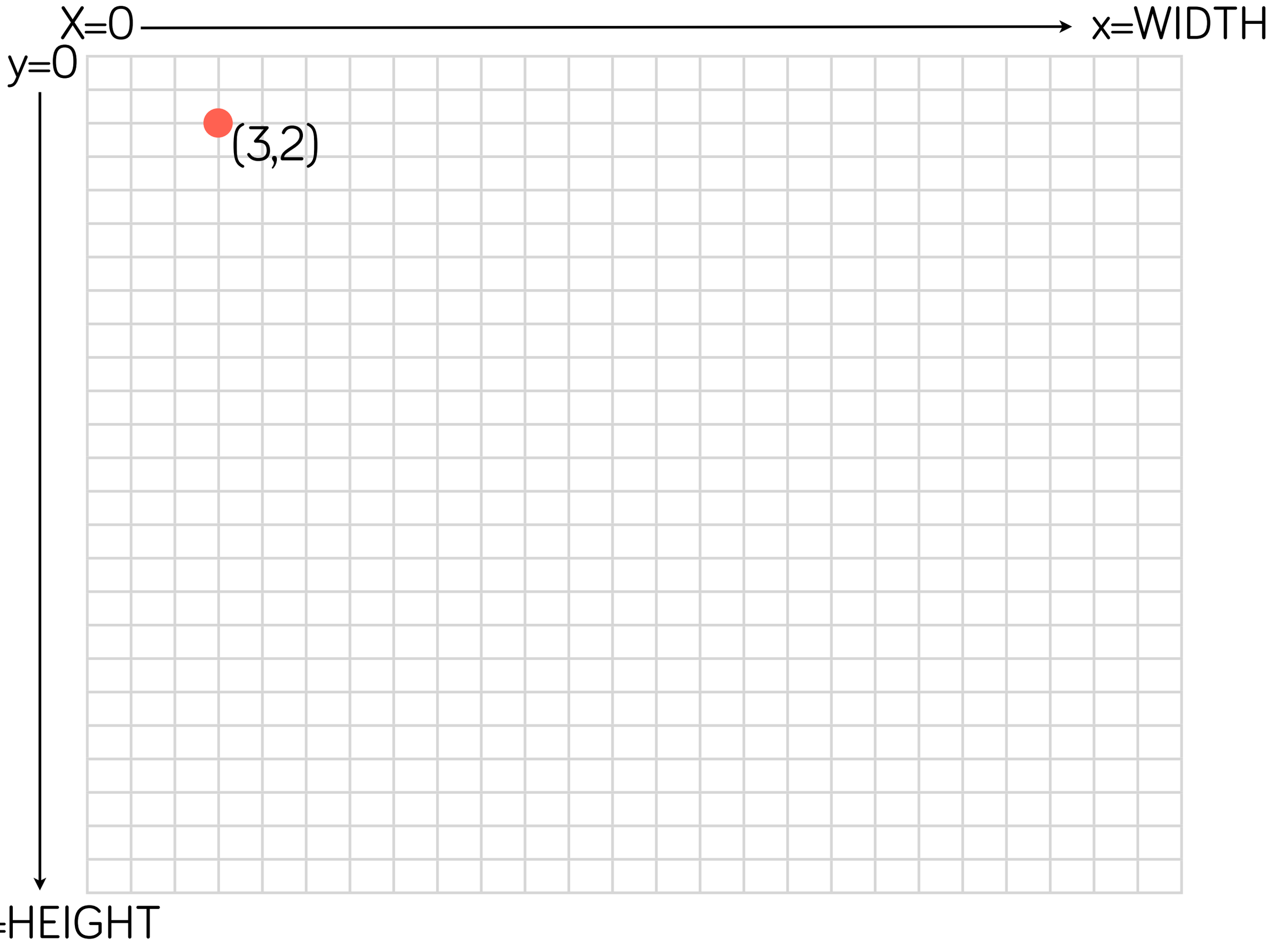
Function	Notes
ofFill();	Fill all following shapes with a color.
ofNoFill();	Don't fill the following shapes.
ofSetColor(r, g, b);	Sets the color to be used on all following shapes. Each value goes from 0-255.

Exercise:
Try drawing some stuff!

(Where would you put this code?)

How does positioning work?


X=0
x=WIDTH
y=0
y=HEIGHT


X=0
x=WIDTH
y=0
(3,2)
y=HEIGHT

Coordinate Plane

- X: horizontal axis, gets larger as you go right
- Y: vertical axis, gets larger as you go down

Question:
How would you draw a circle at x-position 9, y-position 15?

If you drew another circle at y-position 25, would it be higher or lower than the first circle?

Hint: if you don't know, try drawing it!

How can I incorporate interactivity?

Mouse Positions

- Variable: a symbol used to stand in for a value
- mouseX: returns the current x pixel position of the mouse
- mouseY: returns the current y pixel position of the mouse

Exercise:

Try drawing a circle at mouseX and mouseY!

Variables

- Variables are useful for storing data that may change throughout the course of your app (e.g. your player's health)
- To create a variable, you have to tell the computer:
  - What kind of data you're storing (a number? a word?)
  - The name you're going to refer to it by

Some Variable Types

- Float: a decimal number (“I’m 5.4 feet tall.”)
- Integer: a whole number (“I’m 25 years old.”)
- Boolean: a true/false condition (“I’m not from California.”)
- String: text (“My name is Jane.”)
- Char: a single letter (“You all get an A in programming!”)

```
float jane_height;
```

the type of data (datatype)

the variable's name

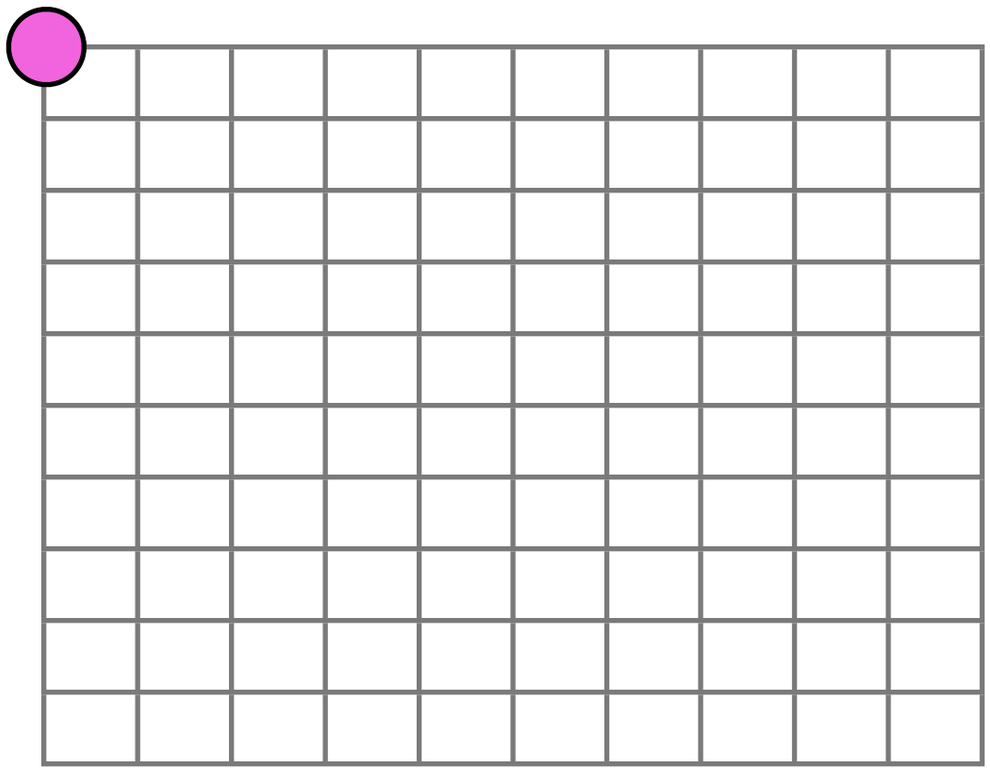
```
jane_height = 5.4;
```

the variable's name

the value of that variable

How can I make stuff move on its own?

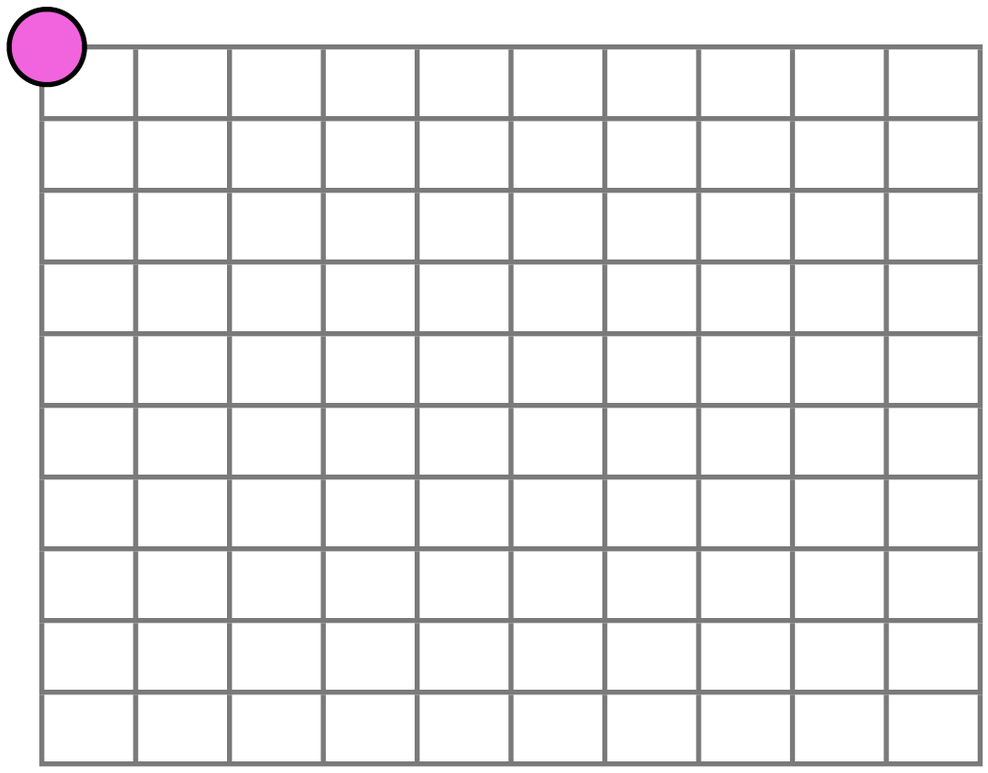
Movement



If our object starts at x=1, at time=0, and moves at a speed of 1 frame/sec, what will x equal at time=1?

How do you know?

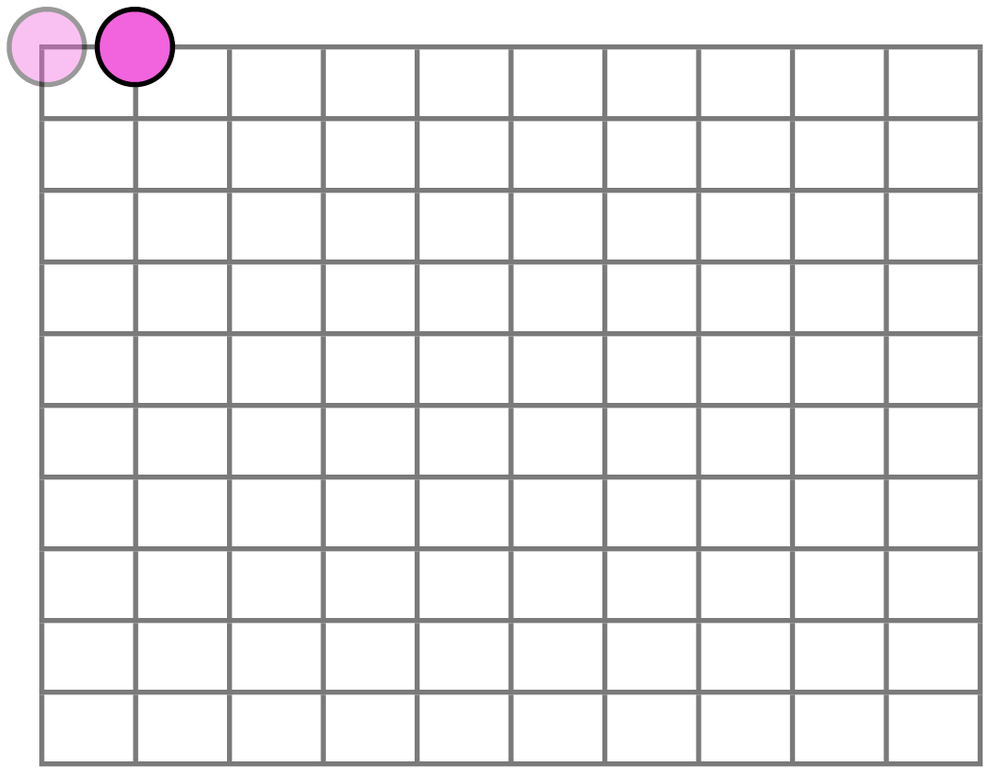
Movement



x will equal 1!

Its current position (0)
+ the total distance it will go over one frame (1)
= 1!

Movement



New position = old position + speed

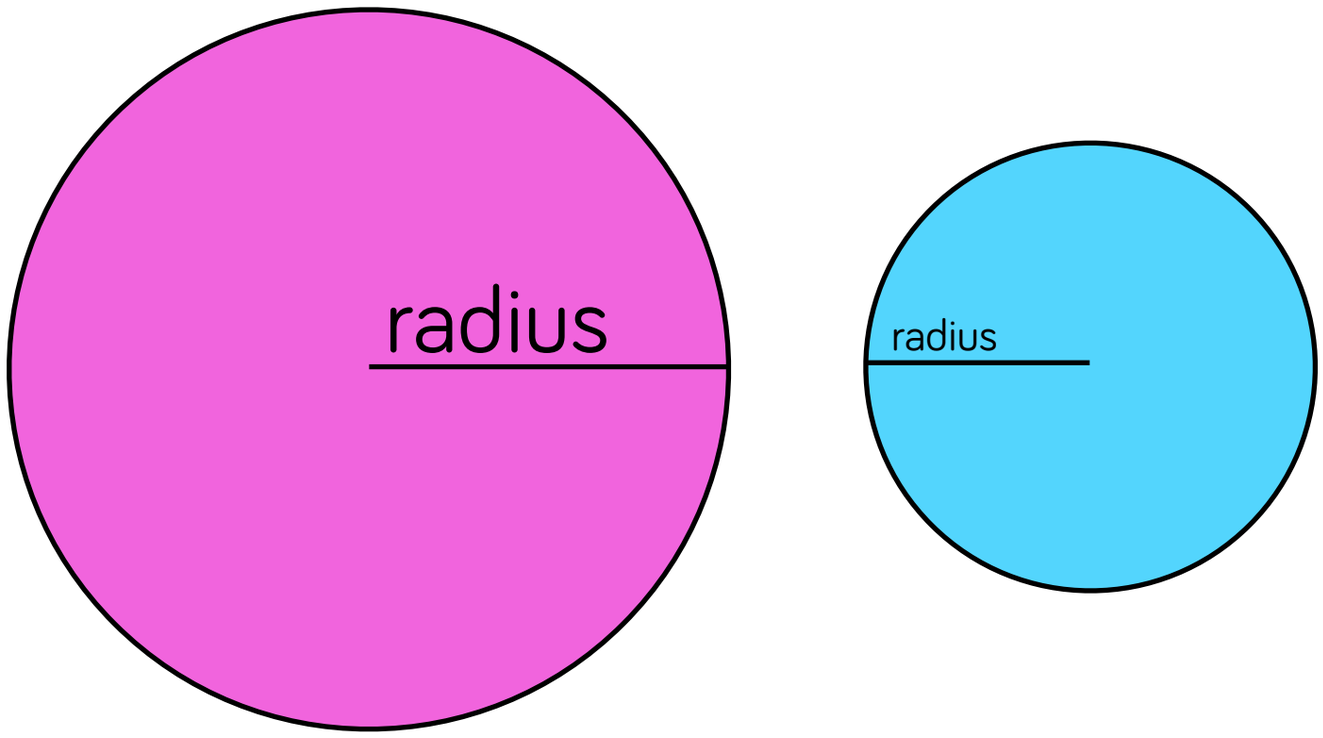
Exercise:
Try drawing a circle that moves vertically down the screen.

Hint: you'll want a variable to hold the circle's position. (Why?)

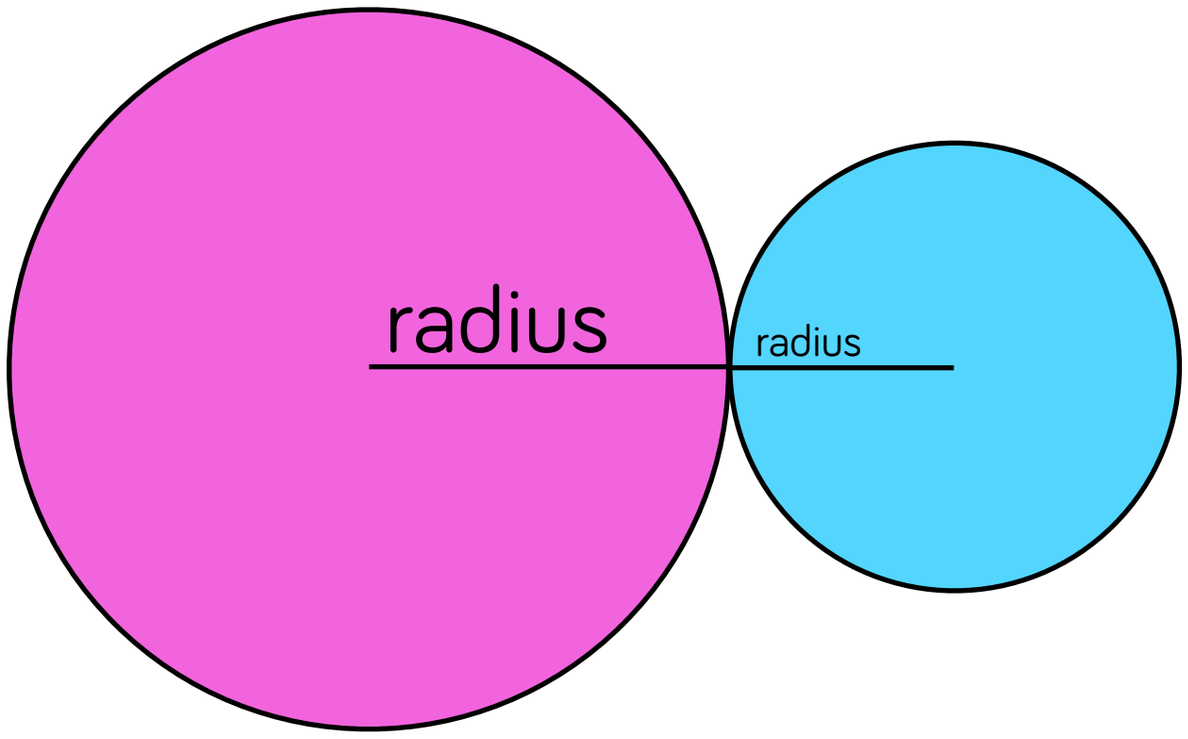
How can I test for collisions?

Collision: when one point is less than a certain distance from another point.

Have these circles collided yet?



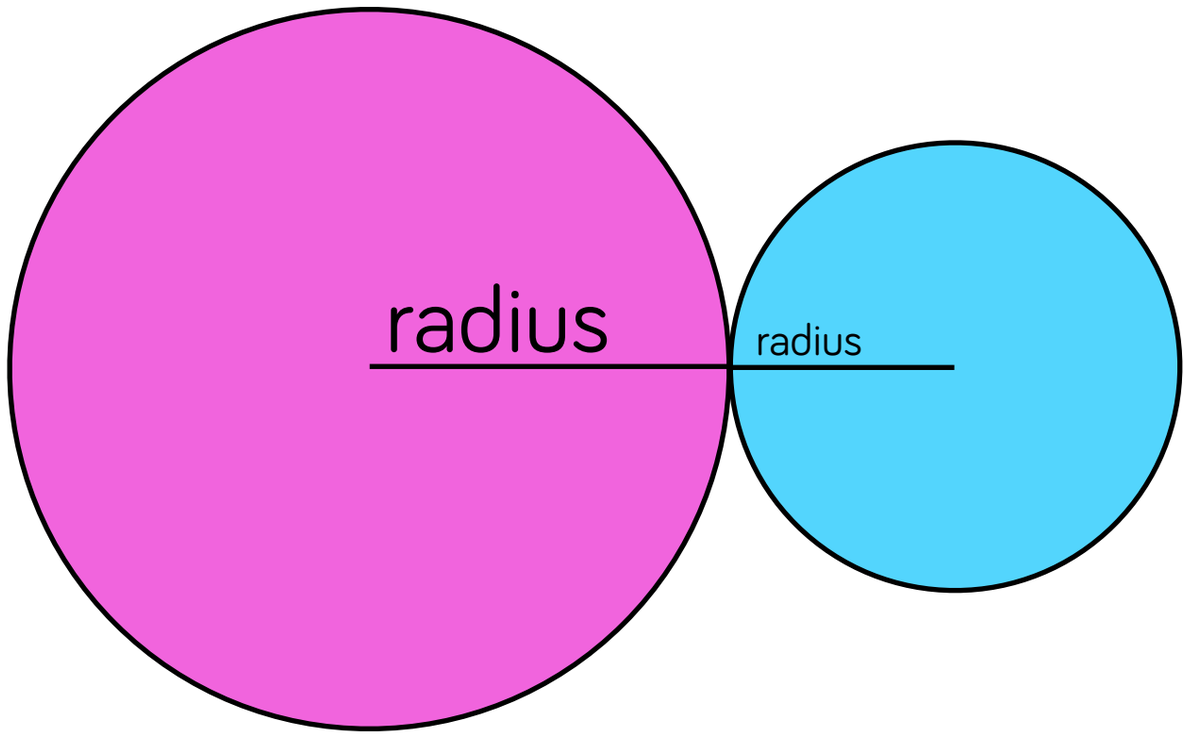
How about now?



Circle Collision

- If the distance between the center-points of the circles is less than or equal to the sum of their radii, they have collided!

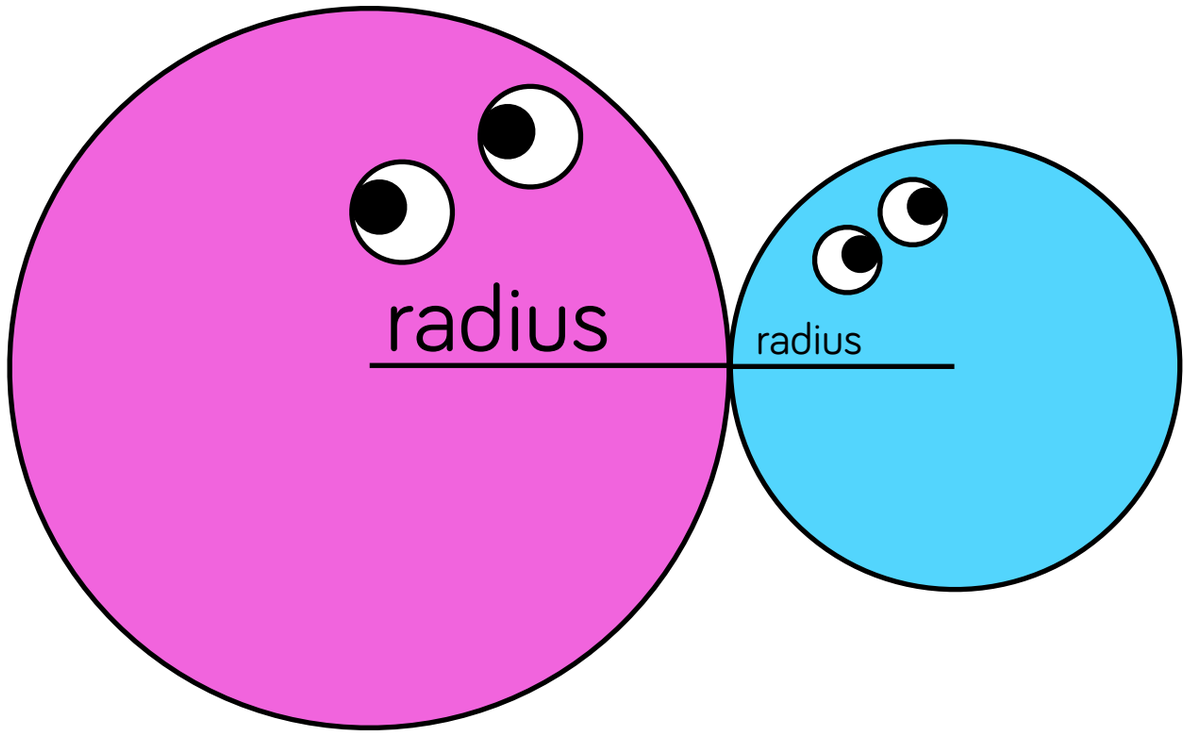
- You can calculate distance in openFrameworks with ofDist(x1, y1, x2, y2).



Circle Collision

- If the distance between the center-points of the circles is less than or equal to the sum of their radii, they have collided!

- You can calculate distance in openFrameworks with ofDist(x1, y1, x2, y2).



How can I check whether something is true?

If I’m hungry, then I’ll eat.

If-statements

- Consist of a condition and an action to take.
- Can have alternatives (if-else) and can put if-statements inside of if-statements, too!

General syntax:

```
if (condition) {
 action to take
}
```

If I'm hungry, then I'll eat.

```
if (hungry) {
 eat();
}
```

If-statements

- Consist of a condition and an action to take.
- Can have alternatives (if-else) and can put if-statements inside of if-statements, too!

If I'm hungry, then I'll eat.

```
if (hungry) {
    eat();
}
```

If I'm hungry, then I'll eat.
Otherwise, I'll dance!

```
if (hungry) {
    eat();
} else {
    dance();
}
```

If I'm hungry, then I'll eat.
If I'm hungry and in the mood for pizza, I'll get pizza.
Otherwise, I'll dance!

```
if (hungry) {
    if (want_pizza) {
        eat(pizza);
    } else {
        eat(something_else);
    }
} else {
    dance();
}
```

How can I incorporate images, sounds, fonts, etc.?

Images

- ofImage: a built-in object that handles the loading and drawing of images

- Must put the file inside “data” folder of project!

- Three steps:

  - Create your image variable: ofImage image;

  - Load your image file: image.loadImage(“image.png”);

  - Draw your image: image.draw(x, y);

Fonts

- ofTrueTypeFont: a built-in object that handles the loading and drawing of fonts
- Must put the file inside “data” folder of project!
- Three steps:
  - Create your font variable: ofTrueTypeFont font;
  - Load your image file: font.loadFont(“font.ttf”, size);
  - Draw your words: font.drawString(string, x, y);

Sounds

- ofSoundPlayer: a built-in object that handles the loading and playing of sounds
- Must put the file inside “data” folder of project!
- Three steps:
  - Create your font variable: ofSoundPlayer sound;
  - Load your image file: sound.loadSound(“sound.mp3”);
  - Play your sound: sound.play();