

# Openframeworks x iPad Game Design (2)

黃怡靜

Janet Huang

2012.05.01

# Topic

- review openframeworks basic architecture
- review basic programming language
- press buttons
- display current time
  - get current time
  - show text

# How to study?

- example code

graphicsExample, advancedGraphics, soundPlayerExample, ...etc

- online document

<http://www.openframeworks.cc/documentation/>

## core

### arduino

ofArduino

### 3d

ofNode

ofEasyCam

ofMesh

ofCamera

### math

ofVec2f

ofMatrix4x4

ofMath

ofQuaternion

ofMatrix3x3

ofVec4f

### graphics

ofPath

ofImage

ofGraphics

ofGetStyle()  
ofEndShape(...)  
ofGetViewportWidth()  
ofSetSphereResolution(...)  
ofSetupScreenPerspective(...)  
ofEnablePointSprites()  
ofGetFill()  
ofDisableAlphaBlending()  
ofDrawBitmapString(...)  
ofRotateX(...)  
ofPopMatrix()  
ofEnableSmoothing()  
ofSetCoordHandedness(...)  
ofSetupScreenOrtho(...)  
ofScale(...)  
ofNoFill()  
ofRotateY(...)  
ofVertex(...)  
ofSetLineWidth(...)  
ofEllipse(...)  
ofPopStyle()

### types

ofColor\_

ofStyle

ofRectangle

ofPoint

### events

ofKeyEventArgs

ofAudioEventArgs

ofCoreEvents

ofEventArgs

ofEvent

ofMouseEventArgs

ofResizeEventArgs

ofEvents

ofEventTitle

# Basic programming language

## data type

- int: 1, 4, 3 ...etc
- float: 0.24, 0.33, ...etc
- string: "hello world", "byebye!!" ...etc
- bool: true, false
- char: 'a', 'b', 'c', ...etc

## array

- int iarr[3]
- float farr[5]
- char carr[2]

## if-else

```
int a = 3;
if (a > 0){
    cout<< "YES! a=" << a << endl;
}else{
    cout << "NO! a=" << a << endl;
}
```

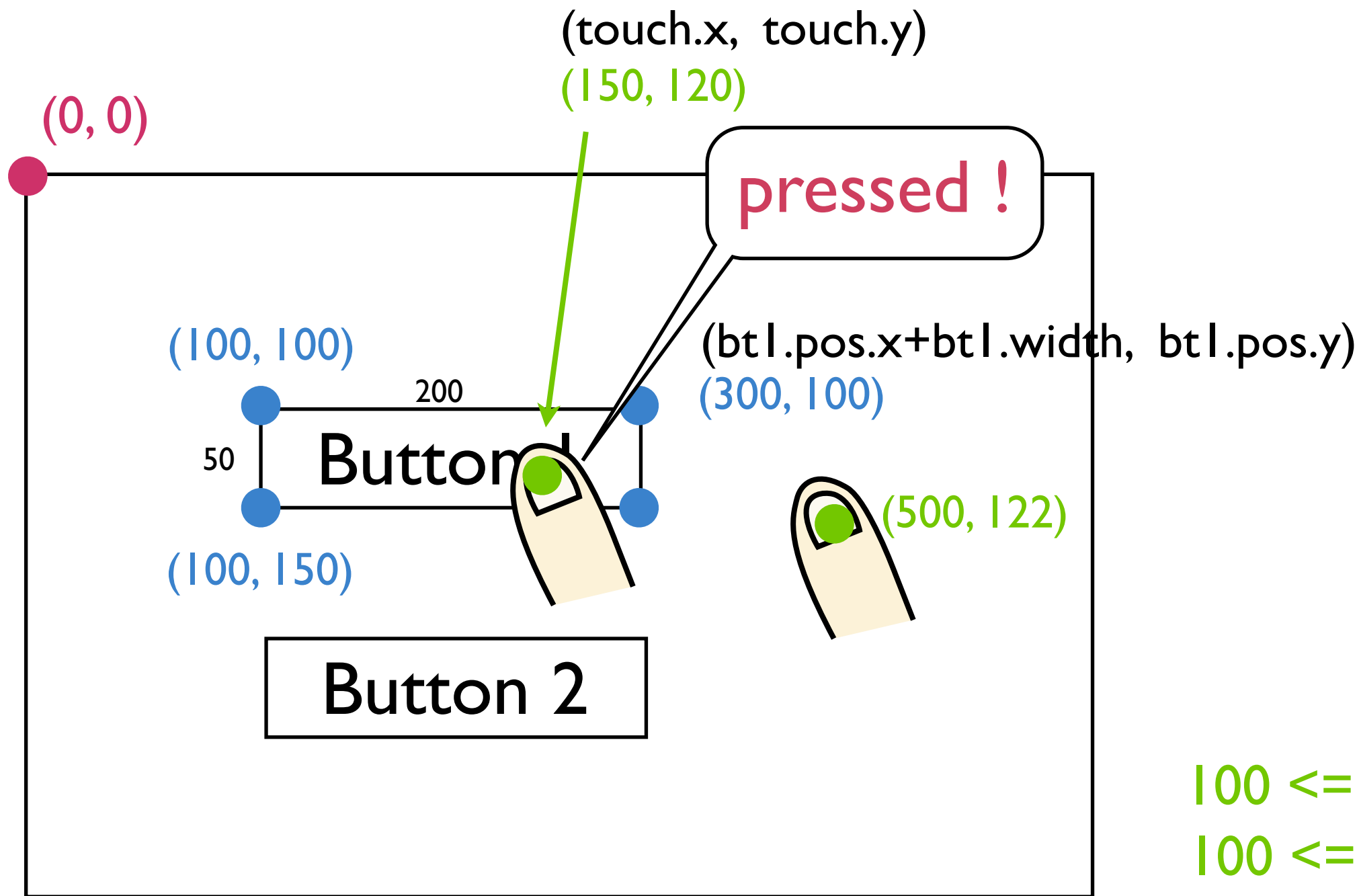
## for-loop

```
for(int i=0; i< 5< i++){
    printf("%d\n", i );
}
```

## function

```
void testFunction(){
    char c[32];
    sprintf(c, "%s, welcome here!", "Janet");
    printf("%s", c);
}
```

# press buttons

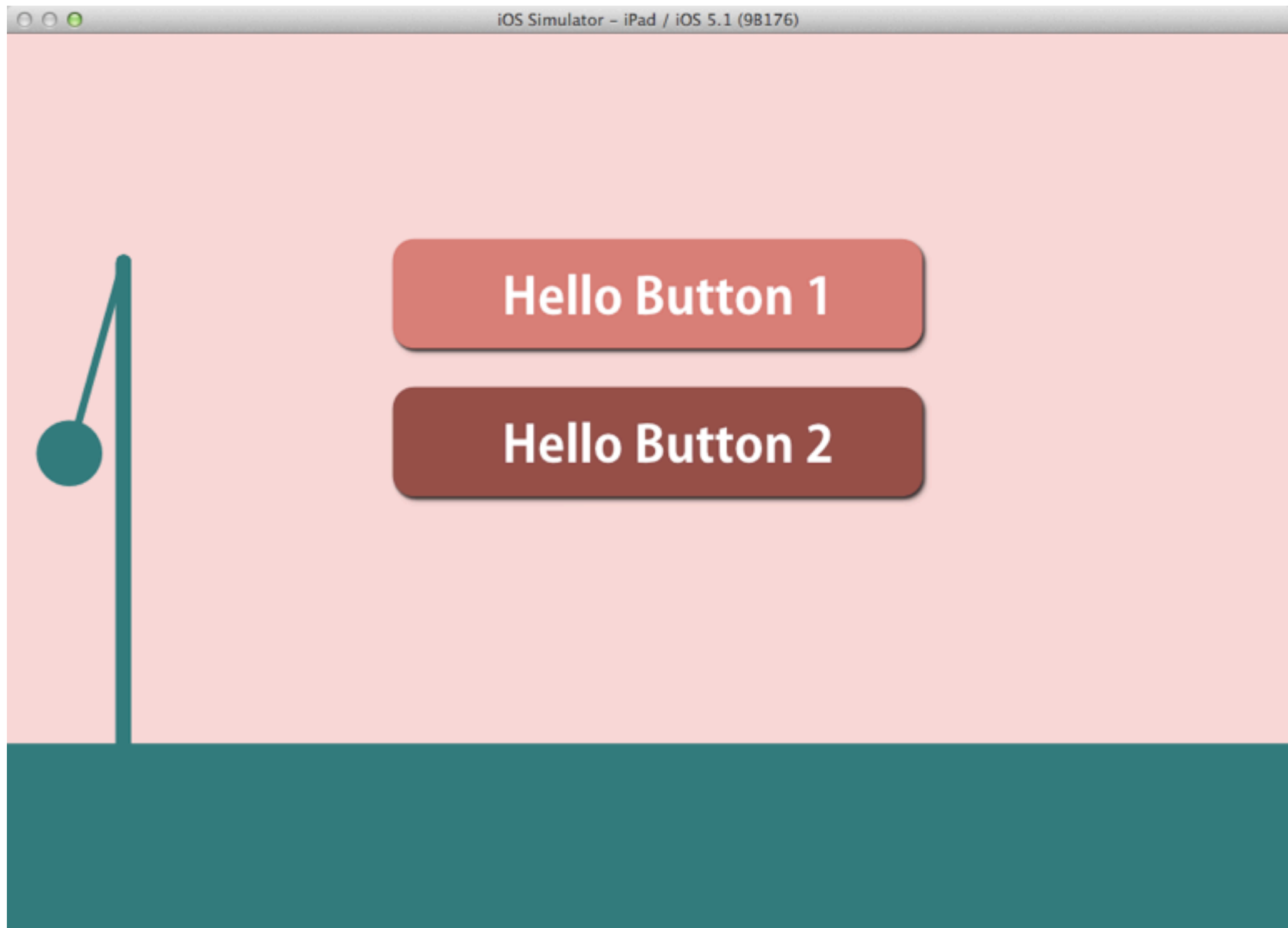


$$100 \leq \text{touch.x} \leq 300$$
$$100 \leq \text{touch.y} \leq 150$$

## pressed condition

$$\text{bt1.pos.x} \leq \text{touch.x} \leq \text{bt1.pos.x} + \text{bt1.width}$$
$$\text{bt1.pos.y} \leq \text{touch.y} \leq \text{bt1.pos.y} + \text{bt1.height}$$

# press buttons



```
#pragma once

#include "ofMain.h"
#include "ofxiPhone.h"
#include "ofxiPhoneExtras.h"

class testApp : public ofxiPhoneApp {

public:
    void setup();
    void update();
    void draw();
    void exit();

    void touchDown(ofTouchEventArgs &touch);
    void touchMoved(ofTouchEventArgs &touch);
    void touchUp(ofTouchEventArgs &touch);
    void touchDoubleTap(ofTouchEventArgs &touch);
    void touchCancelled(ofTouchEventArgs &touch);

    void lostFocus();
    void gotFocus();
    void gotMemoryWarning();
    void deviceOrientationChanged(int newOrientation);

    ofImage backgroundImage;
    ofImage btImg1[2];
    ofImage btImg2[2];

    bool pressed1;
    bool pressed2;
};
```

testApp.h

```
#include "testApp.h"

void testApp::setup(){
    // register touch events
    ofRegisterTouchEvents(this);

    // initialize the accelerometer
    ofxAccelerometer.setup();

    //iPhoneAlerts will be sent to this.
    ofxiPhoneAlerts.addListener(this);

    //If you want a landscape oreintation
    iPhoneSetOrientation(OFXIPHONE_ORIENTATION_LANDSCAPE_RIGHT);
    ofBackground(0,0,0);

    //load background image
    backgroundImage.loadImage("images/background.png");

    //load button image
    btImg1[0].loadImage("images/bt1.png");
    btImg1[1].loadImage("images/bt1_pressed2.png");
    btImg2[0].loadImage("images/bt2.png");
    btImg2[1].loadImage("images/bt2_pressed2.png");

    pressed1 = false;
    pressed2 = false;
}
```



```
void testApp::draw(){
    ofEnableAlphaBlending();
    backgroundImg.draw(0,0);

    //draw button 1
    if (pressed1) {
        btImg1[1].draw(300,200);
    }else {
        btImg1[0].draw(300,200);
    }

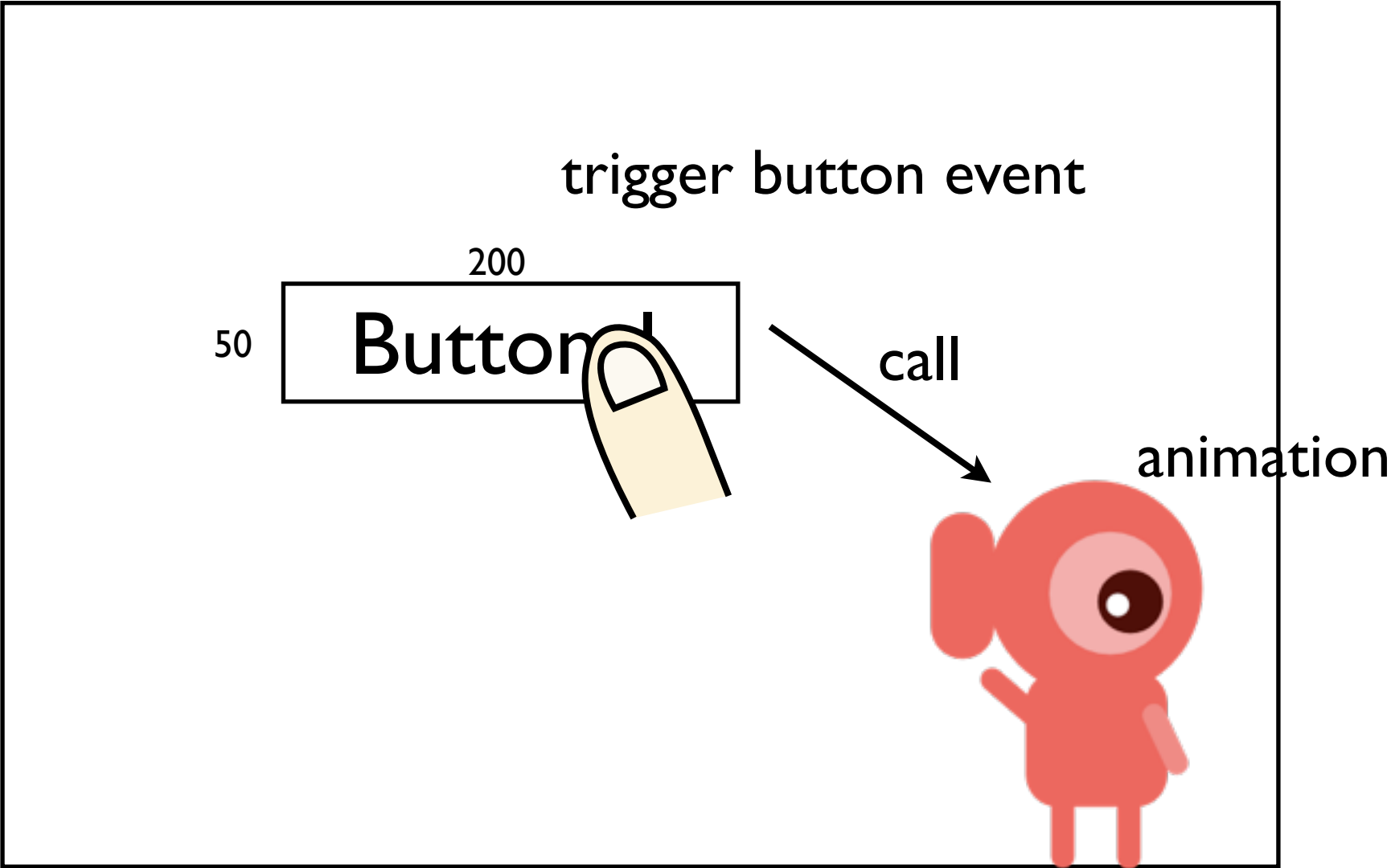
    //draw button 2
    if (pressed2) {
        btImg2[1].draw(300,320);
    }else {
        btImg2[0].draw(300,320);
    }
}
```

```
void testApp::touchDown(ofTouchEventArgs &touch){
    //check if button 1 is pressed or not
    if ((300 <= int(touch.x)) && (int(touch.x) <= (300+btImg1[0].width))) {
        if ((200 <= int(touch.y)) && (int(touch.y) <= (200+btImg1[0].height))) {
            pressed1 = true;
        }
    }

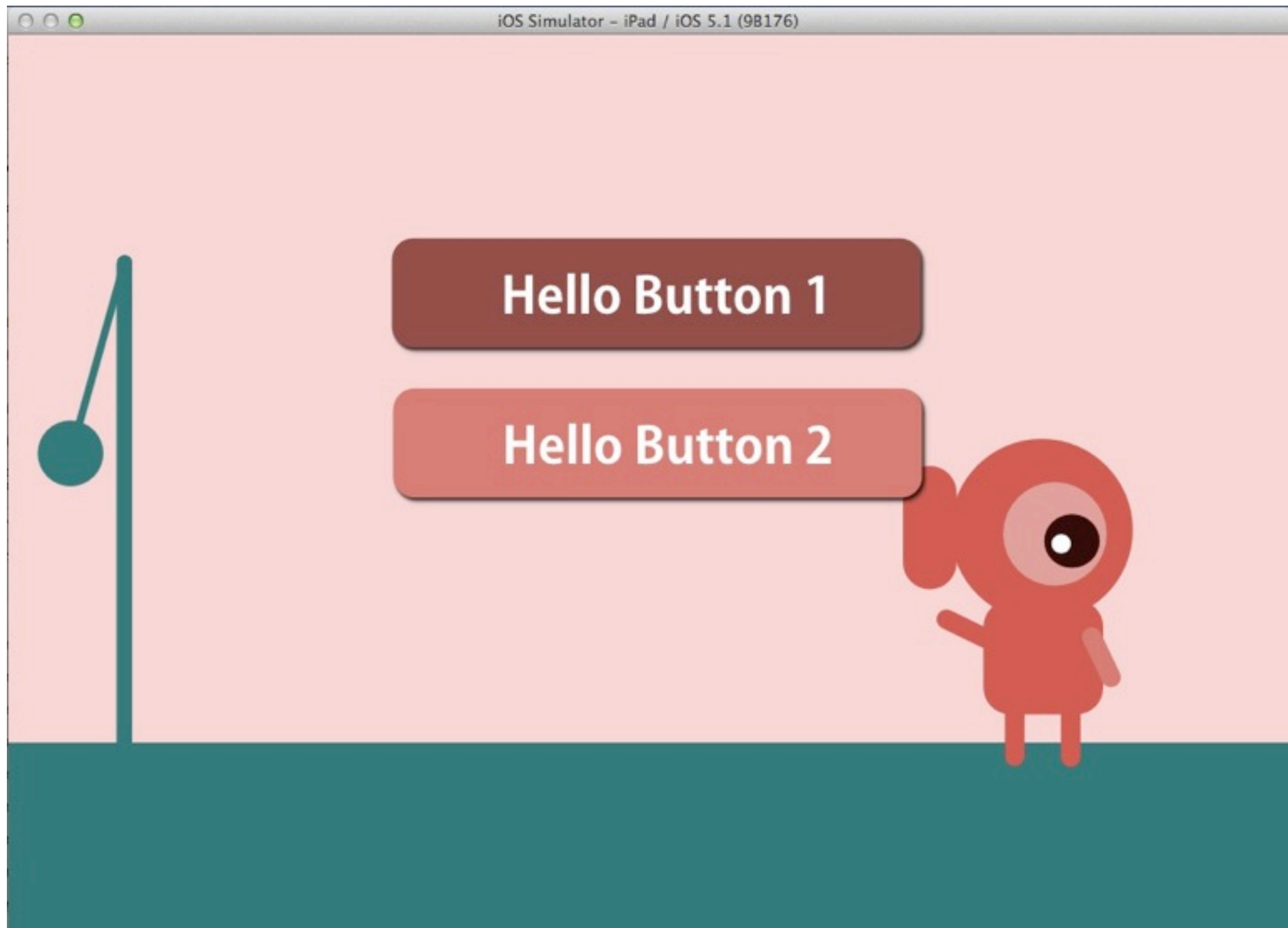
    //check if button 2 is pressed or not
    if ((300 <= int(touch.x)) && (int(touch.x) <= (300+btImg1[0].width))) {
        if ((320 <= int(touch.y)) && (int(touch.y) <= (320+btImg1[0].height))) {
            pressed2 = true;
        }
    }
}

void testApp::touchUp(ofTouchEventArgs &touch){
    pressed1 = false;
    pressed2 = false;
}
```

# press button + animation



# press button + animation



```
#pragma once
```

```
#include "ofMain.h"  
#include "ofxiPhone.h"  
#include "ofxiPhoneExtras.h"
```

```
#define AINMATIONFRAMENUM 13
```

```
class testApp : public ofxiPhoneApp {  
  
public:  
    void setup();  
    void update();  
    void draw();  
    void exit();  
  
    void touchDown(ofTouchEventArgs &touch);  
    void touchMoved(ofTouchEventArgs &touch);  
    void touchUp(ofTouchEventArgs &touch);  
    void touchDoubleTap(ofTouchEventArgs &touch);  
    void touchCancelled(ofTouchEventArgs &touch);  
  
    void lostFocus();  
    void gotFocus();  
    void gotMemoryWarning();  
    void deviceOrientationChanged(int newOrientation);  
  
    void playAnimation();  
  
    ofImage backgroundImage;  
    ofImage animationImg[AINMATIONFRAMENUM];  
    ofImage btImg1[2];  
    ofImage btImg2[2];  
  
    bool pressed1;  
    bool pressed2;  
  
    bool animationPlay;  
    int currFrame;  
};
```

```

void ofApp::setup(){

    ofRegisterTouchEvents(this);
    ofxAccelerometer.setup();
    ofxiPhoneAlerts.addListener(this);
    iPhoneSetOrientation(OFXIPHONE_ORIENTATION_LANDSCAPE_RIGHT);

    ofBackground(0,0,0);

    //load background image
    backgroundImg.loadImage("images/background.png");

    //load button image
    //button 1
    btImg1[0].loadImage("images/bt1.png");
    btImg1[1].loadImage("images/bt1_pressed2.png");

    //button 2
    btImg2[0].loadImage("images/bt2.png");
    btImg2[1].loadImage("images/bt2_pressed2.png");

    ofSetFrameRate(24); // 24fps
    //load animation image
    for (int i = 0; i < AINMATIONFRAMENUM; i++) {
        char char1[32];
        sprintf(char1, "images/creature%d.png", i+1);
        animationImg[i].loadImage(char1);
    }

    pressed1 = false;
    pressed2 = false;

    animationPlay = false;
    currFrame = 0;
}

```

```

void testApp::draw(){
    ofEnableAlphaBlending();
    backgroundImage.draw(0,0);

    //play animation
    playAnimation();

    //draw button 1
    if (pressed1) {
        btImg1[1].draw(300,200);
        animationPlay = true;
    }else {
        btImg1[0].draw(300,200);
    }

    //draw button 2
    if (pressed2) {
        btImg2[1].draw(300,320);
    }else {
        btImg2[0].draw(300,320);
    }
}

```

function playAnimation

```

void testApp::playAnimation(){
    if (animationPlay) {

        if (currFrame < AINMATIONFRAMENUM) {
            animationImg[currFrame].draw(700,350);
        }else {
            animationPlay = false;
            currFrame = 0;
        }
        currFrame++;
    }
}

```

# display current time

## get current time

It's very easy to get current time in openframeworks

- Hours: ofGetHours();
- Minutes: ofGetMinutes();
- Seconds: ofGetSeconds();

## show texts in the screen

ofDrawBitmapString("Hello", 100, 100);



The diagram consists of two pink labels, 'string' and 'float', positioned above the function call. Three pink arrows point from these labels to the arguments of the function: one arrow points from 'string' to the string literal 'Hello', and two arrows point from 'float' to the numerical values 100 and 100.



```
#include "testApp.h"

int h, m, s; //時、分、秒

void testApp::setup(){
    // register touch events
    ofRegisterTouchEvents(this);

    // initialize the accelerometer
    ofxAccelerometer.setup();

    //iPhoneAlerts will be sent to this.
    ofxiPhoneAlerts.addListener(this);

    //If you want a landscape oreintation
    iPhoneSetOrientation(OFXIPHONE_ORIENTATION_LANDSCAPE_RIGHT);
    ofBackground(0,0,0);
}

void testApp::update(){
    s = ofGetSeconds(); //秒
    m = ofGetMinutes(); //分
    h = ofGetHours(); //時
}

void testApp::draw(){
    char time[32];
    sprintf(time, "%02d : %02d : %02d", h, m, s);

    ofSetColor(255, 255, 255);
    ofDrawBitmapString(time, 20, ofGetHeight()/2);
}
```

# Homework

- finish your mockups
- finish buttons and animation in your app
- show your tangible object design