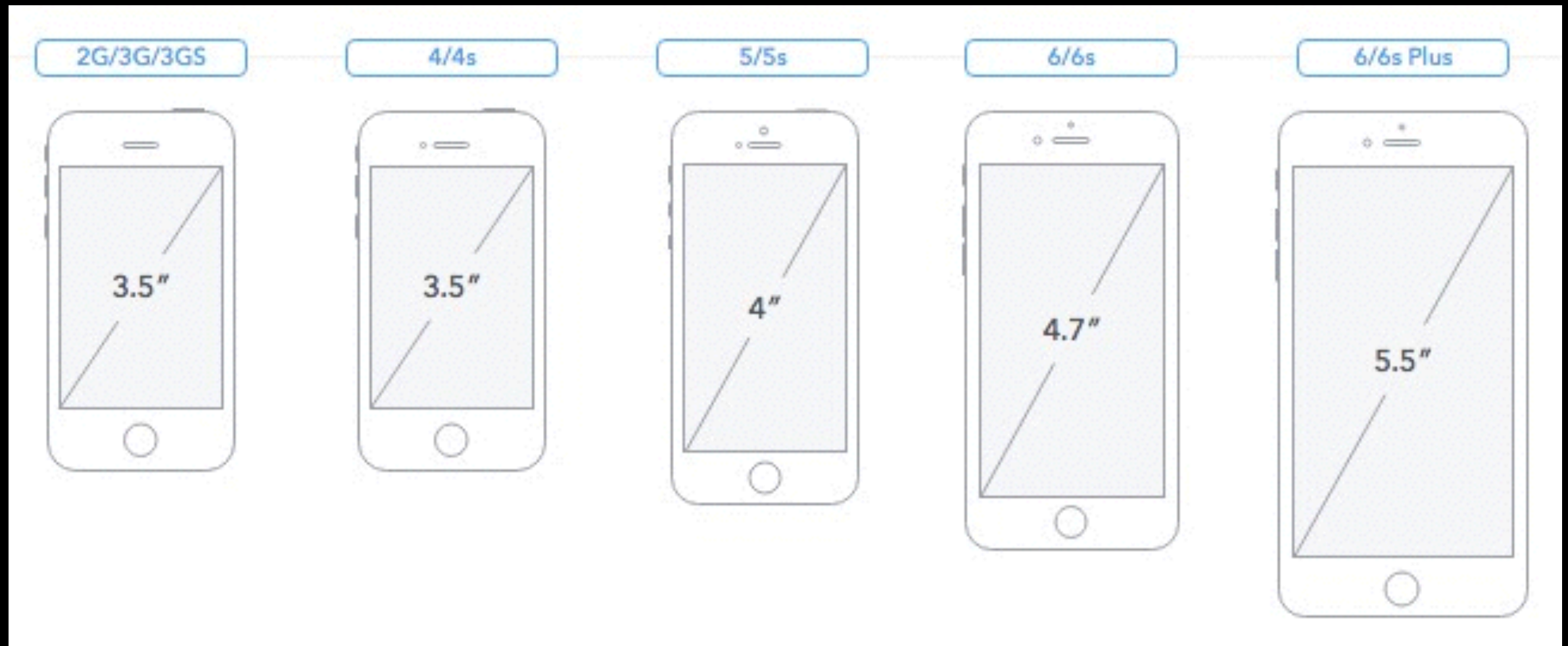


# iPhone Screen & AutoLayout

2016-08 明林清

- iPhone Screen
- UI 适配原则
- AutoLayout
- Masonry
- More

**iPhone Screen**



# iPhone Screen

# Sizes

***2G/3G/3GS***

***4/4s***

***5/5s***

***6/6s***

***6/6s Plus***

**320\*480**

**320\*480**

**320\*568**

**375\*667**

**414\*736**

**320\*480**

**640\*960**

**640\*1136**

**750\*1334**

**1242\*2208**

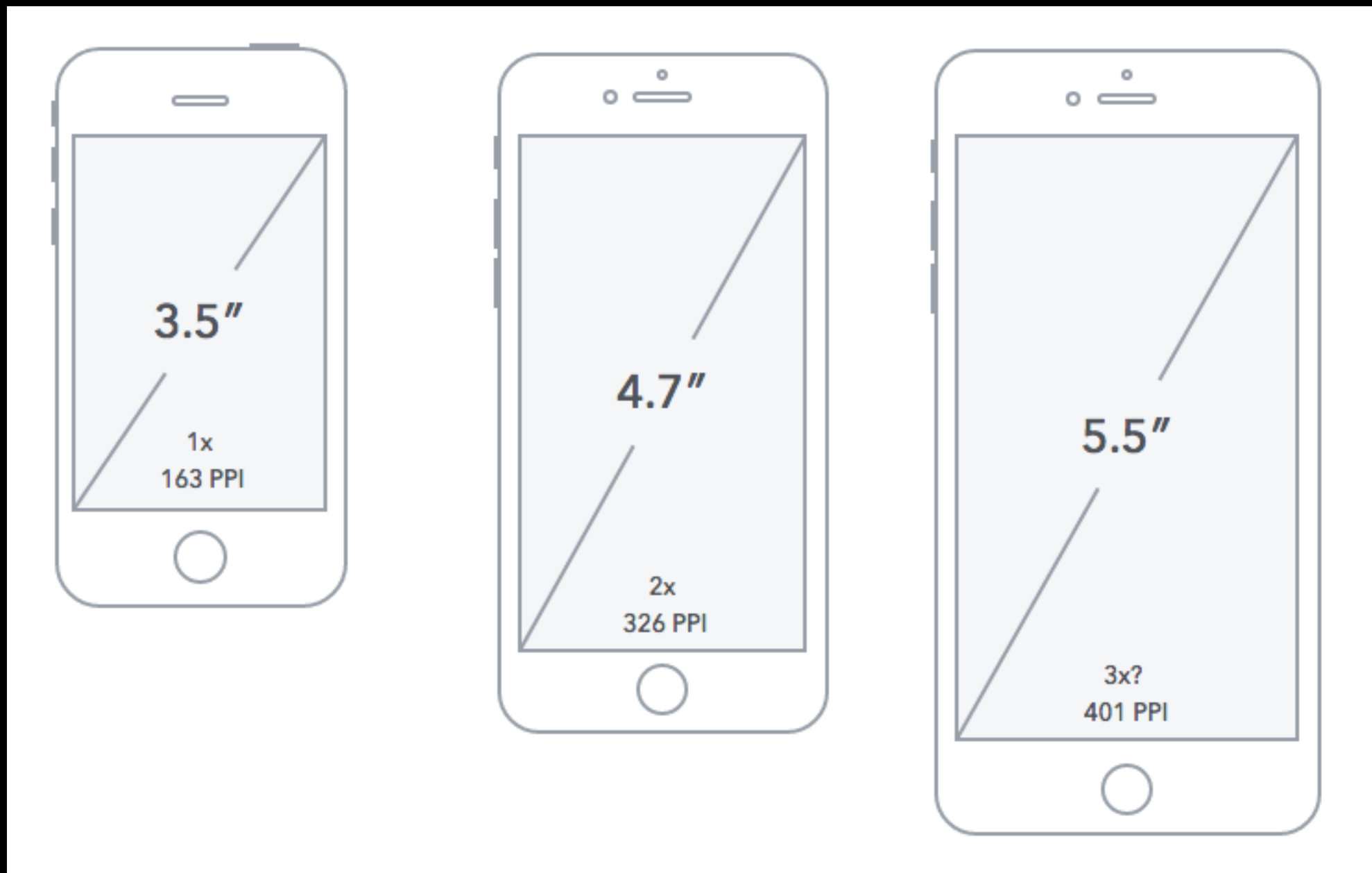
**320\*480**

**640\*960**

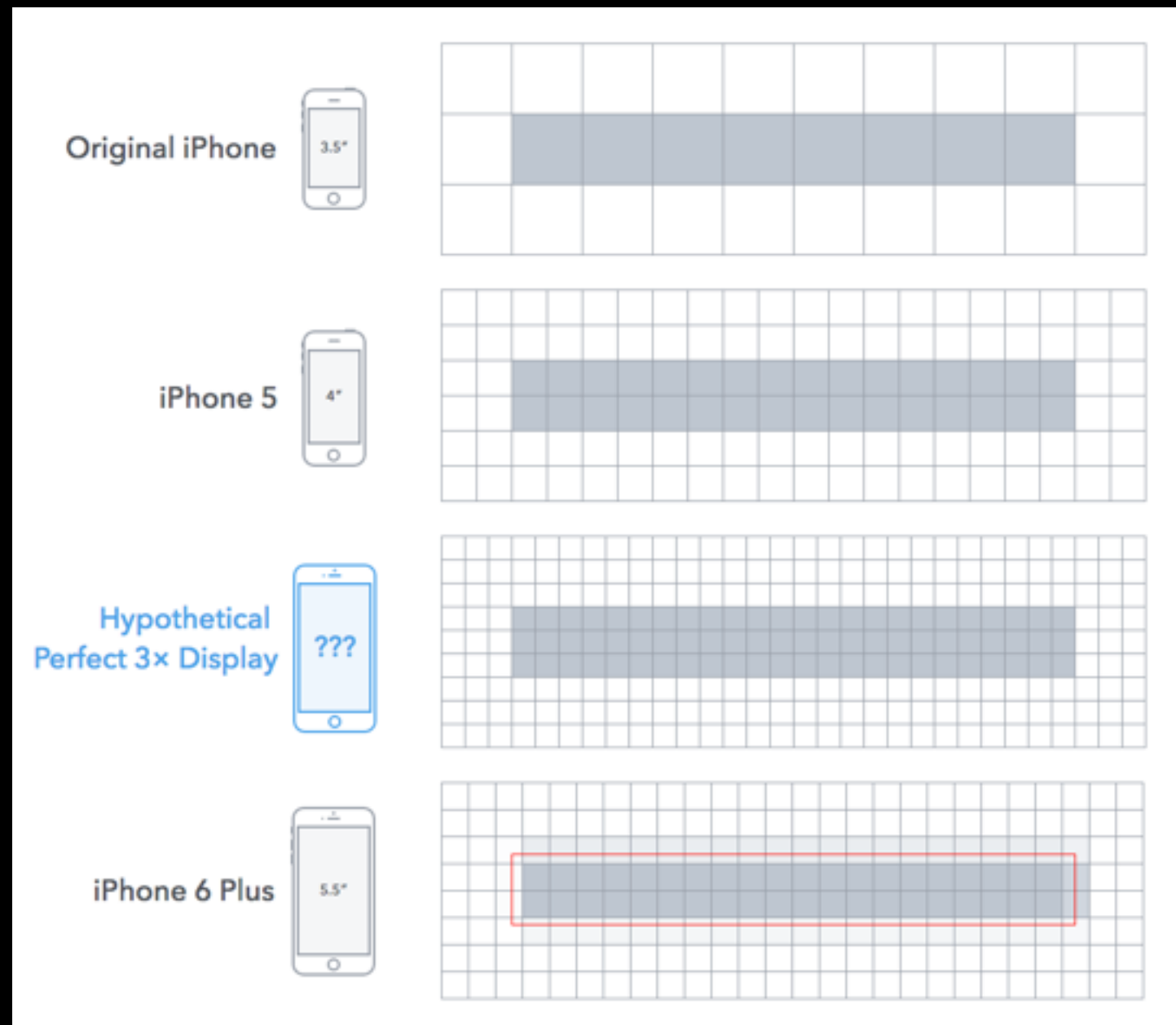
**640\*1136**

**750\*1334**

**1080\*1920**



# Scale & PPI



# Scale & PPI

# PPI

*screen*

*1x*

*2x*

*3x*

*!3x*

**PixelsPerInch**

**163**

**326**

**489**

**401**



# downsampling

<i>screen</i>	<i>1x</i>	<i>2x</i>	<i>3x</i>	<i>!3x</i>
scale	1.0	2.0	3.0	3.0
downsampling				/ 1.15
nativeScale	1.0	2.0	3.0	2.608696

# formula

**scale** = pixels / points

**nativeScale** = scale / downsampling

**renderedPixels** = points \* scale

**physicalPixels** = points \* scale / downsampling

= points \* nativeScale

**physicalSize** = physicalPixels / PPI

**screenSize** = (screenWidth ^ 2 + screenHeight ^ 2) ^ 0.5

# result

<i>scale/PPI</i>	<i>1x/163</i>	<i>2x/326</i>	<i>3x/401</i>
points	320*480	375*667	414*736
renderedPixels	320*480	750*1334	1242*2208
physicalPixels	320*480	750*1334	1080*1920

# Important to 🦁

攻城獅

射擊獅

4/4s

320\*480

649\*960

5/5s

320\*568

649\*1136

6/6s

375\*667

750\*1334

6/6s *Plus*

414\*736

1242\*2208

# Physical Size

<i>scale/PPI</i>	<i>1x/163</i>	<i>2x/326</i>	<i>3x/401</i>
points	44	44	44
renderedPixels	44	88	132
physicalPixels	44	88	114.782609
physicalSize	0.269939	0.269939	0.286241
	1 :	1 :	1.06039142

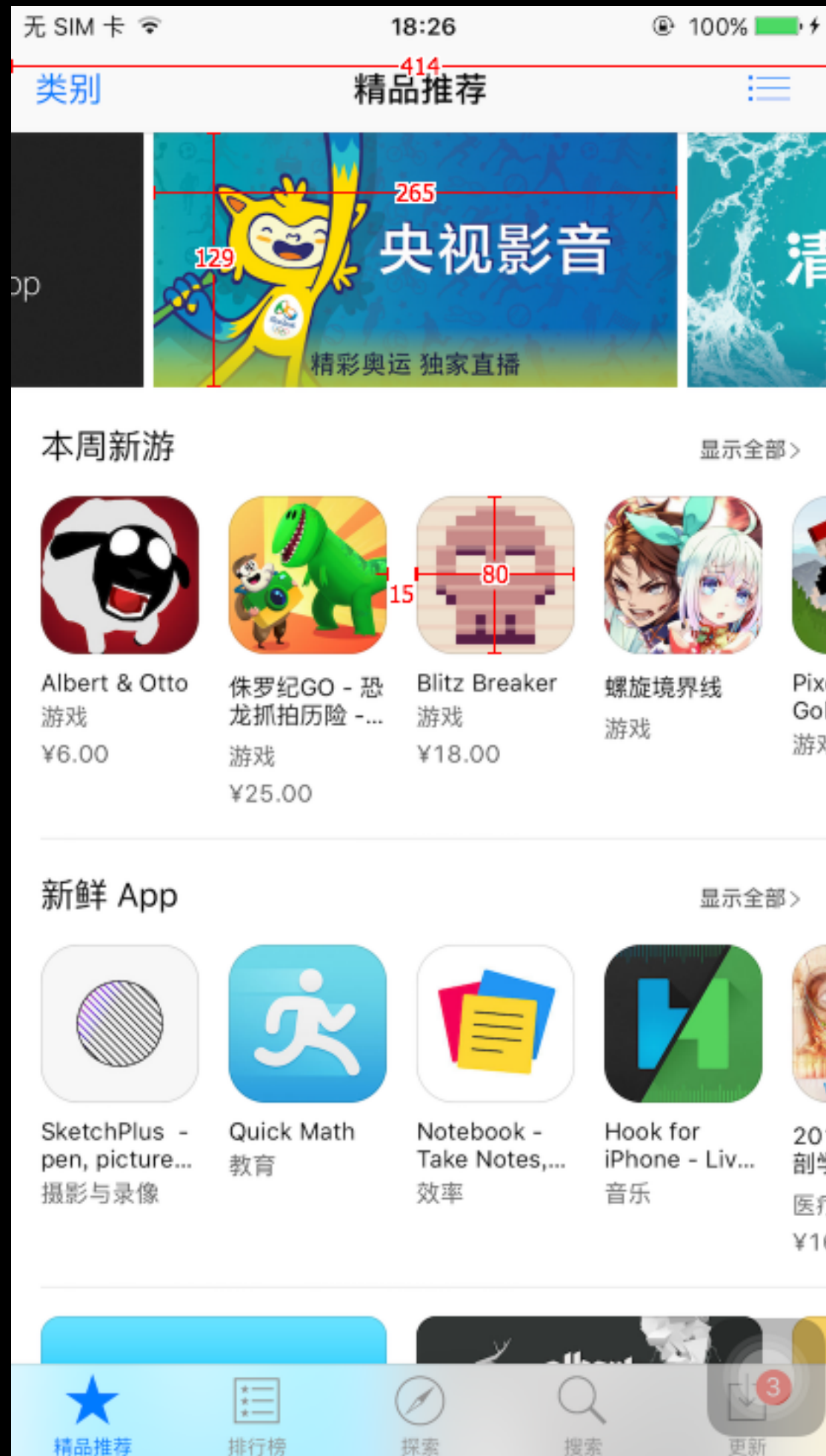
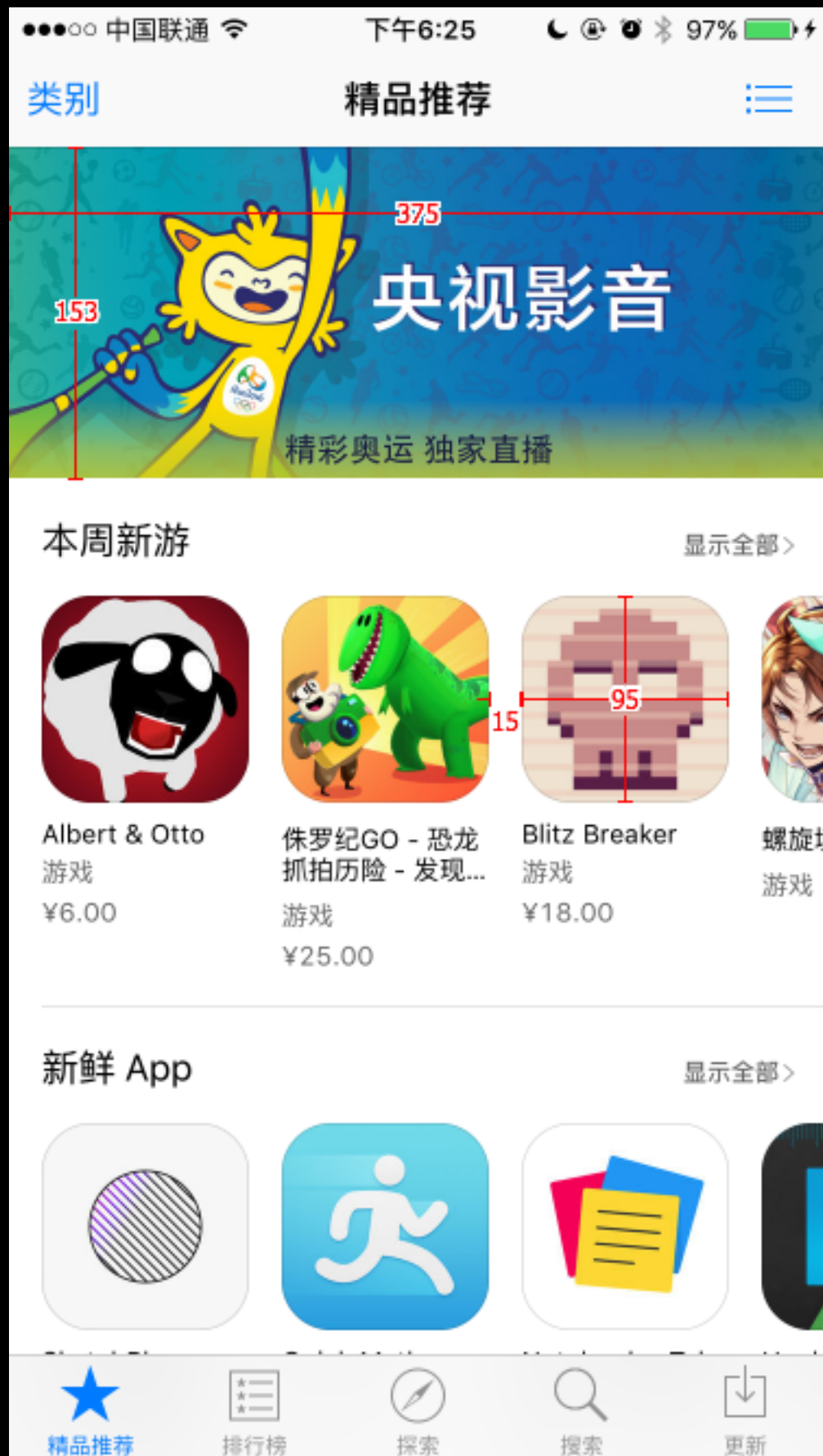
# UI 适配原则

**Bigger than Bigger**

**More than More**







**Sometimes Bigger, Sometimes More**

# Content

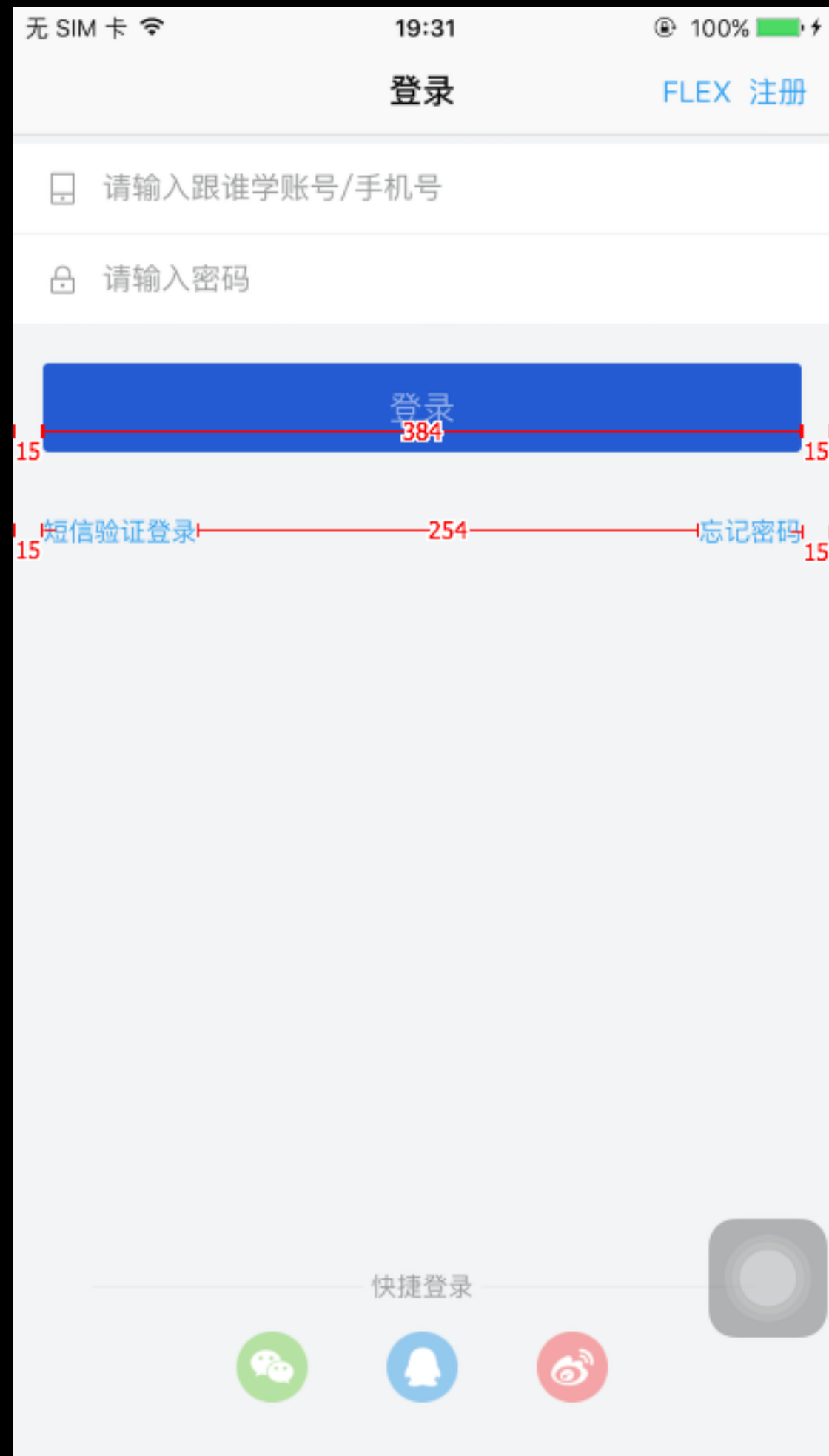
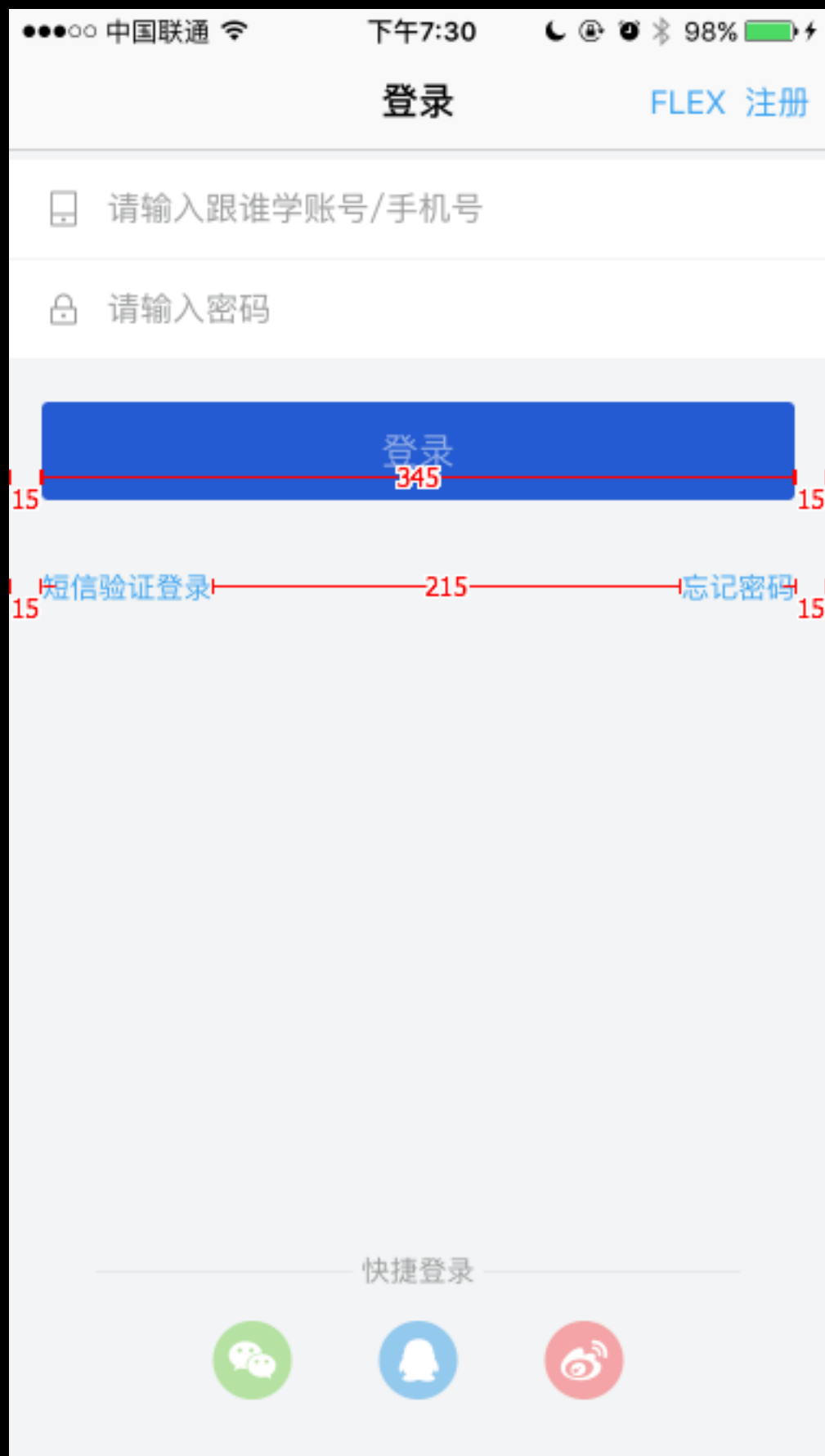
1. 与屏幕宽度有关的 view 宽变 高不变
2. 与屏幕宽度有关、并且自身宽高比固定的 view  
宽高一起变
3. 其它 不变

# Space

1. 固定间距 不变

2. 可变间距 变（与对齐方式有关）





# AutoLayout



# Why AutoLayout

- 写：不需要复杂计算
- 读：容易理解
- 运行：自动适应屏幕变化
- 功能：强大到可以搞定一切

# Constraint

`item1.attr1 == item2.attr2 × multiplier + constant`

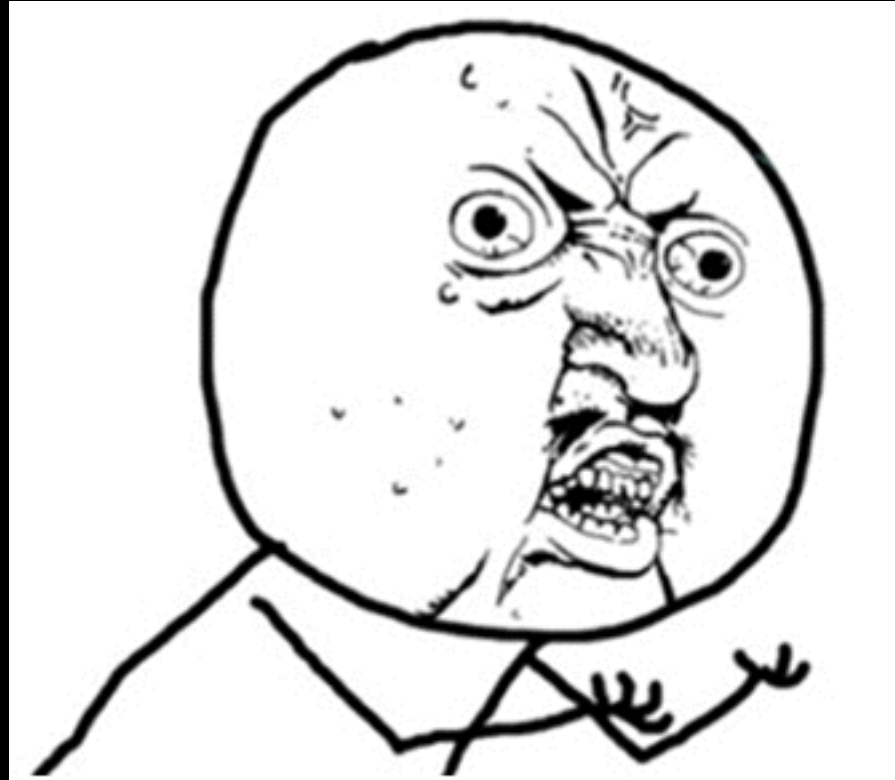
# Constraint

`item1.attr1 == [item2.attr2] × multiplier + constant`  
                  `nil`    `N/A`        `1.0`            `0.0`

`view1 left == view2 ...`  
      `right <=`  
      `top >=`  
      `bottom`  
      `width`  
      `height`  
      `centerX`  
      `centerY`

# Constraint?

```
button.left    == 20;  
button.top     == 30 + 44 + 20;  
button.width   == CGRectGetWidth(screen.bounds) - 20 * 2;  
button.height  == 44;
```



are you kidding me?!

# Constraint!

```
static CGFloat const margin = 20.0;
static CGFloat const buttonHeight = 44.0;

button.left    == superview.left + margin;
button.right   == button.right - margin;
button.top     == lastButton.bottom + margin;
button.height  == buttonHeight;
```

# Relation

**HOW TO**



# HOW TO

- XIB/Storyboard
- NSLayoutConstraint
- VFL(Visual Format Language)
- PureLayout
- ...
- Masonry

# Why Masonry

- ~~XIB/Storyboard~~
- ~~NSLayoutConstraint~~
- ~~VFL(Visual Format Language)~~
- ~~PureLayout~~
- ~~┌─┐~~
- Masonry ✓

# XIB/Storyboard

NO

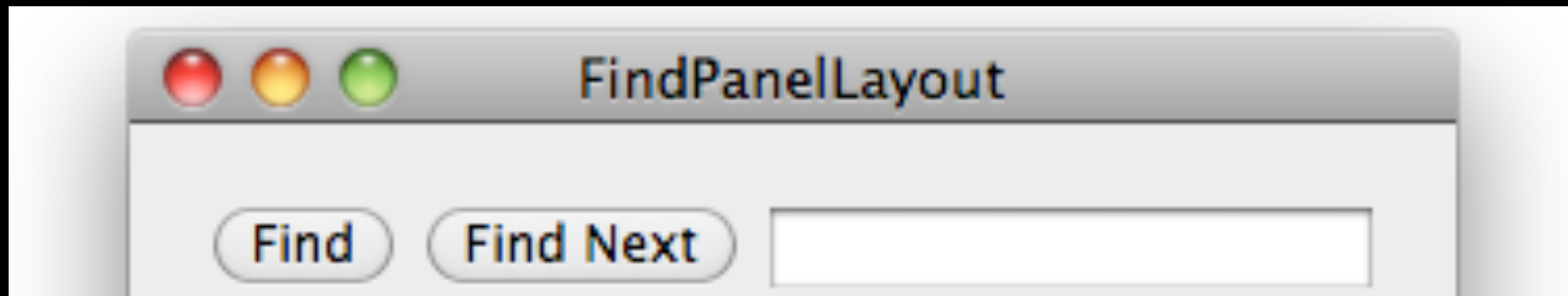
# NSLayoutConstraint

```
NSLayoutConstraint *topConstraint = [NSLayoutConstraint
                                     constraintWithItem:view1
                                     attribute:NSLayoutAttributeTop
                                     relatedBy:NSLayoutRelationEqual
                                     toItem:view2
                                     attribute:NSLayoutAttributeBottom
                                     multiplier:1.0
                                     constant:20];

[view1 addConstraint:topConstraint];
...
...
...
```

# VFL

```
|-[find]-[findNext]-[findField(>=20)]-|
```





# Masonry

```
make.top.equalTo(self.line2.mas_top).offset(margin);
```

```
item1.attr1 == item2.attr2 × multiplier + constant
```

# Masonry

```
make.left.right.and.top.equalTo(view2).with.insets(insets);
```



# Masonry

```
[self.upLinkTypeLabel mas_makeConstraints:^(MASConstraintMaker *make) {
    make.left.offset(normalMargin);
    make.top.equalTo(self.line2).with.offset(normalMargin);
}];

[self.upLinkTypeTCPRadioButton mas_makeConstraints:^(MASConstraintMaker *make) {
    make.left.offset(normalMargin);
    make.top.equalTo(self.upLinkTypeLabel.mas_bottom).with.offset(normalMargin);
    make.size.mas_equalTo(CGSizeMake(radiusButtonWidth, radiusButtonIconSize));
}];

[self.upLinkTypeUDPRadioButton mas_makeConstraints:^(MASConstraintMaker *make) {
    make.right.offset(- normalMargin);
    make.centerY.equalTo(self.upLinkTypeTCPRadioButton);
    make.size.mas_equalTo(CGSizeMake(radiusButtonWidth, radiusButtonIconSize));
}];

[self.downLinkTypeLabel mas_makeConstraints:^(MASConstraintMaker *make) {
    make.left.offset(normalMargin);
    make.top.equalTo(self.upLinkTypeTCPRadioButton.mas_bottom).with.offset(innerMargin);
}];

[self.downLinkTypeTCPRadioButton mas_makeConstraints:^(MASConstraintMaker *make) {
    make.left.offset(normalMargin);
    make.top.equalTo(self.downLinkTypeLabel.mas_bottom).with.offset(normalMargin);
    make.size.mas_equalTo(CGSizeMake(radiusButtonWidth, radiusButtonIconSize));
    make.bottom.offset(- normalMargin);
}];

[self.downLinkTypeUDPRadioButton mas_makeConstraints:^(MASConstraintMaker *make) {
    make.right.offset(- normalMargin);
    make.centerY.equalTo(self.downLinkTypeTCPRadioButton);
    make.size.mas_equalTo(CGSizeMake(radiusButtonWidth, radiusButtonIconSize));
}];
```

# PureLayout

```
[self.upLinkTypeLabel autoPinEdgeToSuperviewEdge:ALEdgeLeft withInset:normalMargin];
[self.upLinkTypeLabel autoPinEdge:ALEdgeTop toEdge:ALEdgeBottom ofView:self.line2
withOffset:normalMargin];
[self.upLinkTypeTCPRadioButton autoPinEdgeToSuperviewEdge:ALEdgeLeft withInset:normalMargin];
[self.upLinkTypeTCPRadioButton autoPinEdge:ALEdgeTop toEdge:ALEdgeBottom ofView:self.upLinkTypeLabel
withOffset:normalMargin];
[self.upLinkTypeTCPRadioButton autoSetDimension:ALDimensionWidth toSize:radioButtonWidth];
[self.upLinkTypeTCPRadioButton autoSetDimension:ALDimensionHeight toSize:radioButtonIconSize];
[self.upLinkTypeUDPRadioButton autoPinEdgeToSuperviewEdge:ALEdgeRight withInset:normalMargin];
[self.upLinkTypeUDPRadioButton autoAlignAxis:ALAxisCenterY
toSameAxisOfView:self.upLinkTypeTCPRadioButton];
[self.upLinkTypeUDPRadioButton autoSetDimension:ALDimensionWidth toSize:radioButtonWidth];
[self.upLinkTypeUDPRadioButton autoSetDimension:ALDimensionHeight toSize:radioButtonIconSize];

[self.downLinkTypeLabel autoPinEdgeToSuperviewEdge:ALEdgeLeft withInset:normalMargin];
[self.downLinkTypeLabel autoPinEdge:ALEdgeTop toEdge:ALEdgeBottom ofView:self.upLinkTypeTCPRadioButton
withOffset:innerMargin];
[self.downLinkTypeTCPRadioButton autoPinEdgeToSuperviewEdge:ALEdgeLeft withInset:normalMargin];
[self.downLinkTypeTCPRadioButton autoPinEdge:ALEdgeTop toEdge:ALEdgeBottom
ofView:self.downLinkTypeLabel withOffset:normalMargin];
[self.downLinkTypeTCPRadioButton autoSetDimension:ALDimensionWidth toSize:radioButtonWidth];
[self.downLinkTypeTCPRadioButton autoSetDimension:ALDimensionHeight toSize:radioButtonIconSize];
[self.downLinkTypeUDPRadioButton autoPinEdgeToSuperviewEdge:ALEdgeRight withInset:normalMargin];
[self.downLinkTypeUDPRadioButton autoAlignAxis:ALAxisCenterY
toSameAxisOfView:self.downLinkTypeTCPRadioButton];
[self.downLinkTypeUDPRadioButton autoSetDimension:ALDimensionWidth toSize:radioButtonWidth];
[self.downLinkTypeUDPRadioButton autoSetDimension:ALDimensionHeight toSize:radioButtonIconSize];
[self.downLinkTypeTCPRadioButton autoPinEdgeToSuperviewEdge:ALEdgeBottom withInset:normalMargin];
```

**Masonry**

# Masonry

<https://github.com/SnapKit/Masonry>

# Masonry

- Hello World
- 一组视图的横向居中对齐、等宽等间距对齐
- 纵向上中下对齐
- 内容尺寸
- 动画
- 登录页面

# Debug

- (lldb) po self.view.constraints
- Debug View Hierarchy
- Flex

**More**

# Workflow

- (void)setNeedsUpdateConstraints;
- (void)updateConstraintsIfNeeded;
- (void)updateConstraints;
  
- (void)setNeedsLayout;
- (void)layoutIfNeeded;
- (void)layoutSubviews;
  
- (void)setNeedsDisplay;
- (void)drawRect:(CGRect)rect;



# Content

// 我就这么大

- (CGSize)intrinsicContentSize;

// 开始变形

- (void)invalidateIntrinsicContentSize;

// 抱紧

- (void)setContentHuggingPriority: forAxis::;

// 别挤我

- (void)setContentCompressionResistancePriority: forAxis::;

# UIViewController

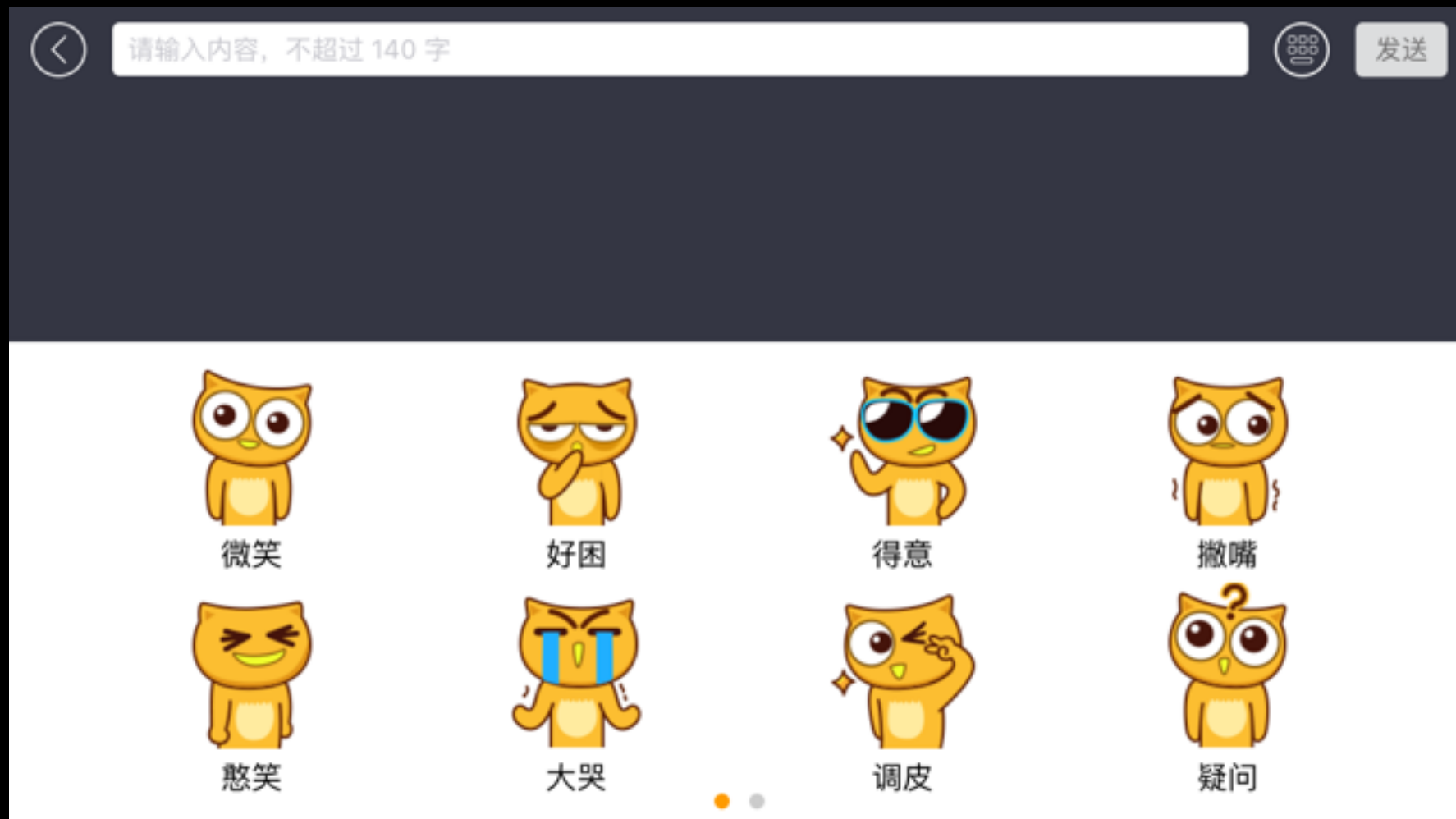
// !!!: 不要碰

```
@property UIRectEdge edgesForExtendedLayout;  
@property BOOL extendedLayoutIncludesOpaqueBars;  
@property BOOL automaticallyAdjustsScrollViewInsets;
```

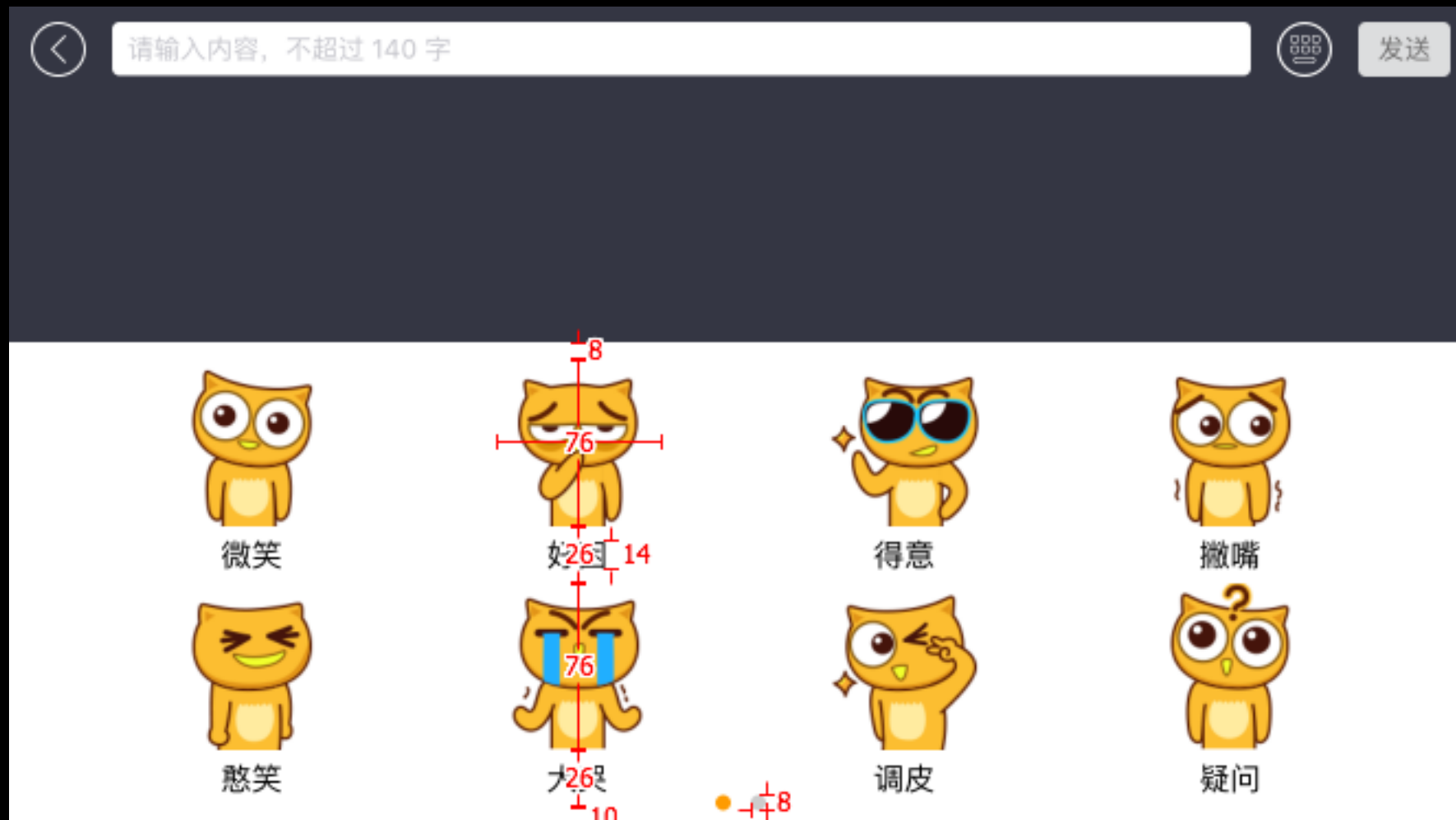
# UIViewController

```
@property id<UILayoutSupport> topLayoutGuide;  
@property id<UILayoutSupport> bottomLayoutGuide;  
  
- (void)viewDidLayoutSubviews;
```

# Homework



效果图



标注图

# Mark Man

<http://www.getmarkman.com/>

# Homework

[minglinqing@baijiahulian.com](mailto:minglinqing@baijiahulian.com)

Earlier than Next Tuesday



# git

<http://git.baijiahulian.com/iOS/training>

**the end**