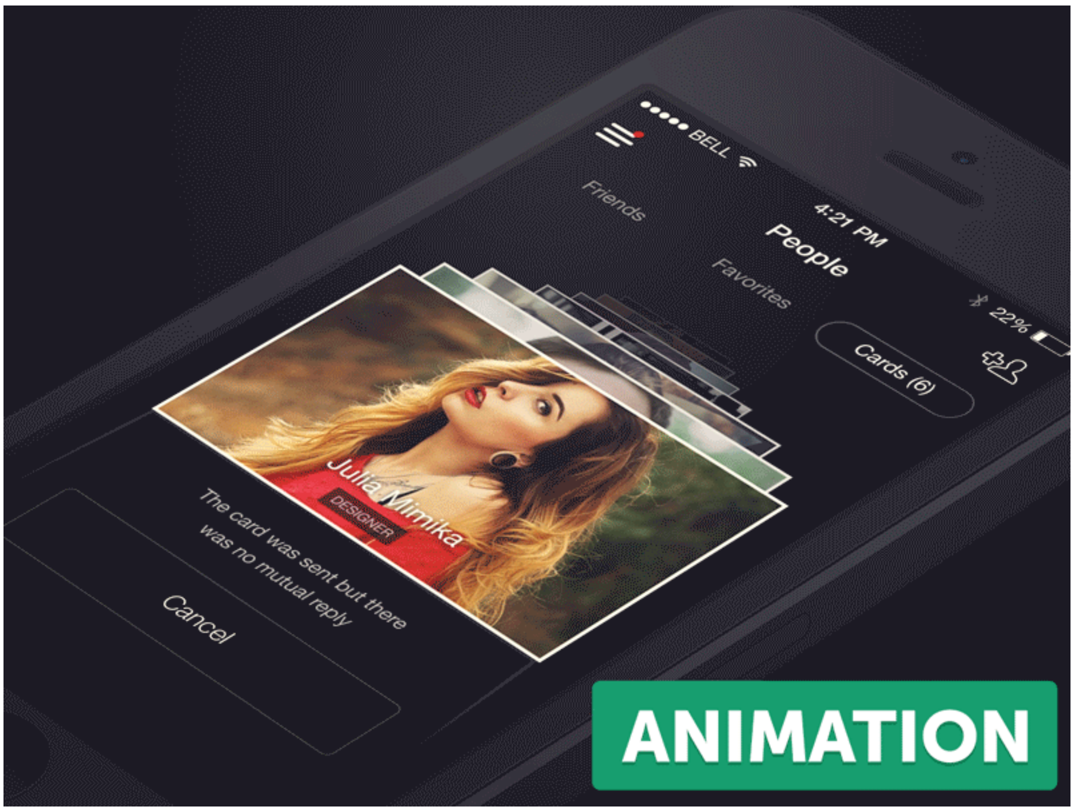
Card Animation Card Animation

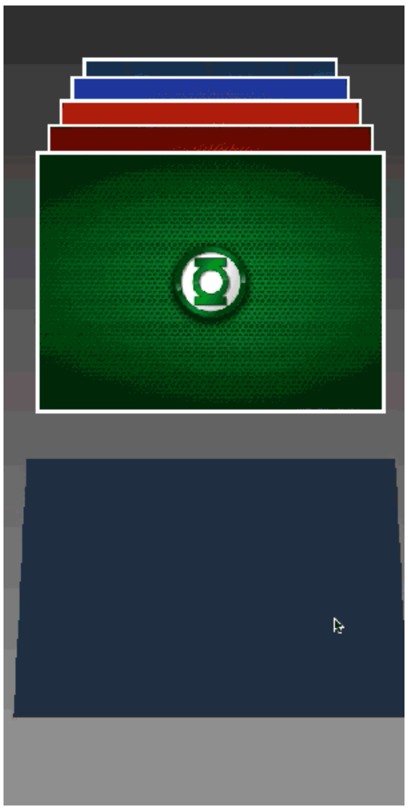
The author seedante
2015.10.08 23: 42 * Word Count 3552

Animated prototypes from Dribbble .



Card Animation.gif

Demo effect:



Renderings and reality .gif

Update: Recently refactored, easy to integrate.
Update: There are foreign enthusiastic users for my Demo under the modified can be very convenient in your project to use the effect, see the specific instructions on Github.

Source code: https://github.com/seedante/CardAnimation.git
Keywords: CALayer, transform, anchorPoint, Auto Layout

Animation analysis

First to solve the flip action. Look at the rotation of the diagram below, using the UIView transform property is unable to complete the action of the figure, because it only supports the rotation of the Z axis; here must use CALayer's transfrom attribute, which supports three latitude rotation.

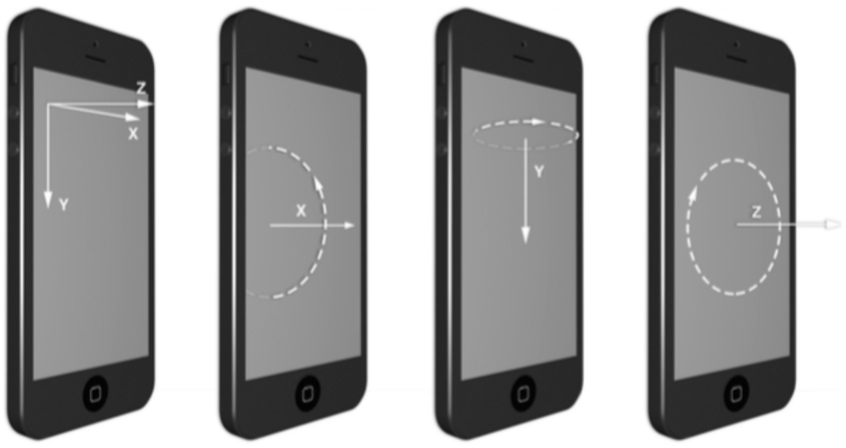


Figure 5.7 The X, Y, and Z axes, and the planes of rotation around them

Core Animation Rotation - iOS Core Animation Advanced Techniques

The above card animation is rotated along the X axis using `CATransform3DRotate(baseTransform, angle, 1, 0, 0)` . The transform values generated by the `CATransform3D` family of functions are all cumulative changes based on the incoming baseTransform, so adjusting the effect in the code often sees the iterations of a certain transform value.

```
var flipTransform3D = CATransform3DIdentity//从原始状态开始flipTransform3D.m34 = -1.0 / 1000.0//设定视觉焦点，分母越大表示视图离我们的距离越远，数值大有什么好处呢，你会发现翻转
```

Next, the rotation here does not rotate along the default center point, but at the bottom of the view, which means that we need to adjust the anchorPoint to `(0.5, 1)` , but we all know that adjusting the anchorPoint will cause the view's position to move, And anchorPoint relationship, recommend one I have seen the most clearly the blog .

On the release of the card, many people should know the "near big and small" perspective principle, I first know that this is a bird in the small comic book to see this painting skills to achieve in the two-dimensional plane to achieve different Distance feeling of scenery of scenery. The above animation in each card in the X-axis and Y-axis direction of the gap does not want to wait, the details of the praise. And the width of the photo frame is not the same, which is also in front of this detail match. So, we have to set the vertical spacing between the cards and the horizontal spacing, as well as the card frame width, you can set a function to calculate these parameters, the appropriate value needs to be debugged until you meet your requirements so far. After the rollover is complete, the subsequent cards are forwarded to the front card, which is the main frame (size and position) and borderWidth changes.

The brightness of the card in the animation is not the same, very much like the details. The solution is to add a foreground view in front of the image, a transparent view of black, and adjust the degree of darkness by adjusting the alpha property. Here the alpha value, the vertical distance of the card and the white border adjustment can be calculated by setting the formula, this part is not long-winded.

The details of the animation are very important.

Parts related to Auto Layout

These days rewrite the animation, completely using Auto Layout to achieve. Ray Wenderlich.com's articles Part I and Part II summarize the differences between the traditional Spring-Strut layout and the AutoLayout layout, which describes the layout relationship between the superView and the subView, but lacks a description of the layout between parallel subViews, the Auto Layout Make up the missing.

In the world of Auto Layout, the position and size of the view are determined by the constraints added to the view. This process is like we set a bunch of equations for the size and position of the view to give the Auto Layout. If the equation is solvable, then the layout of the view is ok; if there is no solution to the equation, there will be a conflict, you will see a lot of reports in the console; if the equation is not enough, Auto Layout can not give the only Solution, you can not determine the layout of the view.

In the adjustment of the card distance, Auto Layout can be automatically adjusted: the back of the card to add the card on the front of the distance constraints, modify the first card position constraints, you can automatically adjust the location of other cards, if the frame to achieve , Have to modify the frame of each card. But in this Auto Layout implementation, I still use the frame strategy, modify each card relative to the parent view centerY constraints. Why? Because the preceding card may be removed from the view, the constraint will disappear, or the previous card will be reused to modify the constraint, and the constraint between the two needs to change. Then, all the constraints on the centerY of the parent view, although the trouble needs to be modified one by one, but this constraint is much more stable.

The view in the storyboard only holds the constraints between itself and its own subviews as well as the constraints between their own subviews. Then the constraints of the card view are preserved in the parent view of the constraint, find out to modify:

```
UIView.animateWithDuration(0.3, { let centerXConstraint = superView.constraints.filter({$0.firstItem as? UIView == subView && $0.secondItem as? UIView == superView && $0.firstAttribute
```

Before the rollover, keep the anchorPoint of the card to be flipped to the bottom of the card, ie (0.5, 1) . So before dealing with anchorPoint, the reasons for the above mentioned blog :

```
//调整anchor point，并且保持视图位置不漂移。 cardView.frame = frame cardView.layer.anchorPoint = CGPointMake(0.5, 1) cardView.frame = frame
```

What about using Auto Layout? Statckoverflow two years ago to discuss this issue , the highest votes answered very exciting, but also incidentally answered the problem of transform and Auto Layout, the solution is to adjust the view embedded in the container view, in the container view to adjust the anchorPoint and rotation , Serve two purposes However, I have found another simpler way. Auto Layout will be the view of the constraints into frame, we only need to use the same way to do the same thing on it. After modifying the anchorPoint, the position of the view has moved, and the movement is compensated. The specific calculation method may be based on the constraints of the conditions to decide, this is not as simple as adjusting the frame.

```
 let oldConstraintConstant = centerYConstraint.constant subView.layer.anchorPoint = CGPointMake(0.5, 1) //关键代码：anchor point从(0.5,0.5)–>(0.5,1)，视图会往上移动自身高度的一半，那
```

How do I adjust the frame view of all card views? Use the transform to adjust the size of the Y-axis will also be on the pitch, for example, you set the two cards on the Y-axis spacing of 10, after scaling this distance is also reduced by the scale of the card itself, Pass; use Auto Layout , Can only be directly modified width and height constraints, and this animation I set the aspect ratio, only need to modify the width of the constraints on it, the problem is to use the constraints to determine the location, the available options are many, but the final effect And the constraints used have a great relationship (anchorPoint is the center of the zoom, modify the width and height of the effect seems to be a bit related to it, a bit confused, for the time being). I want to achieve the effect is in the top distance according to the results of the arrangement, but often modify the width of the constraints will also affect the distance. For example, the added constraint is Bottom, then the bottom of the card is the same as the plan; use Top will achieve the effect we want.

Animation need to use transform to achieve flip, then Auto Layout and Transform conflict? From iOS 8 from the two get along fairly happy, if you need to fit 8 before the version, sorry, this article is not resolved, in iOS 8 or before the frame to adjust it, the two disputes recommended reading this Article: Constraints & Transformations . In fact, there is no interaction with the Auto Layout, AutoLayout only valid for the constraints, transform and no modification of the constraints, the two do not interfere with each other. The relationship between the transform and the frame is also very interesting, the boundaries of the view of the two properties and the center has no effect, only the impact of the frame, they can verify in the code. Frame of the document that the view of the transform is not identity, the value is undefined, should be ignored. Many places undefined literal translation is undefined, in this context feel very strange, I think it should be said at this time can not determine the value of frame.

When generating UIView in the code, remember to change the translatesAutoresizingMaskIntoConstraints property to false . The view generated in the storyboard This property defaults to false, the view generated in the code The default property is true. This attribute is used to determine whether the frame-driven traditional Spring-Strut layout mode is the autoresize mask mixed with the AutoLayout mode. When the translatesAutoresizingMaskIntoConstraints property is true, the view's frame, bounds, center and other attributes of the changes into constraints. The intention of this mechanism is to automatically add Autoplaces without adding constraints manually. In general, the view is used before we are added to the constraints (I personally mainly out of insurance psychology), open the mechanism of the results is often a constraint conflict, so many people by surprise, that good mix of it. If the layout of the view will change in the future, do not open this property, the layout of things to the Auto Layout. On how to use this property, I explored in this article: the use of Auto Layout of the typical pain points and skills .

Transparent flip animation

When you flip the card, when the card is perpendicular to the screen and you continue to flip, you should only see the back of the card because the contents of the front of the card should be blocked. However, the view layer in the flip mode is transparent. The solution is: set the background color of the card view to the desired color, in the flip process when the card view and the screen perpendicular to the picture view hidden, Bingo, at the same time, improve the details of the borderWidth will be changed to 0. Why hide the picture view? Even if the background color of the card view is non-transparent, and the card view itself is set to be non-transparent, it will not achieve the desired effect. Only the image view will be hidden.

In addition, in the ordinary UIView animation flip 180 ℃ is a problem, and we need to flip to 90 ℃ when the contents of the card hidden, the use of keyFrame animation can be a perfect solution to this problem. In addition, modify the borderWidth in all UIView animation will not produce the effect of animation, it can not be the same as in the middle of the animation to modify borderWidth, this time had to use dispatch_after .

```
var flipDownTransform3D = CATransform3DIdentity //m34这个值用来表示视觉上焦点的位置，不明白的话，只需要知道设置的值越大相当于卡片离你的距离越远， flipDownTransform3D.m34 = -
```

Interactive animation

Add pan gestures to implement interactive animations. In the beginning of the gesture according to the positive and negative speed to determine the implementation of the operation.

```
case .Began: if velocity.y > 0{ //向下翻转卡片isInitiallyDown = true }else{ //将下方的卡片翻回上面isInitiallyDown = false }
```

In the gesture of the change stage, does not submit the animation, but adjust the card view of the flip angle and schedule match.

```
//在pan 手势里，根据手势在屏幕上移动的距离来判断进度： let percent = gesture.translationInView(view).y/150 //y 值可以为负，代表手指向上移动//0<percent<1，向下翻卡片flipTransform3D
```

At the end of the gesture, the animation is submitted to complete the rest of the thing.

```
//翻转初始方向为向下时，翻转程度达到一半时自动完成这个翻转if percent >= 0.5{ UIView.animateWithDuration(0.3, animations: { //此时卡片剩余翻转的角度小于180℃，没有问题，这个奇怪。 c
```

In-depth reading:

1. Completely understand position and anchorPoint
2. How I Learned to Stop Worrying and Love Cocoa Auto Layout
3. Auto Layout Guide

4. WWDC15 Session 219: Mysteries of Auto Layout, Part 1
5. WWDC15 Session 219: Mysteries of Auto Layout, Part 2
6. Auto Layout Tutorial in iOS 9 Part 1: Getting Started
7. Auto Layout Tutorial in iOS 9 Part 2: Constraints
8. Stackoverflow: How do I adjust the anchor point of a CALayer, when Auto Layout is being used?
9. Constraints & Transformations: How Auto Layout quietly arrived transform-friendly in iOS 8

seedante

Ask me to drink Coke ^ _ ^

Animation

Report an article

Seedante

Message is the answer to the premise that I have time and energy, do not have to add any complacency of the call, such as the gods and the like, with such a call I will not return. Secondly, there are questions or run my Demo error, please describe the problem clearly, including Xcode version and iOS version, simulator or real machine; limited to personal energy, can not be updated for each version, if the demo when running version The problem is wrong, uh, you try to solve the next. My Github: https://github.com/seedante Recommended iOS Website: 0. https://developer.apple.com/videos/ 1. http://www.raywenderlich.com 2. http: // nshipster. Com 3. http://www.objc.io 4. https://realm.io iOS Latest APIs and New Features Usage: https://www.shinobicontrols.com/blog's day-by-day series

More sharing