Core Animation Essentials

Session 421

Michael Levy
Graphics & Imaging (Canada)

Tim Oriol
Graphics & Imaging

Demo 1

What can you do with Core Animation?

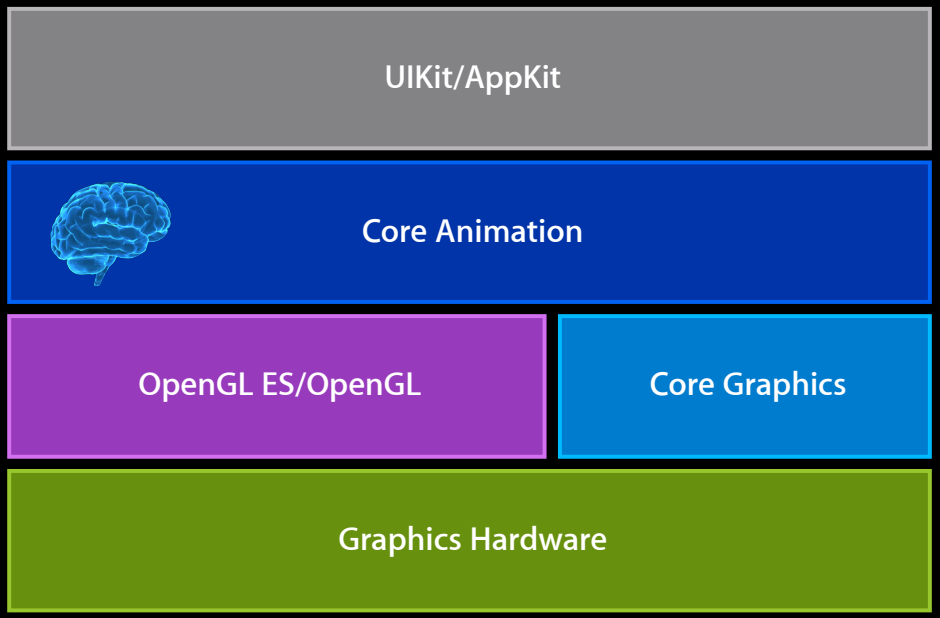
Core Animation in Practice

- What will we cover?
  - Part 1: Fundamental Concepts
    - Layers and layer properties
    - Animating layer properties
  - Part 2: Topics in Core Animation
    - Layers, 3D transforms, and perspective
    - Presentation versus model
    - Notifications and timing
    - Performance
    - Masks and shadows
  - Part 3: A little bit extra

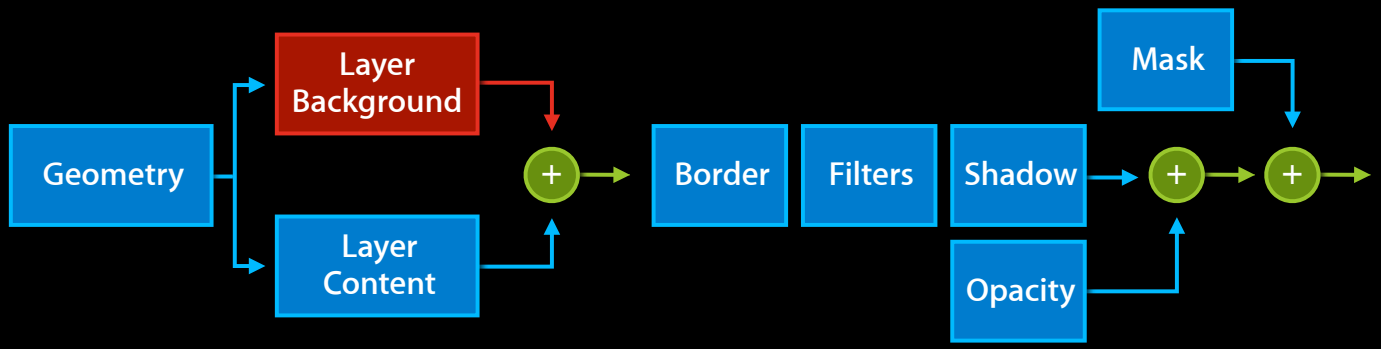
Architectural Overview

Core Animation

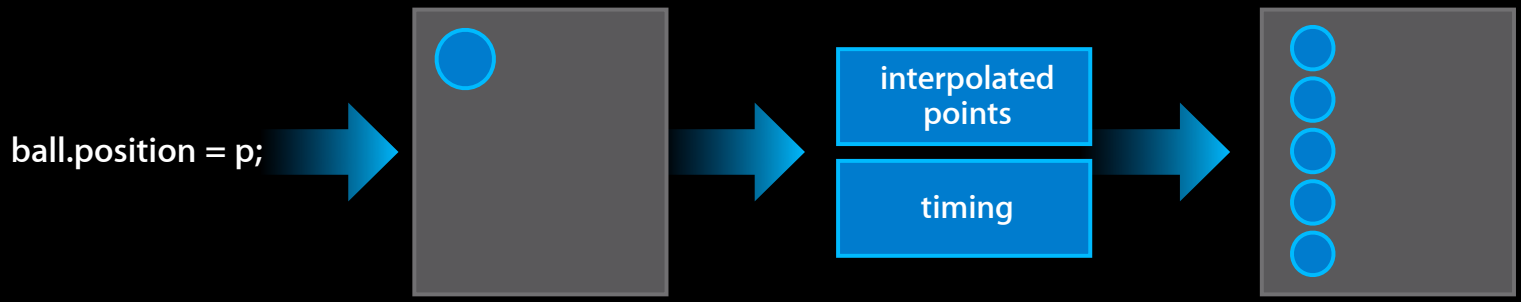
Architecture



Compositing Model



Animation



Part 1

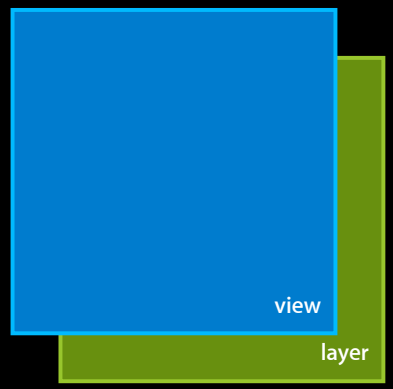
The basics

Welcome to the Layer Cake

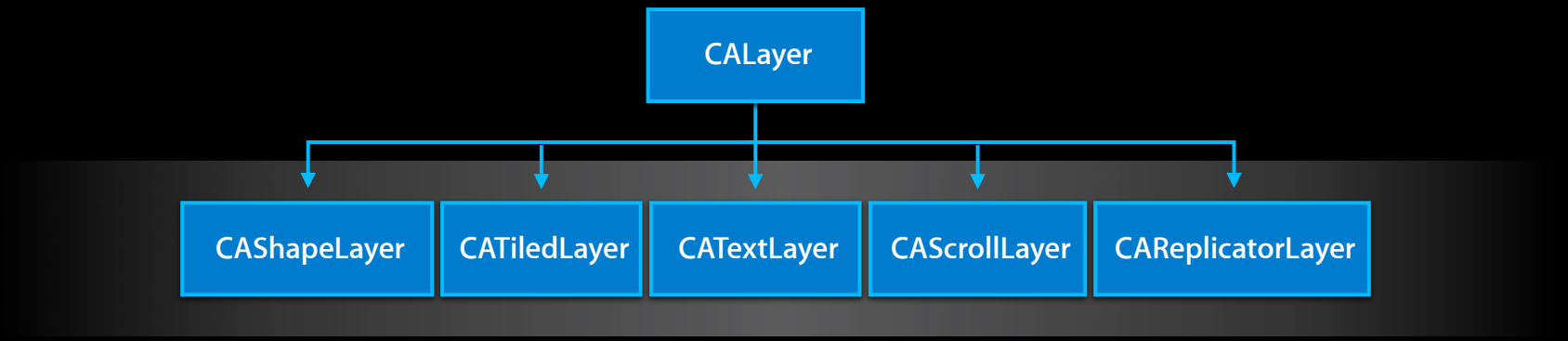
CALayer

Layers in iOS

- Every UIView has a CALayer
  - view.layer
  - drawRect renders to the layer



Layer Hierarchy



Demo

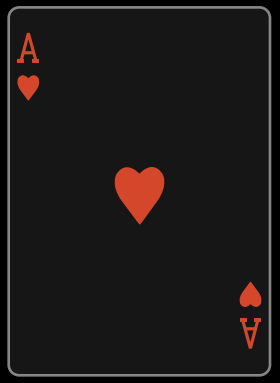
Card

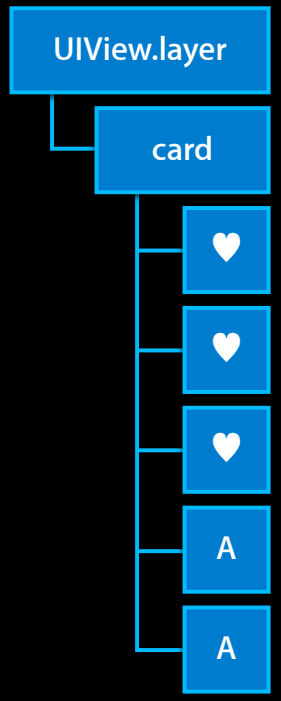
Creating Layers

```
#import <QuartzCore/QuartzCore.h>
...
CALayer* myLayer = [CALayer layer];
myLayer.bounds = CGRectMake(0,0,w,h);
myLayer.position = CGPointMake(x,y);
myLayer.content = familyImage;
[view.layer addSubLayer:myLayer];
```

Layers and Sublayers

- Model similar to UIView
  - addSublayer:
  - insertSublayer:above: (etc.)
  - setNeedsLayout
  - layoutSublayers
  - setNeedsDisplay
  - drawInContext:
- delegate
  - drawLayer:inContext: (delegate)
- 2.5D model
  - Transform is 3D

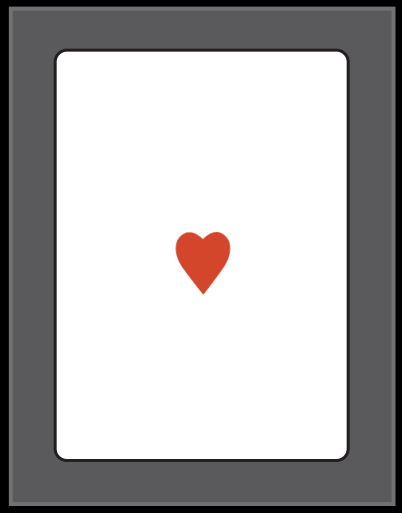




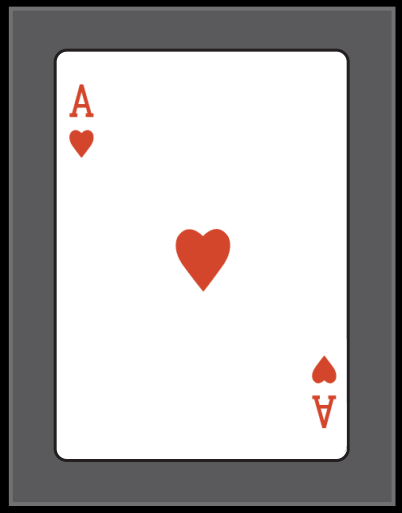
Declarative Style

```
heartAce = [CALayer layer];
heartAce.cornerRadius = 14;
...
// Add to view's layer
[self.layer addSublayer: heartAce];

// Add the pips
// Center
CAShapeLayer* centerPip = [Cards heartPip];
centerPip.position = CGPointMake(...,...);
[heartAce addSubLayer:centerPip];
```

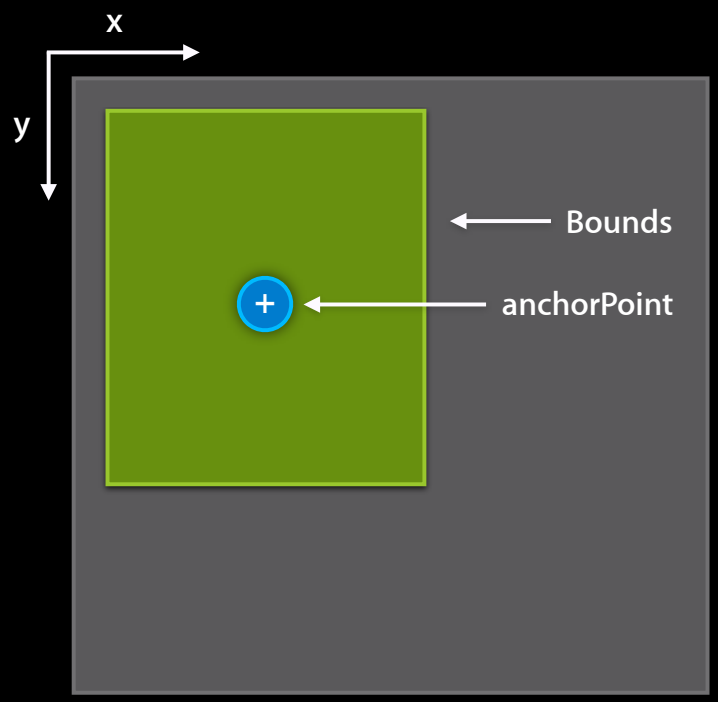


```
// The other pips
CAShapeLayer* bottomPip = [Cards heartPip];
CATransform3D transform = CATransform3DMakeScale(0.5, 0.5, 1);
transform = CATransform3DRotate(transform, M_PI, 0, 0, 1);
bottomPip.transform =transform;
bottomPip.position = ...
[heartAce addSublayer:bottomPip];
...
```



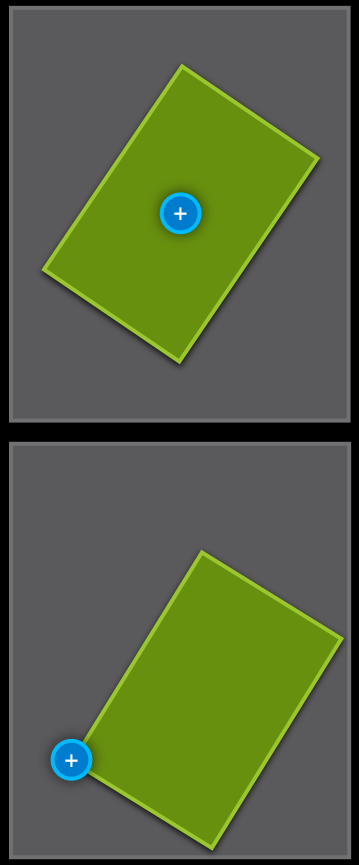
Layers

```
bounds—CGRect
position—CGPoint (superlayer coordinates)
anchorPoint—CGPoint
transform—CATransform3D
```



```
// A: Rotate about center
layer.anchorPoint = CGPointMake(0.5,0.5);
layer.transform = rotationTransform;
```

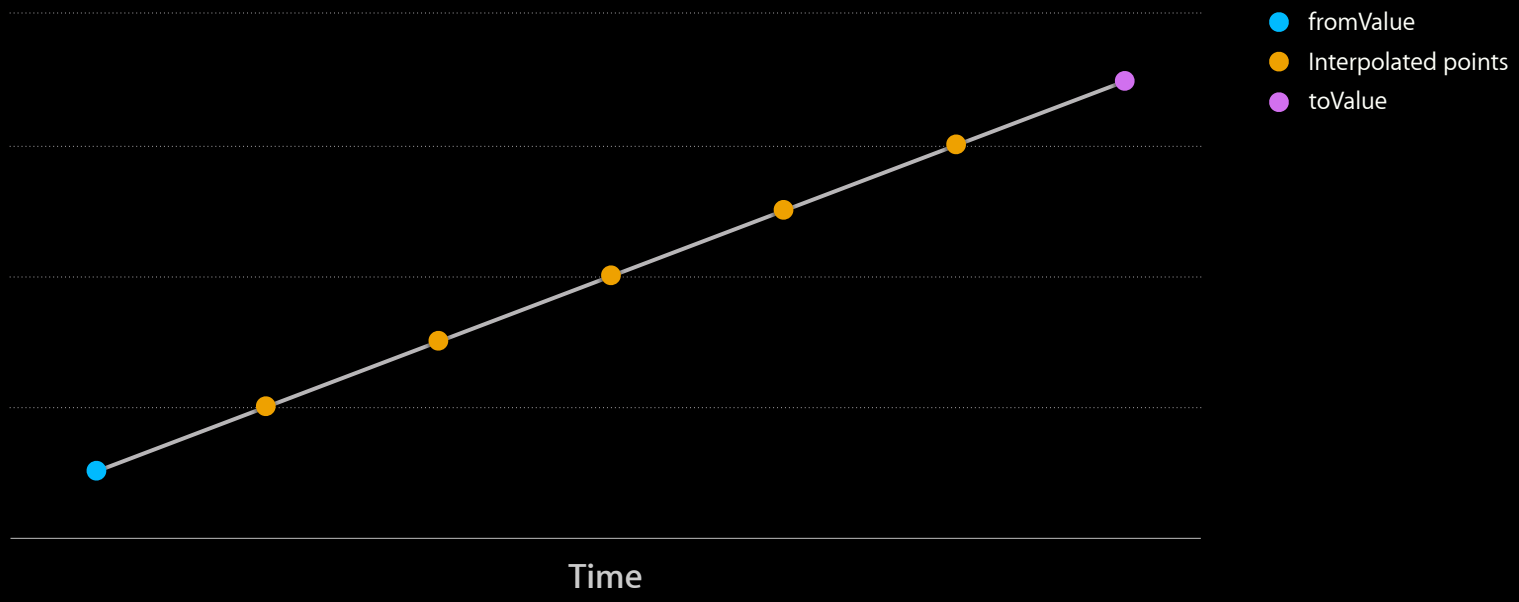
```
// B: Rotate about lower left
layer.anchorPoint = CGPointMake(0.0,1.0);
layer.transform = rotationTransform;
```



Animation

Animation

What does animation mean?



Implicit Animation

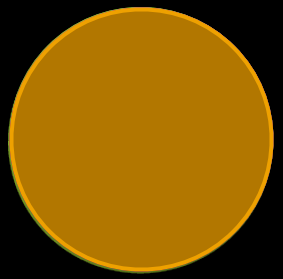
Animation

```
myLayer.opacity = 0;
```

```
[CATransaction setAnimationDuration:2]
myLayer.position = nearBottom;

[CATransaction setAnimationDuration:5]
myLayer.opacity = 0;
```

What Type of Things Can Be Animated?



Transactions

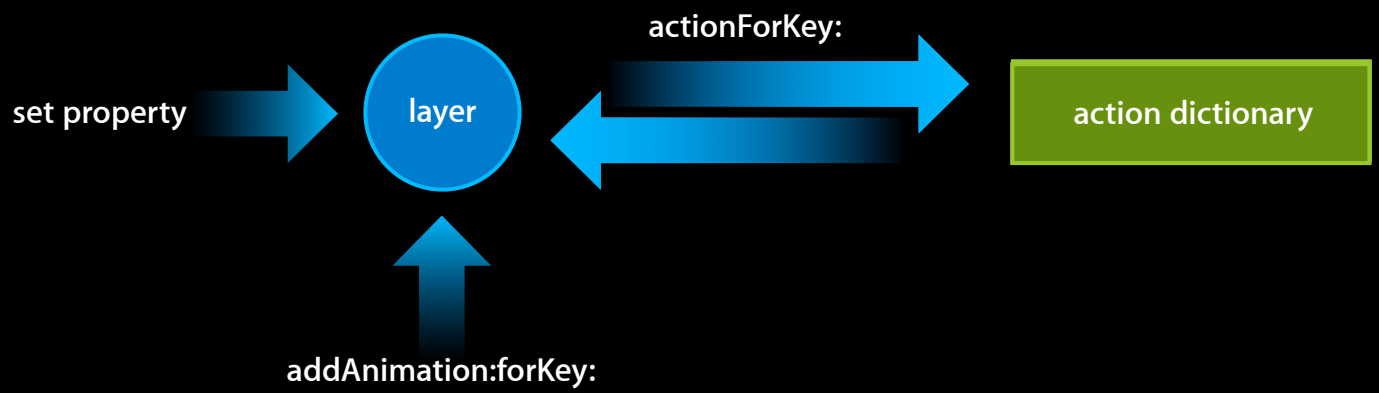
- All applied together in run loop
- CATransaction class
  - Duration
  - Timing function
  - CATransaction properties at time implicit animation is created

```
[CATransaction setDisableActions:YES]
```

Implicit Animation

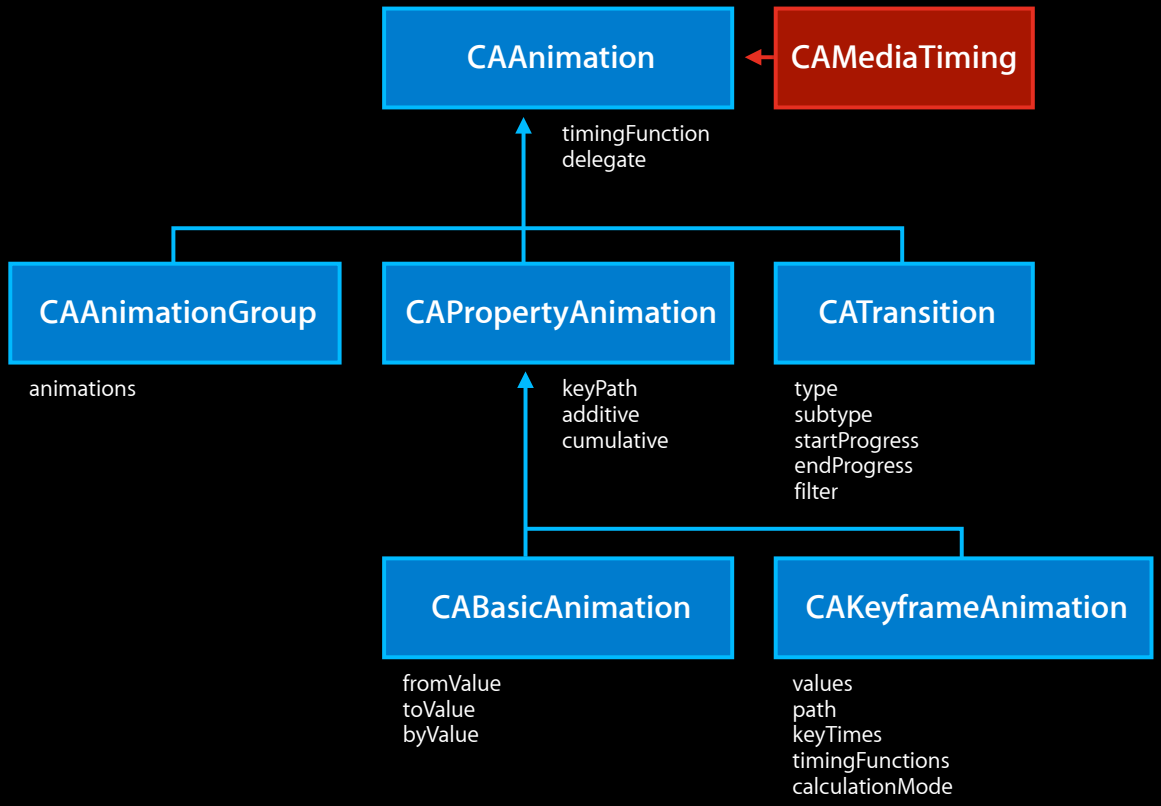
Inside the mind of Core Animation

- CAAction protocol
  - Action is an object that can respond to events
  - CAAnimation implements CAActionProtocol
  - Action (if found) added using `[self addAnimation:forKey:]`



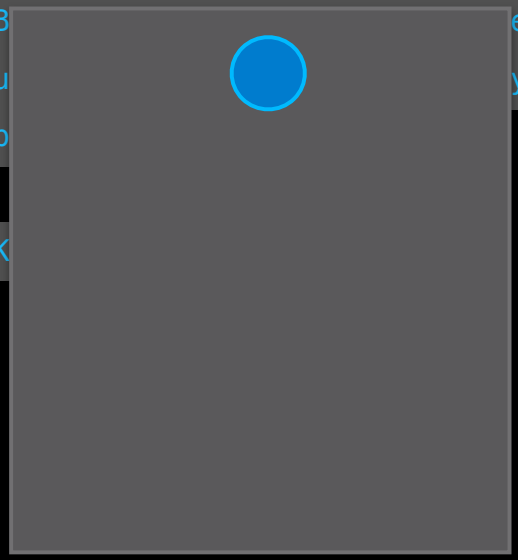
Explicit Animation

Explicit Animation



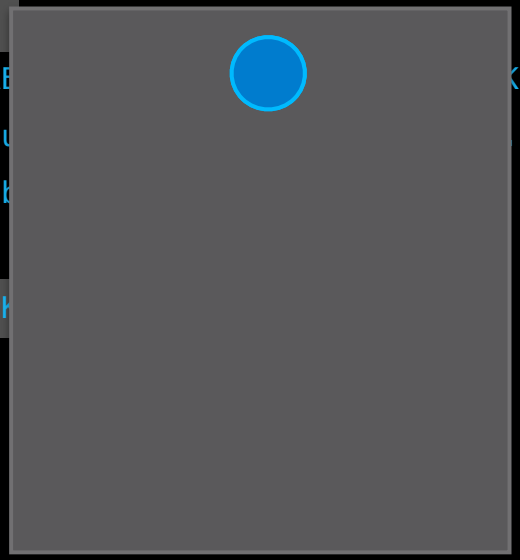
Example

```
CABasicAnimation* drop = [CAB                    eyPath:@"position.y"];
drop.fromValue = [NSNumber nu                    y];
drop.toValue = [NSNumber numb
drop.duration = 5;
[layer addAnimation:drop forK
```



Example

```
layer.position = toPosition;
CABasicAnimation* drop = [CAB                    KeyPath:@"position.y"];
drop.fromValue = [NSNumber nu                    y];
drop.toValue = [NSNumber numb
drop.duration = 5;
[layer addAnimation:drop forK
```

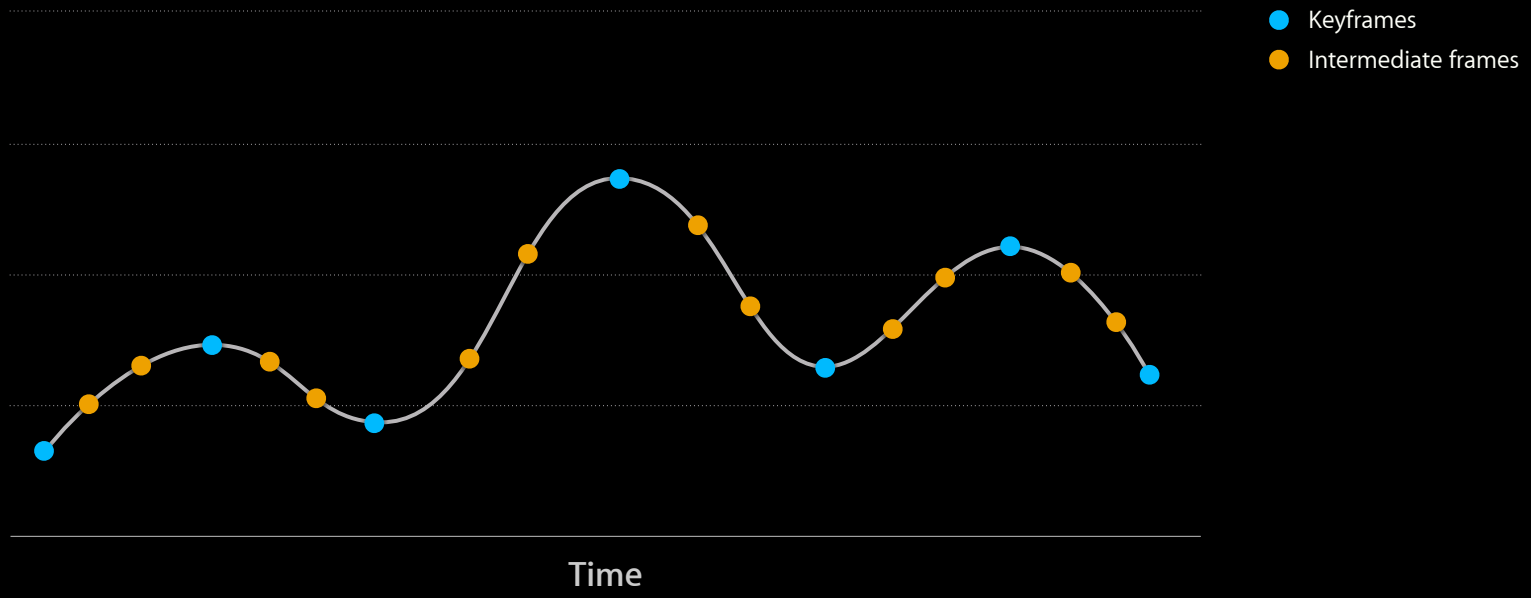


Animation

Keyframe animation

Animation

Keyframes



Keyframe Animations

- Use either
  - `path`
  - `values`
- `keyTimes` (optional)—Fraction of total time for each keyframe segment
- Interpolation either between values or along path
- `calculationMode`
  - Linear
  - Discrete
  - Cubic

Group Animation

- Collection of animations
  - Applied simultaneously to layer's properties
  - Timings clipped to group timing

```
CABasicAnimation* a1 = ....
CAKeyFrameAnimation* a2 = ...
....
CAAnimationGroup* group = [CAAnimationGroup animation];
group.animations = [NSArray arrayWithObjects:a1,a2,...,nil];
group.duration = ...
[layer addAnimation:group forKey:nil];
```

Demo

Bouncing ball

Demo Code Snippet

```
// Create a KeyFrame animation for the path of the ball
// 4/4 time
CGFloat tempo = 150; // 150 quarter notes per minute

CAKeyframeAnimation* bounce
      = [CAKeyframeAnimation animationWithKeyPath:@"position"];
bounce.values = values;
bounce.duration =   numberOfNotes/tempo*60; // seconds
bounce.calculationMode = kCAAnimationCubic;
```

Demo Code Snippet Continued

```
//Animate opacity out between each line
CAKeyframeAnimation *opacityAnim
    = [CAKeyframeAnimation animationWithKeyPath:@"opacity"];

opacityAnim.duration = delay + (lines * notesperline / tempo * 60.0);

for (int i=1; i < lines; i++) {
   float time_out = delay + i * ....;
   float time_in = 2 / tempo * 60.0 + time_out;
   ... add to times and val arrays
}

opacityAnim.keyTimes = times;
opacityAnim.values = val;
```

Demo Code Snippet Conclusion

```
CAAnimationGroup* group = [CAAnimationGroup animation];
group.animations = [NSArray arrayWithObjects:bounce,opacityAnim, nil];
group.duration = bounce.duration;

[_ball addAnimation:group forKey:@"karaoke"];
```

Summary

- Use CALayers for content
- Change of layer property will schedule an implicit animation
- CATransaction class for animation attributes
- Explicit animation
  - CABasicAnimation
  - CAKeyFrameAnimation
  - CAGroupAnimation

Part 2

Topics

Demo

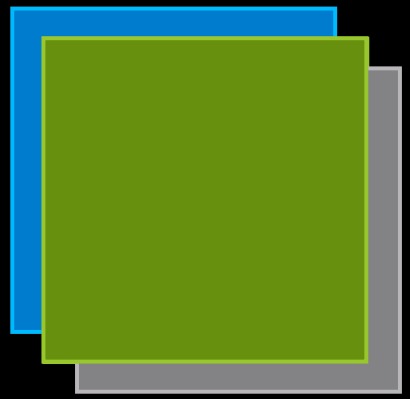
Layers in perspective

Core Animation Topics

The Z coordinate in perspective

2.5D Model

```
[parent addSublayer:blueLayer];
[parent addSublayer:greenLayer];
[parent addSublayer:grayLayer];
greenLayer.zPosition = 500;
```



- Summary
  - zPosition can be used to determine composite order
  - Layer's size in parent layer does not change

Demo

Visualizing perspective

Perspective

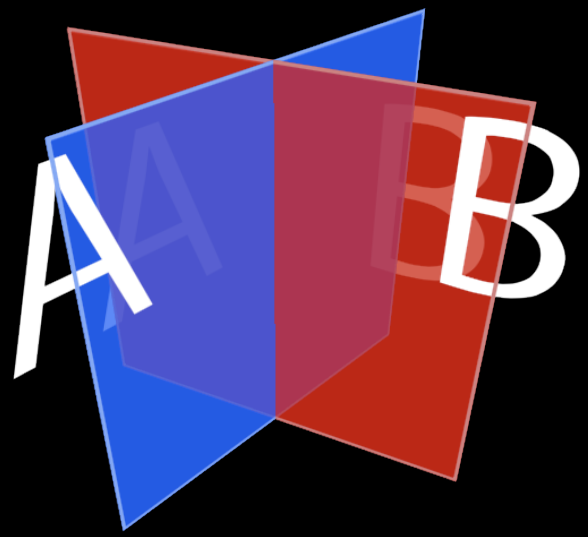
- Use “subLayerTransform” property
  - Homogeneous perspective transform

```
CATransform3D perspective = CATransform3DIdentity;
perspective.m34 = -1./EYE_Z;

CALayer* parent = [CALayer layer];
...
parent.sublayerTransform = perspective;
...
blueLayer.zPosition = -100;
[layer addSublayer:blueLayer];
...
[layer addSublayer:greenLayer];
...
grayLayer.zPosition = 100;
[layer addSublayer:grayLayer];
```

2.5D—Depth Sorting

- Intersecting layers are supported but best avoided
- Nontrivial extra work for renderer
- Layers are rendered more than once
- Depth-sorting uses layer bounds and position to determine occlusion



Core Animation Topics

Notifications and timing

Notifications and Timing

- CAMediaTiming protocol
  - Adopted by CAAnimation and CALayer
  - Properties
    - `beginTime`
    - `repeatCount, repeatDuration`
    - `duration`
    - `autoreverses`
    - `fillMode`

Notifications and Timing

- Animations created explicitly can use a delegate

```
myAnimation.delegate = self;
.....
- (void)animationDidStart:(CAAnimation *)theAnimation {
...
}

- (void)animationDidStop:(CAAnimation *)theAnimation finished:(BOOL)flag {
...
}
```

Notifications and Timing

- Implicit animations can use completion block

```
[CATransaction setCompletionBlock:^{
    // block that runs when animations have completed
    [CATransaction setDisableActions:YES];
    [layer removeFromSuperlayer];
}];
layer.opacity = 0;
layer.position = CGPointMake (2000, layer.position.y);
```

Notifications and Timing

Tip

- Use notifications for setup and teardown
  - For timing, can use CAMediaTiming protocol

Using CAMediaTiming

```
...
CFTimeInterval localMediaTime = [_host convertTime:CACurrentMediaTime() fromLayer:nil];
NSUInteger k = 0;
for(balloon in _balloons) {
    CABasicAnimation* animation = [CABasicAnimation animationWithKeyPath:@"position.y"];
    animation.autoreverses = YES;
    ...
    floatAnimation.duration = 5;
    floatAnimation.beginTime = localMediaTime+k;
    [balloon addAnimation:floatAnimation forKey:@"position.y"];
    k += 5;
}
```

Notifications and Timing

Example: Two seconds after scrolling stops, fade a HUD

```
myHud.opacity = 0;
CFTimeInterval now = [myHud converTime:CAMediaCurrentTime() fromLayer:nil];
CABasicAnimation* fadeOut = [CABasicAnimation animationWithKeyPath:@"opacity"];
fadeOut.fromValue = [NSNumber numberWithFloat:.5];
fadeOut.toValue = [NSNumber numberWithFloat:0];
fadeOut.duration = 5;
fadeOut.beginTime = now + 2;
fadeOut.fillMode = kCAFillModeBackwards;
[myHud addAnimation:fadeOut forKey:@"opacity"];
```

Core Animation Topics

Presentation versus model

Presentation Versus Model

- Layer properties do not reflect active animations
- Use -presentationLayer method to get screen values
  - Creates a temporary layer with animations applied
  - Asking for sublayers returns presentation versions
- Useful for from values of animations

```
anim = [CABasicAnimation animationWithKeyPath:@”borderColor”];
anim.fromValue = [[layer presentationLayer] borderColor];
```

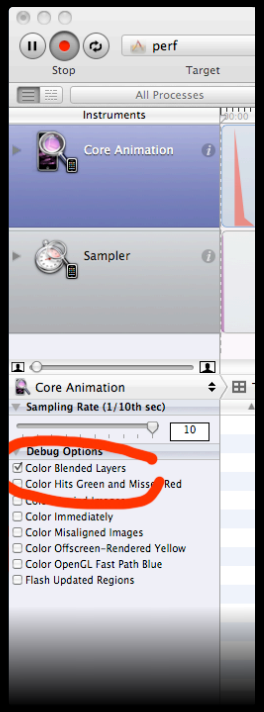
- And for hit-testing against real geometry

```
hitLayer = [[[layer presentationLayer] hitTest:p] modelLayer];
```

Performance

Performance

- Design with performance in mind
- Use opaque layers
- Avoid CAShapeLayer with complex paths
- Avoid offscreen rendering, e.g.:
  - Masks, dynamic shadows, group opacity
- General tips
  - Reduce size of content
  - Remove expensive compositing steps


perf
Stop
Target
All Processes
Instruments
Core Animation
Sampler
Core Animation
Sampling Rate (1/10th sec)
10
Debug Options
Color Blended Layers
Color Hits Green and Miss
Red
Color Immediately
Color Misaligned Images
Color Offscreen-Rendered Yellow
Color OpenGL Fast Path Blue
Flash Updated Regions

Core Animation Topics

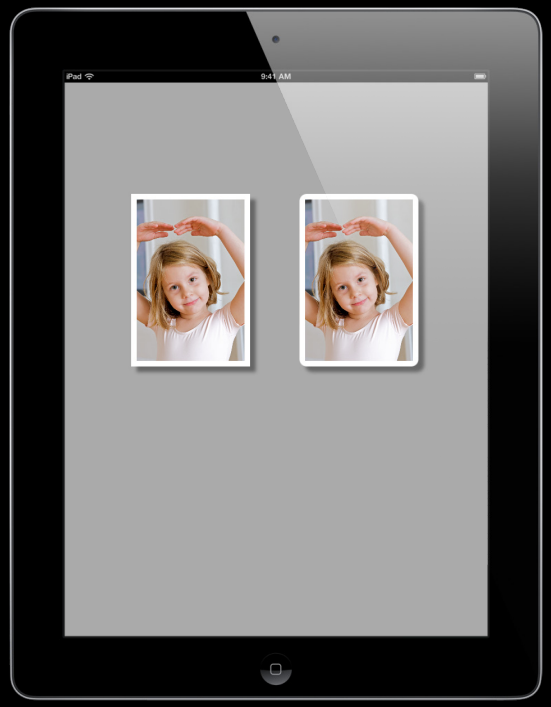
Masks and shadows

Shadows

- Shadows provide crucial depth cues in 2D user interfaces
- Uses alpha channel to compute shadow

```
layer.shadowOpacity = 0.8;
layer.shadowColor = shadowColor;
layer.shadowOffset = CGSizeMake(10, 10);

layer.shadowPath = somePath;
```

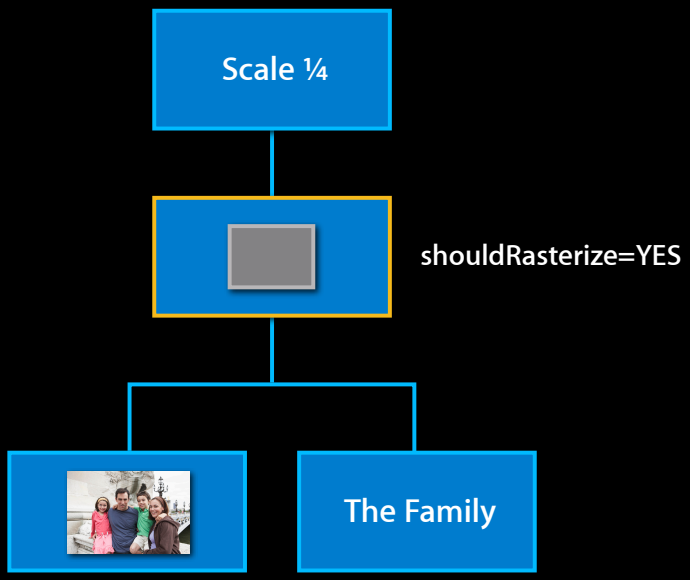


Masks

- Mask property
  - Alpha channel used to mask
  - Can use any CALayer as mask

```
CALayer* subLayer = [CALayer layer];
subLayer.contents = (id) beachImage;
CALayer* star = [CALayer layer];
star.contents = [self makeStarImage];
subLayer.mask = star;
[parent addSubLayer: subLayer];
```

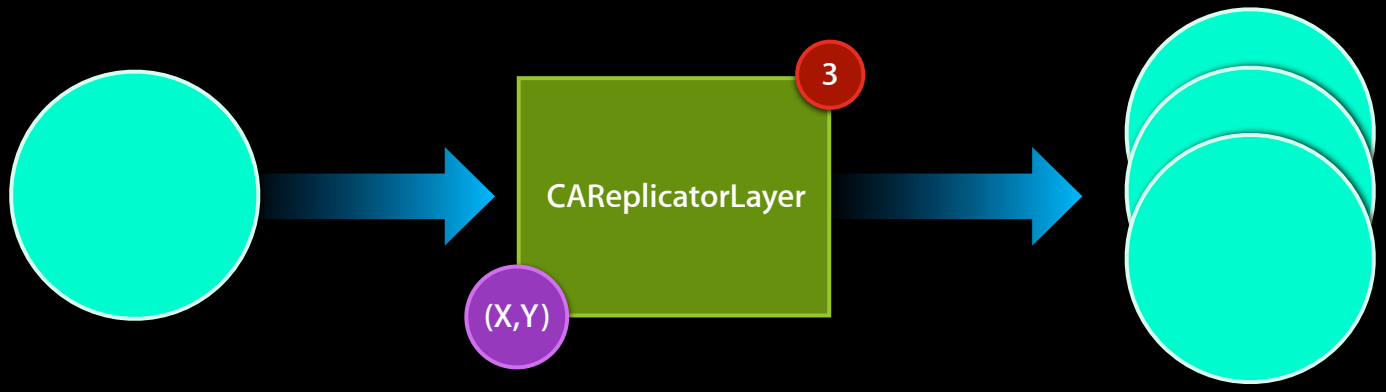
Bitmap Caching



- Subtree is rendered once into cache
- Cache used subsequently
- Caveats
  - limited cache space
  - Caching and not-reusing more expensive than not caching
  - Rasterizing locks layer image to a particular size
  - Rasterization occurs before mask is applied

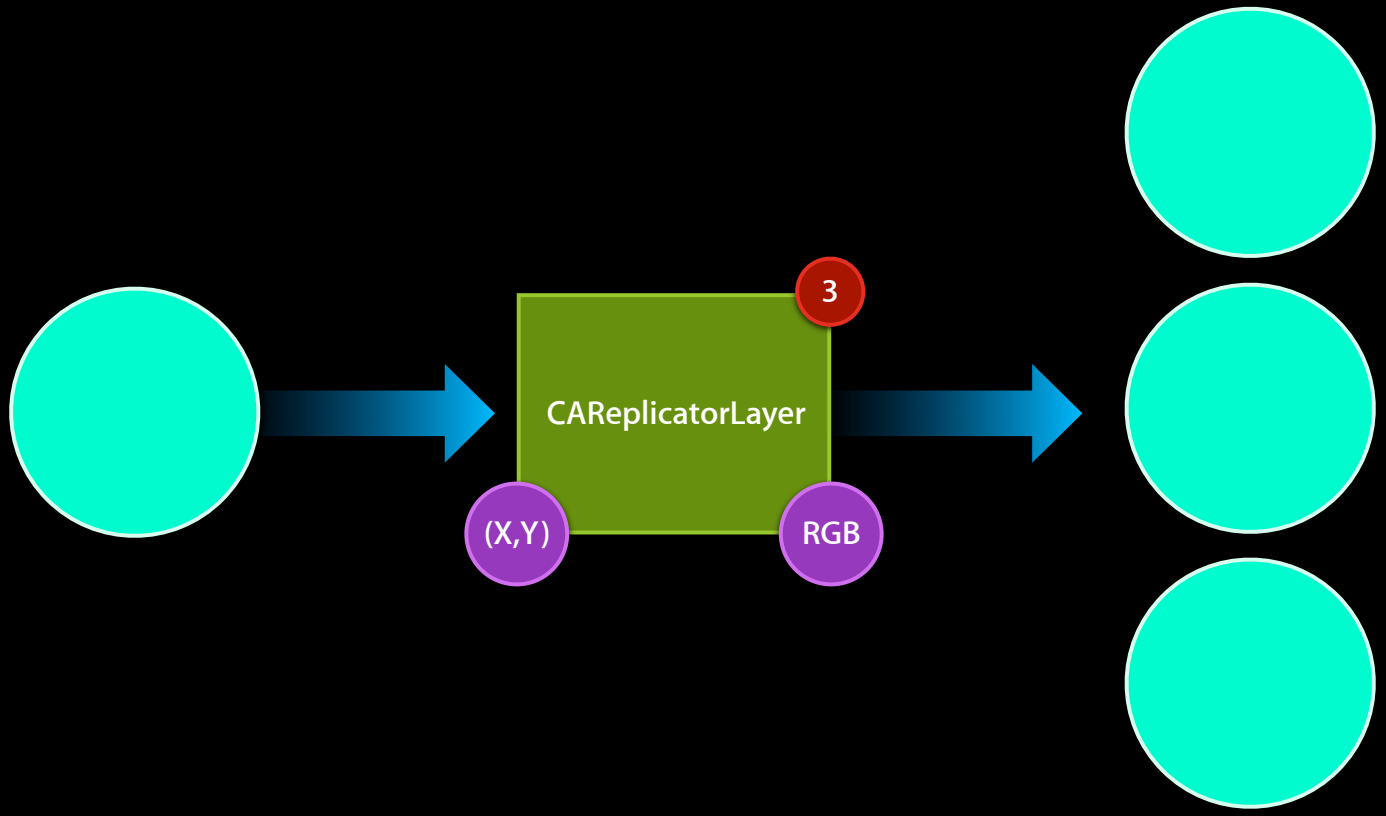
A Little Extra

Replicators



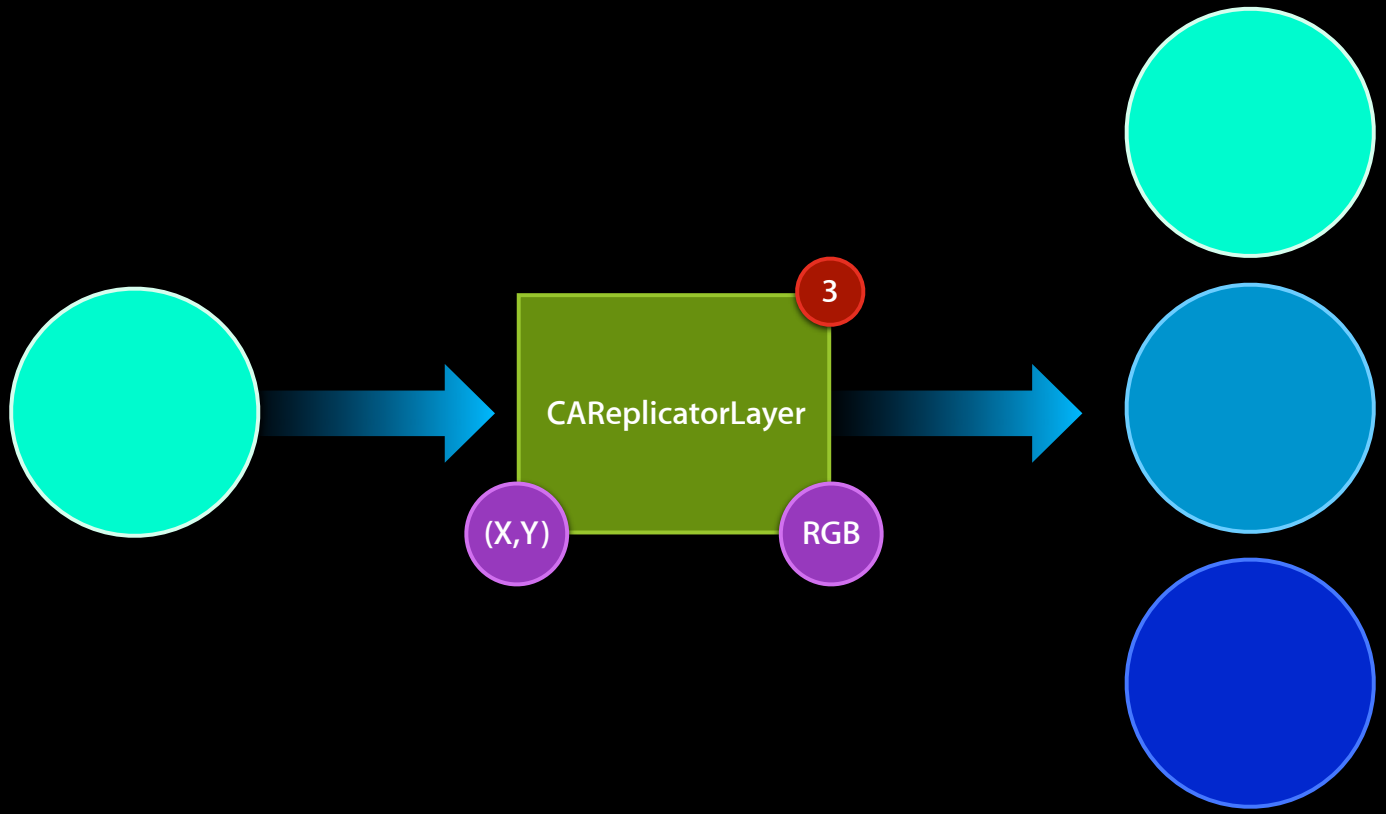

```
replicator.instanceCount = 3;
replicator.instanceTransform = CATransform3DMakeTranslation(0, 50, 0);
[replicator addSublayer: aLayer];
```

Replicators



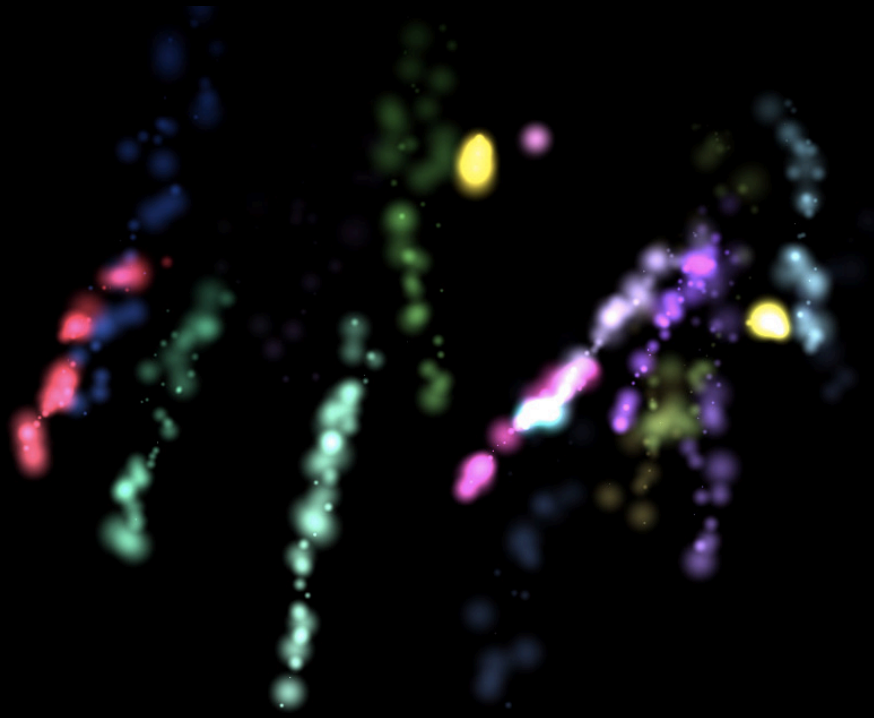

```
replicator.instanceTransform = CATransform3DMakeTranslation(0, 50, 0);
```

Replicators

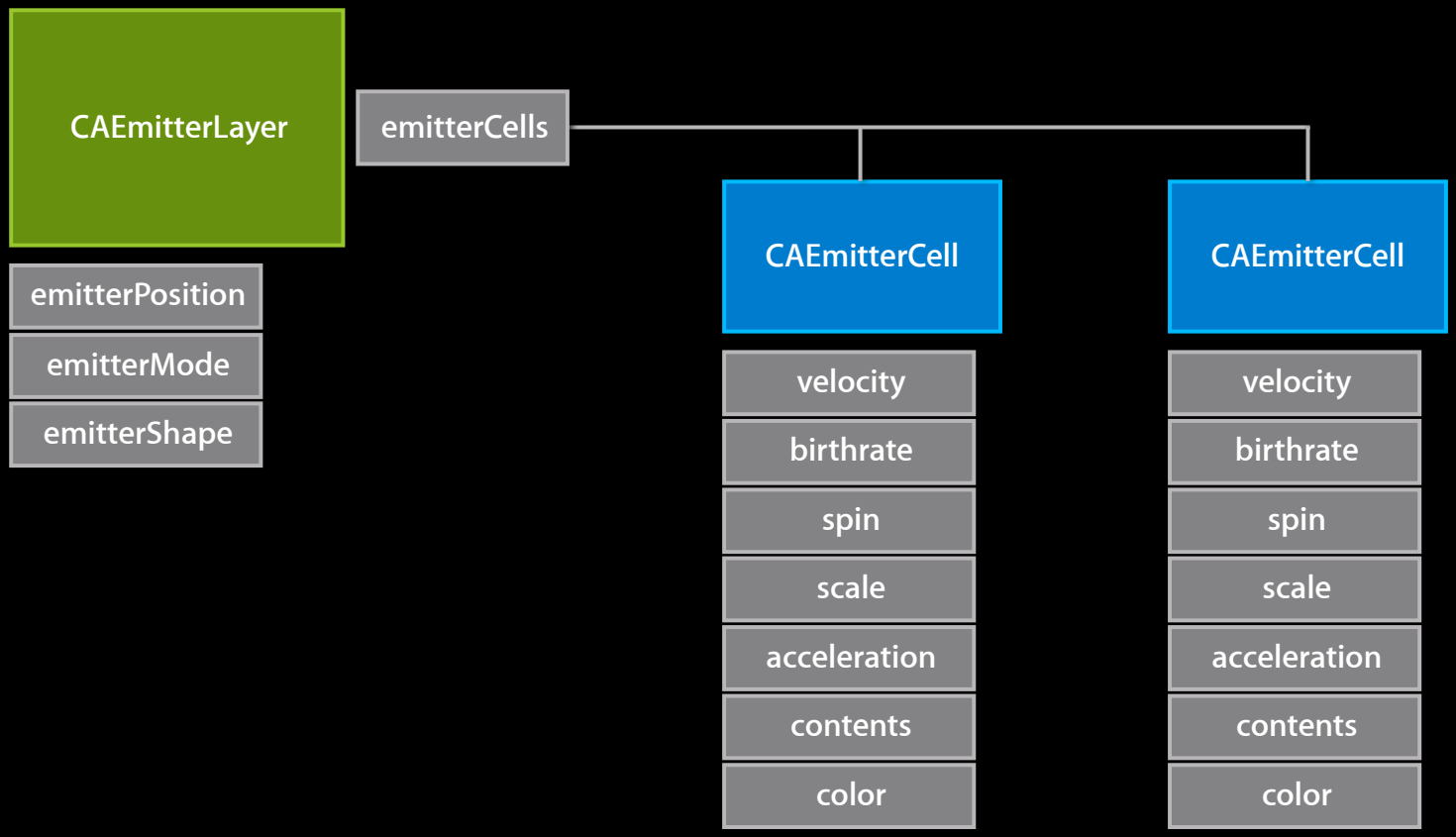



```
replicator.instanceGreenOffset = -0.5;
```

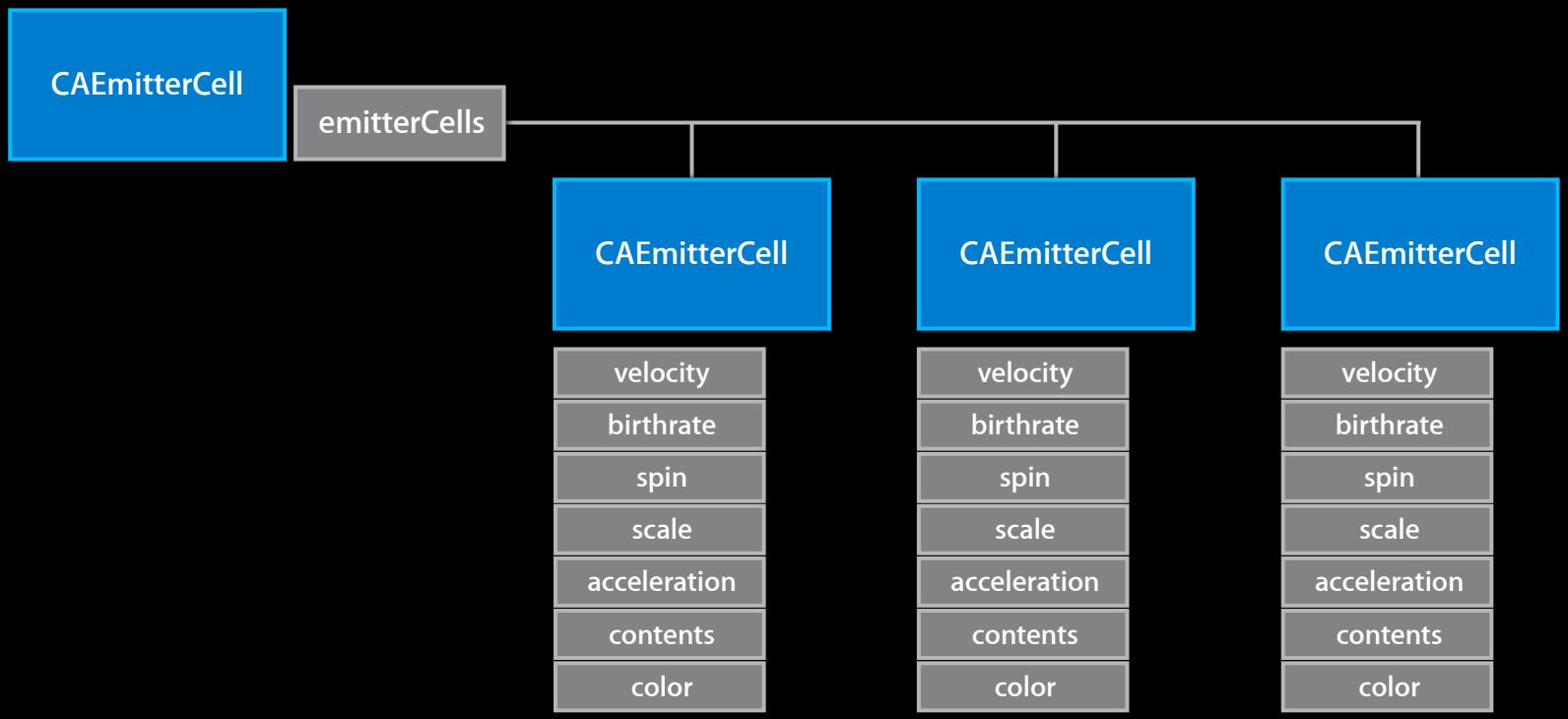
Particles



Particles


CAEmitterLayer
emitterCells
CAEmitterCell
CAEmitterCell
emitterPosition
emitterMode
emitterShape
velocity
birthrate
spin
scale
acceleration
contents
color
velocity
birthrate
spin
scale
acceleration
contents
color

Particles


CAEmitterCell
emitterCells
CAEmitterCell
CAEmitterCell
CAEmitterCell
velocity
birthrate
spin
scale
acceleration
contents
color
velocity
birthrate
spin
scale
acceleration
contents
color
velocity
birthrate
spin
scale
acceleration
contents
color

Demo

Particles

Related Sessions

Understanding UIKit Rendering	Mission Thursday 10:15AM

Labs

Core Animation Lab	Graphics, Media & Games Lab C Thursday 2:00PM

More Information

Allan Schaffer
Graphics and Game Technologies Evangelist
aschaffer@apple.com

Apple Developer Forums
http://devforums.apple.com


WWDC2011