iPhone Application Programming
Lecture 6: Drawing

Simon Voelker
Media Computing Group
RWTH Aachen University

Winter Semester 2013/2014

http://hci.rwth-aachen.de/iphone

RWTH AACHEN UNIVERSITY

Drawing

Quartz & CoreGraphics	**Core Animation**	**Spirte Kit**	**OpenGL**
Mostly Vector Drawing	Mostly Bitmap Drawing	Spirte Game Engine	Mostly Polygon Drawing
2D	2.5D	2.5D	
Tell how do draw	Tell what to draw and how to animate it	Create scene graph, and physics; apply actions	Tell how do draw

Not covered this year

Quartz

Quartz & CoreGraphics

- C-based
- 2D drawing engine
- Path-based drawing
- Transparency, shading, shadows, layers
- Hardware acceleration whenever possible

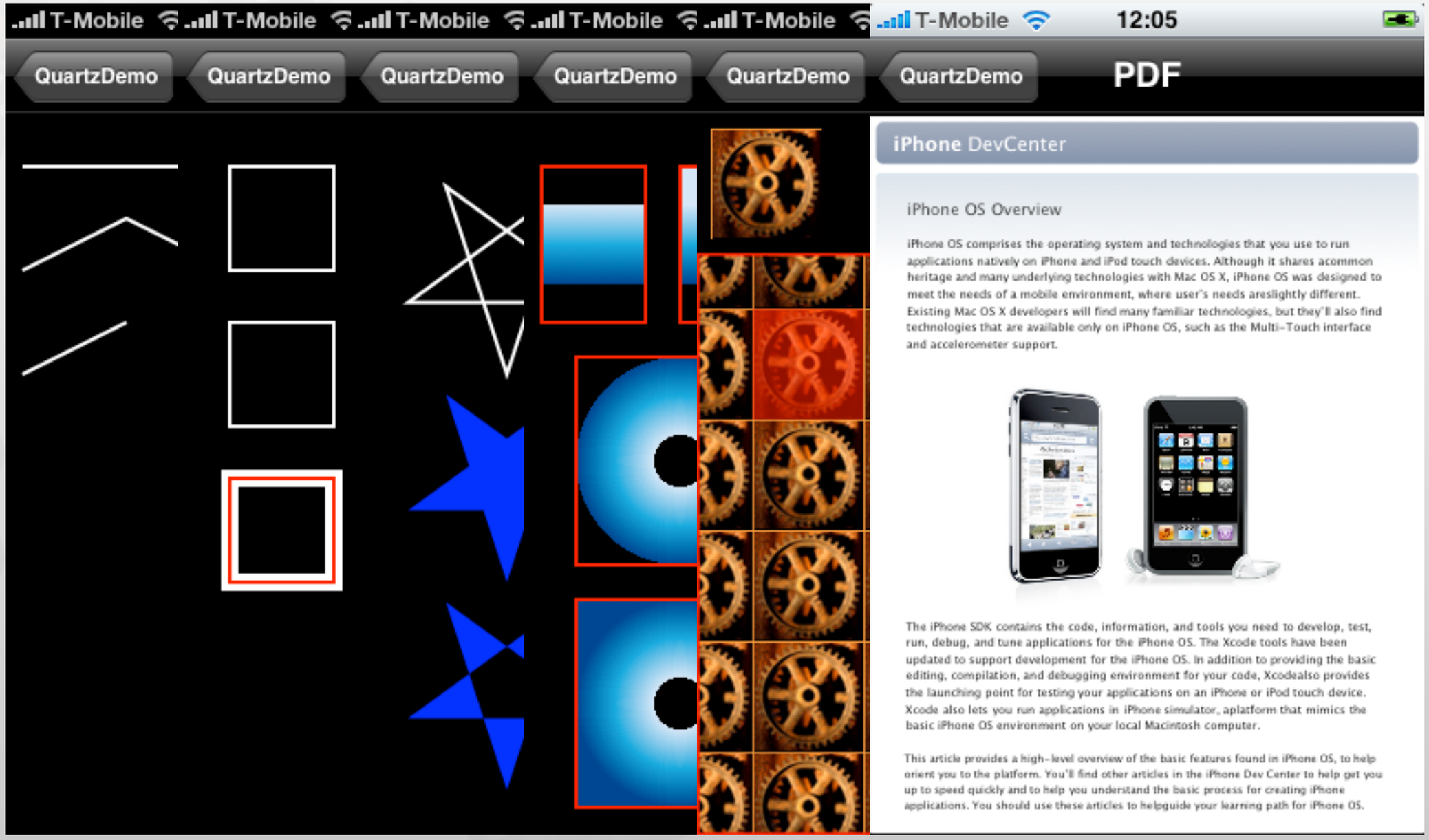
CoreGraphics Primitives

- Graphics context
- Paths
- Transformations
- Colors & Fonts
- Images & PDF

The Graphics Context

- Opaque data type (CGContextRef)
- Window, view, bitmap, PDF document
- Encapsulates drawing
  - Color
  - Line width
  - ...

CoreGraphics Examples

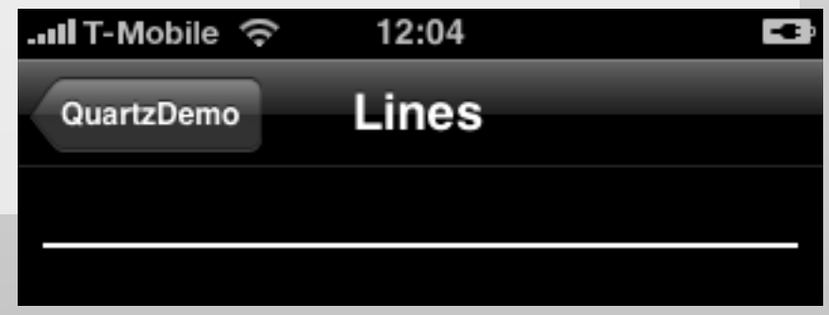


QuartzDemo Sample Code

Painters Drawing Model

Simple Drawing Example

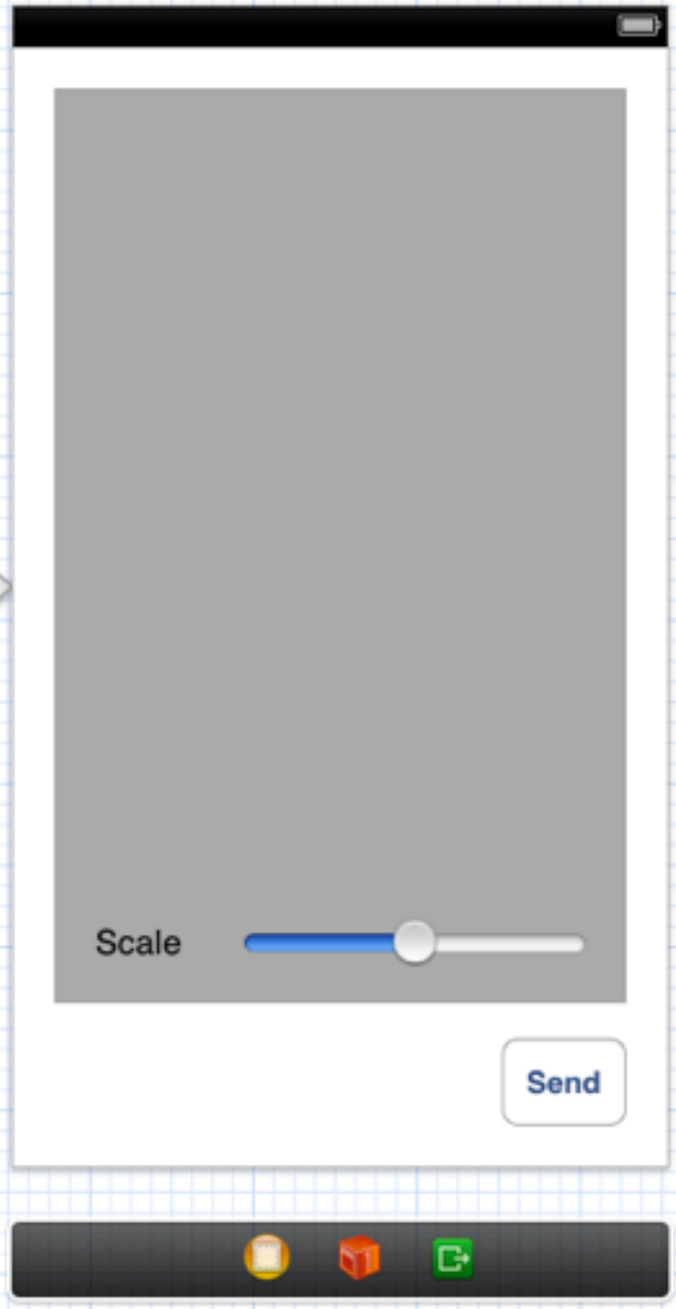
```
-(void)drawRect:(CGRect)rect
{
  //Get the current drawing context
    CGContextRef context = UIGraphicsGetCurrentContext();

    // Drawing lines with a white stroke color
    [[UIColor whiteColor] set];

    // Alternatively: Drawing lines with a white stroke color
    CGContextSetRGBStrokeColor(context, 1.0, 1.0, 1.0, 1.0);

    // Draw a single line from left to right
    CGContextMoveToPoint(context, 10.0, 30.0);
    CGContextAddLineToPoint(context, 310.0, 30.0);
    CGContextStrokePath(context);

}
```



The View Drawing Cycle

- When does drawRect: get called?
  - A part of the view was revealed
  - Unhiding a view
  - The view was scrolled off the screen and back on
  - setNeedsDisplay was called
- Parameter defines the area to be redrawn
  - Full view at first call
  - Can be smaller in subsequent calls

DrawRect Cascade



Backing View

Custom UIView Subclass

UILabel

UISlider

UIButton

Window has dirty regions

Prepares context

Calls drawRect: in all subviews

Managing Multiple Graphics Contexts

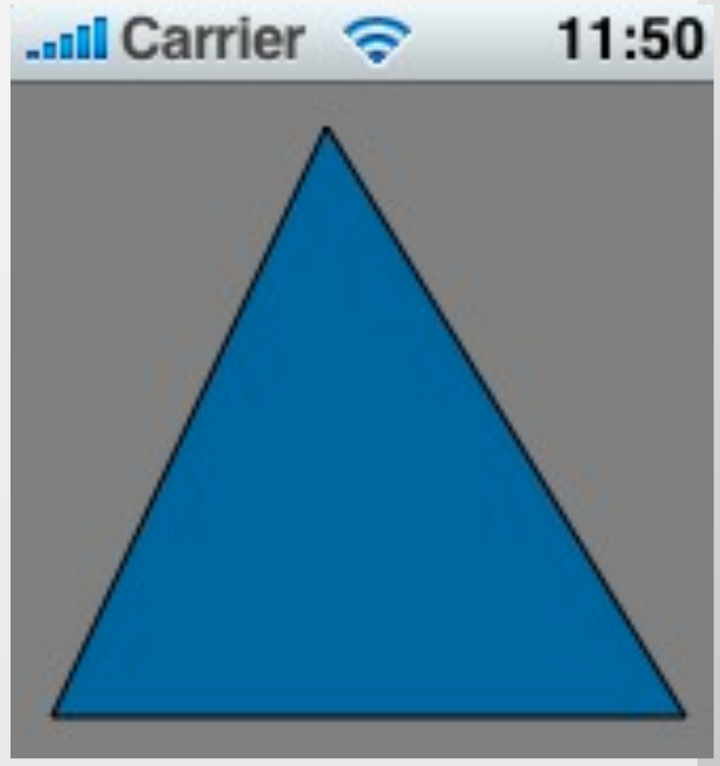
- UIGraphicsPushContext(CGContextRef context)
  - Save current context
  - Make specified context current
  - Balance calls with UIGraphicsPopContext()
- UIGraphicsPopContext()
  - Remove topmost context from stack
  - Restore the previous context

CGPath

- Construct a reusable path
- Draw multiple times
- Building blocks:
  - Points
  - Lines
  - Arcs
  - Curves
  - Ellipses
  - Rectangles

CGPath Example

```
- (void) drawRect:(CGRect)rect
{
   //Get the current drawing context
   CGContextRef context = UIGraphicsGetCurrentContext();

   // Create a triangle
   CGContextBeginPath(context);
   CGMutablePathRef trianglePath = CGPathCreateMutable();
   CGPathMoveToPoint(trianglePath, NULL, 75, 10);
   CGPathAddLineToPoint(trianglePath, NULL, 75, 10);
   CGPathAddLineToPoint(trianglePath, NULL, 160, 150);

   // Draw in blue with black stroke color
   CGColorRef blueColor = [[UIColor colorWithRed:0.0
                                           green:0.37
                                            blue:0.65
                                           alpha:0.8] CGColor];
   CGContextSetFillColorWithColor(context, blueColor);
   [[UIColor blackColor] setStroke];

   // Draw the path
   CGContextAddPath(context, trianglePath);
   CGContextDrawPath(context, kCGPathFillStroke);
}
```



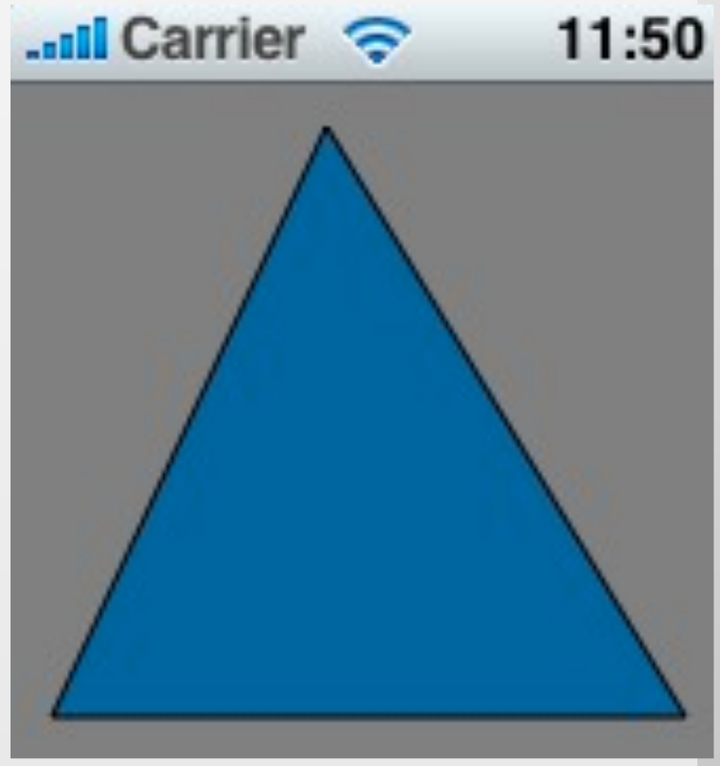
Remember: Memory Management

	Java	C	Core Foundation	Cocoa / UIKit
Garbage collection	✓			
Malloc/free		✓	✓	
Retain/Release			✓	✓
ARC				✓

CoreGraphics does not support ARC!

CGPath Example

```
- (void) drawRect:(CGRect)rect
{
   //Get the current drawing context
   CGContextRef context = UIGraphicsGetCurrentContext();

   // Create a triangle
   CGContextBeginPath(context);
   CGMutablePathRef trianglePath = CGPathCreateMutable();
   CGPathMoveToPoint(trianglePath, NULL, 75, 10);
   CGPathAddLineToPoint(trianglePath, NULL, 75, 10);
   CGPathAddLineToPoint(trianglePath, NULL, 160, 150);

   // Draw in blue with black stroke color
   CGColorRef blueColor = [[UIColor colorWithRed:0.0
                                           green:0.37
                                            blue:0.65
                                           alpha:0.8] CGColor];
   CGContextSetFillColorWithColor(context, blueColor);
   [[UIColor blackColor] setStroke];

   // Draw the path
   CGContextAddPath(context, trianglePath);
   CGContextDrawPath(context, kCGPathFillStroke);
}
```



Find the memory leak

```
- (void) drawRect:(CGRect)rect
{
   //Get the current drawing context
   CGContextRef context = UIGraphicsGetCurrentContext();

   // Create a triangle
   CGContextBeginPath(context);
   CGMutablePathRef trianglePath = CGPathCreateMutable();
   CGPathMoveToPoint(trianglePath, NULL, 75, 10);
   CGPathAddLineToPoint(trianglePath, NULL, 75, 10);
   CGPathAddLineToPoint(trianglePath, NULL, 160, 150);

   // Draw in blue with black stroke color
   CGColorRef blueColor = [[UIColor colorWithRed:0.0
                                           green:0.37
                                            blue:0.65
                                           alpha:0.8] CGColor];
   CGContextSetFillColorWithColor(context, blueColor);
   [[UIColor blackColor] setStroke];

   // Draw the path
   CGContextAddPath(context, trianglePath);
   CGContextDrawPath(context, kCGPathFillStroke);
   CGPathRelease(trianglePath);
}
```

"Create" warrants a "release"

Transforms & Shadows

- CGContextTranslateCTM(...)
- CGContextRotateCTM(...)
- CGContextScaleCTM(...)
- CGContextSetShadow(...)

Drawing to Bitmaps or PDFs

- Create a new Graphics Context
  - UIGraphicsBeginImageContext(...)
  - CGBitmapContextCreate(...)
  - CGPDFContextCreate(...)
- Different coordinate system

And a Lot More

- Several blending modes available
- Clipping along paths
- Patterns
- Gradients
- Transparency layers

Drawing

Quartz & CoreGraphics	Core Animation	Spirte Kit	OpenGL
Mostly Vector Drawing	Mostly Bitmap Drawing	Spirte Game Engine	Mostly Polygon Drawing
2D	2.5D	2.5D	
Tell how do draw	Tell what to draw and how to animate it	Create scene graph, and physics; apply actions	Tell how do draw

CoreAnimation

Core Animation

- Collection of Objective-C classes for animation
- High level of abstraction
  - Dynamic (animatable) attributes
  - CAAnimation class

List of Animatable Properties

- Geometric: frame, bounds, position, transform...
- Background: backgroundColor, backgroundFilters
- Border: borderColor, borderWidth
- Content: contents, contentsGravity
- Sublayers: sublayers, sublayerTransform...
- Filters, Shadow, Composing, Masks

CALayer

- UIView equivalent for animation
  - All animation is performed in CALayers
- All UIViews are backed up by CALayers
  - (only Cocoa Touch, on demand for Cocoa)
  - Layer hierarchy in parallel to view hierarchy
  - view.layer
- You can create and animate your own layers
  - No need for a view

Custom CALayers

- Do not subclass CALayer
  - special classes exist for video, text, ...
- Assign content or a delegate
- Content variable or delegate is queried for drawing
  - drawLayer:inContext:

Example: Custom CALayer

```
 // in any UIView
- (void) awakeFromNib;
{
  // create the box layer
  boxLayer = [[CALayer alloc] init];
  // give it a size and location
  boxLayer.bounds = CGRectMake(0.0, 0.0, 85.0, 85.0);
  boxLayer.position = CGPointMake(160.0, 100.0);
  // set the delegate
  boxLayerDelegate = [[BoxLayerDelegate alloc] init];
  boxLayer.delegate = boxLayerDelegate;
  [boxLayer setNeedsDisplay];
  // make it a sublayer to the view's layer
  [self.layer addSublayer:boxLayer];
}

// ----------
@implementation BoxLayerDelegate

- (void) drawLayer:(CALayer *)layer inContext:(CGContextRef)context
{
  CGContextSetRGBFillColor(context, 1.0, 0.0, 0.0, 1.0);
  CGContextFillRect(context, layer.bounds);
}
@end
```

Implicit Animations

- Layers offer many animatable properties
- Changing their value creates an implicit animation
  - The presented value is changed over time (0.25s)
- Every layer has a presentation and a model layer
  - Presentation Layer: currently displayed values
  - Model Layer: target values

Demo

Example

```
- (void)showAdvancedOptions {
  // assume polygonView and optionsView
  [UIView beginAnimations:@"advancedAnimations”
context:nil];
  [UIView setAnimationDuration:0.3];

  // make optionsView visible (alpha is currently 0.0)
  optionsView.alpha = 1.0;

  // move the polygonView down
  CGRect polygonFrame = polygonView.frame;
  polygonFrame.origin.y += 200;
  polygonView.frame = polygonFrame;

  [UIView commitAnimations];
}
```

Explicit Animation

- Create animation object
  - CABasicAnimation
  - CAKeyframeAnimation
- Configure animation
  - Duration
  - Timing function
- Configure animation target
  - Key path of animated property
  - fromValue: and toValue:

Example: Move Animation

```
- (void)startMoveAnimation;
{
  CGPoint orgPoint = timeLabel.layer.position;
  CGPoint targetPoint = CGPointMake(orgPoint.x, orgPoint.y +
100.0);

  CABasicAnimation *move = [[CABasicAnimation alloc] init];
  move.keyPath = @"position";
  move.fromValue = [NSValue valueWithCGPoint:orgPoint];
  move.toValue = [NSValue valueWithCGPoint:targetPoint];
  move.duration = 0.5;

  timeLabel.layer.position = targetPoint;

  // animate
  [timeLabel.layer addAnimation:move forKey:@"moveAnimation"];

}
```

Example: Spin Animation

```
- (void)startSpinAnimation;
{
  // create the spin animation
  CABasicAnimation *spin = [[CABasicAnimation alloc] init];
  spin.keyPath = @"transform.rotation";
  spin.toValue = [NSNumber numberWithFloat:M_PI * 4.0];
  spin.duration = 1.0;

  // set ease-in, ease-out as timing function
  spin.timingFunction = [CAMediaTimingFunction
    functionWithName:kCAMediaTimingFunctionEaseInEaseOut];

  // set the delegate
  spin.delegate = self;

  // set the spin animation
  [timeLabel.layer addAnimation:spin forKey:@"spinAnimation"];
  [spin release];
}
```

Example: Bounce Animation

```
- (void)startBounceAnimation;
{
   CAKeyframeAnimation *bounce = [[CAKeyframeAnimation alloc] init];
   bounce.keyPath = @"transform";

   // create the values it will pass through
   CATransform3D forward  = CATransform3DMakeScale(1.3, 1.3, 1.0);
   CATransform3D back     = CATransform3DMakeScale(0.7, 0.7, 1.0);
   CATransform3D forward2 = CATransform3DMakeScale(1.2, 1.2, 1.0);
   CATransform3D back2    = CATransform3DMakeScale(0.9, 0.9, 1.0);
   bounce.values = [NSArray arrayWithObjects:
                    [NSValue valueWithCATransform3D:CATransform3DIdentity],
                    [NSValue valueWithCATransform3D:forward],
                    [NSValue valueWithCATransform3D:back],
                    [NSValue valueWithCATransform3D:forward2],
                    [NSValue valueWithCATransform3D:back2],
                    [NSValue valueWithCATransform3D:CATransform3DIdentity],nil];

   // start animation
   [timeLabel.layer addAnimation:bounce forKey:@"bounceAnimation"];
}
```

Combining Animations

- Multiple animations can be added to a layer
  - But: only one per key
- Animations will be played in parallel

Working with Animations

- Animations have a delegate
  - Informed when animation started / stopped
- Animations can be aborted
  - Add new animation to same layer for same key
- Animations can be grouped

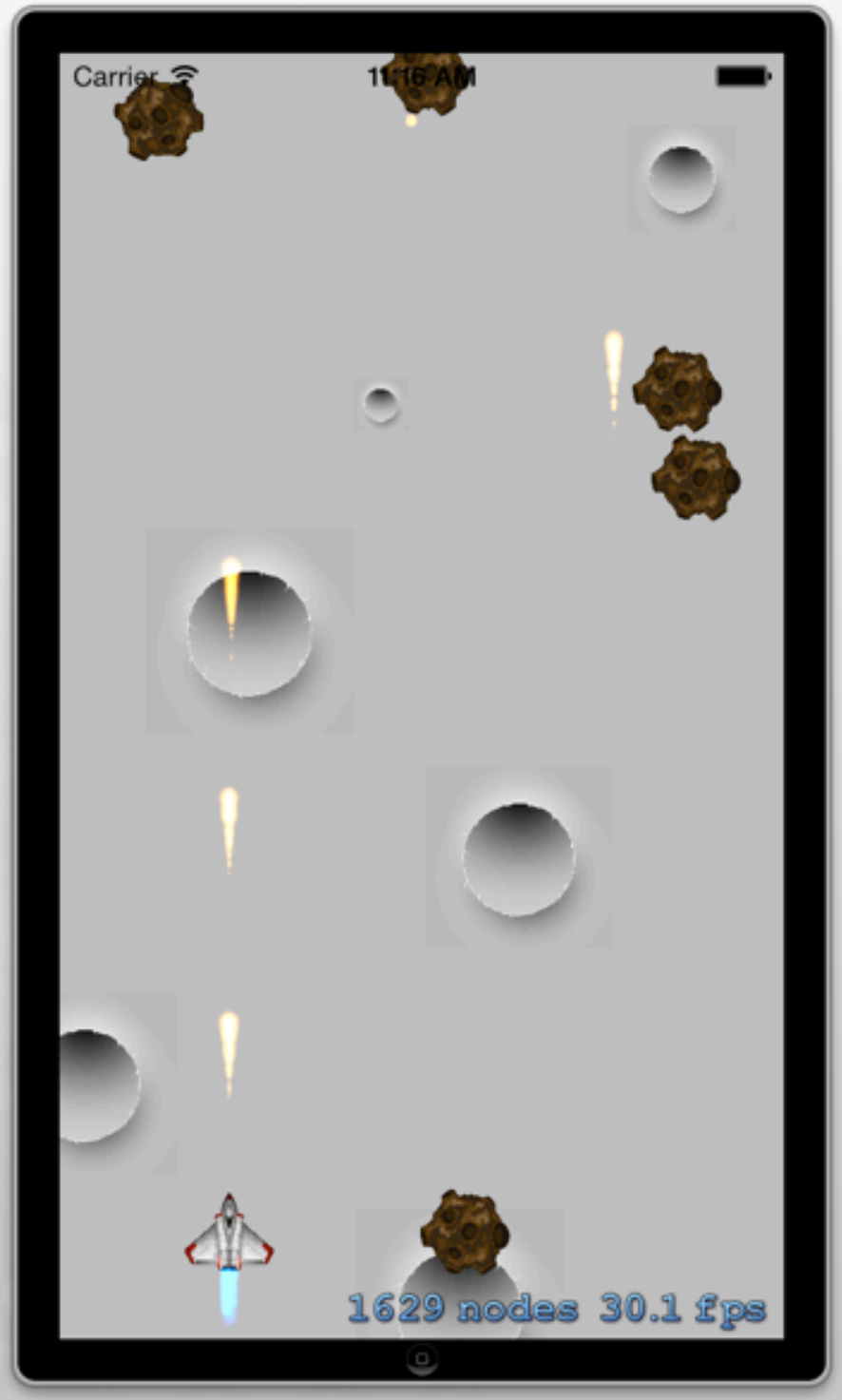
Demo

Drawing

Quartz & CoreGraphics	**Core Animation**	**Sprite Kit**	**OpenGL**
Mostly Vector Drawing	Mostly Bitmap Drawing	Sprite Game Engine	Mostly Polygon Drawing
2D	2.5D	2.5D	
Tell how do draw	Tell what to draw and how to animate it	Create scene graph, and physics; apply actions	Tell how do draw

Sprite Kit

Sprite Games



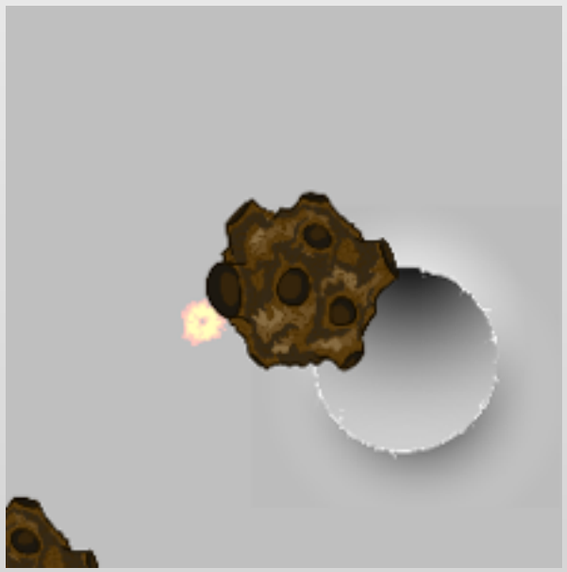
Other Sprite Game Engines

Demo

Basic Part of a Sprite Kit

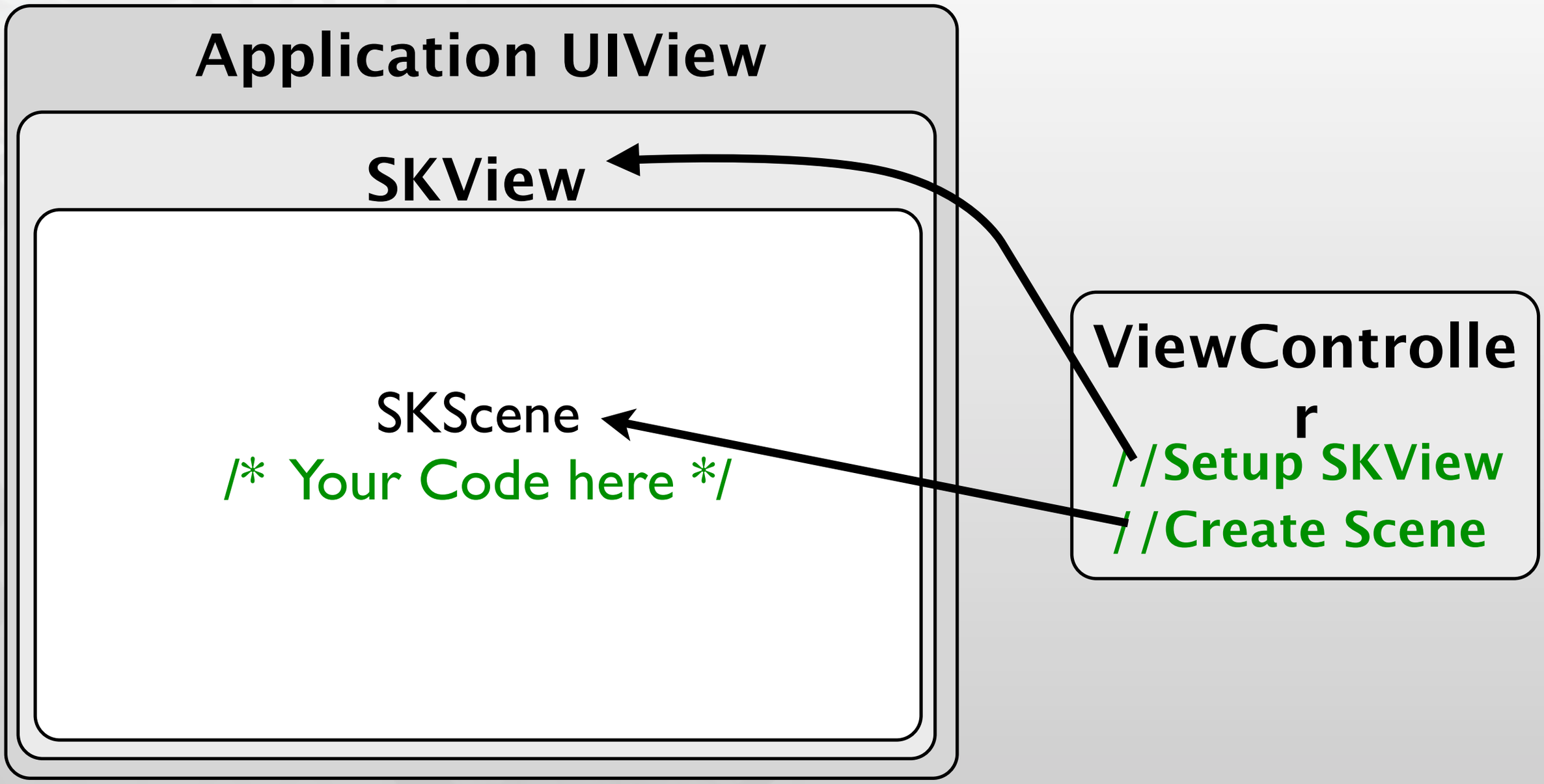
Objects

Actions

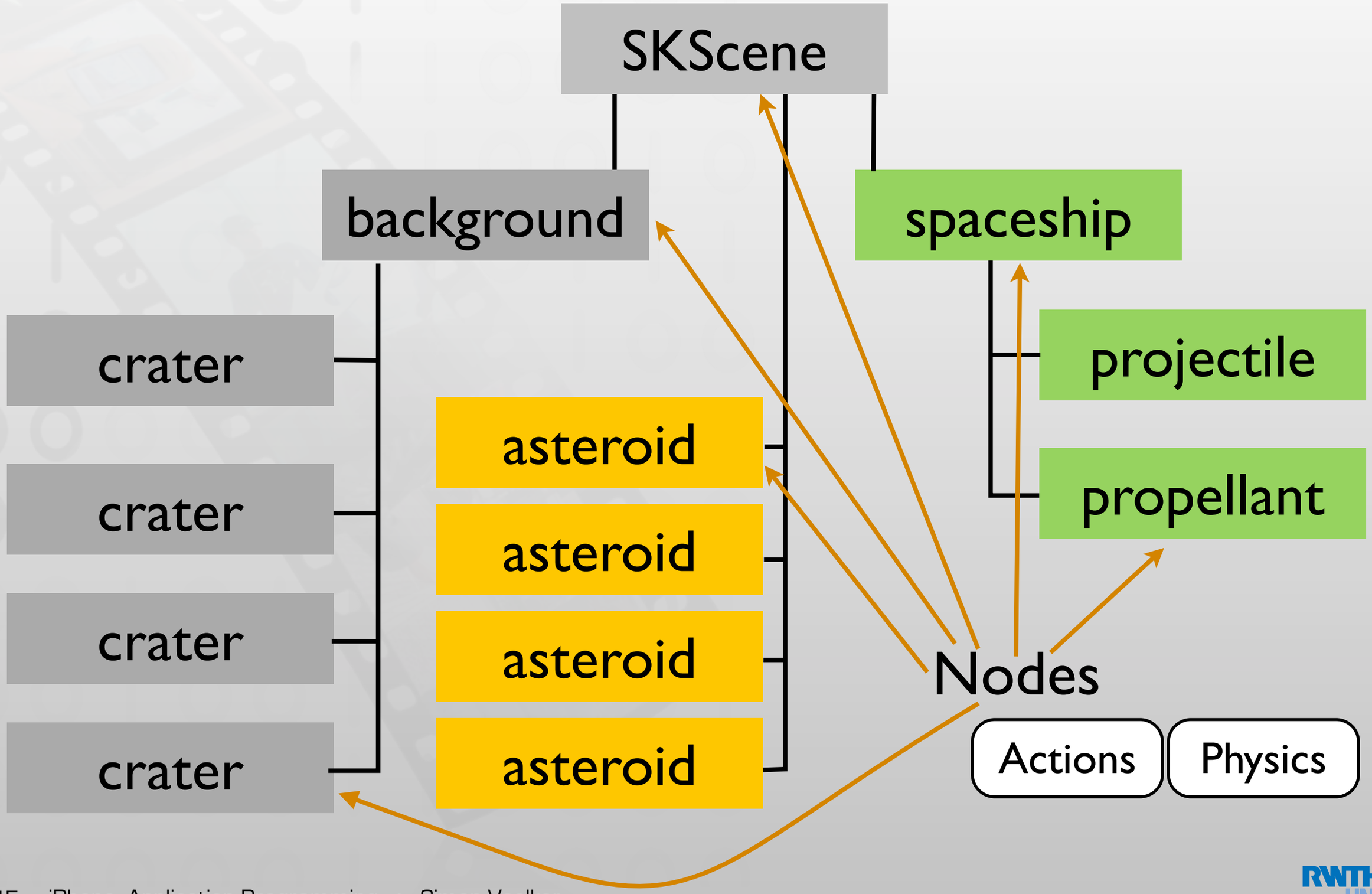


Physics

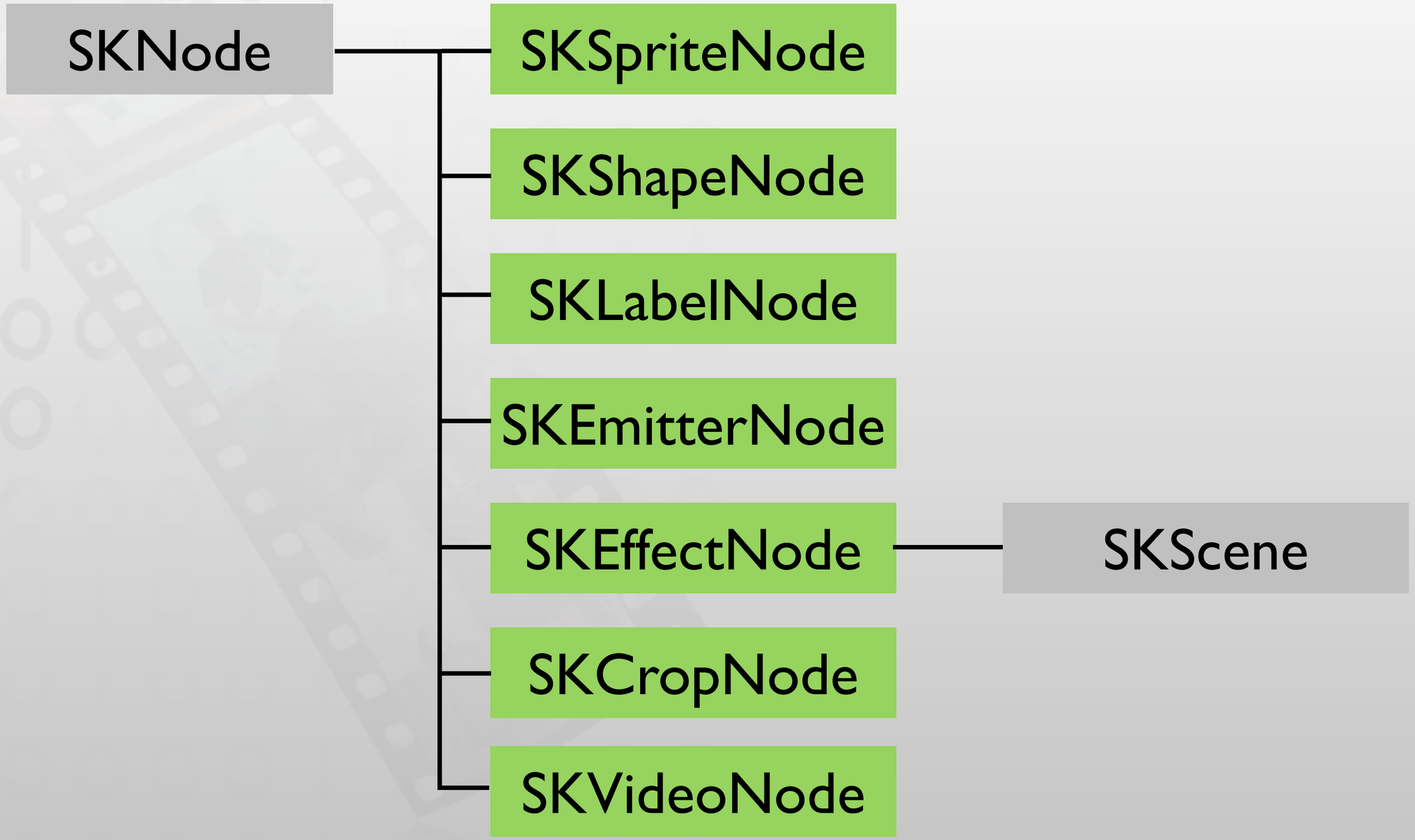
Root Object: SKScene



Scene Graph

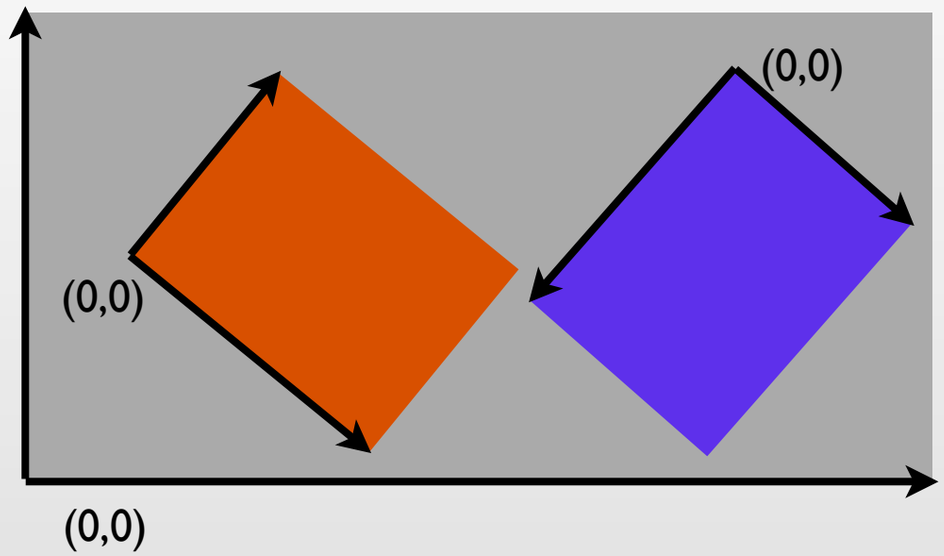


Sprite Kit Nodes



SKNode

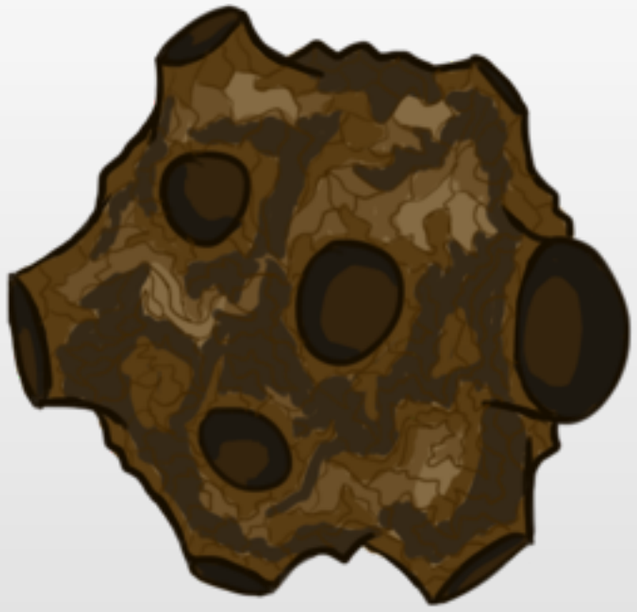
- Basic node (used for grouping)
- Position, rotation, scale
- zPosition




```
//Hit Test
[node containsPoint:aCGPoint];

//Converts a point from the coordinate system
[node convertPoint:aCGPoint fromNode:aSKNode];

//Converts a point in this node's coordinate system
[node convertPoint:aCGPoint toNode:aSKNode];
```

SKSpriteNode

solid color

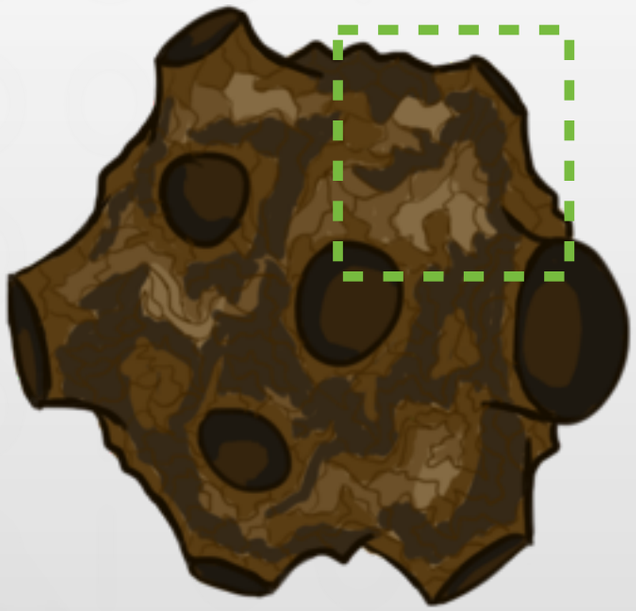


texture

color blending

```
 SKSpriteNode *green =
[SKSpriteNode spriteNodeWithColor:
[SKColor greenColor] size:CGPointMake(200, 200)];

SKSpriteNode *asteroid =
[SKSpriteNode spriteNodeWithImageNamed:@“asteroid.png"];

asteroid.color = [SKColor greenColor];
asteroid.colorBlendFactor = 0.5;
```

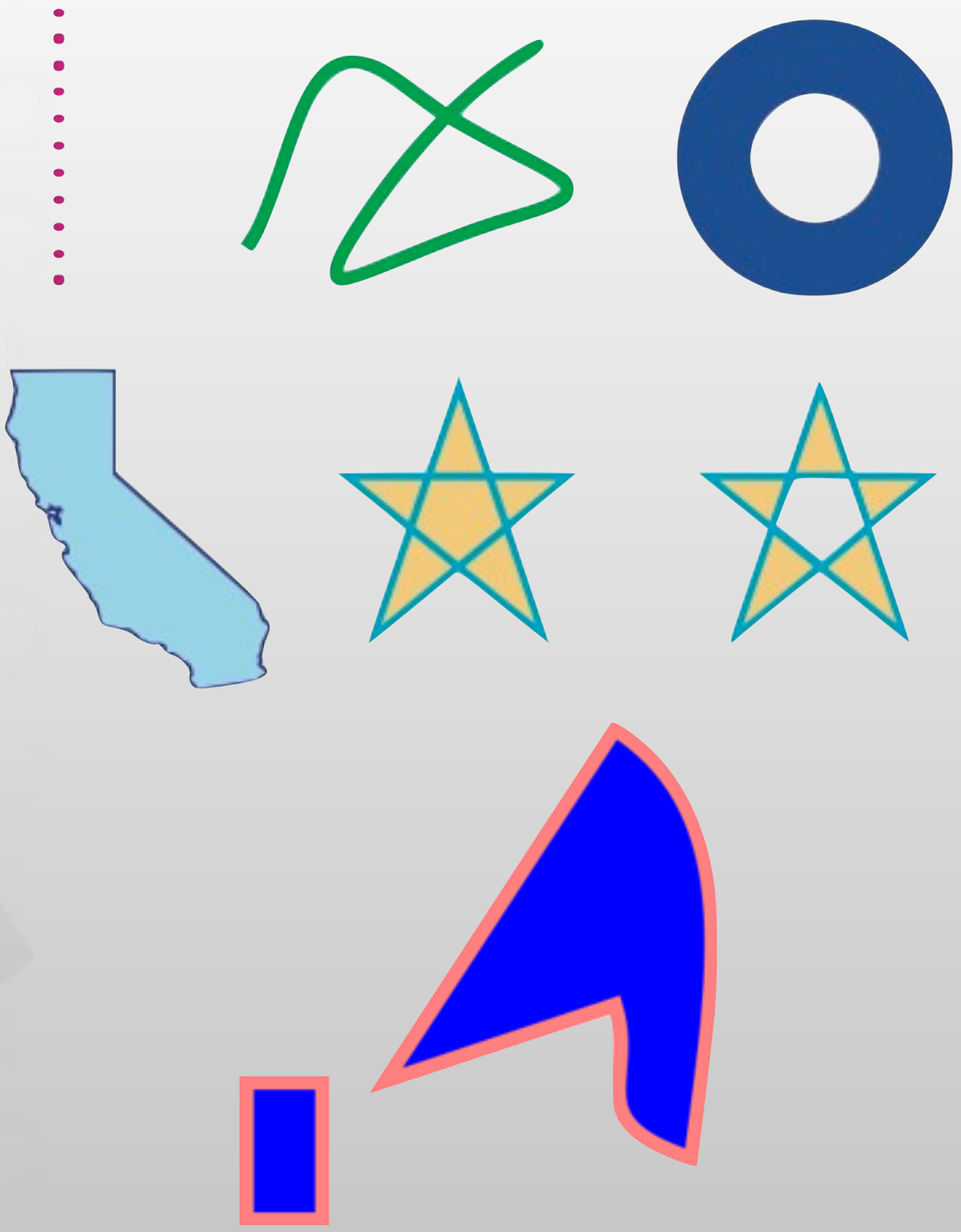
SKTexture



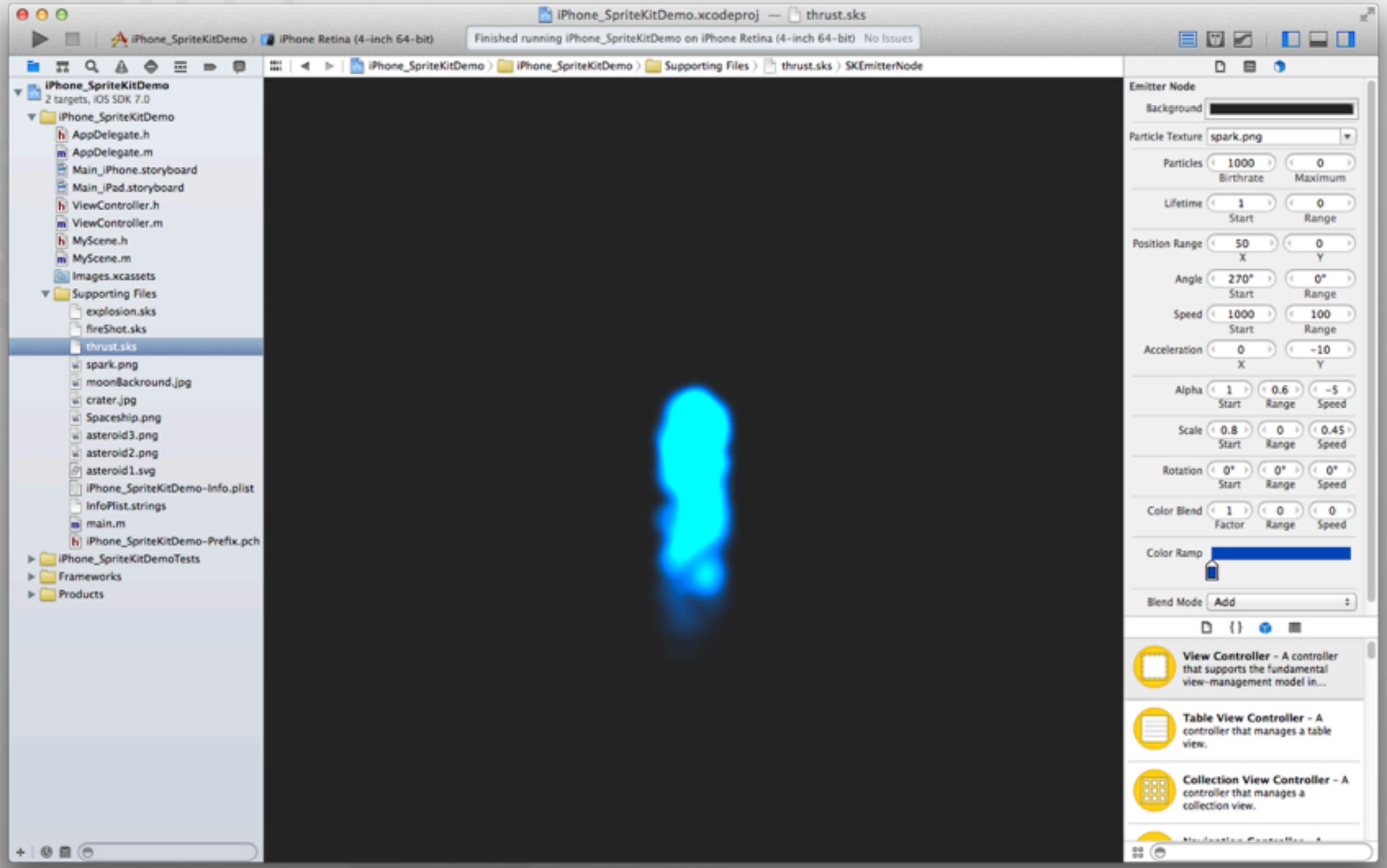
```
[SKTexture textureWithImageNamed:@"asteroid"];

[SKTexture textureWithRect: CGRectMake(100, 100, 80, 80)
 inTexture:tex1];
```

SKShapeNode

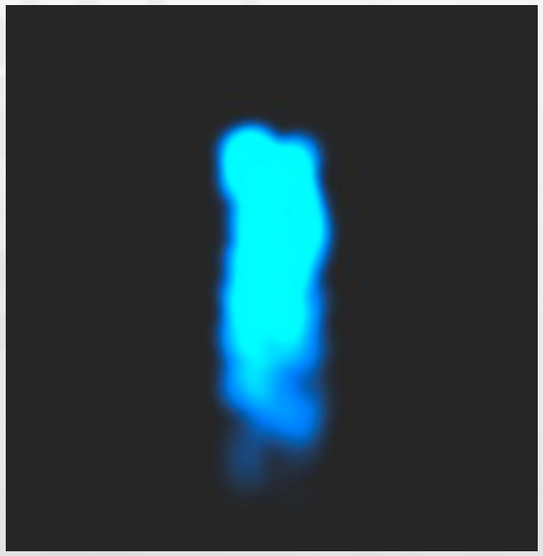
- Draws CGPath
- Stroke, Fill Color
- Glow effect



Particle Editor



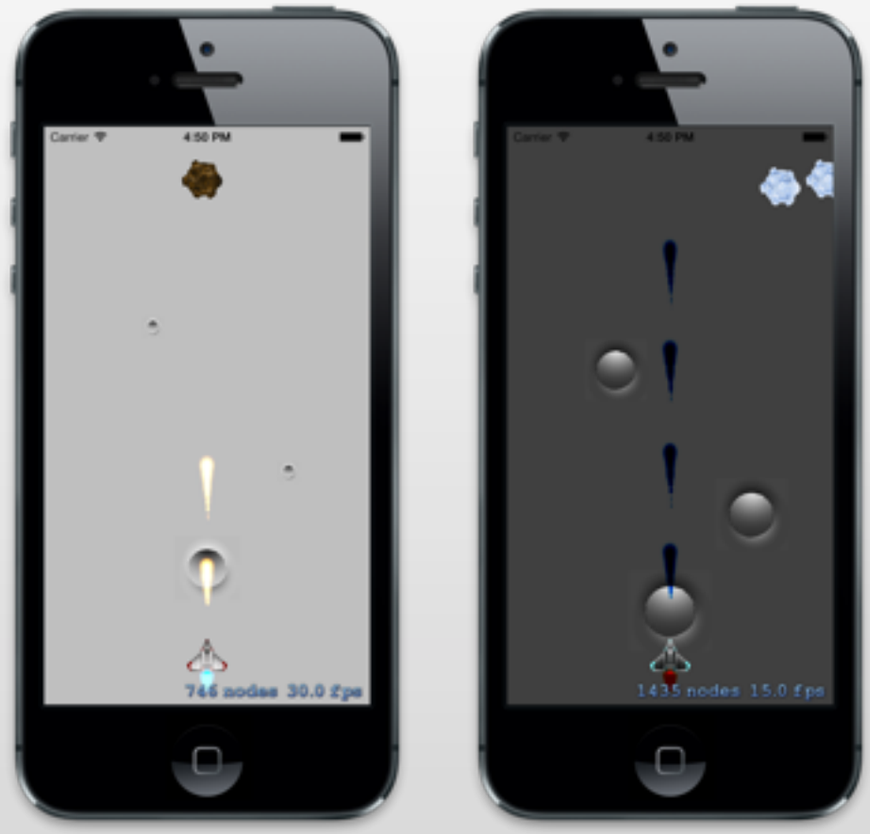
SKEmitterNode



```
NSString *path = [[ NSBundle mainBundle]
                    pathForResource:@"thrust" ofType: @"sks"];
 SKEmitterNode *thrust = [NSKeyedUnarchiver
                     unarchiveObjectWithFile:path];

thrust.position = CGPointMake(0, self.spaceship.size.height -10);

[self.spaceship addChild: thrust];

thrust.particleScale = 2;
thrust.particleScaleSpeed = -10;
```

SKEffectNode

- Applies CIFilter to its children
- CIFilter is a powerful Core Image filter
- Can be used on the entire Scene




```
CIFilter* filter = [CIFilter filterWithName:@"CIColorInvert"];
[filter setDefaults];

self.filter = filter;
self.shouldEnableEffects = YES;
```

SKEffectNode: CIFilter

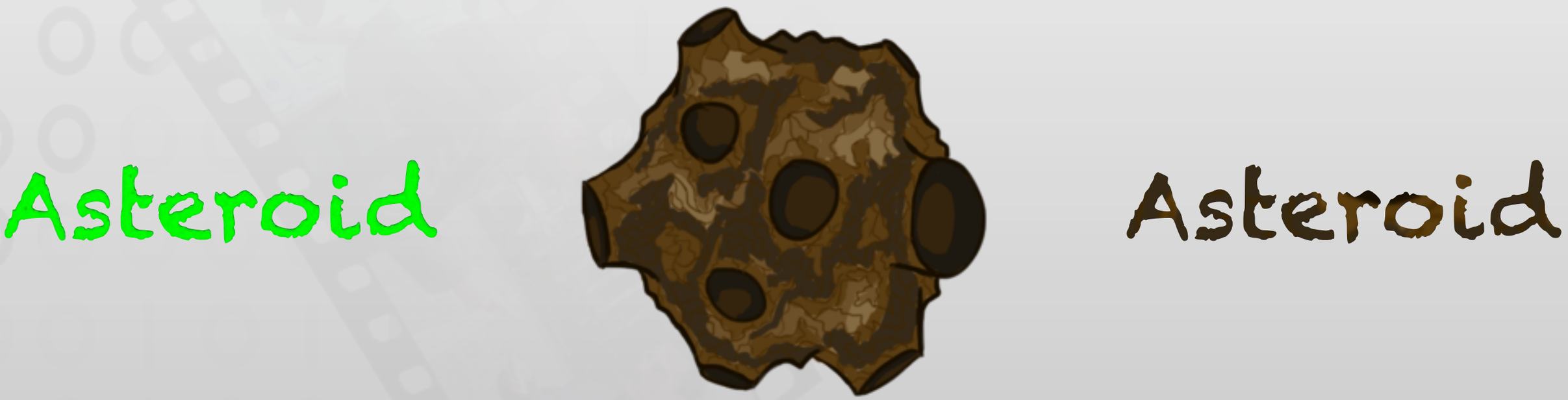
- More then 100 different Filter
- Glow effects:
  - CIBloom

CIBloom

```
CIFilter* filter = [CIFilter filterWithName:@“CIBloom"];
self.filter = filter;
self.shouldEnableEffects = YES;

[filter setValue:
        [NSNumber numberWithFloat:20.0 forKey:@"inputCenter"];
[filter setValue:
        [NSNumber numberWithFloat:2.0 forKey:@"inputIntensity"];
```

SKCropNode

- Creates a mask the children
- Mask is defined as a SKNode



Mask child result

SKVideoNode

- Video as Node
- AVPPlayer (AVFoundation.framework)
- All the functionality from AVFoundation


```
[SKVideoNode videoNodeWithVideoFileNamed:@"video.mp4"];

[SKVideoNode videoNodeWithAVPlayer:player];
```

Sprite Kit: Actions

Simple Actions

Create Action

```
[SKAction moveTo:CGPointMake(100,100) duration:1.0];

[SKAction rotateByAngle:M_PI duration:1.0];

[SKAction fadeAlphaTo:0.75 duration:1.0];

[SKAction scaleBy:10.0 duration:1.0];
```

Move the spaceship

```
SKAction *move = [SKAction moveBy:aVector duration:0.0]
[spaceShip runAction:move];

[spaceShip runAction:[SKAction moveBy:aVector duration:0.0]];
```

Repeating Actions

```
SKAction *move = [SKAction moveBy:aVector duration:0.0];

SKAction *repeat = [SKAction repeatAction:move count:3];

SKAction *repeatForever = [SKAction repeatActionForever:move];
```

Combining Actions


```
[node runAction:[SKAction sequence:@[action1, action2, action3]]];
```


```
[node runAction:[SKAction group:@[action1, action2, action3]]];
```

Other Actions

Texture animate

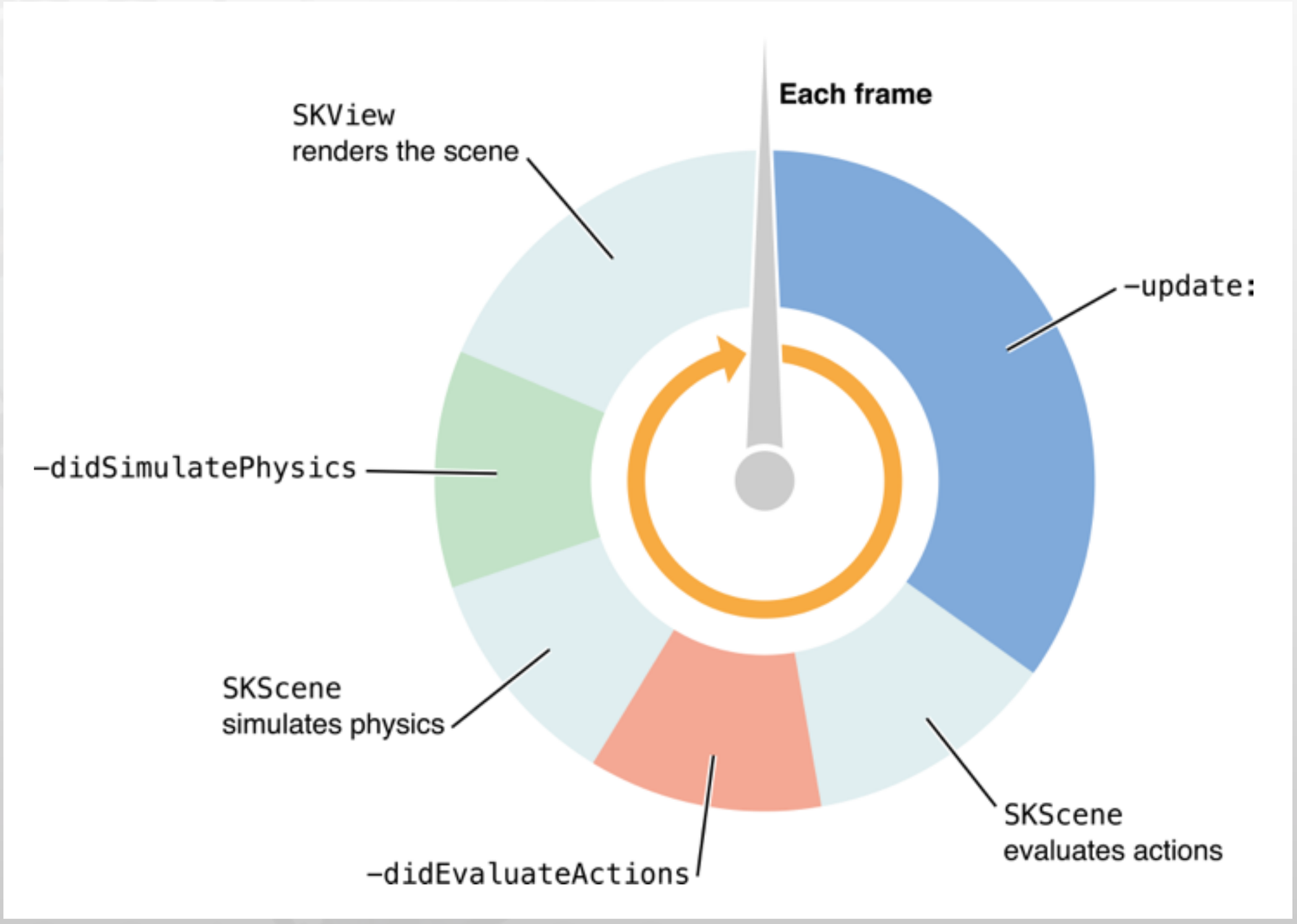
```
[SKAction animateWithTextures:@[tex0, tex1] timePerFrame:0.1];
```

Path animate

```
[SKAction followPath:aPath duration:2.5];
```

and many more: colors, sounds, custom blocks ...

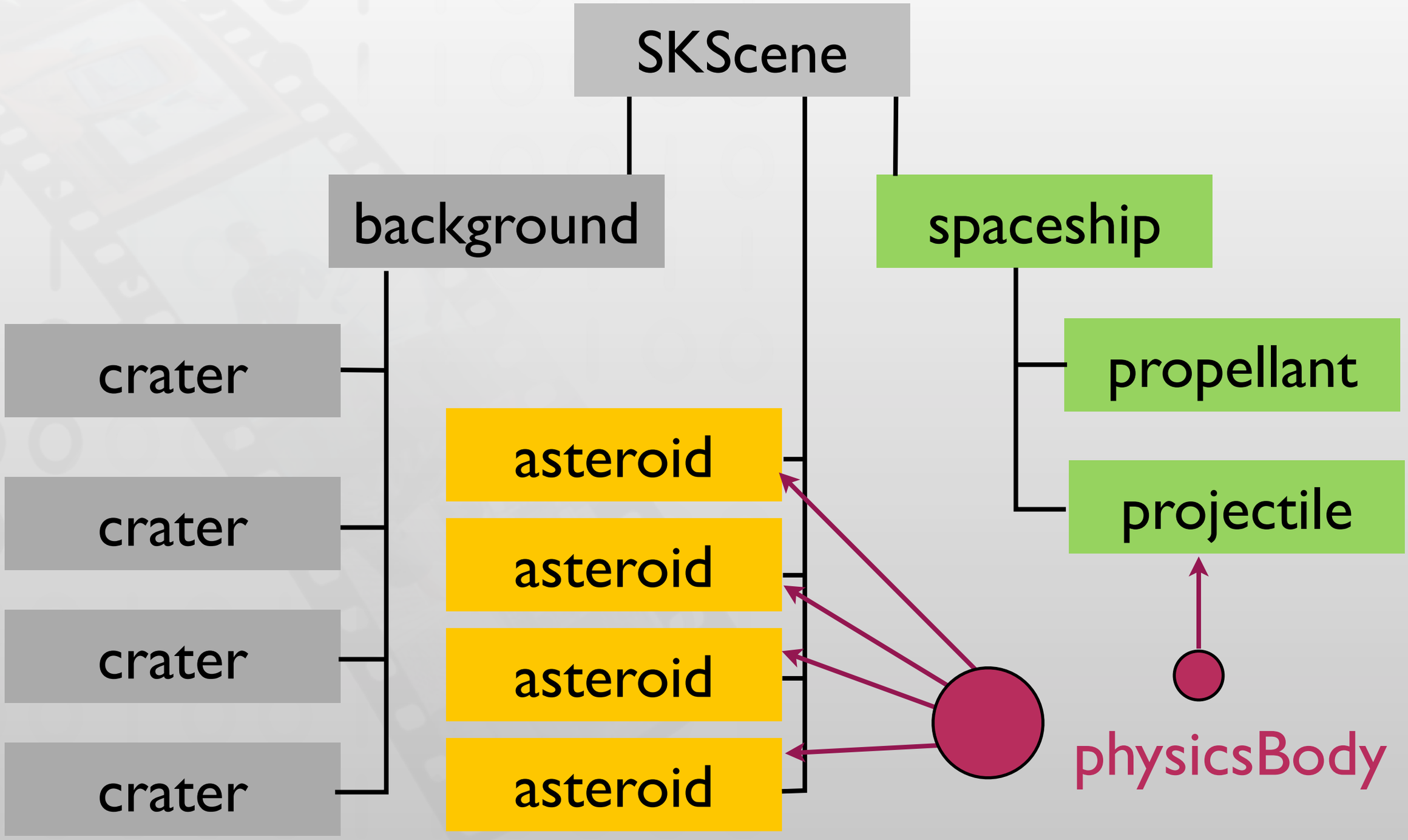
Sprite Kit Render Loop



[Apple iOS 7 API]

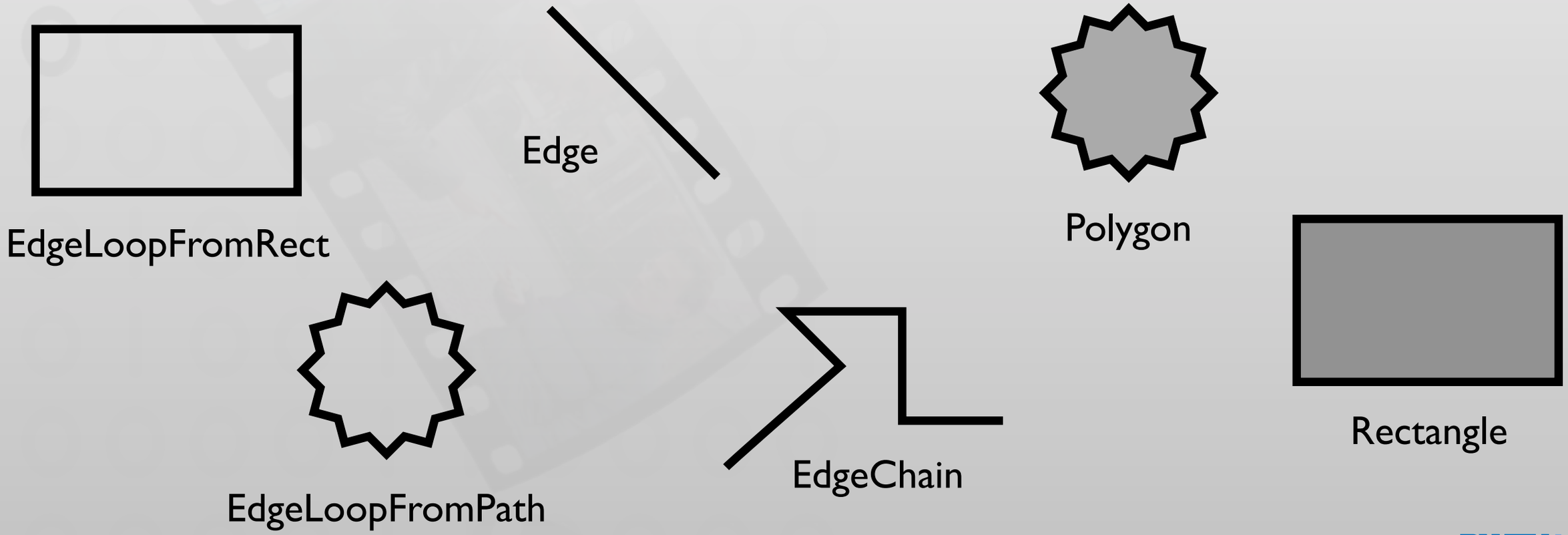
Physics

Scene Graph



SKPhysicsBody

```
asteroid.physicsBody =
[SKPhysicsBody bodyWithCircleOfRadius: asteroid.size.width / 2 ];

asteroid.physicsBody.mass = 10;
asteroid.physicsBody.linearDamping = 0;

asteroid.physicsBody.velocity = aCGVector;
```



SKPhysicsWorld

- Each scene as its own PhysicsWorld
- Performs contact and collision tests

Global gravity

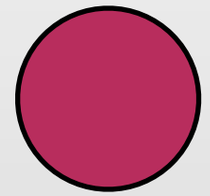
```
/* normal gravity */
self.physicsWorld.gravity = CGPointMake(0.0, -9.8);

/* inverted gravity */
self.physicsWorld.gravity = CGPointMake(0.0, +9.8);
```

SKPhysicsContact

Contact Delegate

```
self.physicsWorld.contactDelegate = myContactDelegate;
```

projectile



asteroid

```
-(void)didBeginContact:(SKPhysicsContact *)contact
```

```
@interface SKPhysicsContact
SKPhysicsBody *bodyA;
SKPhysicsBody *bodyB;

CGPoint contactPoint;

CGFloat collisionImpulse;
@end
```

Collision Groups

```
@property (assign) uint32_t categoryBitMask;

@property (assign) uint32_t collisionBitMask;

@property (assign) uint32_t contactTestBitMask;
```

```
static const uint32_t noneCategory          =  0;
static const uint32_t asteroidCategory      =  0x1 << 0;
static const uint32_t shotCatergory         =  0x1 << 1;
static const uint32_t spaceshipCategory     =  0x1 << 2;
static const uint32_t allCategory           =  UINT32_MAX;
```

Physics Demo

Summary

- CoreGraphics
- Core Animation
- Sprite Kit

- Reading Assignment:
  - Core Animation Programming Guide
  - View Programming Guide for iOS
  - Sprite Kit Programming Guide

Looking for Thesis Students

- Using Sprite Kit on large interactive tabletop
- Tangible on interactive tabletops

voelker@cs.rwth-aachen.de

Image I/O

- Read and write image files
  - PNG, JPEG, TIFF, GIF
- Highly efficient
- Metadata access
- Color management

Creating an Image

```
CGImageRef MyCreateCGImageFromFile (NSString* path)
{
   // Get the URL for the pathname passed to the function.
   NSURL url = [NSURL fileURLWithPath:path];
   CGImageRef          myImage = NULL;
   CGImageSourceRef    myImageSource;
   CFDictionaryRef     myOptions = NULL;
   CFStringRef         myKeys[2];
   CFTypeRef           myValues[2];

   // Set up options
   // caching the image in a decoded form and for using floating-point
   // values if the image format supports them.
   myKeys[0] = kCGImageSourceShouldCache;
   myValues[0] = (CFTypeRef)kCFBooleanTrue;
   myKeys[1] = kCGImageSourceShouldAllowFloat;
   myValues[1] = (CFTypeRef)kCFBooleanTrue;
   // Create the dictionary
   myOptions = CFDictionaryCreate(NULL, (const void **) myKeys,(const void **) myValues, 2,
                  &kCFTypeDictionaryKeyCallBacks, & kCFTypeDictionaryValueCallBacks);
   // Create an image source from the URL.
   myImageSource = CGImageSourceCreateWithURL((CFURLRef)url, myOptions); CFRelease(myOptions);
   // Make sure the image source exists before continuing
   if (myImageSource == NULL){ fprintf(stderr, "Image source is NULL."); return NULL; }
   // Create an image from the first item in the image source.
   myImage = CGImageSourceCreateImageAtIndex(myImageSource, 0, NULL);
   CFRelease(myImageSource);
   // Make sure the image exists before continuing
   if (myImage == NULL){ fprintf(stderr, "Image not created from image source."); return NULL; }
   return myImage;
}
```

Accessing Properties

- Format-specific dictionaries
- Camera-maker dictionaries
- Image source container properties
- Individual image properties
- Color model values
- EXIF dictionary keys

Retrieving Properties

```
    // Create an image source
CGImageSourceRef source = CGImageSourceCreateWithURL((CFURLRef)url, NULL);


// Copy the properties
CFDictionaryRef fileProps = CGImageSourceCopyProperties(source, nil);

// Get the file size for example
NSString *fileSize = (id)CFDictionaryGetValue(fileProps,
kCGImagePropertyFileSize);
```

Writing Images to File

```
float compression = 1.0; // Lossless compression if available.
int orientation = 4; // Origin is at bottom, left.
CFStringRef myKeys[3];
CFTypeRef   myValues[3];
CFDictionaryRef myOptions = NULL;
myKeys[0] = kCGImagePropertyOrientation;
myValues[0] = CFNumberCreate(NULL, kCFNumberIntType, &orientation);
myKeys[1] = kCGImagePropertyHasAlpha;
myValues[1] = kCFBooleanTrue;
myKeys[2] = kCGImageDestinationLossyCompressionQuality;
myValues[2] = CFNumberCreate(NULL, kCFNumberFloatType, &compression);
myOptions = CFDictionaryCreate( NULL, (const void **)myKeys, (const void **)myValues,
   3, &kCFTypeDictionaryKeyCallBacks, &kCFTypeDictionaryValueCallBacks);
```

```
- (void)writeCGImage:(CGImageRef)image toURL:(NSURL*)url
                            withType:(CFStringRef)imageType andOptions:
(CFDictionaryRef)options
{
   CGImageDestinationRef myImageDest =
          CGImageDestinationCreateWithURL((CFURLRef)url, imageType, 1, nil);
   CGImageDestinationAddImage(myImageDest, image, options);
   CGImageDestinationFinalize(myImageDest);
   CFRelease(myImageDest);
}
```