

はじめてのSprite Kit

Twitter: [@studioshin](https://twitter.com/studioshin)

第52回 Cocoa勉強会関西

2013年10月5日

Sprite Kitとは？

iOS 7から搭載された
iOS/OS X向けの2Dゲーム開発の
ためのフレームワーク。

WWDC2013で発表！



Sprite Kitでできること

シーンによるゲーム画面管理

スプライトによるキャラクターの配置

アクションでキャラクターを動かす

物理シミュレーション

Sprite Kitの主なクラス

- **SKView**

スプライトを表示する土台となるビュークラス

- **SKScene**

スプライトの集合を管理しレンダリングを行うクラス

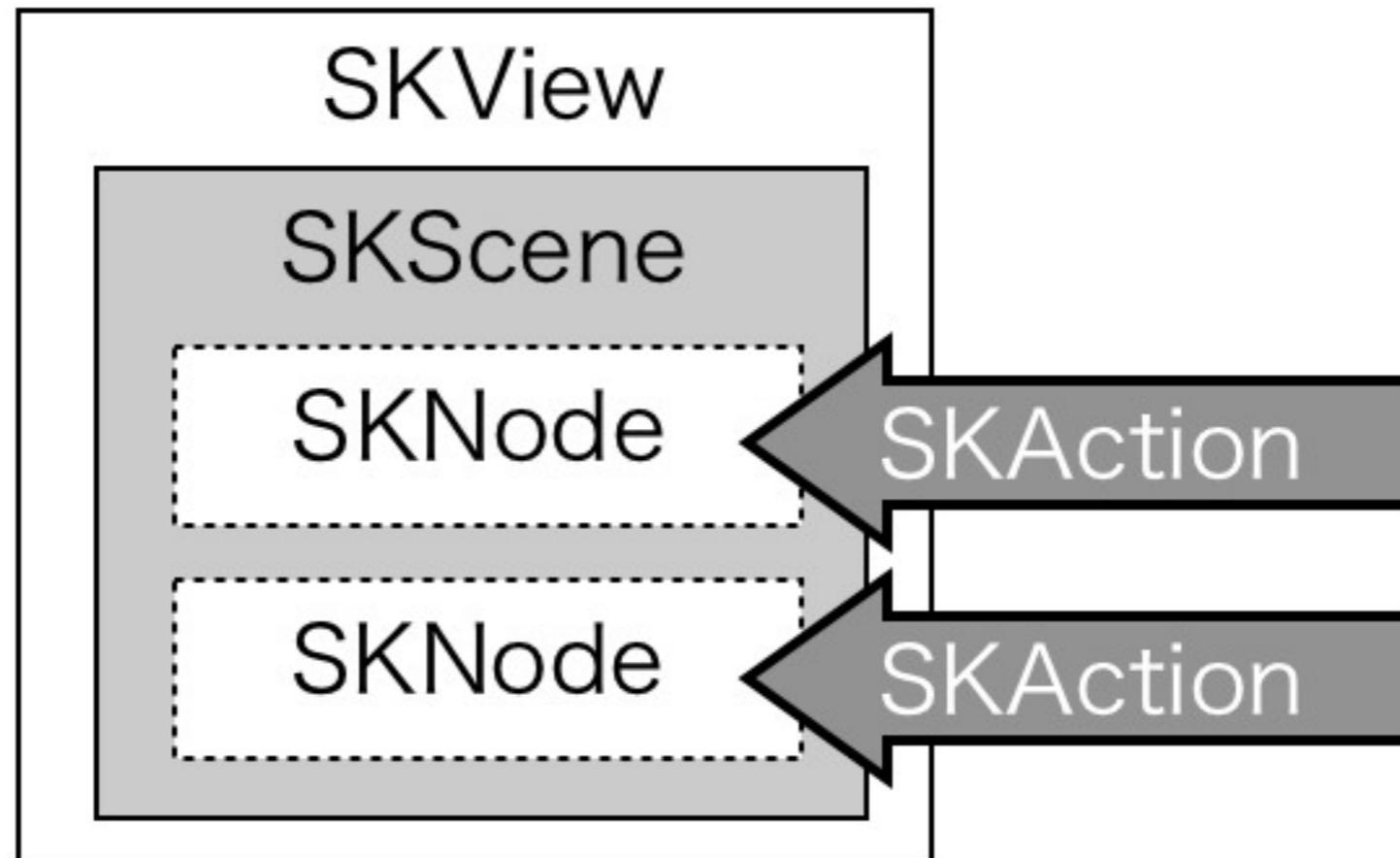
- **SKNode**

テキスト、画像、動画、パスやカラーを使ったスプライト表示を行うクラス

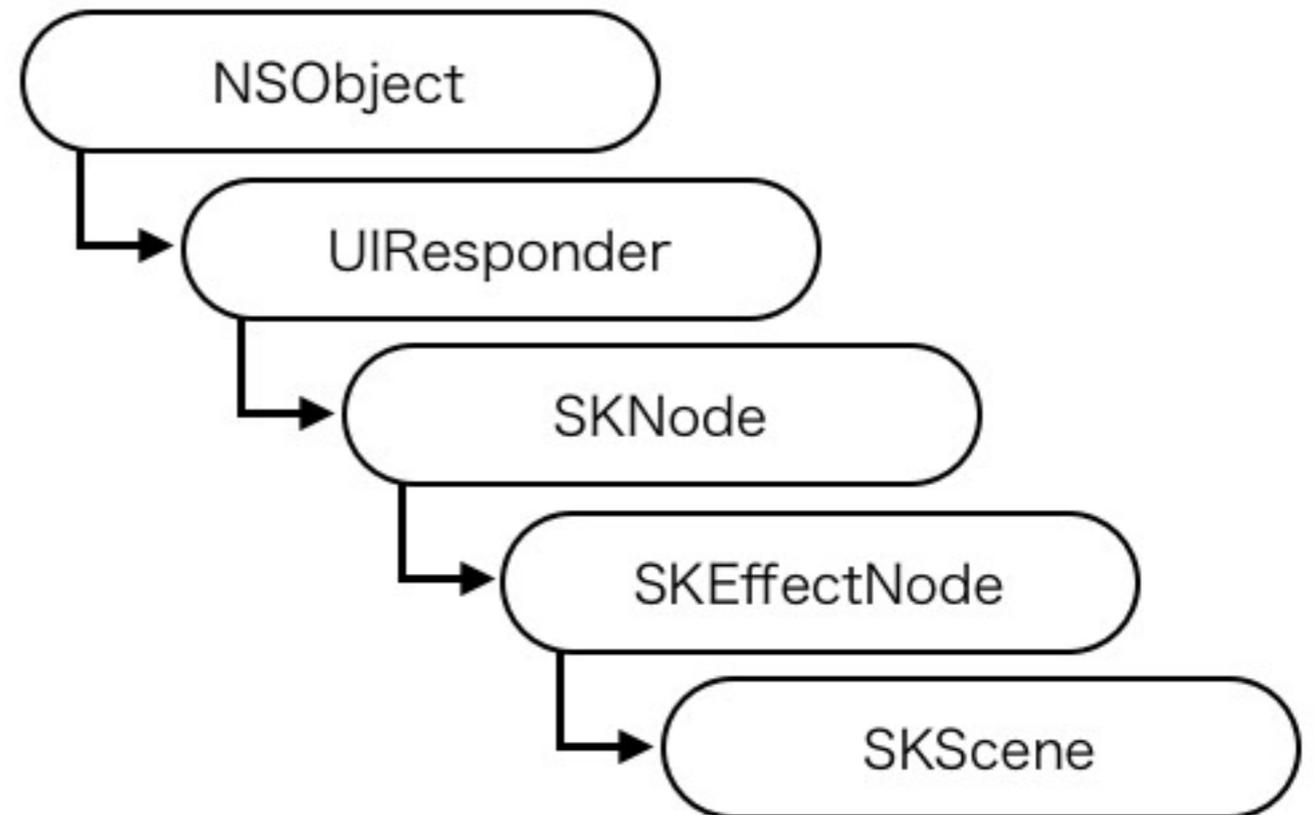
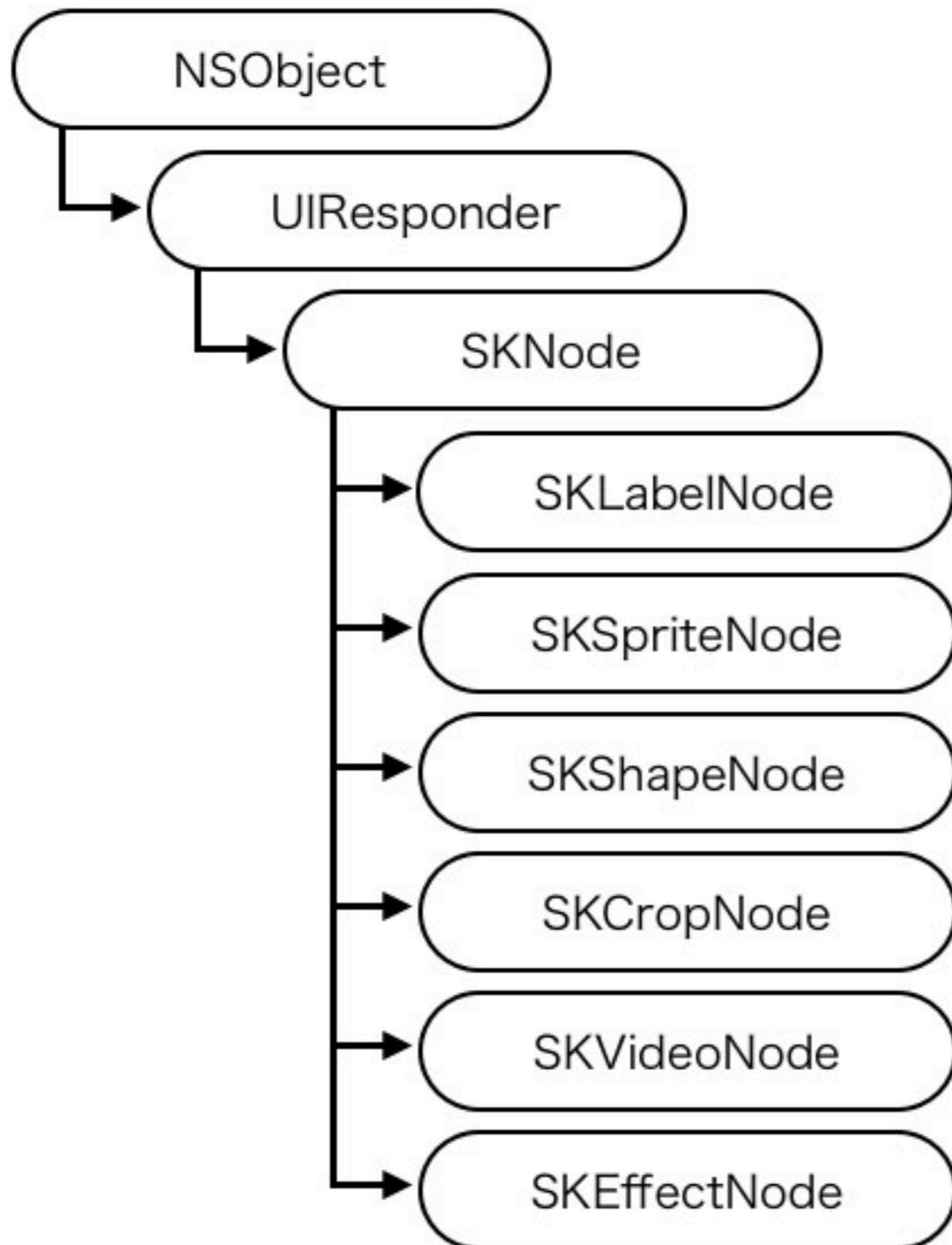
- **SKAction**

スプライトに動きを与えるクラス

Sprite Kitクラスを使ったゲーム構成イメージ

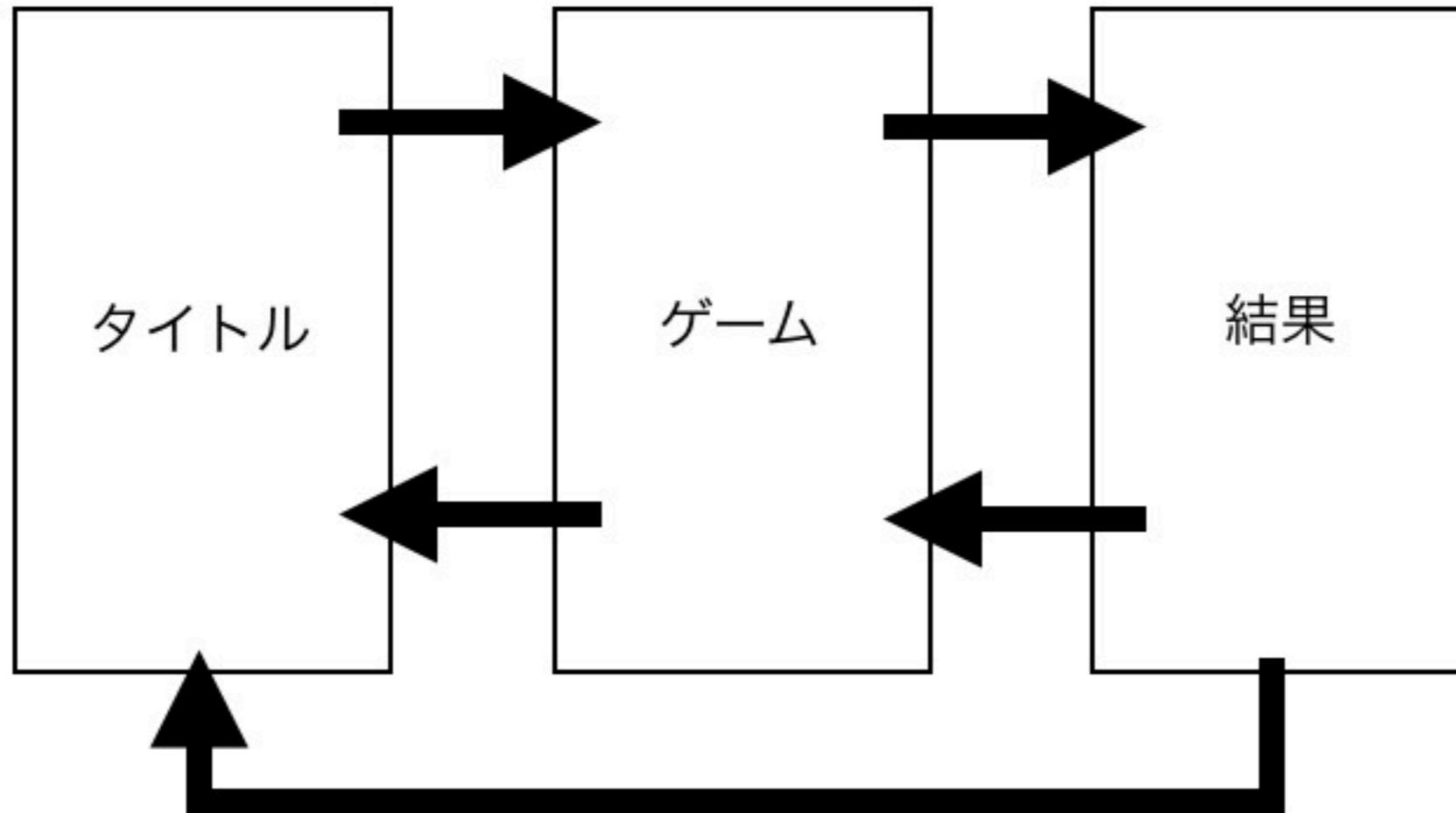


Sprite Kitのクラス階層

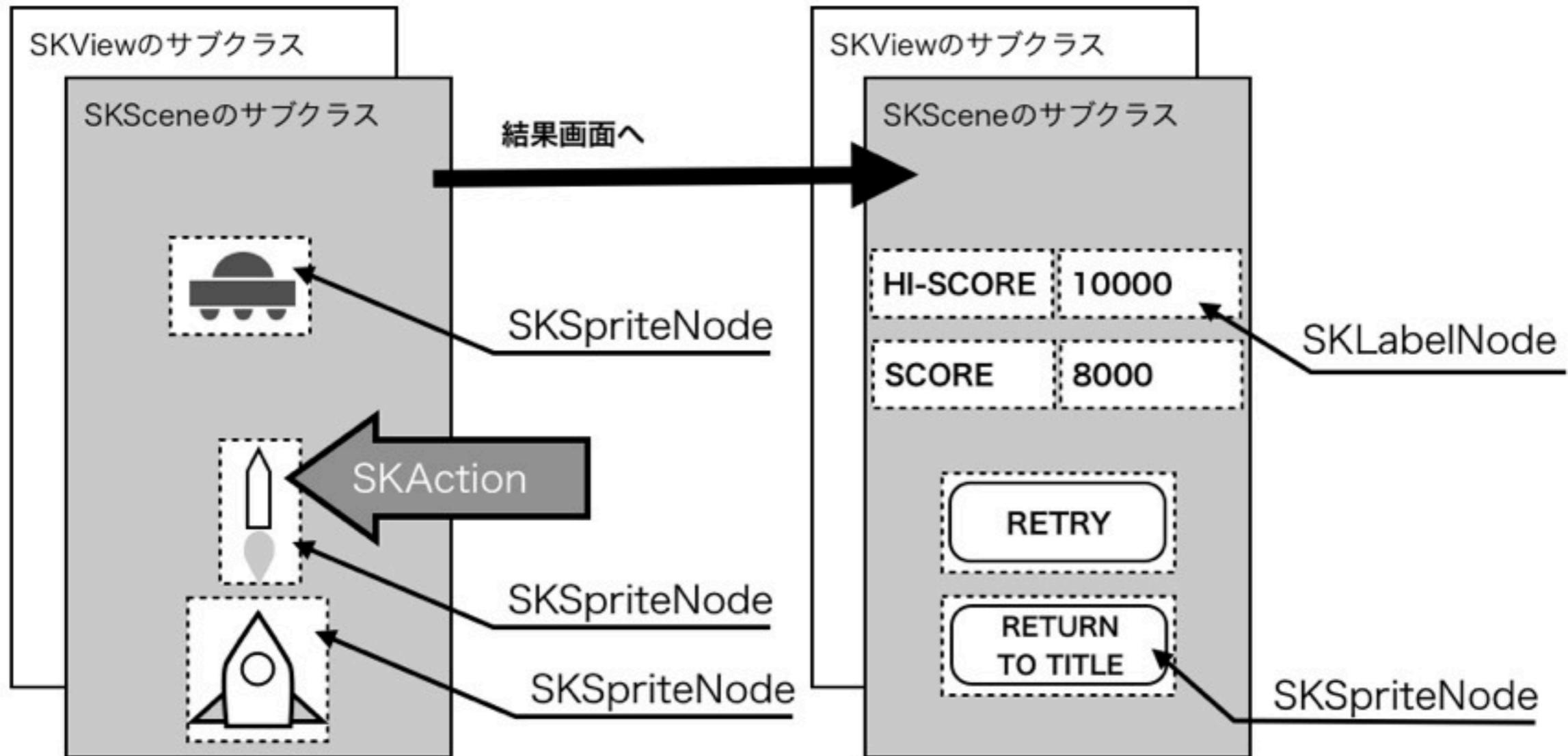


**SKSceneはSKNode
の孫クラス**

Sprite Kitを使ったゲーム画面構成



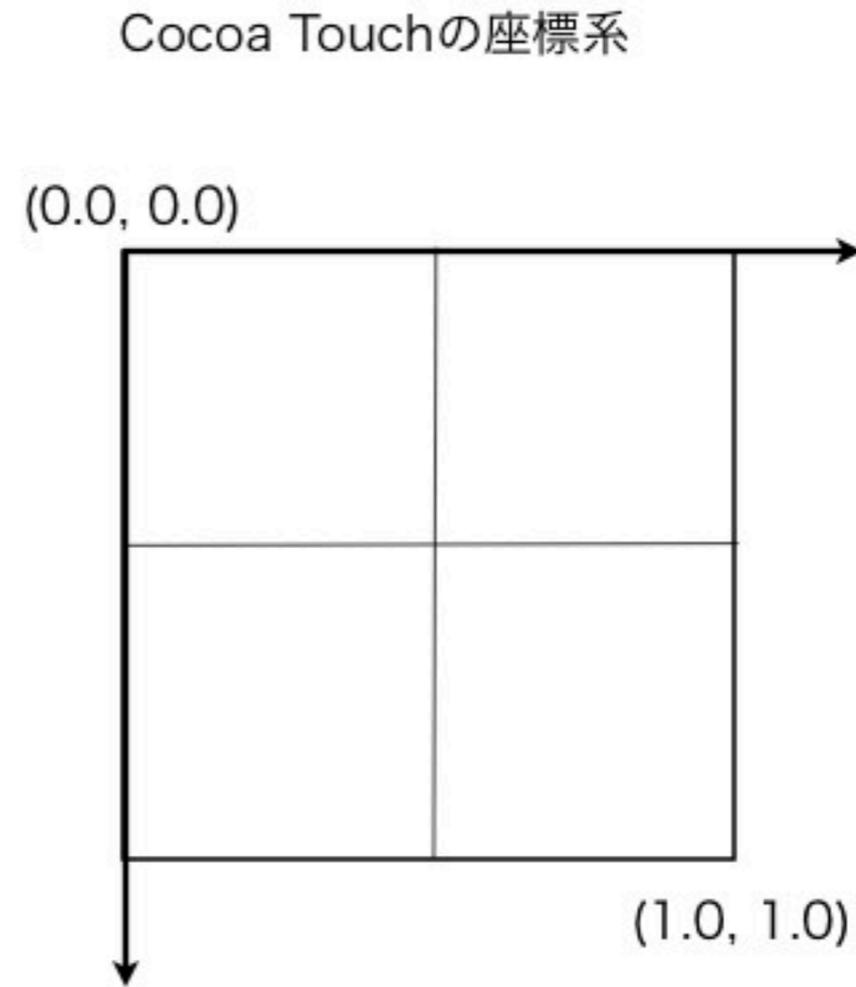
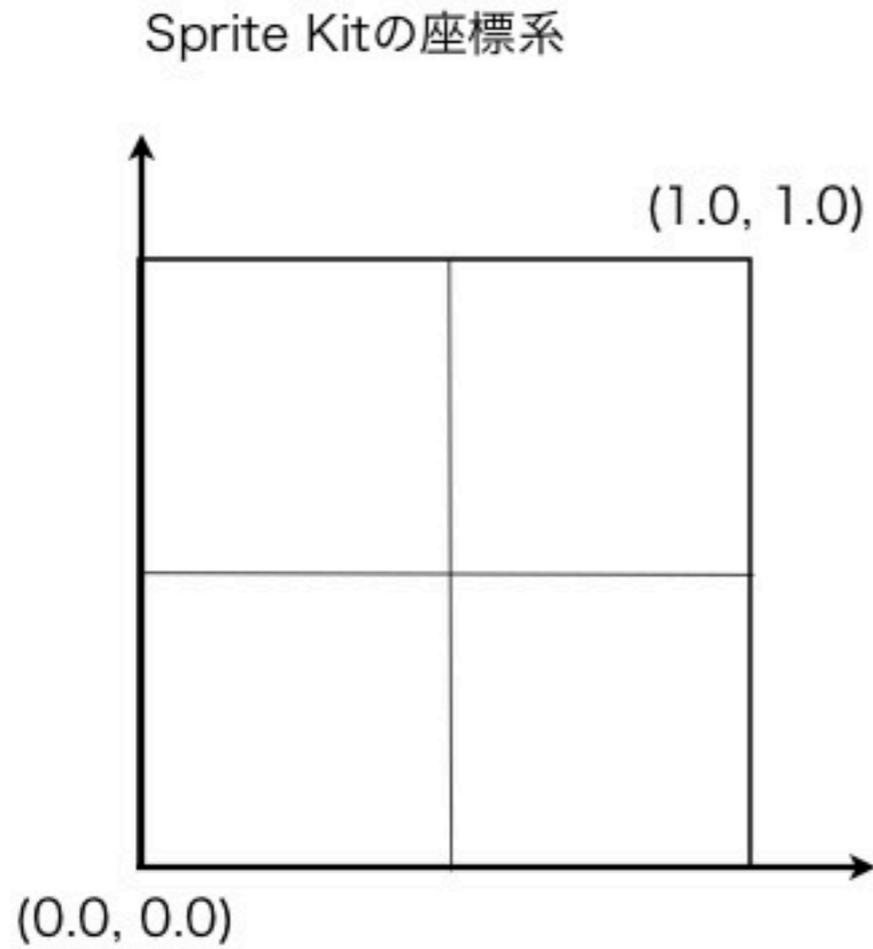
Sprite Kitを使ったゲーム画面構成



シーン単位で画面を切り替えるイメージ

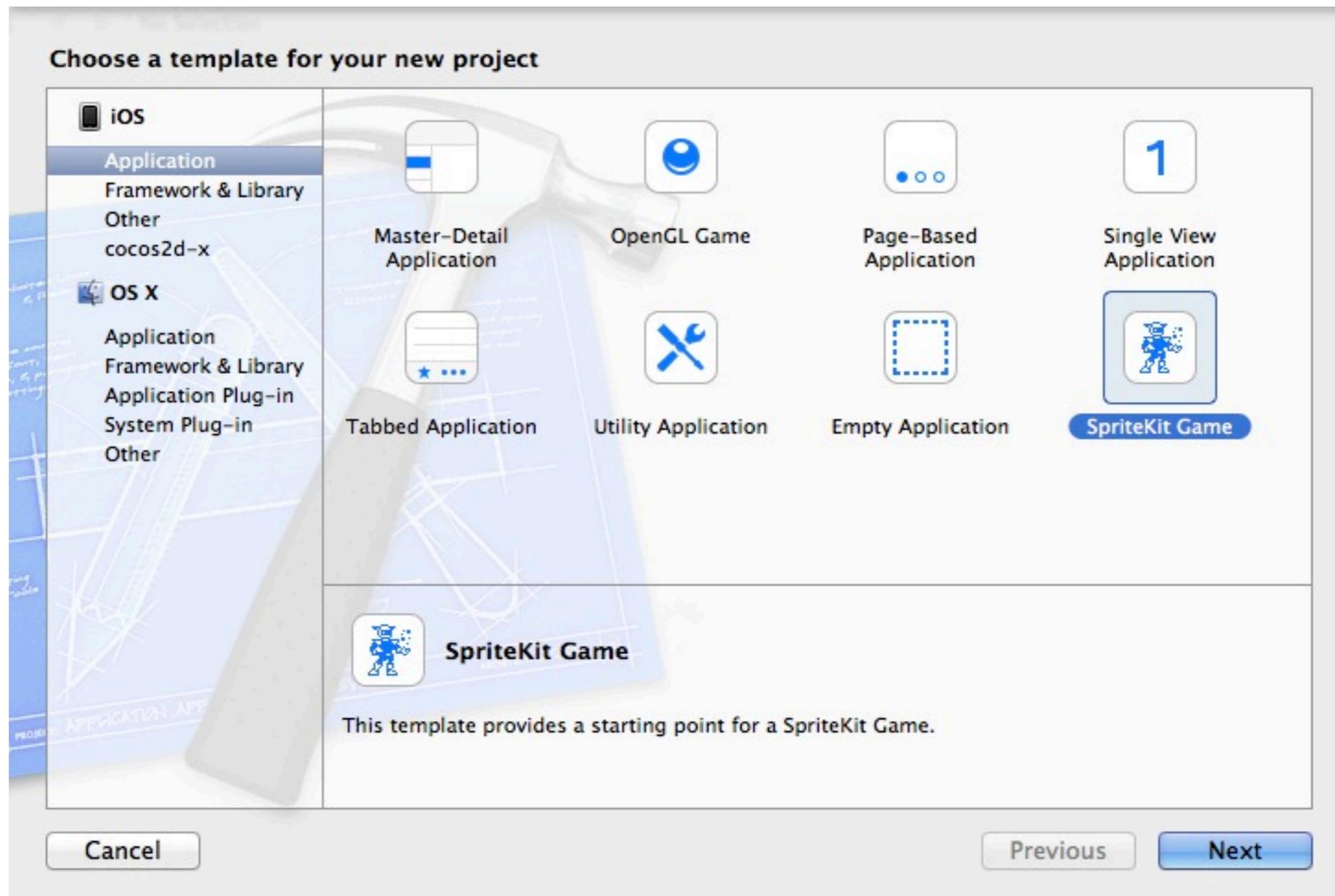
Sprite Kitにはシーン切り替えのAPI用意されている。

Sprite Kitの座標系

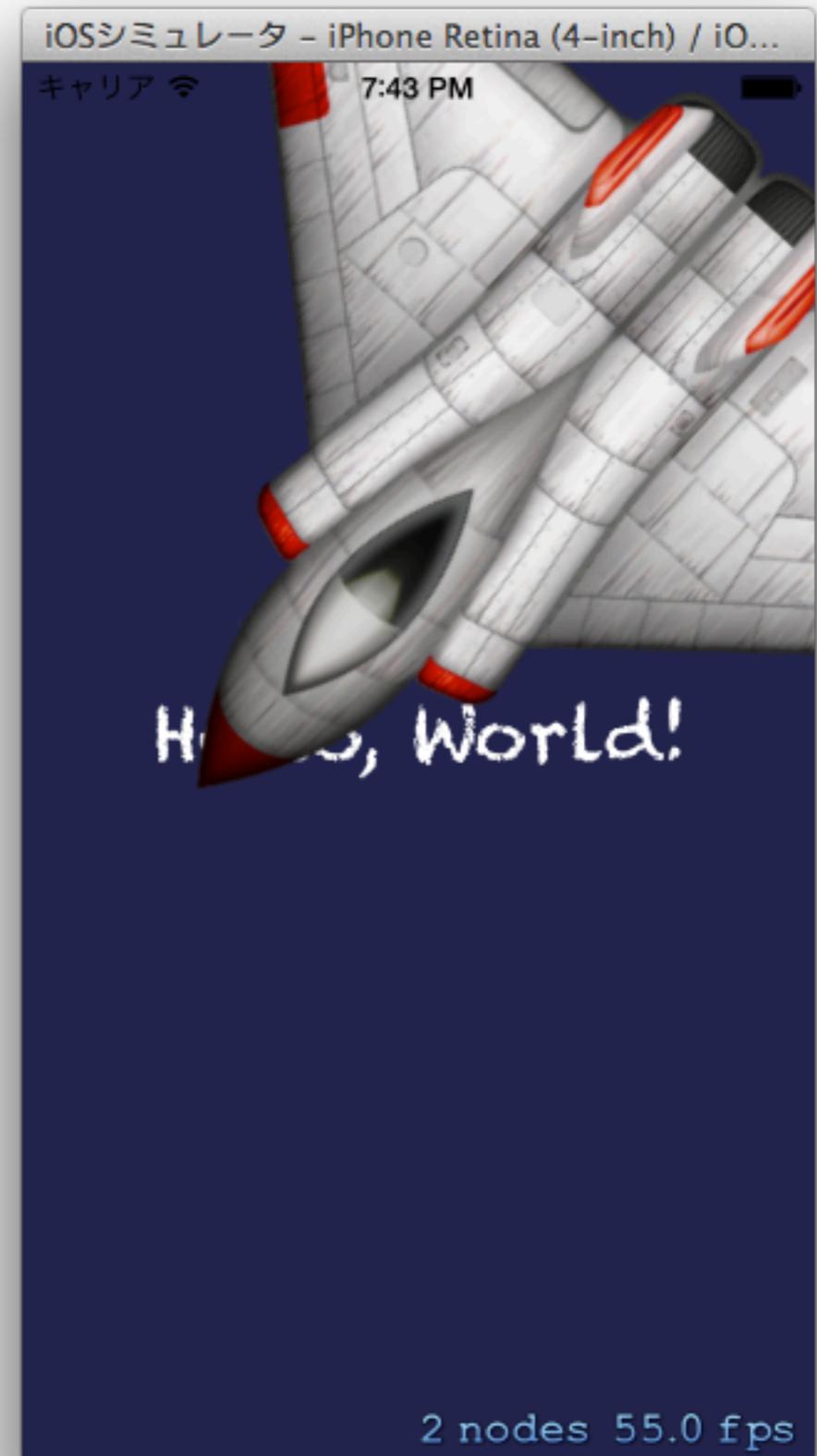
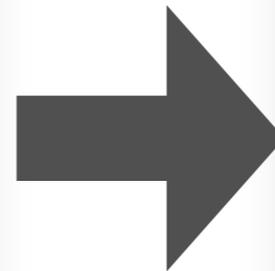


左下原点！

Sprite Kitを使ったXcodeプロジェクトの作成



Sprite Kitを使ったXcodeプロジェクトの作成



シーンの作成

```
- (void) viewDidLoad
{
    [super viewDidLoad];

    // Configure the view.
    SKView * skView = (SKView *)self.view;
    skView.showsFPS = YES;
    skView.showsNodeCount = YES;

    // Create and configure the scene.
    SKScene * scene = [MyScene sceneWithSize:skView.bounds.size];
    scene.scaleMode = SKSceneScaleModeAspectFill;

    // Present the scene.
    [skView presentScene:scene];
}
```

スプライトの作成

文字スプライト

```
SKLabelNode *myLabel = [SKLabelNode labelNodeWithFontNamed:@"Chalkduster"];

myLabel.text = @"Hello, World!";
myLabel.fontSize = 30;
myLabel.position = CGPointMake(CGRectGetMidX(self.frame),CGRectGetMidY(self.frame));

[self addChild:myLabel];
```

画像スプライト

```
SKSpriteNode *sprite = [SKSpriteNode spriteNodeWithImageNamed:@"Spaceship"];

sprite.position = CGPointMake(CGRectGetMidX(self.frame),CGRectGetMidY(self.frame));

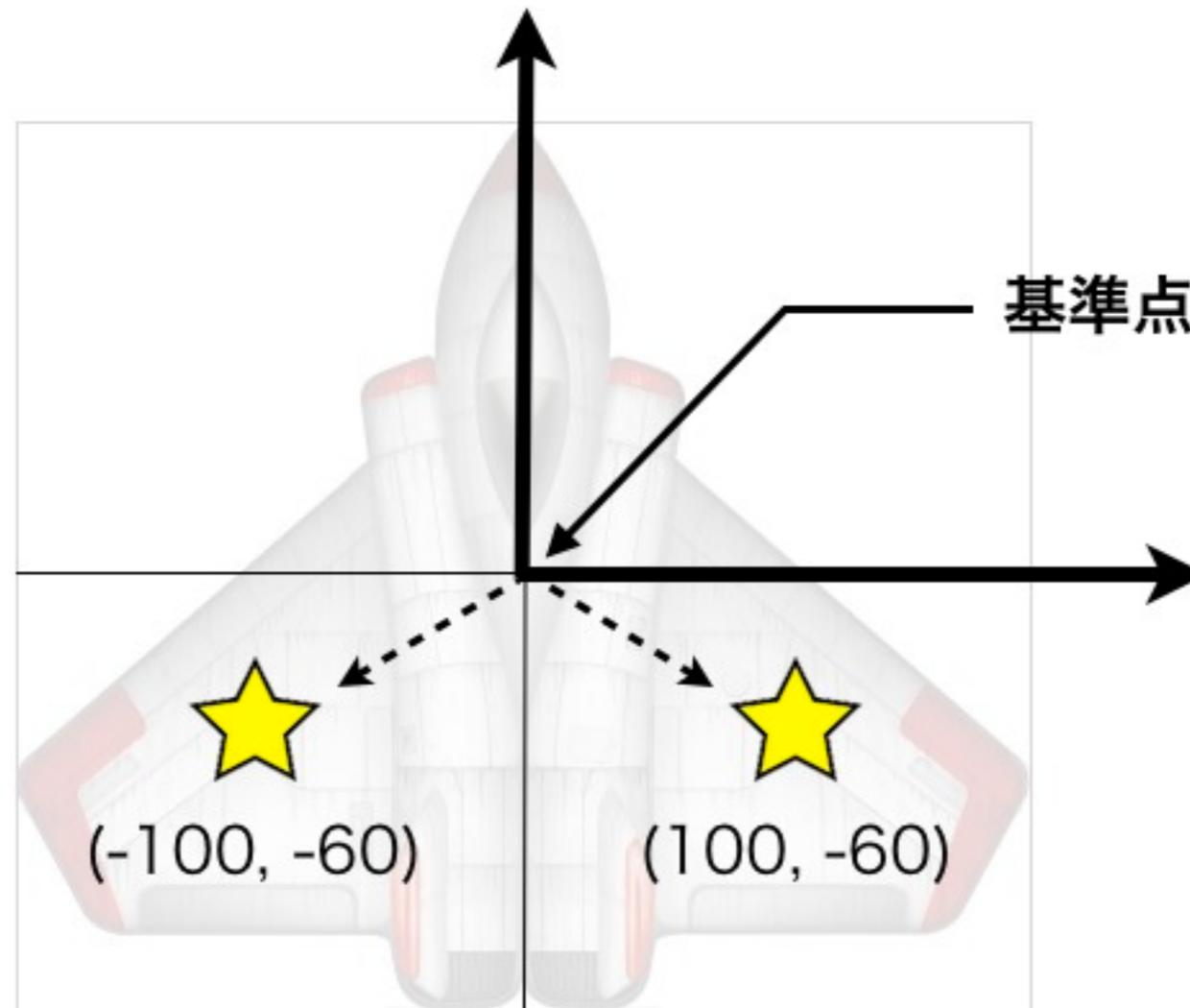
[self addChild:sprite];
```

DEMO

テクスチャと子ノードの追加

Xcode project : SpriteKitDemo1

ノードの原点



アクションの作成とノードへの追加

```
SKSpriteNode *sprite;  
sprite = [SKSpriteNode spriteNodeWithImageNamed:@"Spaceship"];
```

...

```
SKAction *action = [SKAction rotateByAngle:M_PI duration:1];  
[sprite runAction:[SKAction repeatActionForever:action]];
```

アクションは、**By**と**To**の2種類

DEMO

アクションあれこれ

Xcode project : SpriteKitDemo1

アクションの平行実行と連続実行

平行実行

```
NSArray* ary = @[ [SKAction moveByX:100 y:50 duration:1.0],  
                  [SKAction rotateByAngle:M_PI/2 duration:1.0],  
                  [SKAction scaleBy:2.0 duration:1.0] ];  
SKAction* action = [SKAction group:ary];  
[sprite runAction:action];
```

連続実行

```
NSArray* ary = @[ [SKAction moveByX:100 y:50 duration:1.0],  
                  [SKAction rotateByAngle:M_PI/2 duration:1.0],  
                  [SKAction scaleBy:2.0 duration:1.0] ];  
SKAction* action = [SKAction sequence:ary];  
[sprite runAction:action];
```

DEMO

アクションの平行実行と連続実行

Xcode project : SpriteKitDemo1

カスタムアクション

子ノードにアクションを実行させる

ブロックを呼ぶアクション

繰り返し実行するアクション

ノードを削除するアクション

DEMO

カスタムアクション

Xcode project : SpriteKitDemo1

物理シミュレーション

物理シミュレーションのクラス

- **SKPhysicsWorld**

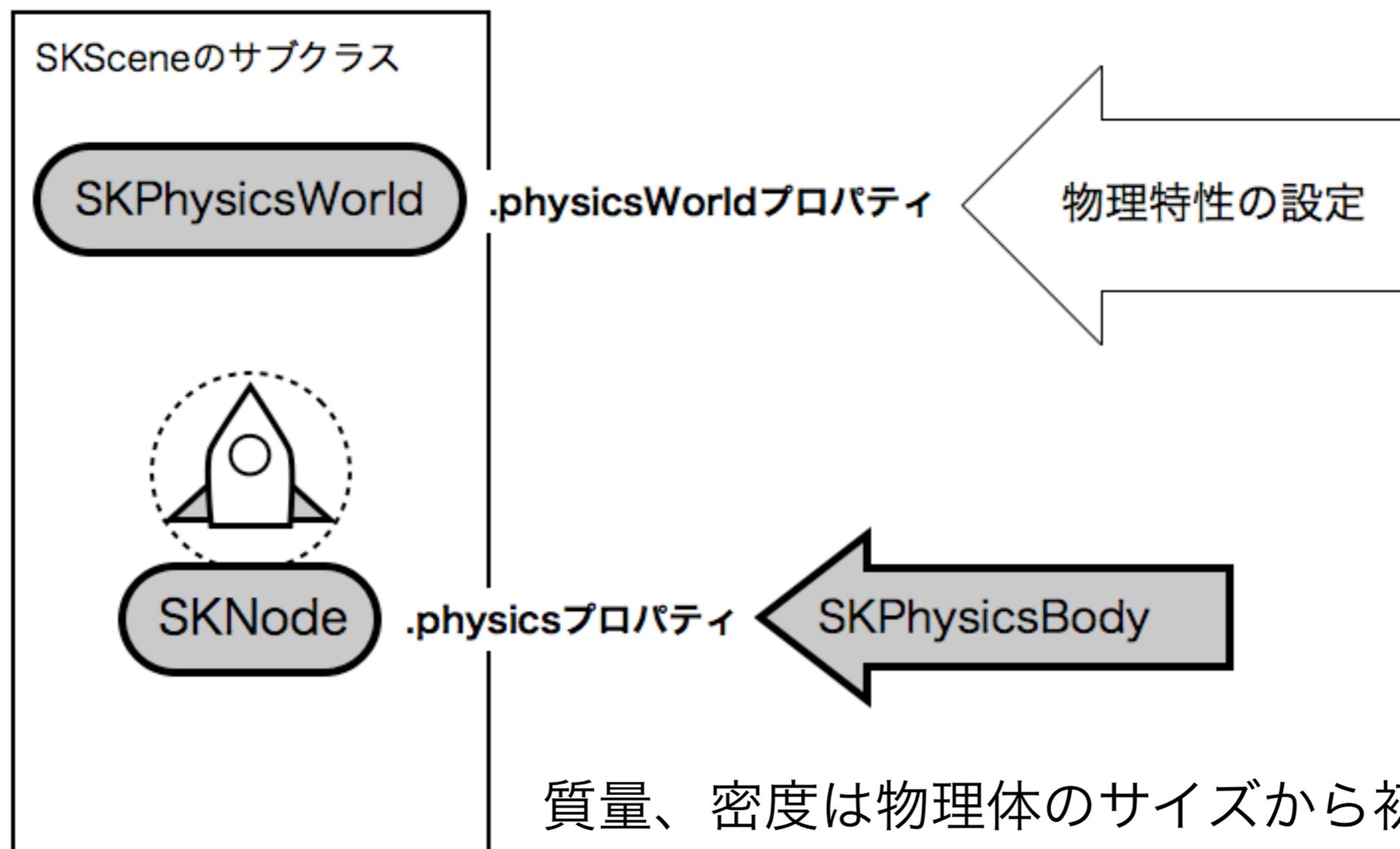
物理シミュレーションの世界。重力や物体同士の衝突判定。

- **SKPhysicsBody**

物理シミュレーション上に存在する物体。物理的特性をもつオブジェクト。

物理特性：サイズ／質量／密度／摩擦／空気抵抗／弾み

Sprite Kitでの物理シミュレーションイメージ



質量、密度は物理体のサイズから初期値が自動的に計算される。

物理体には2つの種類がある

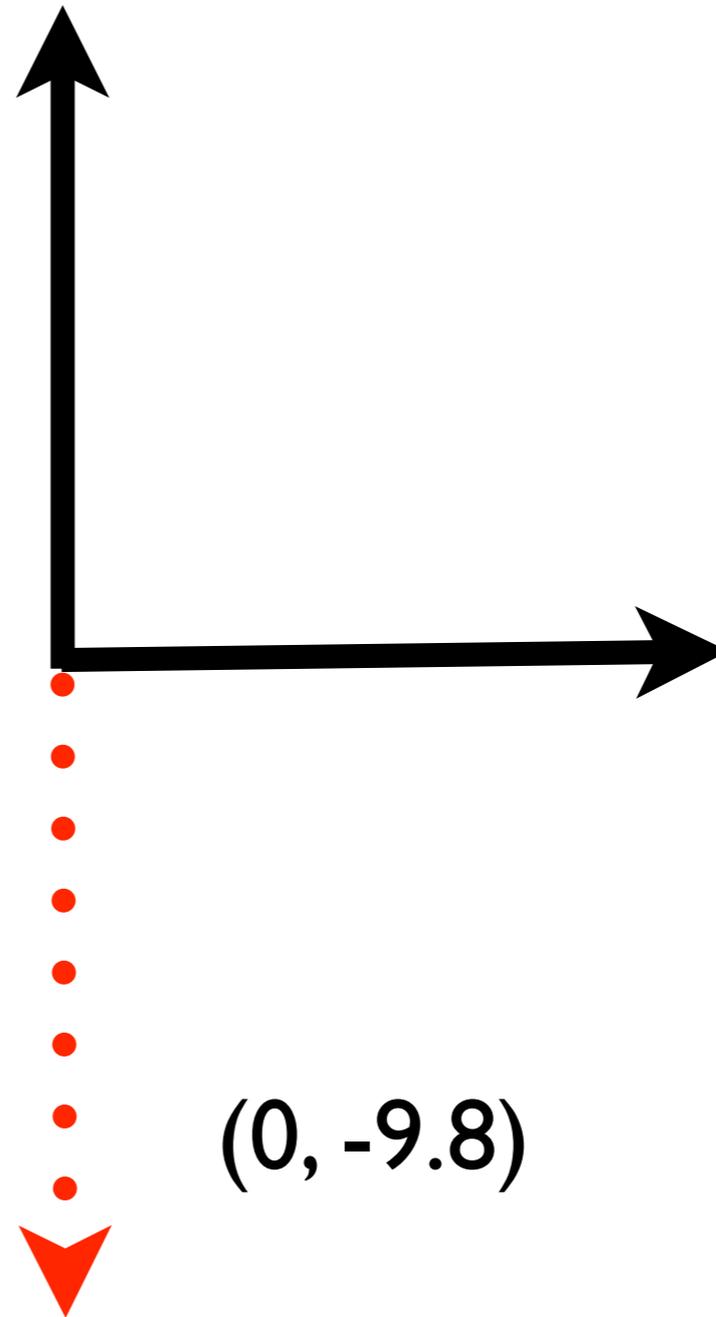
- **ボリウムベース物理体**

質量と体積を持ち重力や衝突といった他からの力の影響を受ける。ゲームキャラなどのゲーム内で動き回るオブジェクトに使用。

- **エッジベース物理体**

質量を持たず重力や衝突などの影響を受けない。物理シミュレーションの境界や空間を表すために使用され

重力のベクトル (Default)



物理体へ力を加える

velocity

方向と速度を与える。

impulse

衝動を加える。

Force

勢いを加える。

DEMO

重力シミュレーション

Xcode project : SpriteKitDemo2

接触と衝突の判定

接触と衝突の判定方法

contactDelegateプロパティ (**SKPhysicsWorld**)

シミュレーションされている物理体の衝突判定をデリゲートメソッドで受ける。

■ **SKPhysicsContactDelegate** プロトコル

2つの物体が衝突したときに呼ばれる

– (**void**)didBeginContact:(**SKPhysicsContact ***)contact

2つの物体が接触が終了したときに呼ばれる

– (**void**)didEndContact:(**SKPhysicsContact ***)contact

接触と衝突の判定方法

categoryBitMaskプロパティ (**SKPhysicsBody**)

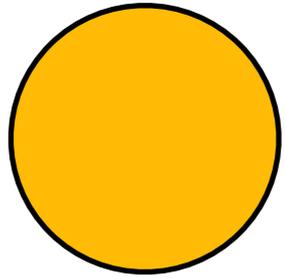
物理体をカテゴリ分けするための32ビットマスク。

contactTestBitMaskプロパティ (**SKPhysicsBody**)

当たりチェックするための32ビットマスク。categoryTestBitMaskとのAND和が1なら接触デリゲートメソッドが呼ばれる。

collisionBitMaskプロパティ (**SKPhysicsBody**)

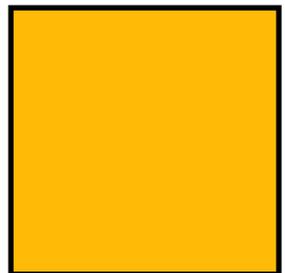
他物理体との接触判定をするための32ビットマスク。



categoryBitMask : 00000001

contactTestBitMask : 00000010

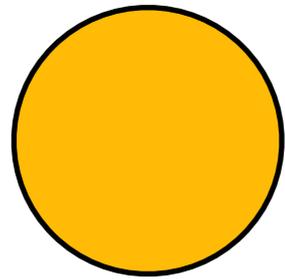
collisionBitMask : 00000010



categoryBitMask : 00000010

contactTestBitMask : 00000001

collisionBitMask : 00000001



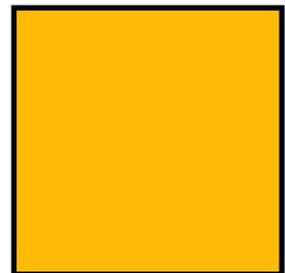
categoryBitMask : 00000001

contactTestBitMask : 00000010

collisionBitMask : 00000010

物理的接触！

AND=I



categoryBitMask : 00000010

contactTestBitMask : 00000001

collisionBitMask : 00000001

categoryBitMask : 00000001

contactTestBitMask : 00000010

collisionBitMask : 00000010

衝突！

デリゲートメソッド

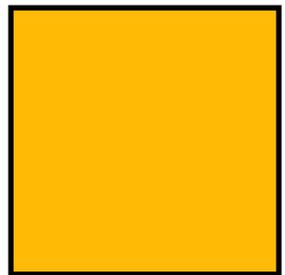
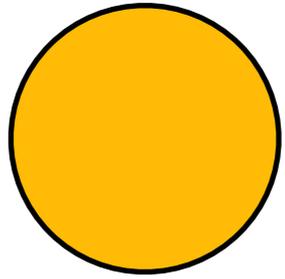
呼ばれる！

AND=I

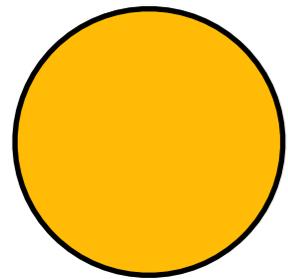
categoryBitMask : 00000010

contactTestBitMask : 00000001

collisionBitMask : 00000001



各ビットマスクのデフォルト値



`categoryBitMask` : 11111111

`contactTestBitMask` : 00000000

`collisionBitMask` : 11111111

DEMO

Meteo Impact!!

Xcode project : SpriteKitDemo3