

# nendSDK cocos2d-x モジュール(v3.0 用) ver 1.0.0

## 設定手順書

## 更新履歴

バージョン	公開日付	更新内容
ver 1.0.0	2014/4/4	新規作成

## 目次

◆nendSDK cocos2d-x モジュールについて.....	4
1.nendSDK 導入のおおまかな流れ.....	4
2.ファイル構成.....	4
3.対応環境.....	4
【重要】 インフォメーションボタンについて.....	5
◆組み込み方法.....	6
1.iOS への組み込み.....	6
2.android への組み込み.....	8
3.Cocos2d-x v3.0 rc0 への組み込み.....	11
◆実装手順 cocos2d-x.....	14
1.バナー広告を表示する.....	14
2.アイコン広告を表示する.....	15
◆nendSDK cocos2d-x の内容.....	17
1.NendModule の内容.....	17
2.NendIconModule の内容.....	19
3.Observer パターンによる通知（任意） .....	21
◆よくある質問.....	24

## ◆nendSDK cocos2d-x モジュールについて

### 1. nendSDK 導入のおおまかな流れ

① nend 管理画面でアプリ登録と広告枠登録を行います。

- ※ 本マニュアルは広告枠登録後、「apiKey」「spotID」を発行し SDK を入手している前提で説明を行います。
- ※ 広告枠を登録していない場合には、別紙「管理画面マニュアル」をご参照の上、ご申請ください。
- ※ 広告枠を申請後、広告枠の管理> 広告枠> SDK> 「SDK をダウンロード」 で SDK を入手できます。

②本マニュアルに従って nendSDK cocos2d-x モジュールをアプリに組み込みます。

- ※ 本マニュアルは、nendSDK cocos2d-x モジュール導入に関するマニュアルとなります。
- ※ nendSDK については、Android・iOS それぞれの SDK マニュアルをご参照ください。

### 2. ファイル構成

nendSDK\_cocos2dX\_Module\_for\_v3.0.zip

NendModule/	モジュールフォルダ	
NendModule.h	バナー広告用ヘッダファイル	(iOS、android 共通)
NendIconModule.h	アイコン広告用ヘッダファイル	(iOS、android 共通)
NendModule.mm	バナー広告用実装ファイル	(iOS 用)
NendIconModule.mm	アイコン広告用実装ファイル	(iOS 用)
NendModule.cpp	バナー広告用実装ファイル	(android 用)
NendIconModule.cpp	アイコン広告用実装ファイル	(android 用)
javaPackage/	android 実装パッケージ	(android 用)
nendCocos2d-x_Module_Guide.pdf	本マニュアル	

### 3. 対応環境

#### ・ 動作保証対象

- Android : AndroidOS 2.3 以上
- iOS : iOS 5.0 以上

#### ・ 開発環境

- cocos2d-x : 3.0 RC1
- Xcode : Xcode5.0以上

※モジュール内部で使用している nendSDK のバージョンは以下の通りです。

- Android : 2.3.3
- iOS : 2.3.3

## 【重要】 インフォメーションボタンについて

nendSDK バージョン 2.0.0 より、アプリに配信される広告バナーに「インフォメーションボタン」が表示されるようになりました。

この機能が追加された nendSDK を利用することにより、弊社システムが提供する一部ターゲティング広告に対して、よりユーザ（アプリ利用者）自身が容易にオプトアウトの設定をすることが可能になります。



平成 24 年より、総務省においてスマートフォンのプライバシーに関する議論がなされ、「スマートフォンプライバシーイニシアティブ」として取りまとめがされております。

### 参考 URL

[http://www.soumu.go.jp/menu\\_news/s-news/01kiban08\\_02000087.html](http://www.soumu.go.jp/menu_news/s-news/01kiban08_02000087.html)

アプリ提供者、情報収集モジュール提供者、広告配信事業者の自主的かつ積極的な取り組みを期待されておりますので、メディアパートナー様におかれましてもご確認頂きますようお願いいたします。また、弊社においても広告配信システム提供事業者として、より一層透明性の高い取り組みを実施していく所存でございます。

尚、当 SDK における取得情報は、以下ようになっており、個人情報に該当するものは取得しておりません。

・匿名 ID（android では U I I D / iOS では UIID または広告識別子）、端末情報（OS 情報、端末機種種類、URL スキーム情報、Package 名等）

※ URL スキーム情報、Package 名に関しては、SDK 内でのみ利用しており、外部送信および保存等を行うことは一切ございません。

## ◆組み込み方法

### 1. iOS への組み込み

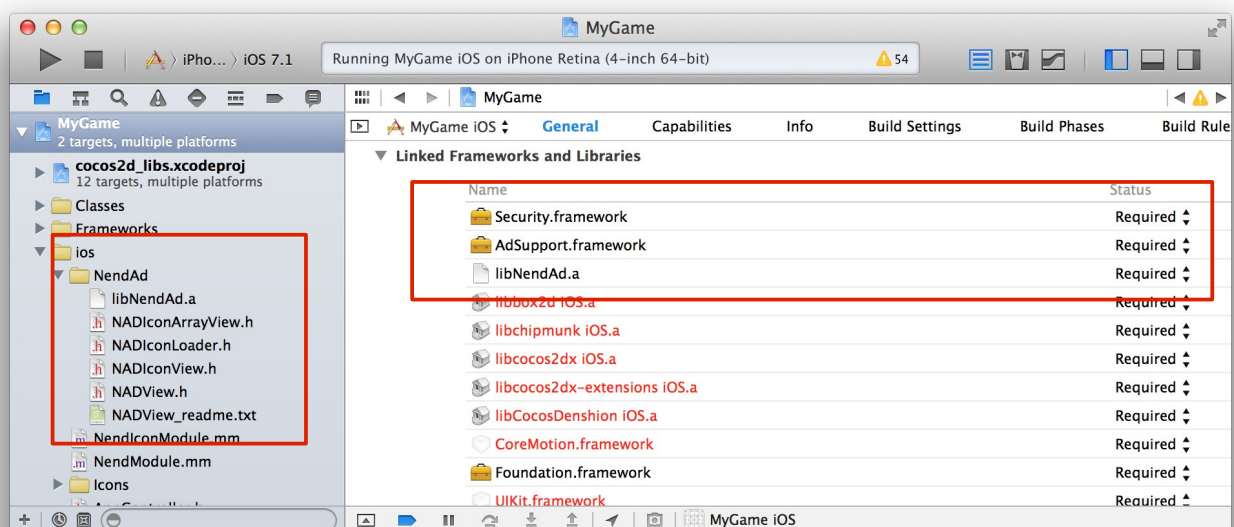
#### ① NendSDK の組み込み

nendSDK iOS 設定マニュアルに従い、

1. プロジェクトの nendSDK 追加

2. 必須フレームワークの追加

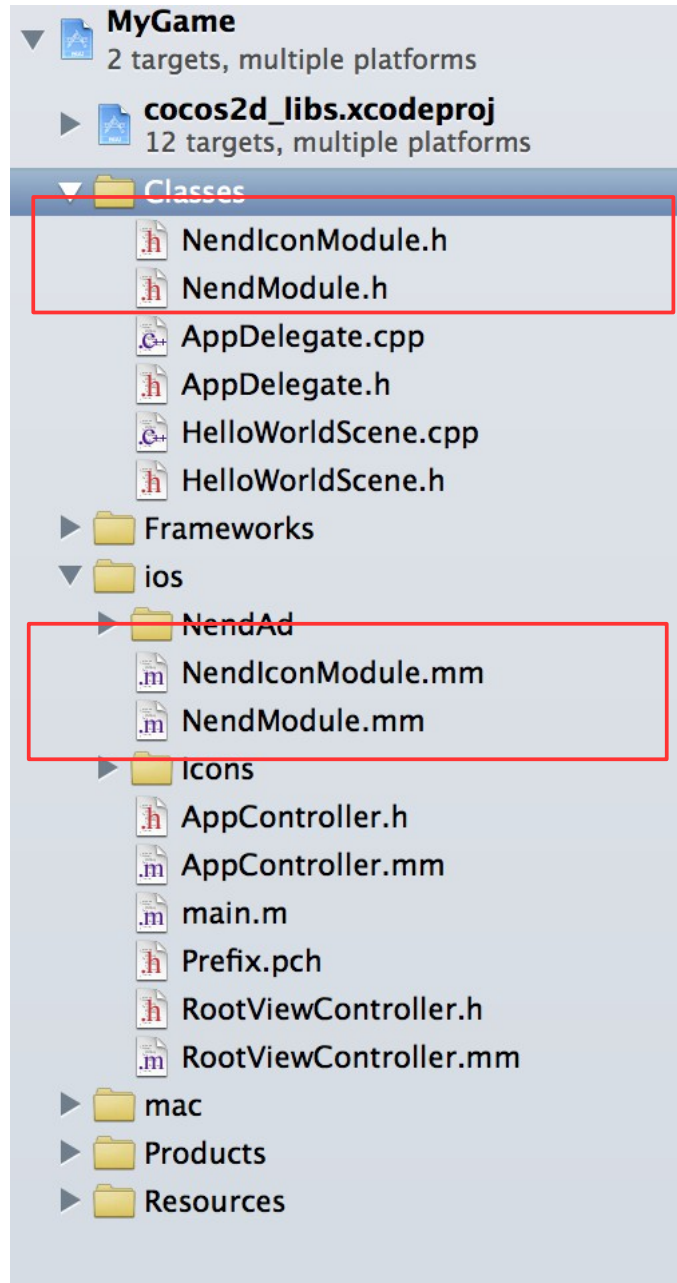
を行う。



## ② cocos2d-x モジュールの組み込み

Nend モジュールのファイル

(NendModule.h,NendIconModule.h,NendModule.mm,NendIconModule.mm)をプロジェクトに追加する。



## 2. android への組み込み

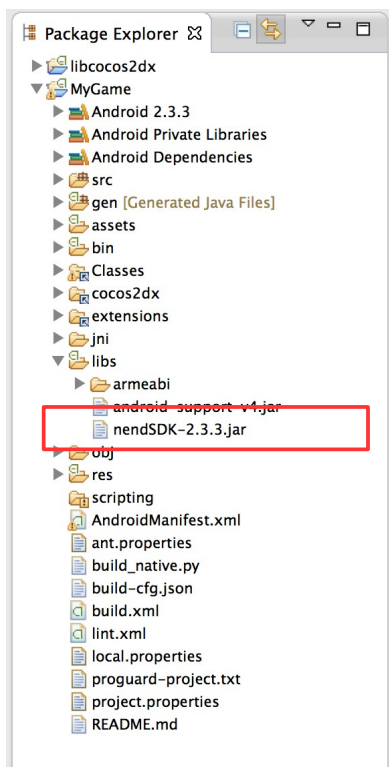
### ① SDK の組み込み

nendSDK Android 設定マニュアルに従い、

1. nendSDK-x.x.x.jar ファイルの格納

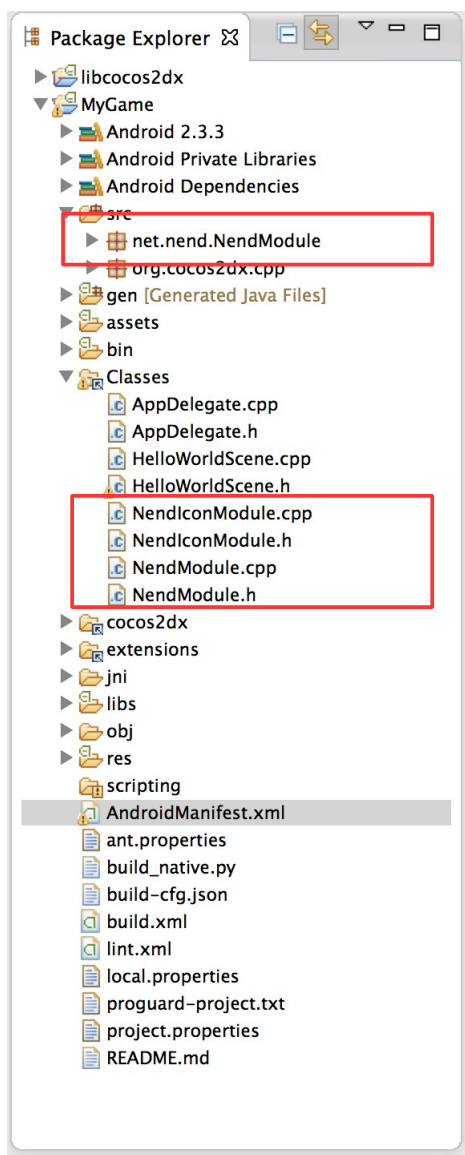
2. パーMISSIONの設定

を行う。





- ② **Nend モジュールのヘッダーファイル(NendModule.h,NendIconModule.h)、**  
**c++ソースファイル(NendModule.cpp,NendIconModule.cpp)、**  
**java ソースファイル(パッケージ)(NendModule.java,NendIconModule.java)をプロジェクトに**  
**追加する。**



### ③ビルドの設定変更

Android.mk の LOCAL\_SRC\_FILES に、

**../../Classes/NendIconModule.cpp ¥**

**../../Classes/NendModule.cpp**

を追加する。

```
LOCAL_PATH := $(call my-dir)

include $(CLEAR_VARS)

LOCAL_MODULE := cocos2dcpp_shared

LOCAL_MODULE_FILENAME := libcocos2dcpp

LOCAL_SRC_FILES := hellocpp/main.cpp \
    ../../Classes/AppDelegate.cpp \
    ../../Classes/HelloWorldScene.cpp \
    ../../Classes/NendIconModule.cpp \
    ../../Classes/NendModule.cpp

LOCAL_C_INCLUDES := $(LOCAL_PATH)/../../Classes

LOCAL_WHOLE_STATIC_LIBRARIES += cocos2dx_static
LOCAL_WHOLE_STATIC_LIBRARIES += cocosdeshion_stati
```

既存のソースファイルに追記します。  
その際、既存の行の最後に \ を追加します。

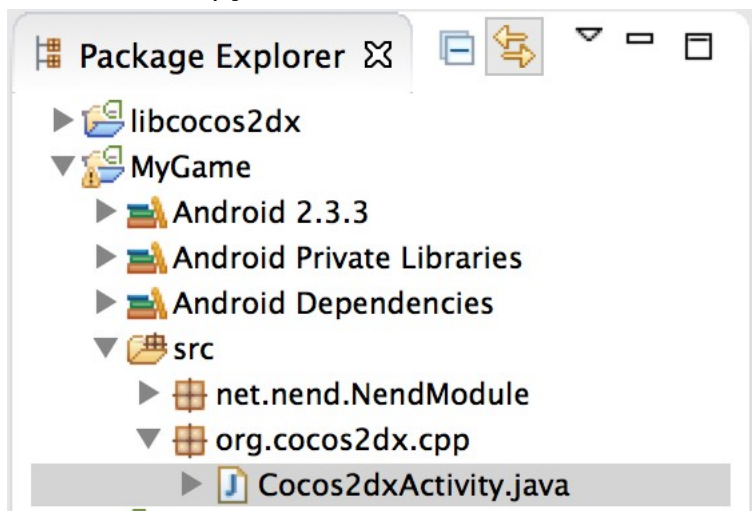
### 3. Cocos2d-x v3.0 rc0 への組み込み

cocos2d-x v3.0 rc0 に組み込む場合は、java ソースファイルに編集を行います。

#### ① Cocos2dxActivity.java の編集

Cocos2dxActivity.java に以下の編集を行います。

Cocos2dxActivity.java は、以下に格納されています。



#### ・メンバ変数の追加

Cocos2dxActivity クラスにメンバ変数を追加します。

```
private static Context sContext = null;
```

例) Cocos2dxActivity.java (一部)

```
package org.cocos2dx.cpp;

import android.app.NativeActivity;

// The name of .so is specified in AndroidManifest.xml. NativeActivity will load it automatically
// You can use "System.loadLibrary()" to load other .so files.

public class Cocos2dxActivity extends NativeActivity{
    private static Context sContext = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
    }
}
```

### ・ onCreate メソッドに処理の追加

onCreate メソッド内にメンバ変数に値を設定する処理を追加します。

**sContext = this;**

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);

    //For supports translucency

    //1.change "attribs" in cocos\2d\platform\android\nativeactivity.cpp
    /*const EGLint attribs[] = {
        EGL_SURFACE_TYPE, EGL_WINDOW_BIT,
        EGL_RENDERABLE_TYPE, EGL_OPENGL_ES2_BIT,
        //EGL_BLUE_SIZE, 5,    -->delete
        //EGL_GREEN_SIZE, 6,   -->delete
        //EGL_RED_SIZE, 5,     -->delete
        EGL_BUFFER_SIZE, 32,  //-->new field
        EGL_DEPTH_SIZE, 16,
        EGL_STENCIL_SIZE, 8,
        EGL_NONE
    };*/

    //2.Set the format of window
    // getWindow().setFormat(PixelFormat.TRANSLUCENT);

    sContext = this;
}
```

### ・ getContext メソッドの追加

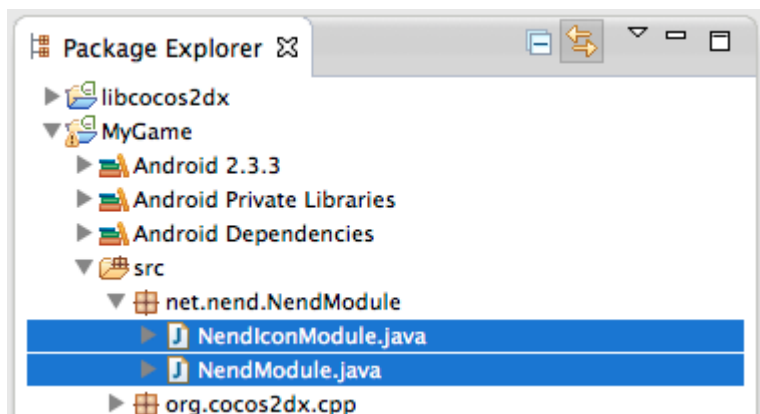
getContext メソッドを Cocos2dxActivity クラス内に追加します。

```
public static Context getContext() {
    return sContext;
}
```

## ② NendIconModule.java, NendModule.java の編集

Nend モジュールの java ソースファイルに以下の編集を行います。

対象のファイルは、以下に格納されています。

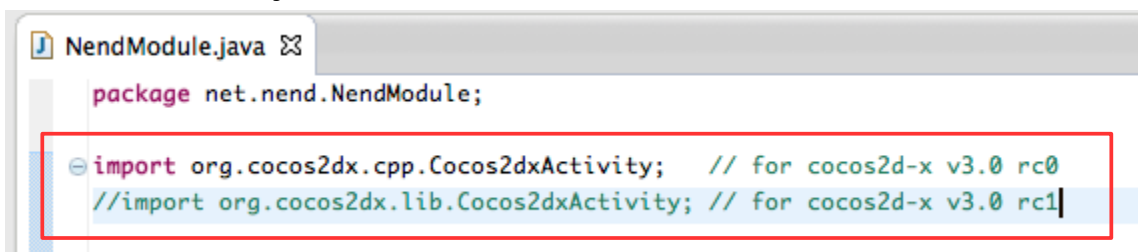


### ・ import するクラスの編集

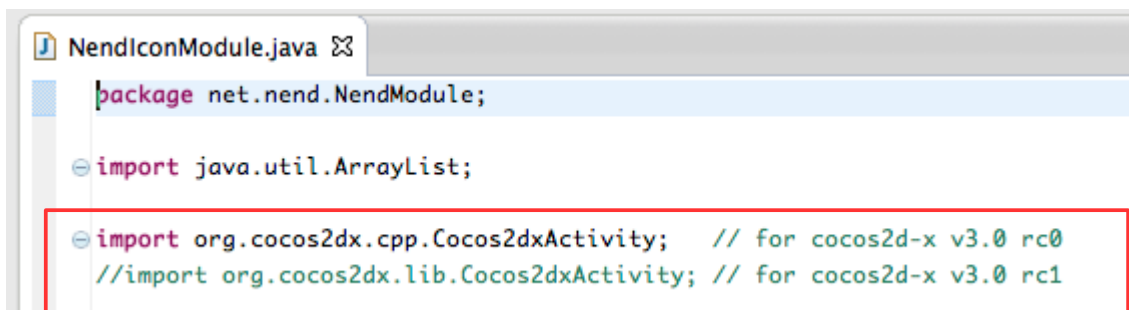
NendIconModule.java, NendModule.java で記述されている import に以下の編集をします。

- ・ org.cocos2dx.**lib**.Cocos2dxActivity の import を削除する。（コメントにする）
- ・ org.cocos2dx.**cpp**.Cocos2dxActivity の import を追加する。（コメントを解除する）

例) NendModule.java



例) NendIconModule.java



## ◆実装手順 cocos2d-x

cocos2d-x のソースファイルに広告を表示するプログラムを実装します。

### 1. バナー広告を表示する

#### 2 – 1. 広告を表示する scene のソースファイルにヘッダーファイルの宣言を追加する。

例) HelloWorldScene.cpp (一部) ヘッダーファイル宣言部分

```
#include "HelloWorldScene.h"
#include "NendModule.h"

USING_NS_CC;

CCScene* HelloWorld::scene()
{
    . . .
```

#### 1 – 2. 広告を表示する処理を呼び出す

例) 画面上部にバナー広告を表示する場合

例) HelloWorldScene.cpp (一部) - HelloWorld::init()

```
// on "init" you need to initialize your instance
bool HelloWorld::init()
{
    . . .

    // add the sprite as a child to this layer
    this->addChild(pSprite, 0);

    char apiKey[] = "[管理画面より発行された apiKey]";
    char spotID[] = "[管理画面より発行された spotID]";
    NendModule::createNADViewTop(apiKey, spotID);

    return true;
}
```

## 2. アイコン広告を表示する

### 2-1. 広告を表示する scene でヘッダーファイルを追加する。

例) HelloWorldScene.cpp (一部) ヘッダーファイル宣言部分

```
#include "HelloWorldScene.h"
#include "NendIconModule.h"

USING_NS_CC;

CCScene* HelloWorld::scene()
{
    . . . . .
}
```

### 2-2. 広告を表示する処理を呼び出す

例) HelloWorldScene.cpp (一部) - HelloWorld::init()

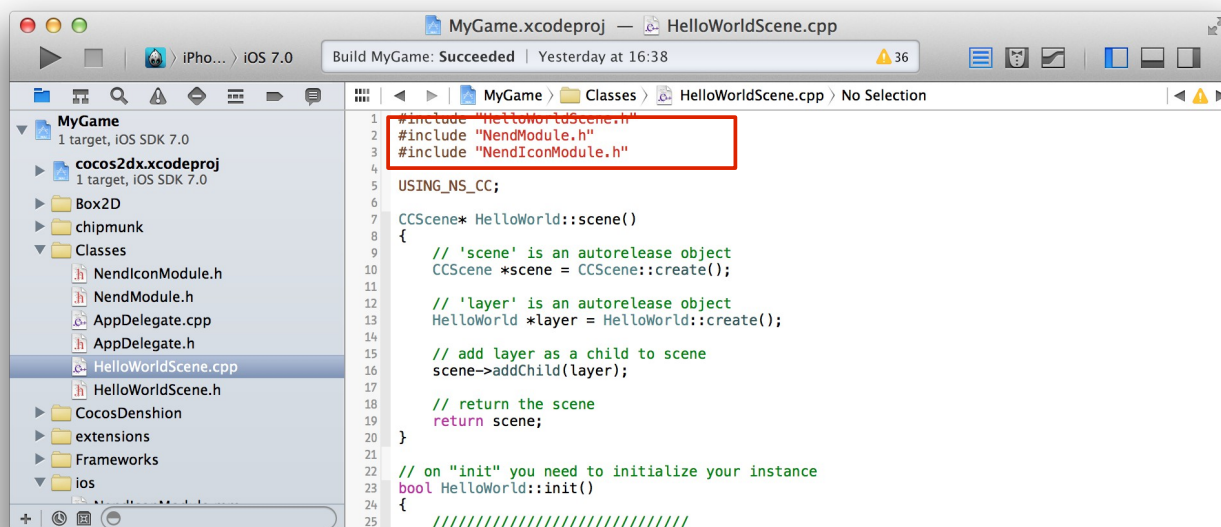
```
// on "init" you need to initialize your instance
bool HelloWorld::init()
{
    . . .
    // add the sprite as a child to this layer
    this->addChild(pSprite, 0);

    char iconApiKey[] = "[管理画面より発行された apiKey]";
    char iconSpotID[] = "[管理画面より発行された spotID]";
    NendIconModule::createNADIconLoader(iconApiKey, iconSpotID);
    NendIconModule::createNADIconViewTop();
    NendIconModule::load();

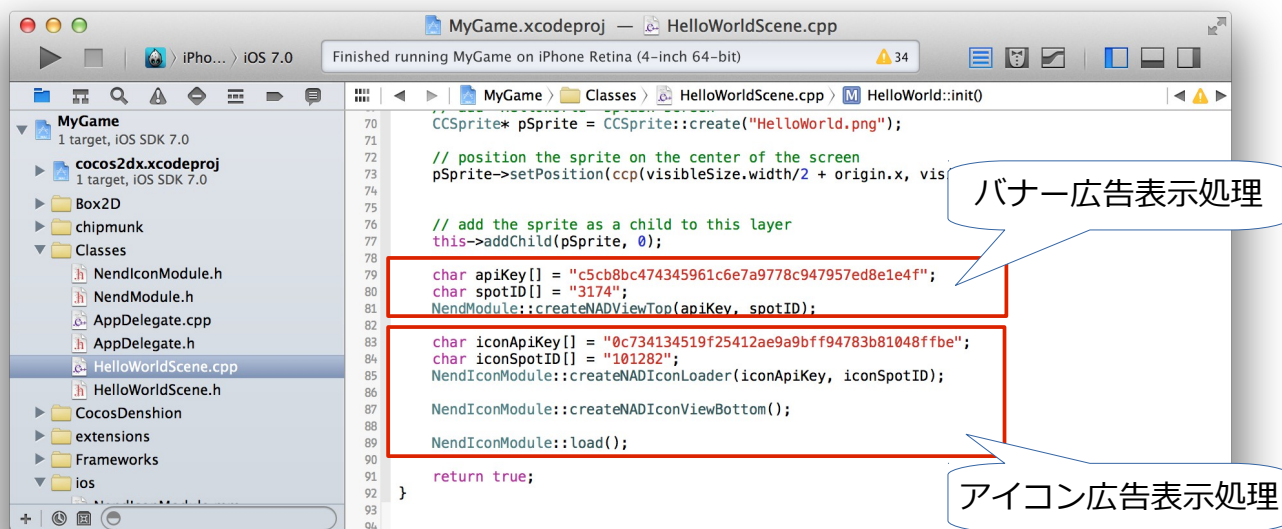
    return true;
}
```

## バナー広告とアイコン広告を同時に表示する例

### 例) ヘッダー宣言の追加



### 例) バナー広告の表示とアイコン広告の表示処理



※この例では、HelloWorld::init()内に処理を実装しています



## ◆nendSDK cocos2d-x の内容

### 1. NendModule の内容

#### ○メソッド

##### ・位置を指定してバナー広告を表示する

**static void createNADView(char\* apiKey, char\* spotID, CCPoint pos);**

nend 管理画面で発行した該当アプリの広告枠の apiKey,spotID,表示する座標をセットします。

```
NendModule::createNADView("[管理画面より発行された apiKey]",
                           "[管理画面より発行された spotID]",
                           Point(0, 50));
```

##### ・画面下部中央にバナー広告を表示する

**static void createNADViewBottom(char\* apiKey, char\* spotID);**

nend 管理画面で発行した該当アプリの広告枠の apiKey,spotID をセットします。

```
NendModule::createNADViewBottom("[管理画面より発行された apiKey]",
                                  "[管理画面より発行された spotID]");
```

##### ・画面上部中央にバナー広告を表示する

**static void createNADViewTop(char\* apiKey, char\* spotID);**

nend 管理画面で発行した該当アプリの広告枠の apiKey,spotID をセットします。

```
NendModule::createNADViewTop("[管理画面より発行された apiKey]",
                              "[管理画面より発行された spotID]");
```

##### ・バナー広告を非表示にする

**static void hideNADView();**

バナー広告を非表示にします。

```
NendModule::hideNADView();
```

##### ・バナー広告を再表示する

**static void showNADView();**

非表示となったバナー広告を再表示します。

```
NendModule::showNADView();
```

##### ・広告の load 一時停止

**static void pause();**

広告の定期ロードを中断します。

```
NendModule::pause();
```

- ・ 広告の load 再開

**static void resume();**

中断中の広告の定期ロードを再開します。

```
NendModule::resume();
```

- ・ リソースの解放

**static void release();**

広告の表示で使用されたリソースを解放します。

```
NendModule::release();
```

## 2. NendIconModule の内容

### ○メソッド

#### ・ NADIconLoader 生成

**static void createNADIconLoader(char\* apiKey, char\* spotID);**

nend 管理画面で発行した該当アプリの広告枠の apiKey,spotID をセットします。

```
NendIconModule::createNADIconLoader("[管理画面より発行された apiKey]",
                                     "[管理画面より発行された spotID]");
```

#### ・ 位置を指定してアイコンを生成する

**static void createNADIconView(CCPoint pos);**

アイコン広告を表示する座標をセットします。

アイコンを一つ一つ個別に配置する場合に使用します。

```
NendIconModule::createNADIconView(Point(0, 75));
```

#### ・ 画面下部中央に並べてアイコンを生成する

**static void createNADIconViewBottom();**

アイコン広告を画面下部に表示します。 (**※標準では4個表示されます。)**

```
NendIconModule::createNADIconViewBottom();
```

#### ・ 画面上部中央に並べてアイコンを表示する

**static void createNADIconViewTop();**

アイコン広告を画面上部に表示します。 (**※標準では4個表示されます。)**

```
NendIconModule::createNADIconViewTop();
```

**※並べて表示するアイコンの数を変更する場合は、以下の数値を変更してください。**

- ・ iOS : NendIconModule.mm 内の define 値  
***#define ICON\_DISPLAY\_COUNT 4***
- ・ android : NendIconModule.java 内の変数  
***public static final int ICON\_DISPLAY\_COUNT = 4;***

- ・ **アイコンを非表示にする**

**static void hideNADIconView();**

表示中のアイコン広告を非表示にします。

```
NendIconModule::hideNADIconView();
```

- ・ **アイコンを再表示する**

**static void showNADIconView();**

非表示となったアイコン広告を再表示します。

```
NendIconModule::showNADIconView();
```

- ・ **広告の load**

**static void load();**

広告の定期ロードを開始します。

```
NendIconModule::load();
```

- ・ **広告の load 一時停止**

**static void pause();**

広告の定期ロードを中断します。

```
NendIconModule::pause();
```

- ・ **広告の load 再開**

**static void resume();**

広告の定期ロードを中断します。

```
NendIconModule::resume();
```

- ・ **リソースの解放**

**static void release();**

広告の表示で使用されたリソースを解放します。

```
NendIconModule::release();
```

### 3. Observer パターンによる通知（任意）

バナー広告及びアイコン広告からの通知は、Observer パターンにより行います。

Observer パターンの登録は行わなくても動作に障害はありません。

#### ・ロード完了（バナー広告）※iOS のみ

広告のロード完了時、

通知名"NEND\_NOTIFICATION\_FINISH\_LOAD"に登録されたメソッドを呼び出します。

例) ロード完了の Observer 登録

```
cocos2d::CCNotificationCenter::sharedNotificationCenter()->addObserver(this,
    callfuncO_selector([呼び出すメソッド]), "NEND_NOTIFICATION_FINISH_LOAD", NULL);
```

#### ・広告受信通知（バナー広告）

広告の受信成功時、

通知名"NEND\_NOTIFICATION\_RECEIVE\_AD"に登録されたメソッドを呼び出します。

例) 広告受信通知の Observer 登録

```
cocos2d::CCNotificationCenter::sharedNotificationCenter()->addObserver(this,
    callfuncO_selector([呼び出すメソッド]), "NEND_NOTIFICATION_RECEIVE_AD", NULL);
```

#### ・広告受信エラー通知（バナー広告）

広告の受信失敗時、

通知名"NEND\_NOTIFICATION\_FAIL\_TO\_RECEIVE\_AD"に登録されたメソッドを呼び出します。

例) 広告受信エラー通知の Observer 登録

```
cocos2d::CCNotificationCenter::sharedNotificationCenter()->addObserver(this,
    callfuncO_selector([呼び出すメソッド]), "NEND_NOTIFICATION_FAIL_TO_RECEIVE_AD", NULL);
```

#### ・広告バナークリック通知（バナー広告）

広告バナーをクリック時、

通知名"NEND\_NOTIFICATION\_CLICK\_AD"に登録されたメソッドを呼び出します。

例) 広告バナークリック通知の Observer 登録

```
cocos2d::CCNotificationCenter::sharedNotificationCenter()->addObserver(this,
    callfuncO_selector([呼び出すメソッド]), "NEND_NOTIFICATION_CLICK_AD", NULL);
```

### ・ロード完了（アイコン広告）※iOSのみ

広告のロード完了時、

通知名"NEND\_ICON\_NOTIFICATION\_FINISH\_LOAD"に登録されたメソッドを呼び出します。

例) ロード完了の Observer 登録

```
cocos2d::CCNotificationCenter::sharedNotificationCenter()->addObserver(this,
callfuncO_selector([呼び出すメソッド]), "NEND_ICON_NOTIFICATION_FINISH_LOAD", NULL);
```

### ・広告受信通知（アイコン広告）

広告の受信成功時、

通知名"NEND\_ICON\_NOTIFICATION\_RECEIVE\_AD"に登録されたメソッドを呼び出します。

例) 広告受信通知の Observer 登録

```
cocos2d::CCNotificationCenter::sharedNotificationCenter()->addObserver(this,
callfuncO_selector([呼び出すメソッド]), "NEND_ICON_NOTIFICATION_RECEIVE_AD", NULL);
```

### ・広告受信エラー通知（アイコン広告）

広告の受信失敗時、

通知名"NEND\_ICON\_NOTIFICATION\_FAIL\_TO\_RECEIVE\_AD"に登録されたメソッドを呼び出します。

例) 広告受信エラー通知の Observer 登録

```
cocos2d::CCNotificationCenter::sharedNotificationCenter()->addObserver(this,
callfuncO_selector([呼び出すメソッド]), "NEND_ICON_NOTIFICATION_FAIL_TO_RECEIVE_AD", NULL);
```

### ・広告クリック通知（アイコン広告）

広告バナーをクリック時、

通知名"NEND\_ICON\_NOTIFICATION\_CLICK\_AD"に登録されたメソッドを呼び出します。

例) 広告バナークリック通知の Observer 登録

```
cocos2d::CCNotificationCenter::sharedNotificationCenter()->addObserver(this,
callfuncO_selector([呼び出すメソッド]), "NEND_ICON_NOTIFICATION_CLICK_AD", NULL);
```

### ・ 広告の非表示と定期ロードの中断

広告を画面内に表示しない、もしくは画面遷移等で Scene 自体が表示されない場合には pause メッセージを送信、広告の定期ロードを中断し、広告を非表示状態にします。

例) 画面が隠れたら定期ロードを中断し、広告を非表示にする。

```
void HelloWorld::onExit(){
    CCLayer::onExit();

    NendModule::pause();
    NendModule::hideNADView();
    NendIconModule::pause();
    NendIconModule::hideNADIconView();
}
```

### ・ 広告の再表示と定期ロードの再開

広告を再び画面内に表示、または画面遷移等で Scene 自体を表示する場合には resume メッセージを送信、広告の定期ロードを再開し、広告を再表示します。

例) 画面が表示されたら定期ロードを再開し、広告を表示する。

```
void HelloWorld::onEnter(){
    CCLayer::onEnter();

    NendModule::resume();
    NendModule::showNADView();
    NendIconModule::resume();
    NendIconModule::showIconNADView();
}
```

以上で設定完了となります。

## ◆よくある質問

詳しくはメディアパートナー様向けヘルプをご覧ください

<https://www.nend.net/m/help/index/20>

## お問合せ先

nend - お問い合わせフォーム(<https://www.nend.net/inquiries/form/>)

※お問い合わせ時には、メディア登録名、ApiKey と spotID、SDK バージョン番号を情報として頂ければ幸いです。