
E-MOTE COCOS2D-X ドライバ マニュアル

このテキストは？

このテキストは、E-mote for Cocos2d-x ドライバのマニュアルです。

リリースノート

2015/10/28 リリースの E-mote3.58 より、配布形態が「cocos2d-x のプロジェクトまるごと」と言う形式から「生成したばかりの cocos2d-x プロジェクトに上書きして使う差分」形式へと変更になりました。また、これに伴い cocos2d-x ver 3.1～3.8 までの前バージョンで動作確認と正式対応を行いました。

特徴

E-mote for Cocos2d-x ドライバは、ドライバ本体と、cocos2d-x で生成したプロジェクトに上書きすることで機能するサンプル差分の 2 種類のファイル群で構成されています。

具体的には以下のフォルダが含まれます。

- sdk SDK 本体を含むフォルダ
- cocos31 cocos2dx v3.1～3.2 のプロジェクト用のサンプルプロジェクト差分
- cocos33 cocos2dx v3.3～3.4 のプロジェクト用のサンプルプロジェクト差分
- cocos35 cocos2dx v3.5～3.6 のプロジェクト用のサンプルプロジェクト差分
- cocos37 cocos2dx v3.7～3.8 のプロジェクト用のサンプルプロジェクト差分

サンプルの動かし方

サンプルプロジェクトを作成するには、以下の手順を取ります。

1. cocos2d-x で、「EmotePlayer」という名称で新規プロジェクトを作成する。
2. sdk フォルダの内容を上書きコピーする。
3. 対応するバージョンの cocosXX フォルダの内容を上書きコピーする。

あとは通常の cocos プロジェクトとしてビルドすることでサンプルの動作確認が出来ます。

Cocos2d-x では、v3.7 から CC_CONTENT_SCALE_FACTOR を参照するようにサンプルの実装が変更されています。それに合わせて、cocos37 フォルダのプロジェクト差分も、CC_CONTENT_SCALE_FACTOR を参照するように記述してあります。

SDK のバージョンアップの仕方

既存のプロジェクトに対して SDK をバージョンアップする際には、sdk フォルダの内容のみをプロジェクトに上書きコピーして下さい。

SDK フォルダの内訳

sdk フォルダのファイル構成は以下のようになっています。

- proj.win32/* Windows 用の E-mote ライブラリとヘッダファイル
- proj.android/* Android 用の E-mote ライブラリとヘッダファイル
- proj.ios_mac/* iOS 用の E-mote ライブラリとヘッダファイル
- Classes/CCEmote.{cpp,h} E-mote キャラを Node として表示可能にするクラス

COCOSXX 差分フォルダの内訳

cococsXX 差分フォルダの内訳は以下のようになっています。

- proj.win32/* Windows のプロジェクトファイル
- proj.android/* Android の mk ファイル
- proj.ios_mac/* iOS のプロジェクトファイル
- Resources/* サンプルの E-mote キャラのバイトファイル
- Classes/* E-mote キャラ表示サンプルプログラムソース

WINDOWS プロジェクトのセットアップ

sdk 内のファイルは以下のような構成になっています。

- proj.win32/iemote.h ヘッダファイル
- proj.win32/libemotedriver_cocos2dx.{lib,dll} リリースライブラリ stub, dll
- proj.win32/libemotedriver_cocos2dx_d.{lib,dll} デバッグライブラリ stub, dll
- proj.win32/libemotedriver_cocos2dx_v.{lib,dll} バージョナライズライブラリ stub, dll (冗長デバッグ出力あり)

ビルドを可能にするにはプロジェクトに対して以下のセットアップを行って下さい。

- \$(ProjectDir) を追加のインクルードパスに加える
- libemotedriver_cocos2dx.lib, libemotedriver_cocos2dx_d.lib (もしくは_v)をそれぞれ Release/Debug 構成の追加の依存ファイルに加える。
- ビルド後イベントに、ビルド後に dll をコピーするように設定する。

ANDROID プロジェクトのセットアップ

sdk 内のファイルは以下のような構成になっています。

- proj.android/iemote.h ヘッダファイル
- proj.android/{armeabi, armeabi-v7a, armeabi-v7a-neon}/* ライブラリ

ビルドを可能にするには、プロジェクトに以下のセットアップを行って下さい。

- proj.android/jni/Android.mk を開き include \$(BUILD_SHARED_LIBRARY)の前に以下の記述を追加

```
EMOTEDRIVER_DIR := $(APP_PROJECT_PATH)
```

```
LOCAL_LDLIBS := -L$(EMOTEDRIVER_DIR)/$(TARGET_ARCH_ABI) -lemotedriver
```

```
LOCAL_C_INCLUDES += $(EMOTEDRIVER_DIR)
```

- proj.android/jni/Application.mk を開き、APP_STL := c++_static の記述を
APP_STL := gnuSTL_static に変更。

Android 用の E-mote cocos2d-x SDK は現在 gnuSTL_static でビルドされているため、STL の設定を変更しないとリンク出来ません。この制限については、ご要望があれば別バージョンの C++ライブラリでビルドした版を同梱するなどの対応を行う予定です。

IOS プロジェクトのセットアップ

sdk 内のファイルは以下のような構成になっています。

- proj.ios_mac/ios/iemote.h ヘッダファイル
- proj.ios_mac/ios/{Debug-iphoneos, Debug-iphonesimulator, iphoneos, iphonesimulator, Verbose-iphoneos, Verbose-iphonesimulator} ライブラリ

ビルドを可能にするには、プロジェクトに以下のセットアップを行って下さい。

- -BuildSetting -> Linking -> OtherLinkerFlags に -lemotedriver を追加
- -BuildSetting -> Search Paths -> Header Search Paths に ./ios を追加
- -BuildSetting -> Search Paths -> Library Search Paths
に ./ios/\${PLATFORM_NAME} を追加

E-MOTE キャラの制御方法

E-mote キャラの制御は、Node を継承した CCEmote クラスで行います。

CCEmote は cocos2d::Node の派生クラスで、表示位置、拡大縮小、回転等 cocos2d::Sprite と同じように扱うことができます。

```
CCEmote *emote = CCEmote::create("emote_test.psb");  
  
emote->setPosition(256, 100);  
  
emote->setScale(0.5f);
```

通常の Emote キャラ制御用の関数は CCEmote::player() を経由して使用します。

こちらは E-mote の native インターフェースそのままなので、各機種のリファレンスマニュアルをそのまま参照して下さい。

```
emote->player()->StartWind(-100, 100, -10.0f, 0.0f, 1.0f);  
  
emote->player()->SetVariable("face_eye_LR", -30.0f, 15, 0.0f);
```