****

1. **GAFインストール**
2. **GAFとは**

GAFとはGeneric Animation Formatと言う意味です。フラッシュファイルからGAFファイルに交換したら、Cocos2dxとUnityも使う事ができます。

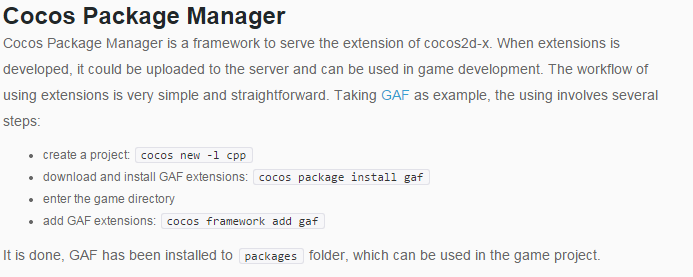
GAFのメリットには下のリンクを参照：<http://gafmedia.com/>

1. **サーポト（必要環境）**

|  |  |
| --- | --- |
| Cocos2dxバーション | Cocos2dx V3.2からGAFを表示できました。でも、Cocos2dxV3.2にはたくさんの事をまだサーポトしていないのでV3.3の以降、十分になりました。  C:\Users\chu\Desktop\キャプチャ3.PNG |
| Adobe Flash バージョン | フラッシュのバーションにとって関係ない |
| 制限事項 | 参照したリンク：   * <http://www.cocos2d-x.org/news/173> * <http://gafmedia.com/documentation/converter/limitations> * <http://gafmedia.com/faq#faq_2_1>   C:\Users\chu\Desktop\not_suporter.PNG |

Cocos2dx　V3.4のチャレンージログによると、V3.4以上GAFライブラリには自動的にCocos2dxプロジェクトに追加する事ができます。

参照したリンク：<http://www.cocos2d-x.org/news/416>



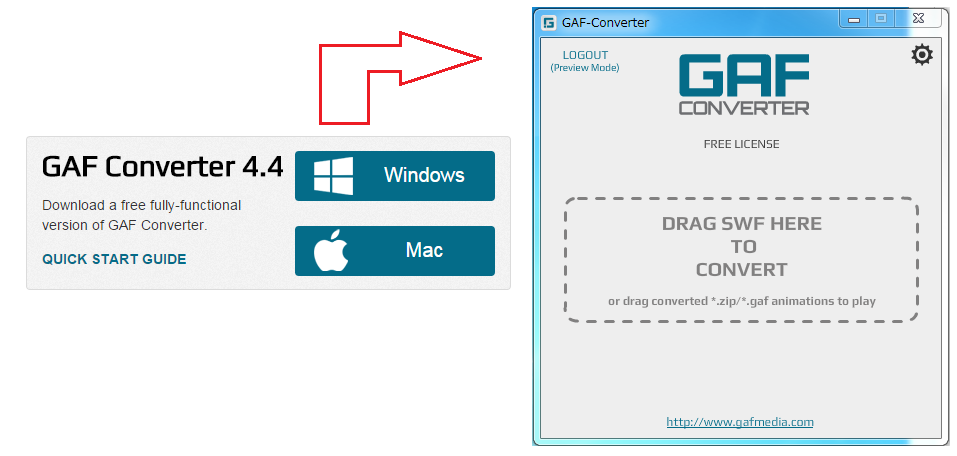
そして、GAFの発展している者によると、V3.3からCocos2dxと共に、アップデートするはずですので、LWFに比べてサーポトの方面にはいいと思いますね。

1. **GAFファイルに交換**

SWFファイルからGAFファイルに交換するの件、GAF開発者から交換ツールを紹介しました。

* **GAF Converter**: 下のリンク通りに、GAFファイルに交換アップリをダウンロードします。

Link: <http://gafmedia.com/downloads>



ダウンロードしたら、アカウントを登録しなくても使えます。でも、登録したほうがいいですね。

* **GAFライブラリ**：GAFファイルを使う為に、ゲームエンジンがGAFライブラリを使わないといけない

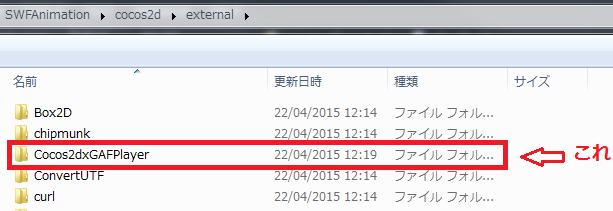
このリンク通りに、GAFファイルに交換アップリをダウンロードします：<http://gafmedia.com/downloads>



ダウンロードしたライブラリの名前は「Cocos2dxGAFPlayer-gaf-4.0」なので使いやすい為に「Cocos2dxGAFPlayer」に変更したほうがいいと思います。(変更しなくてもいいですけれど、綺麗になるように変更したほうがいいと思います)

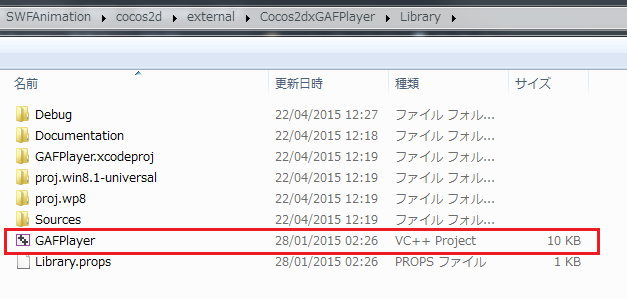
1. **GAFライブラリインストール: Visual Studioの対応とEclipseの対応**

説明が分かりやすい為に、Cocos2dxのSWFAnimationプロジェクトをやってみています。サンプルプロジェクトを作成した後に、プロジェクトの.../cocos2d/externalフォルーダにGAFライブラリに入れてください。

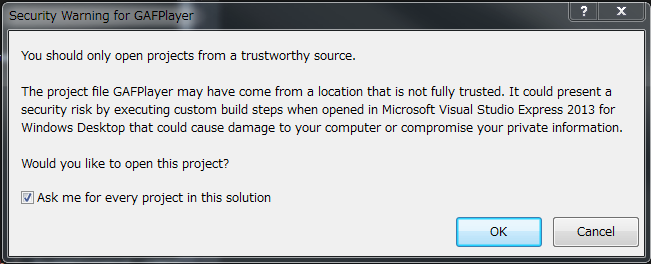


* **Visual Studioにライブラリを追加**
* **ステップ１**：Visual Studioにライブラリソースを追加

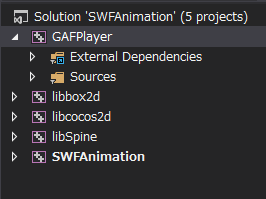
「SWFAnimation」プロジェクトを生成した後に、Visual Studio IDEでオプンして、Solutionを右マウスをクリクして、Add ->Existing Project を選択し、SWFAnimationプロジェクトでSWFAnimation /cocos2d/ Cocos2dxGAFPlayer/Libraryフォルーダの中にGAFPlayer.vcxprojファイルを選択します。



GAFライブラリはCocos2dxの最初ライブラリではないのでセキュリーティダイアログを確認しております。



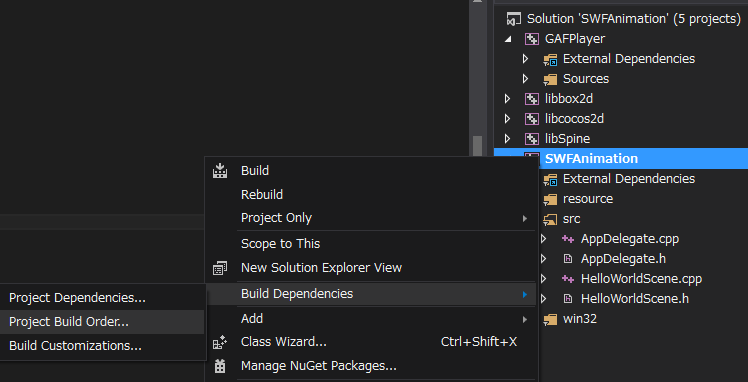
それからGAFPlayerライブラリを追加しました。



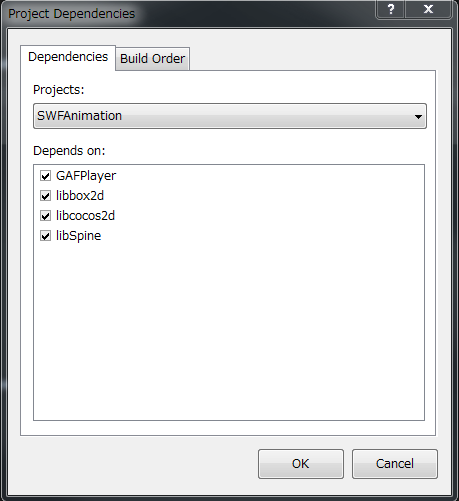
しかし、今までにGAFPlayerライブラリがまだ使わないんです。

* **ステップ2**：Visual StudioでGAFPlayerライブラリを宣言

Visual Studioでプロジェクトを右マウスにクリクして、Build Dependecies -> Project Dependecies...を選択します。

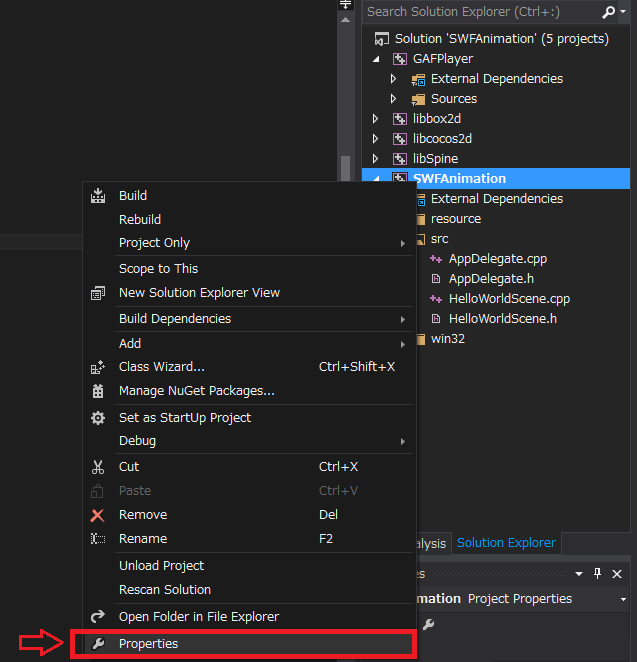


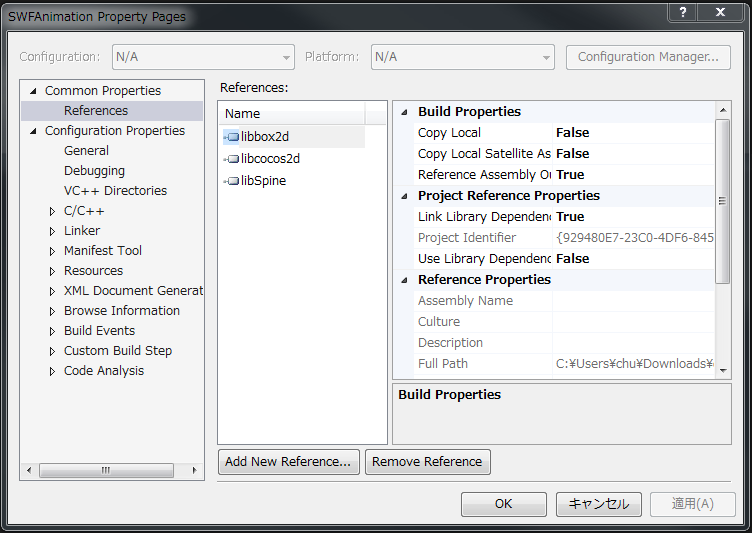
ダイアログを表示下後、GAFPlayerを選択します。



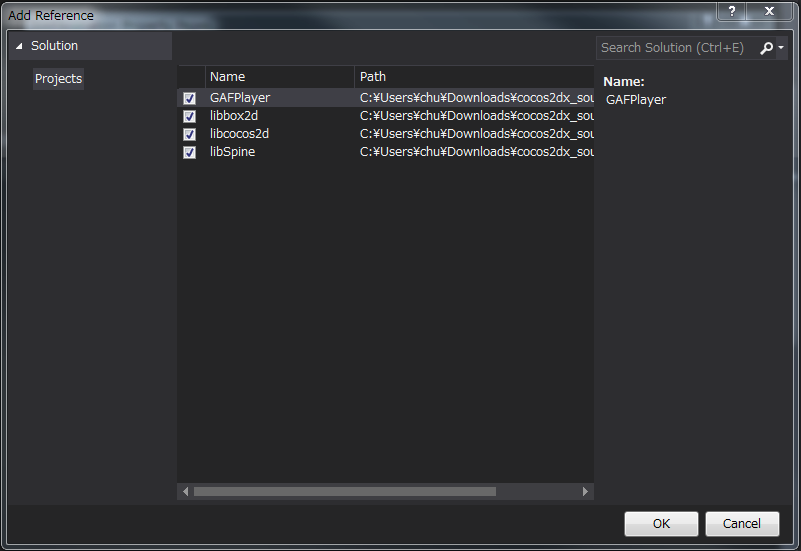
* **ステップ3**：Visual Studioのリファレンスを追加

Visual Studioでプロジェクトを右マウスにクリクして、Property-> Referencesを選択します。





Add New Reference...を選択します。



それからOKを引き続いて押した後、「Ctr + S 」でSolutionをSaveしてからGAFPlayerライブラリをCocos2dxで使う事ができます。(これは大切な操作です！)。

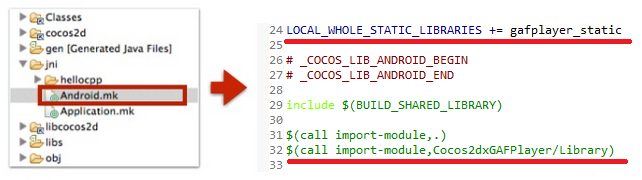
成功かどうか確認したい場合、プロジェクトのproj.win32フォルーダでVC++Projectファイルをオプンして下のようなコードがあれば成功になりました。

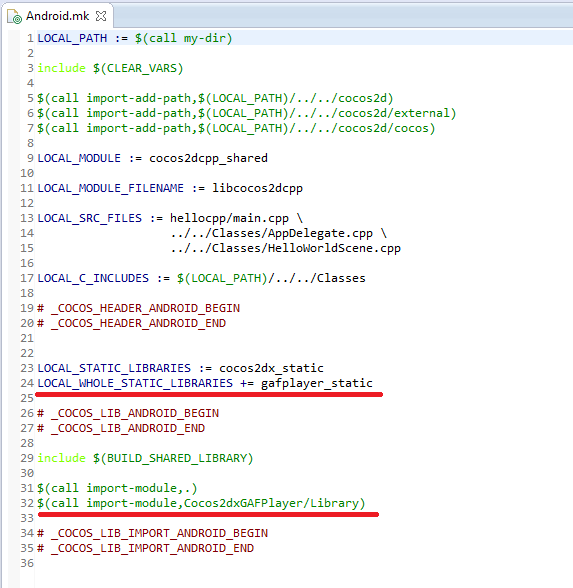
|  |
| --- |
| <ProjectReference Include="..\cocos2d\external\Cocos2dxGAFPlayer\Library\GAFPlayer.vcxproj">  <Project>{4e240210-2d26-4d4c-9329-e912fdac94cc}</Project>  </ProjectReference> |

* **Eclipseにライブラリを追加**

Eclipseでjni/Android.mkファイルを二つ宣言を追加してください。

|  |
| --- |
| LOCAL\_WHOLE\_STATIC\_LIBRARIES += gafplayer\_static  $(call import-module,Cocos2dxGAFPlayer/Library) |





1. **GAF利用**
2. **フッラシュマニュアル**

* タイムライン

|  |  |  |
| --- | --- | --- |
| フラッシュのスクリプト | Cocos2dxで適当なソース | 意味 |
| play() | gafObj->start(); | アニメーションをプレイーの命令 |
| stop() | gafObj->stop(); | アニメーションをストップの命令 |
| gotoAndPlay(frameIndex又はframeName) | gafObj->gotoAndPlay( uinit32\_t frameNumber);  又は  gagObj->gotoAndPlay(std::string& frameLabel); | 呼び出すフレームに移動してから、プレイーする |
| gotoAndStop(frameIndex又はframeName) | gafObj->gotoAndStop( uinit32\_t frameNumber);  又は  gagObj->gotoAndStop(std::string& frameLabel); | 呼び出すフレームに移動してから、ストップする |
| nextFrame | 適当な命令がないんですが、現在のフレームを確認したら、次と前のフレームを確認できます。  uint32\_t currentFrame = gafObj->getCurrentFrameIndex();  uint32\_t nextFrame = currentFrame + 1;  uint32\_t prevFrame = currentFrame – 1;  それから、目的によってフレームを使います。 | |
| prevFrame |

* イベント発生とボタンスクリプト

他の交換方法(LWFを使うかもしれない)が間違って、GAFにはイベントを登録するとき、フラッシュファイルを参照しないで、直接的にコードを使えます。

* フラッシュの座標系

GAF の再生エンジン上では、変換後の Flash のコンテンツはCocos2dxのオブジェクトの座標＝Flash コンテンツのステージの原点という座標系で扱われます。

|  |  |
| --- | --- |
| C:\Users\chu\Desktop\01.jpg | C:\Users\chu\Desktop\無題.png |

これはちょうど、Flash Player の中で swf ファイルをムークリップに読み込んで再生させた場合と同じ状態です。実際のコンテンツでは、次のようになります。



GAFに関して勉強した後、Cocos2dxでGAFオブジェクトの位置設定や回転や拡大・縮小などにとって、フラッシュの原点は中心として実行します。それで、GAFを使う時にフラッシュの原点をよく注意してください！（Cocos2dxのアンカーポイントの可能性があると言われる）

今まで、BOIのデザイナーが作ったフラッシュファイルによると二種類作る方法があるかもしれませんと思います。

|  |  |
| --- | --- |
| C:\Users\chu\Desktop\ヴィエン_Cocos2dx資料\stage2.png | C:\Users\chu\Desktop\ヴィエン_Cocos2dx資料\stage1.png |

Cocos2dxでのGAFオブジェクトを使う時にフラッシュファイルの原点によって位置の設定も違います。位置設定や回転など原点を設定する時にエラーが発生したら、原点を確認してください。

* **GAFの簡単な使い方ライブラリ宣言**

まず、Cocos2dxでGAFライブラリを使う為に、ヘッダファイルでライブラリを宣言します。

|  |
| --- |
| #include "external\Cocos2dxGAFPlayer\Library\Sources\GAF.h"  #include "external\Cocos2dxGAFPlayer\Library\Sources\GAFAsset.h"  #include "external\Cocos2dxGAFPlayer\Library\Sources\GAFObject.h" |

それから、簡単な例で説明としてCocos2dxでのGAFの使い方をご紹介しております。

* **基本の操作**

Cocos2dxでGAFオブジェクトはSpriteクラスから継承しているのでGAFオブジェクトはSpriteの属性もあります。

* GAFオブジェクトを生成する

|  |
| --- |
| auto gafAccet = GAFAsset::create("GAF/testGAF/testGAF.gaf", **nullptr**);  gafObj = gafAccet->createObject();  gafObj->start();　// 大切なコードです。ない場合は実施できない  addChild(gafObj); |

又は：

|  |
| --- |
| auto asset = GAFAsset::create("slot\_machine/slot\_machine.gaf", nullptr);  auto machine = asset->createObjectAndRun(true);　// true:繰り返す　false:１回  machine->setPosition(0, 1920);  addChild(machine, 1); |

* 位置を設定と取得する

|  |
| --- |
| gafObj->setPosition(visibleSize.width /2 , visibleSize.height/2);  log("GAF Default Position( %f , %f )", gafNode->getPosition().x, gafNode->getPosition().y); |

―→　Cocos2dxでGAFオブジェクトのディーフォルト位置はVec2(0,0)　です。

* アンカーポイントを設定と取得

|  |
| --- |
| // GAFオブジェクトのアンカーポイントを設定する  gafObj->setAnchorPoint(Vec2::ANCHOR\_MIDDLE);  // GAFオブジェクトのアンカーポイントを取得する  log("GAF APoint: (%f,%f)", gafObj->getAnchorPoint().x, gafObj->getAnchorPoint().y); |

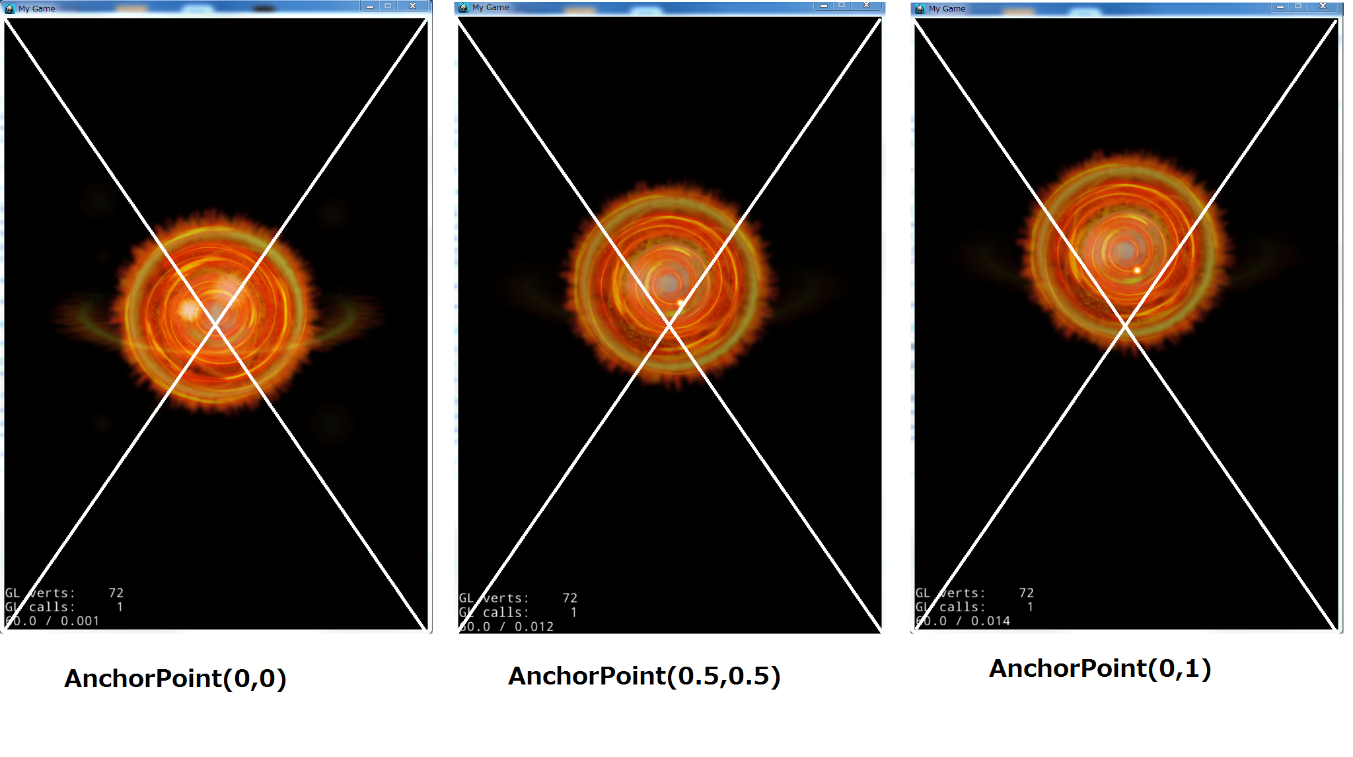
―→　GAFオブジェクトのディーフォルトアンカーポイントはVec2(0,0)　です。

LWFオブジェクトにはアンカーポイントが位置を影響しないのでGAFオブジェクトもチェックして行ってみます。

例1：ステージの原点の周りに穴メーションを作られる場合

|  |
| --- |
| auto gafAccet = GAFAsset::create("GAF/110002/110002.gaf", nullptr);  gafObj = gafAccet->createObject();  // アンカーポイントを設定  gafObj->setAnchorPoint(Vec2::ZERO); // ②アンカーポイントはVec2(0 , 0)に設定する  // gafObj->setAnchorPoint(ANCHOR\_MIDDLE); //　①アンカーポイントはVec2(0.5,0.5)に設定する  // gafObj->setAnchorPoint(Vec2:: ANCHOR\_TOP\_LEFT); // ③アンカーポイントはVec2(0,1)に設定する  gafObj->setPosition(Vec2(visibleSize.width / 2, visibleSize.height / 2));  gafObj->start();  addChild(gafObj); |

実装した後に、結果を比較します。



それではアンカーポイントはGAFオブジェクトの位置を影響することがあるのでアンカーポイントを使う時ご注意してください。

参照リンク：

<http://gafmedia.com/forum/viewtopic.php?f=3&t=21&start=10>

<http://gafmedia.com/forum/viewtopic.php?f=8&t=21>

* アニメーションの繰り返す管理

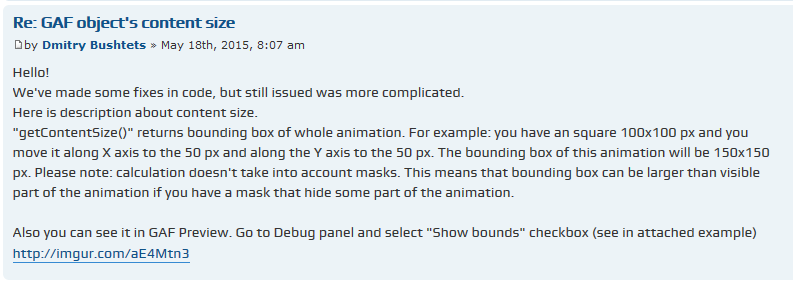
|  |
| --- |
| gafObj->setLooped(true); // 繰り返す：true,1回：false |

* GAFオブジェクトのコンテンツサイズを取得

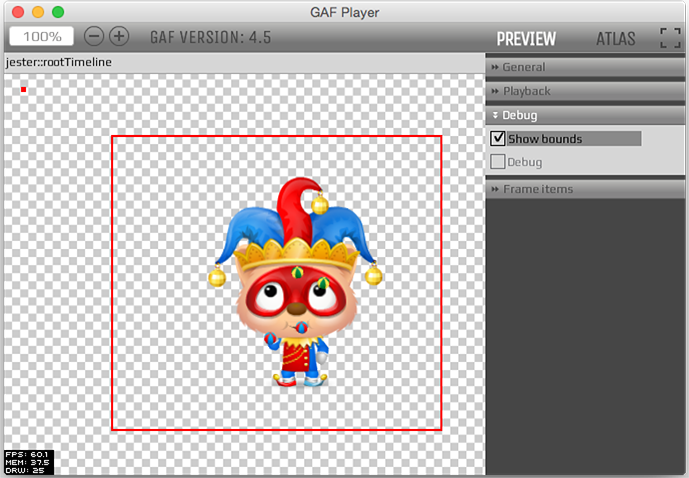
|  |
| --- |
| log("GAF ContentSize: (%f , %f )", gafObj->getContentSize().width, gafObj->getContentSize().height); |

でも、時々、フラッシュのコンテンツサイズとGAFオブジェクトのコンテンツサイズが違います。この問題にはGAFサイトのフォーラムで質問もありましたので参照してください。

参照したリンク：　<http://gafmedia.com/forum/viewtopic.php?f=8&t=292>



GAFConverterツールでGAFに交換する時、GAFオブジェクトのコンテンツサイズを検査することができます。



* GAFライブラリのドキュメント

GAF Libarary V2: <http://gafmedia.com/docs/cocos2dx/index.html>

GAF Library V3: <http://gafmedia.com/docs/cocos2dx_v3/index.html>

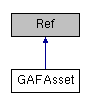
* GAFホームページのサンプルプロジェクト

このサンプルプロジェクトはGAFVer4.0を使っているので、すばらしい例だと思います。

Video: <http://gafmedia.com/examples/casino_slot_machine>

Source: <https://github.com/CatalystApps/GAFSlotMachine>

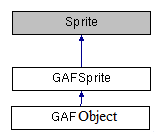
* よく使うクラスと関数の説明
* GAFAssetクラス[GAFAsset ← Ref]



Cocos2dxでGAFオブジェクトを生成する為に、GAFAsset オブジェクトを生成しないといけません。

|  |  |  |
| --- | --- | --- |
|  | Function | Note |
| GAFAsset\* | create(const std::string& gafFilePath, GAFTextureLoadDelegate\_t delegate); | GAFファイルパスからGAFAssetオブジェクトを生成する  例：  auto asset = GAFAsset::create("slot\_machine/slot\_machine.gaf", nullptr); |
| GAFAsset\* | create(const std::string& gafFilePath); |
| GAFAsset\* | createWithBundle(const std::string& zipfilePath, const std::string& entryFile, GAFTextureLoadDelegate\_t delegate); |
| static float | desiredCsf(); | desired content scale factor  例：  log("Desired content scale factor : %f " , gafAsset->desiredCsf()); |
| static void | setDesiredCsf (float csf); | sets desired content scale factor  例：  gafAsset->setDesiredCsf(2.0); |

* GAFObjectクラス　[GAFObject ←　GAFSprite　←　Sprite]



Cocos2dxでGAFオブジェクトって言うのはGAFObjectクラスのオブジェクトです。しかし、GAFVer4.0以降、GAFAnimatedObjectクラスはGAFObjectクラスを変わっていたので今のドキュメントがまだサーポトしていません。

変更を行っているバージョンのログ：

<https://github.com/CatalystApps/Cocos2dxGAFPlayer/commit/cebda25b29d19add453e8f8ec604dc8bbbb2cdb0>

|  |  |  |
| --- | --- | --- |
| void | start() | 呼び出したごろからアニメーションをスタートする |
| void | stop() | 呼び出すごろからアニメーションを禁止する |
| void | pauseAnimation(); | メソッドを呼び出すとき、実行しているフレームをポーズする |
| void | resumeAnimation(); | メソッドを呼び出すとき、ポーズしたフレームをレジュームする |
| void | setLooped(bool looped, bool recursive = false) | 繰り返す可能性を設定する。  引数はtrueの件、繰り返しになり、falseの件、1回だけになる |
| void | setReversed(bool reversed , bool fromCurrentFrame = true); | 呼び出したフレームから転倒に実行する。 |
| bool | isReversed() | リータンの結果によって転倒するかどうかチェックする  trueの件、転倒した。  falseの件、転倒しなかった |
| bool | isDone() | アニメーションが終るかどうかチェックスルメソッド  繰り返すの件、リータンの価値はfalse  1回の件、リータン価値はtrue |
| bool | isLooped() | アニメーションを繰り返すかどうかチェックするメソッド  繰り返すの件、リータンの価値はtrue  1回の件、リータン価値はfalse |
| bool | gotoAndStop(const std::string& frameLabel); | 呼び出すフレームに移動してから、禁止する |
| bool | gotoAndStop(uint32\_t frameNumber); |
| bool | gotoAndPlay(const std::string& frameLabel) | 呼び出すフレームに移動してから、プレイーする |
| bool | gotoAndPlay(uint32\_t frameNumber) |
| uint32\_t | getStartFrame(const std::string& frameLabel); | フレームのラベルにより、スタートフレームのインディークトを取得する |
| uint32\_t | getEndFrame(const std::string& frameLabel); | フレームのラベルにより、エンドフレームのインディークトを取得する |
| bool | playSequence(const std::string& name, bool looped, bool resume = true); | 適当な名前によってシークエンスを実行する  @param name シークエンスの名前  @param looped 繰り返すの設定：trueなら繰り返す、falseなら1回  @param resume 最初のフレームを表示してから、trueならアニメーションを引く続いて実行する、falseならすぐに禁止する |
| void | clearSequence(); | 実行しているシークエンスを禁止する |
| void | setAnimationFinishedPlayDelegate(GAFAnimationFinishedPlayDelegate\_t delegate) | アニメーションを終ったら、このメッソドを呼び出す。繰り返す場合にはメソッドを呼び出さない |
| void | setAnimationStartedNextLoopDelegate(GAFAnimationStartedNextLoopDelegate\_t delegate) | 繰り返すの場合は毎回、アニメーションを終るたびにこのメッソドを呼び出す  繰り返さない場合はこのメッソドを呼び出さない |
| void | setFramePlayedDelegate | アニメーションを実装すると共にこのメッソドを呼び出す |
| Rect | getBoundingBoxForCurrentFrame(); | 実行しているフレームの大きさの矩形 |

* GAFオブジェクトをやってみる

ヘッダファイルで変数とメッソドを宣言する。

|  |
| --- |
| // 変数の宣言  GAFObject\* gafObj; // GAFオブジェクトの宣言  int loopCount = 0;  int nextLoopCount = 0;  // 関数の宣言  void onFramePlayer(GAFObject\* ,uint32\_t);  void animationStartedNextLoop(GAFObject\*);  void animationFinishedPlay(GAFObject\*);  void animationSequence(GAFObject\*, const std::string); |

実装ファイルでメソッドを宣言します。

|  |
| --- |
| [initの中断]  // アニメーションのフレーム数を取得  uint32\_t frameCount = gafObj->getTotalFrameCount();  log("Frame Count = %d ", frameCount);  // アニメーションを実装すると共にこのメッソドを呼び出す  gafObj->setFramePlayedDelegate(CC\_CALLBACK\_2(HelloWorld::onFramePlayer , **this**));  // 繰り返すの場合は毎回、アニメーションを終るたびにこのメッソドを呼び出す  // 繰り返さない場合はこのメッソドを呼び出さない  gafObj->setAnimationStartedNextLoopDelegate(CC\_CALLBACK\_1(HelloWorld::animationStartedNextLoop, this));  // アニメーションを終ったら、このメッソドを呼び出す。繰り返す場合には呼び出さない  gafObj->setAnimationFinishedPlayDelegate(CC\_CALLBACK\_1(HelloWorld::animationFinishedPlay , this)); |

メソッドを受け取った後に、処理します。

|  |
| --- |
| // このメッソドはアニメーションを実装すると共に呼び出される。アグメントのframeは実施ているフレームインディクト(frameIndex)  void HelloWorld::onFramePlayer(GAFObject\* gafObject , uint32\_t frame)  {  log("%d", frame);  　　if (frame == gafObj->getTotalFrameCount() - 1)  　　{  loopCount++;  　　　 // 4回の後、アニメーションをストップすると画面の上に削除する  **if** (loopCount == 4)  {  gafObj->stop();  gafObj->removeFromParentAndCleanup(**true**);  Size visibleSize = Director::*getInstance*()->getVisibleSize();    　　　// 他のアニメーションを実施  **auto** gafAccet1 = GAFAsset::create("GAF/110002/110002.gaf", **nullptr**);  **auto** gafObj1 = gafAccet1->createObject();  gafObj1->setAnchorPoint(Vec2::*ZERO*);  gafObj1->setPosition(Vec2(200, visibleSize.height / 2));  gafObj1->setLooped(**true**);  gafObj1->start();  addChild(gafObj1);  }  　　}  }  void HelloWorld::animationStartedNextLoop(GAFObject\* gafObject)  {  nextLoopCount++;  log("Next Loop: %d ", nextLoopCount);  }  void HelloWorld::animationFinishedPlay(GAFObject\* gafObject)  {  log("Animation Finish");  } |