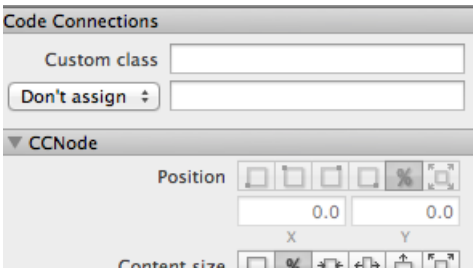
CocosBuilder and Cocos2D-X HelloWorld example

COCOSBUILDER AND COCOS2D-X HELLOWORLD EXAMPLE

After some experience with CocosBuilder and Cocos2d-x we present you the simplest CocosBuilder and Cocos2d-x use case!
For this tutorial we'll use the latest Cocos2d-x version (2.1) and we assume you have a working simple CocosBuilder file named "testLayer.ccbi" with no code connections (custom classes and variable assignments).



We'll start coding a standard CCScene that will hold the CCLayer/CCNode that the CocosBuilder file contains:

```
#ifndef __CCBTESTSCENE_H__
#define __CCBTESTSCENE_H__

#include "cocos2d.h"

class CCBTestScene : public cocos2d::CCScene
{
private:    // methods
        CCBTestScene()  { }
        ~CCBTestScene() { }

        bool init();

public:     // methods
        CREATE_FUNC(CCBTestScene);
};

#endif
```

Nothing special so far, the scene will be created calling *CCBTestScene::create()*, which will call *CCBTestScene*'s

(empty) constructor and then the *init()* method where we'll load the CocosBuilder file:

```
#include "CCBTestScene.h"
#include "cocos-ext.h"              // including CocosBuilder implementation

using namespace cocos2d;
using namespace cocos2d::extension;  // We need classes from this namespace

bool CCBTestScene::init()
{
    if (!CCScene::init())
        return false;

    // Create a default CCNodeLoaderLibrary. As we won't be using
    // code connections in this tutorial don't worry about it.
    CCNodeLoaderLibrary* nodeLoaderLibrary;
    nodeLoaderLibrary = CCNodeLoaderLibrary::newDefaultCCNodeLoaderLibrary();

    // Create a new CCBReader with a default CCNodeLoaderLibrary
    // This can take a lot of parameters to use code connections and more
    CCBReader* ccbReader = new CCBReader(nodeLoaderLibrary);

    // Load the main node from the CocosBuilder file
    CCNode* node = ccbReader->readNodeGraphFromFile("testLayer.ccbi");

    this->addChild(node); // Add the loaded node to the scene (this)

    // As nobody called ccbReader->autoRelease(), returning now would cause
    // a memory leak. We can call autoRelease or delete it ourselves.
    delete ccbReader;

    return true;
}
```

That should be it! If you need to use code connections or any other advanced feature from CocosBuilder this code needs some changes:

- **Line 16**: if you're using custom classes they need to be registered into the *nodeLoaderLibrary* that we've just created (that's it's propose after all, know how a type of node is loaded).
- **Line 19**: as said the CCBReader constructor can receive a lot of parameters (which equal *NULL* by default). If you want to use member variable assignment, menu selectors or animation events this is the place to add a pointer to a delegate that handles them.
- **Line 22**: when you load a file you can assign a "owner" for their nodes. This allows you to select "owner" for a code connection's delegate in CocosBuilder instead of using the document root (top most node of the file's node herarchy).

If you need some insight about code connections you can either take a look at our previous CocosBuilder/Cocos2d-x tutorial or dive into the tests code that comes with the framework. It that's not enough ask and we may end up doing some more tutorials about it!