

```
    action.tag = kActionAttack;  
    [sprite runAction:action];  
  
(void) gotHit:(id)sender data:(void*)data  
  
// verify that we can still can be hit  
Character* attacker = (Character*)data;
```

Nat Weiss

The iPhone

RPG

Engine



THE IPHONE RPG ENGINE

NAT WEISS



Bullets in a Burning Box



Copyright © 2012 Bullets in a Burning Box, Inc.

All rights reserved.
www.iPhoneGameKit.com

The burning box logo is a trademark of Bullets in a Burning Box, Inc.

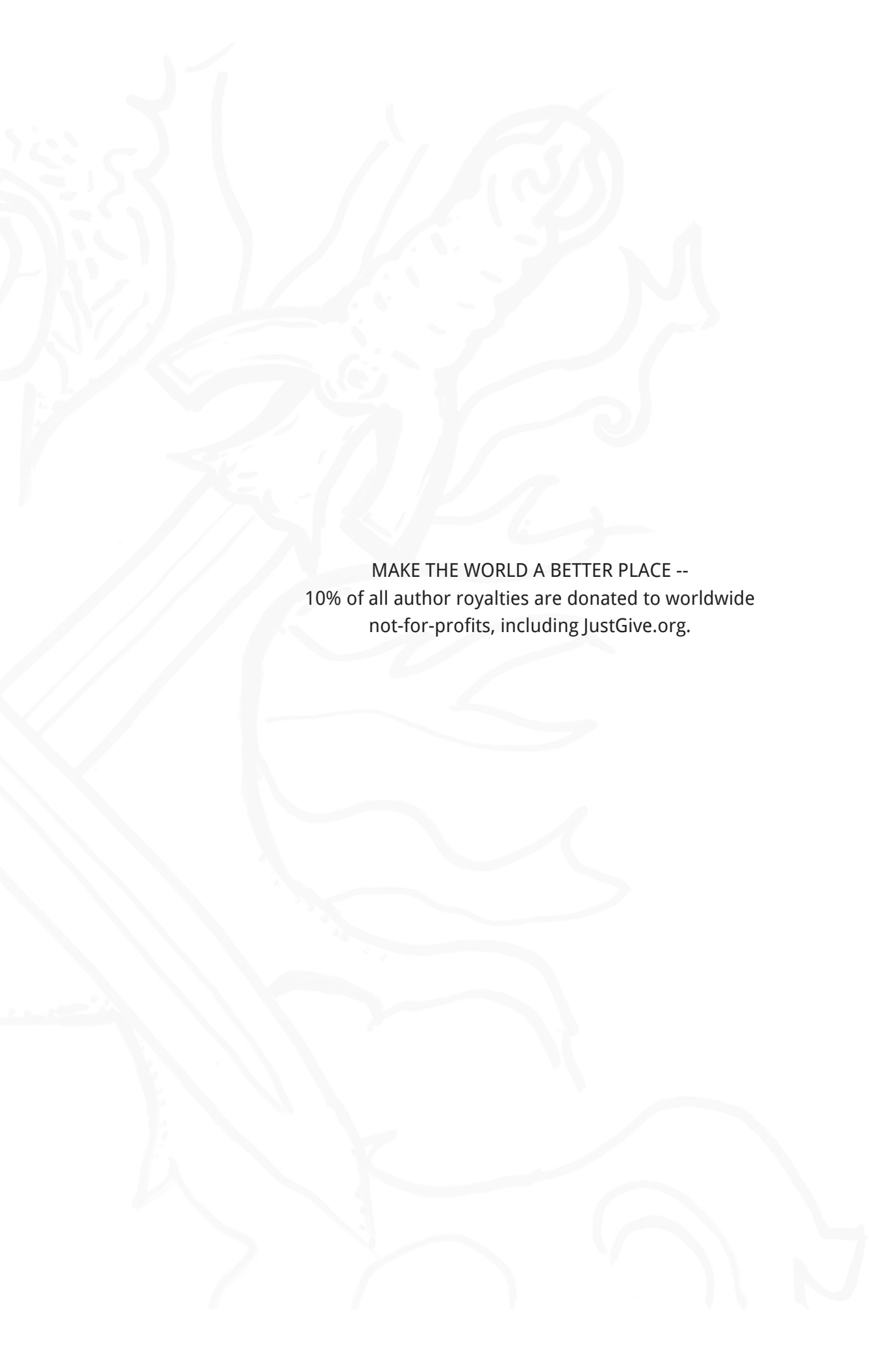
iPhone, iPad, iPod touch, Xcode, Mac, and Mac OS are trademarks of Apple Inc., registered in the U.S. and other countries. Bullets in a Burning Box Inc. is an independent corporation and has not been authorized, sponsored, or otherwise approved by Apple Inc.

Cocos2D is a trademark of Ricardo Quesada, registered in the U.S.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Bullets in a Burning Box shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

Printed in the United States of America.

First edition.



MAKE THE WORLD A BETTER PLACE --
10% of all author royalties are donated to worldwide
not-for-profits, including JustGive.org.

C O N T E N T S

Fantasize Your World

Welcome!	10
What You Should Already Know	10
Get Excited About Your World	11
Sketch It & They Will Come	12
The Making of an RPG	13
A More Flexible RPG Engine	15
Running the RPG Engine	16

Make Your Own Levels

Starting a Tile Map Level	10
Orthogonal vs Isometric	11
Adding a Tileset Image	12
The Tilesets Palette	12
The Obstacles Layer	14
The Objects Layer	15
Inserting a New Object	15
Creating the Player Object	16
Modifying an Object's Attributes	17
Saving Levels In The Right Place	18
Adding Files to the Xcode Project	19
Running the Engine	21
Other Layer Properties	22
More Objects Classes	23

Warp Attributes	24
Other Object Attributes	25
List of All Objects	26
Remember Base64 Gzip	27

RPG Engine Overview

RPG Engine Overview	28
The Camera Layer	29
The Level Layer	29
Level Objects	30
Level Object Attributes	31
Level Update	31
The Warp Object	32
The HUD Layer	32
The Character Menu	33
The Victory Menu	33
Broadly Modifying the Engine	33

Lofty Resolutions

The High Resolution Challenge	36
Multiple Projects?	37
A Single, Universal Project	38
The Scale Method	39
Choosing SD vs HD Resources	40
Creating High Resolution Menus	40
The Principle: Always Go Hi-Res	41

HD TMX Maps	41
HD Sprites	42
Base SDK	42
Deployment Target	43
Setting the Game's Icon	44
Concluding Thoughts on Hi-res	44

Start Your Game

Develop a Strategy	46
Sketch Your Game	46
Simplify	47
List & Gather Assets	47
Copy & Rename the Template	48
Applying New Art	50
Applying New Attributes	50
Feedback	51
About the Author	51

FANTASIZE YOUR WORLD

“The artist must create a spark before he can make a fire and before art is born, the artist must be ready to be consumed by the fire of his own creation.”

-- Auguste Rodin (1840-1917), French sculptor, considered the progenitor of modern sculpture



WELCOME!

Welcome to the iPhone Game Kit's role-playing game (RPG) engine starter kit! This book is a companion to the source code and documentation included in the download you purchased. You can use this book as a guide to understanding the broad overview of how an RPG, action, adventure or even a basic real-time strategy (RTS) game is programmed. Along the way you'll learn how the levels are built and how all the characters and objects interact.

WHAT YOU SHOULD ALREADY KNOW

This book asserts that you already know the basics of Objective C, Cocos2D, Xcode and the iOS software development kit (SDK). If you're brand new to Cocos2D, need help with Objective C or still learning Xcode, then you'll need to catch up before this book makes complete sense.

If that's the case, then check out our [Checkers Game Starter Kit](#). It will bring you up to speed, as it's intended to teach you the fundamentals of Xcode, Cocos2d and Objective C even if you have zero programming experience. It includes source code, an ebook, documentation and artwork similar to the format of this starter kit.

GET EXCITED ABOUT YOUR WORLD

On to the good stuff! Do you like to watch fantasy and science fiction movies?

Have you ever been pressured by friends to get off your butt and stop reading a fantasy novel all weekend?

These are good signs. It means you have the potential to be a world creator because only world creators dive headfirst into creative worlds. Fantasy and sci-fi are so creative because essentially *anything* is possible, and not just in the “you can do it if you put your mind to it” kind of way. We’re talking flying cities and the borg.

Yep, you’re a world creator and one of the coolest ways to express that potential inside you is to make a video game and allow other people to experience your worlds first hand.

A hundred years ago all we had were books. Now we have **games**. This is huge!



Some potential hero characters

Imagine the characters you can use to populate *your* world. What will the hero look like? How many heroes will there be?

What about enemies? Friends? Imagine the *story* they can tell!



Potential enemy and friend characters

And what kind of system will you use to govern the characters’ attributes? Will they have strength, dexterity, magic, skills?



Quexlor's character attributes

SKETCH IT & THEY WILL COME

Time to take the first step. Grab a pen, some paper and invest a chunk of time into sketching out your dream game. Draw a map, list your heroes & villains, write a brief plot synopsis.

It doesn't have to be long. It doesn't require massive detail. You might only spend twenty minutes sketching out a broad overview. You might spend the entire time drawing characters. It doesn't matter so much what you've sketched, it matters that you've put a bit of your ideas on to paper.

This step is critical. Writing down your idea frees up that space in your mind, allowing you to proceed to make it reality without a distraction.

That's right. Writing down your ideas helps you become more focused.

When you have finished, make sure to **keep your world a secret** until you've got a playable game of your own. That's right. Don't tell anybody that you are going to make the game. Let them *see* it when you have something to show.

Why keep it a secret? If you were to go and tell your friends you're about to make such and such a game with such and such heroes, you would psychologically consider the game done and likely never work on it. Really. So save your talking for when the game is ready to be talked about.

THE MAKING OF AN RPG

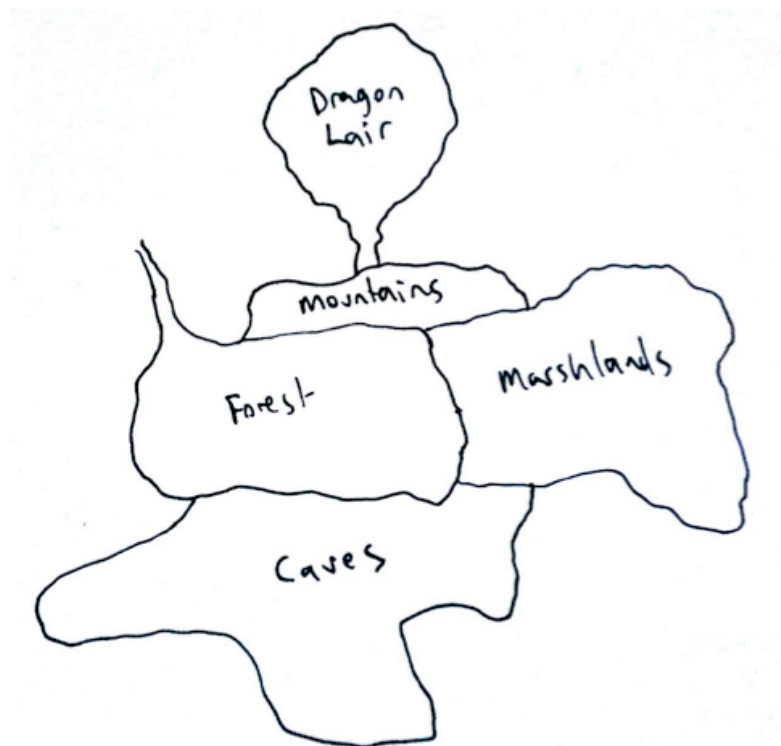
The great news is that you now have an RPG engine at your disposal and making your own world become reality is practically at your fingertips. All you have to do is alter a few property lists and throw in some new artwork and your game world can be considered playable.

Before we get to altering those property lists, how about a little story about the making of an RPG?

During the winter of 2011, game developers like you and me got together through our online [forum](#) and agreed to make the game *Quexlor: Lands of Fate*.

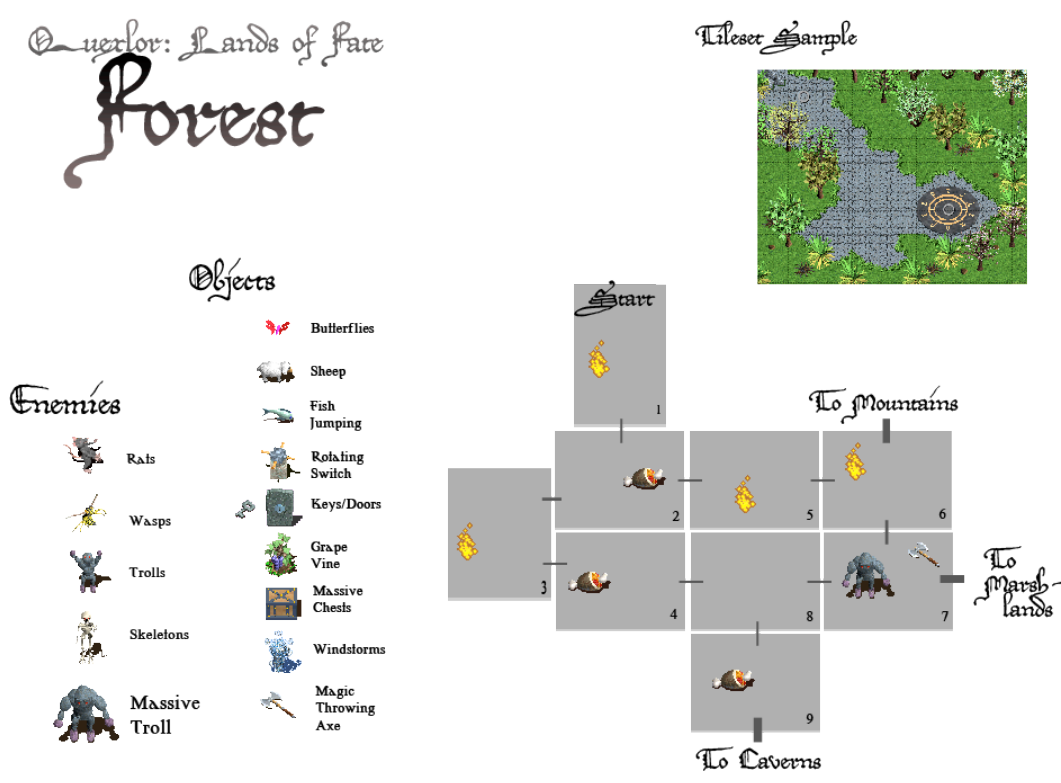
We started with a game design document that outlined how everybody could help out either making levels (easy), sprite sheets (easy), sound effects & music (easy) or artificial intelligence (AI) (difficult). We used a public Google spreadsheet where everyone could sign up for the areas in which they wanted to contribute.

We took a look at the tile artwork available and decided on four main areas: forest, marshlands, caves and mountains.



An early sketch of the overall game world for Quexlor: Lands of Fate

We made detailed maps for each area that depicted the individual levels, the connections between the levels, enemies, key items and bosses.



The Forest area level design for Quexlor: Lands of Fate

With this organization in place, it became possible for people to submit a single level, a single sprite sheet or a snippet of AI code.



The Forest01.tmx level of Quexlor: Lands of Fate

Lots of people ended up contributing and over the course of two months all the submissions had come in. It was time to piece everything together and play test like crazy. After another month of polishing our game was complete.

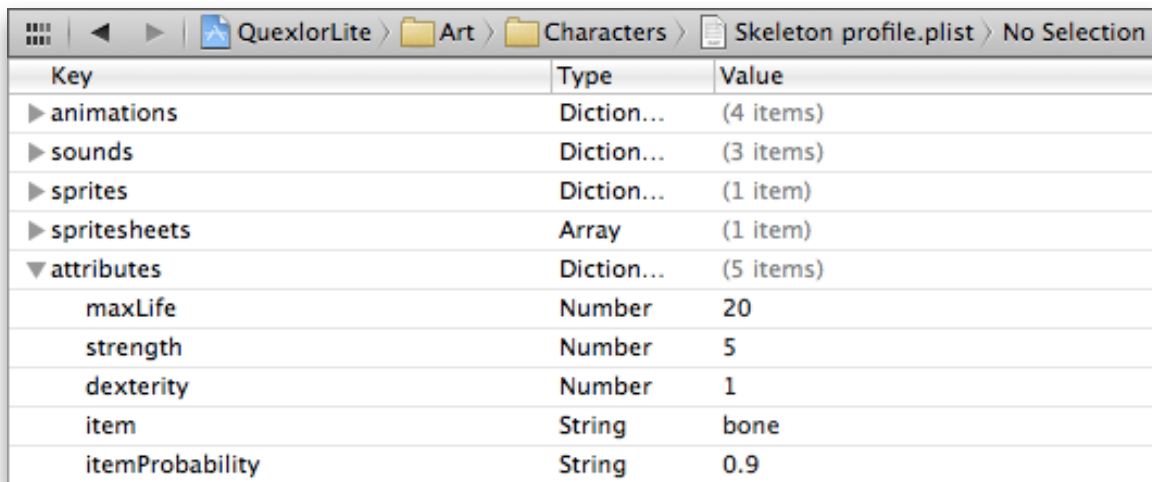
We released the game on the app store for free and since then we've consistently gotten above 500 downloads a week!

A MORE FLEXIBLE RPG ENGINE

After all that time invested making an RPG, it seemed natural to take the engine and make it more flexible and robust. We wanted to be able to quickly make a new game without changing the code very much.

We went back to work simplifying and modularizing the code until we came up with this new RPG engine. One of the biggest enhancements we made is to give each object in the game, whether it be an enemy or an item, a property list of default attributes like strength, maximum life force and scale. It then became possible to alter the game's system quickly and easily without changing any code.

The coolest part about attributes is that they can be overridden in the level files. For example, you can give a skeleton a scale of 2.0 in a level file's object properties and when you play that level the skeleton will appear twice the regular size.



The screenshot shows a plist editor window with the following path: QuexlorLite > Art > Characters > Skeleton profile.plist > No Selection. The main content is a table with three columns: Key, Type, and Value.

Key	Type	Value
▶ animations	Diction...	(4 items)
▶ sounds	Diction...	(3 items)
▶ sprites	Diction...	(1 item)
▶ spritesheets	Array	(1 item)
▼ attributes	Diction...	(5 items)
maxLife	Number	20
strength	Number	5
dexterity	Number	1
item	String	bone
itemProbability	String	0.9

The Skeleton character's attributes property list

RUNNING THE RPG ENGINE

To illustrate just how easy it is to alter the game without changing any code, let's modify the skeleton's property list and see what happens.

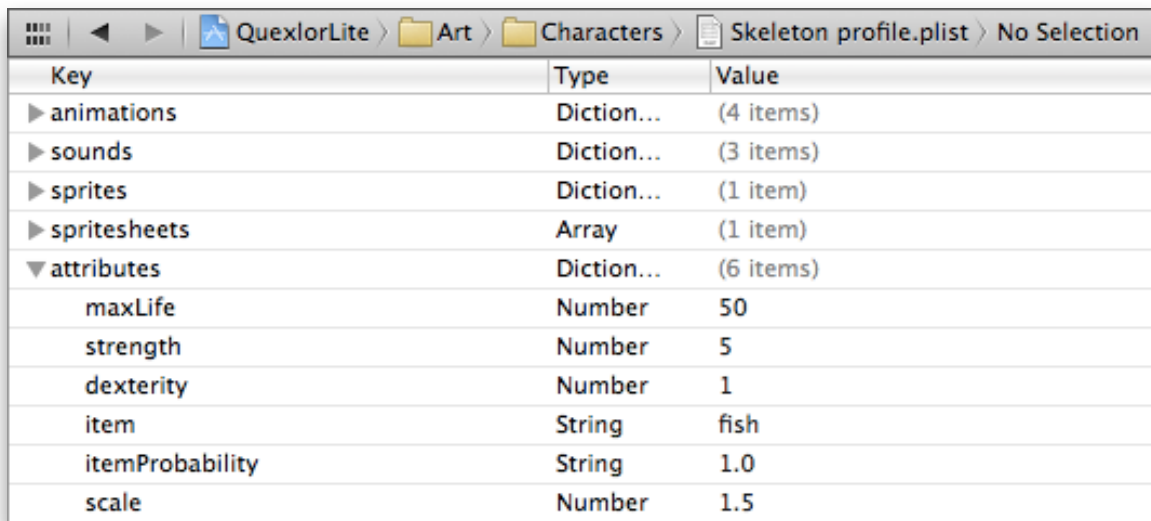
First, open up the **QuexlorLite** Xcode project in your download's **Quexlor** directory and disclose the files in the Project navigator. You will see the highest-level code files like **Quexlor.h** and **Quexlor.m**. These files make up the *specifics* for the Quexlor game. The more general code files can be found in the **Foundation** folder.

The actual RPG engine code is in the **Foundation > RPG Engine** folder. These files are generic to any RPG and you theoretically shouldn't have to change them very much, but feel free to get in there and edit whatever you'd like.

The code in the **Foundation > iPhone Game Kit** folder, particularly **main.m**, **KITApp.h** and **KITApp.m** are what launch the app and pass off control to **Quexlor.m's startApp** method.

The artwork and property lists can be found in the **Art** folder. Open up the **Art > Characters** folder, click on **Skeleton profile.plist** and open up the **attributes** dictionary. Change the skeleton's **maxLife** to **50**, the item to **fish** and the **itemProbability** to **1.0**. Give him one more **number** property called **scale** with the value **1.5**.

Your **Skeleton profile.plist** will now look like this:

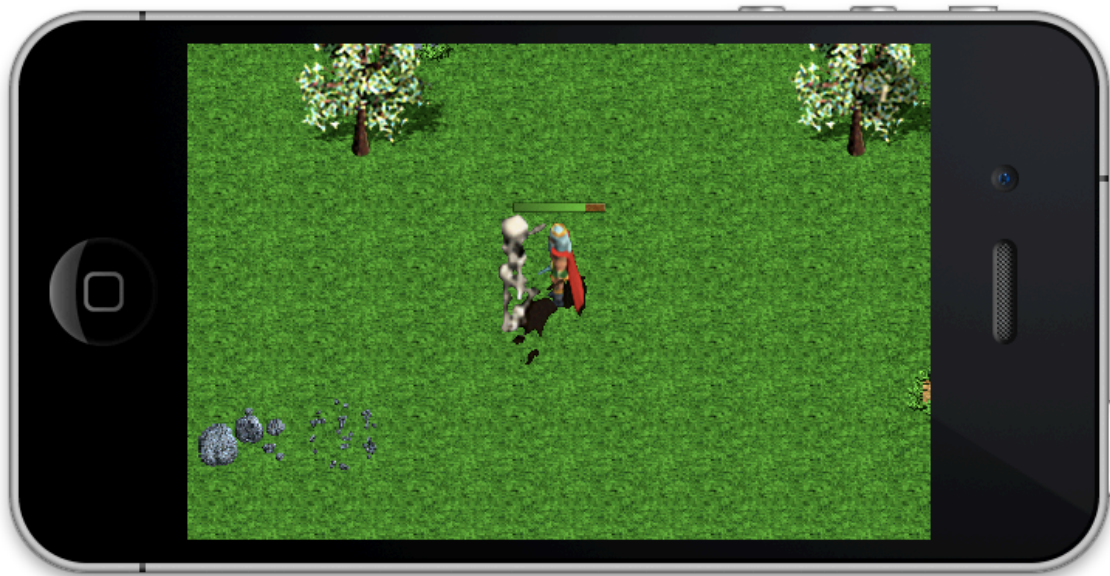


Key	Type	Value
▶ animations	Diction...	(4 items)
▶ sounds	Diction...	(3 items)
▶ sprites	Diction...	(1 item)
▶ spritesheets	Array	(1 item)
▼ attributes	Diction...	(6 items)
maxLife	Number	50
strength	Number	5
dexterity	Number	1
item	String	fish
itemProbability	String	1.0
scale	Number	1.5

Skeleton profile.plist with modified attributes

Important note: if you have already run the game in the simulator prior to reading this chapter, be sure to delete the app from the simulator so that the new default skeleton attributes will apply.

Now, run the game and walk the main barbarian hero down and to the right. You will come across a skeleton, he'll be big and he'll take quite a few swings to bring down. When he does die, guaranteed, he'll drop a fish!



An extra-big, extra-strong, fish-laden skeleton

Chapter Ideas:

- Skim over the code files in the Quexlor project to get familiar with everything.
- Use the Documentation folder to browse the included class reference.
- Sketch out a world map for your game. List out heroes & villains and write a short plot synopsis.



GET IT ALL

Thanks for checking out this sample! :)

Grab the rest of the book, the source code, the art package, and everything else at iphonegamekit.com/game-source-code/cocos2d-iphone-rpg-engine/

Enjoy making games!

-- Nat Weiss