Project name: AndEngine(<http://www.andengine.org/>)

***Introduction***

AndEngine is a framework for developing 2D games for Android.   
AndEngine is a very interesting project from point of view of modeling.  
There are a few elements in the project that contradicts the classic modeling   
methodology even the object oriented design.  
The funder of the project don’t allow (as a principle) to add code documentation. His argument is to obligate the developers to think of the design before writing code and write readable clean code, citation from [Droid blog](http://droid-blog.net/2012/12/28/introduction-to-andengine-getting-the-code/) “Nicolas tries to write the code in a way that it’s documenting itself”. Moreover, it force the developers to write samples.  
From other point of view, there is a lot of tutorials and two books about developing games using AndEngine.   
To improve the simplicity and the efficiency, except using the obvious OOP model he uses the event driven design.

***Overview***

AndEngine is a popular and "easy" to use game framework for Android.

AndEngine is currently available in two versions: GLES1 and GLES2.

The GLES1 version supports OpenGL ES 1.x. (99% of Android devices can run a game built using this version)

And GLES2 supports OpenGL ES 2.0. This branch is actively being worked on, and nearly 93% of current Android devices can run a game that’s been made with this branch.

Popular games:

* Bunny Brawler
* Bunny Shooter
* Constant C
* Cow and Pig go Home
* Draw and Rite
* Fireworks
* Greedy Spiders
* Etc.

**Similar projects:**

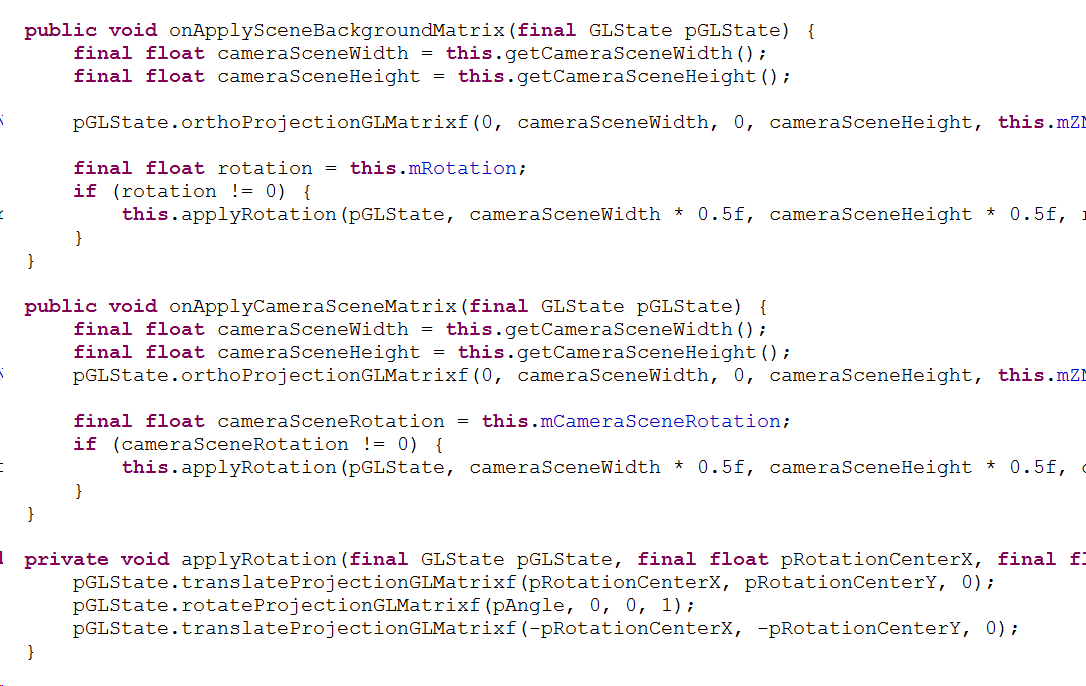
* **Unity 3D**
  + 2D/3D cross platform (iOS, Android, PC)
  + PhysX physics engine, C#, javascript or Boo scripting language
  + Multiplayer and network support
* **Cocos2D**
  + Cross platform (Android, iPhone, Win32) 2D engine
  + C++ based engine
  + Full source code
* **Project Anarchy by Havok**
  + FREE Cross-Platform Engine and Toolkit for Mobile Game Developers
  + Develop and release titles on iOS, Android and Tizen for free.
  + Extendible C++ plugin based architecture
* **GameMaker by YoYo Games**
  + 2D Cross platform game engine
  + support for iOS, Android, HTML5, Mac OS, Windows 8, Ubuntu and Windows Phone 8
* **A lot of more**
  + <https://software.intel.com/en-us/android/blogs/2012/03/13/game-engines-for-android>
  + <http://en.wikipedia.org/wiki/List_of_game_engines>
  + <http://gamedev.stackexchange.com/questions/19219/is-there-a-free-and-open-source-3d-engine-for-android>

**About Documentation**

Code documentation is about 5%.

The project design and the function names are very user friendly that each developer with basic knowledge with Android, java and game development can understand very fast the mining of the function and the design. This is moto of AndEngine developers.

In Addition There is several books and territorials that created for the AndEngine developers that explain how to work with AndEngine from the beginning.



***Modeling***

Game Life cycle:

Game Activity – Android Activity is the first application display.

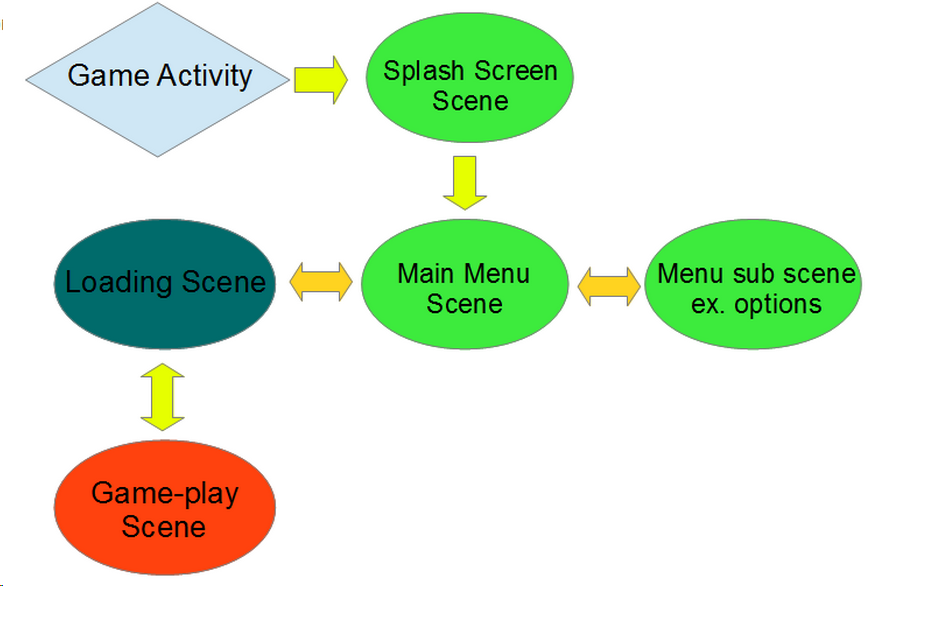
Splash Screen Scene – Logo Scene of the game in this point the game loads all the resources.

Main Menu Scene – User Menu Scene – choose the next step

Menu sub scene – options, etc.

Loading Scene – Loading logo, load the game scene.

Game-Play Scene – The game Scene

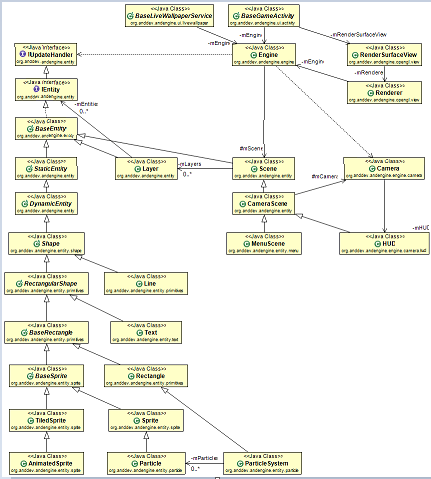


Game Engine – Class diagram:

Engine Class – The main management class

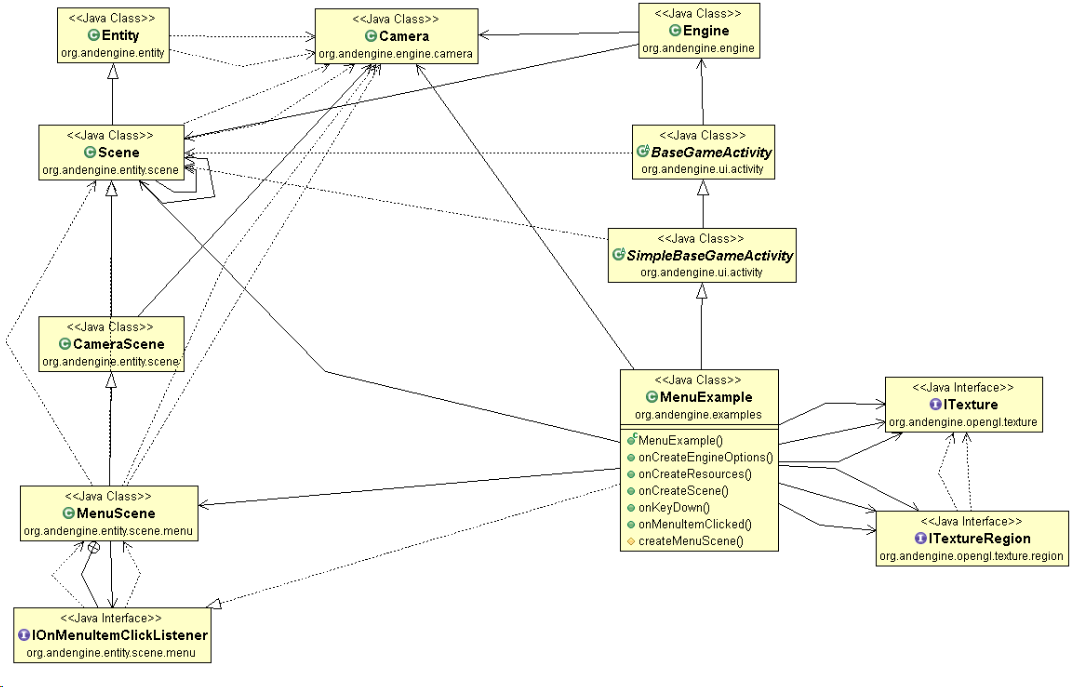
Class that extends from the Entity are all the graphical elements that showed in the Game.

All that extends from IUpdateHandler that give the Entities the ability to control as event driven design.



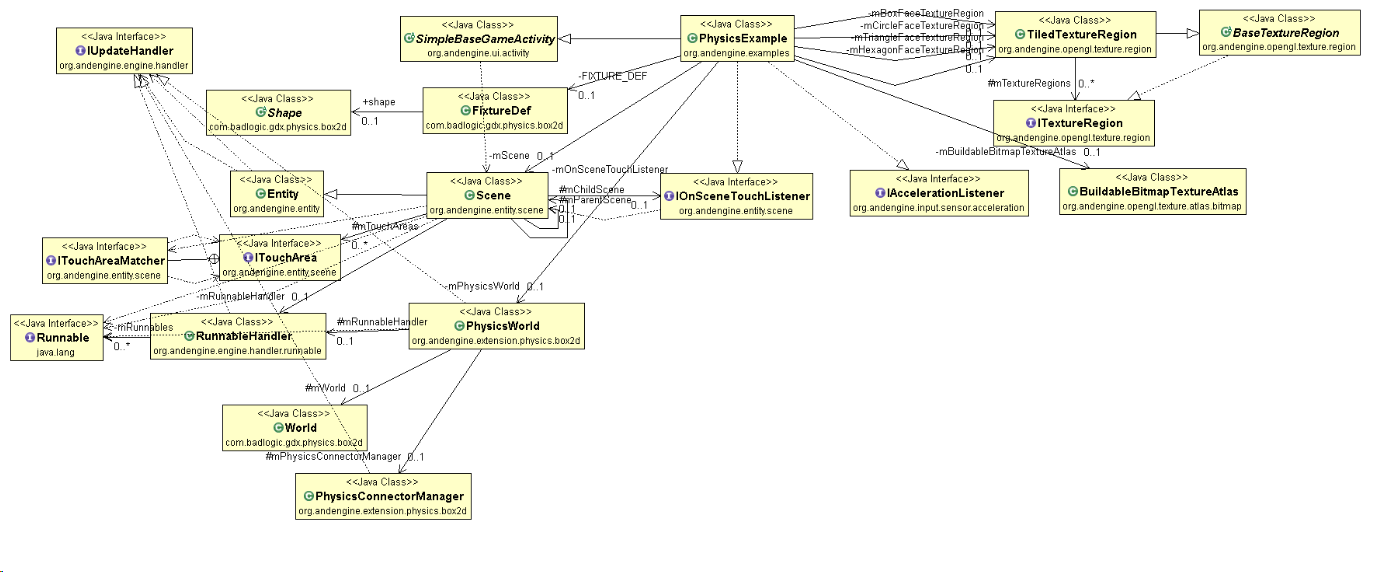
Menu Example – Class diagram:

The class diagram of the Menu Scene

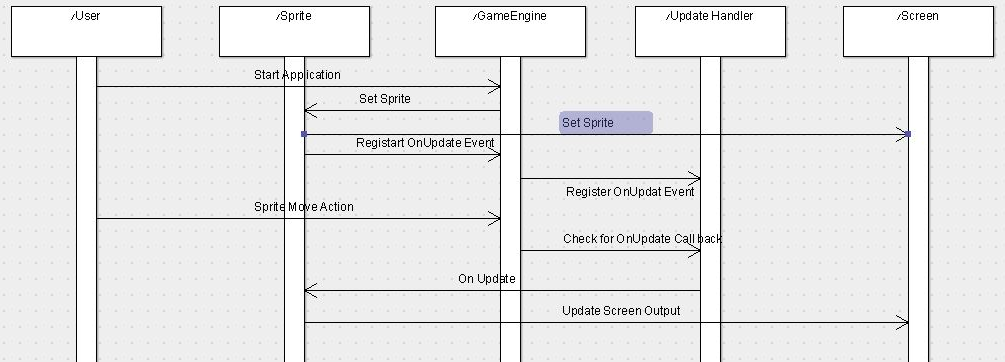


Physics Example – class diagram

The class diagram physical extention.



The Sequence Diagram of the Scene element control by events.



**Summary**

**AndEngine Pros:**

It has a complete 2-D scene graph, with a "easy-to-use" API.

It works really well with the Android activity lifecycle.

It has a number of extensions that can be added as plugins.

It’s free

It's open

**AndEngine Cons:**

The biggest problem I’ve faced while using AndEngine is that the API is undocumented.

This can increase the development time, as you sometimes need to go through the engine source code to figure things out.

The reason why there is not documentation is that the developers doesn't believe that their code need documentation.

We (Natan and Oshri) tried to understand and use the code, it is really very readable code but still some documentation would

make it easier.

The bright side of the story is that there are a lot of online tutorial and a book that helps to undersend the code,

and the most helpful are the examples project.

\*\*\*Given documentation and/or source code:\*\*\*

There are two main branches to this project, the original project created by

Nicolas and a new branch with bugfixes and new libraries created by Mayo:

[Nicloas branch](https://github.com/nicolasgramlich?tab=repositories)

[Mayo branch](https://github.com/realmayo)

Few of the available tutorials:

- [Andengine for android game development cookbook](http://www.andengine.org/blog/2013/01/andengine-for-android-game-development-cookbook/)

- [Andengine-the-basics-of-sprites] (http://stuartmct.co.uk/2012/07/18/andengine-the-basics-of-sprites/)

- [Tiled mapEditor + tutorial] (http://paulsonapps.wordpress.com/2010/06/09/tutorial-tiled-mapeditor-part-1-create-a-tmxtiledmap-game)

- [Full game from scratch] (http://andenginefromscratch.blogspot.co.il/2011/03/andengine-from-scratch-v.html)

- [Puzzle theÂ Tower of Hanoi] (http://www.raywenderlich.com/12065/how-to-create-a-simple-android-game)

- [Full game tutorial](http://www.matim-dev.com/full-game-tutorial---part-3.html)

- [Mimminito's Tutorial List](http://www.andengine.org/forums/tutorials/mimminito-s-tutorial-list-t417.html)

\*\*\*Main requirements/features of the product:\*\*\*

- Allows to create Android games with ease using AndEngine’s framework and its extensions.

- Make your games playable across a vast range of devices (multi-touch, performance-optimizations, and accurate, screen-resolution scaling)

- Allows to construct a customizable, front-end framework that simplifies menu and level creation.

- Physics-based gameplay and simulations

- Vector-based graphics (SVG extension)

- Excellent performance