

ArenaF\_Js

Файтинг по типу мортал комбат

-----

### **Соглашения по оформлению кода.**

1. В одном файле будет один корневой объект обозначенный буквой `_R` от слова корень(root), либо еще дополнительно функция конструктор обозначенная сочетанием `_CL` от слова класс(class). Корневой объект всегда именуется как файл, а вот функция класс может именоваться с изменениями, однако все равно она должна быть одна.

2.1 Используемые внутри модуля внешние функции всегда помещаются в функции-обертки в названии которых будет обозначение `_IN` (input).

Например: `ArenaFjs_R.Timer_R_updateTimeBeforeTick_IN()`; Эти функции расположены в начале модуля.

2.2 Внешний класс сам по себе не используется. Для функций из него тоже создаются функции обертки.

2.3 Функции предназначенные для использования в других модулях содержат в названии обозначение `_OU` (output).

Например: `ArenaScene_R.drawAll_OU()`;

3.1 Файлу с кодом на скрипте будет соответствовать текстовый файл описания. Например  
файл кода - `ArenaFjs.js`  
файл описания - `ArenaFjs.txt`

3.2 Файлы описания размещаются в папке `txt`.

3.3 Файлы с документацией размещаются в папке `doc`.

### **Размышления.**

Имя файла модуля совпадает с именем корневого объекта. Понятно что так делается в джаве. Только там основной единицей является класс. И от этого класса порождается сколько угодно объектов.

Тут же все извращенно. Тут нет классов однако есть функции конструкторы. Причем обычные функции являются объектами к которым можно цеплять внешние поля. К тому же контекст функции зависит от того вызываем мы функцию саму по себе или как поле объекта.

В общем много моментов совершенно не очевидных и способных вызвать шок. Очень часто все работает совсем не так как планировалось.

Дальше очевидно что нужно как то отличать функции и переменные которые мы используем в других модулях. Это функции интерфейса(interface).

Следует сделать функции обертки для входа и выхода. Тогда в случае изменения входных функций все что нужно будет сделать это исправить вызов в функциях обертках для входа. А если изменятся выходные функции то тоже интерфейс не поменяется, разве что только в случае кардинальных изменений.

Все эти ухищрения нужны для этого проекта потому что я пишу его в блокноте. Нету возможности тестирования, нету возможности поиска вхождений по модулям. Все это нужно будет отображать в ручном режиме.