

Понедельник, 11 ноября 2019 г.

Стало понятно почему так долго не писал новый код. Дело в том, что старый сделан так, что дальнейшее развитие пока не возможно. Я говорю про модель бойца, его состояния и его методы. Все это пока не продумано. Одна из целей это минимизировать количество объектов с состоянием. Боец это объект с состоянием.

Воскресенье, 10 ноября 2019 г.

Даже не ожидал что так долго не заходил в дневник. Читаю предыдущие записи и понимаю что уже ничего не помню.

Создавать клоны корневых объектов это была плохая идея потому что вложенные объекты не клонируются. В силу этого и еще некоторых других соображений от этой идеи я ухожу. Корневые объекты вызываются в любом месте программы без всякого клонирования.

Почему когда я решил использовать `Object.create`, то нигде не прочел что он делает только поверхностное копирование? Теперь же я от него отказываюсь и опять возвращаю функцию-конструктор.

Переписал код по новой.

Ноябрь (18)

Октябрь (17)

Сентябрь (16)

Август (15)

Воскресенье, 28 июля 2019 г.

Сделал массив спрайтов бойца в виде двумерного массива.

Раньше такое уже было. Потом я уходил на одномерные массивы и вот сейчас вернулся. Анимационные последовательности сделал в виде массива объектов.

Вторник, 16 июля 2019 г.

В процедуре клонирования объектов нарвался на засаду. Если в объекте есть вложенный объект то он не копируется. Вложенный объект остается один для всех клонов.

Вообще это клонирование не просто копирование. Это наследование т.е. мы создаем новый объект который ссылается на поля предка и только если мы перезаписываем поле используя ссылку на новый объект то только тогда создается новое поле. До этого все клоны используют поля своего предка.

Так что от вложенных объектов пришлось отказаться и теперь я приведу в порядок все объекты игры.

Перевел все состояния бойца в один объект бойца. Объекты обеспечивающие все в одном экземпляре и не имеют состояний.

Объект передающий приказы от игроков тоже в двух экземплярах. Надо будет подумать как от этого уйти. Прототипное наследование используемое для клонирования объектов достаточно опасная штука. Тут легко что либо не учесть.

В итоге все объекты будут одноуровневые. Ссылки на все используемые объекты явно проходят через параметры функций.

На этом предварительная переделка закончится и нужно будет писать новый код. Конечно же прежде всего это перемещение бойцов по арене. Дальше собственно бой.

Воскресенье, 13 июля 2019 г.

Пока переделывал объект пришел к выводу, что все корневые объекты надо создать на главной странице. Это все в обеспечении идеи что нужно уходить от вложенности объектов в объекты. Корневой объект это объект имеющий свой файл как класс в джаве. И именно их я создам на главной странице игры(сейчас это ArenaFjs.js)

Воскресенье, 7 июля 2019 г.

Все это время изучал html. А также работу с базами на php. Добавил на сайт примитивный счетчик. Отредактировал исходный html код страницы игры. Теперь он проходит верификацию на сайте validator.w3.org.

Тут подумал, что состояние бойца должно содержаться в одном объекте. Это нужно для того что бы минимизировать количество объектов с состоянием. Некоторые объекты дублируются именно ради состояния а это уже не оптимально, да и не уследить за всеми. Например это графическая часть бойцов.

Июль (14)

Суббота, 1 июня 2019 г.

Переписал код без функции конструктора. Еще применил правило, что внешний объект проводим через параметры функции в которой он используется. Так что в объекте мы порождаем другой объект только если он часть этого.

Так что мы уходим от проблемы скрытой неограниченной сложности когда в одном объекте может быть скрыта целая вселенная. Такого не будет. В объекте могут быть другие объекты но только если они являются частями данного объекта. Другие же используемые объекты явно проводятся через параметры функций. Так что нет ни скрытого присутствия ни скрытого использования.

Июнь (13)

Воскресенье, 19 мая 2019 г.

Кажется я осознал что такое прототипное наследование. Функция конструктор мне всегда казалась искусственной. И это все потому что идею классов мы притягиваем за уши. В данном языке это не нужно и не работает, но приходилось использовать так как я не понимал как делать экземпляры объектов. Теперь когда я обратил внимание на конструкцию `Object.create()` все стало понятно. Можно породить один объект от другого, а потом мы переопределяя изменяемые поля на самом деле просто создаем их в новом объекте так что если нужного поля нет в данном объекте браузер ищет его в прототипе.

В связи с этим я решил отказаться от методов оберток. Достаточно просто порождать местный объект-наследник. Мы избавимся от больших списков функций в начале модуля. Как правило они не нужны и только сильно засоряют код.

Воскресенье, 12 мая 2019 г.

За все это время ничего не сделал. Как всегда при достижении определенного уровня все становится непонятным и не интересным.

Смотришь на код и не понимаешь идеи лежащей в основе и насколько она правильна и насколько оптимальна ее реализация.

Я уже решил что будут файлы описания для каждого модуля. В этом направлении и надо двигаться.

Что такое модуль? Это отдельный текстовый файл содержащий тематическую подборку функций. У меня это реализовано в виде объекта имеющего то же название что и файл и плюс еще `_R` на конце. И когда мы вызываем функцию из другого модуля то оформление в виде объекта позволяет увидеть из какого модуля мы используем функцию.

Май (12)

Апрель (11)

Суббота, 16 марта 2019 г.

Мое решение проводить инициализацию по мере загрузки модулей было неправильно. Это нарушает поток выполнения программы и заставляет скакать по модулям. Гораздо удобнее когда мы видим в одном месте что и за чем инициализируется, а потом и запускается.

Пятница, 08 марта 2019 г.

Собрал все файлы игры в одной папке. Раньше код был в одном месте, а файлы описания в другом. Группировать надо не по коду и тексту, а именно по проекту и все хранить в одном месте. Раньше до меня это не доходило и я прыгал по папкам в поисках нужных файлов.

Поразительно что пошел уже девятый месяц разработки игры именно на скрипте и до сих пор нету работающего прототипа. В начале все двигалось довольно быстро, а потом все остановилось. Это тем более удивительно, что я уже написал работающую игру на джаве и на си++ и тут надо то всего лишь перенести все на скрипты.

Как всегда после некоторого перерыва ничего не помню. Просто писать в коде обширные комментарии это не правильно. Необходимо писать текстовый файл с названием как у файла кода и с описанием идеи модуля и основными решениями. Файлы описания будут собраны в папке txt. Файлы документации типа дневника и т. д. в папке doc.

Просто код никому не интересен потому что практически нет людей которые будут просто так копаться в чистом коде. Для этого им нужна дополнительная мотивация в виде оплаты или поиска решения какой то проблемы.

Март (10)

Февраль (9)

Воскресенье, 13 января 2019 г.

Переписал код с использованием входных функций. Решил, что отдельно выходные функции не нужны. Буду просто помечать их как используемые в других модулях т.е. в имени добавляется _OU (output).

Что дальше? Надо описать архитектуру самого верхнего уровня. Посмотреть что уже сделано и что нужно сделать. Проработать завершенную структуру игры.

У меня уже есть работающие прототипы на джаве и на си. Причем все в них писал я сам. Кажется ну что сложного сделать то же самое только на скриптах тем не менее идет все достаточно трудно. Понятно что архитектуру игры опишу в отдельном файле.

Январь (8)

2019

Воскресенье, 23 декабря 2018 г.

За месяц ничего не сделано. Поставил новый редактор — атом. Кое как настроил плагины на проверку кода. Проверка нашла только то что в нескольких местах не поставил точку с запятой.

Главная проблема в том что бы оформить код понятным образом. До сих пор я не решил эту задачу. Потому что каждый раз открывая код вижу, что я ничего в нем не понимаю, а ведь это мой код. В связи с этим решил еще раз проработать в отдельном документе правила оформления кода.

То что входные функции и переменные нужно обертывать в начале модуля это для меня сейчас бесспорно. Это может быть очень громоздко но другого выхода нет.

Однако большой вопрос нужно ли так поступать с выходными функциями. Или достаточно их просто помечать как выходные.

Декабрь (7)

Воскресенье, 4 ноября 2018 г.

За месяц ничего не сделано. Т.е. практически весь сентябрь, октябрь, ничего не делал. За это время собрал новый компьютер. Перешел на виндовс 10. Таким образом я полностью забросил игру. Как всегда все забыл.

Надо сказать что пишу код просто в текстовом редакторе. Это накладывает колоссальные ограничения на проект. Самое первое это то что все функции которые используются вне модуля должны быть как то обозначены и должно быть описание где именно их вызывают. Так же должно быть описано какие внешние функции каких модулей вызываются внутри модуля.

Ноябрь (6)

Октябрь (5)

Суббота, 8 сентября 2018 г.

За неделю ничего не сделано.

Надо сказать, что мне не нравится решение что аниматор использует ядро из другого объекта что бы не добавлять состояние в себя. Надо подумать надо основной структурой кода. Нижний уровень не должен использовать верхний для решения своих задач.

Поэтому данное решение не годится. Если аниматор для решения своих задач нуждается в состоянии, то необходимо его добавить, а не заимствовать со следующего уровня. Видимо это одна из причин остановки.

Плюс еще очень большое описание класса. Оно само по себе должно быть простым, а я уже в нем тону.

В итоге сделал так что то, что требует состояния вынесено в функцию конструктор, а то что общее то оставил в общем объекте.

Повторю ранее принятое мной:

«В одном файле будет либо 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.»

Плюс еще добавить, что функция конструктор обращается к корневому объекту по общим вопросам, а в ней оставляем только то что требует экземпляра т.е. используется одновременно с разными состояниями.

Обязательно в корневом объекте должна быть тестовая функция, которая показывает работоспособность объекта.

И обязательно должно быть описание какие объекты использует и какими объектами используется данный объект. Это потому что программировать приходится в блокноте и все ошибки отлавливать самому. И если не знаешь, что использует твоя функция и где она

вызывается то изменения или ошибки, превращаются в кошмар. Программа не работает, а почему не понятно.

Это сильно поднимает планку к понятности кода и само тестируемости модулей.

Суббота, 1 сентября 2018 г.

Две недели опять остановка. Обычно нужно отрисовать последовательность кадров в рамках одного действия. Например, ряд состоит из пяти кадров. Но что делать, когда ряд кончился? Повторять по новой? Или останавливаться? Что делать, когда во время выполнения поступила новая команда? Сортировать их или нет?

У аниматора есть счетчик, и чтобы его обнулять он должен понимать, что поступило задание на новый ряд либо на отрисовку старого ряда по новой. Для контролируемого управления следует выделить отдельно команду на новый ряд. И когда ряд закончится нужно останавливаться, выводя картинку заглушку. Поступление той же команды по новой начинает ряд заново.

Сентябрь (4)

Вторник, 14 августа 2018 г.

Сегодня понял, что взаимодействовать с сервером можно только по инициативе клиента и только через гет, пост запросы. Никаких постоянных соединений на джаве скрипт не создать.

Среда, 08 августа 2018 г.

Модули отрисовки картинок и анимации в начале перевел на функции конструкторы что бы иметь возможность создавать экземпляры, а теперь вернул назад как отдельный объект. Т.е. у них нет собственного состояния. Зачем я это сделал? Мне кажется, что так код будет проще и работать будет быстрее.

Хотя момент, когда что бы уйти от состояния я передаю в функцию объект кажется достаточно сомнительным. Но тут альтернативы тоже не лучше. И одна из которых я ушел это индивидуальные объекты анимации, думаю лучше пусть будет один, но работающий с состоянием вызывающего объекта.

С аниматором разобрались. Теперь надо подумать, как сделать бойца. Бойцов однозначно несколько и у каждого свое состояние. Так что делать один объект глупо. Так что для бойцов оставляю функции конструкторы ну — это аналогия классов в джава скрипт. Сейчас надо сделать сцену. Это будет один объект потому что сцена только одна.

Воскресенье, 05 августа 2018 г.

Писал код практически каждый день. Однако последние несколько дней застопорился. Писал код без комментариев и без описания. В итоге настал момент, когда после некоторого перерыва я уже не знаю, что делать и что сделано. Придется опять начать документировать. Хотя на это времени уходит больше чем на написание кода, однако без

этого невозможно понять, что я там наделал. Завел файл с основными решениями как я делаю всегда.

В начале была идея основной объект назвать также как файл, в котором он содержится и потом методы и объекты экспортировать ссылкой на него. Но так не получилось потому что функции конструкторы объектов ни к чему не привязаны.

Так что многие файлы это просто тематические сборники функций прототипов. Но так не правильно. Функция конструктор это альтернатива классу. Значит все равно следует придерживаться идеи один класс в одном файле.

В итоге решил что в одном файле будет либо 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.

Сейчас надо остановиться и добавить в код комментариев и описания.

Еще надо добавить разделы где описать какие файлы входящие а какие используют код.

Иначе трудно искать концы. Видимо стоит добавить разделы `import` и `export`. Дело в том что я использую текстовый редактор и он не ищет ошибок и не показывает вхождений.

Все приходится делать в ручном режиме.

Август (3)

Июль (2)

Воскресенье, 17 июня 2018 г.

Сегодня решил начать делать игру в браузере.

До этого первая попытка была написать на джаве. Там все хорошо кроме одного колоссального минуса. Для того что бы запустить игру нужно ставить виртуальную джава машину. И нужно запускать джар. Совершенно очевидно что никто не будет этого делать. Кроме того развернута компания против джавы так что не то что бы скачать джар но даже что бы запустить апплет необходимо постараться. Т.о. данные обстоятельства сделали проект на джаве бесперспективным. Однако понял это я не сразу и поначалу казалось что вопрос развертки это мелочь.

Вторая попытка была написать проект на си. Надо сказать что в целом получилось конвертнуть файтинг с джавы на си. Все принципиальные вопросы были решены. Но как всегда все остановилось. На си все проще. Надо просто скачать папку с программой и запустить обычный исполняемый файл. Тут только один момент надо скачивать. Это тоже как правило никто не будет делать потому что боятся занести вирус и этот страх будет решающим. И не менее решающим будет лень что то делать.

Это будет третья попытка написать проект на джава скрипте. Основная идея заключается в том что не надо ничего скачивать и ничего искать. Зашел на сайт и играй. Так и сделаю.

Сразу надо сказать что игра будет плоской. Ну это не новость.

Июнь (1)

2018

начало