

Этот гайд/справочник предназначен для того, чтобы объяснить структуру Дервинбея, привлечь пару - тройку безумцев к его тыканию, если они хотят создать свой миниатюрный мирок на его базе.

### Файлы:

base\_build.dm - Содержит конфигурацию Дервинбея. Здесь вы можете настраивать его плавность, можете отключать запуск разных контроллеров, изменять имя в шапке окна.

- `init()` - При создании мира инициализирует атмосфер. Если вы хотите отключить атмосфер, достаточно закомментировать это и все контроллеры связанные с атмосфером(атмос, трубы, вентиляцию).
- `move_gas()` - Выполняет функцию контроллера атмоса. Та самая вторая строчка для отключения атмоса.
- `new /datum/Kate()` - Создает Кейт. Кейт - искусственный помощник, вызывается через ООС. Выполняет ряд функций, таких как справочные данные о предметах на станции.
- `new /datum/pipe_controller/pipe_controller_master_n` - Создание контроллеров труб. Обрабатывают трубы в пределах своих секторов.
- `new /datum/innerouter_controller()` - Создание контроллера вентиляции.
- `CABLES()` - Выполняет функцию контроллера проводки.

code/delay.dm - Скорость движения мобов. Меняете `gundelay` - меняете скорость.

code/KATE\_CORE(ENGLISH).dm и code/KATE\_CORE(RUSSIAN).dm - Кейт на двух языках.

Этот кусок кода представляет из себя наборы слов и поиск во фразах игрока названий разных объектов и их параметров. Ничего особо сложного, можете поэкспериментировать с этим.

code/lagfilter.dm - Сглаживает резкость движений, микрофризы и прочее.

code/other\_proc - Содержит некоторые функции.

- `do_after` - делает что-то по истечении некоторого времени
- `is_type_in_list` - проверяет, есть ли какой-либо объект в списке

code/area/area.dm - Зоны. Ничего интересного.

code/chemistry - Станционная химия. С ней разбирайтесь сами.

code/food - Станционная кухня.

code/GAS/connector.dm - Коннекторы и канистры. У коннекторов, канистр и прочего атмосохлама имеются переменные для каждого из газов. `oxy` - кислород, `nit` - азот, `plas` - плазма, они показывают, сколько внутри содержится этих газов. `connected` и `open` -

переменные отвечающие за то, приконекчена ли канистра к коннектору и открыта ли она.

code/GAS/filter.dm - Фильтры. Если мы к примеру фильтруем кислород, то он пойдет по средней трубе в другую ветку, а остальной газ спокойно пойдет прямо.

code/GAS/fire - Код огня и зажигалки. У огня есть максимальная генерируемая им температура(19 строчка), также установлена температура при которой на станции начинает загораться кислород (21 строчка). Если кислорода слишком мало, огонь гаснет. Огонь оказывает тепловое воздействие, код которого представлен ниже.

code/GAS/GAS.dm - непосредственно атмос. Суть в чем - есть матрицы 250 на 250(стандартный размер карты), каждая из них какой-то газ или параметр атмоса, ячейка - значение конкретного турфа. Атмос состоит из:

1. plasma - плазма
  2. oxygen - кислород
  3. nitrogen - азот
  4. temperature - температура
  5. windoors\_on\_turf - есть ли на конкретном турфе, что-нибудь, что делает движение газа невозможным(стены, двери, стекла и тд)
  6. space\_is\_here - космос
  7. pressure - давление
- init() - расставляет начальные значения
  - Переменные ниже хранят в себе значения параметров на соседних от текущего турфах.
  - move\_gas() - бесконечный цикл, разбивает всю карту на сектора и в порядке очереди выполняет атмосферорасчеты с задержкой delay.
  - my\_min\_my\_max(var/a, var/b) - выбирает максимальное и минимальное значение. Это нужно, чтобы сократить количество инициализаций цикла до минимума и тем самым слегка оптимизировать код. Прием честно подсмотрен в википедии. Вообще если говорить о оптимизации, с этой же целью я объявил все переменные до цикла, ввел задержку в зависимости от нагрузки.
  - part\_of\_processing(var/X, var/Y, var/A, var/B) - непосредственно атмосферорасчеты на конкретном секторе от X, Y до A, B.

code/GAS/inner.dm - Вентиляция, которая работает на вдув. Все просто, берет значени из атмосфероматриц, отнимает немного оттуда и прибавляет немного себе.

code/GAS/manifold.dm - Трехтрубники. Они должны иметь как минимум одну входную трубу и одну выходную. Так как каких-то в любом случае будет больше, они распределяют газ между этими трубами.

code/GAS/outer.dm - Вентиляция, которая работает на выдув. Все просто, берет значения из соседних труб, отнимает немного оттуда и прибавляет немного в соседние атмосферички.

code/GAS/pipe.dm - Трубы. Минус труб Дервинбея в том, что в трубах нет давления, потому на них размещены стрелочки, которые показывают направление движения газа в них.

code/GAS/pipe\_switcher.dm - Выключатели. Меняют свою атмосферопроводимость по клику.

code/hydroponics - Станционная гидропоника

code/mob/clothes.dm - Одежда. Константы наверху (SUIT\_LAYER, HEAD\_LAYER и тд) означают слои отрисовки.

- get\_armor() - Возвращает суммарное значение брони для конкретной зоны на теле человека.
- iszone(var/my\_zone) - Возвращает 1 в случае, если my\_zone есть в списке защищаемых одеждой областей.

code/mob/damage.dm - Код потери крови.

- bloodloss() - потеря крови
- blood\_new() - регенерация крови

code/mob/farm - Сломанный код петушков и немного кода гостей.

code/mob/human.dm - Основной файл с кодом людей.

В самом верху мы можем видеть код органов. Органы имеют следующие переменные:

- hit\_points - очки здоровья конкретного органа, из среднего значения всех органов высчитывается общее состояние здоровья человека.
- burn\_points - ожоги. Влияют на очки здоровья.
- inner\_organ - внутренний орган, если он имеется. Например голова содержит мозг, а туловище - сердце.
- step\_of\_operation - текущий этап операции на конкретном органе.

Процедуры и переменные самого человека:

- white\_overlay() - Придание персонажу внешности белого мужчины.
- white\_fem\_overlay() - Придание персонажу внешности белой женщины.
- black\_overlay() - Придание персонажу внешности черного мужчины.
- black\_fem\_overlay() - Придание персонажу внешности черной женщины.
- html\_me() - лобби, связано с файлом code/other/topic.dm.
- my\_key - хранит skey с момента входа в игру
- S - мелодия в лобби.
- lying - лежит ли игрок или стоит.

- stop - я м сам забыл зачем это нужно.
- state\_of\_health - состояние игрока.
- stamina - усталость.
- my\_oxygen - есть ли воздух для дыхания.
- pulse - пульс.
- min\_pulse - минимальный пульс при нормальных условиях.
- max\_pulse - максимальный пульс при нормальных условиях.
- signal - мертв или жив.
- reagents\_max - сколько единиц реагентов может хранить.
- cur\_val - сколько единиц реагентов содержит.
- mcolor - цвет кожи
- my\_gend - пол
- oldloc - предыдущий турф, на котором стоял моб
- my\_weight - сколько веса персонаж несет на себе.
- lying() - положить персонажа на пол.
- unlying() - поднять персонажа на ноги.
- process() - главная процедура обработки состояния человека. В ней вызываются различные процедуры, которые проверяют состояние человека и меняют его.
- Move() - То, что происходит при движении.
- start\_game() - то, что происходит при входе в игру.
- punch\_organ(var/obj/item/organs/O) - удар кулаком по человеку.

code/mob/item\_attacks.dm - Обработка ударов различными предметами по человеку.

- help\_me\_doctor(var/obj/item/I, var/obj/item/organs/O) - залечить рану мазью.
- EXTINGUISH(var/obj/item/I, var/obj/item/organs/O) - залечить ожог мазью.
- shake\_me(var/client/C) - Потрясти камеру игрока(при ударах например)
- punch\_me\_doc(var/obj/item/I, var/obj/item/organs/O) - ударить человека предметом.

code/mob/life.dm - Влияние газов на человека, усталости и пульса на человека. Ну тут все просто, меняются иконки датчиков сбоку, человеку становится хуже или лучше и тд.

code/mob/new\_clothing.dm - код отрисовки одежды.

- wear\_to\_slot(var/mob/human/M, var/list/L, var/obj/O) - добавляет иконку одежды на слот.
- wear(var/mob/human/M) - Добавляет на человека оверлей одежды и добавляет ему в контейнер(contents) эту одежду.
- undress(var/mob/human/M) - Убирает оверлей и одежду с человека.
- lying\_me(var/mob/human/M) - Переводит оверлей одежды в лежачее положение.
- unlying\_me(var/mob/human/M) - Переводит оверлей одежды из лежачего положения.

code/mob/new\_hud.dm - код отрисовки интерфейса.

Константы, переменные и процедуры:

- RHAND и LHAND - обозначают руки соответственно
- active - активная рука
- var/obj/screen/x - объекты интерфейса
- my\_pull\_eba - пуллит ли человек объект
- can\_get - максимальный размер, который может тащить человек
- my\_weight - вес персонажа

code/mob/new\_interact.dm - Обработка кликов игрока.

- client/show\_popup\_menus - показывать меню при клике или нет.
- pew(var/atom/movable/A) - выстрел
- act() - Действие свободной рукой
- hit(var/atom/movable/A, var/direction, var/force) - попадание чего-либо во что-либо.
- act\_by\_item(var/obj/item/I) - действие предметом в руке.
- act\_self(var/obj/item/I) - действие по предмету в руке.
- del\_from\_hand(src) - удалить предмет со слота руки.
- drop() - уронить предмет из активной руки.
- drop\_all() - уронить предметы из всех рук.

code/mob/organs.dm - Содержит одну процедуру, которая высчитывает состояние здоровья из среднего состояния всех органов.

- organ\_damage\_calculate() - Да, это она.

code/mob/say.dm - Чаты и фикс буквы я. Содержит в себе эмоты, внутриигровой чат и ООС. Фикс буквы для html(1103) и для разговоров в чатиках(255).

code/food&drinks/reagents.dm - Различные напитки и еда.

code/obj/logic.dm - Логические элементы. Могут выступать в роли различных триггеров для всяких событий и ловушек.

- calculate() - Выполняет логическое действие.
- connect() - Ищет соседние элементы, чтобы получить или передать свой сигнал.
- start() - Начало работы элемента.

code/obj/signs.dm - Вывески.

code/obj/EFFECTS.dm - Ландмарки и иконки некоторых эффектов.

code/obj/other/job.dm - Опять ландмарки. Ландмарки профессий.

code/obj/structure/window.dm - Код окон.

code/objects/items.dm - Различные предметы.

code/objects/Kate.dm - Пока здесь почти ничего.

code/objects/machine.dm - Различные машины.

- check() - Проверка электропитания машины.
- process() - Главная процедура машины.
- need\_amperage - Необходимое для работы количество энергии
- on - включена или выключена машина
- upd\_disp() - Сменить картинку на экране терминала со сменой диска.
- find\_machines() - Вывести на экран все доступные для управления с консоли машины.
- computer\_access\_me - ID доступных для компьютера машин

code/objects/material.dm - Материалы для строительства. Кода пока нет.

code/objects/medical.dm - Различный медицинские шутки.

code/objects/paper.dm - Ну тут все просто. Код бумаги и ручки.

code/objects/structure.dm - Стулья, столы, окна, кровати и тд.

code/objects/accelerator.dm - Код синги, полей, ускорителя и коллекторов.

- input\_(var/my\_power) - Получить с коллекторов энергию my\_power.
- walk\_() - Прогулки сингулярности.

code/objects/conveyor.dm - Конвейеры. Включаются и выключаются. Двигают говно на тайле с ними. Ничего интересного.

code/objects/door.dm - Аирлоки и фаерлоки.

- open() - Открыть дверь
- close() - Закрыть дверь

code/objects/laser.dm - Раньше здесь был код лазеров. Но он был убог и я его переписал в другом месте. Здесь хранятся константы, определяющие силу выстрелов.

code/machine/robots.dm - Пиздецки сырой код роботов. Даже лучше и не трогать.

code/machine/sensors.dm - Сенсоры. Отправляют сигнал, если вы на них наступили.

code/machine/vendings.dm - Вендоматы.

- my\_items и my\_types - содержат типы предметов внутри
- items\_num - число предметов для каждого типа в автомате

code/weapon/book.dm - Книжки. Все просто.

code/weapon/melee.dm - Всякое оружие ближнего боя.

code/weapon/proj.dm - Лазеры.

code/other/math.dm - Математические функции.

code/other/topic.dm - Обработка запросов с разных интерфейсов.

code/other/UI.dm - Стили отображения.

code/physics\_plus/BOOM.dm - Взрывы и ударная волна.

- force\_to(var/value\_of\_force, var/dir\_of\_force) - Воздействовать силой value\_of\_force в направлении dir\_of\_force на объект.
- BOOM(var/turf/center\_of\_explode, var/force\_v) - Сделать взрыв, где center\_of\_explode эпицентр взрыва, а force\_v сила взрыва
- frag(var/force\_) - Осколки. Сила определяет дальность распространения осколков.
- frag\_damage() - урон от осколков.
- destroy() - Разрушить нечто.

code/physics\_plus/heat.dm - Тепловое воздействие.

- FLAMMABLE - Объект горюч.
- NOFLAMMABLE - Объект не горит.
- MELTS - Объект плавится.
- EXPLODE - Объект взрывается.
- fire() - возгорание объекта.
- melts() - плавление объекта.
- heat(var/temp, var/sq) - Воздействие теплом с температурой temp на поверхность sq

code/physics/main.dm - Старый код взрывной волны.

code/physics\_plus/materials.dm - Код материалов. Содержит различные значения физ.величин для разных материалов, которые могут содержаться в объектах.

- get\_endurance() - Получить общую прочность объекта.
- get\_resilience() - Получить общую упругость объекта.
- material\_damage(var/damage) - Нанести урон объекту, снизить прочность материалов в нем.

code/power/old\_power.dm - Древний код энергии, который получил вторую жизнь. Работает с апц.

code/turf/main.dm - Турфы.

code/turf/simulated.dm - Симулируемые турфы.

code/wall.dm - Мерж стен.

code/waves/main.dm - Радиоволны, различные другие волны и радиоприборы.

