

all.spawn

Материал из xrWiki

all.spawn — важная часть игровой системы ИИ. В этом файле хранятся настройки спавна различных игровых сущностей. Файл бинарный, блокнотом не распаковывается.

Структура

all.spawn, как и подавляющее число игровых файлов сталкера, имеет RIFF-формат. Он состоит из многочисленных вложенных друг в друга чанков. На самом нижнем уровне вложенности находятся чанки секций предметов, которые в свою очередь, в точности соответствуют формату нет-пакета этого предмета в игре. В принципе, упрощенно можно считать, что **all.spawn** содержит начальные нет-пакеты, которыми движок инициализирует объекты сталкера. Также в нем содержатся настройки автоматического спавна артефактов на локациях и пути движения объектов.

Редакторы

all.spawn может редактироваться несколькими способами:

1. Официальный способ — создание спавна в **Level Editor**. После создания или правки спавна для каждой локации запускается утилита xrAI, которая объединяет все локальные спавны в один большой глобальный. Способ, пожалуй, самый неплохой, но требующий некоторого количества времени и наличия исходников локации. Отлично подходит для заселения локации с нуля.
2. **xrSpawner**. Упоминаю только как дань истории. Пользоваться утилитой не рекомендую — может испортить спавн.
3. **ACDC**. Скрипт легендарного bardak'a, написанный на Perl. Идеально подходит для мелких правок и анализа спавна. На сегодня существуют версии acdc под все части сталкера и все патчи, их вы можете найти тут. Отдельно выделю Universal ACDC, поддерживающий сразу все версии спавна, включая билд-спавны.

Описание

ACDC в процессе работы создает ряд файлов формата ltx, для редактирования которых подойдет обычный блокнот. Файлы содержат ряд секций с параметрами, как правило, соответствующими параметрам объектов в LE. Если будет необходимо, буду давать пояснения по ходу.

all.ltx

Содержит общую информацию о спавне:

- **graph_version** — версия game.graph. ТЧ – 8, ЧН, ЗП – 10.
- **guid** — глобальный идентификатор спавна. Подробнее о guid.
- **graph_guid** — глобальный идентификатор game.graph. Используется для проверки соответствия спавна и графа.
- **level_count** — количество уровней

Далее идут имена файлов, полученных при декомпиляции спавна.

alife_***.ltx

В этих файлах содержатся секции с настройками спавна объектов игры. В all.spawn объекты идут подряд, при распаковке разбиваются по принадлежности к локации (которая в свою очередь определяется по глобальному вертексу). Каждая секция начинается с индекса в квадратных скобках, что означает начало новой секции. Индексы выдаются asdc произвольно и в игре ничего не означают.

Перед разбором немножко теории. Как известно, в сталкере существует развитая система классов объектов. Немного опишу систему серверных классов. Есть абстрактный класс CSE_Abstract, от которого наследуют все остальные классы (намеренно упрощаю). Если взять, скажем, оружие (автомат), то для него дерево наследования будет выглядеть так:

CSE_AlifeWeaponMagazined->CSE_AlifeWeapon->CSE_AlifeItem->CSE_AlifeDynamicObject->CSE_AlifeObject->CSE_Abstract.

В данном случае нам важен такой факт: при наследовании класса наследуются также и все параметры объекта, из чего следует, что у объектов есть общие параметры, а есть характерные только для них. Родственные объекты (оружие, например) имеют больше общих параметров. Следует сказать и о типах параметров. Их два — **M_SPAWN** и **M_UPDATE**. Параметры первого типа определяют начальный спавн объекта. В параметрах второго типа сохраняется текущее состояние объекта при игровых апдейтах. В распакованном спавне это параметры с индексом **upd**. В спавне для этих параметров следует указывать те же значения, что и для одноименных обычных параметров, так как движок при первом же апдейте начнет использовать апдейт-параметры. Например, если поменять координаты спавна в **position**, но забыть поменять координаты **upd:position**, изменения не будут учтены при спавне. Апдейт-параметры есть не у всех типов объектов, по тексту будет указано, у каких есть. Пока опишу только полезные для модостроителей параметры. Остальные не нужны практически никогда.

Общие для всех объектов параметры (параметры cse_abstract)

- **section_name = climable_object**
Имя секции объекта. Описание секции задается в конфигах, имя секции имеет критически важное значение – именно по этому имени движок узнает, какой класс имеет данный объект.
- **name = clmbl#0**
Имя объекта. Давайте своим объектам уникальные имена.
- **position = 127.544509887695, 8.15257263183594, 67.6282119750977**
Координаты спавна объекта.
- **direction = 0.0420370548963547, 0.0266562905162573, 1.56232643127441**
Ориентация объекта в пространстве. Каждый параметр представляет собой угол поворота объекта в радианах относительно одной из осей.
- **version = 123**
Версия спавна. Править обычно не надо, параметр нужен для нормальной работы universal asdc.
- **script_version = 8**
Версия скриптовой системы серверных объектов. Править обычно не надо, параметр нужен

для нормальной работы universal acdc.

Общие для всех объектов параметры (параметры cse_alife_object)

Эти параметры также общие для всех объектов, которые правят модостроители. Этих параметров нет у граф-поинтов, но они есть только в level.spawn и их не надо править – они представляют собой заготовки для хгАИ.

Также перед чтением этого раздела советую изучить статью ИИ в Сталкере. Отпадут вопросы про глобальный и локальный вертексы.

▪ game_vertex_id = 1112

Глобальный вертекс уровня. Необходим для корректного спавна на нужной локации.

▪ distance = 32.899998

Дистанция от локального вертекса до ближайшего глобального. Обычно не нужна, но некоторые типы объектов требуют правильного значения.

▪ level_vertex_id = 29696

Локальный вертекс уровня. Обычно можно поставить -1, но некоторые типы объектов требуют правильного значения

▪ object_flags = 0xfffffbf

Флаги объекта. Представляет собой бинарный флажок, в котором каждый бит соответствует какому-нибудь переключателю. Вот список переключателей и соответствующих битов:

```
|-----|
|flUseSwitches      = 0x00000001;  //Используется только редактором уровней ([[LevelEditor]]
|                               //из X-Ray SDK) и отвечает за видимость в редакторе
|                               //флажков flSwitchOnline и flSwitchOffline
|flSwitchOnline     = 0x00000002;  //Можно ли переводить в онлайн
|flSwitchOffline    = 0x00000004;  //Можно ли переводить в оффлайн
|flInteractive      = 0x00000008;
|flVisibleForAI     = 0x00000010;  //Для зон отвечает за возможность реакции на контакт
|flUsefulForAI      = 0x00000020;
|flOfflineNoMove    = 0x00000040;  //По идее должен отвечать за отсутствие движения в
|                               //оффлайне, но не используется
|flUsedAI_Locations = 0x00000080;  //Использовать ли левел вертекс для спавна
|flUseGroupBehaviour = 0x00000100;
|flCanSave          = 0x00000200;  //Сохранять ли объект
|flVisibleForMap     = 0x00000400;  //Устанавливает видимость на миникарте
|flUseSmartTerrains = 0x00000800;
|flCheckForSeparator = 0x00001000; //Флаг для компилятора хгАИ. Устанавливает, проверять
|                               //ли данный спейс-рестриктор на разделение АИ-сетки.
|                               //В игре не используется.
|flCorpseRemoval    = 0x00002000;
|-----|
```

Работает просто: если соответствующий бит в object_flags установлен в 0, переключатель выключен. Если 1 – включен. Таким образом в 32х битном значении можно хранить 32 переключателя, но в игре используется только 14.

▪ custom_data = <<END

Содержимое конфига текущего объекта – логика, описание респавна, конфиг смарт-террейна и много другое. Можно писать прямо тут, можно делать ссылки на игровые конфиги. Вот пример:

```
|-----|
|[[logic|
|cfg = scripts\agroprom\zooman.ltx
|END
|
|
```

▪ **story_id = 1023**

Уникальный игровой идентификатор объекта. Используется для быстрого получения объекта в скриптах, в которых есть специфические функции для этого. Обычно задается для сюжетных персонажей и предметов для облегчения манипулирования ими.

Больше у объектов общих параметров нет. Объекты (и, соответственно, параметры) можно далее разделить на две группы: имеющие визуал (игровую модель) и не имеющие ее. К первым относится большая часть объектов — живые существа и различные предметы. Ко вторым — лестницы, области пространства (рестрикторы, аномалии, смарт-каверы, смарт-террейны, переходы между уровнями). Рассмотрим их по отдельности.

Параметры объектов без визуала

Каждый объект описывается **формой (shape)**.

▪ **shapes = 1**

Количество форм, которыми описывается объект

▪ **shape_0:type = sphere**

Тип формы. Может быть сфера (sphere) или коробка (box)

Для сферы:

▪ **shape_0:offset = 0,0,0**

Смещение центра сферы от координат объекта, указанных в position.

▪ **shape_0:radius = 2**

Радиус сферы

Для коробки:

▪ **shape_0:axis_x = 1,0,0**

Длина по оси X

▪ **shape_0:axis_y = 0,1,0**

Длина по оси Y

▪ **shape_0:axis_z = 0,0,1**

Длина по оси Z

▪ **shape_0:offset = 0,0,0**

Смещение центра коробки от координат объекта, указанных в position.

Длины коробки так странно выглядят, потому что в спавне они заданы как матрица трансформации объекта, поскольку движок оперирует именно ими. Если описываете объект правильным параллелепипедом, значение имеют только «диагональные» координаты: первая для X, вторая для Y, третья для Z. Остальные должны быть нулевые. Если же меняете их, грани шейпа в игре не будут параллельны, коробка будет «кривой».

Для аномалий обычно используется сфера, для всего остального – коробка.

▪ **restrictor_type = 3**

Тип рестриктора для объекта.

Параметры аномалий

Есть два типа аномалий: **статичные** (не двигающиеся с места) и **динамические**.

Динамические аномалии могут двигаться в пространстве (помните аномалии в X-18?), но не

могут порождать артефактов. Оба типа имеют разный набор параметров.

Статические аномалии:

- **offline_interactive_radius**
Радиус аномалии в оффлайне
- **artefact_spawn_count**
Количество случайных точек спавна артефактов
- **artefact_position_offset**
Смещение в section2.bin, по которому расположены конфиги точек спавна артефактов

Существует тип статических аномалий, который имеет анимации. Это **cse_alife_zone_visual**, в игре это жарки, например. У них есть еще два параметра:

- **idle_animation**
Имя анимации в режиме ожидания
- **attack_animation**
Имя анимации при активации аномалии.

Динамические аномалии:

- **motion_name**
Путь до файла с объектной анимацией (anm)

Параметры переходов между уровнями

- **dest_game_vertex_id**
Глобальный вертекс точки, куда будем прыгать. Необходимо корректное значение на новой локации
- **dest_level_vertex_id**
Локальный вертекс точки назначения
- **dest_position**
Позиция, в которую прыгаем
- **dest_direction**
Направление взгляда после прыжка
- **dest_level_name**
Имя локации назначения. Используется вроде только в LevelEditor
- **dest_graph_point**
Граф-поинт назначения. Используется вроде только в LevelEditor

Параметры смарт-каверов

- **description**
Тип смарт-кавера
- **enter_min_enemy_distance**
Минимальное расстояние до врага, при котором непись туда лезет
- **exit_min_enemy_distance**
Минимальное расстояние до врага, при котором непись вылезает из укрытия
- **is_combat_cover**
Боевое ли укрытие

Параметры объектов с визуалом

К этому типу объектов принадлежит большая часть объектов сталкера. Это монстры, сталкеры, актер, лампочки, стекла, предметы и оружие. Рассмотрим их с разделением по категориям.

Общий параметр:

- **visual_name = actors\stalker_neutral\stalker_neutral_1**
Тут все просто: ссылка на модель объекта от папки meshes. Пишется без расширения ogf, будьте внимательны.

Лампочки

- **main_color**
Цвет
- **main_brightness**
Яркость
- **main_color_animator**
Световая анимация (к примеру, мигание)
- **main_range**
Дальность свечения
- **light_flags**
Флаги лампочки
- **lamp_fixed_bones**
Кость, за которую фиксируется лампочка
- **health**
«Здоровье» лампочки
- **main_cone_angle**
Растр светового конуса
- **glow_radius**
Радиус свечения
- **ambient_bone**

Живые существа

У сталкеров, монстров и актора есть общие параметры:

- **g_team**
Команда. Оставляйте нулем.
- **g_squad**
Отряд. Оставляйте нулем.
- **g_group**
Группа. Оставляйте нулем.
- **health**
Здоровье. Поставите 0 — заспавнится труп.
- **dynamic_out_restrictions**
Рестрикторы, за пределами которых объекту нельзя быть.
- **dynamic_in_restrictions**
Рестрикторы, внутри которых объекту нельзя быть.
- **equipment_preferences = 0, 0, 1, 1, 0**
Предпочтение снаряжения. Имеет отношение к файлам *ai*.efd*, лучше не трогать.

- **main_weapon_preferences = 2, 0, 2, 1**

Предпочтение оружия. Имеет отношение к файлам *ai*.efd*, лучше не трогать.

- **upd:health = 1**

Здоровье. Должно соответствовать **health**

- **upd:timestamp**

Временная метка. Не трогать

- **upd:position**

Апдейт координат спавна. Должно соответствовать **position**.

- **upd:o_torso**

Апдейт направления туловища объекта. Задается так же, как и **direction**, в радианах.

У актора, stalkеров и торговцев также можно задать следующие параметры:

- **money**

Количество денег

- **trader_flags**

Флаг торговли.

- **character_profile**

Ссылка на профиль.

Предметы

Физические предметы, которые нельзя брать в инвентарь:

- **physic_type**

- **mass**

Масса предмета

- **fixed_bones**

Кость фиксации предмета (если предмет нельзя сдвинуть)

Инвентарные предметы

Оружие, еда, аптечки – все, что можно взять в инвентарь.

- **condition**

Состояние. Изменяется от 0 до 1.

Дополнительные параметры для оружия:

- **ammo_current**

Текущее количество патронов в стволе

- **addon_flags**

Навешанные аддоны. Бинарный флаг по сути, но распишу подробнее:

- **0** - нету

- **1** - прицел

- **2** - гранатомет

- **3** - прицел+подствольный гранатомет

- **4** - глушитель

- **5** - прицел + глушитель

- **6** - подствольный гранатомет + глушитель

- **7** - все сразу

- **ammo_type**

Тип патронов. 0 - обычные, 1 - бронебойные. Если есть еще - идут в том же порядке, что

указан в конфигах.

- **upgrades**

Только ЗП или ЧН старше 4 патча. Тут можно указать через запятую список сделанных апгрейдов для оружия.

- **upd:current_fire_mode**

Режим стрельбы. 0 - одиночными, 1 - два патрона, 2 - очередью. Если режима по 2 патрона у оружия нет, то режим очередью будет 1.

- **upd:grenade_mode**

Только у оружия с возможностью крепления подствольника. Режим подствольника. 0 - выключен, 1 - включен

- **upd:condition = 1**

Состояние оружия. Необходимо обязательно ставить этот параметр, иначе оружие заспавнится сломанным.

section2.bin

Представляет собой набор данных для периодического рандомного спавна артефактов на локациях. Состоит из блоков по 20 байт, которые содержат координаты спавна, локальный вертекс и дистанцию от локального вертекса до ближайшего глобального. Модостроителю блок обычно не нужен, для интереса можно посмотреть его содержимое с помощью Universal ACDC, запустив распаковку с ключом *-af*.

section4.bin

Имеется только в спавнах ЧН и ЗП. Представляет собой **game.graph**, начиная с ЧН его зашивают прямо в спавн.

way_***.ltx

В файлах **way** содержатся пути для объектов спавна. Каждый путь начинается с секции в квадратных скобках, в которой указывается имя пути:

- **[agr_smart_terrain_4_4_sleep_3]**

Далее идут настройки пути:

- **points = p0,p1**

Перечисление индексов всех точек (waypoint), содержащихся в пути

Конфиги точек:

- **p0:name = name00**

Имя точки. Также может содержать флаги для парсера логики. Подробнее о флагах (и вообще путях) читайте тут.

- **p0:position = 19.5853118896484,0.166345998644829,24.0984535217285**

Координаты точки

- **p0:game_vertex_id = 1111**

Глобальный вертекс точки

- **p0:level_vertex_id = 168909**

Локальный вертекс точки

- **p0:links = p1(1)**

Связи с другими точками. Задаются в формате:

индекс_точки(вероятность_перехода_в_точку). Если указано 1, объект обязательно перемещается именно в эту точку, если, скажем, 0.5 – объект переместится в точку с вероятностью 50%.

Автор

Статья написана **K.D.**

Источник — «<https://xray-engine.org/index.php?title=all.spawn&oldid=676>»

Категории:

Справка

Форматы файлов

-
- Страница изменена 2 декабря 2017 в 21:30.
 - К этой странице обращались 11 578 раз.
 - Содержимое доступно по лицензии GNU Free Documentation License 1.3 или более поздняя (если не указано иное).

