

# Передача параметров в функции

Материал из xrWiki

Ниже перечислен набор функций с параметрами, присутствующими в `xr_conditions` и `xr_effects`, к которым можно обращаться из кастом даты и при этом передавать в них переменные.

Во всех функциях `xr_conditions` и `xr_effects`, которые обращались к гулагам по имени, теперь можно использовать как имя, так и `story id`. Причем если мы указываем имя, то использовать функцию можно только, когда гулаг находится в онлайне, а если мы вешаем на самрттеррейн `story_id`, то можем обращаться к гулагу и в оффлайне.

## Содержание

- 1 `xr_conditions`
- 2 `xr_effects`
- 3 Функции для работы с HUD'ом
- 4 Функция запуска `camera_effector'a`
- 5 Функция запуска постпроцесса
- 6 Функции для работы с инвентарём
- 7 Настройка отношений
  - 7.1 Функции
  - 7.2 Проверки
  - 7.3 Пример

## `xr_conditions`

```
┌-----┐  
|fighting_dist_ge(p)|  
└-----┘
```

Универсальная функция для `combat_ignore`, проверка расстояния для игрока (в метрах).

```
┌-----┐  
|distance_to_obj_le(sid:dist)|  
└-----┘
```

Проверка дистанции до объекта, заданного `story_id`.

Можно использовать, например, в секции `follower` для определения того, что сталкер подошел на нужную дистанцию к лидеру и переключать в другую секцию (лидер при этом стоит где-то в ремарке). Эта ситуация возникает, когда после боя надо подогнать одного сталкера к другому, а их позиций мы не знаем. Если используется в секции `follower`, то **dist** надо ставить большим **distance** фолловера, поскольку если поставить их одинаковыми, то данная функция не всегда будет срабатывать.

```
|health_le(health)
```

Проверка того, что здоровье npc <= health

```
|heli_health_le(health)
```

Аналогично предыдущему, только для вертолета.

```
|enemy_group(group1:group2:...)
```

Проверка на принадлежность врага к одной из групп (правильность работы пока не проверялась).

```
|hitted_by(sid1:sid2:...)
```

Проверка того, что удар был нанесен кем-то из NPC, указанных в списке. NPC задаются с помощью story\_id. Функцию удобно использовать в секции hit.

Пример:

```
|[hit]  
|on_info = {=hitted_by(407:408)} %+val_escort_combat%
```

```
|killed_by(sid1:sid2:...)
```

Аналогично предыдущему, но для случая смерти npc. Используется в секции death.

```
|is_alive(sid)  
|is_alive_one(sid1:sid2:...)  
|is_alive_all(sid1:sid2:...)
```

Проверка того, что один, один из нескольких или все из списка соответственно npc, заданные по story\_id, живы.

```
|is_dead(sid)  
|is_dead_one(sid1:sid2:...)  
|is_dead_all(sid1:sid2:...)
```

Аналогично предыдущему, только проверка на "мертвость".

```
|check_fighting(sid1:sid2:...)
```

Проверка того, не является ли кто-то из перечисленных (с помощью story\_id) npc врагом данного. Как правило, используется в combat\_ignore\_cond.

```
lgulag_empty(gulag_name)
```

Проверка того, что гулаг пуст или вообще не существует.

```
lgulag_population_le(gulag_name, num)
```

Проверка того, что количество народу в гулаге  $\leq$  num

```
lgulag_casualties_ge(gulag_name:num)
```

Проверка того, что гулаг понес потери  $\Rightarrow$  num

Потери гулага не обнуляются, так что с этой функцией работать аккуратно.

```
lsignal(строка)
```

Проверяет, установлен ли у данного НПС в текущей схеме указанный сигнал.

## xr\_effects

```
lheli_set_enemy(story_id)
```

Сделать npc с указанным story\_id врагом вертолёту. В одной секции можно задавать только 1 врага.

```
lset_gulag_enemy_actor(gulag_name)
```

Сделать актера врагом для данного гулага.

```
lhit_npc(direction:bone:power:impulse:reverse=false)
```

Нанести хит по npc.

Параметры:

- **direction** - если строка, то считается, что это имя пути и в сторону первой точки производится толчек. Если же это число, то оно рассматривается как story\_id персонажа от которого должен поступить хит;

- **bone** - строка. Имя кости, по которой наносится удар;
- **power** - сила удара;
- **impulse** - импульс;
- **reverse (true/false)** - изменение направления удара на противоположное. По умолчанию false.

Пример:

```
[[death]
|on_info = {=killed_by(404)} %=hit_npc(404:bip01_spine1:100:2000)%, {=killed_by(405)}
%=hit_npc(405:bip01_spine1:100:2000)%
```

```
|set_friends(sid1:sid2:...)
|set_enemies(sid1:sid2:...)
```

Установить друзьями/врагами данного прс и указанных в списке по story\_id.

```
|play_snd(snd_name:delay=0)
```

Играть звук в голове актёра.

Параметры:

- **snd\_name** - путь к звуку относительно папки sounds
- **delay** - задержка перед проигрыванием. По умолчанию 0 - проигрываем сразу.

```
|play_snd_now (sid:snd_name)
```

Играть звук от указанного объекта. Звук играетсЯ от объекта с указанным story id, без задержки с громкостью. Указывается не имя звуковой схемы, а имя файла.

```
|hit_obj(sid, bone, power, impulse, hit_src=np:position())
```

Дать объекту, заданому story\_id, хит. Отличается тем, что может прописываться в любой кастом дате. Параметры: actor, npc,  
p[sid,bone,power,impulse,hit\_src=np:position()]

- **sid** - story\_id объекта, по которому наносится хит.
- **bone** - строка. Имя кости, по которой наносится удар.
- **power** - сила удара
- **impulse** - импульс
- **hit\_src** (необязательный параметр) - точка (waypoint), из которой по объекту наносится хит. Если не задано, то берется позиция объекта, из которого была вызвана данная функция.

```
factor_has_item(section)
```

Проверка на наличие у игрока соответствующего предмета. Проверка проходит по секции в ltx.

## Функции для работы с HUD'ом

```
disable_ui_elements(...), enable_ui_elements(...)
```

Отключение/включение элементов HUD'a.

Параметры:

- **weapon** - спрятать/показать руки с оружием;
- **input** - отключить/включить клавиатуру;
- **hud** - спрятать/показать индикаторы на экране;
- **all** - отключить/включить все элементы.

Пример:

```
lon_info = %=disable_ui_elements(weapon:input)%
```

Есть также сокращенные варианты: `disable_ui`, `enable_ui` (вызываются без скобок и параметров).

Аналогичны вызовам `disable_ui_elements(all)`, `enable_ui_elements(all)` соответственно.

Пример:

```
lon_info = %=enable_ui%
```

## Функция запуска camera\_effector'a

```
run_cam_effector(имя_файла)
```

- `имя_файла` (указывается без расширения) - это имя анимационного файла (с расширением `anm`) из папки `gamedata\anim\camera_effects\`

Пример:

```
lon_info = %=run_cam_effector(prison_0)%
```

## Функция запуска постпроцесса

Есть 2 функции для работы с постпроцессами:

```
run_postprocess(file_name:id:loop)
```

- запуск постпроцесса.

- **file\_name** - имя файла постпроцесса (без расширения) из папки *gamedata\anim*s. Указывается без расширения.
- **id** - номер постпроцесса. Задается опционально. Используется в *stop\_postprocess*.
- **loop** - (true/false) определяет заикленность постпроцесса. Опциональный параметр. По умолчанию false.

```
stop_postprocess(id)
```

Принудительная остановка постпроцесса.

- **id** - номер постпроцесса заданный в *run\_postprocess*.

## Функции для работы с инвентарём

```
drop_actor_inventory(имя_пути)
```

Выбрасываем все предметы из инвентаря актера в первую точку заданного пути. Пример:

```
on_info = %=drop_actor_inventory(drop_point)%
```

## Настройка отношений

### Функции

Установить уровень отношения фракции к игроку на определенном уровне игры:

```
=set_level_faction_community(faction:level_name:relation)
```

Параметр	Описание
faction	Имя фракции (bandit, stalker, duty, csky, monster, freedom и т.д.)
level_name	Имя уровня (escape, garbage, agroprom и т.д.)
relation	отношение к игроку (friend, enemy, neutral)

Пример:

```
%=set_level_faction_community(stalker:escape:neutral)%
```

## Проверки

Проверка отношения группировки к игроку лично:

```
=is_faction_enemy_to_actor(faction)
=is_faction_neutral_to_actor(faction)
=is_faction_friend_to_actor(faction)
```

Пример проверки — враждебна ли нам группировка сталкеров:

```
{=is_faction_enemy_to_actor(stalker)}
```

Проверка отношения группировки к группировке игрока:

```
=is_factions_enemies(faction)
=is_factions_friends(faction)
=is_factions_neutrals(faction)
```

## Пример

```
[[logic]
|active = remark
|relation = enemy
|
|[remark]
|combat_ignore_cond = true
|combat_ignore_keep_when_attacked = true
|anim = hide_na
|on_info = {+lim_condition5} %=hit_npc_from_actor =make_actor_visible_to_npc%
|следующая@секция
```

Источник — «[https://xray-engine.org/index.php?title=Передача\\_параметров\\_в\\_функции&oldid=779](https://xray-engine.org/index.php?title=Передача_параметров_в_функции&oldid=779)»

Категория:

A-Life

- 
- Страница изменена 25 марта 2018 в 15:57.
  - К этой странице обращались 4891 раз.
  - Содержимое доступно по лицензии GNU Free Documentation License 1.3 или более поздняя (если не указано иное).

