

Оглавление

1. Задания
 - 1.1. Создание задания
 - 1.2. Тексты задания
 - 1.3. Функции для работы с заданиями
2. Персонажи
 - 2.1. Создание профиля персонажа
 - 2.2. Создание описания персонажа
 - 2.3. Создание спаун-секции
3. Диалоги
 - 3.1. Создание диалога
 - 3.2. Подключение диалога персонажу
 - 3.3. Перекрытие стартового диалога из секции [meet]

1. Задания

Игроку можно выдать задание. Для этого нужно задать параметры задания, написать его заголовок и текст, и выдать его при помощи функции.

1.1. Создание задания

Параметры задания описываются в файлах «gamedata\configs\misc\tm_<имя_уровня>.ltx». Для добавления нового задания необходимо:

- a) Создать секцию параметров задания [<идентификатор_задания>].
- b) Добавить в эту секцию необходимые ключи параметров задания (**жирным** выделены обязательные параметры):

type	Тип задания по геймплею. На данный момент может быть только storyline .
task_type	Тип задания по важности. Может быть storyline либо additional . По умолчанию – additional .
target_cond	Проверка на возможность выдать задание. Если этот кондлист возвращает true – задание может быть выдано. По умолчанию – true .
icon	Иконка на карте (например, ui_mt_image_attack_tower)
prior	Приоритет задания относительно других заданий. Чем выше это число, тем приоритетней задание. По умолчанию – 0 .
name	Идентификатор строки, содержащей заголовок задания. По умолчанию - NO_TEXT. Поддерживает кондлист.
descr	Идентификатор строки, содержащей текст задания. По умолчанию - NO_TEXT. Поддерживает кондлист.
spot	Идентификатор мапспота, которым задание будет обозначено на карте. Значение по умолчанию зависит от параметров type и task_type.
target	Указание на цель задания. Может задаваться как story_id объекта, либо как работа смарт-террейна, на которой находится необходимый объект, в формате <имя_секции_logic_работы>, <идентификатор_смарт-террейна>. Поддерживает кондлист.
condlist_0 condlist_1 ...	Группа параметров, позволяющая задавать условия выполнения, провала или отмены задания. Число в имени ключа позволяет задать несколько последовательно пронумерованных (начиная с нуля) параметров. Кондлист каждого параметра может возвращать следующие значения: complete (выполнить задание), fail (провалить задание), reversed (отменить задание).

on_init	Кондлист, позволяющий выполнить какое-либо действие при выдаче задания.
on_complete	Кондлист, позволяющий выполнить какое-либо действие при выполнении задания.
on_reversed	Кондлист, позволяющий выполнить какое-либо действие при отмене задания.
on_fail	Кондлист, позволяющий выполнить какое-либо действие при провале задания.
reward_money	Награда деньги, которая выдается после выполнения задания. Измеряется в числе. Поддерживает кондлист.
reward_item	Награда предметы, которая выдается после выполнения задания. Поддерживает кондлист.

1.2. Тексты задания

Для добавления заголовка задания необходимо в файл «gamedata\configs\text\<локализация>\st_quests_<имя_уровня>.xml» добавить строки:

```
<string id="<идентификатор_задания>_name">
  <text><текст_заголовка></text>
</string>
```

Для добавления текста задания необходимо в тот же файл добавить строки:

```
<string id="<идентификатор_задания>_text">
  <text><текст_задания></text>
</string>
```

Конструкции <идентификатор_задания>_name и <идентификатор_задания>_text прописываются в параметры **name** и **text** задания соответственно.

1.3. Функции для работы с заданиями

Для того, чтобы выдать задание, используется следующая функция эффекта:

```
give_task(<идентификатор_задания>)
```

2. Персонажи

Для создания персонажа необходимо создать его профиль, описание и спаун-секцию.

2.1 Создание профиля персонажа

Чтобы создать профиль персонажа, необходимо в файл «gamedata\configs\gameplay\npc_profile.xml» добавить строки:

```
<character id="<идентификатор_профиля>">
  <class><идентификатор_класса></class>
</character>
```

Идентификатор профиля конструируется как <префикс_уровня>_<идентификатор_сцены>_<группировка_персонажа>_<описательное_название>, например, zat_b7_stalker_raider_leader. Во избежание путаницы, класс персонажа указывается такой

же, как идентификатор профиля. Класс персонажа используется для связи профиля и описания персонажа.

2.2. Создание описания персонажа

С одним профилем персонажа может быть ассоциировано несколько описаний персонажей. Это позволяет, добавив в игру несколько персонажей с одним и тем же профилем, получить в результате персонажей с различным набором вещей, разным внешним видом (visual), и т.д. Для того чтобы создать описание персонажа, необходимо в файл «gamedata\configs\gameplay\character_desc_<имя_уровня>.xml» добавить следующую конструкцию:

```
<specific_character id="<идентификатор_описания>" team_default = "1">
  <name><идентификатор_строки_имени></name>
  <icon><идентификатор_текстуры_иконки></icon>
  <map_icon x="1" y="0"></map_icon>
  <bio><идентификатор_строки_биографии></bio>

  <class><идентификатор_класса></class>
  <community><группировка_персонажа></community> <terrain_sect>stalker_terrain</terrain_sect>

  <rank><ранг></rank>
  <reputation><репутация></reputation>

  <money min="<минимум_денег>" max="<максимум_денег>" infinitive="<восполняемость>" />

  <visual><внешний_вид></visual>
  <snd_config><папка_озвучки></snd_config>

  <supplies>
    [spawn] \n
    <предмет> \n
    <предмет> \n
  #include "gameplay\character_items.xml"
  #include "gameplay\character_food.xml"
  #include "gameplay\character_drugs.xml"
  </supplies>

  #include "gameplay\character_criticals_4.xml"

  <start_dialog><идентификатор_стартового_диалога></start_dialog>
  <actor_dialog><идентификатор_диалога_актёра></actor_dialog>
  <actor_dialog><идентификатор_диалога_актёра></actor_dialog>
  #include "gameplay\character_dialogs.xml"
</specific_character>
```

Описание элементов конструкции:

<идентификатор_описания>	<p>Во избежание путаницы идентификатор описания:</p> <ul style="list-style-type: none"> a) аналогичен идентификаторам класса и профиля в случае, если с профилем ассоциируется только одно описание; b) конструируется как <идентификатор_профиля>_default_<номер> (начиная с нуля), если с профилем ассоциируется несколько описаний.
<идентификатор_строки_имени>	<p>Идентификатор строки имени конструируется, как <идентификатор_описания>_name. Строка имени персонажа добавляется в файл «gamedata\configs\text\<локализация>\st_characters.xml» конструкцией:</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre><string id="<идентификатор_строки_имени>"> <text><текст_имени></text> </string></pre> </div> <p>Также, имя можно сгенерировать, поставив в описание персонажа вместо идентификатора строки имени конструкцию GENERATE_NAME <имя_группировки>.</p>
<идентификатор_текстуры_иконки>	Идентификатор текстуры иконки берётся из файла «gamedata\configs\ui\textures_descr\ui_npc_unique.xml». Эта иконка видна во время диалога с персонажем.
<идентификатор_строки_биографии>	На данный момент в игре не используется, так что это поле можно оставить пустым.
<идентификатор_класса>	Идентификатор класса, который связывает описание персонажа с соответствующим профилем персонажа.
<группировка_персонажа>	Имя группировки, к которой принадлежит данный персонаж.
<ранг>	<p>Число, определяющее относительную «крутизну» персонажа:</p> <ul style="list-style-type: none"> 0 - 300 – новичок (novice) 301 - 600 – опытный (experienced) 601 - 900 – ветеран (veteran) 901 - 999 – мастер (master)
<репутация>	На данный момент в игре не используется, так что это поле можно оставить пустым.
<минимум_денег> <максимум_денег> <восполняемость>	<p>Количество денег, имеющихся у персонажа, будет выбрано случайным образом от указанного минимума до указанного максимума.</p> <p>Восполняемость может принимать значения 0 или 1. В последнем случае количество денег у персонажа будет восполняться после каждой сделки, но каждая отдельная сделка будет ограничена количеством денег персонажа.</p>
<внешний_вид>	<p>Путь к файлу «*.ogf» от папки «gamedata\meshes», содержащему внешний вид (visual) персонажа.</p> <p>Указывается без начальной «обратной косой», без точки и без расширения файла.</p> <p>От внешнего вида зависит бронированность персонажа.</p>

<папка_озвучки>	Путь к папке универсальной озвучки персонажа от папки «gamedata\sounds». Имеет вид characters_voice\human_<номер>\<группировка>. В этой конструкции <номер> может быть 01 либо 02 (обычные голоса), или же 03 (приглушённый голос из-за дыхательной маски, противогаза либо купола).
<предмет> \n <предмет> \n	При помощи конструкций <предмет> задаётся список предметов, которые должны быть у данного персонажа. Конструкции имеют следующий вид: <идентификатор_предмета>< = количество><, навес> При этом, < = количество> и <, навес> необязательны. Параметр навес имеет смысл только для оружия и может принимать следующие значения: scope (оптический прицел) silencer (глушитель) launcher (подствольный гранатомёт) Можно добавить несколько конструкций <, навес>, но все они должны стоять после конструкции < = количество>. В конце всей вышеописанной конструкции обязательно должны стоять символы \n (перенос строки). Пример: wpn_abakan = 1, scope, launcher
#include "gameplay\character_items.xml"	Эта и все остальные строки #include должны стоять в начале строки (без отступов перед ней). Она добавляет в список предметов персонажа бинокль, фонарь, детектор и, с долей вероятности, гитару.
#include "gameplay\character_food.xml"	Эта строка добавляет в список предметов персонажа хлеб, колбасу и водку.
#include "gameplay\character_drugs.xml"	Эта строка с определённой долей вероятности добавляет в список предметов персонажа бинты и аптечку.
<идентификатор_стартового_диалога>	Идентификатор диалога, являющегося для персонажа стартовым. Если этот параметр (вместе с тэгами <start_dialog></start_dialog>) не будет указан – будет использован стандартный стартовый диалог.
<идентификатор_диалога_актёра>	Идентификатор диалога актёра. Можно указать несколько диалогов актёра (каждый – в тэгах <actor_dialog></actor_dialog>).
#include "gameplay\character_dialogs.xml"	