# 2. Общая часть.

[2.1. Работа с xml-файлами.](#_Xml-файлы)

[2.2 Система string\_table](#_Система_string_table)

[2.3 Настройка профилей персонажей.](#_Профили_персонажей)

[2.4 State manager](#_State_manager)

[2.4.1 Основные положения](#_1._Основные_положения:)

[2.4.2 Запрещенные функции](#_2._Запрещенные_функции.)

[2.4.3 Правила создания дескрипторов](#_3._Правила_создания_дескрипторов)

[2.4.4 Список дескрипторов](#_4._Список_дескрипторов.)

[2.5. Мануал по сбору билда](#_2.5._Мануал_по_сбору билда)

[2.6. Префиксы уровней](#_2.6._Сокращенные_имена_уровней для )

[2.7. Озвучка диалогов.](#_2.7._Озвучка_диалогов)

[2.8. Список коммьюнити монстров](#_2.8._Список_коммьюнити_монстров)

[2.9. Важные моменты](#_2.9._Важные_моменты)

[2.9.1. Prefetch текстур сталкеров](#_2.9.1._Prefetch_текстур_сталкеров.)

[2.9.2. Правила спавна](#_2.9.2._Правила_спавна.)

[2.9.3. AI-nodes для монстров.](#_2.9.3._AI-nodes_для_монстров.)

[2.9.4. Расстановка источников света](#_2.9.4._Расстановка_динамических_ист)

[2.9.5. Звуковые темы в группе.](#_2.9.5._Звуковые_темы_в группе)

## 2.1. Работа с xml-файлами.

Файлы XML можно всего редактировать в обычном текстовом редакторе, типа notepad либо FAR. Подойдет и встроенный редактор sourceSafe. Но для удобства и уменьшения количества ошибок лучше использовать что-то с подсветкой синтаксиса. Например, Visual Studio. К тому же он неплохо интегрирован и облегчает работу с SourceSafe.

Теги в XML бывают двух типов: парные и одиночные. Парные теги обязательно должны закрываться соответствующим тегом. Например:

<bio>Детальная информация отсутствует.</bio>

Одиночные теги не закрываются. Например:

<?xml version='1.0' encoding="windows-1251"?>

Все, что относится к парному тегу, находится внутри него, между открывающим и закрывающим тегом. При получении определенного инфопоршна запустить квест с id = bar\_bar\_hunter\_quest\_start

<info\_portion id="bar\_bar\_hunter\_quest\_start">

<task>bar\_hunter\_rifle</task>

<action>level\_tasks.notify\_new\_task</action>

</info\_portion>

Для комментариев используется особый тэг:

<!-- тут текст комментария -->

Следует также понимать, что если вы внутри тэга используете знак “--“, то xml ожидает что это закрывающий тег и не найдя знака > будет жутко ругаться. Поэтому если вы хотите сделать что-то вроде:

<!-- ------------ тут текст комментария ------------------- -->

То у вас ничего не получиться, следует делать так:

<!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* тут текст комментария \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

И последнее. После окончания редактирования xml, если вы не уверены, что не допустили ошибки в структуре и xml прочитается движком нормально, это можно проверить следующим образом: открыть его через Интернет эксплорер. Если он не сможет открыть xml, то и движок не сможет. К тому же он укажет сам в какой строке и на какой позиции была допущена ошибка.

P.S. В \gamedata\config\system.ltx перечисляются все xml, которые загружает движок.

## 2.2. Система string\_table

1. Следует избегать указания текста в игре напрямую. Для этого создана хитроумная система string\_table.

2. Логика работы предельно проста. Каждый текст, перед выводом на экран прогоняется через особую таблицу. В таблице указаны соответствия входящий\_текст – исходящие\_тексты. То есть для каждого входящего текста может быть несколько исходящих. По идее по одному исходящему тексту для каждого языка. Если для текущего входящего текста соответствия не найдено, то на экран выводится он. Таким образом, если вы в диалоге напишите фразу русским языком, то она так и будет выводиться русским языком даже в английской версии игры.

3. Поэтому везде, где нужно указать текст, который выводится на экран, вместо текста указывается уникальный идентификатор, по которому из стрингтейбла в зависимости от версии игры (английская, русская) выберется необходимый текст. Если же текст не будет найден, то на экран выведется уникальный идентификатор, и по нему можно будет увидеть, что не так и пофиксить.

4. Все стрингтейблы задаются в папке GameData\Config\Text. По этому пути созданы несколько папочек, по одной на каждый язык. Сейчас там есть лишь русский и английский. Внутри каждой папочки абсолютно идентичный набор файлов (в идеале).

5. В каждом файле стрингтейбла описываются фразы. Например:

<string id="escape\_factory\_assault\_hello\_2">

<text>Ты, наверное, тот малый, о котором Волк говорил? Вообщем смотри, бандитов в принципе немного. Вот только мы своей толпой тудя врядли тихо проберемся. На втором этаже снайпер сидит, все подходы пасет. Если бы не он - мы бы их как следует наказали.</text>

</string>

Таким образом уникальному идентификатору escape\_factory\_assault\_hello\_2 ставится в соответствие текст фразы. А там, где нам нужно ставить данную фразы – мы указываем ее идентификатор.

Аналогичная запись в англоязычном файле:

<string id="escape\_factory\_assault\_hello\_2">

<text>You are supposed to be that guy Wolf has been talking about? Well it’s up to you now, basically there are not so many bandits there. Although we can hardly penetrate it together. Sniper has settled on the first floor staring at all passing nearby. If it were not for him – we would punish them swiftly.</text>

</string>

В диалоге же соответствующая запись задается так:

<phrase id="2">

<text>escape\_factory\_assault\_hello\_2</text>

</phrase>

По поводу описания диалогов более подробно смотрите соответствующий файл.

6. Файлы со стрингтейблами диалогов называются следующим образом: GameData\Config\Text\rus\stable\_dialogs\_название\_уровня.xml (или \egn\ для английской версии)

Файлы со стрингтейблами тасков (заданий и подзаданий квестов) называются следующим образом:

GameData\Config\Text\rus\string\_table\_tasks\_название\_уровня.xml

Файлы со стрингтейблами иноформации в энциклопедии называются следующим образом:

GameData\Config\Text\rus\string\_table\_enc\_название\_уровня.xml

## 2.3. Профили персонажей

Устанавливаются и настраиваются в файлах \gamedata\config\gameplay\character\_desc\_название группы.xml

Профиль персонажа – это собственно полное описание персонажа, включающее имя, группировку, ранг, обмундирование, описание и внешний вид. Используется для типизированной простановки персонажей, что экономит и время и позволяет достичь определенной рандомности от игры к игре.

Профиль создается следующим образом:

В gamedata\config\gameplay\npc\_profile.xml создается уникальный идентификатор типа персонажа.

<character id="stalker\_novice">

<class>Stalker\_novice</class>

</character>

Класс персонажа – это уникальный идентификатор класса, по которому будет браться конкретный профиль для данного персонажа (ссылаться на определенный файл с профилями)

В character\_desc\_название класса/xml…. Описываются конкретные профили. Они умышленно разбиты на файлы по группировкам, чтобы облегчить навигацию и восприятие. Обычно на каждый класс создается несколько профилей. Некоторые из них детерминированы (имеют ФИО, уникальную биографию и тд) Некоторые – базовые. Базовые отмечены флажком team\_default = "1". Логика работы такая: При создании персонажа считывается его тип, по типу получается класс. Затем составляется список всех доступных профилей для данного класса и из них рандомно выбирается один. Уникальные детерминированные профили (без team\_default = "1" ) Будучи выбраны однажды – больше не выбираются. Профиль единственный в своём классе должен иметь team\_default = "1".

<specific\_character id="stalker\_novice\_default1" team\_default = "1">

Имя персонажа.

<name>Неизвестный новичок</name>

Иконка, которая отображается при общении. Берется с текстуры ui\_icons\_npc. Отсчитывается с верхнего левого угла но не по пикселям, а по позициям (равным одной иконке)

<icon x="20" y="2"></icon>

Иконка, отображаемая на карте. Берется с текстуры ui\_icons\_map. Отсчитывается с верхнего левого угла но не по пикселям, а по позициям (равным одной иконке)

<map\_icon x="0" y="0"></map\_icon>

Биография персонажа. Отображается в ПДА и по идее должна отображаться при общении.

<bio>Новичок. Недавно в Зоне. Детальная информация отсутствует.</bio>

Класс сталкера. То, что связывает этот профиль с идентификатором персонажа

<class>Stalker\_novice</class>

Группировка. По ней определяются отношения.

<community>stranger</community>

Ранг персонажа

<rank>20</rank>

Репутация персонажа

<reputation>0</reputation>

<panic\_threshold>0.1</panic\_threshold>

Значение может изменяться от 0 до 1. Если функция оценки вероятности победы падает ниже указанного порога, то сталкер впадает в панику. Если вы установится 0 - в панику он впадать вообще не будет. Если в 1 - то будет бегать от крысы.

Модель персонажа.

<visual>actors\novice\green\_stalker\_1</visual>

То, что спавнится у персонажа в инвентаре.

[spawn]

<section> [ = <count>[, <parameters> ] ]

где

<section> = название секции в LTX

<count> = количество предметов, integer > 0 =1, если не задано

<parameters> = [<addons>] [<condition>] [<probability>]

<addons> = <addon> [<addons>]

<addon> = {scope | silencer | launcher} – соответственно прицел, глушак, подствольник

<condition> = {cond=float\_value} – состояние вещи, изменяется от 0 (сломаный), до 1 (новый)

<probability> = {prob=float\_value} – вероятность спауна предметов. Тоже от 0 до 1.

важно: если задаем дополнительные параметры, нужно обязательно задавать количество предметов

например:

<supplies>

[spawn]

device\_torch

wpn\_ak74 = 1, silencer

wpn\_ak74 = 4, launcher cond=0.8

wpn\_ak74 = 1, scope prob=0.3

wpn\_ak74 = 1, scope launcher silencer cond=0.5

wpn\_fort = 8, cond=0.75 prob=0.5

</supplies>

Стартовый диалог персонажа. Более подробно в описании диалогов. <start\_dialog>hello\_dialog</start\_dialog>

Диалоги, доступные актеру. Более подробно в описании диалогов.

<actor\_dialog>general\_pda\_talk</actor\_dialog>

<actor\_dialog>escape\_lager\_access</actor\_dialog>

<actor\_dialog>escape\_factory\_assault\_talk</actor\_dialog>

<actor\_dialog>escape\_talk\_about\_escape</actor\_dialog>

</specific\_character>

Для некоторых персонажей требуется четкое соответствие тип персонажа – персонаж. Например для уникальных квестовых или сюжетных персонажей. В таком случае делаем так, чтобы соответствующему классу персонажа соответствовал только один профиль. Важные элементы выделены большим шрифтом.

<character id="stalker\_esc\_wolf">

**<class>stalker\_esc\_wolf</class>**

<specific\_character>stalker\_esc\_wolf</specific\_character>

</character>

<specific\_character id="stalker\_esc\_wolf" **team\_default = "1"**>

<name>Сталкер "Волк"</name>

<icon x="16" y="4"></icon>

<map\_icon x="0" y="0"></map\_icon>

<bio>Главарь лагеря новичков.</bio>

**<class>stalker\_esc\_wolf</class>**

<community>stranger</community>

<rank>30</rank>

<reputation>0</reputation>

<visual>actors\neytral\stalker\_neytral\_balon\_1</visual>

<supplies>

[spawn] \n

wpn\_pb \n

ammo\_9x18\_pbp \n

af\_medusa \n

wpn\_ak74 \n

ammo\_5.45x39\_fmj \n

device\_torch \n

</supplies>

<start\_dialog>escape\_lager\_volk\_hello</start\_dialog>

<actor\_dialog>escape\_lager\_volk\_job</actor\_dialog>

<actor\_dialog>escape\_lager\_volk\_info</actor\_dialog>

<actor\_dialog>escape\_lager\_volk\_dynamite\_done</actor\_dialog>

</specific\_character>

Следует обратить внимание, что несмотря на то, что класс волка отличается от класса сталкера-новичка – он относится к той же группировке.

## 2.4. State manager

### 2.4.1. Основные положения:

Схема создана для централизованного управления телом сталкеров. Она отвечает за положения тела, оружия, анимацию, тип перемещения и тд и тп. Всего схема контролирует семь свойств: оружие, движение, ментальное состояние, движковое состояние тела, направление взгляда, анимационное состояние тела, рандомные анимации.

Все эти семь свойств объединяются в дескриптор состояний. Дескрипторы создаются в файле \gamedata\Scripts\state\_lib.script.

Для того, чтобы перевести персонажа в новое состояние, достаточно просто вызвать функцию:

state\_mgr.set\_state(npc, state\_name, callback, timeout, target)

state\_name - состояние, которое включает в себя режим оружия, набор анимаций и т.д.

callback - таблица с полями object и func [имена взять как в move\_mgr]

timeout = n msec [через какое время вызвать коллбек]

Если время не задано, коллбек вызывается после окончания проигрывания \_вступительной\_ анимации, иначе - когда истек timeout со старта \_зацикленной\_ анимации (или неанимированного idle).

target - таблица, в которой задано одно из полей:

look\_position = координаты

look\_object = объект

Например:

state\_mgr.set\_state(self.object, "wait", { obj = self, func = self.time\_callback }, 5000, {look\_object = db.actor})

В данном случае “wait” – представлен следующим дескриптором:

wait = { weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

direction = "object",

animstate = "sit",

animation = "eating"

}

### 2.4.2. Запрещенные функции.

Следующие функции запрещено вызывать непосредственно из скрипта:

Add\_animation()

Clear\_animations()

Set\_item()

Set\_mental\_state()

Set\_body\_state()

Set\_movement\_type()

Set\_sight()

### 2.4.3. Правила создания дескрипторов

3.1. Оружие (weapon):

“strapped” - оружие висит на плече

“unstrapped” - оружие находится в руках.

Nil - безразлично

3.2. Движение (movement):

move.stand – стоять

move.walk – идти

move.run - бежать

Nil - безразлично

3.3. Ментальное состояние (mental):

Anim.free – спокойное состояние

Anim.danger – возбужденное состояние

Nil - безразлично

3.4. Движковое положение тела (bodystate)

Move.standing – стоим

Move.crouch – присели. Возможно только при mental = anim.danger

Nil - безразлично

3.5. Not implemented yet

3.6. Анимационное состояние тела (animstate):

“stand” - стоим

“sit” – сидим (только при weapon = “strapped”)

“sleep” – спим (только при weapon = “strapped”)

“wounded” - ранены (только при weapon = “strapped”)

3.7. Рандомные анимации

Тут пока предоставлены далеко не все. В принципе это самый легкорасширяемый раздел, поэтому при необходимости сюда можно добавлять новые анимации.

“idle” стоим, чешем яйца, ожидаем чего-то.

“caution” прислушиваемся

“search” поиск чего-то под ногами

“hide” нычкуемся (юзать в крауч денжере)

“hello” приветствуем рукой

“refuse” отказ

“claim” вымогательство

“backoff” вали отседа!

“backoff2” вали отседа2!

“sleeping” спим (только при animstate = “sleep”)

“eating” жрем кавбасу (только при animstate = “sit”)

### 2.4.4. Список дескрипторов.

-- Дефолтовый

idle = { weapon = nil,

movement = nil,

mental = nil,

bodystate = nil,

animstate = "stand",

animation = nil

},

-- Ходячие состояния

walk = { weapon = "strapped",

movement = move.walk,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

run = { weapon = "strapped",

movement = move.run,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

sprint = { weapon = "strapped",

movement = move.run,

mental = anim.panic,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

patrol = { weapon = "unstrapped",

movement = move.walk,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

patrol\_fire = { weapon = "fire",

movement = move.walk,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

raid = { weapon = "unstrapped",

movement = move.walk,

mental = anim.danger,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

raid\_fire = {weapon = "fire",

movement = move.walk,

mental = anim.danger,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

sneak = { weapon = "unstrapped",

movement = move.walk,

mental = anim.danger,

bodystate = move.crouch,

animstate = "stand",

animation = nil

},

sneak\_fire = { weapon = "fire",

movement = move.walk,

mental = anim.danger,

bodystate = move.crouch,

animstate = "stand",

animation = nil

},

assault = { weapon = "unstrapped",

movement = move.run,

mental = anim.danger,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

assault\_fire = { weapon = "fire",

movement = move.run,

mental = anim.danger,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

rush = { weapon = "unstrapped",

movement = move.run,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

-- Стоячие состояния

wait = { weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "idle"

},

wait\_na = { weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

guard = { weapon = "unstrapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "idle"

},

guard\_na = {weapon = "unstrapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

guard\_fire = { weapon = "fire",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

threat = { weapon = "unstrapped",

movement = move.stand,

mental = anim.danger,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

threat\_fire = { weapon = "fire",

movement = move.stand,

mental = anim.danger,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

threat\_sniper\_fire = { weapon = "sniper\_fire",

movement = move.stand,

mental = anim.danger,

bodystate = move.standing,

animstate = "stand",

animation = nil

},

hide = { weapon = "unstrapped",

movement = move.stand,

mental = anim.danger,

bodystate = move.crouch,

animstate = "stand",

animation = "hide"

},

hide\_na = { weapon = "unstrapped",

movement = move.stand,

mental = anim.danger,

bodystate = move.crouch,

animstate = "stand",

animation = nil

},

hide\_fire = { weapon = "fire",

movement = move.stand,

mental = anim.danger,

bodystate = move.crouch,

animstate = "stand",

animation = nil

},

hide\_sniper\_fire = { weapon = "sniper\_fire",

movement = move.stand,

mental = anim.danger,

bodystate = move.crouch,

animstate = "stand",

animation = nil

},

caution = { weapon = nil,

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "caution"

},

choose = {weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "choosing"

},

press = {weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "press"

},

ward = {weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "warding"

},

search = {weapon = nil,

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "poisk"

},

salut = {weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "salut"

},

-- Сидячие состояния

sit = { weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "sit",

animation = "sit"

},

sit\_knee = {weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "sit\_knee",

animation = "sit\_knee"

},

sit\_ass = { weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "sit\_ass",

animation = "sit\_ass"

},

sleep = { weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "sleep",

animation = "sleeping"

},

eat\_bread = { weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "sit\_knee",

animation = "eat\_bread"

},

hello = { weapon = "unstrapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "hello",

reset = true

},

hello\_wpn = {weapon = "unstrapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "hello",

reset = true

},

refuse = { weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "refuse",

reset = true

},

claim = { weapon = "unstrapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "claim",

reset = true

},

backoff = { weapon = "unstrapped",

movement = move.stand,

mental = anim.danger,

bodystate = move.standing,

animstate = "stand",

animation = "backoff",

reset = true

},

backoff2 = {weapon = "unstrapped",

movement = move.stand,

mental = anim.danger,

bodystate = move.standing,

animstate = "stand",

animation = "backoff2",

reset = true

},

suicide = { weapon = "unstrapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "suicide",

reset = true

},

punch = { weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "punch",

reset = true

},

hide\_s\_left = {weapon = "unstrapped",

movement = move.stand,

mental = anim.danger,

bodystate = move.crouch,

animstate = "stand",

animation = "hide\_s\_left",

reset = true

},

hide\_s\_right = {weapon = "unstrapped",

movement = move.stand,

mental = anim.danger,

bodystate = move.crouch,

animstate = "stand",

animation = "hide\_s\_right",

reset = true

},

dynamite = {weapon = "strapped",

movement = move.stand,

mental = anim.danger,

bodystate = move.crouch,

animstate = "stand",

animation = "dynamite"

},

binocular = {weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "binocular"

},

hide\_rac = {weapon = "unstrapped",

movement = move.stand,

mental = anim.danger,

bodystate = move.crouch,

animstate = "stand",

animation = "cr\_raciya"

},

wait\_rac = {weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "raciya"

},

guard\_rac = {weapon = "unstrapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "raciya"

},

-- Раненый

wounded = { weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "wounded",

animation = nil

},

wounded\_heavy = { weapon = "drop",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "wounded",

animation = "wound\_heavy"

},

help\_me = { weapon = "strapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "wounded",

animation = "help\_me"

},

-- Пси раненый

psy\_pain = {weapon = "drop",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "psy\_armed"

},

psy\_armed = {weapon = "unstrapped",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "psy\_armed"

},

psy\_shoot = {weapon = "fire",

movement = move.stand,

mental = anim.free,

bodystate = move.standing,

animstate = "stand",

animation = "psy\_shoot"

},

Еще есть набор анимаций у бармена, но так как они уникальны и уже все проставлены, то здесь они не указаны.

## 2.5. Мануал по сбору билда

1. Подготовить уровень, проверить на глюки, собрать симуляцию

2. Запустить процесс билда уровня добавив ключ "-build"

Пример билда эскейпа: "@\bin\xr\_3da.exe -build -start server(l01\_escape/single/alife) client(localhost)"

3. После того как уровень загрузится (это процесс продолжительный, предупреждаю сразу;)), набрать в консоли:

al\_switch\_distance 20000 (перевести весь уровень в онлайн)

4. На диске X:\ появилась директория xray\_build - это то, что сбилдилось. Переписывайте на диск S:\ в папку, куда хотите собрать билд.

5. Внимание! Билдится только то, что загрузилось в память, а грузится далеко не все, что есть на уровне.

Поэтому необходимо вручную дописать нехватающие файлы из нижеследующих папок:

S:\GameData\anims

S:\GameData\config

S:\GameData\meshes

S:\GameData\scripts (из подпапок ничего не нужно писать!!!)

S:\GameData\sounds\characters\_voice

6. Из самого свежего билда взять файл fsgame.ltx

7. Теперь осталось дописать текущие DLL, EXE, PDB, BPL, LTX (из директории s:\bin), сделать батник и запускать версию

## 2.6. Сокращенные имена уровней для использования в названиях объектов, расположенных на этих уровнях, в качестве префикса:

|  |  |
| --- | --- |
| **esc** | L01\_Escape |
| **gar** | L02\_Garbage |
| **agr** | L03\_Agroprom |
| **agu** | L03u\_Agroprom |
| **val** | L04\_[**DarkValley**](http://tiger/x-wiki/DarkValley) |
| **dar** | L04u\_[**LabX18**](http://tiger/x-wiki/LabX18) |
| **bar** | L05\_Rostok |
| **yan** | L06\_Yantar |
| **yau** | L06U\_[**LabX16**](http://tiger/x-wiki/LabX16) |
| **mil** | L07\_Military |
| **cit** | L08\_[**DeadCity**](http://tiger/x-wiki/DeadCity) |
| **swa** | L09\_Swamp |
| **rad** | L10\_Radar |
| **bun** | L10u\_Bunker |
| **pri** | L11\_Pripyat |
| **aes** | L12\_Stancia |
| **sar** | L12u\_Sarcofag |
| **mon** | L12u\_Monolith |
| **gen** | L13\_Generators |
| **lab** | L13u\_[**MainLab**](http://tiger/x-wiki/MainLab) |

## 2.7. Озвучка диалогов

Озвучка диалогов происходит в момент, когда персонаж говорит фразу в окне диалогов. В каждый момент времени может отыгрываться только одна фраза, т.е. если в данный момент проигрывается озвучка сказанной фразы, и в этот момент собеседник гороврит другую фразу, то первая фраза останавливается и начинает проигрываться вторая.

Для того, чтобы фраза озвучивалась, необходимо создать звуковой файл с именем, идентичным идентификатору фразы в DialogEditor-e и разместить ее в папке S:\gamedata\sounds\dialogs.

Напр. hello\_dialog\_2.ogg

Если для какой-либо фразы не будет найден соответствующий звуковой файл, то она озвучиваться не будет.

## 2.8. Список коммьюнити монстров

Вот список коммьюнити монстров, которые указываются в гулагах для них

"bloodsucker"

"boar"

"burer"

"cat"

"chimera"

"controller"

"pseudodog"

"dog"

"flesh"

"fracture"

"poltergeist"

"pseudo\_gigant"

"snork"

## 2.9. Важные моменты

Ниже будет приведен список тех моментов и правил, про которые не стоит забывать, но некуда занести в документацию ☺

### 2.9.1. Prefetch текстур сталкеров.

При спавне сталкеров и монстров, машина подгружает их текстуры и сильно тормозит при этом. Чтобы этого не происходило – надо их подгружать сразу при загрузке уровня. Делается это следующим образом:

В файле S:\Gamedata\config\prefetch\prefetch\_single.ltx заносите в список visual тех сталкеров, которые у вас на уровне подгружаются при спавне.

### 2.9.2. Правила спавна.

Нельзя использовать спаун рядом с игроком. Особенность инструментария такова, что после спауна проходит до 5-10 секунд, когда сталкеры перейдут в необходимые состояния. Поэтому никогда нельзя проектировать сцены, где предполагается, что когда игрок зайдет в зону за следующим углом проспавнятся противники.

Проектируйте сцены, чтобы спаун сталкеров и объектов происходил не ближе радиуса появления сталкеров/предметов из офлайна в онлайн (100-200 метров).

А лучше проектировать схемы так, чтобы не нужно было что-то доспавнивать, чтобы участники сцены корректно и адекватно реагировали на вокруг происходящие события, потому как в основном, спавны используются для того, чтобы никто до этого не мог повредить сценку. Также не воспринимайте спавны, как возможность прооптимизировать количество сталкеров и монстров на уровне. Для этих целей прекрасно работает радиус онлайна. Все, что в офлайне практически не снижает производительность.

### 2.9.3. AI-nodes для монстров.

Учитывая, что случаются ситуации, когда монстры залазят в геометрию, в частности в сетку раббица, рекомендуется:

Ставить ai-nodes на некотором расстоянии от геометрии (желательно не менее 40-50 см).

### 2.9.4. Расстановка источников света

1. Все костры надо делать со статическим светом - т.е. отдавать в цепкие руки левел-дизайнеров.

2. ОЧЕНЬ аккуратно баловаться динамическим светом и особенно расставлять около него физику.

### 2.9.5. Звуковые темы в группе

Если персонажи находятся в одном и том же team, squad, group, при этом group не равен 0, то считается, что данные персонажи находятся в одной звуковой группе.

Персонажи, находящиеся в одной звуковой группе, не могут отыгрывать скриптовые звуки одновременно. Это сделано для того, чтобы избежать гомона в скоплениях сталкеров.

Как этим пользоваться: Одну звуковую группу следует ставить \_только\_сталкерам, находящимся долго в одном месте. Примером таких случаев являются посиделки у костра. Во всех остальных случаях ставьте сталкерам 0 группу.