В последнее время люди постоянно меня спрашивали как создать скрипт гулага для смартеррейнов. Чтобы постоянно не отвечать на каждое отдельное письмо, я решил написать небольшой тутор. Прежде чем начать, хотелось бы предупредить, что не являюсь экспертом в подобного рода вещах и некоторые моменты мне всё еще не ясны до конца, так что вероятно я не смогу объяснить всё правильно. Также я буду использовать ST как абревиатуру смартеррейна. Этот тутор посвящен НПС в ST, но он также сгодится и для мутантов. Предполагаю что вы знакомы с:

- луа программированием

- редактированием all.spawn

- работа с вейпоинтами(патрульными путями)

- утилитами аи

В первую очередить подумайте о своём ST, где бы его заспавнить, сколько нпс/мутантов будут в нем выполнять определенную "работу", когда ST должен активироваться/отключаться, сколько и какие состояния будет у вашего ST и тд. Думаю, наиболее эффективный способ объяснить вам как это работает на примере. Давайте предположим.. мы хотим поставить ST(smart terrain) на Кордоне для 3х бандитов.

1. Спавн ST:

[2515]

; cse\_abstract properties (основные параметры)

section\_name = smart\_terrain

name = esc\_bandits\_smart\_terrain

position = 131.02030944824,0.065616846084595,-248.9094543457

direction = 0,0,0

; cse\_alife\_object properties (параметры объекта)

game\_vertex\_id = 635

distance = 9.09999942779541

level\_vertex\_id = 363757

object\_flags = 0x==3e

custom\_data = <<END

[smart\_terrain]

type = esc\_bandits\_smart\_terrain

cond = {-infoportion}

capacity = 3

squad = 1

groups = 5

respawn = esc\_respawn\_inventory\_box\_0002

END

; cse\_shape properties (параметры шейпа объекта)

shapes = shape0

shape0:type = sphere

shape0:offset = 0,0,0

shape0:radius = 20.55957102775574

; cse\_alife\_space\_restrictor properties (параметры рестриктора)

restrictor\_type = 3

; se\_smart\_terrain properties (параметры смарттеррейна)

Эта наиболее важная часть:

type = esc\_bandits\_smart\_terrain

cond = {-infoportion}

capacity = 3

squad = 1

groups = 5

respawn = esc\_respawn\_inventory\_box\_0002</source>

type название вашего нового ST(обязательно)

cond описывает условия, которые необходимы для включения гулага(по желанию)

capacity количество мутантов/нпс, которое может вместить смарттеррейн(обязательно)

squad, groups - номер сквада и количество групп(по желанию)

respawn название тайника(синяя коробка) куда будут спавниться предметы, когда мы вызовем респавн в ST.(по желанию)

2. Спавн нпс/мутантов и назначение(биндинг) их к нашему ST: для этого мы должны добавить каждому мутанту/нпс определенную логику:

custom\_data = <<END

[smart\_terrains]

esc\_bandits\_smart\_terrain = true

END

Если таковых нпс не обнаружится то гулаг выберет себе население из числа заспавнившихся в Зоне нпс с подходящими параметрами. Даже если они на другой локации.

3. Добавляем "работу" (логика) для каждого нпс/мутанта из нашего ST(для каждого состояния). Предположии у ST их два: состояние 0 (описывает какие нпс/мутанты "работают" днем) и состояние 1 (ночью). У нас 3 бандита, определяемся:

- bandit1: walker (состояние 0) и kamp (состояние 1)

- bandit2: guard (состояние 0) и sleeper (состояние 1)

- bandit3: walker (состояние 0 и 1 <= он делает тоже и днём, и ночью)

У нас есть 3 способа добавить логику (работу) для каждого нпс/мутанта, мы будем использовать наиболее общепринятый способ, добавим логику в фаил config\misc\gulag\_escape.ltx. Она должна выглядеть примерно так:

;-- bandit1 (walker(прогуливающийся) -> состояние 0, днем)

[logic@esc\_bandits\_smart\_terrain\_bandit1\_walker]

active = walker@esc\_bandits\_smart\_terrain\_bandit1

[walker@esc\_bandits\_smart\_terrain\_bandit1]

path\_walk = bandit1\_walk

danger = danger\_condition@esc\_bandits\_smart\_terrain

def\_state\_moving1 = patrol

def\_state\_moving2 = patrol

def\_state\_moving3 = patrol

meet = no\_meet

;-- bandit1 (kamp(лагерь) -> состояние 1, ночью)

[logic@esc\_bandits\_smart\_terrain\_bandit1\_kamp]

active = kamp@esc\_bandits\_smart\_terrain\_bandit1

[kamp@esc\_bandits\_smart\_terrain\_bandit1]

center\_point = bandit\_kamp

path\_walk = bandit\_kamp\_task

;-- bandit2 (guard(охранник) -> состояние 0, днем)

[logic@esc\_bandits\_smart\_terrain\_bandit2\_walker]

active = walker@esc\_bandits\_smart\_terrain\_bandit2

[walker@esc\_bandits\_smart\_terrain\_bandit2]

path\_walk = bandit2\_walk

path\_look = bandit2\_look

danger = danger\_condition@esc\_bandits\_smart\_terrain

;-- bandit2 (sleeper(спящий) -> состояние 1, ночью)

[logic@esc\_bandits\_smart\_terrain\_bandit2\_sleeper]

active = sleeper@esc\_bandits\_smart\_terrain\_bandit2

[sleeper@esc\_bandits\_smart\_terrain\_bandit2]

path\_main = bandit2\_sleep

wakeable = false

;-- bandit3 (guard -> состояние 0 и 1, днем/ночью)

[logic@esc\_bandits\_smart\_terrain\_bandit3\_walker]

active = walker@esc\_bandits\_smart\_terrain\_bandit3

[walker@esc\_bandits\_smart\_terrain\_bandit3]

path\_walk = bandit3\_walk

path\_look = bandit3\_look

[danger\_condition@esc\_bandits\_smart\_terrain]

ignore\_distance\_corpse = 0

ignore\_distance = 0

4. Теперь нам нужно заскриптовать наш ST. Так что добавим наш код в фаил скрипта \gulag\_escape.script. Есть еще несколько моментов, которые мы должны здесь доделать (каждый из этих шагов обязателен):

- грузим логику (работу) для каждого нпс/мутанта и для каждого состояния -> function load\_job(...)

if type == "esc\_bandits\_smart\_terrain" then

t = {}

;-- "соеденительная секция" для логики, определяем ltx фаилом

t.section = "logic@esc\_bandits\_smart\_terrain\_bandit1\_walker"

;-- no idea, probably describes after what time

;-- npc will use this job again (?)

t.idle = 0

;-- no idea but i guess it's optional

t.timeout = 0

;-- пріоритет

t.prior = 100

;-- нпс будет использовать эту логику,

;-- если ST переключится в это состояние

;-- в этом случае - состояние 0 (день)

t.state = {0}

;-- Какой squad и group назначится персонажу принявшему эту работу.

t.squad = squad

t.group = groups[1]

;-- no idea what means position\_threshold

t.position\_threshold = 100

;-- описывает нпс в этом состоянии: онлайн или офлайн

;-- онлайн = истина по дефолту

t.online = true

;-- описывает рестрикторы (куда нпс могут/не могут пойти)

t.in\_rest = ""

t.out\_rest = ""

;-- ввиду особого способа присвоения работ в

;-- smart\_terrain.script вы никогда не знаете, какая работа

;-- будет использоваться каждым нпсом; если вы хотите быть уверенным

;-- что конкретный нпс взял конкретную работу, тогда

;-- вам нужно заюзать предикатную функцию; в этом слуае

;-- мы хотим чтобы эта работа присвоилась мастеру бандиту

t.predicate = function(obj\_info) return obj\_info.rank >= 900 end

table.insert(sj, t)

t = {section = "logic@esc\_bandits\_smart\_terrain\_bandit1\_kamp",

idle = 0, timeout = 0, prior = 100, state = {1},squad = squad,

group = groups[1], position\_threshold = 100, online = true, in\_rest = "",

out\_rest = "", predicate = function(obj\_info) return obj\_info.rank >= 900 end}

table.insert(sj, t)

;-- bandit2 -> состояние 0 (день)

t = {section = "logic@esc\_bandits\_smart\_terrain\_bandit2\_walker",

idle = 0, prior = 5, state = {0}, squad = squad, group = groups[1],

in\_rest = "", out\_rest = ""}

table.insert(sj, t)

;-- bandit2 -> состояние 1 (ночь)

t = {section = "logic@esc\_bandits\_smart\_terrain\_bandit2\_sleeper",

idle = 0, prior = 5, state = {1}, squad = squad, group = groups[1],

in\_rest = "", out\_rest = ""}

table.insert(sj, t)

;-- bandit3 -> состояние 0 (день) и состояние 1 (ночь)

t = {section = "logic@esc\_bandits\_smart\_terrain\_bandit3\_walker",

idle = 0, prior = 5, state = {0, 1}, squad = squad, group = groups[1],

in\_rest = "", out\_rest = ""}

table.insert(sj, t)

end

Еще один момент о состояниях ST, всё зависит от того сколько у вас их в ST. Еще одна важной вещью является добавление логики для каждого состояния. Например в вашем ST такие состояния:

0 - нпс оффлайн

1 - нпс онлайн (день)

2 - нпс онлайн (ночь)

3 - нпс онлайн, они решили напасть на другой ST

4 - нпс онлайн, актор напал на них

К сведению, меня не прет заполнять такое большое количество таблиц, так что обычно я использую эту функцию:

function fill\_tbl(section, idle, prior, states, squad, group, in\_rest, out\_rest, online, gulag\_name)

local tbl = {}

tbl.section = "logic@" .. gulag\_name .. "\_" .. section

tbl.idle = idle

tbl.prior = prior

tbl.state = {}

for index = 1, #states do

table.insert(tbl.state, states[index])

end

tbl.squad = squad

tbl.group = group

tbl.in\_rest = in\_rest

tbl.out\_rest = out\_rest

tbl.online = online

return tbl

end

Используя функцию выше, мы можем загружать логику наподобие этой:

if type == "esc\_bandits\_smart\_terrain" then

local t = table.insert(sj, fill\_tbl("bandit1\_walker", 0, 100, {0}, squad, groups[1], "", "", true, type))

t.timeout = 0

t.position\_threshold = 100

t.predicate = function(obj\_info) return obj\_info.rank >= 900 end

table.insert(sj, t)

t = table.insert(sj, fill\_tbl("bandit1\_kamp", 0, 100, {1}, squad, groups[1], "", "", true, type))

t.timeout = 0

t.position\_threshold = 100

t.predicate = function(obj\_info) return obj\_info.rank >= 900 end

table.insert(sj, t)

table.insert(sj, fill\_tbl("bandit2\_walker", 0, 5, {0}, squad, groups[1], "", "", true, type))

table.insert(sj, fill\_tbl("bandit2\_sleeper", 0, 5, {1}, squad, groups[1], "", "", true, type))

table.insert(sj, fill\_tbl("bandit3\_walker", 0, 5, {0, 1}, squad, groups[1], "", "", true, type))

end

- автоматически изменяем работу для каждого нпс/мутанта -> function load\_states(...)

if type == "esc\_bandits\_smart\_terrain" then

return function(gulag)

if not db.actor then

return gulag.state

end

if level.get\_time\_hours() >= 5 and level.get\_time\_hours() <= 22 then

return 0 -- переключает всех мутантов/нпс на дневную работу

else

return 1 -- ереключает всех мутантов/нпс на ночную работу

end

end

end

- убедитесь что наш ST будет использоваться только для бандитов -> function checkStalker(...)

if gulag\_type == "esc\_bandits\_smart\_terrain" then

return npc\_community == "bandit"

end

Так же существуют универсальные гулаги General\_lager для сталкеров. Они считаются упрощенными гулагами.

Пример, создаем смарт:

[9999]

; cse\_abstract properties (основные параметры)

section\_name = smart\_terrain

name = esc\_gen\_lager

position = 131.02030944824,0.065616846084595,-248.9094543457

direction = 0,0,0

; cse\_alife\_object properties (параметры объекта)

game\_vertex\_id = 635

distance = 9.09999942779541

level\_vertex\_id = 363757

object\_flags = 0x==3e

custom\_data = <<END

[smart\_terrain]

type = general\_lager

capacity = 3 ;вместимость

communities = bandit ; комьюнити населения

END

; cse\_shape properties (параметры шейпа объекта)

shapes = shape0

shape0:type = sphere

shape0:offset = 0,0,0

shape0:radius = 20.55957102775574

; cse\_alife\_space\_restrictor properties (параметры рестриктора)

restrictor\_type = 3

; se\_smart\_terrain properties (параметры смарттеррейна)

General\_lager автоматически соберет все точки путей на уровне начинающиеся на имя смарта(в нашем случае esc\_gen\_lager) и разделит их названия таким образом: (имя\_смарта)\_(аи схема)\_(номер если требуется,напрмер если 2 walkerа то 1 и 2 соответственно)\_(поднастройка схемы)\_(состояние гулага 1 или 0,день или ночь) пример: esc\_gen\_lager \_walker \_1 \_walk \_1 (walk то же что и path\_walk,look это path\_look соответственно)

Пришедший на general\_lager сталкер рандомно примет любую свободную работу работой считается комбинация схем с одним номером. например esc\_gen\_lager\_walker\_1\_walk\_1 и esc\_gen\_lager\_walker\_1\_look\_1 это схема дневной работы одного персонажа из гулага.

Так что для каждого general\_lager нужны заранее расставленые точки путей для него.

"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

Создание Смарт террэйна на основе схемы General Lager

Материал из S.T.A.L.K.E.R. Inside Wiki.

Перейти к: навигация, поиск

Итак, вы хотите создать смарт, но не охота копошится в скриптах и конфигах и создавать свои скрипты для смарта? тогда этот способ именно для вас. Генерал лагер это универсальный смарт, с его помощью можно заселить локации или переобустроить ПЫСовские. Начнем с простенького.

[править]

Создадим смарт для мутантов.

Открываем сдк, жмем spawn element>ai>smart terrains, шлепаем сие на локацию, затем переходим в shape>sphere(можно и box по желанию) и снова вставляем сие на локацию, увеличиваем до нужных размеров (не сильно больших), переходим обратно в режим spawn element, жмем кнопку attach object и кликаем на созданной нами сфере. Мы создали смарт с зоной влияния. открываем properties правым щелчком мыши или eneter по умолчанию. открываем custom data и пишем в ней [smart\_terrain] type = general\_lair capacity = 4 где capacity = 4, это число мутантов, которые могут в ходить в смарт (тобешь есле capacity = 4, то в смарте будет 4 места для мутанта) переходим в режим way points, добавляем новый путь (он будет состоять из одной точки) и обзываем его smart\_name\_home\_1, где smart\_name это имя вашего смарта, все, смарт готов, осталось только создать самих мутантов, берем любого, нужного вам мутанта, ставим в зону влияния смарта и идем в custom data, в ней пишем [smart\_terrain] smart\_terrain\_name = true где smart\_terrain\_name - имя вашего смарта. Все, смарт террэйн для мутантов готов, собираем уровень и идем туда, куда поставили свой смарт и любуемся тем, что мутанты не разбредаются кто куда, а чем то занимаются рядом со смартом

\* Примечание.

\* Есле вы создали смарт на своем, новом, уровне то нужно добавить в config\misc\general\_lair

[Имя уровня] boar = weak, normal, strong bloodsucker = weak, normal, strong flesh = weak, normal, strong dog = weak, normal, strong pseudodog = weak, normal, strong giant = weak, normal, strong zombie = weak, normal, strong burer = weak, normal, strong controller = weak, normal, strong poltergeist = weak, normal, strong snork = weak, normal, strong tushkano = weak, normal, strong

и в тои же папке в smart\_terrain\_presets добавить

[Имя уровня] stalker = novice, experienced, veteran, master monolith = novice, experienced, veteran, master military = novice, experienced, veteran, master killer = novice, experienced, veteran, master ecolog = novice, experienced, veteran, master dolg = novice, experienced, veteran, master freedom = novice, experienced, veteran, master bandit = novice, experienced, veteran, master zombied = novice, experienced, veteran, master

boar = weak, normal, strong bloodsucker = weak, normal, strong flesh = weak, normal, strong dog = weak, normal, strong pseudodog = weak, normal, strong giant = weak, normal, strong zombie = weak, normal, strong burer = weak, normal, strong controller = weak, normal, strong poltergeist = weak, normal, strong snork = weak, normal, strong tushkano = weak, normal, strong

[править]

Создадим смарт для НПС

И так, повторяем действия сделанные выше Открываем сдк, жмем spawn element>ai>smart terrains, шлепаем сие на локацию, затем переходим в shape>sphere(можно и box по желанию) и снова вставляем сие на локацию, увеличиваем до нужных размеров (не сильно больших), переходим обратно в режим spawn element, жмем кнопку attach object и кликаем на созданной нами сфере. открываем properties правым щелчком мыши или eneter по умолчанию. открываем custom data и пишем в ней [smart\_terrain] type = general\_lager capacity = 5 что такое capacity вы уже знаете. Создаем пути для НПС (Работы для НПС) вот примеры путей

\* smart\_name\_kamp\_1

\* smart\_name\_sleep\_1

\* smart\_name\_walker\_1\_walk \ smart\_name\_walker\_1\_look

\* smart\_name\_walker2\_1\_walk \ smart\_name\_walker2\_1\_look

\* smart\_name\_patrol\_1\_walk

\* smart\_name\_guard\_1\_walk \ smart\_name\_guard\_1\_look

\* smart\_name\_sniper\_1\_walk \ smart\_name\_sniper\_1\_look

Соответсвенно слева путь НПС, а справа направление его взгляда. smart\_name - имя вашего смарта (его нужно вставлять перед любым названием пути, так же необходимо писать номер пути между видом пути и walk\look) теперь по подробнее о видах путей.

\* kamp - Это лагерь сталкеров у костра. Вокруг этой точки они будут рассказывать анекдоты, есть колбасу\хлеб, пить водку.

\* sleep - место для сна НПС состоит из двух точек, НПС ложится на нулевой точке и смотрит в первую.

\* walker - обычное бродяжничество по точкам НПС, при использовании точек для пути больше двух - smart\_name\_walker\_1\_look не применяется (путь у этого вида волкера должен быть не замкнут и состоять из точек пути больше одного)

\* walker2 - тот же волкер что и выше, только он ходит по замкнутым путям, лук пути почти не используются.

\* patrol - патруль лагеря, используется во множественном количестве (количество точек пути в одном пути больше единицы, путь должен быть замкнут),не применяются look пути

\* guard - часовой, используется в двойном количестве (в пути имеется две точки), очень часто используется look пути

\* sniper - работа для снайпера, задается двумя точками и look путями. Почему то НПС игнорируют данную работу.

Так, теперь, когда вы узнали виды работ - создайте по вкусу работы (их количество не должно превышать количество сталкеров в смарте) и не забудьте приписывать в начале имя смарта (например red\_bridge\_blokpoats\_kamp\_1) все, собираем уровень и смотрим на наш смарт, есле вы создали несколько лагерей на одном уровне с одинаковой кастом датой и ваши НПС начали разбредатся - сначала проследите, куда они пойдут, может их принял под себя другой смарт (лично у меня такое было)

smart\_name\_kamp\_1 имя\_гулага\_kamp\_коментарии типа [esc2\_tunnel3\_home\_1] причем вместо 1 может быть все что угодно

esc2\_tunnel3 название смарта

home используемая схема поведения

\_1 коментарии

да и незабутьте файл

gulag\_general.script именно в нем указанно все что касается гулагов этого типа

Получено с http://stalkerin.gameru.net/wiki/index.php/%D0%A1%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5\_%D0%A1%D0%BC%D0%B0%D1%80%D1%82\_%D1%82%D0%B5%D1%80%D1%80%D1%8D%D0%B9%D0%BD%D0%B0\_%D0%BD%D0%B0\_%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D0%B5\_%D1%81%D1%85%D0%B5%D0%BC%D1%8B\_General\_Lager

Категория: SDK