Точки пути (Way Points)

Материал из xrWiki

Содержание

Именование точек пути

Имя вейпоинта должно иметь следующий вид:
`имя|поле=значение|поле=значение|…`
Первое слово является именем и игнорируется парсером. Остальные фразы, разделенные символом `|` будут обработаны.

Примеры имён:
`wp00|a=hide`
`wp01`
`wp02|a=hide|s=weather`
и т.д.

Если задано имя поля, но не задано значение, то парсер автоматически подставит true.
Т.е. не надо писать `wp0|r=true|d=true`, достаточно просто написать `wp0|r|d`

Система расстановки путей

Пути можно делить на два типа: **path_walk** и **path_look**. Как можно понять из названия, по первым сталкеры ходят, а во вторые смотрят при ходьбе.

Как взаимодействуют между собой **path_walk** и **path_look**? Придя на точку **path_walk**, где установлена какая-то комбинация флажков, сталкер найдёт такую же комбинацию флажков в **path_look** и посмотрит в эту точку. Если же ни один флажок не установлен, сталкер пойдёт дальше не останавливаясь.

В точках путей можно задавать флаги, изменяющие поведение персонажа. Флаги задаются прямо в имени waypoint-а, например, для точки с именем **wp00**:
`wp00|flag1|flag2`

Флаги точек пути path_walk

- **n = 0 .. 9999**
  Номер точки синхронизации. Рекомендуется первой точке задавать значение **0**, остальным – числа по возрастанию с произвольным шагом. Прийдя в точку с большим **n**, сталкер будет ждать отстающих напарников. Примечание: сталкер дожидается опаздывающих напарников **только** в точках остановки (т.е. только в тех местах, где точка **path_walk** имеет общие флаги с одной из точек **path_look**).

Внимание – для поддержки зацикленных маршрутов, сталкеры на точке с **минимальным n** дожидаются сталкеров на точке с **максимальным n**. Поэтому минимальное количество точек синхронизации для корректной работы схемы должно составлять 3 точки или больше!

- **s = имя_звуковой_схемы**
  Пробегая через эту точку, сталкер включит указанную звуковую схему. Звук стартует ДО начала поворота и старта анимации. Для того, чтобы звук стартовал синхронно с анимацией – задавайте его в **path_look** соответствующей точки, а не в **path_walk**. Если нужно стартовать звук одновременно с ЛЮБОЙ из анимаций в этой точке, можно воспользоваться параметром **sa**.
- **sp =**
  С какой вероятностью будет проигран звук (по умолчанию 100)
- **sa = true**
  Ждать начала анимации в точке, прежде чем стартовать проигрывание звука (по умолчанию false).
- **sc = true**
  Разрешить проигрывать звуки схемы неоднократно (по умолчанию false).
- **sf, st**
  Временной интервал повторения фраз из выбранной звуковой схемы в секундах (по умолчанию от 5 до 10 сек).
- **p = 0...100**
  При наличии в точке флажка, общего с одной из точек **path_look**, задает вероятность того, что персонаж остановится в точке (по умолчанию 100).
- **c = true**
  Дальше перемещаться в присяде (по умолчанию false)
- **r = true**
  Дальше перемещаться бегом (по умолчанию false)
- **d = true**
  Перемещаться в состоянии danger (по умолчанию false)
- **ds = имена_диалогов**
  Имена диалогов, которые разрешено стартовать начиная с этой точки (разрешение действует до следующей точки). Имена задаются в виде текстовой строки, разделенной запятыми: `ds=bandits_talk,weather_talk` и т.д.
- **ret = число**
  Сразу же по прибытии в точку вызывает зарегистрированный при инициализации movement manager-a callback с этим числом в качестве второго аргумента.
- **rel = name,name,...**
  Меняет отношение других персонажей к себе. Используется вместе с параметром ret. Вызывается менеджером перемещения подобно пользовательской callback-функции, при этом в rel перечисляются через запятую персонажи, у которых нужно сменить отношение к себе, а в ret задается, какое именно отношение нужно установить: **1** – хорошее, **0** – плохое. Пример: `ret=0|rel=bandit1,bandit2` установит у бандитов плохое отношение к персонажу,

который пришел в данную точку пути.

- **w = имя_walk_пути**
  Переводит схему на новый **path_walk**. Рекомендуется также задать новый **path_look** с помощью параметра **l**, иначе текущий **path_look** будет сброшен. Персонаж идет на стартовую точку с режимом перемещения, заданным в точке с параметром **w**. Настройки функции коллбека при переключении пути сохраняются.
- **l = имя_look_пути**
  Сбросит схему на новый **path_look**. Задавать параметр **l** нужно вместе с параметром **w**, иначе **l** будет проигнорирован. По умолчанию **path_look** при смене **path_walk** будет сброшен.
- **a = state**
  Выбирает состояние тела при перемещении (только из раздела "Ходячие состояния"). Список состояний можно найти в `gamedata\scripts\state_lib.script`
- **sig = name**
  Установить сигнал с именем **name** сразу по прибытию в точку (до поворота) для последующей его проверки с помощью поля **on_signal** логической схемы. Если нужно установить сигнал после поворота – используйте соответствующий флажок пути **path_look**.

Флаги точек пути path_look

- **p = 100**
  Вероятность, с которой персонаж посмотрит именно в эту точку. Значения **p** всех возможных точек суммируются, т.е. если у одной точки **p = 100**, а у другой **300**, то персонаж посмотрит в первую с вероятностью 25%! (т.е. 100 из 400).

Рекомендуется задавать **p** так, чтобы их сумма составляла 100.
По умолчанию у всех точек **p = 100**

- **nowpn = true**
  Если персонаж должен спрятать оружие.
- **c = true**
  Смотреть в точку в присяде (по умолчанию используется значение одноименного поля из **path_walk**)
- **d = true**
  Смотреть в точку в состоянии danger (по умолчанию используется значение одноименного поля из **path_walk**)
- **att = 1 или 2**
  Номер атаки (основная, вспомогательная). Можно использовать вместо анимации, например: `a=nil|att=1`), а можно вместе с анимацией: `a=стреляем_в_потолок|att=1`
- **t = число**
  Время, которое персонаж будет ждать, играя анимацию или стреляя (по умолчанию 5000). Если требуется ждать бесконечно долго (например, это финальная точка пути), нужно задать **t** равным **-1**

```
Примечание: если персонаж ждет синхронизации в точке, то он будет играть анимацию столько времени,
сколько нужно для того, чтобы дождаться напарников, но только по прибытию всех напарников на точки
засечет заданное в t время. Исключение составляет стрельба – персонаж не станет стрелять сразу по
прибытию в точку, а сперва дождется напарников, а потом уже начнет стрелять в течение заданного времени.
```

- **s = имя**
  Звук, который персонаж разово проиграет, посмотрев в эту точку.
- **sp = число**
  С какой вероятностью будет проигран звук (по умолчанию 100)

- **sl = имя_прожектора**
  Если задано, то при повороте в указанную точку персонаж также повернёт и прожектор в неё.
- **ret = число**
  После поворота в целевую точку вызывает зарегистрированный при инициализации movement manager-а callback с числом **ret** в качестве второго аргумента. При этом время ожидания (поле **t**) игнорируется, т.е. после того как callback вызовет update_movement_state, персонаж сразу же пойдёт дальше.
- **a = state**
  Анимация, которую проиграет персонаж, стоя или сидя на месте (из разделов "Стоячие и сидячие состояния"), по умолчанию idle. Для того, чтобы персонаж стоял в точке без анимации, задайте значение nil: `a=nil` Список состояний можно взять в `gamedata\scripts\state_lib.script`
- **t = msec**
  Время в миллисекундах, которое персонаж должен смотреть в заданную точку. * — бесконечное время. Допустимы значения в диапазоне [1000, 30000], по умолчанию — 5000. Для конечных (терминальных) вершин пути **path_walk**, у которых не более 1-й соответствующей точки **path_look**, значение **t** всегда считается бесконечным и его явно задавать не нужно.
- **sig = name**
  После поворота в точку **path_look** установить сигнал с именем **name**.
- **syn**
  Наличие флажка задержит установку сигнала до тех пор, пока в точку с флажком **syn** не прибудут все персонажи данной **team** (**team** задается в виде текстовой строки в **customdata**). До тех пор, пока остальные персонажи не прибудут, ожидающей персонаж будет отыгрывать свою idle-анимацию.
- **sigtm = signal**
  Устанавливает сигнал при вызове **time_callback**-а **state manager**-ом. Соответственно, если **t=0**, то сигнал будет установлен после отыгрывания init-анимации. Это используется, например, с анимацией press, которая состоит из двух частей: 1 — нажимаем на кнопку, 2 — опускаем руку. В пути **path_look** можно сделать так: `wp00|a=press|t=0|sigtm=pressed`, а затем переключить схему: `on_signal = pressed | другая_схема`

Более подробное описание путей на примере логики walker

На карту для каждого **walker**-а нужно поставить:

1. Путь **path_walk**, по которому **walker** ходит.
2. Путь **path_look**, состоящий из точек, в которые **walker** смотрит.

Walker-ов может быть 1 или больше. Они могут действовать независимо, или взаимодействовать друг с другом. Если персонаж должен только **ходить** по маршруту, **path_look** можно не задавать.

[walker]

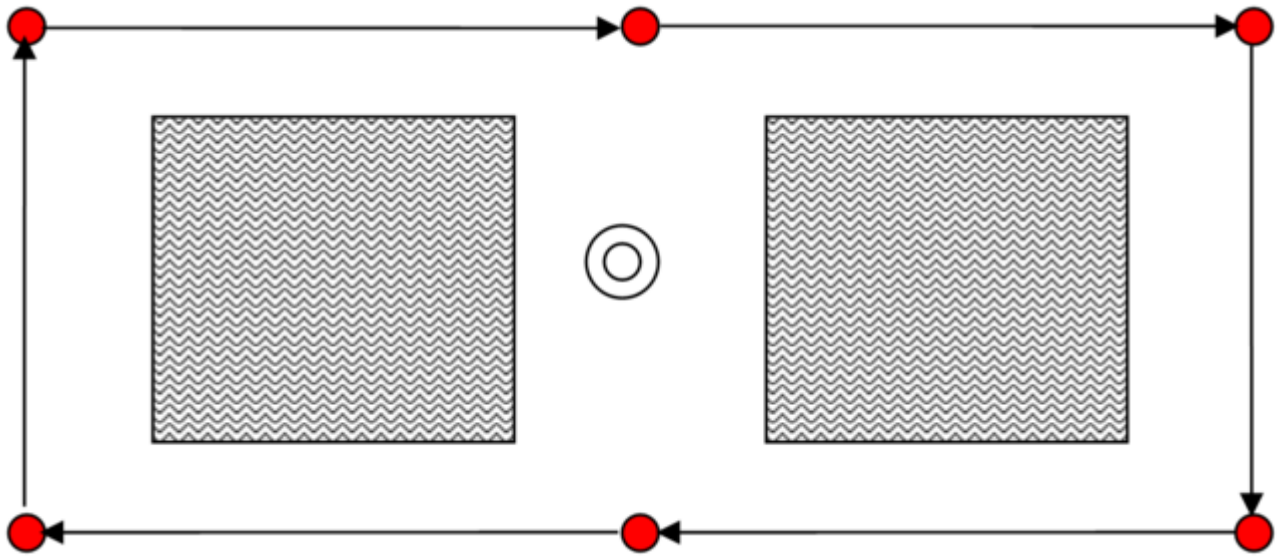
- **team = ...**
  Имя команды, произвольная текстовая строка. Все walker-ы в одной команде должны иметь один и тот же **team**. Желательно в **team** задавать имя уровня и имя места, где стоят walker-ы, например: escape_bridge, escape_factory. Это уменьшит шанс ошибиться и дать разным командам общее имя.

- **path_walk = ...**
  Имя пути, описанного в п. 1.
- **path_look = ...**
  Имя пути, описанного в п. 2 (необязательно). Если персонаж должен только ходить по маршруту, path_look можно не задавать. Если персонаж должен стоять на месте, то ему задается одна точка пути path_walk и как минимум одна точка пути **path_look**.

Правила расстановки флажков в путях рассмотрим на нескольких примерах.

Пример 1

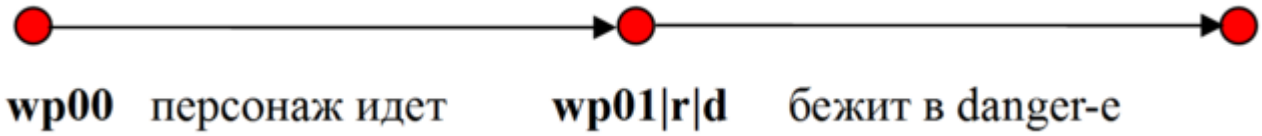
Персонаж патрулирует территорию вокруг двух домиков. Маршрут строится следующим образом:



Как сделать, чтобы персонаж между определенными точками бежал или крался? Для этого в пути **path_walk** существуют флажки. У каждого вейпоинта есть имя: **wp00**, **wp01** и т.д. Флажки задаются в имени. Их нужно отделять от самого имени с помощью символа **|**. Пишется **a=anim**, где **anim** – название анимации из *gamedata\scripts\state_lib.script*. Если мы напишем **a=threat**, то персонаж пойдет в состоянии **danger**, если **a=raid**, то побежит с оружием наизготовку и т.д.

```
Внимание: в точках пути path_walk используются анимации только из раздела «Ходячие состояния»!
```

Пример 2



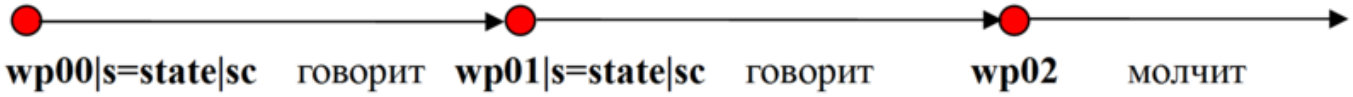
Чтобы персонаж говорил, перемещаясь по маршруту, нужно определить в каждой точке список тем, на которые он может говорить. Для этого существуют следующие поля:

- **s = имя_звуковой_схемы** (по умолчанию звук отключен). Несколько тем можно перечислять через запятую.
- **sp = вероятность произнесения фразы** (1..100)
- **sa** – флажок, который нужно поставить, если персонаж должен дождаться начала проигрывания анимации (об анимациях – ниже в этом документе), прежде чем включить звук, а не сделать это сразу по прибытию в точку. Если анимация в этой точке не играется, то звук

никогда не стартует.
- **sc** – флажок, который разрешает звуки указанных тем проигрывать неоднократно.
- **sf, st** – временной интервал повторения фраз из выбранной звуковой темы в секундах (по умолчанию от 5 до 10 сек).

Обратите внимание, что действие любых флажков распространяется только до следующей точки! Т.е. если нужно, чтобы персонаж говорил на заданную тему на протяжении какого-то отрезка маршрута, нужно в КАЖДОЙ точке маршрута перечислять эти темы.

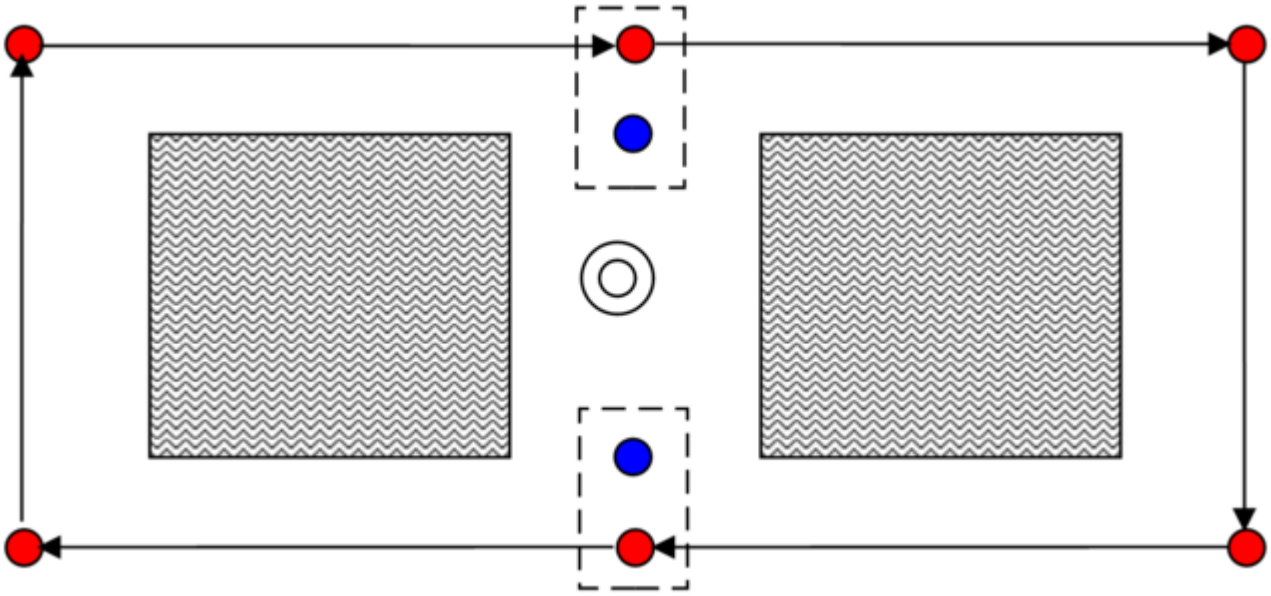
Пример 3



В примере 3 используется только поле **s**, чтобы задать тему разговора, и флажок **sc**, чтобы показать, что звук проигрывается не разово, а периодически. Остальные параметры (**sa**, **sf**, **sp**, **st**) задавать не рекомендуется, значения по умолчанию приемлемы для большинства скриптов. Если нужно стартовать звук одновременно с анимацией, лучше воспользоваться полями пути **path_look**, о котором будет написано ниже. Если персонаж не только ходит по маршруту, но должен также останавливаться и играть анимации, нужно задать ему путь **path_look**.

Пример 4

Усовершенствуем пример 1, чтобы персонаж, проходя мимо проема между домами, останавливался и заглядывал в него:



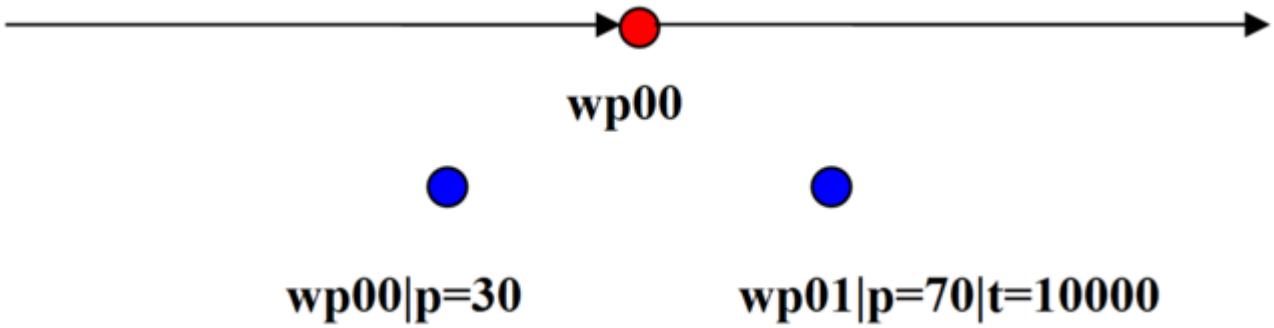
Что добавилось в этом примере? Путь **path_look** с двумя точками. Связь между точками этого пути рекомендуется сразу же удалить в редакторе, поскольку она все равно не используется. Далее, в точках путей **path_walk** и **path_look**, которые обведены на рисунке пунктирной линией, в редакторе ставим общие флажки. Например, в верхней паре точек ставим флажок **0**, а в нижней паре точек — флажок **1**.

Теперь персонаж будет останавливаться в точках **path_walk**, помеченных флажком, и смотреть в точку **path_look**, помеченную тем же самым флажком. Если точка **path_walk** не помечена флажком, персонаж проходит её не останавливаясь. Одной точке **path_walk** может соответствовать несколько точек **path_look**. Тогда персонаж случайным образом выберет одну из подходящих точек.

По аналогии с **path_walk**, в точках пути **path_look** можно использовать различные флажки, меняющие поведение:

- **p = ...** — вероятность, с которой персонаж посмотрит именно в эту точку. Значения **p** всех подходящих точек суммируются, т.е. если у одной точки **p = 100**, а у другой **300**, то персонаж посмотрит в первую с вероятностью **25%** (т.е. 100 из 400).
  Во избежание путаницы, рекомендуется задавать **p** так, чтобы их сумма составляла **100**. По умолчанию у всех точек **p = 100**.
- **t = ...** — время, на которое персонаж задержится в этой точке (по умолчанию 5000 мсек).

Пример 5



В этом примере, проходя через точку **wp00**, персонаж с вероятностью **30%** посмотрит в точку **wp00** в течение 5 секунд, и с вероятностью **70%** посмотрит в точку **wp01** в течении 10 секунд.

По умолчанию при остановках персонаж играет анимацию **idle**, если он не в состоянии **crouch**, либо анимацию **hide**, если он в состоянии **crouch**.

Если требуется другая анимация, можно ее указать с помощью флажка:

- **a = имя_анимации** (по умолчанию idle).
  Если мы напишем **a=hide**, то персонаж сядет в состоянии **danger**, если **a=guard**, то встанет с оружием наизготовку и т.д.

```
Внимание: в точках пути path_look используются анимации только из раздела «Стоячие и сидячие состояния»!
```

Источник — «https://xray-engine.org/index.php?title=Точки_пути_(Way_Points)&oldid=1177»

Категории:

Level Editor
A-Life
