show

Показывает спрайт персонажа.

Синтаксис

Описание синтаксиса данной команды будет разделёно на две части.

В первой будет описан синтаксис основных параметров, во второй — синтаксис дополнительных параметров.
Оба синтаксиса одинаково применимы к обоим видов спрайтов (статичным и Live2D) с небольшими отличиями, о которых будет написано далее.

Основные параметры

```
show [charID или alias] ([poseInfo]) (animations)
```

Параметры

- `charID/alias` — ID или псевдоним персонажа. При первичном спавне модели всегда указывается ID персонажа. Если во время спавна вы присвоили персонажу псевдоним при помощи параметра `as`, то дальнейшие команды для данного персонажа должны использовать этот самый псевдоним (`alias`).
  Обязательный параметр.
- `poseInfo` — поза и скин персонажа, разделённый пробелом. Данный параметр является обязательным, если у спрайта не прописаны поза и скин по умолчанию.

  Также в качестве значения этого параметра может быть

использовано краткое название сочетания позы и скина (`shortname`), если оно определено.
- `animations` — список анимаций.
  Подробнее о том, как указываются анимации, будет написано в отдельной главе ниже.

Дополнительные параметры

```
{at} {size} ({layer} или {behind} или {ahead}) {spritecolor} {with} {as} (|skip| или |await|)
```

Параметры

- `at` — положение спрайта на экране. Значением может быть название положения или координаты, разделённые пробелом.
  При первичном спавне, если значение не указано, спрайт будет выведен в положении `center`.
- `size` — размер спрайта на экране. Значением может быть название размера или числовые значения, разделённые пробелом.
  При первичном спавне, если значение не указано, спрайту будет присвоен размер `normal`.
- `layer/behind/ahead` — определяет приоритетность отображения спрайта относительно других спрайтов на экране.

  `layer` может принимать до двух числовых значений, разделённых пробелом — номер слоя и позиция в слое. Чем выше значение — тем приоритетнее спрайт, т.е. он будет отображаться поверх других спрайтов, у которых номер слоя или позиция в нём меньше.

  Оба значения должны быть больше или равны нулю.
  Значения по умолчанию для обоих параметров при первичном спавне — `0`.

  `behind` и `ahead` спавнят спрайт за или перед каким-то

определённым спрайтом соответственно. Данные параметры принимают только имя персонажа, относительно которого нужно расположить спрайт на слое (они не меняют положение спрайта, только приоритетность отображения на экране).

- `spritecolor` — оверлей-цвет для спрайта. Если указан, то будет переопределять глобальные настройки оверлей-цветов, заданные командой `scene`. Аналогично параметру команды `scene`, значение может быть как названием цвета, так и конкретным цветом;
  `none` — отключает цвет у спрайта и игнорирует любые глобальные изменения цветов спрайтов;
  `default` — выставляет спрайту текущий активный глобальный цвет;
- `with` — название (и опционально параметры) перехода появления спрайта.
- `as` — выводит спрайт под псевдонимом, что позволяет вывести на экран несколько спрайтов одного и того же персонажа.

  Также копирует параметры отображения спрайта на экране, если в `charID`
  используется выведенный на экран персонаж (работает только для Live2D спрайтов).

- `skip/await` — по умолчанию, при первичном спавне спрайта на экране, игра будет ожидать завершения команды (т.е. пока спрайт полностью появится), все дальнейшие команды для спрайта (пока он на экране) игра ожидать не будет.

  При необходимости подобное поведение можно изменить при помощи этих двух флагов.

  `skip` пропускает ожидание выполнения, а `await` — наоборот — ждёт, пока команда будет выполнена.

Примеры использования

```
# выводит фон akihibara_night
scene bg akihibara_night

# спавн спрайта Химицу под псевдонимом test_sprite с переходом dissolve
show hi with dissolve as test_sprite

"..."

# перемещает спрайт в левую часть экрана и меняет ему размер
show test_sprite at left size close with dissolve

"..."
```

```
# выводит фон akihibara_night
scene bg akihibara_night

# спавн спрайта Химицу
show hi at left size close with dissolve

"..."

# выводит спрайт Кетрин поверх спрайта Химицу
show ca at left size close ahead hi with dissolve

"..."

# выводит копию спрайта Кетрин под псевдонимом ca2 справа
show ca at right as ca2 with dissolve

"..."

# меняет спрайту ca2 скин
show ca2 ca_front dom with dissolve

# перемещает Химицу в центр экрана, а также меняет ей позу и скин
```

```
show hi himitsu_l plate at center with dissolve

"..."
```

Анимации

У всех Live2D спрайтов, встроенных в игру, существуют следующие слои анимации:

- emotions (эмоции)
- head (голова)
- hands (руки)
- legs (ноги)
- additive (вспомогательные анимации)

> Указанный выше порядок действителен только для встроенных в игру спрайтов. Для Live2D спрайтов, определённых модом, порядок и названия слоёв определяются в `resources.yaml`

Все слои, кроме `additive`, принимают название одной конкретной анимации. В `additive` могут прописываться несколько анимаций одновременно, т.к. этот параметр объединяет все вспомогательные слои.

> Данное правило распространяется только на встроенные в игру спрайты.

Синтаксис

Анимации для Live2D-спрайтов могут прописываться двумя способами — кратким и полным.

Краткий

Краткий синтаксис выглядит так:

```
(emotions) (head) (hands) (legs) (additive)
```

Названия анимаций для конкретных слоёв прописываются друг за другом, при этом необязательно указывать анимации для всех слоёв — допустим, можно указать только название эмоции (emotions) и положение рук (hands), пропустив положение головы (head). При использовании данного синтаксиса необходимо строго соблюдать порядок слоёв.

Пример:

```
# выводит спрайт Химицу с эмоцией angry, положением головы head2left и положеним рук hands3
show hi angry head2left hands3
```

Полный

В полном же синтаксисе указываются названия слоя и анимации, разделённые двоеточием:

```
layer:animation layer:animation layer:animation
```

При этом порядок, в котором они указываются, неважен.

Пример:

```
# выводит спрайт Химицу с эмоцией angry, положением головы head2left и положеним рук hands3
# порядок, в котором прописываются анимации, неважен
show hi emotions:angry head:head2left hands:hands3
```

> При использовании полного синтаксиса, для вспомогательных анимаций также необходимо указывать названия слоёв.

Статичные спрайты

Статичные спрайты — это спрайты, которые собираются из изображений. У них нет анимации. Такие спрайты можно определить внутри файла `resources.yaml`, и они могут содержать различные части, которые могут меняться.

Для отображения или скрытия тех или иных частей спрайта используется синтаксис, аналогичный синтаксису анимаций, только вместо названия анимаций (и слоев) используются названия частей и их групп.

Например, мы определили в моде спрайт `mt`, в котором определена группа `uniform` — различные вариации одежды, и группа `face` — различные вариации эмоций:

```
# resources.yaml
---
  characters:
    mt:
      poses:
        front_1:
          parts:
            body: sprites/mt/mt_1_body.png # это постоянная часть, которую мы определили ранее
            ~face:
              default: normal # по умолчанию будет отображаться эмоция normal
              parts:
                normal: sprites/mt/mt_1_normal.png # нейтральное выражение лица
                smile: sprites/mt/mt_1_smile.png # улыбка
                sad: sprites/mt/mt_1_sad.png # грусть
            ~uniform: # группа uniform
              default: pioneer # по умолчанию при вызове спрайта будет выводиться пионерская форма
              parts:
                pioneer: sprites/mt/mt_1_pioneer.png # пионерская форма
                dress: sprites/mt/mt_1_dress.png # платье
                swimsuit: sprites/mt/mt_1_swim.png # купальник

      # остальные параметры
```

Чтобы поменять эмоцию и одежду, мы можем воспользоваться кратким синтаксисом:

```
# выводит спрайт mt с улыбкой и в платье
show mt smile dress
```

Как и в случае с анимациями, при использовании краткого синтаксиса важен порядок групп. В данном случае порядок определяется внутри файла ресурсов.

Также мы можем воспользоваться полным синтаксисом:

```
# выводит спрайт mt с улыбкой и в платье
# порядок групп неважен
show mt face:smile uniform:dress
```

Кроме того, можно убрать какую-то часть при необходимости. Для этого достаточно прописать точку вместо названия части спрайта:

```
# выводит спрайт mt с улыбкой и без одежды, краткий синтаксис
show mt smile .

# делает то же самое, полный синтаксис
show mt face:smile uniform:.
```

Revision #35
Created 21 July 2023 21:12:49 by Admin
Updated 21 January 2025 16:21:28 by Admin