characters

Тип: список

Определяет статичные спрайты персонажей для команды show.

Внутри нода прописываются ноды с параметрами статичных (неанимированных) спрайтов персонажей:

```
# resources.yaml
---
  characters:
    dv:
      # ... параметры
    un:
      # ... параметры
    sl:
      # ... параметры
    us:
      # ... параметры
```

Каждый спрайтовый нод может содержать следующие параметры:

- `poses`
- `positions`
- `sizes`
- `shortnames`
- `default`

poses

Тип: список

Определяет позы спрайта.
Под позами понимаются различные вариации спрайта:



3 различных позы на примере спрайта из Бесконечного Лета

Каждая поза может содержать несколько параметров:

- `parts` — обязательный параметр
- `canvasSize`
- `positions`
- `sizes`
- `spritecolors`

```
# resources.yaml
---
  characters:
    mt:
      poses:
        front_1:
          parts:
           # ... параметры
          canvasSize: 900 1080
          positions:
           # ... параметры
          sizes:
           # ... параметры
          spritecolors:
           # ... параметры
```

parts

Тип: список

Определяет части, из которых собирается спрайт.

Части могут быть двух типов:

- постоянные
- непостоянные

Тип ассетов у частей — `texture`.

Постоянные части

Постоянные части видны всегда и никогда не меняются (например, тело персонажа).

Определяются следующим образом:

```
# resources.yaml
---
  characters:
    mt:
      poses:
        front_1:
          parts:
            body: sprites/mt/mt_1_body.png # тело персонажа, которое должно отображаться всегда


          # ... остальные параметры
```

На примере выше мы определили часть `body` — тело спрайта, на которое будут накладываться остальные элементы.

Непостоянные части

Непостоянные — соответственно, те части, которые могут меняться. Это эмоции, одежда, различные аксессуары и прочее. Такие части объединяются в группы, которые позволяют менять активную (отображаемую) часть из сценария.

Непостоянные части определяются так же, как и постоянные, только внутри отдельного нода, начинающегося со знака `~` — такой нод называется группой:

```
# resources.yaml
---
  characters:
    mt:
      poses:
        front_1:
          parts:
            body: sprites/mt/mt_1_body.png # это постоянная часть, которую мы определили ранее
            ~uniform: # группа uniform
              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
```

У частей спрайта в группе также может указываться параметр `position`, синтаксис аналогичен.

На примере выше мы создали группу `uniform` с различными вариациями одежды, которые можно использовать в игре. Однако мы не указали, какая часть должна отображаться по умолчанию — в случае, если мы явно не обозначили это в сценарии.

Чтобы сделать какую-либо часть в группе частью, отображаемой по умолчанию, можно воспользоваться параметром `default`:

```
# resources.yaml
---
  characters:
    mt:
      poses:
        front_1:
          parts:
            body: sprites/mt/mt_1_body.png # это постоянная часть, которую мы определили ранее
            ~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
```

Как видно из примера, ID части, выводимой по умолчанию, прописывается в ноде `default`, а сами части — в `parts`.

Если нод `default` не указан, то по умолчанию ни одна часть не будет выводиться.

Позиционирование частей

Каждой части спрайта можно задать положение относительно канваса:

```
# resources.yaml
---
  characters:
    mt:
      poses:
        front_1:
          parts:
            body:
              asset: sprites/mt/mt_1_body.png
              position: 0 100

          # ... остальные параметры
```

На примере выше изображение будет смещено на 100 пикселей по вертикали **относительно левого верхнего угла** канваса спрайта. Ассет же прописывается в ноде `asset`.

> Использование параметра `position` у части спрайта возможно только если указан параметр `canvasSize` у позы.

Порядок наложения частей

Части спрайта (будь то постоянная часть или группа) накладываются друг на друга согласно порядку, в котором они определены:

```
# resources.yaml
---
  characters:
    mt:
      poses:
        front_1:
          parts:
            body: sprites/mt/mt_1_body.png # сначала будет выведено тело
```

```
      ~uniform: # затем форма
        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
      ~face: # затем лицо
        normal: sprites/mt/mt_1_normal.png # нейтральное лицо
        smile: sprites/mt/mt_1_smile.png # улыбка
        sad: sprites/mt/mt_1_sad.png # грустное лицо

    # ... остальные параметры
```

canvasSize

Тип: текстовый нод

Определяет размер канваса спрайта. Этот параметр нужен для использования параметра `position` у частей спрайта, т.к. относительно значения этого параметра производится позиционирование изображений.

Значение должно состоять из двух целых чисел, разделённых пробелом, и представлять собой корректное разрешение:

```
# resources.yaml
---
  characters:
    mt:
      poses:
        front_1:
          canvasSize: 900 1080 # 900x1080

          # ... остальные параметры
```

positions

Тип: список

Определяет именованные положения спрайта относительно игровой сцены. Не стоит путать с параметром `positions` у частей спрайта.

```
# resources.yaml
---
  characters:
    mt:
      poses:
        front_1:
          positions:
            center: 0 0 # центр
            left: -0.5 0 # лево
            right: 0.5 0 # право

          # ... остальные параметры
```

Данный параметр, указанный внутри позы, позволяет задавать положения, действительные только для этой позы. Это может быть полезно, если у разных поз спрайта разные размеры изображений, и они должны использовать разные координаты.

> Указанные внутри нода персонажа позиции будут переопределять одноимённые позиции выше по иерархии.

Также можно указать значение только для одной оси, прописав `.` в качестве значения для второй:

```
# resources.yaml
---
  characters:
    mt:
      poses:
        front_1:
          positions:
            center: -0.5 . # изменена только X-координата
            left: . -0.5 # изменена только Y-координата
```

```
        # ... остальные параметры
```

Значение для недостающей координаты будет браться из позиций с таким же ID выше по иерархии, если же таких нет — будет выставлен 0.

Подробнее о положениях можно почитать здесь.

sizes

Тип: список

Определяет именованные размеры спрайта относительно игровой сцены. Синтаксис схож с `positions`:

```
# resources.yaml
---
  characters:
    mt:
      poses:
        front_1:
          sizes:
            normal: 1 1
            close: 2 .

          # ... остальные параметры
```

Внутри каждого нода можно также прописать позиции — они будут использоваться, когда у спрайта будет указанный размер:

```
# resources.yaml
---
  characters:
    mt:
      poses:
```

```
front_1:
  positions:
    center: 0 0 # это значение будет использоваться при любых других размерах
  sizes:
    normal: 1 1
    close:
      value: 2 .
      positions: # позиции при размере close
        center: 0 0.25 # это значение будет использоваться при размере close

  # ... остальные параметры
```

Размер в таком случае указывается внутри нода `value`*.*

На примере выше, когда у спрайта в сценарии будут прописаны `size close` и `at center`, в качестве координат для `center` будет использоваться значение `0 0.25`, а не `0 0`.

Аналогично `positions`, размеры, указанные внутри нода персонажа, будут переопределять одноимённые размеры выше по иерархии.

Подробнее о размерах можно почитать здесь.

spritecolor

Тип: список

Цвета для наложения на спрайт. Используются для более естественного отображения спрайтов на различных фонах (день, ночь и т.п.).

Значением может быть как цвет, заданный в `colors`, так и HEX-код или RGB(A)-код.

```
# resources.yaml
---
  characters:
    mt:
      poses:
        front_1:
          spritecolor:
            tokyo_street_sunset: #FFF1ED

          # ... остальные параметры
```

Подробнее о цветах спрайтов можно почитать здесь.

shortnames

Тип: список

Определяет короткие имена для поз и различных вариаций частей групп, которые можно использовать в сценарии.

Короткие имена прописываются так же, как прописываются позы и группы внутри сценария:

```
# resources.yaml
---
  characters:
    mt:
      shortnames:
        front_sd: front smile dress # или front face:smile uniform:dress
```

После определения коротого имени, его можно использовать внутри сценария:

```
show mt front_sd with dissolve
```

default

Тип: текстовый нод

Определяет позу и части групп, которые будут выводиться по умолчанию (если конкретные значения не указаны в команде сценария).

В качестве значения можно прописать короткое имя или позу и параметры групп:

```
# resources.yaml
---
  characters:
    mt:
      default: front smile dress
      # ... остальные параметры
```

Теперь, когда в сценарии будет прописано `show mt`, по умолчанию будет спавнится вариация `front smile dress`.

positions

Тип: список

Определяет позиции, которые могут использовать все позы спрайта. Любые позиции внутри конкретной позы будут переопределять данные позиции, если у них совпадают ID.

Синтаксис идентичен синтаксису positions внутри позы:

```
# resources.yaml
---
```

```
  characters:
    mt:
      positions:
        center: 0 0
        left: -0.5 0
      # ... остальные параметры
```

sizes

Тип: список

Определяет размеры, которые могут использовать все позы спрайта. Любые размеры внутри конкретной позы будут переопределять данные размеры, если у них совпадают ID.

```
# resources.yaml
---
  characters:
    mt:
      sizes:
        normal: 1 1
        close: 2 .
        far:
          value: 0.7 0.7
          positions:
            center: 0 -0.25
      # ... остальные параметры
```

spritecolor

Тип: список

Определяет оверлей-цвета спрайтов, которые могут использовать все позы спрайта. Любые цвета внутри конкретной позы будут

переопределять данные цвета, если у них совпадают ID.

```
# resources.yaml
---
  characters:
    mt:
      spritecolor:
        tokyo_street_sunset: #FFF1ED
      # ... остальные параметры
```

Revision #100
Created 4 July 2023 20:43:11 by Admin
Updated 19 August 2023 21:08:43 by Admin