Camera API

Edit New Page

Jump to bottom

Hans Jorgensen edited this page on Dec 22, 2015 · 2 revisions

In Sigma, multiple views may be made of the same graph, and multiple renderers may subscribe to the exact same view. Cameras in Sigma encapsulate the information needed for such a view, separate from the renderer.

Basic use

To add a camera to the graph, simply call Sigma's `addCamera` method, providing a unique ID for the camera (or omitting it and allowing Sigma to generate one for you):

```
sigInst.addCamera('cam1');
```

Then, to assign a specific renderer to it, add the ID to the configuration object for the renderer:

```
s.addRenderer({
  container: document.getElementById("graphContainer"),
  camera: 'cam1',
  /* rest of settings */
});
```

Camera methods

Camera objects (returned from `sigInst.cameras`) expose the following methods:

- **sigma.classes.camera(id, graph, settings)**
  - Constructor for creating a camera manually, using the specified string ID, Sigma graph, and settings object. This is rarely called from outside Sigma - calling `sigInst.addCamera` is much preferred.
- **goTo(coordinates)**
  - Update the camera's position immediately. The coordinates argument expects an object with the following properties:
    - `x` : The center x position of the camera in graph space
    - `y` : The center y position of the camera in graph space
    - `ratio` : The zoom ratio of the graph and its items

        - `angle`: The rotation angle of the camera
    - Returns the camera instance
- **applyView(read, write)**
    - Applies a view to the attached graph by reading all of the coordinates prefixed with the string given by `read` and transforming them to the coordinates given by `write`.
    - This is called whenever the renderer detects a camera move and needs to re-render the nodes. It is rarely called from outside Sigma.
    - It returns the camera instance
- **graphPosition(x, y)**
    - Takes coordinates in camera space and returns them in graph space as an object `{x, y}`
- **cameraPosition(x, y)**
    - Takes coordinates in graph space and returns them in camera space as an object `{x, y}`
- **getMatrix()**
    - Gets the WebGL-compatible matrix for the graph, useful for applying rendering transformations
- **getRectangle(width, height)**
    - Takes the screen width and height and returns a rectangle (in the form of an object `{x1, x2, y1, y2, height}`) that represents the camera on screen.
    - To help prevent node labels from going out of the screen, the method keeps a small margin around the screen in the rectangle automatically.

Camera properties

Cameras expose the following properties:

Property	Description
graph	**(Read only)** Returns the graph this camera is bound to
id	**(Read only)** Returns the unique string ID of this camera
readPrefix	**(Read only)** Returns the preferred input prefix for node coordinate transformations
prefix	**(Read only)** Returns the preferred output prefix for node coordinate transformations
x	The center x position of the camera on the graph
y	The center y position of the camera on the graph
ratio	The scaling ratio of the camera
angle	The rotation angle of the camera
isAnimated	Indicates whether or not a camera's position is being animated
settings	The settings object for this camera

The properties `x`, `y`, `ratio`, and `angle` should be set via the `goTo()` method so that other objects may be informed that the camera has moved.

Camera events

Cameras in Sigma also implement `sigma.classes.dispatcher`, which means that they fire events. Currently, cameras only fire one event with no arguments:

Event	Description
coordinatesUpdated	Fires whenever the camera's coordinates have been updated via `goTo()`. Use this event to watch for camera changes and update things such as scrolling backgrounds.