Linaïa
Innover Tester Développer

Technical Documentation

CNRS

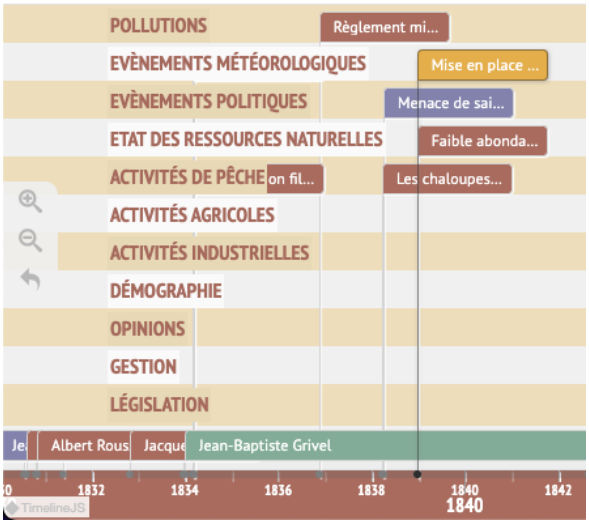
TimeLineJs improvement

Contents

Highlighting of categories

Initialization of the style

Category names are displayed in the foreground.



Add a padding on each category.

Assign the color of the category line to the text background color.

Darkening of the text background color to match the category line color.

Increase of the z index TL-TimeGroup.less line 27

Modified code:

File TimeGroup.js

```
87    _initLayout () {
88
89        // Create Layout
90        this._el.message = DOM.create("div", "tl-timegroup-message ", this._el.container);
91
92        //Handle Color categories
93        this._el.message.style.backgroundColor = this.data.color;
94        this._el.message.style.filter = "brightness(98%)";
95        this._el.message.style.padding = "2px 2px 3px 2px";
96
97        this._el.message.innerHTML = this.data.label;
98    }
```

Manage categories and their colors

Assigning the order and color of the categories

The order and color of the categories are to be defined in the Google Sheets file, at the very beginning of the table, starting from line n°1.

Group	GroupColor	GroupOrder
Législation	#EFE1C2	11
Gestion		10
Opinions	#EFE1C2	9
Etat des ressources naturelles		4
Activités de pêche	#EFE1C2	5
Activités agricoles		6
Activités industrielles	#EFE1C2	7
Démographie		8
Pollutions	#EFE1C2	1
Evènements météorologiques		2
Evènements politiques	#EFE1C2	3

Recovery of values

File TimeScale.js

```
if (slides[i].group) {
    if (groups.indexOf(slides[i].group) < 0) {
        var colorLigne = slides[i].GroupColor;
        var group_order = slides[i].GroupOrder;
        colors.push(colorLigne);
        group_orders.push(group_order);
        groups.push(slides[i].group);
    }
```

So for each existing group, the color and the order are added.

group_info is accessible in the TimeNav.js file which allows to modify the Timeline.

Once the values are defined in the Google Sheets table, they are retrieved and assigned in two tables colors and group_orders.

```
for (var i = 0; i < groups.length; i++) {
    group_info[i] = {
        label: groups[i],
        color: colors[i],
        order: group_orders[i],
        idx: i,
        positions: [],
        n_rows: 1, // default
        n_overlaps: 0
    };
}
```

Creation of groups

File TimeNav.js

```
_createGroups() {
    this._groups = [];
    var group_labels = this.timescale.getGroupLabels();

    var group_ordered = this._reorderGroup(group_labels);
    if (group_labels) {
        this.options.has_groups = true;
        for (var i = 0; i < group_ordered.length; i++) {
            this._createGroup(group_ordered[i]);
        }
    }

}
```

The values of the groups are retrieved in the TimeScale.js file.

group_labels now holds the groups with their order and color.

The groups are not yet ordered.
Creation of group_ordered by calling _redorderGroup.

The _reorderGroup function takes groups as parameters and reorders them.

Once reorganized, the groups are created in the TimeGroup.js file.

```
_reorderGroup(groups) {
    var group_ordered = [];
    //sort groups by group_order
    for (var i = 0; i < groups.length; i++) {
        var group = groups[i];
        if (group.group_order) {
            group_ordered[group.group_order - 1] = group;
        }
    }
    return group_ordered;
}
```

File TimeMarker.js

```
_initLayout() {

    // Create Layout
    let isHidden = this.data.GroupOrder ? 'hidden' : '';
    this._el.container = DOM.create("div", `tl-timemarker ${isHidden}`);
```

The groups created at the beginning of the table are considered as events by default and are therefore displayed with a "marker" in the form of a box. These boxes must therefore be hidden for better readability of the Timeline. So, in the Google Sheets table, if an event contains a value in the GroupOrder column, it automatically hides the marker in the Timeline.

Color assignment

File TimeGroup.js

```
// Data
this.data = {
    label: "",
    rows: 1,
    color: ""
};
```

We add the color attribute to the TimeGroup class.

Then, we access it via this.data.color.

```
// Change the group line color
this._el.container.style.backgroundColor = this.data.color;
```

Manage marker colors

Initialization

The marker color is defined in the hexadecimal format #RRVVBB in the Google Sheets table, in the column `MarkerColor`.

MarkerColor
#8db5a1
#b18869
#b16972
#b16972
#b18869
#b18869
#8db5a1
#8db5a1
#8db5a1
#8db5a1

File ConfigFactory.js

```
function extractEventFromCSVObject(orig_row) {

    let row = {}
    Object.keys(orig_row).forEach(k => {
        row[k] = trim(orig_row[k]) // get rid
    })
    var d = {
        id: row['n°ID'] || '',
        type: row['Type'] || '',
        categories: row['Categorie'] || '',
        markerColor: row['MarkerColor'] || '',
        GroupOrder: row['GroupOrder'] || '',
        GroupColor: row['GroupColor'] || '',
        parentOf: row['parentOf'] || '',
        childOf: row['childOf'] || '',
        typeOfLink: row['TypeOfLink'] || '',
        ZoomOnClick: row['ZoomOnClick'] || '',
```

Once the color is defined, we add a property to the marker object.

This property takes the value of row['MarkerColor'] which corresponds to the value written in the `MarkerColor` column in the Google Sheets table.

If there is no color value specified, the box will default to gray.

File TimeMarker.js

When initializing the layout and markers, their background color is changed by the value in the `MarkerColor` column.

```
// Handle color
this.data.markerColor != "" ? this._el.content_container.style.backgroundColor = this.data.markerColor : "";
```

Highlighting of events and their sequences

Relationship between Parent and Child

In order to highlight the events and their chains, the relationship between the elements is activated by listing the identifiers (id) in the `parentOf` and `childOf` columns.

parentOf	childOf
	E0035
E0034,E0036	
	E0035

An event can have several parent or child relationships. Thus all relationships can be mentioned, just separate the event identifiers by a comma or a semi-colon in the dedicated columns.

Thus, when an event is clicked on in the Timeline, all other events will be faded out to show only the events related to it and indicated in the `parentOf` and `childOf` columns.

Information retrieval

File ConfigFactory.js

```
        parentOf: row['parentOf'] || '',
        childOf: row['childOf'] || '',
    }
    if(d.parentOf != ''){
        var parents = d.parentOf.split(/[,;]/);
        d.parentOf = parents;
    }
    if(d.childOf != ''){
        var children = d.childOf.split(/[,;]/);
        d.childOf = children;
    }
```

The fields are retrieved from the Google Sheets table and assigned to parentOf and childOf.
Once retrieved, they are processed to separate the values, via a comma or semicolon.

Click interaction

File TimeMarker.js

```
_onMarkerClick(e) {
    //Remove highligh from other markers
    $(".tl-timemarker-content-container").removeClass('highlighted');
    $(this._el.content_container).addClass('highlighted');
```

A highlighted class is added to the clicked marker to make it stand out.

Then, highlighting (_setHighlight) of all linked markers thanks to the identifiers contained in the parentOf and childOf.

```
//highlight the parent and its children
this._setHighlight(this.data.parentOf);
this._setHighlight(this.data.childOf);
```

The function takes as parameter data, and loops on data.

For each parent and/or child, the spaces are removed and the highlighted class is added.

```
_setHighlight(data) {
    for (var i = 0; i < data.length; i++) {
        //remove space
        var str = data[i].replace(/\s/g, '');
        $("." + str).addClass("highlighted");
    }
}
```

''. "+str Refers to the marker, in fact when the marker is created, the id value contained in the Google Sheets table is assigned to a Marker id

```
let idMarker = this.data.id;
idMarker = idMarker.replace(/\s/g, '');
this._el.content_container = DOM.create("div", `tl-timemarker-content-container ${idMarker} ${hasParent} ${isHidden} `, this._el.container);
```

Marker opacity management

On click, the opacity changes only if the marker contains children or parents.

```
//Change opacity if marker has childs or parents
if (!this.data.parentOf.includes('') || !this.data.childOf.includes('')) {
    $(".tl-timemarker-content-container").not('.highlighted').not('.tl-timemarker-active').css('opacity', 0.1);
}
```

Remote relationship

When a child marker is clicked, the other child markers of its parent are displayed thanks to the fire method and the markerclick parameter, which refers us to the TimeNav.js method

File TimeMarker.js

```
//Fire event that this marker has been selected
this.fire("markerclick", { unique_id: this.data.unique_id, zoomLevel: this.data.ZoomOnClick ,parent: this.data.childOf });
```

Once in the TimeNav.js file _onMarkerClick is called.

File TimeNav.js

```
marker.on('markerclick', this._onMarkerClick, this);
```

```
_onMarkerClick(e) {
    if (e.parent) {
        let parents = this._removeBlankSpace(e.parent);

        // Find the child of the parent and highligh it
        for (var i = 0; i < parents.length; i++) {
            let index = this._markers.findIndex(x => x.data.id == parents[i]);
            if(index == -1){
                //skip loop
                continue;
            }
            let childs = this._markers[index].data.parentOf;
            if (childs.length > 0) {
                childs.map((child) => {
                    //clear space
                    var str = child.replace(/\s/g, '');
                    $('.' + str).addClass('highlighted');
                })
            }
        }
    }
```

This method searches for each parent, the children of this one.

Then, it applies for each child a highlighted class.

The ToolTip

If a marker is clicked, a tooltip appears when hovering over its children.
The text of the tooltip is to be defined in the Google Sheets table in the column TypeOfLink.

n°ID	parentOf	childOf	TypeOfLink
E0034		E0035	Conséquence
E0035	E0034,E0043		Cause indirecte
E0036		E0035	Cause direct

In this example:
“Engouement pour la praire en r...” is E0035
“La pêche aux coquilles Saint-Ja...” is E0034
“Décret tail...” is E0036

File TimeMarker.js

In order to create a tooltip, the _initiateToolTip function is used with the concerned parents and children.

```
//initialization of tooltip on only parent and child
this._initiateTooltip(this.data.parentOf);
this._initiateTooltip(this.data.childOf);
```

```
_initiateTooltip(data) {
    if (!Array.isArray(data)) {
        return;
    }
    data.forEach(element => {
        var str = element.replace(/\s/g, '');
        $("." + str).mouseenter(function () {
            $(".tooltip-" + str).removeClass('tl-tooltip-hidden').addClass('tl-tooltip');
        })
        $("." + str).mouseleave(function () {
            setTimeout(function () {
                $(".tooltip-" + str).removeClass('tl-tooltip').addClass('tl-tooltip-hidden');
            }, 1000);
        }
        )
    });
}
```

The function changes the class of the tooltip of the children and parents if the mouse passes over it.

Also if the mouse comes out.

The change is made via the classes tl-tooltip and tl-tooltip-hidden

Implementation of the tooltip on each element with a TypeOfLink

```
if (this.data.typeOfLink != "") {

    this._el.tooltip = DOM.create("div", `tl-tooltip-hidden tooltip-${this.data.id}`, this._el.timespan);
    this._el.tooltip.innerHTML = this.data.typeOfLink;
}
```

On Blur

When a user clicks on another marker or elsewhere on the Timeline, the markers, tooltips, and opacity are reset.

These actions are performed with the _onMarkerBlur function see below.

File TimeMarker.js

```
_onMarkerBlur(e) {
    this.fire("markerblur", { unique_id: this.data.unique_id });
    $(".tl-timemarker-content-container").css('opacity', 1);
    $(".tl-timemarker-content-container").removeClass('highlighted');
    this._deinitiateTooltip(this.data.parentOf);
    this._deinitiateTooltip(this.data.childOf);
}
```

```
_deinitiateTooltip(data) {
    if (!Array.isArray(data)) {
        return;
    }
    data.forEach(element => {
        data.forEach(element => {
            var str = element.replace(/\s/g, '');
            $("." + str).unbind('mouseenter');
        })
    });
}
```

The _deinitiateTooltip function allows you to remove mousenter and mouseleave binds from tooltips that have become useless.

Otherwise the tooltips would remain displayed on the hover even after choosing another marker.

Setting up a zoom when clicking

Setting up

In order to display a specific zoom when a marker is clicked, in the Google Sheets table, the ZoomOnClick column is filled in and defines the zoom level associated with the event.

ZoomOnClick
10
2
4
10

By default, the zoom value ranges from 0 to 10, with 10 representing the highest zoom level.

Recovery

File ConfigFactory.js

Definition of ZoomOnClick for the TimeMarker class

```
ZoomOnClick: row['ZoomOnClick'] || '',
```

File TimeMarker.js

As for the remote relation, use the fire method with this.data.ZoomOnClick as parameter.

```
//Fire event that this marker has been selected
this.fire("markerclick", { unique_id: this.data.unique_id, zoomLevel: this.data.ZoomOnClick ,parent: this.data.childOf });
```

Then we add in _onMarkerClick the desired zoom level.

```
// Set the zoom level
if (e.zoomLevel != "") {
    this.setZoom(e.zoomLevel);
}
```