Project Documentation

Interactive Learning with HistoGlobe

Winter Semester 2014/15

Virtual Reality Group
Faculty of Media
Bauhaus University Weimar

Authors:
Marcus Kossatz
Chris Hornischer
Max Weber
Sebastian Köpsel

Supervisor :
Dipl.-Med.sys.wiss. Patrick Riehmann
M.Sc. Henning Gründl
Prof. Dr. Bernd Fröhlich

HistoGlobe®

VR-Systems Group
Bauhaus-Universität Weimar

Contents

1 Introduction

HistoGlobe is a interactive website designed for people who want to get an overview of history. The appear of this events in function of time on the map helps you to get an impression of the behave of history.

With HistoGlobe there is much less need of unwieldy things like big maps of historical places or books and many piece of paper. Everyone should be able to learn our past in a nice, intuitive and simple way to be prepared for future.

This is HistoGlobe.

2 User Centered Design Approach

HistoGlobe wanted to be helpful and usable for a specific target group: history teachers in schools. Therefore we looked for a school in the area of Weimar that could serve as a location for a field study. We found a school in Jena, 25 km east of Weimar, that offered us to develop an instance of HistoGlobe directly for the usage in class.

Lobdeburgschule in Jena-Lobeda is a public school for all students from grade 1 to 13. A history teacher in grade 12 invited us to conduct a field study in his class to test HistoGlobe directly in school. This gave us the chance to develop the visualization in a User Centered Design approach thoughout the semester. Two members of the project group went every two or three weeks to the teacher in Jena in the time from October 2014 until April 2015. We presented new concepts, asked specific questions about the interface and the usage of the visualization in class and new problems and questions about the concept raised that had to be clarified until the next meeting.

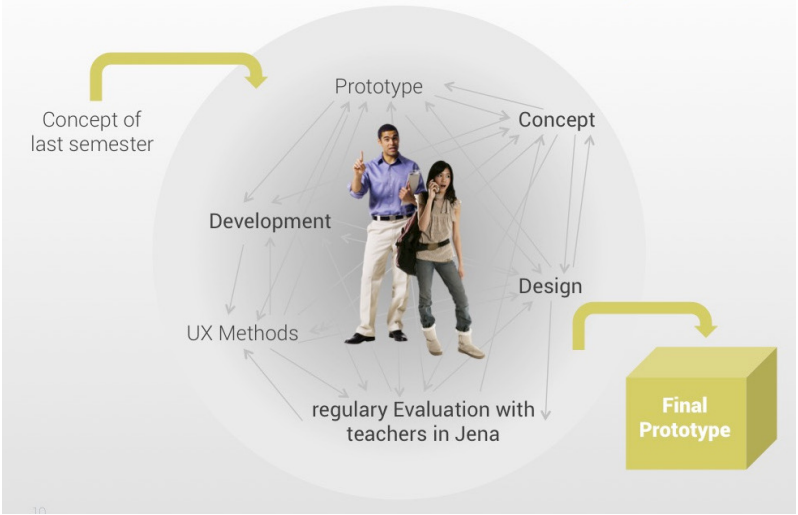


Figure 2.1: User Centered Design

Design Iterations

We had a lot of different design concepts. On the one hand we wanted to maximize the utility for the teacher to help him convey the necessary information in class but on the other hand design HistoGlobe in a way that we found suitable. We played around with orientation and the functionality of the timeline, the information about historical events on the map or the colors of the interface.

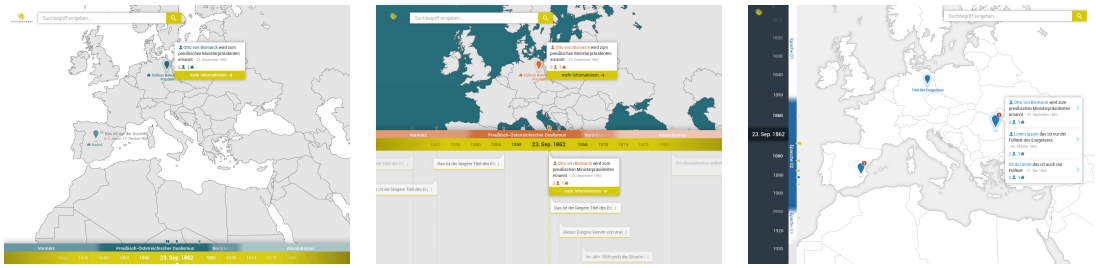


Figure 2.2: Several iterations of the design throughout the semester

In the next chapter we want to introduce the final elements of the user interface that were the result of the design iterations with the teacher.

3 User Interface Elements

The interface consists of five main elements: The **Map** is the central element showing the current countries with their names and their borders and the position of historical events, calles **Hivents** happening around the current date. This date is set on the **Timeline** which allows to control the temporal dimension: Set a new date and see the status of this day in history on the map. There are also **Topic Bars** on top of the timeline showing historical epochs in a specific time period. The sidebar on the right contains a **Search Bar** for retrieving information about historical events and a **Hivent List** for hivents of the selected topic. If an hivent is selected, there is a **Hivent Box** opening presenting the name, a short description and an image or video about the Hivent. Additionally there are **Control Buttons** for zooming the map or timeline, toggling the full screen or high contrast modus for better readability in problematic lighting conditions in the classroom.

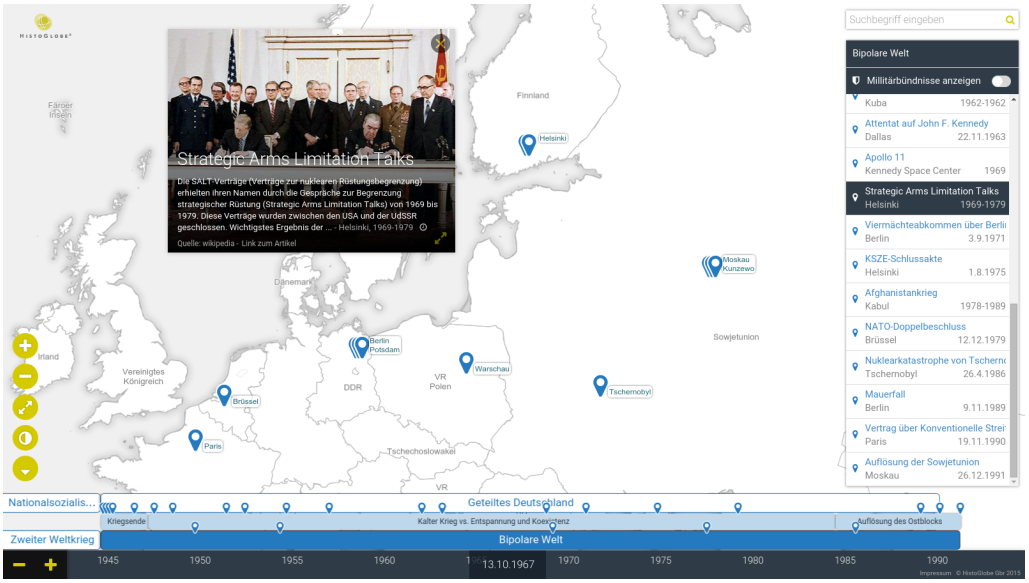


Figure 3.1: The final User Interface of HistoGlobe

3.1 Map

The map shows the status of the countries on Earth at this certain moment of history set in the timeline – the **NowDate**. For this project we used a self-made dataset of historic countries of whole Europe from 1945 until today and from Western, Northern, Southern and Central Europe from 1871 until 1945. We organized the data in a way that we can visualize historic changes of countries on the map. Finally we provided a functionality to style the country areas due to a current theme, for example all countries belonging to NATO get a blue background color.

3.1.1 Historic Countries

A country consists of an **area**, represented as a multipolygon geometry and a **label** with the name of the country and the position of the label.

Areas Everything is based on a dataset of the current countries in Europe from *Natural Earth Data* [1]. We extracted only the countries of Europe and loaded them into *QuantumGIS*, an open source GIS software for organizing, analyzing and visualizing areas on Earth. For each historic country we found an historic map online and created the area of the country using the *Vector Geoprocessing Tools* of QuantumGIS. Each area is stored in a single `area_id.geojson` file.

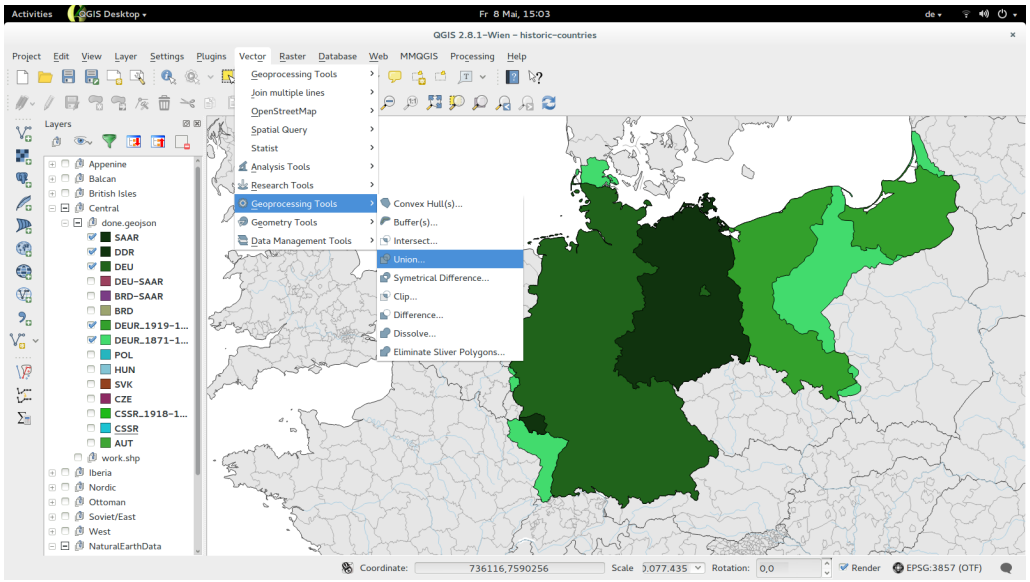


Figure 3.2: Geometry Manipulation of Historic Countries in QuantumGIS

In 3.2 you can see the areas of Germany from 1871 until today, from light green to dark green:

- 1871 - 1919 German Empire
- 1919 - 1945 Weimar Republic and Third German Reich (after WW I)
- 1945 - 1949 Occupied Germany (after WW II)
- 1949 - 1990 GDR without West Berlin and
  1949 - 1956 Saarland

Because of the very problematic usability of QuantumGIS and the mass of data that would have needed to be processed we have not reached the goal to create a database of all historic countries in Europe from 1871 on, due to the time constraint.

of a country are defined in an table consiting of the id, the name, the position (lat, lng) and a priority. Labels are stored separately from the areas to account for independent changes of names and geometries: A country can be renamed and borders can change, but both events do not need to correlate. The list of labels is stored in `labels.csv`.

[1] 1:10m Cultural Vectors | `http://www.naturalearthdata.com/downloads/10m-cultural-vectors/`

The visualization of the labels works with priorities: Each label has a bounding box. According to the label priority, the box increases inversly in size (low priority labels get large bounding boxes). Showing and hiding a label is based on the question if the labels collide, i.e. if their bounding boxes intersect. The algorithm works in the following way:

- If a new label L_n gets added to the map it checks for each shown label L_S^H with a higher priority if it collides with L_n
  - If so, L_n will be hidden
  - If it does not collide with any L_S^H, L_n will be shown. Now it checks for each shown label with the same or lower priority L_S^L if it collides with L_n
    - If so, L_S^L gets hidden
- If a label L_R gets removed from the map, it checks with all hidden labels L_H^L of the same and lower priority if they collided with L_R
  - If so, L_H^L gets shown
- If the user zooms in, the algorithm checks for each hidden label L_H if it can be shown now: it checks for each shown label L_S^H of the same or higher priority if it collides with it
  - If L_H does not collide with any L_S^H, it will be shown
- If the user zooms in or out, the algorithm checks for each shown label L_S if if collides with any shown label of the same lower priority L_S^L
  - If so, L_S^L will be hidden

The problem with this approach is that the label position and priority can not be deducted from the area. Therefore, both have to be set by hand. We are aware that this manual approach is not optimal, but for the scope of that project it is suitable. For the future, a data model should be found in which areas and labels are connected but can still change separately from each other.

3.1.2 Historic Changes

Because of the way areas and labels are organized, an historic change can easily be modelled, like seen in 3.1.

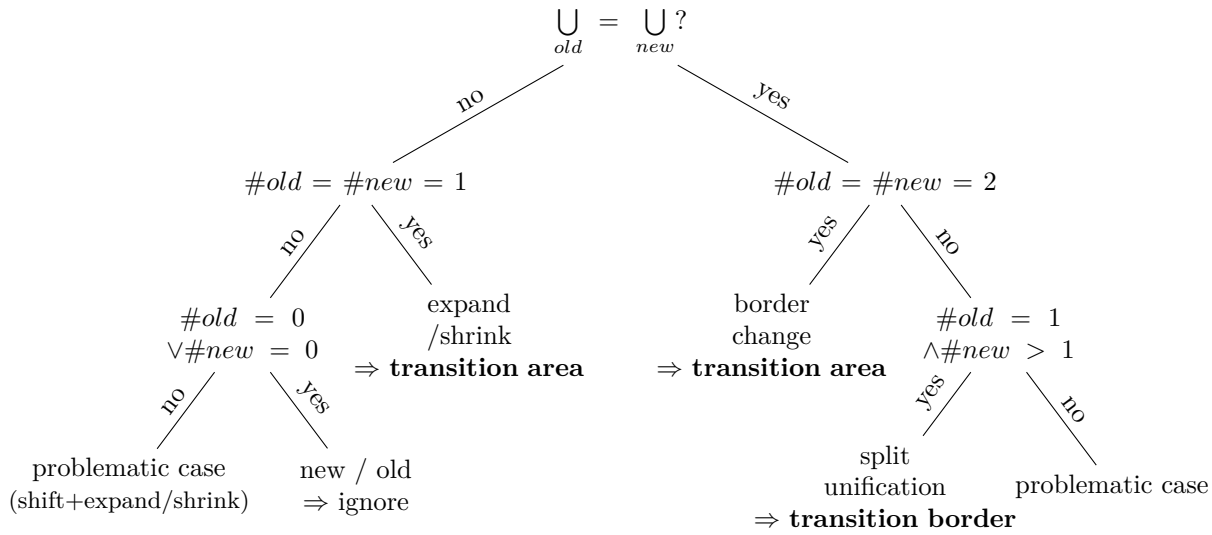
This event-based data model is maintained like this: if an historic change appears, it needs to have a date of change, a description of what happened, a set of areas that stop exist and a set that starts to exist from this date in history on – the same for the labels. Afterwards, the new areas have to be created in *QuantumGIS*, which is most of the work, and new labels have to be defined in the table.

In order to visualize the data on the client side, the areas, labels and changes have to be preprocessed on the server. Especially the transition areas and borders have to be generated.

Table 3.1: Examples of Historic Changes

date / **domain**	**name of change** / **old**	**new**
25.12.1991	Dissolution of Soviet Union	
area:	CCCP	EST, LVA, LTU, BLR, UKR, MDA, RUS
label:	CCCP	EST, LVA, LTU, BLR, UKR, MDA, RUS
03.10.1990	German reunification	
area:	BRD, DDR	DEU
label:	BRD, DDR	DEU
01.01.1990	End of Socialistic Republics	
area:		
label:	PR-ROU, PR-BGR, PR-HUN, PR-POL	ROU, BGR, HUN, POL
01.01.1979	Separation of Greenland	
area:	DNK-with-GRL	DNK, GRL
label:		GRL
01.01.1881	Init state	
area:		DEU-REICH-1871, POL-1871, ...
label:		DEU-REICH-1871, POL-1871, ...

Transitions are the geometric changes in a change event. There is either a transition area, which is the area that changes the membership of a country (e.g. Alsace-Lorraine 1919 from the German Empire to France) or a transition border, that splits two countries (e.g. The border between Czech and Slovak Republic after the dissolution of Czechoslowakia in 1991). These transitions shall be emphasized with an animation in the moment of the historic change so that it is clearly visible to the user what is currently happening. The transitions are generated like this: For each historic change the set of old and new areas are compared to each other and pass the following decision tree:



Legend: old = set of all old areas, *new* = set of all new areas
$\bigcup_{old}$ = union of all old areas, *#old* = number of old areas

Preprocessing happens with a *Python* script performing the following steps:

1. loading the areas (from *geojson*), the labels and the changes (from *csv*)
2. checking the set of areas and labels for completeness and the changes for consistency
3. generating the transition areas
4. writing the data to *json* files to be delivered to the client:
   - 4.1 `areas.geojson`
   - 4.2 `labels.geojson`
   - 4.3 `trans_areas.geojson`
   - 4.4 `changes.json`

The Workflow at runtime of the program can be seen in 3.3. The diagram is simplified focussing only on the areas, but the process is the same for the labels.

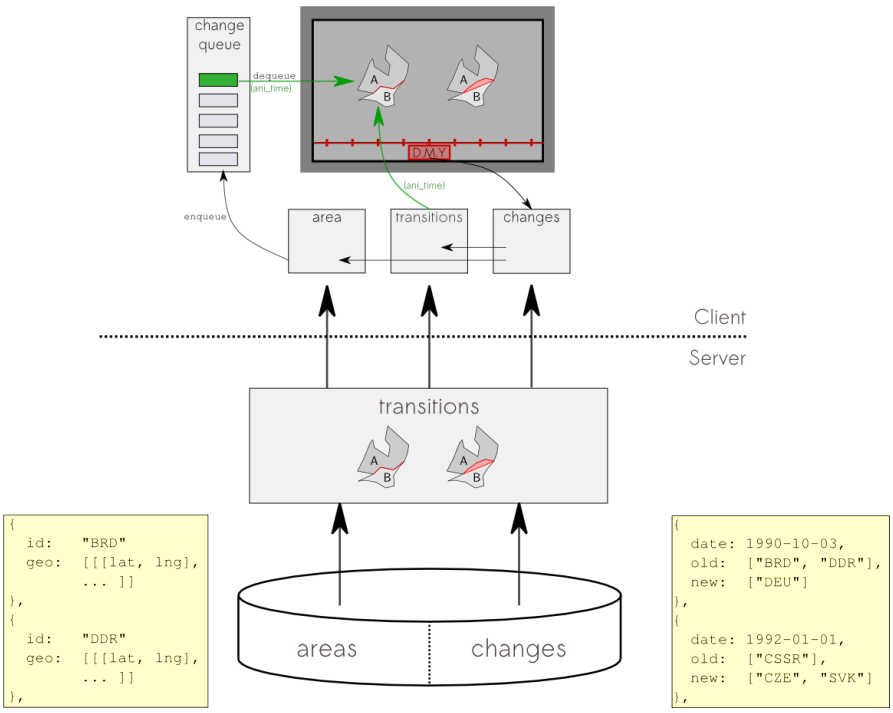


Figure 3.3: Architecture of historic countries on the map

The countries areas, labels and changes are created, preprocessed with the *Python* script and delivered to the server. The client gets the four *json* files and reads the data out of them. When the NowDate on the timeline changes, the timeline sends the old date and the new date to the Controller that finds out all the changes happened in that time period. From each change the transition areas and borders are faded in on the map and the related old and new areas and labels are enqueued as a change event in a **change queue**. Every 50 milliseconds the queue is processed: for the first change event it checks if the related transitions are fully faded in. If so, the new areas

and labels will be added and the old areas and labels deleted from the map. Finally, the transitions will be faded out again.

For moving the timeline backwards the mechanism is the same, it is just that old and new areas and labels are swapped, because the historic change happens the other way.

In order to prevent large amounts of changes on the map if the timeline is moved far, a rule-out mechanism is implemented: There is a list of old and new areas and labels for all historic changes in the period between the old and the new date from the timeline. Areas and labels that would be added in one change but deleted in another one are removed from both lists, because they would not contribute to the current state. With this mechanism it is possible to move the timeline at a high speed there and back and always get a consistent update on the map without irritating the user.

3.1.3 Styling the Countries

Each area and each label can have a certain style in a certain **theme**. An example theme we have implemented is called *Bipolar World* (Figure 3.4) and assigns each country in Europe a membership to the NATO (blue) and the Warsaw Pact (red) based on their dates of joining and leaving the alliance. Moving the timeline can lead to a change of the countries style.

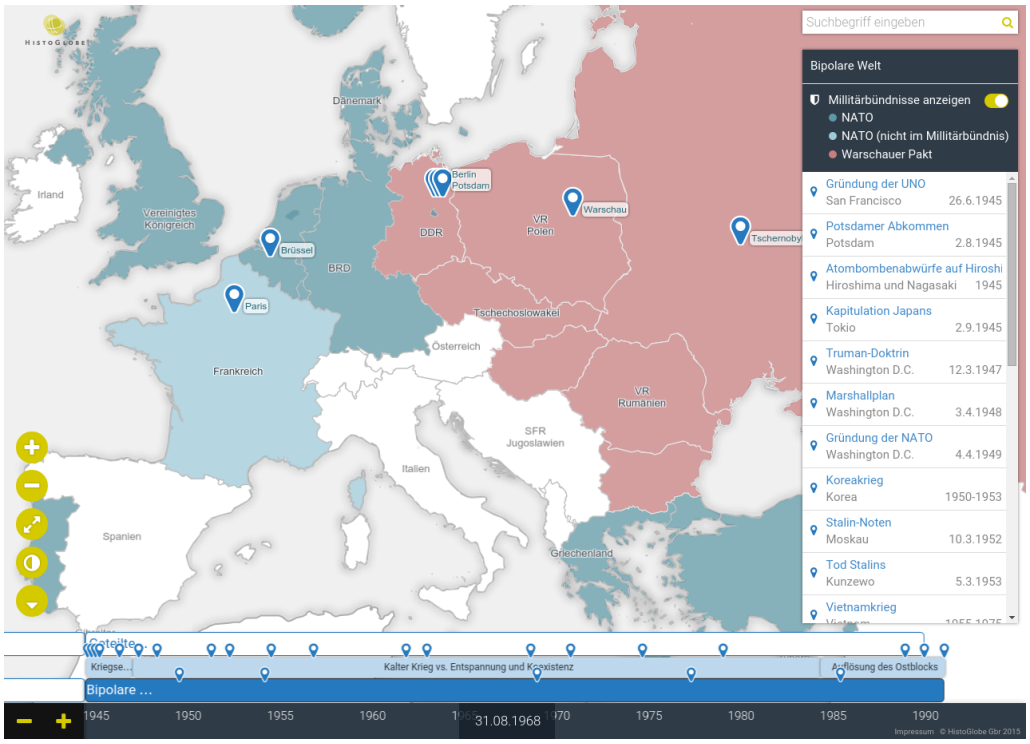


Figure 3.4: Countries colored due to their membership in NATO or Warsaw Pact

The style module was designed in a reusable way: for any new theme there have to be firstly **class**es and their styles defined (e.g. `class: nato`, `background-color: blue`). Secondly, areas and the membership to a class have to be stated in a table with `area_id`, `theme_class`, `start_date`, `end_date` (e.g. `ESP, NATO, 1982, 1986`). The same has to be done with labels.

3.2 Timeline and Topic Bars

3.2.1 General

The Timeline is our interface to control the time in HistoGlobe. It is located at the bottom. In the center on certain moment of history is displayed. With the plus and minus on the left you can zoom in and out. This is done by stretching and compressing the timeline. It is possible to chainge the actual date by pulling the timeline to left or right.

Figure 3.5: Timeline

3.2.2 Structure

One requirement of the teacher is to present time in history not linear but in epochs. The advantage is that he can give focus to special subsets of history. In our implementation we have two different types of epoch bars: German history and world history.

So our actual implementation of the timeline consists of four layers. The top layer describes the German history from "Deutsches Kaiserreich" to "Geteiltes Deutschland". The next two layers belong together. They represent the world history from "Imperialismus" to "Bipolare Welt". The upper of these two layers is only shown if one category of the world history is selected. Like in picture 3.6. Here "Imperialismus" is active and on top of this layer is the subtopic layer with specific topics. The bottom line is used to show the classification of epoch in time and to show the certain moment of history in the DD.MM.JJJJ format as NowDate.

Figure 3.6: Timeline Elements

3.2.3 Behaviour

The following section describes the individual functionalities of the timeline and topic bars. All elements are linked togehter so if you chainge one element, all other elements are chainged in the same way.

With the two control buttons on the left side you can **zoom** the time. We have linited the different displayed zoom levels to four steps and the displayed dates have a fixed distance (1, 2, 5, 10). The date change if they begin to overlap the thresholds of each other. So you can zoom in and out and the timeline is horizontal stretched or compressed.

To change the current date you can **pull** the timeline. You can also move the timeline by activate a Hivent. Then it is centered in the middle and the NowDate displays the date of the event.

One interesting task of the Epoch Bars was to **adapt** the labels **to diffent sizes**. In most cases the names of Epochs do not fit in their boxes. Therefore we tryed diffent ways to fix this problem:

- write epoch name in two lines (destroys layout)
- dynamic resize of Epoch Bars and no chainge of zoom level (historically incorrect because epochs overlap)
- dynamic resize of Epoch Bars and with adaption of zoom level (short historical epochs with long names destroys layout)
- leave out vowels (some epoch names are unreadable)
- leave out the middle of the word and replace it with ... (some epoch names are unreadable)

Finally, we have used a combunation of two replacement rules:

1. replace the entire word with an abbreviation
2. replace the end of the word with ...

To improve readability of the epoch names we placed it in the middle of the epoch boxes. If they come close to the edge they stick there with an threshold till the box box is to small.

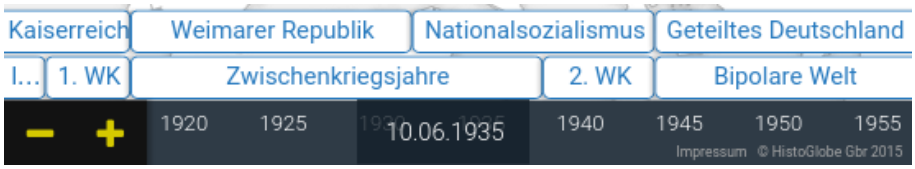


Figure 3.7: Timeline Elements with all improvements

3.3 Search Bar

One way to retrieve information about historical events is our search bar, which is a single-line text box with the dedicated function of accepting user input to be searched for. The user is able to search for events, people, places and time.

Figure 3.8: Search Bar with some search results in drop-down list

Instant Search One or more possible matches are immediately displayed while the user is typing text. So the search is sent automatically to present the user with real-time results which are displayed as a drop-down list. This often allows the user to stop short of typing the entire word they were looking for.

3.4 Hivent List

3.4.1 General

The central modul to navgate HistoGlobe is the Hivent List. Here you can see Hivents with additional informations in one list ordered by date. Each Hivent has a name, a date and a location. She is located on the right edge of HistoGlobe under the search bar. They share the vertical ammount.

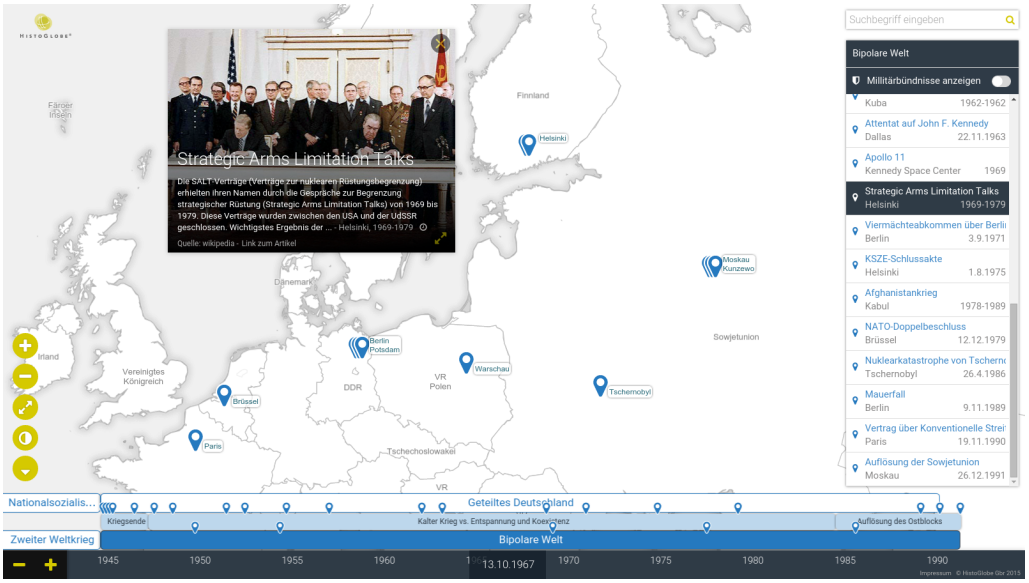


Figure 3.9: Hivent List

3.4.2 Structure

Aufbau: Titel, Options, List

3.4.3 Behaviour

Special Functionality: Knopp Clickbar, Hoverbar Slide into view on active On Search only Headline is shown dynamicly adapt to window heigth

3.5 Hivent Boxes

In HistoGlobe Hivent Boxes are the main method to display information beside the name, the location or the date of an historical event (Hivent). Every Hivent Box consists of a name, a short description about the Hivent, a link to the article on Wikipedia and multimedia content like an image or a video, if such a thing exists for the Hivent.

Figure 3.10: Hivent Box displaying information of an historical event with an image

Small Hivent Box The small version, as displayed in figure 3.10, is the default representation of our Hivent Boxes. This version is moveable on the map per drag and drop and consists of the same information like the the big Hivent Box, but with a smaller description length.

Big Hivent Box In figure 3.11 you can see the big Hivent Box. As mentioned above this version differs only in description length from the small Hivent Box and of course in its size.

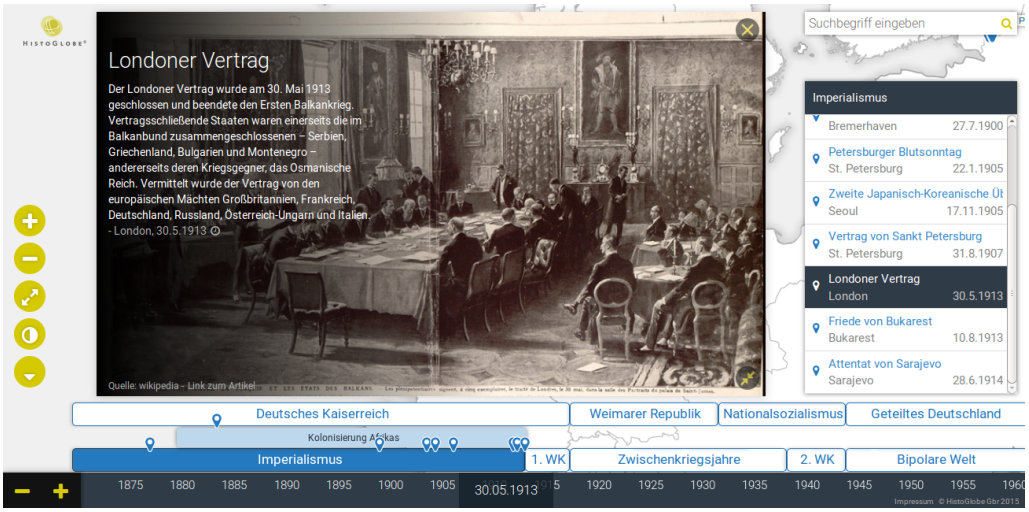


Figure 3.11: Big Hivent Box

Resize Button - Extend This button downright is only available in the small version of the Hivent Box and allows the user to switch to the big Hivent Box.

Resize Button - Compress This button downright is only available in the big version of the Hivent Box and allows the user to switch back to the small Hivent Box.

Multimedia Button This button on the right is available in both versions of the Hivent Box, the small and the big one, but only if there is existing multimedia content. With this button the user can switch to the multimedia content.

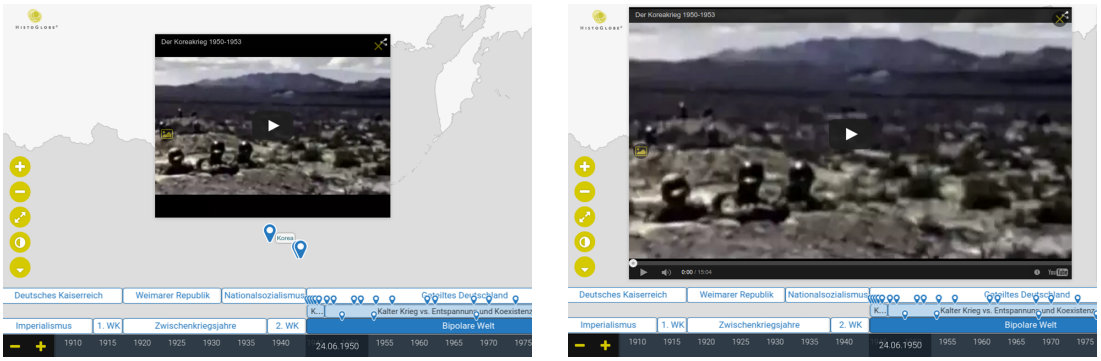


Figure 3.12: Multimedia content in small and big Hivent Box

Image Button This button on the left is only available in the multimedia view of the Hivent Box and allows the user to switch back to normal view.

Close Button This button in the right upper corner is available in every view and version of the Hivent Box and allows the user to close the box.

3.6 Control Buttons

Platzhalter

3.7 Hivents

3.7.1 General

Hivents mean "Historical Events", and are the way historical events are defined and presented. They are stored in a database, and have attributes such as name, location, start- and endyear, description and asocciated media. They are one of the most important concepts to visualize history in HistoGlobe.

3.7.2 Behavior

Hivents are represented on several locations in the UI, with the map, the hivent list and the timeline being the most important ones. Since usually more than one hivent is being shown, it's important to signify the representation between one hivent in different UI-Elements. The main interface events to which the feedback occurs are mouse hovering and clicking, which leads to the hivent being highlighted in all UI-elements.

Upon being clicked on, it changes its status to active. An active hivent gets focused in the map, it's marker are highlighted, it gets tagged in the URL bar and the hivent box opens.

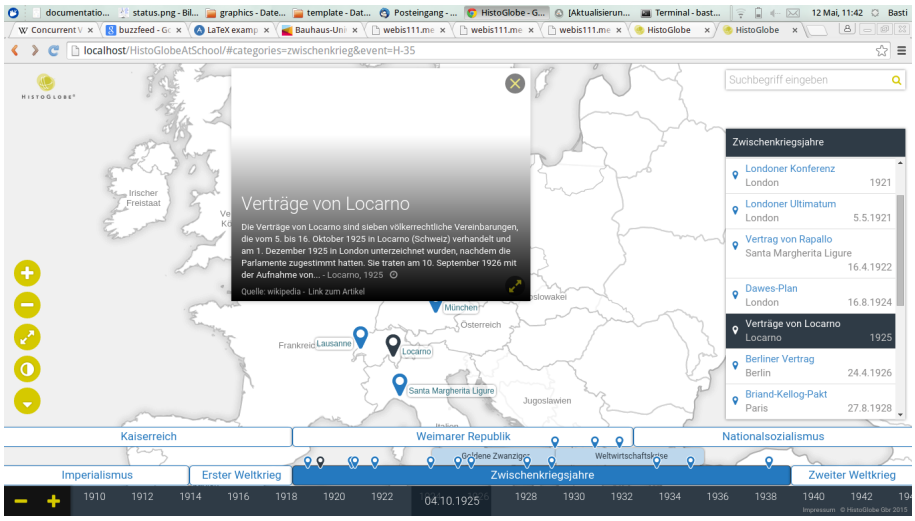


Figure 3.13: Activated Hivent

3.7.3 Labels on Map

Hivents on the Map were only represented by a marker, which automatically clustered when they are close together. The teacher demanded more information, so we attached labels to the hivent markers. We tried a Leaflet addon, but it didn't match our expectations of modifiability. To do labels our own, we used how the markers were implemented in the first space. They are Leaflet DivIcons so they are a div with a background image essentially. To accomplish easily modifiable labels we simply added another div element. In the first implementation the hivents name was shown, but we realized this isn't appropriate on a map, so we switched to show the hivents location. With this solution we got easily modifiable small weight labels. Overlapping labels are a known problem with some solutions todo:REFS. To adjust them to reasonable readability we needed fullfil special requirements. We needed an algorithm which was particulary leightweight, since we can't precompute the labels positioning and because we wanted to use HistoGlobe on schoolPCs and mobile devices. Our solution is done by checking for an overlap by comparing the divs Bounding Boxes and moving the left label to the left on overlap. On high zoom levels the amount of labels was so high the map wasn't readable anymore, so we remove them from a certain zoom level on.

3.7.4 Hivent Regions

A lot of historical events took place over a region, such as wars, so we wanted a region representation of hivents on the map. We implemented an additional type of map marker to do this. We added an additional optional attribute to the hivents, containing the polygon representing the region. The implementation was done using Leaflets Polygon drawing capabilites.

As seen in 3.15.

todo: Bild

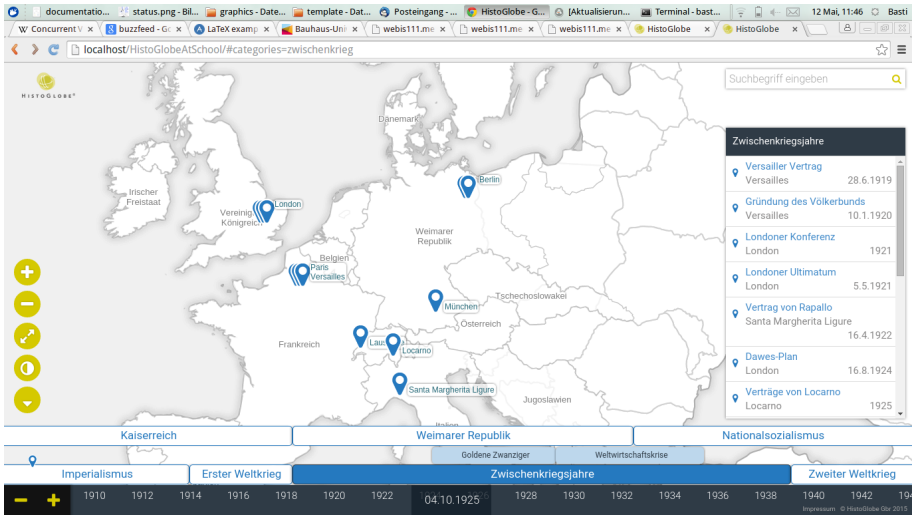


Figure 3.14: Overlapping labels

4 Field Study

Platzhalter

4.1 Usage as Teaching Material

4.2 Usage as Study Material

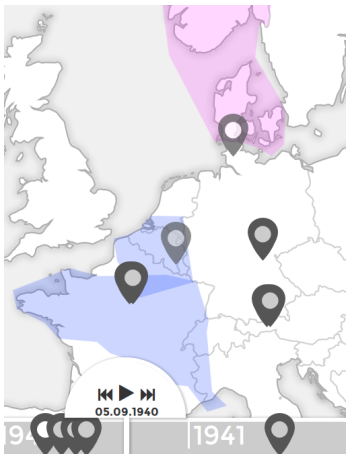


Figure 3.15: Hivent regions with highlights

5 Conclusion

User Centered Design We have implemented a visualisation, which is usable for the teacher and understandable for the students. We also had valuable feedback throughout the entire development process, but we did not strictly apllied the user centered design, which means that we had no prototypes or think aloud protocols. A negative aspect to name ist that we had only one teacher and only a few studets to test our implementation with.

Visualisation Our visualisation can be navigated via map and timeline. Historical events can be found through the Hivent List, the Search Bar, or directly on the Map or the Topic Bars represented as markers with supported linking. We are able to visualize areas and border changes and to show themes. Our high contrast and minimum UI mode allows the users to customize their view on HistoGlobe. Therefore we are able to visualize what where when and how happend. But we are not yet able to show why something happend, so the visualisation of coherences is not yet implemented.

Cooperation with Designer To the work with the designer has to be pointed out that it was very collaborative and very productive. He provided us with greatly designs. The only negative aspect to name is that this was no common project.

Organisation We have worked with git, which was very productive. We also had a masterplan, so everyone of us could see what to do next and which priority the particular feature to implement had.

Goal for the semester The teacher was able to use visualization and the students understood the goal of HistoGlobe. Our field study was sucessful and so we are proud to say that we reached our goal for the semester.

5.1 Study Results

5.2 Discussion

5.3 Future Work

5.3.1 Dynamics and animated transitions

While testing HistoGlobe we found out that the pupils wanted a better visualization of historical connectivity, in the form of more dynamic display of the data with better respect to the context of historical events. We think one should try to show the flow of history with more detail and animation.

5.3.2 Content Generation Backend

For our studies we used a self generated, wikipedia.org based database, which was sufficient for our first study. Nevertheless it would be great if a teacher or students could compile lessons and homework with a tool. This tool should be able to create, modify and delete hivents easily, and to add them in arbitrary form to categories. Another helpful functionality would be the automatic generation of a hivent from a given wikipedia article.

5.3.3 Country Generation Backend

For the future, an editor for storing, managing and analyzing historic changes would be desireable, because the data aqcuisiton part took a large share of the projects time.

5.3.4 Mobile Version

HistoGlobe was designed, on the UI and software level, to be usable on tablets and phones. But at this point, optimization is sub-optimal. Further tuning to style and usability are necessary.