5-2021

Storytelling Maps Classification

Konstantin Biriukov
kostya.igm@gmail.com

Storytelling Maps Classification

by

Konstantin Eduardovich Biriukov

A Thesis

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in Geography

May, 2021

Thesis Committee:
Mikhail Blinnikov, Chairperson
Jeffrey Torguson
Plamen Miltenoff

Abstract

Interactive maps play an increasingly important part in various fields including journalism, education, traveling, and entertainment, among others. Interactive maps require interactive user engagement. The scope of this interaction can vary from the basic mouse scroll to the complex logical sequence of steps including extensive toolboxes. One of the major applications of interactive maps is in storytelling. A map serves as a powerful tool to tell a story and modern technologies make this tool flexible and potent.

This research seeks to analyze and compare cartographic JavaScript APIs[1] and libraries, and to classify storytelling maps with a concentration on 'path visualization' type of maps and their technical implementation with an extensive review regarding the function of maps-related API/libraries work under the hood and their improvement. Additional material to this work, a web platform, demonstrates an example of each class and subclass of the classification. The proposed classification has been evaluated by reviewers working with interactive storytelling maps.

The web platform provides examples discussed in this work. The main chapter always references this platform, therefore readers have to adopt the web platform as an essential part of this work during reading: https://konstantinbiryukov.github.io/storytelling-classification/.

[1] *An application programming interface (API)* is a computing interface that defines interactions between multiple software intermediaries. It defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow, etc.

Table of Contents

Page

Chapter Page

List of Tables

List of Figures

Figure Page

Figure Page

Chapter 1. Introduction

Interactive maps play an important part in various fields. Journalism, education, traveling, entertainment are only some of the many fields utilizing interactive maps. Other terms such as “data storytelling” or “data-driven storytelling” refer to how insights are uncovered from data, interpreted efficiently, and translated into experience and emotions.

As interactive storytelling is an important topic, research conducted to make it more dynamic and efficient is a necessity. Currently, people have become used to dynamics and interaction in most areas of life and take them for granted, therefore maps should be properly fitted with this trend as well. The history of telling stories with maps in atlases and journalism leaves few studies explicitly addressing story mapping despite the recent popularity of data-driven storytelling and web-based story mapping (Margo et al., 2018; Song, 2017).

Dynamic and interactive storytelling involves an emotional component necessary for accurate perception of a story. This component is one of the main reasons for interactive maps to become so powerful and influential. Roth (2013) identified three main aspects of the relationships between maps and emotions: (1) the emotions that we place on maps; (2) the emotions that shape the mapping process and the map; and (3) the emotions experienced in response to maps. All of these aspects become increasingly obvious and stronger in relation to interactive storytelling. This work demonstrates the successful accompaniment of map visualization with photo, music, video, animation, and other visual and sound media to appeal to the user’s emotions more efficiently.

In general, storytelling maps contain numerous characteristics – complexity, efficiency, responsiveness, user interface, user experience, and emotional involvement. A map by itself is a great tool to trigger emotions, especially if a user has a special connection with the place that the map focuses on or has any memories or previous knowledge about the place. The mere addition of colors to a black and white map may immerse the user more deeply into the story. Dynamic visual tools and content as well as interaction with the user can dive deeper and become even more emotional, providing an immersive experience. Since emotional, social, and cultural factors have a significant influence on storytelling maps, an important topic of digital technologies, storytelling maps have a direct interference with the area of Digital Humanities.

The story professionally presented onto the map by combining different data sources, methods, and designs reduces the gap between emotions experienced while visiting the place and emotions experienced while exploring the storytelling map. The link between the emotional context of the map with the real perception of the place remains to be explored. The growing recognition to better understand these aspects of the mapping process may include the feedback from other disciplines such as artistic practice, storytelling, neuroscience, cognitive science, and geography to address this issue properly. The relationship between places and emotions is too complex to be captured only by maps, and mapping is rather too powerful to be deprived a central role in the attempt to better understand our emotional engagements with certain places (Harrower, 2008).

Nowadays, numerous specific skills are essential to build high-quality and efficient interactive storytelling maps. Moreover, the increasing popularity of interactive storytelling maps has produced a completely new job title transforming a *cartographer* into a *data journalist*. Data journalists have created a significant number of new forms of presentation and rhetoric together with new dimensions of the user experience which has been created in the last decade (Wallace, 2016).

The areas for storytelling data-driven maps application:

1) Entertainment: re-tell or complement the plot of a movie, book, article, any other kind of narrative.
2) Journalism: create a geographical chronology of any event.
3) Traveling: demonstrate the traveler’s path and highlight sightseeing.
4) Education and Science: explain numerous topics of geography, paths of birds migration, historical events, specific geographical unit or environmental object (country, city, park, monument).

The Master Thesis structure could be found in Figure 1.1. The main core of the Thesis consists of the proposal of the new classification and its practical implementation examples. The main core is preceded by two overviews to justify the proposal of the new storytelling maps classification and prepare a reader to use it as a developers' guideline.

The 2nd chapter provides an overview of the research about visual-storytelling maps. The literature review analyzes and explains the state of existing storytelling maps classifications, their distinctive features, and the premises for the classification proposed in this work. In addition, the chapter provides a brief historical overview, explains the meaning of the emotional component and the influence of the entertainment industry on storytelling maps.

The 3rd chapter includes a cartographic library overview. Cartographic JavaScript libraries are the main tools to programmatically work with maps. This chapter serves as a useful resource to analyze the features of these libraries before diving into interactive storytelling maps concepts and the new classification.

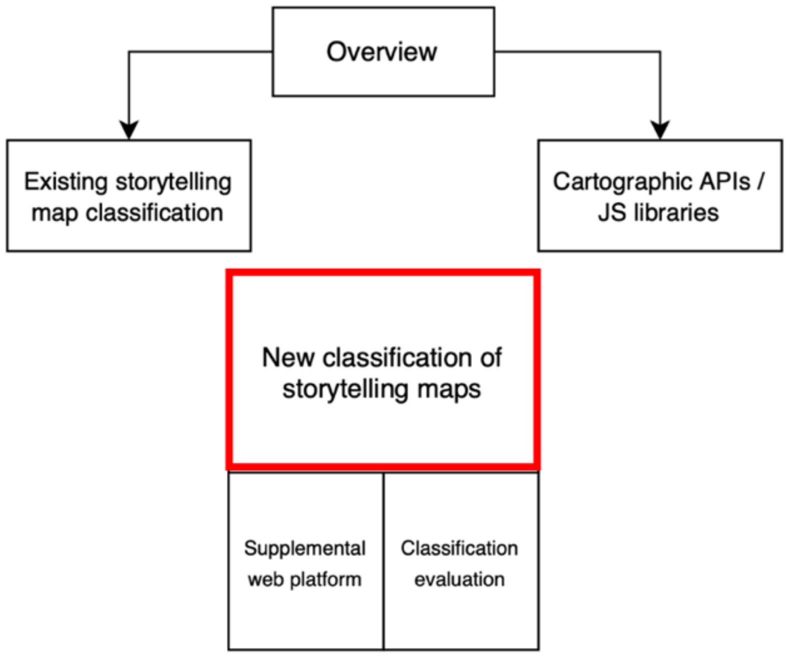
The 4th chapter serves as the main resource of this thesis and an extensive overview for each class and subclass of the new storytelling maps classification. A practical implementation example supplements each class. The chapter demonstrates the different ways of implementations, their pros and cons, analyzes the most efficient ways to build storytelling maps, and scrutinizes the details of those implementations. All the examples of storytelling maps classification are viewable on the web platform which is integral and essential material and a key part of this Thesis. This essential material is accessible online: *https://konstantinbiryukov.github.io/storytelling-classification/* (storytelling-classification web platform).

The 5th chapter provides classification evaluation and feedback collected from the targeted reviewers consisted of web developers, user experience designers, and journalists who specifically worked with web cartography and specialized in interactive storytelling maps.

The 6th chapter reviews final thoughts and future directions regarding research on map-based visual storytelling. This chapter discusses challenges faced during research and the project's contribution to different fields of knowledge.

Figure 1.1

Master Thesis plan scheme



Chapter 2. Literature review

History overview

The concept of animated maps, a direct predecessor to storytelling maps begins in the 1930-1940s (Slocum et al., 2005). Those animated maps were hand-drawn and extremely expensive to produce. They were designed to be shown in movie theaters, and it took months to draw them. When hand-drawn animated maps came out, storytelling maps followed suit almost immediately; for example, the hand-made animated storytelling map showing the German invasion of Poland came out in 1939 (Getty Images, n.d.). The animation demonstrates Germany's plan and invasion of Warsaw through Poland, encircling the city and cutting it off, forcing the surrender of most of the Polish army during World War II (Getty Images, n.d.). This story has numerous hand-drawn animations which contain a significant number of tiny details and the work with a camera. Moreover, this story's audible part gradually describes the process of the German invasion. Such kind of maps is supposed to be shown in cinema theaters and even after a century, this method is still effective, and useful for educational purposes. This early animation involved a significant number of static maps, and map animations were created via photography tricks (Thrower, 1959). With the development of computers in the 1960s, newly created animation programs enabled the growth of animation in mapping, and some of the first computer animations were developed (Tobler, 1970). Further development in animated maps was stalled until the 1990s due to financial constraints on research and reduced attention to the topic (Campbell and Egbert, 1990). In the 1990s, however, the invention of more efficient and accessible computers solved such problems and drew attention to the storytelling maps topic.

The state of storytelling maps evolved significantly in the last decade due to major enhancements in web-based visualization technologies. In the past five years, cartography has changed more significantly than in the previous fifty years, and currently, the field itself amid a significant revolution that has changed the whole perception of the maps, and how and why people use them (Harrower, 2008). More recently, those unprecedented changes became even more increasingly obvious and substantial. However, while technological developments in interactive mapping have become spectacular, cartographic science has failed to keep pace with practice (Roth, 2013).

A large amount of storytelling map practices remains undocumented and disorganized because of the underdevelopment of this branch. The increasing evolution of storytelling maps creates an expectation for new roles with different specializations, including map designers and map developers, a similar way of separation occurring in front-end web development[2]. Web developers usually work with the designs provided by UI/UX designers[3] who have more extensive knowledge and a better understanding of consumer behavior. In the case of the separation of roles, map designers work on the emotional component and the flow of the storytelling map, by setting the mood and the flow of the story, as well as triggering emotions and manipulating them for the purposes of the story. Their goal is to make the user experience convenient and pleasant. Meanwhile, web developers concentrate only on technical implementation.

[2] Front-end web development, also known as client-side development is the practice of building a website so that a user can see and interact with it.
[3] UI – user interface, UX – user experience

Design of the maps is extremely important. Sometimes a mapmaker sacrifices the data to improve the design quality of the map; for example, a mapmaker can decrease the amount of represented data to make the map more readable. A design-first approach is highly desirable for interactive storytelling maps. Numerous examples of the proposed classification explicitly use this approach (subclasses 2.2, 2.8).

Terminology

Cartographic interaction is an important topic for Geographic Information Systems (GIS), as it describes the manipulation of maps by the map user. Hereinafter the term '*interactive'* is used rather than '*dynamic'* to distinguish display updates evoked by the user (interaction) from display updates evoked by the system (such as animation, a form of cartographic representation) (Roth, 2013).

It is also crucial to clarify the definition of the '*interaction'* term. Some researchers insist paper maps are interactive by themselves and people visually interact with any map while they see and explore it (Dodge et al., 2008; Peterson, 1993). However, modern technologies have numerous sophisticated tools to create and manipulate cartographic representation. In this work *Digital interaction* also known as cartographic interaction is defined as the dialogue between a user and a map managed through any web interface.

Storytelling maps in entertainment: video games, movies, literature

Nowadays visual stories and story maps are more often created through code by technical professionals working in computer science, user experience (UX) design, data journalism, and web development (Roth, 2015). This significant change of map creators and techniques has prompted the revision of the storytelling maps classification.

A form of activity that significantly influences interactive maps is entertainment, especially the gaming industry and movie production. In video games, such maps are usually fictional, and sometimes they evolve throughout time because they are highly interactive and connected to a game story or specific in-game events. Games could have other distinguished optional characteristics such as the "Fog of war" feature. This feature was designed to hide unexplored territory from a player. This is one of the many unique features maps are missing in real. Interactive storytelling in video games is indirect and embedded into a video game itself. Moreover, video game maps are a special topic because games utilize game engines to render built-in game maps which are not related to storytelling maps for web browsers. Therefore, this topic is only partially discussed in this work. Nevertheless, video games have a large influence on web-based visual storytelling which has adopted numerous different features and has led to an increased degree of interactivity.

Narrative maps have been used by filmmakers and writers. Sometimes, these internal built-in maps help the audience follow the plot or implicitly explain some events and aspects of the story. Storytelling maps adopt classic features of filmmaking, such as working with different positions and camera angles, limiting the visual borders to emphasize the point of interest, adding the audio and video effects. A large number of features adopted in interactive maps prompted the creation of new different branches known as cinematic cartography and cartographic cinema.

Les Roberts (2012) has classified the different ways of representing the relationships between movies and maps through five “overlapping clusters”: “(1) maps and mapping in films; (2) mapping of film production and consumption; (3) movie mapping and cinematographic tourism; (4) cognitive and emotional mapping; and (5) film as spatial critique.”

Fiction and non-fiction writers extensively use maps. In some genres of fictional narrative stories, maps play a crucial role, especially in fantasy and science fiction, as they help readers to visualize the location of the events and relative interconnection of different places. Thus, the reader becomes more involved in the ‘universe’ and the story flow. It is difficult to identify these kinds of maps by the “*interactive*” term, using the meaning that is defined in this paper, although sometimes the story itself is driven by the map. Russell Kirkpatrick, a New Zealand geographer and writer who wrote the “Fire of Heaven” book series. Kirkpatrick produced the topographic maps of the territory before writing the text of his trilogy, so his story is placed in the “prepared territory” and afterwards the map simulates the story’s narrative. Russell Kirkpatrick emphasizes the importance of the maps for all kinds of storytellers: “I await the day when authors realize they can be as creative—and subversive—with their maps as they are with their text” (Crowe, 2017).

Fiction books are important to be mentioned because of the influence they have on web-based storytelling techniques. Currently, it is possible to make maps for books or movies using web technologies. A great example is an impressive map of The Witcher. The Witcher is a series of fantasy novels and short stories written by Polish author Andrzej Sapkowski and the map was made by Netflix (Netflix, 2020). This map demonstrates the timeline of the events in the fictional universe. The map contains various visual effects, such as snow and rain effects, very sophisticated animations, and a high degree of freedom for the user to explore the map.

Emotional component of the maps

The emotional component is crucial for storytelling maps. Mapmakers of dynamic visual maps have large set of tools to express those emotions onto the map than the mapmakers of static maps. The map is a representation of a place limited for transferring emotions. To build a map with emotional context emotions may require the utilization of computer technologies which offer more opportunities to transfer stronger emotional messages. The map is a powerful tool to trigger emotions just like any other type of entertainment, and it is critical to set up the appropriate mood of the users by establishing the visual tone of the map.

Emotional involvement helps users to understand data more thoroughly. An interactive way to share a story in most cases positively affects the user's involvement. For example, a large number of studies explore reader engagement with infographics versus articles with text only. Research has revealed that articles with graphical storytelling provide up to 34 percent more comments and shares together with 300 percent improvement on the depth of the page's scroll down (Mar Tech Series, 2017). The analogy with static and interactive maps could produce quite similar results.

The integration of emotional components into the cartography is widely recognized (Cartwright et al., 2008; Iturrioz and Wachowicz, 2010) from two major perspectives. The first perspective is related to expressing emotions associated with places. Different practical mapping strategies enable a better representation of the relationships between places and emotions which allows the user to become more involved and attached to the map with the help of visual technologies. Secondly, from a scientific perspective, a growing interest in research of cognitive cartography enables the study of emotional reactions to different cartographic designs as well as the targeted use of social media based on locations to collect emotions expressed in various places (Klettner et al., 2011).

Roberts (2014) provides an example of cinematic cartography by demonstrating the power of video for mapping with strong emotional context and by presenting and discussing his video (Roberts, 2010). This video follows the route taken by two-year-old James Bulger and his two kidnappers during the abduction in the United Kingdom in 1993. Contextualization has a key role in binding the place and story. Roberts calls this kind of video and map connection a “cinemapping strategy” that provides an “embodied spatial engagement” which derives the memories and places associated with the specific event. Visualizing or even filming the path (as Roberts did) is a powerful way to evoke emotions in a way a static map is barely able to do. Another paper emphasizes the importance of connecting maps with other media and options of expression to better capture the emotional connections that some stories have developed with places. Roberts believes in video-making as a spatial storytelling practice (Caquard & Cartwright, 2014; Caquard & Griffin, 2019).

Map constraints and flexibility

A significant number of researchers agreed upon a lack of objectivity regarding communication and transmission of the message from a mapmaker to a map user. In addition, the mapmaker can embed multiple layers of meaning within the map. These ideas are discussed particularly in the well-known geographic works of M. Monmonier "How to Lie with Maps" (Monmonier, 1996) and D. Wood "The power of maps" (Wood, 1992). Similarly, all users have their own set of skills, experience, and memories which help to retrieve various meanings from the story (MacEachren, 1995; MacEachren, 1998). Those meanings could not always be fully predictable, especially in complex stories. In that respect, interactive storytelling maps are usually rather sophisticated and multi-layered.

Work on storytelling maps requires precision to attract the user's attention into the right direction or, on the contrary, to assist the user with finding his or her own meaning in the "multi-layered by sense" map. Paper or static computer maps usually have less flexibility and complexity; therefore it makes easier the user's focus on the specific topic. For example, if a mapmaker enables the user to freely zoom in/out or to roam the entire planet, the user can be easily distracted and become frustrated with the story flow. Therefore, the constraints are an essential topic for storytelling maps or any other interactive maps which grant control to the user.

Different empirical studies indicate a need for increased interface constraint, or a reduction of cartographic interaction and/or the degree of freedom available to the user when interacting with a map (Roth, 2013). According to Dix and Ellis (1998), "virtually any existing static representation can be made more powerful by adding interactivity", and it is specifically important for storytelling maps. Other authors (Dou et al., 2010) conclude that "complete freedom of interaction may make problem-solving more difficult", therefore a storytelling mapmaker should always think about the balance between *constraints* in order to keep the focus on the story and *flexibility in interaction* to make the user experience more interactive, more emotional, and presumably more memorable as a result. The focus of attention matters for static maps, but the importance of focus is much larger for interactive storytelling maps, because of the degree of freedom provided to the user. The design of the interactive storytelling maps must be much more sophisticated and calculated because when using a highly interactive map, it is easier to distract a user with insignificant information.

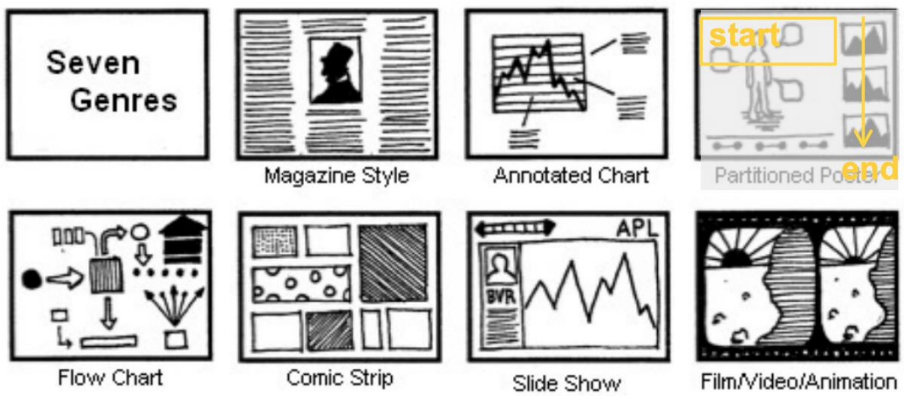
Previous classifications

Several classifications of storytelling maps currently exist in the field of cartography. Segel and Heer (2010) determine different “genres of narrative visualization” and distinguish visual storytelling genres by the number of frames, or individual sections, panels, slides, etc., which contain narrative elements and plot points. Segel and Heer (2010) define seven genres in their analysis of visual storytelling, whereas each can be interactive. These genres include magazine style, annotated chart, partitioned poster, flow chart, comic strip, slide show, and film/video/animation (Figure 2.1) (Segel & Heer, 2010). A significant amount of the work related to visual storytelling uses this classification.

Kosara and Mackinlay (2013) disagree with most of the genres proposed by Segel and Heer (2010). According to Segel and Heer (2010), genres are limited to printed newspaper stories, but nowadays, all dynamic content has a large influence from movies, television, and different visual types of dynamic entertainment. Currently, web technologies are a dominating trend in data journalism and visual storytelling today. Most stories fit map-based visual storytelling, as the events usually occur in geographic contexts (Song, 2017).

Figure. 2.1

Visual storytelling genres. Segel and Heer, 2010



The most recent taxonomy of visual storytelling classes specific to digital mapping was proposed by Roth in 2016. He divided the genres into 6 classes: news maps (encapsulating most of the genres for static maps proposed by Segel & Heer, 2010), dynamic slideshows, longform infographics, personalized story maps, narrated animations, and multimedia experiences (Table 2.1). Segel & Heer (2010) have built the classification based on the number of elements or frames, while Roth's taxonomy (2016) distinguishes each genre as a representation of a linear, interactive narrative. *Linearity* is a key component in Roth's classification (2016), and the cause of the linearity defines the class (genre). Linearity could be enforced by a scroll of the user, highlighting map elements, clicking and swiping through panels, hyperlinking of text, images, and graphics, etc. The key characteristic in Roth's classification (2016) is the way linearity is enforced (Table 2.1).

Roth (2020) determines the story map as a linear-three act spatial narrative referring to Aristotle's Poetics: set-up, conflict, resolution. Each act contains narrative elements recurring across stories that can be visualized as unique symbols within a map or its different frames and sections (Roth, 2020).

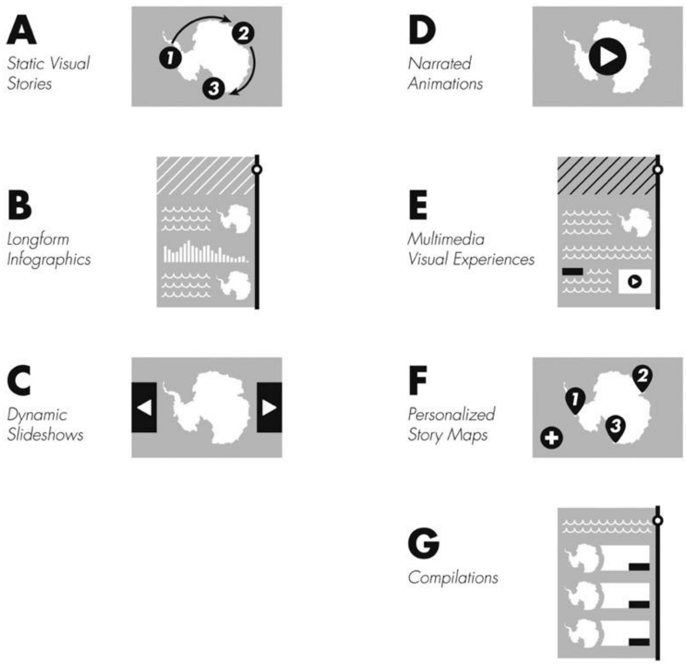
Table 2.1

The Roth's (2016) Taxonomy of Map-based Visual Storytelling Genres

Genre	Definition
Static news maps	linearity enforced by layout, highlighting, and annotation on the map
Longform infographics	linearity enforced through the browser window's scroll functionality
Dynamic slideshows	linearity enforced clicking or swiping through panels presented individually
Narrated animations	linearity enforced by the narration and advancement of time in the animation
Personalized story maps	linearity enforced by the user
Multimedia experiences	linearity enforced through the layout and hyperlinking of text, images, and graphics

Figure 2.2

Visual storytelling genres (Roth, 2016)



As depicted in the Figure 2.2. visual storytelling genres are distinguished by the interactive or visual methods used to enforce linearity in the story. The black components represent interactive elements used by the genre, serving as a distinctive characteristic of each genre (Roth, 2016).

The current classification was inspired by Roth classification (2016), but the key element is the interactive component of the map itself, instead of the linearity and the user action. An interactive component could be a path visualization, video, animation, music, scrolling event, slideshow, etc. The classification proposed in the current work concentrates on technical implementation and the techniques of story map visualization.

Principal prerequisites for new storytelling maps classification

Storytelling interactive maps have significantly evolved in the last decade. The taxonomy of Segel and Heer (2010) does not properly capture visual storytelling classes ('genres') which are currently possible and widely accessible by means of modern geo-related web technologies. As an effect, a number of frames cannot be a sufficient criterion for such a classification anymore. Roth's classification (2016) is more up-to-date and it is developed more 'from a user's point of view'. The key elements of the Roth's classification (2016) consist of the user action and the process of the story's progression along the way. The currently proposed classification was promoted 'from a developer's (mapmaker) point of view', where key elements are a dynamic component and element(s) of interaction (Table 2.2). This classification is intended to be more precise and fractional and to fit well as a guideline of how to tell the story, which tool or class to select, and which interactive component or a combination of them to choose. Thus, the thesis author (Biriukov) proposes this classification for 'builders' of the storytelling maps prompting this classification as a practical guideline. This is one of the reasons to include the cartographic libraries overview chapter before the main chapter proposes the new classification. To ensure understanding of this research by readers, it is necessary to analyze the main developer tools in order to fully comprehend a variety of storytelling maps represented in the classification.

Table 2.2

The prerequisites of the new classification

Previous classifications (Segel & Heer, 2010; Roth, 2016)	**New classification (current work)**
Based on “user’s point of view”: what user sees (Segel and Heer, 2010) or what user does (Roth, 2016);	Developers-oriented, “guideline” how to tell the story, which tool or class to select, and which interactive component or a combination of them to choose;
A key component is linearity, and the user action that caused this linearity of the story (Roth, 2016);	A key element is the interactive component(s) of the map itself;
Earlier interactive storytelling maps were not as popular;	Interactive storytelling maps became much more popular and sophisticated in the last 5 years. They started to prosper when the vector maps era has come (but with a delay);
Influenced by printed newspaper stories (Segel and Heer, 2010);	Influenced by dynamic content: television news reporting, movies and different visual types of dynamic entertainment;
Interactive storytelling maps were not identified like a special branch of geography and there were no specific specialists on this kind of maps;	Nowadays visual stories and story maps are more often created through code by technical professionals working in computer science, user experience (UX) design, and web development;

Chapter 3. Cartographic libraries overview

The cartographic JavaScript libraries intensely used about 5 years ago have evolved significantly. A mapmaker can use a variety of tools to create a storytelling map. Some of the tools do not require any programming skills and serve as constructors (ArcGIS StoryMaps, Mapbox Studio). This approach has significant constraints in functionality. Another approach requires programming skills using high-level cartographic API such as Google Maps API, Mapbox-gl, OpenLayers. These three ones are the major APIs, numerous others are not too popular or are still used but outdated and do not have official support any longer (like Leaflet). Those APIs provide map layout and different functionality for users to make API calls, creating any logic and interface, and using any additional JavaScript libraries combined with a cartographic API. Storytelling maps built on top of these tools are significantly more flexible, and developers have the flexibility to craft their own ideas by being able to add a large amount of functionality and to design it in their own way. Storytelling maps created via cartographic API technically are not significantly different from creating any web application that uses an external API. The goal of this chapter is to compare the main tools available to a developer in order to build a storytelling map.

Until the existence of Mapbox-gl, the foundation of the interactive cartographic world was based on the *raster* files, the maps themselves were built with tiles (*raster tiles*), and server-client interaction also was different. A server preserves all rendering logic, and after the rendering process is finished, the server sends rendered raster tiles to the client. As a result, the client was able to move the cursor, zoom in/out the map, create icons over the map. Any changes that involve re-rendering required client-server interaction. A new request to the server always had to be sent, resulting in significant and crucial overhead. All popular Javascript libraries for interactive maps worked on the principle of raster tiles or rendered images: Leaflet, OpenLayers, previous versions of GoogleMaps API.

In 2015 a new era of *vector tiles* has come. Google, Apple, Esri, Here, Maps.me, Mapbox, Mapzen, Cart, Deck.gl. Cesium – all of them have switched to WebGL[4] technology and vector maps. Currently, map rendering is implemented by WebGL locally in the browser. This technology allows the client's computer to leverage a video card for the needs of the browser. WebGL is a low-level 2D API and its entire responsibility is just drawing the triangles. Essentially WebGL is a fast technology for rendering triangles. Modern video cards are capable of drawing hundreds of millions of triangles per second. To draw something meaningful using WebGL, everything has to be initially turned into triangles. Specific frameworks (Three.js, Pixi.js, Babylon.js) do the same, but they are not suitable for maps, since they are still not fast enough to provide a smooth experience that is crucial for interactive maps. Developers of the major cartographic libraries leverage a native WebGL implementation under the hood of their libraries or develop their own engines using sophisticated algorithms of drawing polygons and objects out of triangles. WebGL technology which is built-in into modern browsers and those web engines created by developer teams of the cartographic libraries interact with the client's GPU[5] under the hood to enhance rendering capabilities, so the rendering process turns out to be hardware-accelerated.

---

[4] *WebGL* (Web Graphics Library) is a JavaScript API for rendering interactive 2D and 3D graphics within any compatible web browser without the use of plug-ins. WebGL is fully integrated with other web standards, allowing GPU-accelerated usage of physics and image processing and effects as part of the web page canvas.

[5] Graphics Processing Unit. Modern GPUs are very efficient at manipulating computer graphics and image processing. The GPU is usually presented on a video card or embedded on the motherboard.

The transition from raster to vector tiles has significantly improved the quality of maps, and maps with a high degree of interaction have become possible. Major distinguishing characteristics of vector maps:

1) Everything changes visibly smoothly in vector maps, without ‘jumps’ between scales. The map can be smoothly zoomed in/out or rotated, animations and transitions are also visibly smooth. This smoothness cannot be achieved by using raster maps. In vector maps data rendering is based on interpolation, therefore, when scaling, the thickness of lines and contours changes gradually.

2) The user receives full control over the display of data in real time without additional client-server interaction. If the user wants to add (or remove) a layer to (or from) data in raster maps or change the style of the map, then a request has to be sent to the server (Figure 3.1). Then, the server must re-render the map and send it back to the client, which is acutely ‘expensive' overhead taken place in raster tiles. With vector maps, all data and any changes are rendered on the client side, map styles can be changed on the fly, and even the shadows are smoothly redrawing in real time when the viewing angle around an object is rotated by the user.

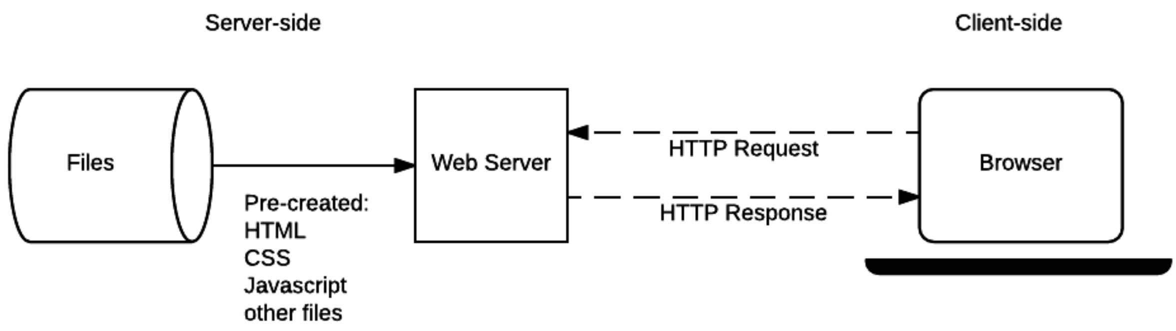
3) Any object on the map can be interactive: building, road, sign, etc. A user can change thousands of objects at the same time, and these changes quickly apply in real-time. Such a solution made it possible to create visual map editors in the browsers because all map elements can be interactive.

4) Perspective views (navigator camera angle), 3D objects, and video on maps are also made possible because of the vector maps.

5) Vector maps consume less traffic, namely 5-10 times less traffic in average.

Figure 3.1

Client-server interaction



The disadvantages of raster tiles become apparent if the development of the dynamic web maps with any data visualization is required. Raster tiles still can be used for static maps though, since in this case no interaction or changes on the fly are required and frequent client-server interaction is also not necessary.

Mapbox-gl is one of the major cartographic APIs. It is based on OpenStreetMap data source, a large database of cartographic data. Mapbox supports Mercator projection and for now,

Mapbox does not support other projections which might be only a temporary restriction. Most of the examples from this work utilize this library. You can find the comparison of the most popular cartographic APIs in Table 3.1.

Table 3.1

The comparison of the most popular cartographic APIs

	Leaflet	OpenLayers (ol)	Mapbox-gl	Google Maps API
Initial release	2011	2006	2013	2005
Original author(s)	Vladimir Agafonkin	MetaCarta	Mapbox	Google
Current state	Not developed any longer but still in use	In active development and support	In active development and support	In active development and support
Vector maps support	No out of the box support	No out of the box support (*ol-mapbox-style* plugin is required)	Native support (pioneers in vector maps technologies)	Native support
Advantages	Completely open-sourced; Very simple and lightweight	Completely open-sourced	Unique customization options; A lot of flexibility and features; Open-source SDKs[6]; Mapbox AR[7]; Offline mode; Native vector maps support;	Street View[8] feature; The best geographical coverage; Extensive language support; Native vector maps support;
Disadvantages	Not officially supported; No many features; Outdated design; Not optimized for mobile phones;	Outdated design;	Less geographical coverage;	Offline mode available only in Google branded apps, but no support in API; High pricing;

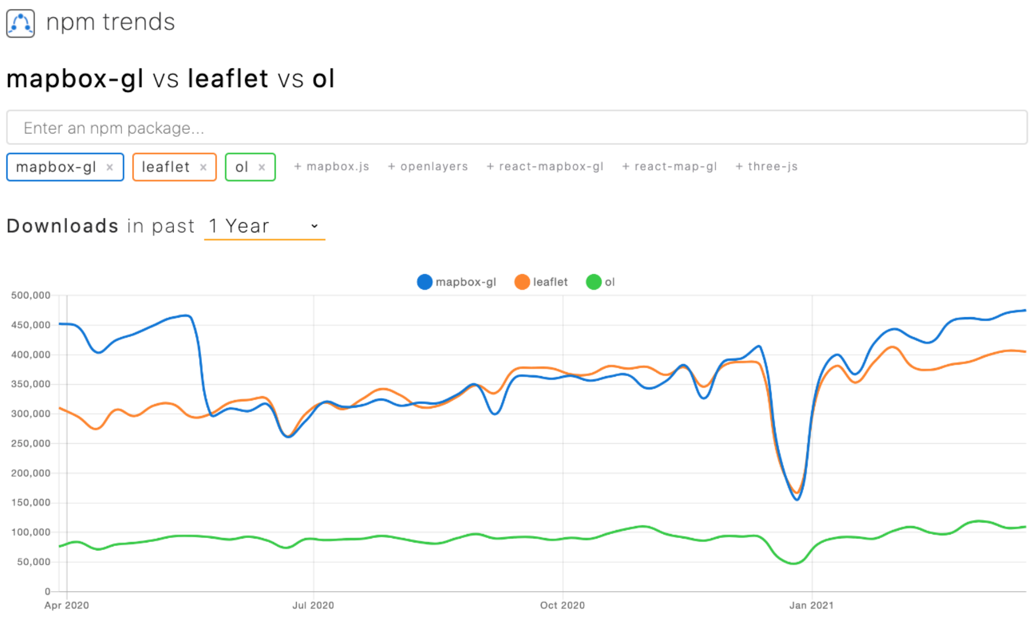
[6] *A software development kit (SDK)* is a collection of software development tools in one installable package. They facilitate the creation of applications by having compiler, debugger and perhaps a software framework.
[7] *Augmented reality (AR)* is an interactive experience of a real-world environment where the objects that reside in the real world are enhanced by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory and olfactory.
[8] *Google Street View* is a technology featured in Google Maps and Google Earth that provides interactive panoramas from positions along many streets in the world.

Notice, that there are many more cartographic APIs, but the ones mentioned in this work are the most popular nowadays. Other cartographic APIs which are worth to be mentioned are the following. *Carto.js* that is used for *Carto* platform. *Carto's* location intelligence platform is used for storing, enriching, analyzing, and visualizing spatial data to make insightful decisions. *Carto* is very specialized, related to specific *Carto* platform and does not have many features for storytelling maps. Moreover, various different cartographic APIs and libraries are working on the foundation of the major ones. Also, constructors do not require any developer's skills and work as web applications; for example, such popular constructors as ArcGIS StoryMaps and Mapbox Studio (not Mapbox-gl). Often all constructors have API which can be used to build more sophisticated maps. For example, ArcGIS StoryMaps and Mapbox both have constructors for building maps from the web browser interface and APIs for more low-level access to the functionality from the JavaScript code. Moreover, numerous libraries can be supplemental to the cartographic ones and to work efficiently in combination. For example, the d3.js library can be also used for creating storytelling maps, but this library is not cartography-oriented, and generally used for data visualization. Thus, a common use-case is to use robust tools provided by d3.js as a supplement library to Mapbox-gl.

Despite the fact the popularity of Leaflet is stable or gradually decreasing in the comparison with Mapbox-gl (Figure 3.2), it is still quite popular because of the existing legacy code produced throughout the last decade. Leaflet usually is not used for new projects anymore, at least not for interactive maps. This is a reason why Leaflet can still be found 'in a good shape' in the relation to a number of downloaded packages. This graphic (Figure 3.2) represents a comparison of npm[9] trends, and it displays a number of downloads over time. There is no official npm package for Google Maps API, so this package was not represented on the graphic.

Figure 3.2

Package download counts over time: Mapbox-gl, Leaflet, OpenLayers (2020-2021) (Npm Trends, n.d.)



	stars	issues	updated	created	size
mapbox-gl	7,379	825	Mar 29, 2021	Mar 7, 2013	minzipped size 214.1 KB
leaflet	30,387	671	Mar 29, 2021	Sep 22, 2010	minzipped size 39.9 KB
ol	7,901	177	Mar 29, 2021	Jun 20, 2012	minzipped size 66.4 KB

[9] *npm* (originally short for Node Package Manager) is a package manager for the JavaScript programming language. It is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry. The registry is accessed via the client, and the available packages can be browsed and searched via the npm website.

Thus, a transition from raster to vector maps has ensured the rapid development of web cartographic tools that influenced the rapid growth of interactive maps' potential. Currently, the development of map constructors and development tools flourish with three major cartographic APIs being the most popular ones: Google Maps API, Mapbox-gl, OpenLayers. The following chapter proposes the classification of interactive storytelling maps with the utilization of Mapbox-gl.

Chapter 4. Interactive storytelling maps classification

A variety of classifications exist for various kinds of maps, including interactive maps or map visualizations. Also, different classifications of interactive storytelling maps exist. Namely, the classifications of Segel and Heer (2010) and the classification of Roth (2016). Both classifications were specifically discussed in the literature review. These classifications do not significantly take into consideration existing modern web tools currently used to create interactive storytelling maps; maps built not only to demonstrate or visualize content but explicitly interact with users. These tools open the door for the creation of a variety of interactive storytelling maps and allow insights into existing classifications from a different perspective.

Modern web technologies provide sophisticated and powerful tools which give significant flexibility for representation of map data. There are numerous ways to present a map story and the classification demonstrates this diversity and flexibility.

This chapter often refers to the following web platform, an integral part of this work: ***https://konstantinbiryukov.github.io/storytelling-classification/***

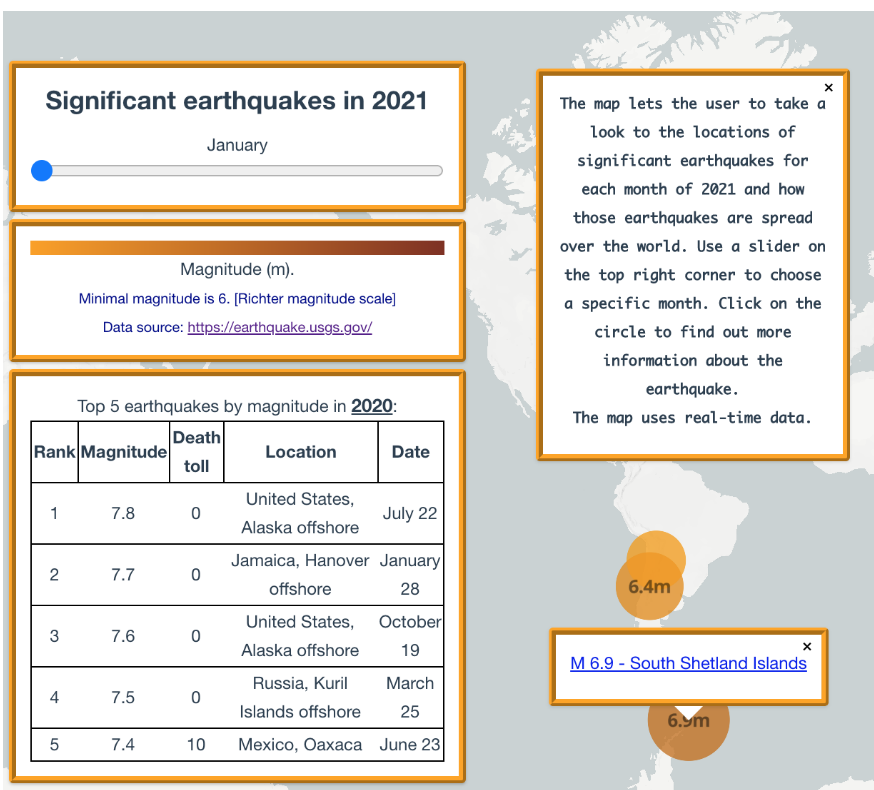
This platform presents live examples of each class described here. Since storytelling maps are highly interactive, it is difficult to express the ideas in photos or any supplemental static graphics, so each class and subclass of this classification is represented on this web platform. The text often refers to that platform and its corresponding parts. The reader can have a fully functional example of each storytelling map class, and a developer-mapmaker can choose the class which fits his or her story and to use this chapter as a kind of guidance.

Examples discussed in this chapter are always provided on the web platform. The whole chapter is a reference to this platform; therefore the web platform is always expected to be used by readers during reading as an integral part of this work. The web platform is designed specifically for this work as its essential part. Despite the photos from the web platform are available in the manuscript, utilization of the web platform is crucial to get a fully interactive experience that accompanies the text as conceived by the author.

All web platform's examples contain "helpers". "Helpers" are the popups map's explanations, they guide the users in their utilization of the map. Those "helpers" always have a clickable button in the right top corner to close them and to not intervene with user experience. The example of the helper represented in Figure 4.1. located on the right side of the image. Figure 4.1. provides an example taken from the subclass *2.2. Changes over time* (Figure 4.5).

Figure 4.1

An example of the "helper" popup



The classification was built by compiling the features from cartographic APIs mentioned in this work. Features available in cartographic APIs documentations are organized and included into classes and subclasses of the classification. The code for the web platform is extensively referenced in this work as well and can be found here: https://github.com/KonstantinBiryukov/storytelling-classification.

The technology stack used to build the platform is Vue.js 3.0, Vuex, Vue-Router, JavaScript, turf, d3, Mapboxgl.js, GitHub Pages.

The author abstracts most of the technologies, and only JavaScript knowledge is required for reading the chapter, but to fully understand the code the familiarity of all mentioned technologies is necessary. The author usually attempts to familiarize readers with the Mapboxgl.js functionality along the way, but the reading of official documentation is always the best comprehensive resource.

Interactive storytelling maps classification

Class 1. On-scroll maps

Class 2. Dynamic interactive maps

- **2.1. Fly to a Location**
- **2.2. Changes over time**
- **2.3. Control by sound**
- **2.4. Video**
- **2.5. Animation**
- **2.6. Slideshow**
- **2.7. 3D Polygons**
- **2.8. Markers & Popups**

Class 3. Path visualization maps

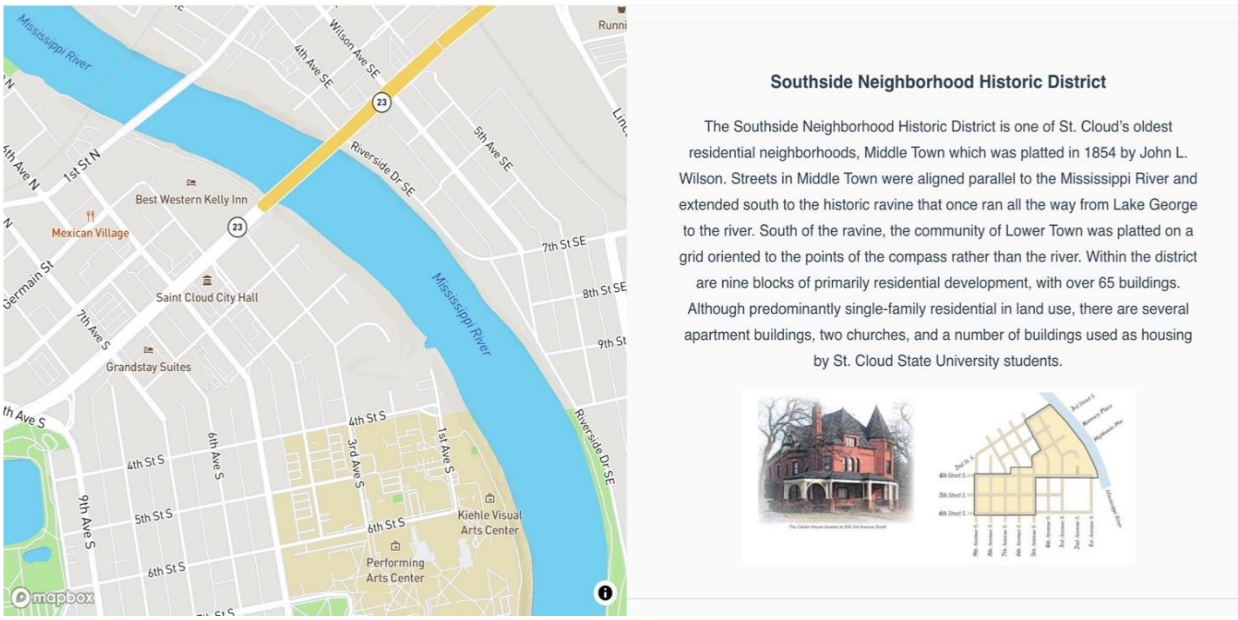
- **3.1. Path following**
- **3.2. Real-time route render**
- **3.3. The object path along the route**

Class 4. Complex stories

Class 1. On-scroll maps

Figure 4.2

On-scroll map



On-scroll (scroll-through) type of maps represents scroll-driven storytelling (Figure 4.2). The advantage of this type is that the user himself is the one who controls the flow of the story. The user is able to slow down or speed up the scroll and take the time to enjoy the story at a comfortable pace. Scroll-driven maps are based on handling a *scroll* event and this is a distinguishing characteristic that separates it from other classes of interactive maps. The on-scroll maps work well for the stories with multiple locations. For example, it works in a story that demonstrates multiple locations by changing camera position and angle according to the scroll position.

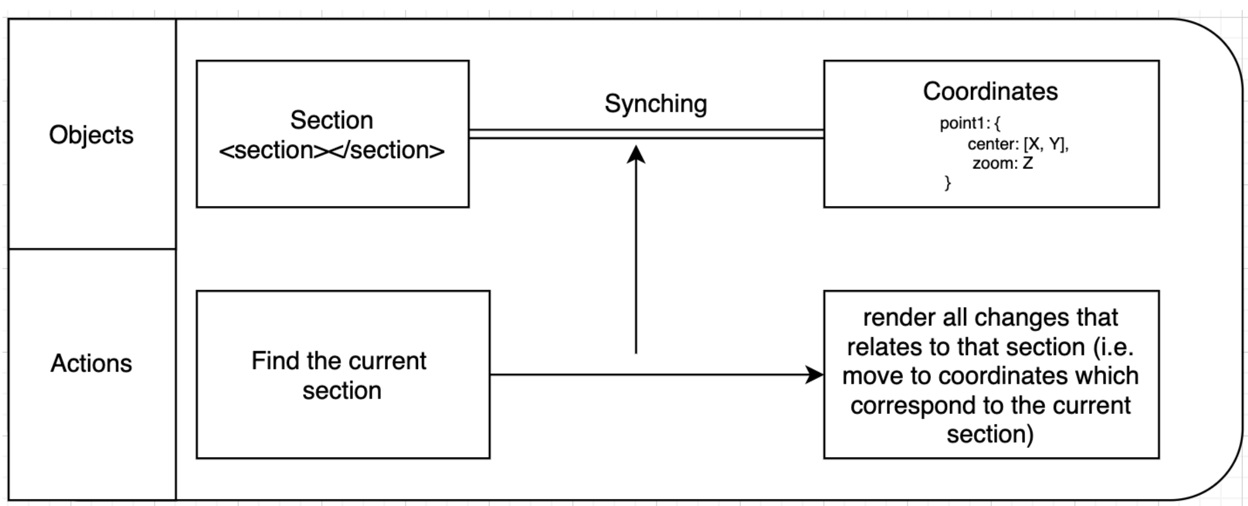
Implementation

The on-scroll map can be divided into several *sections*. The sections usually incorporate text; a reference to the place a developer intends to demonstrate (the result is represented in the Figure 4.2., right part). Nevertheless, sections can contain any other elements as well. The second element is the geographic representation of any points, path, or a specific highlighted object built by *coordinates* (file with points in geojson format) (the result is represented in the Figure 4.3., left part). Apart from coordinates, different camera options can be added, such as zoom value, pitch, bearing, etc.

At this moment these objects (chapter text and chapter coordinates) are existing separately. Building an on-scroll map requires a step of syncing the text story and map representation (Figure 4.3). Therefore, the next step is to add interactivity.

Figure 4.3

Structure of on-scroll map implementation



To make the map scroll-driven JavaScript *onscroll* event handler has to be used. Two actions need to precede the use of this handler, find the current section and render all changes that relate to that section. Therefore, the text sections built on the first step are a key part of code used in the main “rendering” code afterwards. A mapmaker has to connect those two parts together. To find the position of the current section *getBoundingClientRect()* method has to be used. This method returns the size of an element and its position relative to the viewport. The returned value of this method is a *DOMRect* object.[10] Then, a mapmaker has to incorporate and use the map coordinates previously linked to the specific text section.

Type of on-scroll maps

The web platform demonstrates basic on-scroll map (Figure 4.2.), but on-scroll (or scroll-through) maps can be quite sophisticated. For example, on-scroll maps can consist of interactive maps by themselves. In this case, when the user scrolls down the main map, the layers of the built-in interactive maps change one another. Besides, many of the subclasses presented in the following classes of the classification (Dynamic Interactive Maps and Path Visualization) can be easily incorporated into the on-scroll map. Those examples provided in the web platform resemble snippets that could be placed into any number of story experiences, and on-scroll map is one of the most popular and appropriate classes to incorporate features from different classes. Moreover, the on-scroll map can be designed for 3D terrain providing new context and perspective for storytelling.

[10] More about getBoundingClientRect at developer.mozilla.org

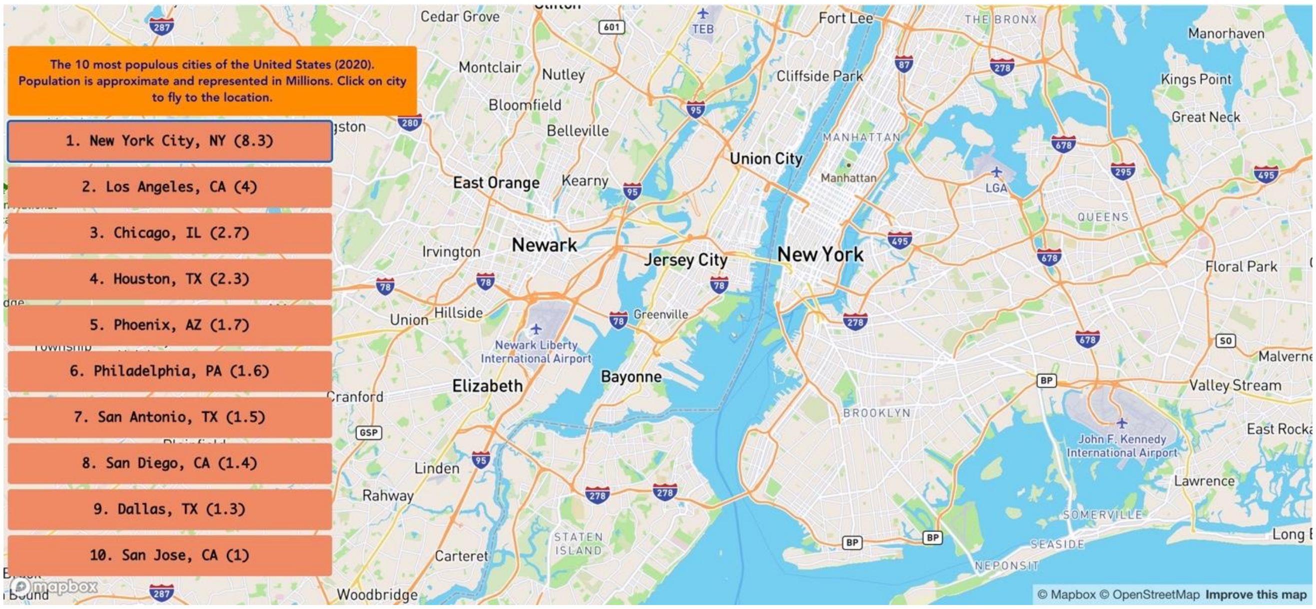
Class 2. Dynamic interactive maps

Dynamic interactive maps can include any kind of interaction including the non-static objects which are placed onto the map. Distinguishing features of dynamic interactive storytelling maps are not limited to audio or video resources to enhance the story's expression. Clickable objects on the map, dynamic icons, and polygons, 3D objects placed on the map, or any additional dynamic and interaction components provided by the map are valid interactive dynamic elements which play an important part in storytelling. This class has different subclasses and can be expanded by new additional features expected in the future.

Subclass 2.1. Fly to a location

Figure 4.4

Fly to a location



In order to demonstrate some places to the user, a mapmaker can set the predefined locations with coordinates and let the user smoothly travel among them. The "Fly to a location" feature is based on the *flyTo()* function, determining smooth interpolation between locations. The *flyTo()* function is called from the *map* object of *mapboxgl* and takes general options of the map object together with *event data*. The most important option is *coordinates*, where a mapmaker sets the central point for the camera to fly. The *flyTo()* function provides smooth interpolation under the hood, but the mapmaker can change the camera's behavior by passing appropriate pairs of property names and values to the function's *options*. A mapmaker can set multiple locations for camera flights as shown in the example of the web platform (Figure 4.4). By solely applying this feature, the mapmaker is able to create a story and translate sufficient scope of information through these multiple camera flights as shown in the web platform, but usually, the “Fly to a location” feature appears as a constituent part of a larger storytelling map.

Subclass 2.2 Changes over time

Figure 4.5

Changes over time

Usually, to compare maps or, more specifically, to see changes of any property over time, the user has to look to different maps trying to capture noticeable changes and evaluate them. Interactivity does not drastically change this process, but it makes it more pleasant by adding some degree of map control.

Changes over time can be represented in different ways. This class aims to present '*sliders*' for the users to control the flow of displaying the changes over time by choosing their desirable time range to see and compare the parts of the story. *Sliders* can be set for different time values such as year, month, or for any other properties specific to the topic.

In the web platform's example, the earthquake map is used (Figure 4.5). The purpose of this map is to enable the user to look at the locations of significant earthquakes for each month of 2021 (with a magnitude is equal to or larger than 6 on the Richter magnitude scale), and the respective location of those earthquakes around the world. The month is represented as a scale division of the slider. The slider is located on the top left side of Figure 4.5. The intended audience is people with interest in the location of major earthquakes in a specific year. Using this interactive map, they are able to find detailed information about each earthquake by clicking on the icons which contain clickable external links to the official data source (bottom right side of Figure 4.5). In this example, the author sacrifices the amount of represented data to improve general design quality. Thus, a user chooses a specific month for displaying earthquake data and the map is not overloaded with information about the earthquake for the whole year.

The used data source for this example is (Earthquakes database, n.d.). Their service provides API to get custom geojson files with required data.

In the current map this request was used: https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2021-01-01&endtime=2021-12-31&minmagnitude=6

If the aforementioned string would be copied and pasted as a URL[11] in the browser, a mapmaker retrieves a geojson file with complete detailed information about all earthquakes with a magnitude is equal to or greater than 6, which happened from the 1st of January to the 31st of December 2021. These settings (period of time and minimal magnitude) can be altered by replacing *query string*[12] values in “field = value” pairs, namely, *starttime, endttime, minmagnitude*.

Each earthquake represented by a circle is a clickable icon. When clicking on the icon, a user retrieves minimal information about the specific earthquake: magnitude and the distance to the major well-known place. If a user clicks on the hyperlink, the page with extended data will be opened. Data Source for this information is (Earthquakes database, n.d.). The data is fully dynamic and all earthquakes until the 31st of December 2021 will appear on the map. The display of the data is also controlled by the *slider*.

A graduated symbol map is used to represent the data. Symbols are distinguished by color and the size of circles represents earthquakes magnitude value from 6 to 7.8. A minimum magnitude of 6 was chosen because usually there are too many earthquakes with a magnitude less than 6. This example demonstrates a necessary or even forced restriction. The author sacrifices data for design purposes and changes the amount of data for the overall design quality of the map.

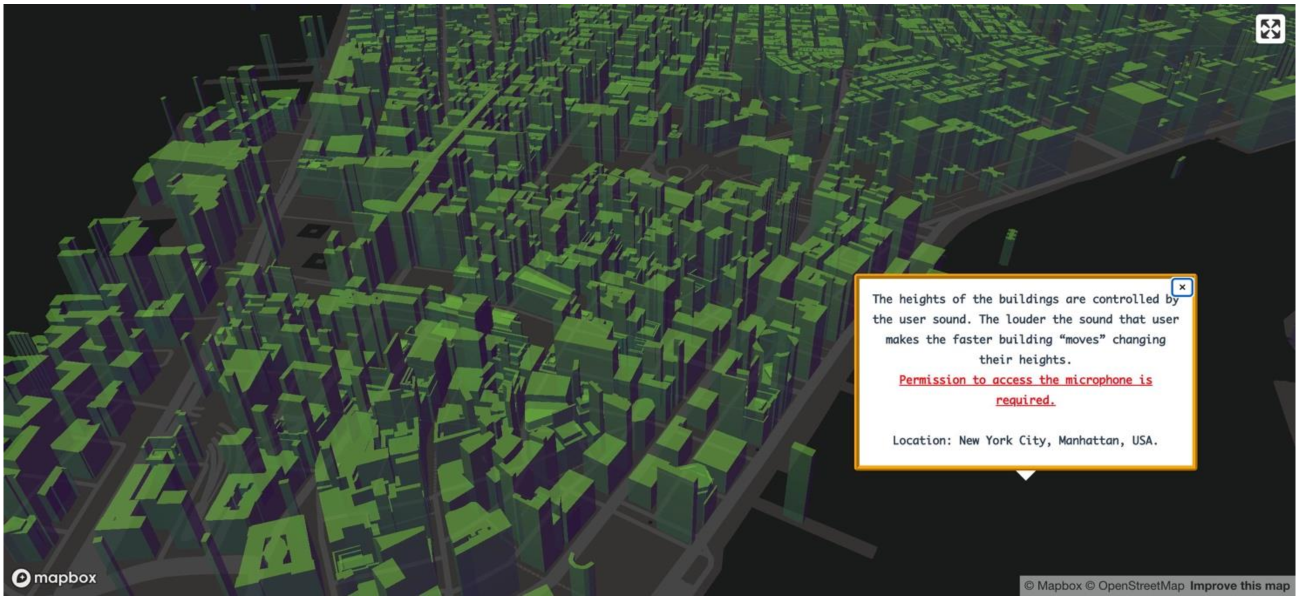
Other examples of changes over time could be found in Mapboxgl documentation: *comparison history maps* and *changes over time with dynamic bar*.

[11] Uniform Resource Locator (URL), colloquially termed a web address, is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it.
[12] A query string is a part of a URL that assigns values to specified parameters. A query string commonly includes fields added to a base URL by a Web browser or other client application.

Subclass 2.3. Control by sound

Figure 4.6

Control by sound



The user also is able to interact with the map by sound. The author assumes that this subclass will be significantly expanded in the following decade, and there will be more applications using this subclass.

This example leverages Web Audio API[13] in a combination with the runtime styling API. In this case, the sound produced by the microphone is tied to the characteristics of the buildings, namely such parameters of the sound as volume and frequency are tied to such characteristics of the building as the height and color. The louder the sound that the user makes the faster buildings 'moves' changing their heights. The colors are changing to the sound beat that the user produces, the more frequent the sound is the faster the color changes (Figure 4.6).

[13] *The Web Audio API* provides a powerful and versatile system for controlling audio on the Web, allowing developers to choose audio sources, add effects to audio, create audio visualizations, apply spatial effects

Subclass 2.4. Video (on top of the map)

Figure 4.7

Video

Sometimes when the users are exploring the maps, they are trying to visualize a specific place in motion: how this place can look in reality. Using video on the surface of the map is a good trick to demonstrate a specific location to the user. A user is able to see a part of the location he or she is looking at in motion. This way creates an extra layer of the map value, demonstrating not only the location of the place on the map but also the recorded video of exactly the same place from the perspective a mapmaker wants to show.

The example available on the web platform adds a georeferenced video on top of a map layer with satellite imagery (Figure 4.7). Three sources have to be initialized in the code: the source for satellite imagery, video source, and 4 coordinates (the points of the rectangle or square a mapmaker wants to fit the video into). Besides, a mapmaker can create the *click* events to enable pause and play video controls to add user interactivity. A good practice is to set a narrow limit for map zoom. Otherwise, a user can be easily distracted losing sight of the video.

Subclass 2.5. Animation

Figure 4.8

Animation

Animation in general is a convenient way to make the user experience highly interactive. At the end of 2020 Mapbox presented Mapboxgl.js, version 2, where they launched several new features:

1) *Camera API*, low-level API[14] which provides the low-level control of the camera;

2) *Sky API* to simulate sun's position based on time of day and specific geolocation;

3) All maps became available in 3D;

All these features provide an opportunity for cinematographic animations, where the difference between a map and a video is gradually erased. The combination of realistic 3D map views and experiments with angles and speed of the camera allows making quite sophisticated animations. This subclass can be used for different purposes which tend to be very

[14] A programming interface (API) allows the programmer to manipulate functions within a software module at a very granular level providing more flexibility.

cinematographic: to fly over a volcano's crater (an example of the web platform, Figure 4.8), to follow the airplane route, to rotate around a point of interest.

Subclass 2.6. Slideshow

Figure 4.9

Slideshow

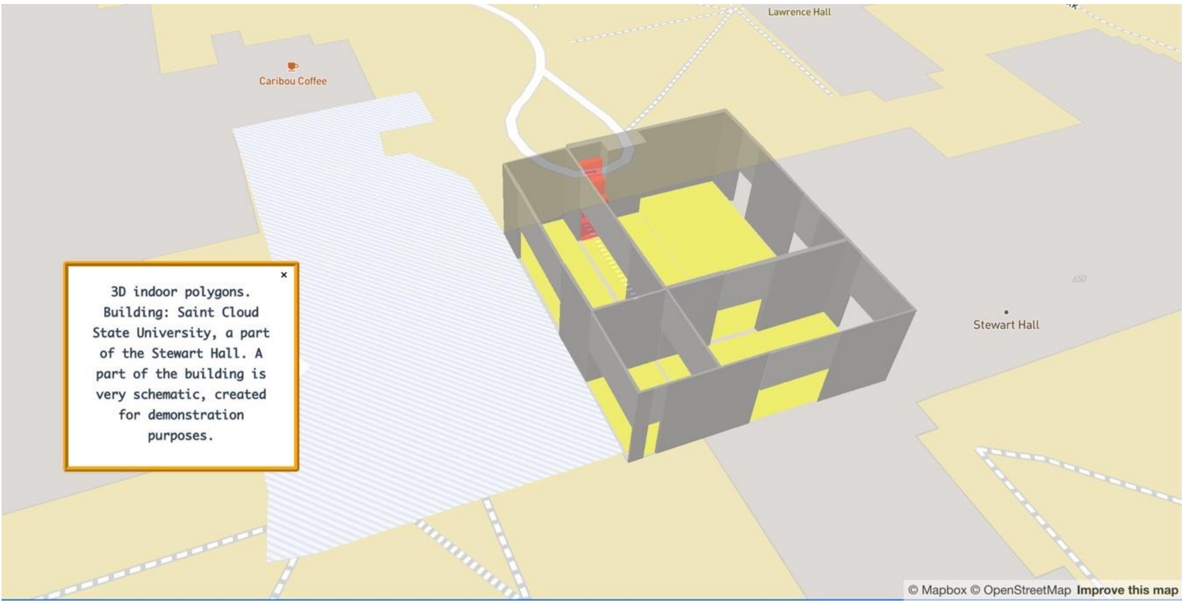
This subclass represents a storytelling map that can be produced by a group of slides with an autoplay function. In the slideshow subclass, the series of locations bounded to the specific coordinates and parts of the story are switching with a delay defined by a storyteller (Figure 4.9). Slideshow subclass uses a separate JavaScript object to define a collection of locations. Each location consists of *title*, *description*, *center*, and other optional properties. A mapmaker creates a custom *playback()* function and uses it with the *flyTo()* function to make an autoplay functionality available through locations on the map and to update the text description's overlay with a predefined time delay.

Since this simple Slideshow is not controlled by the user, the duration of the slide is shown on the screen is of a significant importance. In most cases, the user experience is much better when the user has control like in the On-scroll maps class since it is difficult to predict the perfect time that the average user would want to spend on reading the text and exploring the map. When a mapmaker does not provide any control, the user experience should be carefully calculated. Since the Slideshow subclass relies on delays between switching the slides, these delays should be properly calculated and tested to avoid the cases when the user does not have enough time to read the text or when the user is bored because the story spends too much time for one slide to be shown. A better approach is to grant some kind of control interface to a user; namely, the buttons that manage the slides in the slideshow. In this case, a user would be able to go back and forth, pause, and play the slideshow exploring the storytelling map at a comfortable pace.

Subclass 2.7. 3D Polygons

Figure 4.10

3D Polygons



A 3D polygon is a georeferenced object that stores geographic representation in a series of x and y coordinate pairs. 3D polygons represented as objects can be a part of a storytelling map (Figure 4.10). The user is able to check the indoors design of the building which provides extra information also valuable for specific stories. The polygons have numerous properties related to the painting of the object in a layer. Names of most of the properties start from *'fill-extrusion'*, since the component that these properties apply to is an extruded (3D) polygon. The most important properties are *fill-extrusion-color* (the base color of the feature), *fill-extrusion-height* (the height with which to extrude this layer), *fill-extrusion-base* (the height with which to extrude the base of this layer), *fill-extrusion-opacity* (the opacity of the entire fill extrusion layer). All these properties apply to an extruded polygon and all of them are optional. 3D polygons require a dataset with coordinates that can be produced using Mapbox Studio (Mapbox Studio, n.d.). Check MapboxGL documentation for more information.

Subclass 2.8. Markers & Popups

Figure 4.11

Markers & Popups General view

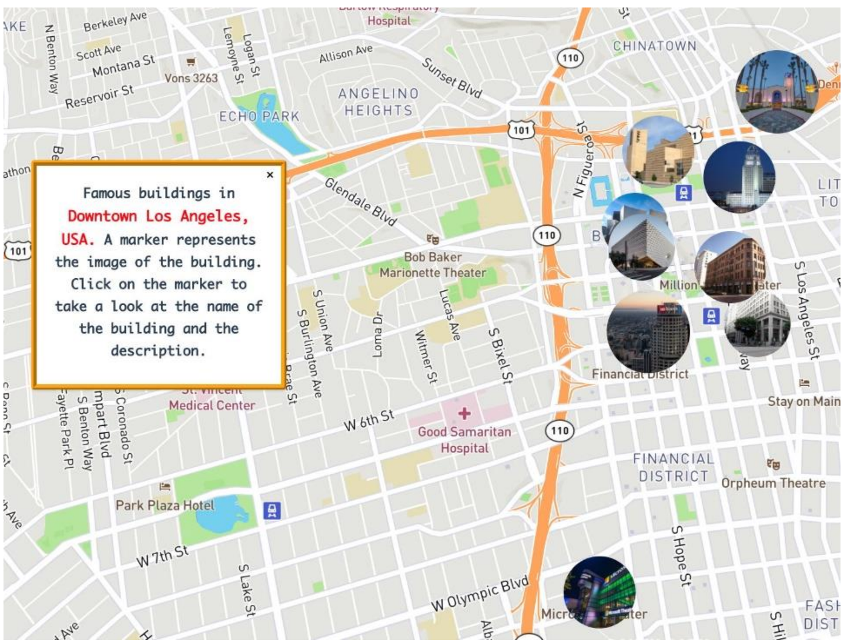
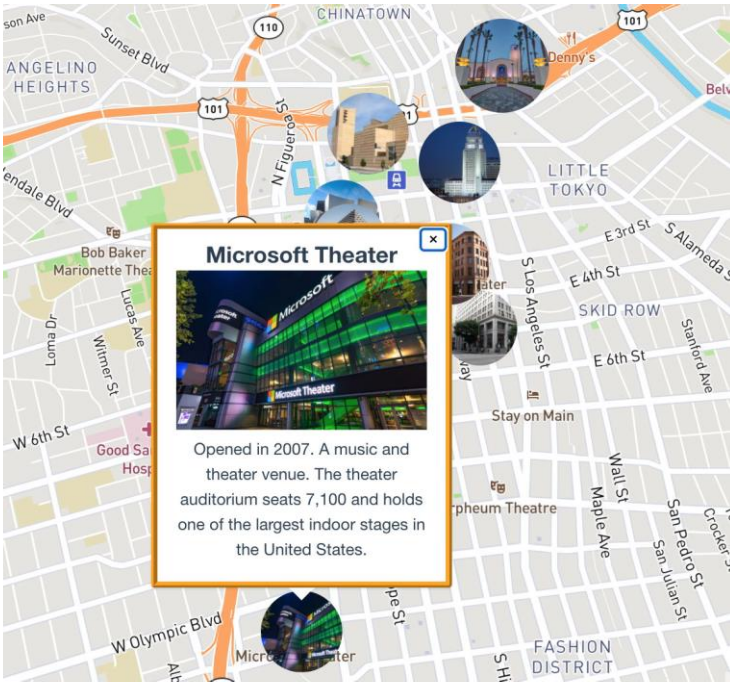


Figure 4.12

Markers & Popups: Popup of the Map's Marker



Purpose of the subclass

This subclass represents storytelling maps based on clickable icons (Figure 4.11). When the user clicks on the icon, a popup appears showing the information related to that icon (Figure 4.12). In this type of map, a user has full freedom to choose the order of clicking the icons, therefore all icons have to be independent. The marker and the popup objects exist outside the map's canvas element. The description in the popup can have any meaning, design, and structure.

Web platform's example

In the web platform's example marker represents the buildings of Los Angeles city depicted in a circle photo (Figure 4.11). The popup appearing by the mouse click has the following structure: a building name, a building image (facade), and a short description. In general, the marker image replicates the popup image (Figure 4.12), but sometimes the marker image represents a facade of the building and the popup image represents the photo that made inside the building. This approach can help to interpret the click on the marker as a metaphor to enter the building: by clicking on the building marker a user enters the building to find out more information about it and take a look at the interior. Usually, inconsistency (for example, different kinds of popup images) indicates a bad design approach, but this example intends to demonstrate the variety of techniques.

Mapboxgl documentation details related to Marker and Popup classed

The Marker and Popup are special user interface elements that appear as one of the Mapboxgl classes. Both classes extend the '*Evented'* Mapboxgl class which consists of the methods mixed into other classes for *event* capabilities. The Marker class takes an HTML element that represents an icon in the constructor. Besides, this class has various different functions. The most important ones are *setLngLat()* to set the marker's geographical position taken in the arguments as an array, *setPopup()* which takes Popup, instance of another Mapboxgl's class, and finally, *addTo()* function which takes map object as a parameter, where a mapmaker intends to add this customized Marker instance. The most useful options of Popups are: *offset* which provides pixel offset applied to the popup's location, *anchor* which indicates the part of the Popup that should be positioned closest to the coordinate set, and *maxWidth* that defines popup's maximum width. The most important Popup instance methods are *setHTML*() to set the popup's content to the HTML provided as a string, and *setLngLat()* together with *addTo()*

which work similarly to Marker instance methods. Documentation is available for a full list of options.

Implementation details of the example from the web platform

All information related to the “Markers & Popups” story is combined into a separate JavaScript object placed into an isolated file. This object represents an array of *places* that are objects as well. Each *place* represents a building with *id* number, *name*, *coordinates*, and *popupHtml* properties. The *name* property is used to bind it with the HTML DOM *id* property. *Coordinates* property is used to set marker’s coordinates in the *setLngLat()* method of *Marker* instance. *PopupHtml* property contains popup’s html (building’s name, photo’s path, and description) and it is used in the *setPopup*() method of the *Marker* instance. More precisely, *setPopup()* method takes *Popup* instance and then, *PopupHtml*() is used in *setHTML()* method of that *Popup* instance. The advantage of this technique is the ability of a mapmaker to change that JavaScript object placed in a separate file and the map will be appropriately updated without any changes in the main file.

Class 3. Path visualization maps

This part of the chapter provides an extensive overview of one of the major classes of storytelling maps - Path Visualization. The overview demonstrates the different ways of implementation, their pros and cons, analyzes the most efficient ways to visualize the path, and investigates what is under the hood of those implementations.

The path visualization allows the mapmaker to demonstrate the specific route in an interactive way. The path could be pre-rendered in advance. In this case, the camera just follows the route visualizing the path (Web platform, subclass 3.1). The path also could be rendered in real-time (Web platform, subclass 3.2). In this case, the path progresses by adding new coordinates to a feature in a *line* layer. This approach is useful for visualizing real-time data sources. The path also could be schematically represented in an object that moves along the route (Web platform, subclass 3.3). The web platform contains all those examples: Path following, Real-time route render, The object path along the route.

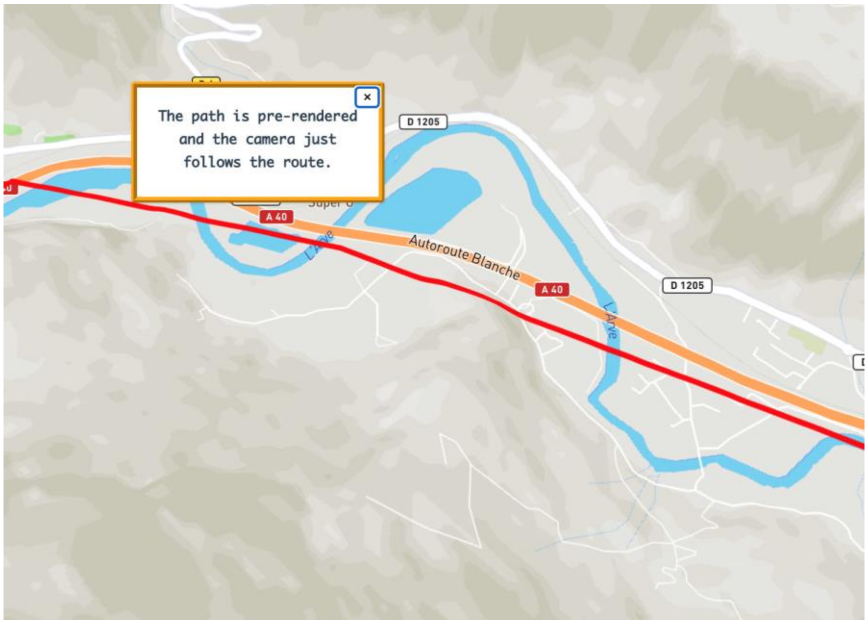
Like any other elements of the storytelling tools, path visualization can play a central role in storytelling or just to be a part of the more complex story. The web platform has an example of path visualization in the ‘Complex Stories’ class described later in this chapter. Path visualization plays a central role in an example of the “Complex story” that represented in the web platform. The current chapter concentrates on the different examples, technical implementation, and its particular features proposed by the author.

The most difficult problem that developers usually encounter during the implementation of the path visualization is optimization. The operation of path rendering in real-time on the user’s device requires decent computer performance because the computer renders a large number of points which is necessary in order to provide smooth rendering without transitions and gaps. To improve performance, it is possible to optimize rendering algorithms or minimize the number of rendering points. The last approach decreases the degree of smoothness. Another example to avoid optimization issues could be avoiding rendering on the user's computer. Such an implementation is used in Relive application (Relive, 2021). Relive is a Dutch technology company specialized in tracking activities via GPS, they turn the activity paths into animated 3D path visualizations. Those visualizations are pre-rendered using the hardware power of Relive company, then visualizations are turning into videos (Relive, 2020) and sending to the end user. In this case, there is no need to use the hardware of the user to render a large number of points, the user only receives a final video. Relying on the user's hardware is as dangerous as expecting the users to use the specific browser or a certain version of the browser. Consistency and equal user experience are the cornerstones of software development, therefore eliminating such unstable factors as relying on user's hardware or browser dependency should be prioritized. When the path is pre-rendered on the specific and predefined hardware, the user experience is more predictable and can be controlled. In this case, a video of decent quality can be obtained. Eventually, all the users receive videos with visualized paths, no matter the number of rendered points, the experience is equally smooth for every user.

Subclass 3.1. Path following

Figure 4.13

Path Following

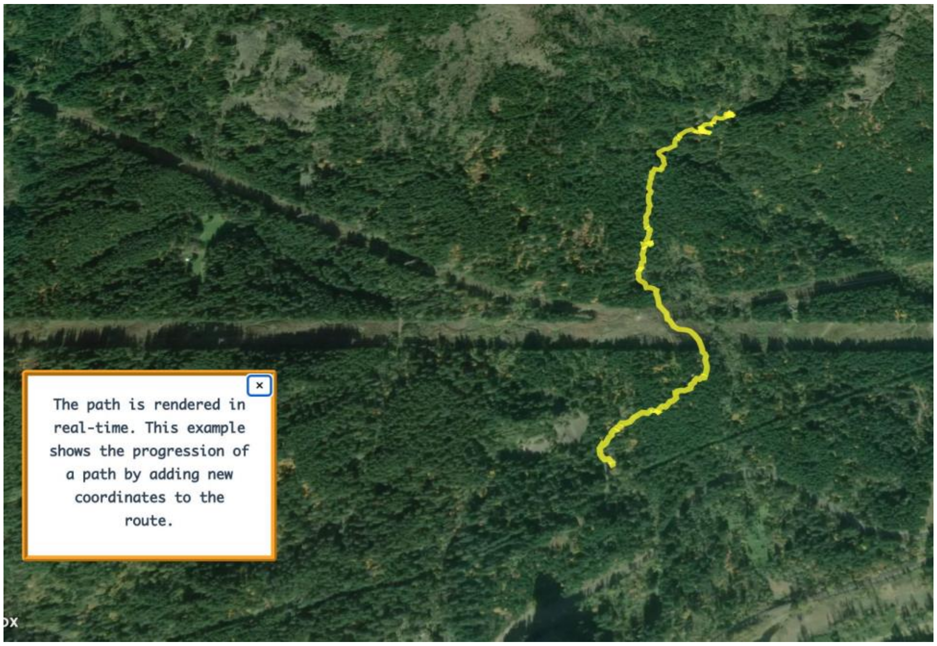


This subclass represents the path pre-rendered in advance. A mapmaker controls the camera that follows the route visualizing the path to demonstrate this drawn path to the user (Figure 4.13). It is convenient to use the techniques with file separation and place all coordinates into an individual file that contains a JavaScript object with arrays of coordinates: one type of array represents the points the camera will look at; another type of array represents the points the camera will move along. The main technique is to use interpolation along the route and to sync the camera and route positions ensuring they move at roughly equal rates even if they don't contain the same number of points.

Subclass 3.2. Real-time route render

Figure 4.14

Real-time route render



This subclass represents the path visualization that progresses in real-time by rendering all path points with a pre-defined speed. The path progresses by adding new coordinates to a feature in a *line* layer (Figure 4.14). This approach is useful for demonstrating the path that evolves step by step which helps the user to imagine how he or she would take this path in real-time, gradually covering the distance. It is also helpful for visualizing real-time data sources. As usual, the path coordinates are preserved in the individual JavaScript file and referenced in the main file. The algorithm is to set the interval and to update points from coordinates in order to simulate the path until the coordinates in the referenced file are not finished. Using *panTo()*

method a mapmaker pans the map to the specified location with an animated transition and the map is updated according to the interval.

Subclass 3.3. The object path along the route

Figure 4.15

The object path along the route

This subclass represents an object that moves along the schematic path. Multiple ways exist to achieve that goal. The web platform's example consists of 2 main objects: a *line* from the origin to destination and a *point* that animates along the route. A point is shown as an airplane icon that travels from *point A* to *point B*. Then, an arc path is drawn to imitate the path of the airplane. Moreover, some extra details were added to improve user experience and make the map more informative: *Replay* button which restarts the airplane path, and some information about the sources and the destinations: the names of the airports and their short comparative characteristics (Figure 4.15).

Class 4. Complex stories

Figure 4.16

Complex stories. Interactive mode

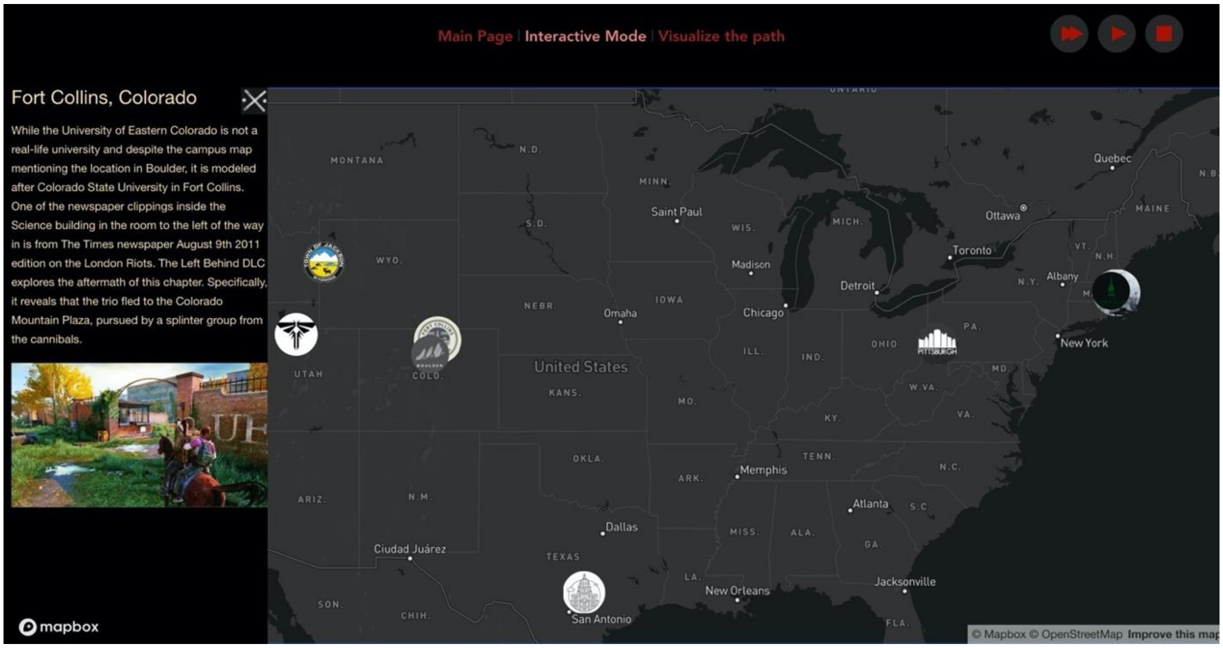
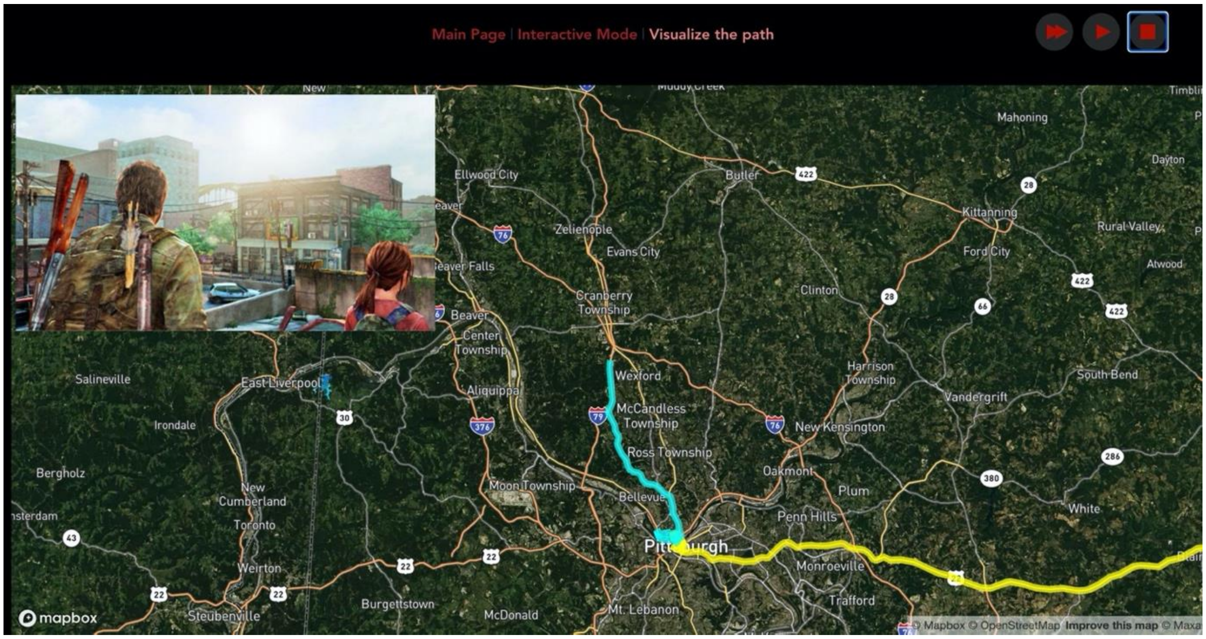


Figure 4.17.

Complex stories. Path Visualization



A complex story is a separate class that combines different features of the previous classes which serve as a set of the components of this "superclass". A complex story is one of the most popular classes and the combinations of components included in this class can provide astonishing results. One example of the complex story is analyzed in the following paragraphs.

One web platform's example which leverages a complex interactive approach will be considered. The name of the example on the web platform is "Complex stories". The main page welcomes a user with music, and hover events of the menu buttons generate the sound. Thus, the story has an explicit audible component. In addition, a simple audio player lets the user control the music: switch to the next song, pause and play the current song. The main menu offers the user a choice to proceed with exploring *an interactive map* or watch *the path visualization*. *An interactive map* has clickable icons and a click event handler displays the description cards with text about the place a user clicked on (Figure 4.16). Clickable markers are actually Markers recognized in *Class 2. Interactive Dynamic maps - Subclass 2.8. Markers & Popups*. Path visualization is implemented in a cunning way (Figure 4.17). The base of the implementation is taken from *Class 3. Path Visualization – Subclass 3.2. Real-time route render.* The algorithm from that subclass is used to render the path. In addition, *Class 2. Interactive Dynamic maps - Subclass 2.1. Fly to a Location* algorithm is used to fly to a location when there are gaps in the rendered path. Thus, the Complex story absorbs many of the different subclasses of the current classification.

The images' carousel which accompanies the path visualization is also worth discussion. Those images are directly related to the path which overlaps with the images, providing insight about events on that path while the story evolves. Thus, a user remains occupied during the path visualization process: the user has the main path visualization, audible (music) and extra visual (images slider) activities, so overall this complex story provides an immersive user experience. Moreover, when the path visualization is completed, control of the zoom will be unlocked, and a user is able to explore the rendered path more thoroughly.

The complex stories often include multiple types of classes and subclasses of the classification. The analyzed complex story contains 3 of the subclasses, but it is possible to make much more sophisticated stories without any limitations.

Chapter 5. Classification evaluation and feedback

During the research and building the classification, the author collected reviews to evaluate the proposed classification. The targeted reviewers consisted of web developers, user experience designers, and journalists who specifically worked with web cartography and specialized in interactive storytelling maps.

The main questions asked to the reviewers were as follows:

1) What criteria do you find the most important for appropriate classification?

2) Would you like to have a classification for storytelling maps? Would you use this classification or whom would you recommend using it?

3) What types of storytelling maps can be added to the classification or removed from the classification, or the classification is complete in your opinion?

The answers:

1) The overall perception about the proposed classification is its dependence on the map functions and behavior. Another approach to classifying the storytelling maps could be the utilization of the degree of interactivity as a basic criterion to reveal the classes of the maps. In that case, the following questions could be asked for each of the examples in the classification: Does the map itself need to be interactive? Will it make sense to have the map visible throughout the story or is it better if some maps include non-map content? How comfortable the mapmaker is with front-end development (e.g., does a mapmaker needs a low-code template or is it better to work on custom code)?

One of the ways to evaluate the classification is to ask the following question related to different kinds of maps: ‘Does this map can be related to one class or another?’. The classification can be satisfactory if all kinds of maps are specifically related to one of the classes. Besides, the decision to fit the map into the specific class or subclass must be obvious for the mapmaker in most cases.

2) The reviewers have agreed that the classification could be useful for people who are new to interactive storytelling maps and for those who want to orient themselves to various options.

When customers ask a mapmaker to create an interactive storytelling map, they can point out the name of the class or subclass referring to the classification. Thus, the classification can be helpful for mapmakers and for customers who can check the existing storytelling map options and indicate the necessary ones in the technical or business specification.

3) Most people recommended making more complex examples of the on-scroll option. Another important note is the utilization of code examples in *‘Class 2 - Interactive dynamic maps*’ within a scroll-through story. These code examples serve as snippets to be compiled into any number of storytelling experiences. For on-scroll (or scroll-through) stories the examples could be highly sophisticated, and often it is the favorite and most frequently used class of most reviewers.

Chapter 6. Conclusion

This work introduces the first practice-oriented storytelling map classification which is designed to be a reference for people who develop storytelling maps. The classification is based on map functionality and behavior. It consists of 3 major classes, 11 subclasses, and 1 “superclass”. The classification covers all basic components of the storytelling maps which can be compiled into complex stories or used on their own after necessary customization.

The author encountered the following major challenges. Firstly, developers tools and different ways to build storytelling maps had to be considered to make the classification comprehensive enough to cover all existing types of interactive storytelling maps. Secondly, the classes of maps had to be sufficiently scalable to include the types of maps that are able to appear in the future to avoid any major amendments.

This work contributes to the field of knowledge at the confluence of Geography, Software Engineering, Data Journalism, and Data Visualization. The main goal of this master’s thesis was to classify interactive storytelling maps according to the current technologies and demands. Each class and subclass were accompanied by a guideline and supplemented by at least one example made by using a modern JavaScript cartographic library: Mapboxgl.js. “Builders” of the storytelling maps possess the skills to pick up the most suitable class to fit the story in. Thus, the classification is highly practice oriented. The findings of this research can be helpful to journalists, geographers, web developers, and user experience designers.

References

Cartwright, W. (2019). Emotion maps. *Abstracts of the ICA* 1:1-2. July 2019.

Caquard, S., Cartwright, W. (2014). Narrative Cartography: From Mapping Stories to the Narrative of Maps and Mapping. May 2014. *Cartographic Journal The*. 51(2), 101-106 DOI: 10.1179/0008704114Z.000000000130

Caquard, S. & Griffin, A. (2019). Mapping Emotional Cartography. *Cartographic Perspectives*, (91), 4–16. https://doi.org/10.14714/CP91.

Campbell, C., Egbert, E. (1990). Animated cartography: 30 years of scratching the surface. *Cartographica*, 27(2), 24-43.

Crowe, J. (2017, September 26). Russell Kirkpatrick on Fantasy Maps. *Maproomblog*. https://www.maproomblog.com/2017/09/russell-kirkpatrick-on-fantasy-maps/

Dix A., Ellis, G. (1998). Starting simple: Adding value to static visualisation through simple interaction. In Proc. *Working Conference on Advanced Visual Interfaces, ACM*, 124–134.

Dodge, M., Mcderby, M., Turner, M. (2008). Book: The power of geographical visualizations. In Geographic Visualization: Concepts, tools, and applications. West Sussex, England, 2008, 1–10. https://doi.org/10.1002/9780470987643.ch1

Dou, W., Ziemkiewicz C., Harrison, L., Jeong, D. H., Ryan, R., Ribarsky, W., Wang, X., And Chang, R. (2010). Comparing different levels of interaction constraints for deriving visual problem isomorphs. *Visual Analytics Science & Technology*, IEEE, 1–8.

Earthquakes database (n.d.). *Earthquakes, United States Geological Survey*. Retrieved February 15, 2021, from https://earthquake.usgs.gov/

Getty Images (n.d.). *WWII, Animation of Germany's plan and invasion of Warsaw through Poland, encircling the city and cutting it off, forcing the surrender of most of the Polish.* Retrieved February 21, 2021, from https://www.gettyimages.com/detail/video/news-footage/898851242

Harrower, M. (2008) The golden age of cartography is now. http:www.axismaps.com/ blog/2008/10/the-golden-age-of-cartography-is-now/.

Iturrioz, T. & Wachowicz, M.. (2010). An Artistic Perspective for Affective Cartography. *Mapping Different Geographies*, 74-92.

Kosara, R., & Mackinlay, J. (2013). Storytelling: The next step for visualization. *Computer*, 46(5), 44-50.

Klettner, S.& Huang, H., Schmidt, M. (2011). EmoMap—considering emotional responses to space for enhancing LBS. *Advances in location-based services, 8th international symposium on location-based services, Vienna.* 01/2011.

MacEachren, A. (1995). How maps work. *The Guilford Press,* New York, NY, USA.

MacEachren, A. (1998). Cartography, GIS and the World Wide Web. *Progress in Human Geography*. 22(4), 575–585.

Mapbox Studio (n.d.). *Creation and management of the datasets with coordinates.* Retrieved February 11, 2021, from https://studio.mapbox.com/datasets/

Mar Tech Series (2017, Dec 22). 3 Reasons Why Data Storytelling Will Be A Top Marketing Trend of 2018. Retrieved February 21, 2021 from https://martechseries.com/mts-insights/guest-authors/3-reasons-data-storytelling-will-top-marketing-trend-2018/

Margo, E. B., Jeffrey, D. H., Geraldm R. W. (2018). Digital Story Mapping to Advance Educational Atlas Design and Enable Student Engagement. *ISPRS Int. J. Geo-Inf.*, 7(3), 125.

Monmonier, M. How to Lie with Maps. (1996). University of Chicago Press, Chicago, IL.

Netflix (2020). *Witcher, Map of the Continent.* Retrieved February 21, 2021, from https://www.witchernetflix.com/

Npm Trends (n.d.). *The comparison of package download counts over time*. *Npm packages: Mapbox-gl, Leaflet, OpenLayers (2020-2021)*. Retrieved March 29, 2021, from https://www.npmtrends.com/

Peterson, M.P. (1993). Interactive Cartographic Animation. *Cartography and Geographic Information Systems*, Vol. 20, No.1, 40-44.

Relive (2020). This is Relive [Video]. Youtube. https://www.youtube.com/watch?v=YZ3If0CTZY8

Relive (2021). *Relive official website.* Retrieved March 30, 2021, from https://www.relive.cc/

Roberts, L. (2012). Cinematic Cartography: projecting Place Through Film, In Roberts, L. (ed.), *Mapping Cultures: Place, Practice, Performance*. Basingstoke, Palgrave Macmillan, 68–84.

Roberts, L. (2014). The Bulger Case: A Spatial Story, *The Cartographic Journal*. 51:2, 141-151, DOI: 10.1179/1743277413Y.0000000075

Roberts, L. [Liminoid1]. (2010, Jun 19). The Bulger Case: a Spatial Story [Video]. Youtube. https://www.youtube.com/watch?v=0eDN2dTcLC0&t

Roth, R.E. (2013). Interactive maps: What we know and what we need to know. *J. Spat. Inf. Sci. 6*, 59–115, Google Scholar.

Roth, R.E. (2015) “Interactivity and Cartography: A Contemporary Perspective on User Interface and User Experience Design from Geospatial Professionals” *Cartographica* 50 (2), 94–115.

Roth, R. E. (2016). Cartographic Design as Visual Storytelling. *2016 Annual Meeting of the American Association of Geographers.*

Roth, R.E. (2020). Cartographic Design as Visual Storytelling: Synthesis and Review of Map-Based Narratives, Genres, and Tropes. *The Cartographic Journal.*

Segel, E. and Heer, J. (2010) “Narrative Visualization: Telling Stories with Data” *IEEE Transactions on Visualization and Computer Graphics* 16 (6), 1139–1148.

Slocum, T.A., McMaster R.B., Kessler F.C., Howard H.H. (2005). Thematic Cartography and Geographic Visualization. Second Edition. Upper Saddle River, NJ: Prentice Hall.

Song, Z. (2017) Map-based visual storytelling : an assessment of emerging genres and tropes. Master Thesis, *M.S. University of Wisconsin--Madison 2017*, 3-25.

Thrower N. (1959) Animated cartography, *The Professional Geographer*, 11:6, 9-12.

Tobler, W. (1970) A computer movie simulating urban growth in the Detroit region. *Economic Geography*. 46(2), 234-245.

Wallace, T.R. (2016) “Cartographic Journalism: Situating Modern News Mapping in a History of Map-User Interaction” (*PhD thesis*) University of Wisconsin−Madison.

Wood, D. (1992). The power of maps. The Guilford Press, New York, NY, USA.