

MapStory: Prototyping Editable Map Animations with LLM Agents

Aditya Gunturu
University of Calgary
Canada
CU Boulder
United States
aditya.gunturu@ucalgary.ca

Ben Pearman
University of Calgary
Canada
ben.pearman@ucalgary.ca

Keiichi Ihara
CU Boulder
United States
keiichi.ihara@colorado.edu

Morteza Faraji
University of Calgary
Canada
morteza.faraji@ucalgary.ca

Bryan Wang
Adobe Research
United States
bryanw@adobe.com

Rubaiat Habib Kazi
Adobe Research
United States
rhabib@adobe.com

Ryo Suzuki
CU Boulder
United States
ryo.suzuki@colorado.edu

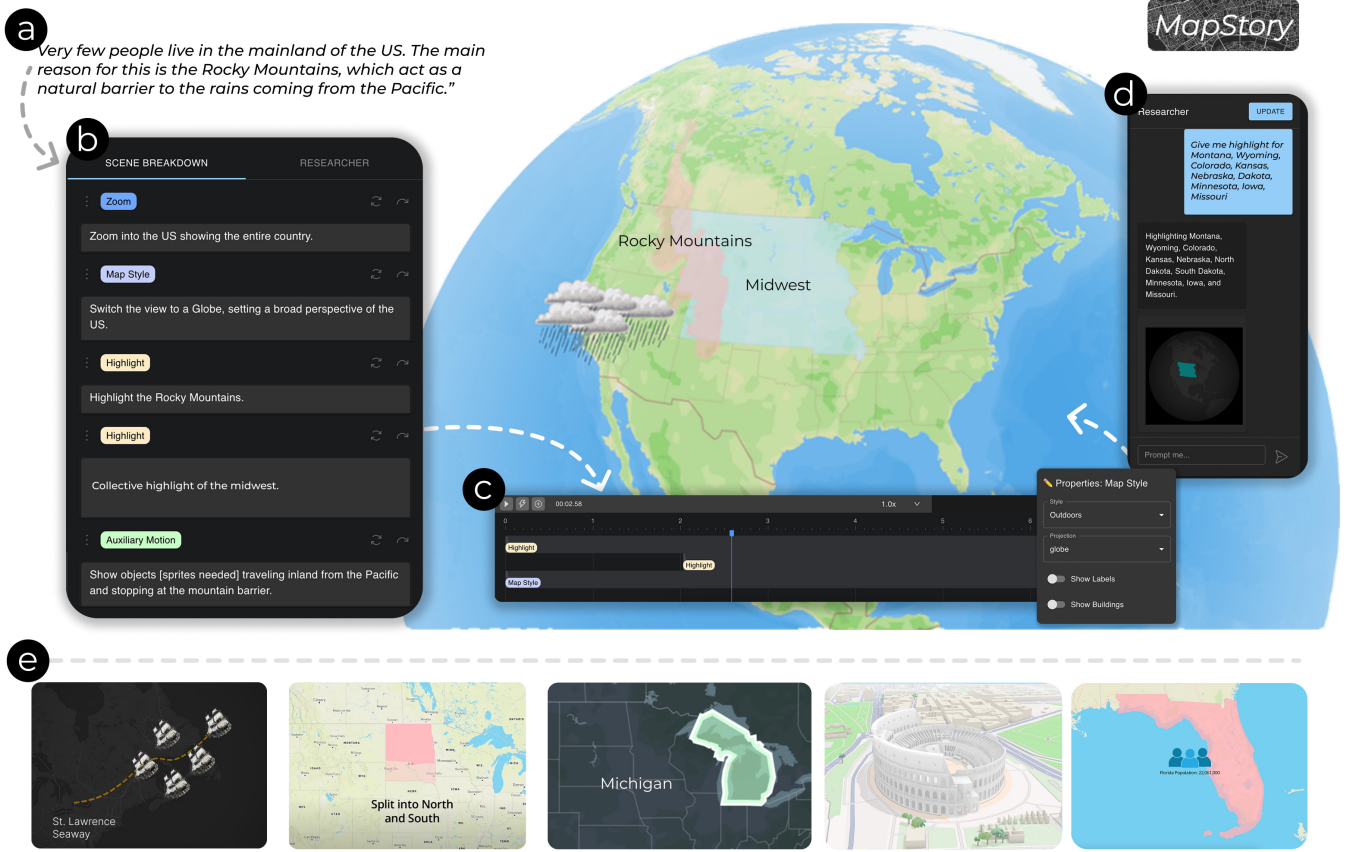


Figure 1: MapStory is an LLM-powered tool for prototyping map-centric animations from natural language scripts. Users write and edit a script (a), which generates an editable scene breakdown (b). The animation is shown on a timeline with a properties panel (c), and a researcher module retrieves and updates geospatial data (d). We showcase various map animation examples (e).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or

republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
UIST '25, September 28-October 1, 2025, Busan, Republic of Korea
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-2037-6/2025/09...\$15.00
<https://doi.org/10.1145/3746059.3747664>

ABSTRACT

We introduce MapStory, an LLM-powered animation prototyping tool that generates editable map animation sequences directly from natural language text by leveraging a dual-agent LLM architecture. Given a user-written script, MapStory automatically produces a scene breakdown, which decomposes the text into key map animation primitives such as camera movements, visual highlights, and animated elements. Our system includes a researcher agent that accurately queries geospatial information by leveraging an LLM with web search, enabling automatic extraction of relevant regions, paths, and coordinates while allowing users to edit and query for changes or additional information to refine the results. Additionally, users can fine-tune parameters of these primitive blocks through an interactive timeline editor. We detail the system's design and architecture, informed by formative interviews with professional animators and by an analysis of 200 existing map animation videos. Our evaluation, which includes expert interviews (N=5), and a usability study (N=12), demonstrates that MapStory enables users to create map animations with ease, facilitates faster iteration, encourages creative exploration, and lowers barriers to creating map-centric stories.

CCS CONCEPTS

• **Human-centered computing** → **User interface design**.

KEYWORDS

text-to-animation, map-based storytelling, LLM-based authoring tools, AI-assisted animation, human-AI collaboration

ACM Reference Format:

Aditya Gunturu, Ben Pearman, Keiichi Ihara, Morteza Faraji, Bryan Wang, Rubaiat Habib Kazi, and Ryo Suzuki. 2025. MapStory: Prototyping Editable Map Animations with LLM Agents. In *The 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*, September 28–October 1, 2025, Busan, Republic of Korea. ACM, New York, NY, USA, 19 pages. <https://doi.org/10.1145/3746059.3747664>

1 INTRODUCTION

Maps have long been a powerful medium for storytelling [71]. Since ancient and medieval times, people have used maps to tell stories [62, 63], such as the Catalan Atlas, which depicted medieval trade routes [2], or Charles Minard's visualization of Napoleon's campaign, considered one of the earliest examples of map-based data storytelling [3]. Today, people bring this form of narrative to life through animated storytelling, which we call **map animation**. This map animation is becoming increasingly popular, as it can effectively communicate historical trajectories [23], migration patterns [38], or geopolitical dynamics [31], often attracting millions of viewers as engaging educational content.

However, creating map animations remains significantly challenging and time-consuming. Producing even a short animation clip requires substantial efforts and expertise with animation tools. In addition, creators must conduct extensive research to ensure factual accuracy throughout the process. As a result, the creation of map animations remains largely inaccessible to non-experts, and even professional animators often spend days or weeks producing a single video, as we observed in our formative study. While

not all users may seek to create animated maps, existing apps like TravelAnimator[1] and many popular YouTube tutorials available online [46, 64, 65] indicate clear novice interest in editable map animation creation.

In this paper, we ask: *what if anyone could immediately create these map-based animations simply by writing a script?* And going beyond that to make them editable at every stage. For example, imagine typing a sentence like “*Very few people live in the mainland of the US. The main reason for this is the Rocky Mountains, which act as a natural barrier to the rains coming from the Pacific*” and instantly generating an animated sequence that visualizes this explanation and is editable at the script level as well as granular animation level (Figure 1). To explore this vision, we present **MapStory**¹, a system that generates editable map animations from natural language text following a human-in-the loop approach throughout the process. Unlike general-purpose text-to-video models like Sora [11] or Veo [4], MapStory is designed to preserve creative control and enable real-time fine-grained editing through a *scene breakdown*, a *conversational researcher*, and *primitive map animation blocks* such as camera motion, location highlights, and animated elements presented in an editable timeline interface. Our goal is not to replace traditional professional tools like After Effects and their low-level functions. Instead, we provide a *prototyping environment* that lets animators quickly draft map animations from text and refine them through rapid iteration with parametric control, enabling faster exploration and ideation.

MapStory is powered by an agentic architecture consisting of two LLM-based agents: 1) **a scene breakdown agent**, which parses the script into map animation primitives, and 2) **a researcher agent**, which grounds the script in factual geospatial data by identifying real-world regions, paths, and coordinates. For instance, in the previous Rocky Mountains example (Figure 1a), the scene breakdown agent identifies zooming actions for the US, highlights for the Rockies and Midwest regions, and animated elements like moving clouds (Figure 1b). Furthermore, the researcher agent automatically fetches polygons to highlight these regions without requiring manual specification; the user may then provide additional modification queries (Figure 1d). These animations can be easily modified or fine-tuned through an interactive timeline interface (Figure 1c). We leverage the geocoding ability of LLMs [43, 50] to generate real-world Cartesian coordinates and implement our architecture based on OpenAI's *o1* model for the scene breakdown agent and Perplexity's *sonar-pro* model for the researcher agent with web-search capability.

Our design was informed by two formative studies. First, we identified *animation primitives* commonly used in map animations (Figure 1e and 4) through an analysis of 200 publicly available map animation videos on YouTube. These primitive animations fall into three categories: 1) **camera movements**, such as *zoom*, *translate*, and *orbit*; 2) **visual highlights**, such as *area*, *path*, and *point highlights*; and 3) **animated elements**, such as *routes*, *spatial transitions*, and *auxiliary motion*. Second, we interviewed professional animators to understand their workflows and challenges, informing our three key design goals: 1) script-driven authoring, 2) research

¹Project Page: <https://adigunturu.github.io/MapStory-UIST25/>



Figure 2: Popular online animated map videos explain knowledge to a general audience in a highly engaging manner, influencing millions of viewers (See list of image sources for credits and links).

integration, and 3) modular map animation blocks for creative controllability. These findings guided the design and development of the MapStory interface and system architecture.

To evaluate our system, we conducted three studies: 1) *a usability study* (N=12) to measure the tool’s expressiveness, creative exploration, and accessibility for novice users; 2) *an expert study* with professional map animators (N=5) to gather feedback on the tool’s workflows, and potential for real-world applicability; and 3) *a technical evaluation* on 20 randomly sampled prompts to assess the generalizability and factual correctness of automatically generated animations. Our findings suggest that MapStory enables faster iteration, encourages creative exploration, and lowers barriers to creating map-centric animations for storytelling. Participants appreciated the effective balance between automation and control, and experts recognized its potential as a creative support tool with a focus on rapid prototyping. On the other hand, we also identified several limitations, including challenges with AI hallucination affecting factual accuracy the need to verify produced visualizations.

Finally, our contributions are as follows:

- (1) MapStory, a text-driven animation prototyping tool powered by an LLM-based agent architecture, composed of scene breakdown and researcher agents, that enables automated yet human-in-the-loop creation of map animations.
- (2) Insights from two formative studies: analyzing 200 map animation videos and interviewing expert animators, that guided our identification of essential primitive map animation blocks and script-driven design principles.
- (3) Results from three evaluations: technical, usability, and expert feedback, that assess the effectiveness of our tool and highlight opportunities and challenges in AI-assisted map animation authoring.

2 RELATED WORK

2.1 Map-Based Visualization and Storytelling

Map-based storytelling systems have leveraged interactive visualizations, especially geographic maps, to contextualize narratives. For example, NewsViews [19] and GeoExplainer [34] automatically generate annotated thematic maps by mining text for locations and linking them to relevant data. Lundblad et al. [41] present a web-based geovisual analytics toolkit that integrates dynamic visual analysis with interactive storytelling. These systems lower the barrier to creating map-based storytelling for understanding statistical data and conducting spatial analyses. Narrative Maps [30]

introduced a “route map” metaphor, depicting events in a story as landmarks connected by routes on a conceptual map. Prior work has extensively explored map-based visualizations for geospatial data. For example, data visualizations rendered on maps or 3D globes to convey global trends [53, 54]. Other work integrates narratives with physical space: Location-Aware Adaptation [35], Story-Driven [8], Believable Environments [22], and How Space is Told [56] present an approach to generate location-based stories, automatically assigning story events to contextually suitable locations. Additionally, GeoCamera [36] introduced an authoring tool that supports users in designing camera movements for storytelling with geographic visualizations. MapStory builds on these threads and extends prior map visualization and narrative authoring systems by leveraging map animation generation through script, allowing authors to easily produce and revise rich map-based stories.

2.2 Animation Authoring Tools

Creating animations traditionally requires significant time, effort, and technical skill, so a number of HCI systems have explored ways to lower this barrier. Early approaches introduced sketch-based animation tools like K-Sketch [16], Draco [28], and Kitty [27]. For example, Draco [28] let illustrators bring static drawings to life by sketching motion paths and applying kinetic textures to create rich path animations and particle effects. These sketch-based interfaces greatly lowered the barrier through more natural interactions. Augmented Physics [21] introduced a CV-based pipeline to convert static physics diagrams into animated physics simulations, providing a AI-assisted, selection oriented tool for creating animated physics visuals. Another line of work provides higher-level building blocks and templates to simplify animation authoring. Motion Amplifiers [29] introduced a set of reusable animation primitives based on the principles of traditional animation. Similarly, Ma et al. [42] extended this idea to 3D animations with a layered authoring interface. This layered design balanced ease-of-use with expressiveness, showing how modular animation blocks and multi-level interfaces can support both novices and professionals. Other systems have explored performance-based and data-driven templates: for example, Moscovich et al. [45] enabled recording motions via hand gestures. More recently, researchers have explored flowchart-based authoring [13, 76]. For instance, FlowMatic [76] introduced an authoring tool that uses flowcharts to create reactive behaviors of virtual objects.

2.3 LLM-Powered Authoring Tools

The emergence of generative AI and large language models (LLMs) has further inspired AI-assisted authoring tools. LLMs have shown promise in supporting creative work like story generation [15]. In particular, recent works have explored text-to-video generation, demonstrated through numerous commercially-available tools, including Sora [48], Veo2 [20], Runway [52], and Pika [49]. These approaches basically generate videos from prompts, but they usually lack iteration and fine-tuning.

In contrast, the HCI community has explored more interactive and user-in-the-loop approaches. Several systems introduce visual authoring techniques, such as graph-based manipulation [7, 26, 74], drag-and-drop interfaces [9, 44], and multi-modal prompt refinement [69], to allow users to edit and steer the generation process. Several notable systems have applied these approaches to animation generation. One example is Katika [25], an end-to-end system for creating explainer-style motion graphics videos from a natural language script. Alternatively, Spellburst [6] uses an LLM to produce p5.js code snippets to generate creative web animations. Similarly, Keyframer [61] explored using GPT-3 to synthesize CSS keyframe animations from design descriptions. LogoMotion [40] is a recent system that helps novices animate static logo graphics through visually-grounded code synthesis. Inspired by these prior works, our system also leverages LLMs not to directly output the video content, but instead generate animation components by leveraging LLM-agent architecture for map-based animation generation.

2.4 Natural Language for Creative Tasks

Natural language has been extensively explored as a medium for authoring, editing, and organizing content across various creative domains. Prior work has shown its effectiveness in video editing and storytelling with text input. For instance, CrossPower [72] leverages script-like natural language to organize visuals in videos, while DataParticles [12] supports language-oriented authoring of unit visualizations.

Natural language is not limited to text input; recent systems explore speech as a modality for real-time creative control. TakeToons [58] leverages structured scripts and the actor's facial poses to translate talking animations to a virtual character in real-time. DrawTalking [51], on the other hand, enables users to add simple motion to sketched objects via speech, and RealityTalk [37] displays relevant graphics based on the user's speech for creating augmented presentations.

Recently, many works have focused on editing and organizing video content via natural language using LLMs. Works such as LAVE [66] and ExpressEdit [59] have explored story level editing video through free-form text. Other systems, such as ChunkyEdit [33], QuickCut [60], and B-Script [24] have explored natural language prompting as a way to make edits, such as trimming or organizing video clips using video and organizing transcripts.

Natural language has also been used for map-based content creation. For instance, Embark [57] has examined parsing natural language text outlines into plans with routes rendered on maps, effectively turning notes with a structured schema into dynamic documents. Meanwhile, tools like Eviza [55] support natural language analytical questions and render the resulting visualization

situated on a map. Finally, CrossTalk [73] leverages conversations in online meetings to organize, recommend, and navigate places on shared maps. Script-driven editing tools for video have proven to be promising in allowing creators to have high-level control over their editing, organization, and overall storytelling [32, 33, 60]. Inspired by these ideas, our system supports creating parameterized animations from the script, which are organizable, controllable, and editable at a high level for conveying map-centric stories.

Recent work in HCI has seen the emergence of LLM-driven editing tools such as LAVE [66] and MoGraphGPT [75], which respectively leverage linguistic augmentation for video editing and modular LLMs for interactive scene creation. In contrast to these approaches, our architecture is novel in its explicit focus on map-centric storytelling: it employs a dedicated *planner-researcher* agent that translates free-form user instructions into GeoJSON operations, including querying, addition, reduction, and generation, while being promptable at every step of the architecture. The user can request changes to the scene breakdown as well as the researcher agent. This modular design not only decouples complex geographic reasoning from traditional video or scene editing pipelines but also provides enhanced transparency and user control by allowing real-time refinement of map animations.

3 FORMATIVE STUDY

To better understand current practices and challenges in creating map animations, we conducted a formative study. Our method is informed by prior work such as k-Sketch [16] and Draco [28], which demonstrated that novices benefit significantly from learning and adopting expert-derived strategies. Moreover, cognitive apprenticeship theory [18] shows that novices perform best by watching and trying expert strategies. The goal of this study was to gather insights that would guide the design of MapStory, grounded in the real-world experiences, needs, and challenges faced by professional map animators.

3.1 Method

We recruited three professional map video animators (3 male, ages 21–28). Each participant had at least three years of experience (3–6 years) producing map animation videos, which they regularly upload to YouTube. One participant does client work. We contacted them via emails, Youtube, and Instagram DMs. We conducted semi-structured interviews of approximately 1 hour over Zoom, and compensated each with 15 USD.

MapStory was intentionally designed using insights from expert animators, following a cognitive apprenticeship model [18] to scaffold novice interactions effectively. This design choice aligns with successful prior systems such as K-Sketch [16] and Draco [28], which demonstrated that novices benefit significantly from learning and adopting expert-derived strategies.

During the interviews, we explored the experts' typical workflows, the tools they use, and the challenges or needs they encounter when producing map animations. Although we acknowledge the relatively small number of experts in our study, we intentionally focused on specialized map animators rather than general video creators, thus recruitment was challenging. To complement our

interviews and mitigate this limitation, we also analyzed a well-known tutorial on creating map animations, titled “How I Make My Maps”². We also analyzed popular creator’s Patreon tutorials³. Our findings, drawn from both the expert interviews and the tutorial analysis, are described in the following sections.

3.2 Insights and Findings

Current Tools, Practices, and Workflows. In our study, all experts use Adobe After Effects for video editing. For map-specific graphics, they use an After Effects plugin called GeoLayers [5], which offers features such as importing real-world map data, customizing map styles, and integrating location-based metadata. In case of area highlights, they use GeoJSON⁴ polygons to render the regions on the maps. Despite these capabilities, the experts noted that complex animations still demand considerable manual scripting and keyframing. As a result, they reported that creating a single map animation typically takes from a few days to weeks, with some projects extending beyond. Moreover, the experts emphasized that After Effects usage is only one component of their overall process. Based on their comments and the tutorial, a typical map animation project generally involves four stages: 1) **script phase**, where the idea and narrative are determined; 2) **research phase**, for gathering factual information and extracting data; 3) **animation prototyping phase**, where the script is translated into a rough animation; and 4) **iteration phase**, during which the animation is refined for publication.

It All Starts with a Script. Participants emphasized that the animation process always begins with a script that outlines the core narrative, pacing, and visual milestones (Figure 3). Before working in After Effects, they typically create a written plan specifying what content must appear on the map and when, in sync with the video’s narration. As P1 explained, “I generally start with an idea [...] then I’ll start writing a script, and it usually takes longer than you’d think”. The script also captures instructions for other visual elements, such as layering different geographic datasets, adding text callouts, or highlighting resource-rich regions. P1 added, “Sometimes, I’ll put directions like ‘zoom in here’ or ‘highlight this region’ in my script”. Once the script is complete, they conduct a “script breakdown” by tagging specific parts of the text with directional notes—such as zooming in or toggling boundaries—to convert the written plan into explicit animation steps. As P2 described, “When I started, I did some markup on my script like ‘move the camera here’ or ‘highlight that region next’ so I know exactly how to animate it in After Effects”. Some participants also described using large language models to outline their animation sequences well before launching any software. As P2 explained, “I give ChatGPT the script and ask it to generate a plan [...] It’s a huge time saver for planning my camera moves before I open After Effects.”

Decoupled Research and Animation Process. After writing the script, creators often go through a dedicated research phase to identify accurate routes, verify historical or geographic facts, and gather supporting data. However, this research occurs outside the

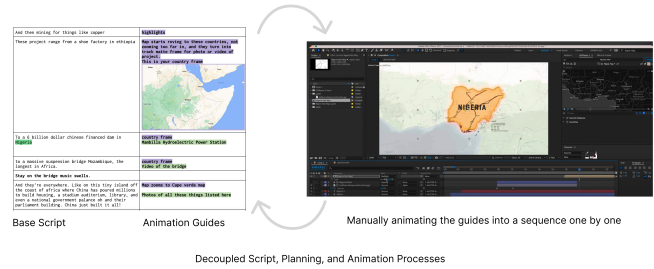


Figure 3: Workflow of a map animator. The animator first breaks down their script into manageable chunks and describes the animation they plan to make for this specific item. The animation guides also include research information like population counts, medals won by a specific country or region boundary maps. ©Patreon - Johnny Harris

animation tools, so animators must manually translate their findings into keyframes, such as plotting accurate paths or highlighting specific regions. As P2 noted, “I do my fact-checking in Google or ChatGPT, but then I have to transfer everything by hand into After Effects. If some information like routes, highlights, or stats changes last minute, I need to rework the whole scene.” This lack of real-time integration between research, planning, and animation leads to frequent misalignments, increased chances for errors, and slower iteration cycles. Ultimately, these fragmented workflows hinder the creative process. P1 echoed similar challenges, noting how “I do a lot of the research with Google, but then I basically integrate it all (borders, stats, routes, facts etc.) out by hand. If I discover something’s slightly off, I have to go back and fix every keyframe.” He also pointed out that “I’ll sometimes use ChatGPT to refine my script, but there’s no direct way to update the animation when I find new facts—I still have to do everything manually.” This repeated back-and-forth between gathering accurate data and implementing it in the animation often results in “days” of extra work whenever details change.

Need for Iteration and Creative Controllability. When prototyping their animations, experts emphasized the importance of an iterative process. As P1 explained, “Being able to quickly tweak an animation and immediately see how it aligns with the narrative is crucial”. Therefore, rather than relying on a single, monolithic timeline, creators prefer to assemble individual building blocks or primitives, where each block is responsible for a particular sequence or visual effect. For instance, one block might highlight a geographic boundary, while another animates a specific route, which are related to the tags in the script. Since these modules are self-contained, changes in one block do not disrupt the rest of the animation. P2 added, “We can’t just fix it on the fly in the same place, so it takes forever if the story shifts suddenly”. Similarly, P3 said “When clients request changes last moment, it can be very frustrating. I try to clarify requirements upfront, but clients sometimes change their requests in the middle of the process,” underscoring how sudden updates force rework, which takes a lot of time. Overall, professional creators prefer granular-level adjustments and creative controllability, rather than generating the entire video in a single pass.

²https://www.youtube.com/watch?v=GsojLuJpe_0

³<https://www.patreon.com/posts/behind-scenes-51811597>

⁴<https://geojson.org/>

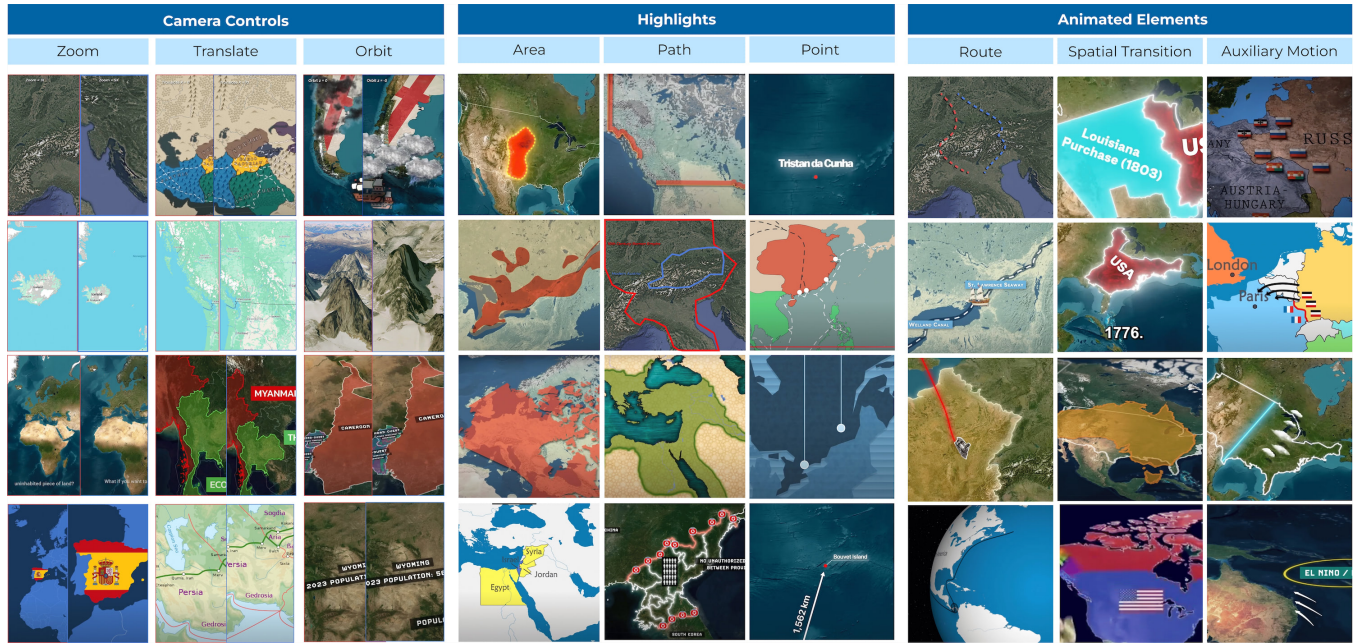


Figure 4: Common map augmentation and animation techniques based on content analysis of 200 map-centric videos (See list of image sources for credits and links).

3.3 Design Implications

Based on these findings, we made the following design decisions.

D1: Script-Driven Authoring. Our system should directly translate scripts to support a script-driven workflow. Moreover, changes to textual instructions should seamlessly reflect in the animated map scenes, enabling rapid iteration.

D2: Research Integration to Extract Map Data. Our system should integrate research into the animation tool, allowing users to gather, verify, and organize geospatial data and story elements in a single environment. These elements should be directly embedded into the scene to minimize manual plotting.

D3: Generating Editable Animation Building Blocks. Our system should generate animations not as one-off videos but as editable animation modules. These modules should allow creators to tweak individual parameters and elements without disrupting other segments, thereby supporting fine-tuned visual adjustments and rapid iteration.

4 COMMON MAP ANIMATION TECHNIQUES

To establish our map animation primitives, we began by identifying the most common techniques used in existing map animation videos. To this end, we conducted a formative analysis of map-based animations, aiming to pinpoint widely adopted methods. The findings from this study helped shape the system design of our primitive map animation blocks.

4.1 Method

We started from collecting popular map animation videos through YouTube search. Two authors (A1 and A3) began with keyword search, such as “map animation”, “animated maps”, and more context-specific terms like “history of [geographic location]”. Since there is no single dominant keyword to discover these map animation videos, we used YouTube’s recommendation system based on the initial subset of relevant videos to discover additional related videos. We included only videos that employed animation techniques using a map as the primary visual canvas. Static map images or videos without any animated transitions or effects were excluded. Through this process, we curated a total of 200 map animation videos.

Recognizing that each video might feature multiple animation techniques across various scenes, A1 and A3 extracted screenshots from each distinct animation scene, producing a total of 1200 screenshots. These screenshots were then categorized with thematic coding. A1 initiated the coding by organizing the screenshots on a Miro board. After this initial categorization, all authors reviewed the emerging themes and iteratively refined them until reaching a consensus.

4.2 Results of Map Animation Techniques

This analysis resulted in the identification of three primary categories of map animation techniques, which we implemented as core components in our system (Figure 4). A complete list of the analyzed videos, along with their corresponding visual overlays, is provided in the supplementary materials.

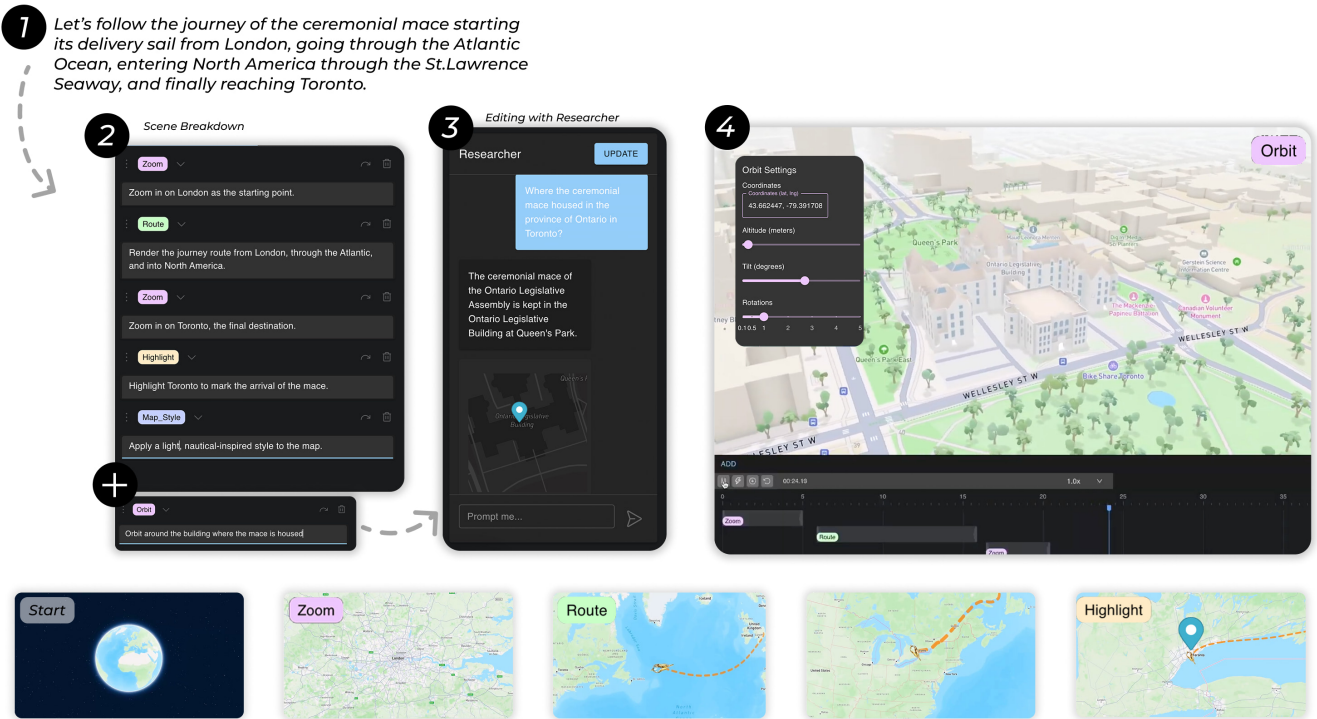


Figure 5: MapStory System Walkthrough.

Highlights. Highlights are a primary method of emphasizing specific geographic areas of interest on a map. They can be implemented as areas, lines, or points:

- **Area highlights** are used to visually emphasize regions or zones by shading, coloring, or bounding the chosen area. This approach effectively focuses the viewer’s attention on a specific region of the map—such as a country, state, or historical territory—by distinguishing it from its surroundings.
- **Line highlights** outline or trace boundaries and routes. For example, line highlights are often used to represent journeys like troop movements and migration paths. Also, it can show borders, national boundaries, or coordinate references to provide contextual grounding or to show expansion over time.
- **Point highlights** are small markers, symbols, or icons placed at particular coordinates. These highlights typically indicate important landmarks, cities, or data points, making it clear where the viewer should focus.

Camera Controls. Camera movement plays a critical role in map-based animation, guiding the viewer’s focus and creating dynamic transitions between scenes.

- **Zooming** is the most common camera motion, used to shift the level of detail by moving in and out of specific regions. It helps create narrative pacing and emphasizes points of interest.
- **Translation** involves shifting or panning the camera’s viewpoint across the map. This technique can guide the audience

from one geographic region to another, showcasing transitions between key locations or highlighting spatial relationships.

- **Orbiting** refers to rotating the camera around a focal point or region on the map. By orbiting, animators can present multiple sides or angles of a location, adding a three-dimensional feel and helping viewers grasp the scope or scale of the geography in question.

These camera movements are frequently orchestrated using keyframes, which define the camera’s position and orientation at specific moments in time, resulting in smooth and deliberate visual transitions.

Animated Elements. Animated elements bring dynamic motion to the map, helping convey movement, changes in territory, or shifting data.

- **Route motion** animates objects, such as ships, planes, and arrows, along paths to illustrate travel, trade routes, invasions, or other directional flows. This technique is especially effective for showing temporal progression or cause-and-effect across space.
- **Spatial transitions** involve the enlargement or reduction of highlighted regions or shapes over time. This technique is typically used to showcase how areas grow or shrink—such as expanding boundaries in historical conquests, shifting population zones, or changes in environmental conditions.
- **Auxiliary motion** includes any additional, supportive movement layered onto the map for emphasis or context. Examples include arrows sweeping across a region, clusters of icons like armies or clouds moving in a swarm-like fashion, or scanning

beams that pass over key areas. This motion is often employed to guide the viewer’s attention and reinforce important narrative points or data trends.

5 MAPSTORY

This section presents MapStory, an interactive tool that generates map-based animations from natural language input.

5.1 Overview

Built on large language models, MapStory provides the authoring interface that translates text into modular, editable map animations. MapStory takes a text script as input and outputs a sequence of structured animation components, offering flexibility and fine-grained control. MapStory decomposes the input into discrete animation steps by identifying high-level actions, such as zoom, highlight, and route animation, which are represented as editable JSON and later automatically converted into corresponding animations. These modules are visualized on a map canvas and can be interactively refined using a timeline editor and a properties panel. Users can edit the animation modules on a timeline, adjust parameters, and preview changes in real time. To support accurate and fact-grounded animations, users can ask follow up queries and modifications to the system’s initial results. MapStory assists animators in resolving vague or ambiguous location references, retrieving precise geospatial data like points, regions, or paths, based on the LLM-powered web search, that is directly integrated into the animation modules.

5.2 System Walkthrough

We now walk through an example of how a user can create an animated scene using MapStory, as illustrated in Figure 5.

Step 1: The User Types a Script into MapStory. The user begins by writing, in natural language, a description of the animation scene they wish to create. In Figure 5-1, the user provides the script: “Let’s follow the journey of the ceremonial mace starting its delivery sail from London, going through the Atlantic Ocean, entering North America through the St. Lawrence Seaway, and finally reaching Toronto.” The input can be as vague or as detailed as the user prefers. MapStory is designed to interpret a wide range of user intentions from open-ended descriptions.

Step 2: The System Breaks Down the Script into Animation Sequences. From the initial text input, the system’s *scene breakdown* module generates a step-by-step breakdown of the scene into modular map animation primitives. In this example, the system automatically generates the following sequence: “a zoom to London”, “a route animation across the Atlantic”, “a zoom to Toronto”, and “a highlight on Toronto”. Each step corresponds to a specific primitive animation block, which can be edited, reordered, added, or removed by the user (Figure 5-2). The user can further refine each block by writing contextual or descriptive text, which the system uses to better understand the animation intention.

Step 3: The User Interactively Retrieves Geospatial Map Data. Next, the system helps retrieve and process relevant geospatial data via the *researcher* module. This includes fetching GeoJSON coordinates for locations, areas, or routes mentioned in the script.

The system first automatically fetches the relevant information for each block in the scene breakdown. However, the user can follow up with additional queries or modifications. For example, when the user asks, “Where is the ceremonial mace housed in the province of Ontario in Toronto?”, the system identifies the Ontario Legislative Building as the destination and extracts its location for use in the animation (Figure 5-2). The system can also pull in additional metadata, such as historical context, the year a building was constructed, or statistics like the number of medals won by a country, allowing users to focus on their story.

Step 4: The User Modifies Parameters, Adjusts Styles, and Controls Timing of Each Animation Module. With the animation structure and map data in place, users can fine-tune the look and feel of each element using the properties panel. For example, they can add images, label annotations, or change the style to enhance the animation. Each module’s visual properties, such as color, transparency, labels, and styling, can be customized. For instance, a user might highlight a country in a vivid color, adjust the opacity of a route, or change the map style to match the narrative tone.

Finally, users can interactively control the timing of each animation module by specifying when it appears and how long it plays, through the timeline sequencer. This allows for fine-grained control over pacing and sequencing in real time. By iteratively refining timing and transitions, users can craft a cohesive and compelling animated story that aligns with their intended narrative. The final result is a text-driven animation that seamlessly integrates geospatial data, visual styles, and temporal coordination.

6 IMPLEMENTATION

Our system is implemented using React.js and TypeScript, composed of three main components: 1) a base map canvas built with Mapbox, 2) a custom timeline editor and animation sequencer, and 3) a text-driven scene breakdown and researcher interfaces powered by our large language model architecture. We use OpenAI’s *o1* as the primary multimodal language model to process user input and assist with animation generation. For geospatial queries, the system uses Perplexity’s *sonar-pro* model to retrieve and integrate geospatial data, which has access to the web. We additionally query GeoJSON data from OpenStreetMap via the Nominatim API⁵, enabling location lookup and metadata retrieval based on natural language queries. For map animation rendering, we use the Mapbox API within the React.js framework. The system can also dynamically change the map’s appearance by selecting from a variety of styles supported by Mapbox⁶.



Figure 6: Map Stylization

⁵<https://nominatim.openstreetmap.org/ui/search.html>

⁶<https://docs.mapbox.com/api/maps/styles/>

6.1 Animation Sequencer

The animation sequencer, which is visually represented as a timeline is implemented in React.js in the browser environment using Mapbox API as a base canvas. We use Fabric.js to render overlays on top of this canvas. The engine comprises of a time interval which runs every second when the user clicks the play button. Each map animation primitive in the system is represented as with a structured JSON block with a *block_name*, *start_time*, *end_time*, and *block_args*, where the block arguments are specific to each block type. These blocks are typically created by the LLM agent but can be manually added by the user as well. When the animation starts, the renderer periodically checks if the time interval matches the start time of any block. If it does, a block specific function is called to start the animation sequence of that block. These block specific functions animate and visualize objects on the Mapbox canvas implemented as a fabric.js overlay. The block arguments with start and end times are passed into the block function, which then performs the animation action rendered on screen.

6.2 Supported Primitive Map Animation Blocks

In this section, we describe how our system implements the core map animation primitives. For each primitive block, the choice of supported animation is informed by our taxonomy analysis, and each category corresponds to the descriptions provided in Section 4. These modules serve as the fundamental *animation primitives* in our system.

Highlights. The Highlight block supports three types of visual emphasis: *Area*, *Line*, and *Point*.

- **Area:** Renders a GeoJSON polygon on the map canvas at a specified time interval. The polygon data is fetched from the Nominatim API or generated by the LLM.
- **Line:** Uses an array of latitude-longitude coordinates provided by the LLM to render a polyline on the map.
- **Point:** Plots a map marker at the specified coordinates.

Users can freely attach text or images to highlights of any type, offering a flexible way to annotate or enhance the visual presentation of geographic features.

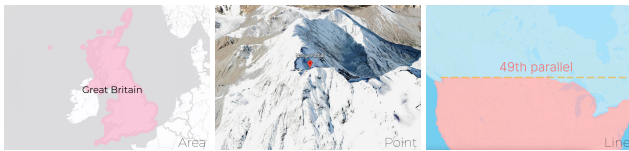


Figure 7: Our system supports three types of highlights 1) area, 2) point, and 3) line.

Camera Controls. Camera movements are handled by animating the viewpoint in Mapbox:

- **Zoom:** Employs the built-in FlyTo function to smoothly zoom into a specified location.
- **Translate:** Interpolates the camera’s target position from start to end coordinates, shifting the focal point without changing the zoom level.

- **Orbit:** Moves the camera along a circular path using incremental angular steps every frame, creating a revolving view of a given focal point.

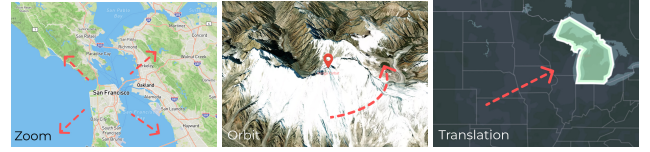


Figure 8: Camera controls: 1) zoom, 2) orbit, and 3) translation.

Animated Elements. To create dynamic motion atop the map canvas, our system employs a custom animator built using browser intervals and Fabric.js⁷. By defining sprite positions over time in latitude–longitude format, we can display animation frames that smoothly transition across the map.

- **Animated Routes:** Drawn via the Mapbox drawing API, routes are specified as an array of latitudes and longitudes, generated by the LLM (or modified by the animator). Once the path is rendered, user-uploaded sprites can be animated along that route. These sprites remain fully editable, and their position, style, or any other attribute can be adjusted.
- **Spatial Transitions:** Implemented as polygon shape transformations or translations. For example, to show a transition between the polygons for North Dakota and a merged North–South Dakota shape, we interpolate these two polygons over a specified time interval with Flubber⁸, creating a morphing animation. Simple translations move the polygon from one location to another.
- **Auxiliary Motion:** Provides secondary movements such as looping animations across a specified coordinate range on the map. Users can upload sprites, which the system duplicates into a cluster. The LLM determines the motion range and cluster count, enabling versatile and visually engaging secondary animations.

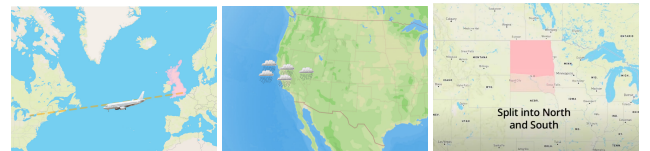


Figure 9: Animated elements: 1) routes, 2) auxiliary motion, and 3) spatial transitions.

6.3 LLM Architecture

MapStory’s agent architecture follows a human-in-the-loop approach to creating, verifying, and querying information for map animations. We chose to separate our architecture into an agentic workflow consisting of multiple calls as LLMs perform better when broken down into modular task due to token limitations, continuity

⁷<https://fabricjs.com/>

⁸<https://www.npmjs.com/package/flubber/v/0.1.0>

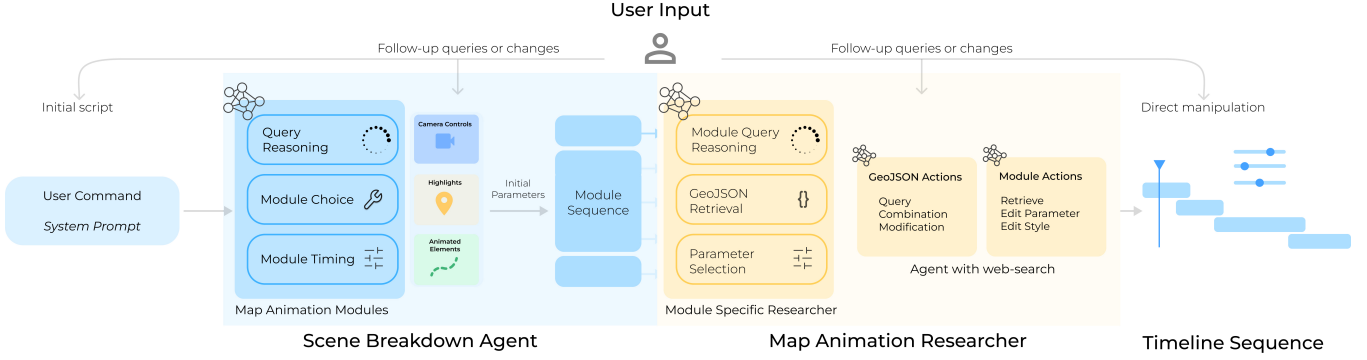


Figure 10: MapStory LLM Architecture

issues or clarity [67]. We chose the two main tasks of creating map animations as found in our formative explorations: an animation planning stage and a research stage. Our architecture features two primary LLM agents: a scene breakdown agent and a map animation researcher agent. We use the *o1* model for scene breakdown tasks, while Perplexity’s *sonar-pro* model serves as the research agent, helping with web-queried outputs. The scene breakdown agent is responsible for converting user input into a structured JSON-based scene layout with key animation blocks in sequence. The researcher agent then examines the relevant modules from that breakdown, performs chain-of-thought reasoning [70] to validate their parameters, and finally invokes the appropriate function call with the confirmed arguments. Below, we detail both agent roles.

Scene Breakdown Agent

The scene breakdown agent is designed to create step-by-step editable animation guides (**D1**), which follow an animator’s traditional approach of creating animation guides from their scripts. The scene breakdown agent is implemented as a single function call and can be re-prompted with additional queries by the user interactively. Given a user *script*, the agent returns a sequence of primitive map animations with each item containing a *module_name*, *timeline_start*, *timeline_end*, *short_description*, *long_description* and *initial_parameters* (module specific). We describe the steps the agent takes to achieve this below.

Query Reasoning. Given an initial script, the scene breakdown agent is first instructed to first perform query reasoning to infer the intent of the user and instructed to simplify the text into a list of requirements. The scene breakdown agent follows a planning approach where a task is divided into smaller manageable sub-tasks [68] while internally conducting CoT reasoning [70].

Module Choice. The agent is then instructed to deconstruct the initial free-form user query into actionable animation guides in plain text. The prompt includes descriptions of each animation block *<primitive_descriptions>* supported by our system within three categories: *highlight*, *camera control*, and *animated element* blocks. The prompt also includes the following heuristic consideration: Always precede highlight or animated element blocks with a *zoom* or *translate* blocks to set the focus on the area.

Module Timing. The agent is instructed to schedule each module in strictly consecutive, discrete, non-overlapping intervals with no gaps with the following heuristic considerations: 1) avoid overlapping camera control blocks; 2) time modules so that, at any given camera focus, all active modules operate within the same visible region; and 3) the mapstyle module must not overlap with any other module and, unless otherwise specified, should span the entire sequence.

Initial Parameter Tuning. For each animation primitive module, the agent is instructed to produce both a short and a long description for each block. The *short_description* is one sentence describing the module and its main parameter. For the *long_description*, the agent sets initial parameters for the animation block, such as the latitude and longitude of the location, the initial query for the Nominatim API, or an initial list of coordinates for a route. This provides additional and specific context for the researcher to perform reasoning, querying, and verification.

Map Animation Researcher Agent

Once the scene breakdown agent has drafted an initial set of animation guides and set initial parameters, the researcher agent parses through each item with its unique user-specified and system-generated context to find or generate appropriate GeoJSON and map-specific coordinates. The researcher is designed to reason about the user’s intent and extract the relevant information from the web or other sources (**D2**). We use Perplexity’s *sonar-pro* with web-search and reasoning capability to query, search, and fetch relevant GeoJSON data to satisfy the user’s specific request.

A dedicated research agent is assigned to each animation module. The module specific researcher agent validates initial parameters from the scene breakdown agent and each research agent’s system prompt is tailored to a specific animation block. Additionally, each research agent can only access one tool: a function call adhering to the OpenAI Function Calling Protocol [47]. This modular structure not only streamlines the system’s architecture but also minimizes context length, given the inverse relationship between context size and instruction-following accuracy [39].

We break down the agent prompt into the following parts:

Query Reasoning. Given a scene breakdown object with user query, long and short description, initial parameters, and overall

scene breakdown, the researcher module is first instructed to reason about user intent, disambiguating queries into map-compatible or queryable terms. For example, when asked to highlight “*Andhra Pradesh before 2014*,” the researcher is instructed to first verify if this is a Nominatim-compatible query. In this case, it is not, as the user is asking for region boundaries in the past. It disambiguates ambiguous terms and flags any request that cannot be satisfied by a direct Nominatim lookup (e.g., “*Andhra Pradesh before 2014*,” which requires historical rather than current boundaries). If not, the agent deconstructs the query into a list of n Nominatim compatible strings which best match the query description. This will later be fetched from the Nominatim call.

GeoJSON Retrieval. Once query reasoning is complete, the agent moves on to the GeoJSON retrieval step. We use the OpenStreetMap Nominatim API to query and retrieve various types of GeoJSONs, like polygons, multipolygons, lines, or points. The arguments taken by the API are a compatible *query_string* and optional arguments like *polygon_type*, *country*, *viewbox* etc. A full list of compatible parameters for the API can be found here ⁹ For example, the “*Avon River*” can be found in both the UK and Canada. We can construct a query to disambiguate between the two by replacing the bounding coordinates (*Example query with the UK as filter*). The researcher agent is then instructed to choose an appropriate *GeoJSON retrieval action*.

6.3.1 GeoJSON Actions. We define actions as a supplementary query to a specialized agent which performs a modify action on the fetched GeoJSON for a given query. Our system supports the following modifications on fetched GeoJSONs:

- **Query:** If a query is determined to simply be Nominatim-compatible, the agent constructs a compatible query and fetches the GeoJSON. For example, querying “*Andhra Pradesh*” results in a polygon of the state in India in the present day.
- **Combination:** The combination action performs basic set operations (*Union*, *Difference*, and *Intersection*) on the fetched GeoJSON polygons to form the desired boundary. The agent first returns a string array of all Nominatim-compatible queries that satisfy a request. The agent is instructed to select the operation that minimizes the number of Nominatim queries. For example, for “*Andhra Pradesh in 2014*,” it retrieves the boundaries of the constituent regions “*Telangana, India*,” “*Andhra Pradesh, India*” and applies a *union*. In another case, it may choose *intersection*; for instance, to fetch only the Bow River segment within Calgary, the agent intersects the queried river and city geometries to return the correct GeoJSON.
- **Modification:** The modification action enables the LLM to directly change the GeoJSON based on the query intention for arbitrary stylistic edits.
 - **Reduction:** Removes unwanted sub-features via LLM-driven JSON edits. For example, to visualize evolving historical borders not present-day, the agent deletes arbitrary shapes as requested by the user and emits the updated boundary without any set operations.
 - **Generation:** For simple requests that are not queryable, such as representative flight or sea routes (which the agent does not

have a database to query), the agent generates an estimated GeoJSON with key waypoints.

Parameter Selection. The agent is then instructed to select relevant module parameters. For example, if a highlight module requires text to display the number of gold medals won by a country or if a route color needs to be changed to contrast with the map style. Finally, the response (which relies on the function calling API) is parsed into a compatible JSON output with the animation functions available in the system. This is then rendered as an editable timeline. The agent can perform **module actions** to change the module animation in three ways: 1) It can retrieve or change relevant geospatial information, 2) It can edit module-specific parameters (such as zoom levels) based on the narrative requirements, and 3) It can edit the style of the module based on the script provided by the user. The agent chooses an appropriate action based on the user’s request.

Human-in-the-Loop (HITL) Editing. Once the agent generates an initial scene breakdown, the user can edit, rearrange, or delete any part and regenerate a scene breakdown. For example, the user can change the order the blocks appear in, add or delete blocks or edit short and long descriptions of individual map animation blocks to give additional context. Similarly, by accessing the researcher tab when a specific item in the timeline or scene breakdown is selected, the user can communicate with the researcher through a chat interface and ask for specific questions or modifications (**D3**). The user can iteratively refine the prototypes and follow up with changes via direct manipulation or natural conversation.

7 TECHNICAL EVALUATION

To evaluate our system’s performance, we conducted a technical evaluation.

7.1 Method

Dataset. We used 20 map-based storytelling prompts generated by GPT 4.5 to evaluate our system. These prompts were generated using 5 user-generated prompts as input, with additional context about our system such as “*In 1492, Christopher Columbus set sail from Palos, Spain, and after weeks at sea, he arrived in the Bahamas, forever altering world history.*”. We then tested these 20 prompts with four different LLM models: 1) GPT-4.5, 2) o1, 3) GPT-4o, and 4) GPT-3.5 to compare the performance. We did not include a baseline because we simply tested our system with text-to-video generation with Sora but with a simple test of 5 prompts, we immediately realized that this approach failed completely, (e.g. generate completely wrong video, generate completely inaccurate information, etc). Therefore, we evaluated our system across different LLM models.

Procedure. We entered each prompt into the Scene Breakdown agent and recorded both the generated modules and the time it took to generate them. After that, we called the Researcher agent to process the modules, with its execution time also measured. After the Researcher completed processing, we added the modules to the timeline and evaluate them. We then graded each module on its accuracy, with criteria varying depending on the module type. Camera control modules must have 1) the correct location, and 2) the correct zoom level. Highlight modules must have accurate

⁹<https://nominatim.org/release-docs/latest/api/Search/>

boundaries, points and lines. Animated element modules must be within 1 km of the real routes, as well as being physically plausible (e.g. train routes musn't go through bodies of water). We evaluated accuracy based on our judgment as no standardized or objective ground truth was available for these types of outputs.

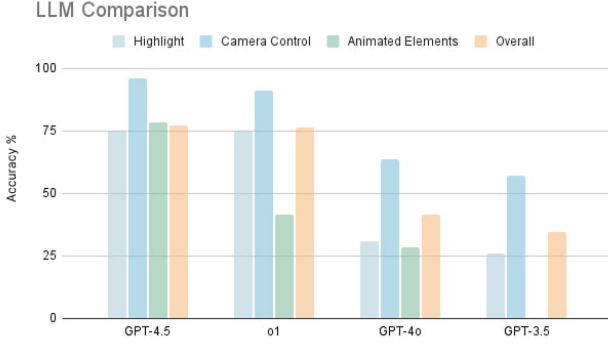


Figure 11: Results of accuracy across four LLM models.

7.2 Results

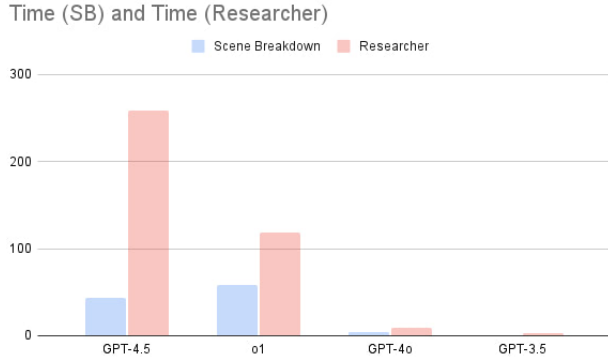


Figure 12: Results of processing time (in seconds) across four LLM models.

Figure 11 presents the accuracy results across the four evaluated models. As expected, we observed a clear improvement in system performance with each successive model. For context, GPT-3.5 is the oldest model, followed by GPT-4o, then o1, then GPT-4.5. These results suggest that the capabilities of the underlying LLM play a substantial role in determining the overall performance of our system. As more sophisticated models become available, we expect corresponding improvements in our system's capabilities.

Figure 12 shows the average processing time (in seconds) for each model used in our system. As expected, GPT-4.5 took the longest to produce an output, followed by o1, 4o, and 3.5 respectively. Interestingly, GPT-4.5 took less time to produce an output for the scene breakdown, this is presumably due to GPT-4.5 not being a reasoning model. As the efficacy of the models increases so does

the time. Failure cases can be attributed to the wrong or inaccurate geocoding by the LLM or complex and historical region queries not available for the LLM to fetch from the databases (even with addition, reduction functions). For example, a test prompt in our technical evaluation involved a path through space to the Moon, which we do not support. Moreover, we observed worse performance when the query involved historical data that needed to be mapped to present-day, queryable geography. For example, when visualizing the Roman Empire, the system failed to cover important provinces or returned only small fragments of its territory (which corresponds to present day) rather than the full extent required.

7.2.1 System Bottlenecks. The primary bottleneck of the system is the Scene Breakdown Agent. In a single shot (single user prompt), the number of animation modules that can be generated is bound by this agent's token limit. At the time of testing, the GPT-4o output token limit is 16k [17]. We measured the average token size for one animation module as 300. Route modules require the most tokens, and zoom modules require the least. The theoretical limit of our system then becomes 54 animation blocks for a single user input.

8 USER EVALUATION

To understand how MapStory supports novice users in creating map animations, we conducted a usability study with twelve participants. We aimed to evaluate the tool's expressiveness, learnability, support for creative exploration, and factual correctness.

8.1 Method

Participants. We recruited 12 participants (5 male, 7 female; ages 21–30, mean=25) via university mailing lists and snowballing sampling. All participants had minimal or no prior experience with animation tools.

Tasks. Participants were asked to complete two animation tasks:

- **Task A (Guided):** Create a short animation based on a pre-written script about the migration of monarch butterflies from Canada to Mexico.
- **Task B (Open-ended):** Create an animation based on a self-authored narrative (e.g., historical event, personal story, or travel journey).

Procedure. Each session lasted approximately 45 minutes and was conducted in person. Participants received a compensation of \$15 USD. Before starting the tasks, participants were given a brief tutorial explaining the study's purpose and the functionality of our system. They were encouraged to ask questions during the tutorial and Task A to ensure a clear understanding. However, to minimize external influence on their creativity, participants were not allowed to ask questions during Task B. For each task, participants were allowed to work on it until they were satisfied and were encouraged to think aloud during the study. To evaluate system accuracy, the experimenter recorded the number of animation modules created during each task. Afterward, participants were asked which modules they believed accurately reflected their prompts, allowing us to compute a subjective accuracy rate. After completing each task, participants filled out a questionnaire covering three aspects: 1) system usability, based on the System Usability Scale (SUS) [10], 2) creativity support, based on the Creativity Support Index (CSI) [14],

and 3) usefulness for each system feature, based on our 7-point Likert scale questionnaire. After both tasks, we conducted a semi-structured interviews to gain in-depth feedback on their experience. We include some example videos generated by participants in the video figure.

8.2 Results

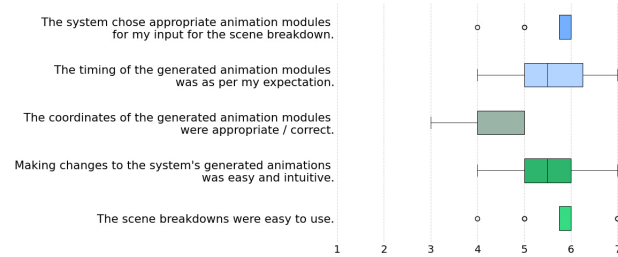


Figure 13: Results of the questionnaire for each system feature. The Likert scale ranges from 1 (strongly disagree) to 7 (strongly agree). Outliers (represented by circles) were defined as values that lie beyond 1.5 times the inter-quartile range (IQR) from the quartiles.

Accuracy, Usability, and Usefulness. Overall, the system generated a total of 114 animation modules during our user evaluations. Of those, participants rated 91 of them to be accurate (79.82%). *Task A* was found to be 89.47% accurate (34 / 38) and the failure cases were the result of users not being satisfied with the start and end position, which they then prompted the system again to change (After the user changes, the *Task A* success rate was 100%). *Task B* was found to be 75% accurate (57 / 76). Failures were almost always a result of the route or the highlight modules having imprecise coordinates, albeit they were typically within 1km of the correct result. For example, when users asks to zoom into “a random seven eleven in japan” the system returned a highlight around 1km away from a seven eleven. Another interesting find was for compound geojson queries like “the roman empire”, the system returned the correct regions but some regions were fully highlighted as returned by the present day boundaries, whereas they should be represented with partial borders. In terms of usability, the average System Usability Scale (SUS) score was 83.7 (SD=6.9), indicating high usability. As shown in Figure 13, system features were rated positively in terms of both accuracy and usability. This indicates that participants found the system features to be reliable and easy to use. In terms of perceived usefulness, the average scores were 5.75 (SD=1.29) for the scripting phase, 5.33 (SD=1.15) for the research phase, and 5.83 (SD=0.83) for the creation phase, demonstrating high perceived usefulness across all phases (Figure 14).

Creativity support by providing scaffolding. Figure 15 presents the results of the Creativity Support Index (CSI). As shown, participants generally felt that MapStory supported their creativity in addition to enabling them to create map animations, a task which they could not see themselves doing before. This perception was supported by interview data, which suggested that the system lowered initial barriers to entry by providing them with an initial “guide”

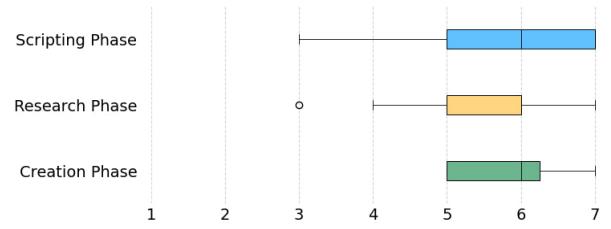


Figure 14: Results of the perceived usefulness of each work phase. The Likert scale ranges from 1 (not useful at all) to 7 (extremely useful). Outliers (represented by circles) were defined as values that lie beyond 1.5 times the inter-quartile range (IQR) from the quartiles.

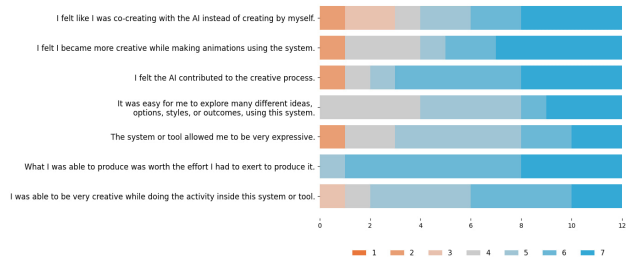


Figure 15: Results of the Creativity Support Index (CSI). The Likert scale ranges from 1 (strongly disagree) to 7 (strongly agree). Each bar in the chart represents the number of participants who selected each corresponding response value for the CSI items.

or a starting point. “In the beginning, it’s probably hard for me because there’s no starting point, and it’s hard to imagine. Everything is like drawing a picture without a draft. But once you have the draft, everything seemed pretty clear, you know, like the immediate next step.” (P9) Additionally, by decomposing the animation process into manageable steps, the system enabled more efficient prototyping and iteration, which participants found particularly beneficial. “A prototyping tool for different shots would be really interesting, and very quick to do actually.” (P6) These findings suggest that MapStory not only makes animation creation more accessible by reducing the cognitive load of starting from scratch but also provides structural support for sustained creative exploration and refinement.

Emerging Use Cases. Participants envisioned novel and often informal use scenarios that went beyond our initial expectations, particularly highlighting casual and presentation-oriented applications outside educational or learning video creation. Notably, 8 out of 12 participants proposed using the tool to plan and share travel itineraries in a visually engaging way. “If I’m going on a trip or a vacation, I want to share with others, in a fun way, the route I’ll be taking and where we’re going. It sounds a lot more exciting than just giving them an Excel sheet with all the details” (P10). “If you were a travel agent, you could use it to visually present your proposed plan to clients” (P6). Also, participants proposed personal documentation scenarios. For instance, one participant mentioned using the tool to

visually record their visits to restaurants around the city. *"I can also use that to document all the restaurants that I've been to around the city"* (P10). Beyond video creation, the tool was seen as a potential resource for real-time classroom presentations. *"This would be a great educational tool. Like, if I were a teacher, I could demonstrate it to the whole class"* (P9). These unexpected use cases highlight the flexibility of the tool and its potential to support a wide range of applications beyond our original scope.

9 EXPERT INTERVIEWS

To assess MapStory's value and feasibility in professional workflows and gain insights into potential use, we conducted expert interviews with five professional map animators (N=5), each with 3–15 years of experience using tools like Adobe After Effects and Davinci Resolve.

9.1 Method

We recruited five professional map video creators (5 male, ages 22–38). Two of these experts had also taken part in our earlier formative study. Each participant had at least two years of experience (2–10 years) producing map animation videos. Three out of the five experts regularly upload videos to YouTube, while the remaining two primarily do client work. Collectively, their Youtube channels have an average of 120k subscribers (5k–300k as of March 2025). We contacted them via Youtube or Instagram, conducted semi-structured interviews of approximately 60 minutes over Zoom, and compensated each with \$50 USD. Experts were given a demo of MapStory and asked to use the system to prototype animations based on a familiar script. We followed up with questions to understand how MapStory could potentially integrate into their workflows.

9.2 Insights and Findings

Experts found script-driven animation intuitive and were satisfied with animation controllability, though several noted a desire for even more fine grained control available in professional software. We condense insights from these interviews below.

Scene breakdown as the spine of story. Participants found that the scene breakdown not only provides animation guides but also acts as a medium of story exploration. Animators felt that they could stay in *"script mode"*, using short text edits to restructure the whole animation instead of dragging keyframes or rewriting code, while visualizing how the story is presented in real-time. E1 explained that the scene breakdown provides an initial guided path, which *"is effectively the basics of a story"* and keeps narrative flow intact, and he can directly control and change the narrative: *"The auto-generated 'scene breakdown' feels like way-points in your story that you can tweak in plain language."* E5 finds the scene breakdown to be a medium of updating story beats first for editing the visuals. He explains that he traditionally wrote scripts in *"Then/but/therefore"* chunks and then assigns timecodes to make animations in After Effects. But now that he can directly make/write those beats, he can spend his creative time adjusting nuance instead of laying every keyframe. He says, *"[MapStory] generates those things, the clips, the timeline items, and then you can manipulate it later to fine-tune what you're doing; that's the most beneficial."*

Animation by means of conversation. Experts described the chat-based Researcher as a dialogue that lets them refine map animations and data with text instead of keyframes. E1 explained that when a border is not up to their liking they can *"just ask for information and not just animation,"* saving them from *"manually rearranging keyframes or remaking the scene by hand"*. E5 echoed the sentiment, saying *"AI can take out the grunt work, but then people can tweak the results to a really high degree afterwards"*. E3 quantified the gain: being able to chat-edit a mock-up *"could save me like 40% of the time"*, and, when modern polygons include unwanted islands, *"being able to iteratively change it for my liking is the best part"*. Across interviews, conversational animate instructions—whether querying historical borders, trimming overseas territories, or requesting fresh coordinates or information, were praised as a faster, more flexible alternative to the traditional cycle of swapping between Wikipedia, Geo editing tools like QGIS and After Effects.

Iterative and editable animation authoring. Creators repeatedly framed MapStory as a way to *rough-cut first, polish later*, using fast textual edits to explore ideas and postpone heavy key-framing until they are sure of the direction. E1 calls the system a *quick tool that still gives high quality output*—good enough to visualize and use for final polish—and estimates it can *shave off about two days on a typical five-day timeline* because the early prototyping phase no longer stalls on full-resolution renders. The creators still preferred to polish the final video to their liking with advanced editing tools that offer precise control. E5 values the fact that MapStory can *generate the timeline items and then manipulate them*, which he says is *the most beneficial* part of his workflow: MapStory lets him iterate on narrative story beats before committing to detailed animation *"The ability to manipulate is paramount"*. E1 notes that polygon edits are now more approachable and iterative instead of time consuming of GeoJSON changes: *It does these computations by itself ... searches for everything and compiles it for you*. Even though the system wasn't perfect, he still appreciated ability to keep prompting or manually make changes *"I really like how you can ask for information and not just animation"*. E2 contrasts this with his current After Effects practice where, *once you make it, you can't change it... But [with MapStory] I can change it, but it will be time-consuming*. In MapStory he simply deletes a label and replaces a location highlight and the highlight and camera jumps from India to France, enabling many quick versions: *In 20 or 10 minutes I can do this kind of base animation... then later move it into After Effects*.

Speed also opens space for parallel ideation. E4 keeps several prototypes open at once: *You can do like three videos at the same time ... it handles all animation by itself so you don't have to worry about it*. Because the Researcher answers questions and returns edits, E2 expects to brainstorm more adventurous storylines, noting that the tool itself becomes a live fact-checking partner. But he also mentions that this would only benefit him during time crunches, and that he would want more precise control (like After Effects) for other projects *"It can't replace After Effects... your product's job is mapping, idea tool for a client demo."* Because each animation block is stored as editable prose, creators immediately thought about portability: E3 asked, *Could you save the specific timeline as a file and then import it later?*, treating the breakdown like an

edit-decision list that can hop between tools, while E2 imagined the same promptable timeline living inside Adobe’s flagship: *If we integrate this with After Effects, then I would love [to use it] ... it gives me reassessing thing*. Text-driven prototyping lowers the cost of experimentation, lets animators iterate early and often, and reserves advanced tools for the final layer of visual details.

Limitations and caution. Several experts flagged the risk that an LLM-powered “Researcher” might invent or distort facts. E4 warned that prompts can return “*wrong information or ... hallucinations ... I will double-check*” before publishing anything. Yet most participants stressed that fact-checking is not new creators have always had to validate sources. E5 summed it up bluntly: “*You can’t trust Wikipedia ... you’ve gotta do a lot of research to make sure things are historically accurate, a lot of stuff out there is just wrong.*”. Likewise, E3 treats AI outputs like any other draft, noting that he needs a “general idea” so he can spot errors and avoid “*accidentally including the AI mistake*” in his maps. Participants did appreciate the controllability our system provides. Moreover, the experts explained that they would not find themselves using MapStory as their final production tool but more for initial exploration, drafting, or pre-production for client demo or storyboarding. They prefer having complete creative control over their animations with advanced software which support keyframing like After Effects or Davinci Resolve.

10 FUTURE WORK

Modular extensibility of AI-assisted, text-driven animation prototyping. Our formative explorations have yielded a corpus of map-centric animation blocks; however, our modular system can be extended with new, custom blocks tailored to animators’ niche tastes. Participants envision creating their own stylistic modules—such as filters, transitions, or bespoke map-animation blocks, while some imagine importing a hand-designed basemap and letting the engine treat it as another selectable style, “*Imagine you can import ... an image, and then it would do the same thing*” (E3). Because each block is merely metadata plus a renderer hook, contributors can publish plug-ins (e.g., “smear-frame transition,” “parchment basemap”) to a shared registry. Experts also envision a community forming around trading map-specific animation blocks, basemaps, transitions, and filters, similar to in-game items.

Geocoding Ability of LLMs. Our system is constrained by how accurately LLMs can map textual descriptions onto geographic coordinates and generate valid GeoJSON (albeit geocoding accuracy improvement was never our goal). Although our functions give animators interactive, iterative control, reliable geocoding remains challenging. For example, when visualizing the Roman Empire we can highlight its key regions, but we must rely on present-day boundaries; portions of the historical empire correspond to only parts of modern countries—a level of detail our system cannot yet represent (users can, however, draw a rough outline manually). Despite this limitation, the text-driven animation workflow itself remains unchanged. We also acknowledge that our evaluation was restricted by API rate limits, which limited the number of prompts we could test. Future work should therefore conduct a more extensive assessment focused on geocoding to explore the limits and

capabilities of LLMs for any map-based visualization, not just animations. Our agent design is also limited by a design features two agents. We chose this design based on our formative evaluations and time constraints of animation authoring. However, future architectures could explore agents interacting over multiple rounds to keep refining the geocoded result of the previous agent until a verification agent is satisfied.

Beyond Maps: Broader implications of text or conversation driven animation workflows. The interviews suggest that MapStory’s text-driven “scene breakdown + Researcher” model could serve as a control layer for broader animations, not just maps. Once the timeline itself is conversational, the Researcher can query far more than geographic data; E5 already envisions layering non-map treatments—“*You can also add transitions or filters to the maps to give it more [board-game] kind of a thing.*”—and the same mechanism could contextually search a studio’s library of transitions, LUTs, or particle systems, insert the chosen effect blocks into the timeline (additionally add transitions or filters based on user’s intention), and expose their parameters for manual tweaking at every stage. This can create a reusable, domain-agnostic interaction that can accelerate storyboarding and live client iteration for any genre of animation.

11 CONCLUSION

This paper introduced MapStory, a text-driven map animation prototyping tool that enables users to create map-based animations through natural language editable at every stage. We contributed to the design space and LLM architecture of our system. Formative evaluations detailed workflow of map animation creation with a design space of map-animation blocks. MapStory introduces modular animation blocks, integrated geospatial querying researcher, and tight coupling between script and animation through a step-by-step scene breakdown. These features were implemented through novel architecture that leverages two LLM agents, one for scene breakdown and another one for research agent. Through technical evaluations, usability studies, and expert interviews, we found that MapStory supports fast iteration, encourages creative exploration, and aligns well with existing professional workflows. On the other hand, our findings also revealed several limitations, including occasional inaccuracies from LLM-generated outputs and needing verification and finer grained control. Future work should address these limitations to enable more accessible, expressive, and verifiable map animations.

ACKNOWLEDGMENTS

This research was partially funded by the NSERC Discovery Grant RGPIN-2021-02857, JST PRESTO Grant Number JPMJPR23I5, and Adobe Collaborative Research Gift. We also thank all of the participants for our user study.

REFERENCES

- [1] [n.d.]. TravelAnimator: Accessible Map Animation Tool. <https://travelanimator.com/>. Accessed: 2025-07-15.
- [2] 1375. Catalan Atlas. https://en.wikipedia.org/wiki/Catalan_Atlas.
- [3] 1869. Carte figurative des pertes successives en hommes de l’Armée Française dans la campagne de Russie 1812–1813. https://en.wikipedia.org/wiki/Charles_Joseph_Minard.

- [4] 2024. Veo 2. <https://deepmind.google/models/veo/>.
- [5] AEscreens. 2025. GeoLayers: The Ultimate Map and Data Visualization Plugin for After Effects. <https://aescreens.com/geolayers/> Accessed: 2025-03-12.
- [6] Tyler Angert, Mirosław Suzara, Jenny Han, Christopher Pondoc, and Hariharan Subramonyam. 2023. Spellburst: A node-based interface for exploratory creative coding with natural language prompts. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–22.
- [7] Ian Arawjo, Priyan Vaithilingam, Martin Wattenberg, and Elena Glassman. 2023. ChainForge: An open-source visual programming environment for prompt engineering. In *Adjunct proceedings of the 36th annual ACM symposium on user interface software and technology*. 1–3.
- [8] Jan Henry Belz, Lina Madlin Weilke, Anton Winter, Philipp Hallgarten, Enrico Rukzio, and Tobias Grosse-Puppenthal. 2024. Story-Driven: Exploring the Impact of Providing Real-time Context Information on Automated Storytelling. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–15.
- [9] Stephen Brade, Bryan Wang, Mauricio Sousa, Sageev Oore, and Tovi Grossman. 2023. Promptify: Text-to-image generation through interactive prompt exploration with large language models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–14.
- [10] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.
- [11] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. 2024. Video generation models as world simulators. *OpenAI Blog* 1 (2024), 8.
- [12] Yining Cao, Jane L E, Zhutian Chen, and Haijun Xia. 2023. DataParticles: Block-based and language-oriented authoring of animated unit visualizations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–15.
- [13] Mengyu Chen, Marko Peljhan, and Misha Sra. 2021. EntangleVR: A visual programming interface for virtual reality interactive scene generation. In *Proceedings of the 27th ACM symposium on virtual reality software and technology*. 1–6.
- [14] Erin Cherry and Celine Latulipe. 2014. Quantifying the creativity support of digital tools through the creativity support index. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21, 4 (2014), 1–25.
- [15] John Joon Young Chung, Woosuk Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. 2022. TaleBrush: Sketching stories with generative pretrained language models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–19.
- [16] Richard C Davis, Brien Colwell, and James A Landay. 2008. K-sketch: a 'kinetic' sketch pad for novice animators. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 413–422.
- [17] Samantha Dawson. 2024. OpenAI launches experimental GPT-4o Long Output model with 16x token capacity. *VentureBeat* (15 Aug. 2024). <https://venturebeat.com/ai/openai-launches-experimental-gpt-4o-long-output-model-with-16x-token-capacity/>
- [18] Vanessa P Dennen and Kerry J Burner. 2008. The cognitive apprenticeship model in educational practice. In *Handbook of research on educational communications and technology*. Routledge, 425–439.
- [19] Tong Gao, Jessica R Hullman, Eytan Adar, Brent Hecht, and Nicholas Diakopoulos. 2014. Newsviews: an automated pipeline for creating custom geovisualizations for news. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 3005–3014.
- [20] Google. 2025. Veo 2 - Google DeepMind. <https://deepmind.google/technologies/veo/veo-2/> [Online; accessed 2025-03-31].
- [21] Aditya Gunturu, Yi Wen, Nandi Zhang, Jarin Thundathil, Rubaiat Habib Kazi, and Ryo Suzuki. 2024. Augmented Physics: Creating Interactive and Embedded Physics Simulations from Static Textbook Diagrams. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–12.
- [22] Anton Gustafsson, John Richard, Liselott Brunnberg, Oskar Juhlin, and Marco Combetto. 2006. Believable environments: generating interactive storytelling in vast location-based pervasive games. In *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*. 24–es.
- [23] Geo History. 2021. History of the Major Trade Routes - Summary on a Map. <https://www.youtube.com/watch?v=Q7xp1-VvtZ0> Accessed: 2025-01-06.
- [24] Bernd Huber, Hujung Valentina Shin, Bryan Russell, Oliver Wang, and Gautham J Mysore. 2019. B-script: Transcript-based b-roll video editing with recommendations. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–11.
- [25] Amir Jahanlou and Parmit K Chilana. 2022. Katika: An end-to-end system for authoring amateur explainer motion graphics videos. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [26] Peiling Jiang, Jude Rayan, Steven P Dow, and Haijun Xia. 2023. Graphologue: Exploring large language model responses with interactive diagrams. In *Proceedings of the 36th annual ACM symposium on user interface software and technology*. 1–20.
- [27] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, and George Fitzmaurice. 2014. Kitty: sketching dynamic and interactive illustrations. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 395–405.
- [28] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, Shengdong Zhao, and George Fitzmaurice. 2014. Draco: Bringing life to illustrations. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. 579–582.
- [29] Rubaiat Habib Kazi, Tovi Grossman, Nobuyuki Umetani, and George Fitzmaurice. 2016. Motion amplifiers: sketching dynamic illustrations using the principles of 2D animation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 4599–4609.
- [30] Brian Felipe Keith Norambuena and Tanushree Mitra. 2021. Narrative maps: An algorithmic approach to represent and extract information narratives. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW3 (2021), 1–33.
- [31] Knowledgeia. 2022. How did The Silk Road Actually Work? <https://www.youtube.com/watch?v=J-pfeBssMw> Accessed: 2025-01-06.
- [32] Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. 2017. Computational video editing for dialogue-driven scenes. *ACM Trans. Graph.* 36, 4 (2017), 130–1.
- [33] Mackenzie Leake and Wilmot Li. 2024. ChunkyEdit: Text-first video interview editing via chunking. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–16.
- [34] Fan Lei, Yuxin Ma, A Stewart Fotheringham, Elizabeth A Mack, Ziqi Li, Mehak Sachdeva, Sarah Bardin, and Ross Maciejewski. 2023. GeoExplainer: A visual analytics framework for spatial modeling contextualization and report generation. *IEEE Transactions on Visualization and Computer Graphics* 30, 1 (2023), 1391–1401.
- [35] Wanwan Li, Changyang Li, Minyoung Kim, Haikun Huang, and Lap-Fai Yu. 2023. Location-aware adaptation of augmented reality narratives. In *Proceedings of the 2023 CHI conference on human factors in computing systems*. 1–15.
- [36] Wenchao Li, Zhan Wang, Yun Wang, Di Weng, Liwenhan Xie, Siming Chen, Haidong Zhang, and Huamin Qu. 2023. GeoCamera: Telling stories in geographic visualizations with camera movements. In *Proceedings of the 2023 CHI conference on human factors in computing systems*. 1–15.
- [37] Jian Liao, Adnan Karim, Shivesh Singh Jadon, Rubaiat Habib Kazi, and Ryo Suzuki. 2022. Realitytalk: Real-time speech-driven augmented presentation for ar live storytelling. In *Proceedings of the 35th annual ACM symposium on user interface software and technology*. 1–12.
- [38] Real life Core. 2019. Why 50% of Canadians Live South of This Line. <https://www.youtube.com/watch?v=DFJAgb7dn78> Accessed: 2025-01-06.
- [39] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the Middle: How Language Models Use Long Contexts. *arXiv:2307.03172 [cs.CL]* <https://arxiv.org/abs/2307.03172>
- [40] Vivian Liu, Rubaiat Habib Kazi, Li-Yi Wei, Matthew Fisher, Timothy Langlois, Seth Walker, and Lydia Chilton. 2024. LogoMotion: Visually Grounded Code Generation for Content-Aware Animation. *arXiv preprint arXiv:2405.07065* (2024).
- [41] Patrik Lundblad and Mikael Jern. 2013. Geovisual analytics and storytelling using HTML5. In *2013 17th International Conference on Information Visualisation*. IEEE, 263–271.
- [42] Jiaju Ma, Li-Yi Wei, and Rubaiat Habib Kazi. 2022. A layered authoring tool for stylized 3d animations. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [43] Rohin Manvi, Samar Khanna, Gengchen Mai, Marshall Burke, David Lobell, and Stefano Ermon. 2023. Geollm: Extracting geospatial knowledge from large language models. *arXiv preprint arXiv:2310.06213* (2023).
- [44] Damien Masson, Sylvain Malacria, Géry Casiez, and Daniel Vogel. 2024. Directgpt: A direct manipulation interface to interact with large language models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–16.
- [45] Tomer Moscovich and John F Hughes. 2001. Animation sketching: An approach to accessible animation. *Unpublished Master's Thesis, CS Department, Brown University* 5 (2001).
- [46] Life of a Scrambler. 2025. How I Make My Maps. https://www.youtube.com/watch?v=GsojLuJpe_0.
- [47] OpenAI. 2023. Function Calling and Other API Updates. <https://openai.com/blog/function-calling-and-other-api-updates>. Accessed: 2023-XX-XX.
- [48] OpenAI. 2025. Sora. <https://openai.com/sora/> [Online; accessed 28-March-2025].
- [49] Pika. 2025. Pika. <https://pika.art/> [Online; accessed 2025-03-31].
- [50] Jonathan Roberts, Timo Lüddecke, Sowmen Das, Kai Han, and Samuel Albanie. 2023. GPT4GEO: How a Language Model Sees the World's Geography. *arXiv preprint arXiv:2306.00020* (2023).
- [51] Karl Toby Rosenberg, Rubaiat Habib Kazi, Li-Yi Wei, Haijun Xia, and Ken Perlin. 2024. DrawTalking: Building Interactive Worlds by Sketching and Speaking. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–25.
- [52] Inc. Runway AI. 2025. Runway | Tools for human imagination. <https://runwayml.com/> [Online; accessed 2025-03-31].
- [53] Kadek Ananta Satriadi, Barrett Ens, Tobias Czauderna, Maxime Cordeil, and Bernhard Jenny. 2021. Quantitative data visualisation on virtual globes. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [54] Kadek Ananta Satriadi, Jim Smiley, Barrett Ens, Maxime Cordeil, Tobias Czauderna, Benjamin Lee, Ying Yang, Tim Dwyer, and Bernhard Jenny. 2022. Tangible

- globes for data visualisation in augmented reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–16.
- [55] Vidya Setlur, Sarah E Battersby, Melanie Tory, Rich Gossweiler, and Angel X Chang. 2016. Eviza: A natural language interface for visual analysis. In *Proceedings of the 29th annual symposium on user interface software and technology*. 365–377.
- [56] Jae-Eun Shin and Woontack Woo. 2023. How space is told: linking trajectory, narrative, and intent in augmented reality storytelling for cultural heritage sites. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [57] Paul Sonnentag, Alexander Obenauer, and Geoffrey Litt. 2023. Embark: Dynamic Documents as Personal Software. In *LIVE 2023: The Ninth Workshop on Live Programming*, Cascais, Portugal. <https://www.inkandswitch.com/embark/>
- [58] Hariharan Subramonyam, Wilmot Li, Eytan Adar, and Mira Dontcheva. 2018. Taketools: Script-driven performance animation. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 663–674.
- [59] Bekzat Tilekbay, Saelyne Yang, Michal Adam Lewkowicz, Alex Suryapranata, and Juho Kim. 2024. ExpressEdit: Video Editing with Natural Language and Sketching. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*. 515–536.
- [60] Anh Truong, Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2016. Quickcut: An interactive tool for editing narrated video. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 497–507.
- [61] Tiffany Tseng, Ruijia Cheng, and Jeffrey Nichols. 2024. Keyframer: Empowering animation design using large language models. *arXiv preprint arXiv:2402.06071* (2024).
- [62] Edward R Tufte. 1991. Envisioning information. *Optometry and Vision Science* 68, 4 (1991), 322–324.
- [63] Edward R Tufte and Peter R Graves-Morris. 1983. *The visual display of quantitative information*. Vol. 2. Graphics press Cheshire, CT.
- [64] MotionArray Tutorials. 2025. Animate Maps Like The WSJ | After Effects Tutorial. <https://www.youtube.com/watch?v=1Pf2EwiP1W4>.
- [65] Boone Loves Video. 2025. GeoLayers 3: Travel Map Animation Tutorial. <https://www.youtube.com/watch?v=qdrvtU6WQ0>.
- [66] Bryan Wang, Yuliang Li, Zhaoyang Lv, Haijun Xia, Yan Xu, and Raj Sodhi. 2024. LAVE: LLM-Powered Agent Assistance and Language Augmentation for Video Editing. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*. 699–714.
- [67] Haochen Li1 Jonathan Leung1 Hao Wang and Zhiqi Shen. [n.d.]. Prompting LLMs to Solve Complex Tasks: A. ([n. d.]).
- [68] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091* (2023).
- [69] Zhijie Wang, Yuheng Huang, Da Song, Lei Ma, and Tianyi Zhang. 2024. Promptcharm: Text-to-image generation through multi-modal prompting and refinement. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–21.
- [70] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.
- [71] Denis Wood and John Fels. 1992. *The Power of Maps*. Guilford Press.
- [72] Haijun Xia. 2020. Crosspower: Bridging graphics and linguistics. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 722–734.
- [73] Haijun Xia, Tony Wang, Aditya Gunturu, Peiling Jiang, William Duan, and Xiaoshuo Yao. 2023. CrossTalk: Intelligent Substrates for Language-Oriented Interaction in Video-Based Communication and Collaboration. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–16.
- [74] Zihan Yan, Chunxu Yang, Qihao Liang, and Xiang’Anthony’ Chen. 2023. XCreation: A graph-based crossmodal generative creativity support tool. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–15.
- [75] Hui Ye, Chufeng Xiao, Jiaye Leng, Pengfei Xu, and Hongbo Fu. 2025. Mo-GraphGPT: Creating Interactive Scenes Using Modular LLM and Graphical Control. *arXiv preprint arXiv:2502.04983* (2025).
- [76] Lei Zhang and Steve Oney. 2020. Flowmatic: An immersive authoring tool for creating interactive scenes in virtual reality. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 342–353.

LIST OF IMAGE SOURCES

In Figure 2 and 4 we have used still frames from videos published by the following creators on YouTube.

Table 1: YouTube Video Frame Sources

Copyright Name	Channel Link
© Copyright Ahmed Sabry	https://www.youtube.com/channel/UCpLQ1u9_KSTVAO3PXkJho4w
© Copyright Atlas Pro	https://www.youtube.com/channel/UCz1oFxmMrgrQ82-276UCOU9w
© Copyright BBC News	https://www.youtube.com/channel/UC16niRr50-MSBwiO3YDb3RA
© Copyright Binkov's Battlegrounds	https://www.youtube.com/channel/UCPdk3JuQGxOCmIZLLt4drhw
© Copyright Business Insider	https://www.youtube.com/channel/UCcyq283he07B7_KUX07mmtA
© Copyright CGP Grey	https://www.youtube.com/channel/UC2C_jShtL725hvb1arSV9w
© Copyright CaspianReport	https://www.youtube.com/channel/UCwnKziETDbHJtx78nIkfYug
© Copyright Cycle V	https://www.youtube.com/channel/UCaCuAz2FXGVWMeT4jWUtTCA
© Copyright EmperorTigerstar	https://www.youtube.com/channel/UCUXqYwTCR6R3Wr-FkLTD4AQ
© Copyright Epic History	https://www.youtube.com/channel/UCvPXiKxH-eH9xq-80vpqgmKQ
© Copyright FactSpark	https://www.youtube.com/channel/UC5EcQrgvXsgGLR4ukxBTM8w
© Copyright Geo All Day	https://www.youtube.com/channel/UCdWuhQg9XRYXTiON98aC03w
© Copyright Geo History	https://www.youtube.com/channel/UC2CL2g2xFTZ0AEldxYVzQFg
© Copyright GeoGlobeTales	https://www.youtube.com/channel/UCPKVNqfbunZCkbvjBgf6CdA
© Copyright Geography By Geoff	https://www.youtube.com/channel/UC7-Z34pbJ0ZAOJRMUCEz0Cg
© Copyright Globe Addict David	https://www.youtube.com/channel/UCXJniNhLPJSeYXAfUzexHSA
© Copyright Histoire Géo	https://www.youtube.com/channel/UCYP6V7c4M026AXMr4ng6aQA
© Copyright History Mapped Out	https://www.youtube.com/channel/UCtzvIHQyRDL2mtetu6ZWsvw
© Copyright History Matters	https://www.youtube.com/channel/UC22BdTgxfuvUivjrjesETjg
© Copyright History On The Map	https://www.youtube.com/channel/UCYNb4d5zyaJ6KIN9knzoevg
© Copyright History Scope	https://www.youtube.com/channel/UCYb6v1AIX6prKI83DswM5iw
© Copyright History on Maps	https://www.youtube.com/channel/UC7JrMLRDJLUh5BPBYkMerQ
© Copyright HistoryMarche	https://www.youtube.com/channel/UC8MX9ECowgDMTOFTe8EUJw
© Copyright Jabzy	https://www.youtube.com/channel/UCoUkea_dZioNSJbi1vWDZKA
© Copyright Jay Foreman	https://www.youtube.com/channel/UCbbQalJ4OaC0oQ0AQaRQJ9g
© Copyright Johnny Harris	https://www.youtube.com/channel/UCmGSJVG3mCRXVOP4yZrU1Dw
© Copyright Kings and Generals	https://www.youtube.com/channel/UCMmaBzfCCwZ2KqaBJjkj0fw
© Copyright Knowledgeia	https://www.youtube.com/channel/UCuCuEKq1xuRA0dFQj1qg9-Q
© Copyright LEMMiNO	https://www.youtube.com/channel/UCRcgy6GzDeccI7dkbbBna3Q
© Copyright Look Back History	https://www.youtube.com/channel/UCnPYviDHmuqLeKG6kflTV8w
© Copyright Map Pack	https://www.youtube.com/channel/UCb0JxFSwHTzH0AEsAzQHV4w
© Copyright Megaprojects	https://www.youtube.com/channel/UC0woBco6Dgcxt0h8SwyyOmw
© Copyright Money & Macro	https://www.youtube.com/channel/UCCKpicnIwBP3VPxBZWDDeNA
© Copyright Odd Compass	https://www.youtube.com/channel/UCpZYcHFVksxNCwrUKGJMDyW
© Copyright Ollie Bye	https://www.youtube.com/channel/UC6gNjP1W4FXWExT5QpYkmhQ
© Copyright OnlyIAS Extended	https://www.youtube.com/channel/UCaIdhU356a0ej2MtFEylybA
© Copyright OverSimplified	https://www.youtube.com/channel/UCNiuvl7V8zACPpTmmNlqP2A
© Copyright Overly Sarcastic Productions	https://www.youtube.com/channel/UCodhH5mUeF-m_BsNueRDjcw
© Copyright PBS Eons	https://www.youtube.com/channel/UCzR-rom72PHN9Zg7RML9EbA
© Copyright RealLifeLore	https://www.youtube.com/channel/UCP5tjEmvPItGyLhmjdwP7Ww
© Copyright Server	https://www.youtube.com/channel/UCY_T0aB9ch-oYv7tBSuTC_w
© Copyright TED-Ed	https://www.youtube.com/channel/UCsooa4yRKGN_zEE8iknghZA
© Copyright TLDR News Global	https://www.youtube.com/channel/UC-uhvujip5deVcEtLxnW8qg
© Copyright The Generalist Papers	https://www.youtube.com/channel/UCN9UPjA8I-uwvAy0-N9maOA
© Copyright The Wall Street Journal	https://www.youtube.com/channel/UCK7tptUDHh-RYDsdxO1-5QQ
© Copyright Theology Academy – Christianity	https://www.youtube.com/channel/UCq7kcinj-b9d57THam2QOBQ
© Copyright True Story Docs	https://www.youtube.com/channel/UCALxmR5IDHHjN25Fb0ODZdw
© Copyright Uncovering	https://www.youtube.com/channel/UCrsm1dqSALPG5BCCi0EBug
© Copyright Virtual High School	https://www.youtube.com/channel/UCveQN5D9y28Xou9jMBQkbGQ
© Copyright Vox	https://www.youtube.com/channel/UCLXo7UDZvByw2ixzpQCufnA
© Copyright WeatherWatchTV	https://www.youtube.com/channel/UC6J_1lBR727yUrcm9m140Ug

continued on next page

continued from previous page

Copyright Name	Channel Link
© Copyright Wendover Productions	https://www.youtube.com/channel/UC9RM-iSvTu1uPJb8X5yp3EQ
© Copyright WonderWhy	https://www.youtube.com/channel/UCcEPmwpXKrKzZahqjwpIAsQ
© Copyright pigeoninanutshell	https://www.youtube.com/channel/UCQQwkFEwF4TfyOr26Yu6-vA