

SVGRAFFITI

Remixing the Web with Vector Illustrations

Tongyu Zhou
Adobe Research
San Jose, CA

Joshua Kong Yang
Brown University
Providence, RI

Eric Nai-Li Chen
Brown University
Providence, RI

Jeff Huang
Brown University
Providence, RI

ABSTRACT

Website experiences are often interpreted differently depending on a viewer's background; these interpretations are typically expressed outside of the website itself through comments, fan art, or other forms of media. In this work, we introduce a proof-of-concept tool, SVGraffiti, that allows users to remix existing webpages through expressive, site-specific visual overlays. Users can link recorded mouse events to changes in imported SVG illustration or DOM states, resulting in reactive visual designs that evolve alongside website navigation. These remixes can be exported as bookmarklets, allowing them to be applied locally and shared with others, or incorporated publicly into the source website itself. Through two gallery examples where we reinterpret New York Times articles, we demonstrate how the tool supports creative, interpretive engagements with web content. We discuss the implications of this approach for digital graffiti, web-based art placement, and participatory meaning-making on the web.

Authors Keywords

digital graffiti; web remix; bookmarklet; vector illustration

CSS Concepts

- Human-centered computing --> Graphical user interfaces; Web-based interaction

CC-BY-NC-ND

This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivs International 4.0 License](https://creativecommons.org/licenses/by-nc-nd/4.0/).

DIS '26 June 13--17, 2026, Singapore, Singapore

© 2026 Copyright is held by the owner/author(s).

ACM ISBN 979-8-4007-2563-0/2026/06

<https://doi.org/10.1145/3800645.3812830>

Figure 1. SVGraffiti enables the remix of existing websites by allowing users to overlay vector-based illustrations and link them to sequences of recorded mouse events, resulting in reactive experiences.

Bookmarklet Maker

Effective URL

SVG

Interactions JSON

Editor

navigation

Enable zoom?



Enable pan?



saved interactions



waypoints



Scrolling takes you from start to end.

component colors



INTRODUCTION

By construction, web experiences—from the initial wireframe design to fully deployed webpages—are typically shaped by the perspective of their authors. Once the website goes live, however, audiences are free to interpret and engage with it in their own ways, through the lens of their own unique backgrounds and beliefs [29, 36].

Today, these interpretations are rarely materialized within the website itself. Instead, visceral responses to online media are typically externalized through separate media such as comments, fan art, reaction videos, and blog posts. While these forms of expression, when viewed alone, can reshape how content is understood, they remain detached from the original source media. We argue that situating these interpretations and perspectives *directly within the webpage*—similar to scribbling notes in a book—provides a more effective and directly accessible way to embed personal perspectives into the same space where the content is encountered.

Existing approaches in web-development and design tools [19, 26, 40, 47, 49] offer partial solutions. For example, annotations [15, 50] on top of webpages allow users to overlay commentary to help guide the viewing experience. Alternatively, visual media can be directly embedded [22, 42] into the source code of the page itself. However, these components are often confined within their own discrete `<div>` containers and cannot interact with the rest of the page. There is still a lack of support for integrated interpretations of web content.

To fill this gap, we propose SVGraffiti, a proof-of-concept web extension tool that allows users to create non-destructive, site-specific remixes of existing webpages. Inspired by graffiti as a form of situated, often unauthorized expression, SVGraffiti enables users to author expressive, site-specific visual compositions onto webpages

without modifying their underlying source code. Using the tool, users can record sequences of mouse events (e.g. clicks, scrolls), then link these sequences to pan and zoom changes in imported vector-based illustrations. This results in immersive web experiences where illustrations dynamically react to user navigation without needing to access the original source code. While SVGraffiti shares surface similarities with other web development and design-focused tools, we do not position it as a mechanism for improving usability or streamlining workflows. Instead, it supports subjective, creative remixes that could challenge, support, or transform the meaning of a webpage through visual composition and interaction.

There are several methods to share these designs with other users, depending on their willingness to install the web extension or availability of the source code of the page the experience was designed for. These options are illustrated in the table on the right. For those with the web extension, the tool can export a JSON file which can be re-imported to launch the interactions locally. For those with access to the source code, the tool can

Has access to?		Usage		
Web Extension	Source Code	Method	Scope	Editable?
✓		Add SVG & JSON to extension	Local	✓
	✓	Embed in <code>index.html</code>	Global	✗
		Add as web bookmark	Local	✗

Figure 1. Table depicting 3 ways SVGraffiti designs may be exported and used.

export an embeddable bookmarklet so that any visitor to the page can globally experience it. For those with neither, the same bookmarklet can be locally added to a user’s private list of bookmarks; clicking on the bookmark, or pasting it into the URL field of the page, would generate the same effect. We demonstrate the potential of SVGraffiti through two gallery examples that reinterpret New York Times articles—one through additive visual commentary and the other through a more transformative narrative reimaging.

Overall, this paper contributes a lightweight system that allows end-users to create **reactive digital graffiti** on the web, expanding how users can engage with, reinterpret, and recontextualize online media.

The Structure of This Pictorial

We begin by discussing existing practices from prior literature on interactive web experiences, one from a more practical DOM recording for authoring and debugging perspective and the other from a customization for personalization perspective. We then synthesize these ideas to frame our system as a device for crafting “digital graffiti” for the web, taking inspiration from their physical counterparts. We describe the implementation of the resultant proof-of-concept tool, SVGraffiti, and demonstrate how the output designs can be easily exported as a lightweight bookmark for local viewing. We further provide two gallery examples demonstrating our proof-of-concept tool in action.

A HISTORY OF INTERACTIVE WEB EXPERIENCES

Past work on interactive web experiences can be divided into two categories: 1) those that focus on the interactions themselves and 2) those that focus on the presentation of web content that instigate or result from those interactions.

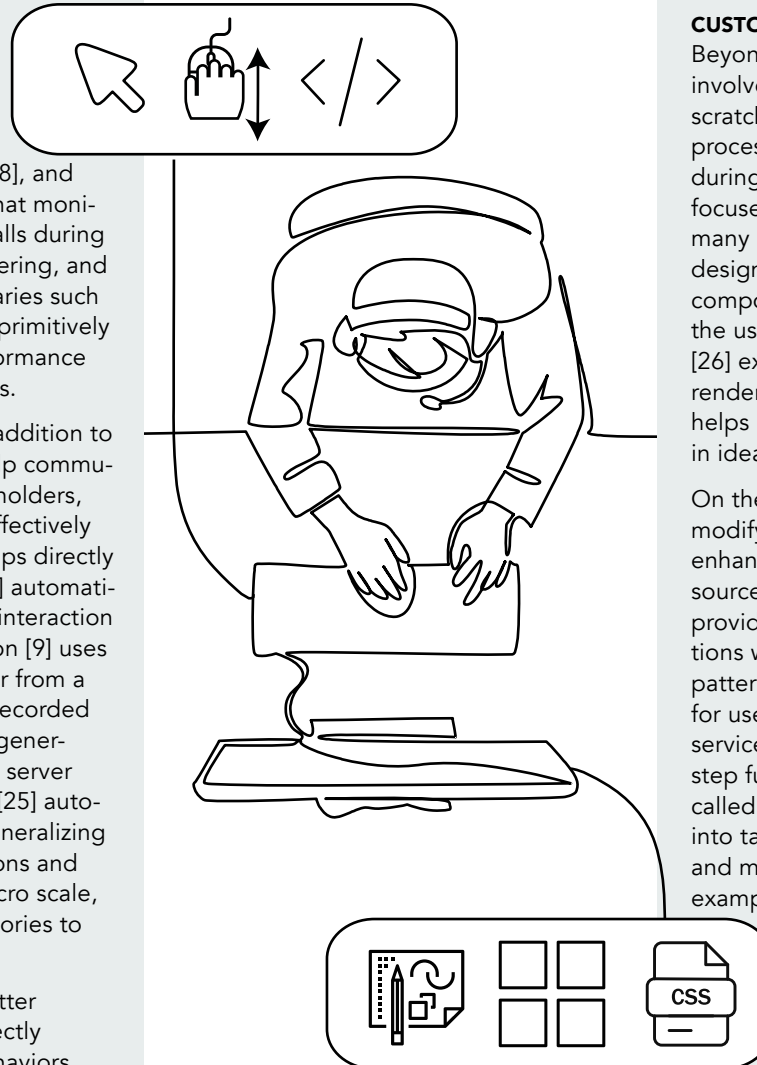
INTERACTION & DOM RECORDING ON THE WEB

Recording and replaying interactions and DOM changes on the web allow users to more deeply understand web behavior. Previous systems such as Scry [8], Unravel [17], Telescope [18], FireCrystal [37], Ply [28], Timelapse [8], and WebVSD [32] all fall into the category of tools that monitor HTML changes and/or JavaScript function calls during interaction for code debugging, reverse engineering, and understanding. Beyond academic research, libraries such as rrweb¹, Fullstory.js², OpenReplay³, and more primitively (no access to full DOM) Chrome DevTools' Performance Recorder all provide APIs to record web sessions.

These recordings also have applied benefits in addition to understanding and analysis. For example, to help communicate design decisions between multiple stakeholders, CoCapture [10] introduces a strategy to more effectively convey UI behavior by remixing existing mock-ups directly and referring to them via hypertext. Doppio [11] automatically generates a screenflow diagram based on interaction flows to serve as visual documentation. Umitation [9] uses the recorded interactions to retarget UI behavior from a source website to a target one. Ringer [5] uses recorded interactions on a presumably unchanging UI to generate automation scripts for presumably changing server requests or underlying function calls. SUGILITE [25] automates arbitrary tasks across Android apps by generalizing the script from the recorded demonstrated actions and linking them to voice commands. At a more macro scale, Irchiver [20] records all naturalistic browsing histories to serve as the user's web archive.

The recording aspect of our tool falls into the latter category of *applied* benefits, and is used to directly program-by-demonstration the sequence of behaviors users wish to associate UI appearance changes with.

- 1 <https://www.rrweb.io/>
- 2 <https://developer.fullstory.com/>
- 3 <https://openreplay.com/>



CUSTOMIZATION & DESIGN OF WEB EXPERIENCES

Beyond traditional web development that predominantly involves writing standard HTML/CSS and JavaScript from scratch, multiple tools have been created to streamline this process. These tools fall into two categories: those beneficial during the beginning of the design process and those more focused on end-user development. In the former category, many systems aim to ameliorate the bottleneck during design-to-code and focus on automatically generating web components from UI mock-ups [3, 41, 49]. Alternatively, if the users prefer to use hand-drawn sketches, FrameWire [26] extracts interaction logic and sketch2code [40] extracts rendered HTML from paper prototypes. Swire [19] additionally helps retrieve UI designs similar to a user's initial sketch to aid in ideation.

On the other hand, end-user development usually involves modifying and adding additional functionality [13] or UI enhancements [44] to deployed web pages, even when the source code to these pages are inaccessible. Chickenfoot [6] provides a way to programmatically integrate web applications without accessing the underlying code through keyboard pattern matching instead. Marmite [48] provides a strategy for users to combine and remix existing web content and services from multiple sites. Applianceizer [14] takes this a step further into the physical realm by introducing a strategy called essential interface mapping to convert GUI components into tangible ones. Other tools focus less on functionality and more on the styling of the website's appearance. One example is Poirot [42], a Dev Tools alternative for designers

to create style edits using direct manipulation on a graphical interface. For those who prefer dictating their preferences instead, Stylette [22] allows users to use natural language to communicate styling, which is then automatically translated to CSS.

SVGraffiti focuses on end-user development, specifically on modifying functionality that *results* in visual changes. We differ from prior work by providing a demonstrative approach to triggering functionalities and the ability to author dynamic, rather than static, visual changes.

HOW SHOULD WE CREATE GRAFFITI FOR THE WEB?

The design of SVGraffiti hinges on two questions: 1) *why do people design or modify user interfaces on their own volition*, and 2) *how they do it*. We then synthesize the answers to these questions into a **metaphor of non-destructive graffiti for webpages**.

IDENTITY-ENHANCING CUSTOMIZATION

Prior studies that investigated the first question have divided interface design into two categories, either *identity-enhancing* or *control-enhancing* [21, 30]. The former focuses on augmenting the user's sense of self through personalization. These types of customizations, implied to be mostly aesthetic, can evoke strong senses of psychological connection [30]. This connection can be established, and at times is even *preferred*, when people are customizing UIs created by others [2]. Creating customizations in this way provides a unique opportunity to share the users' identity with others. Thus, we position our tool to specifically support the process when end-users want to modify existing webpages they did not create, which led us to adopt web extensions as our implementation strategy.

DESIGN-BY-DEMONSTRATION

The other facet of this tool focuses on the second question—how should we best author these customizations? Webpage customizations involve modifications to both the styling and possible interactions users can complete on a page. Standard approaches to altering styling involve writing or converting low fidelity designs to CSS [3, 26, 40, 41, 49]. However, these approaches usually result in static designs that do not account for the actual experiences of interacting with the page. For interactions, approaches usually involve writing code that overwrites existing ones [6] or transferring them from other websites [48]. Multiple rounds of iteration are often necessary as the designer needs to mentally translate between the exact interaction they want and the corresponding code representations. Instead, we suggest that borrowing a concept from end-user programming—*programming-by-demonstration* [12, 27, 31]—and applying that to website interaction design can allow the user to more directly convey both styling and interaction, as well as how interactions affect styling.



Figure 2. Dmitri Vrubel's "My God, Help Me to Survive This Deadly Love" [46] on the Berlin Wall; photo taken by Tony Webster.

WEB REMIX AS GRAFFITI

Altogether, we want our tool to help users create identity-enhancing customizations through demonstration. The resultant designs should help them express themselves on existing public-facing webpages. We can thus draw analogies between this type of customization and the notion of graffiti. Graffiti, formally defined by Merriam-Webster [33] as "usually unauthorized writing or drawing on a public surface," can be a provocative form of personal expression and a way to "make public space into private space [38]." In addition to personal expression, graffiti has historically served as a political and social intervention within public spaces, often challenging authority, ownership, and dominant narratives. Despite being "unauthorized," graffiti done mindfully like Dmitri Vrubel's "My God, Help Me to Survive This Deadly Love" [46] (above) and Banksy's "Balloon Girl" [4] (right) may contain merit valuable enough to warrant the disruption. Thus, we hope that this tool can encourage users to create webpage remixes akin to these exemplar pieces of graffiti—using creativity to enhance or re-imagine the way we perceive a previously familiar web experience through the lens of personal narrative.

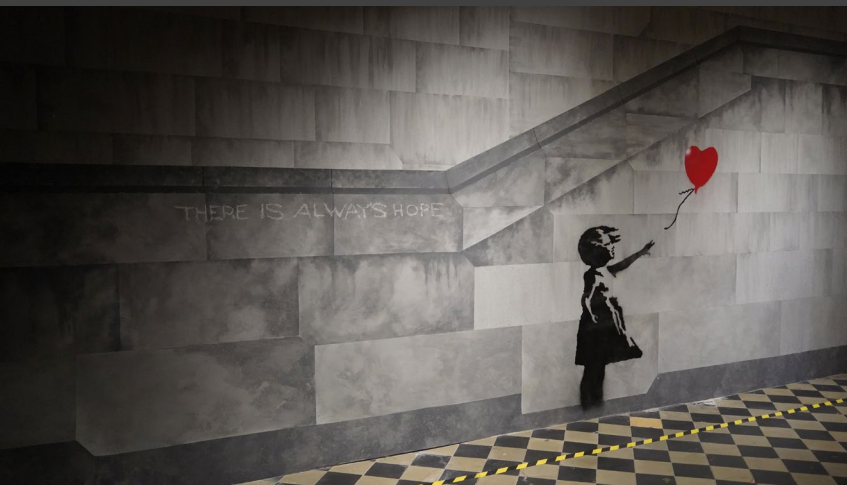
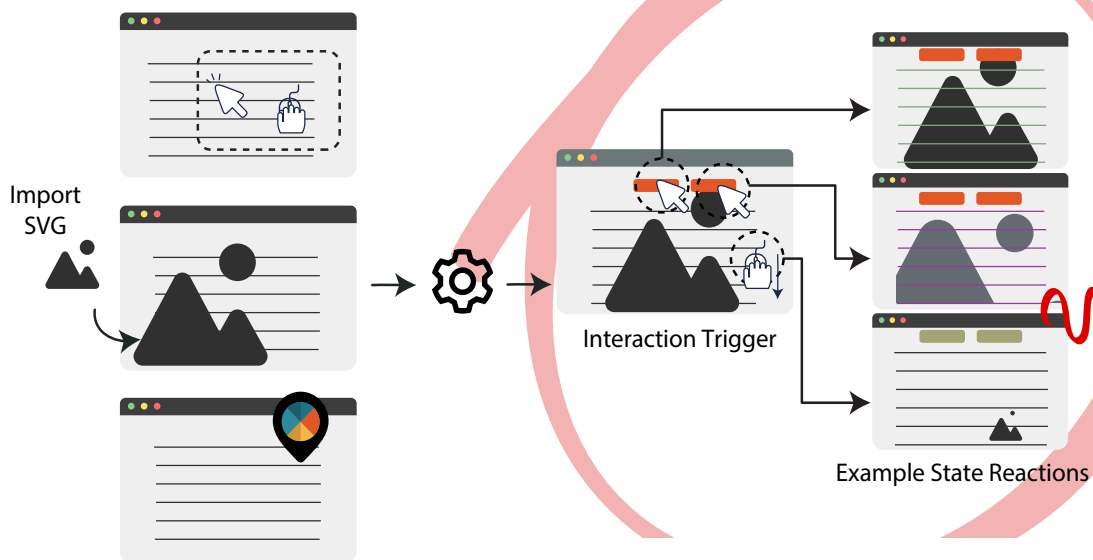


Figure 3. Banksy's "Balloon Girl" [4] in "The world of Banksy" exhibition; photo taken by Miguel Discart.

SVGRAFFITI

SVGraffiti builds upon an open-source vector drawing tool [51] for creating 2.5D parallax-based illustrations. We assume users create their vector illustrations using similar external drawing tools and are interested in using them to modify the experience of an existing webpage. Decoupling illustration creation and interaction authoring like this is intentional: this 1) allows users to use any illustration tool and 2) enables them to focus on mapping interactions to visual states with the webpage. To implement these behaviors, we rely on DOM injection and CSS manipulations. The extension is built using Lit, primarily in TypeScript, and has been tested on Google Chromium. Its workflow, summarized below, features two components. The first is an editor that allows users to record and sync mouse events to SVG and webpage state changes. The second is a bundler that collects all recorded interactions (saved as JSON) and bundles them with the SVG to create a minified JavaScript bookmarklet that can be called directly or added to the `index.html` of a webpage.

INTERACTION & SVG/DOM STATE AUTHORIZING



BOOKMARKLET EXPORT

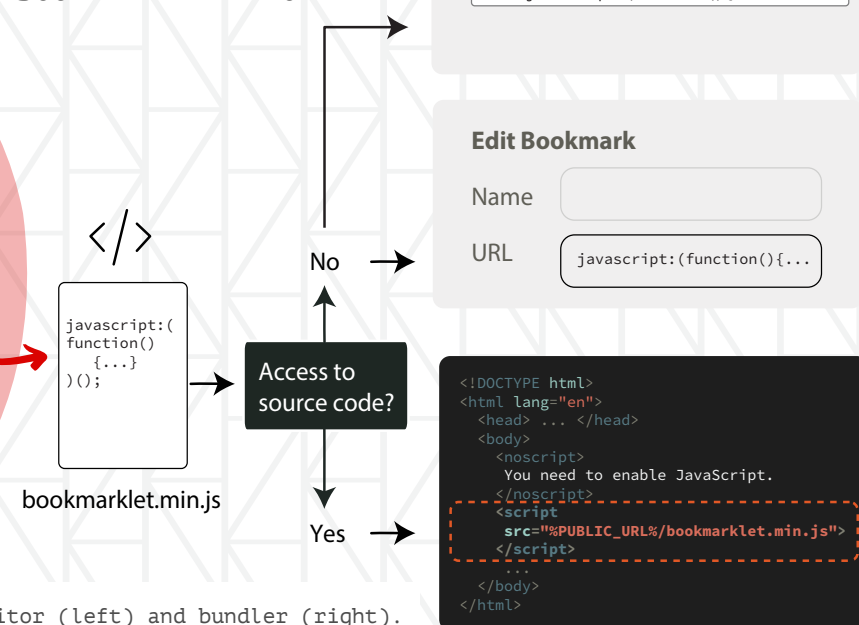


Figure 4. System pipeline of SVGraffiti depicting an editor (left) and bundler (right).

(1) Users can start by:

- Recording a sequence of interactions.
- Importing the SVG of a drawing onto the webpage.
- Selecting potential text colors.

(2) Using SVGraffiti, they can then bind the interaction sequences to SVG/website states, turning them into triggers. The bindings are automatically saved in a JSON file.

(3) The bindings (JSON) and image (SVG), with an optional script for SVG parallax effects (.js), are bundled in a minified script that can be either:

- Pasted directly into the URL field of the target page for local access.
- Saved as a bookmark for one-click local access.
- Injected into the `index.html` of the target page for global access.

INTERACTION & SVG/DOM STATE AUTHORIZING

To tether website interactions to illustration events and website components, the tool must record 1) a list of interactions users would like to serve as triggers and 2) a list of illustration and website states, then 3) link them together.

>> Recording Interactions

The tool currently supports the two of the most common types of interactions, 1) mouse clicks () and 2) mouse scrolls (), both of which can be recorded in any sequence. Upon clicking the “record” button, event listeners are added to the document. The click listener saves the mouse’s x and y positions, in addition to the `HTMLElement` of the clicked target. The scroll listener records the window x and y positions before and after the scroll action. For both events, we also record time displacements between each action and the URL of the page they occurred on. For scroll events, we also specifically disambiguate instances when the user has changed scroll directions or when they have paused scrolling for more than 500 milliseconds.

>> Replaying Interactions

Recorded interaction sequences are saved as circular icons () below the record/play buttons. Clicking on any of these icons () displays a preview of the interaction sequence, visualized as emojis. Clicking the “play” button will first verify that the URL of the interactions match the current URL (in case interactions span multiple subpages), then replay this sequence on the browser page with the previously saved time displacements between each.

Illustration & Webpage States

Recorded interactions can be linked to illustration and/or webpage states. Illustration states refer to locations, or “waypoints,” in the 2.5D drawing that have their unique depth (**d**) and position (**x, y**) values that control where the viewport of the SVG is relative to the entire illustration [51]. After the user imports an SVG, our tool sets it as the background of the webpage. Using scroll for zoom and click and drag for pan, they are able to navigate around the illustration directly, which sets the depth and position values, to identify which location on the illustration best fits the current website. This location is stored as a waypoint along with a PNG preview in a list; given any current location on the illustration, clicking on a waypoint will then morph the viewport via interpolation to that location over the span of a couple of seconds.

Webpage states refer to CSS properties of HTML elements that are on the page. The user can select which `HTMLElement` they want to modify by directly clicking on it, after which our tool will extract all the associated CSS of the `class` of the element for editing. While the tool only supports color properties for simplicity, this list of properties can be extended in the future for greater customization.



Figure 5. Sequences of recorded interactions are saved as circular icons.

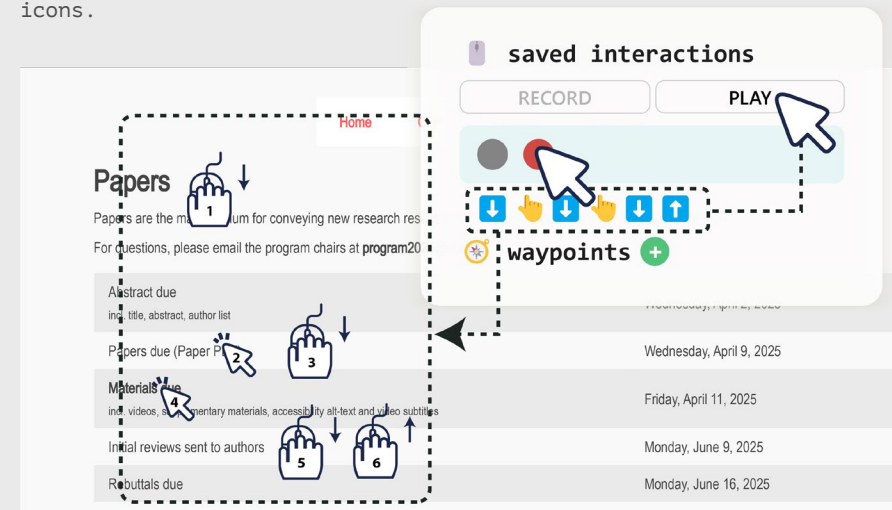


Figure 6. Recorded interactions are previewed as emojis and can be replayed on the webpage.

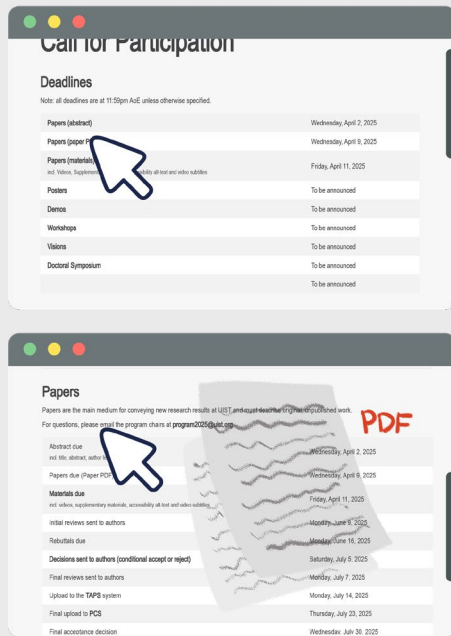


Figure 7. Click interactions only require one waypoint/webpage state per interaction.

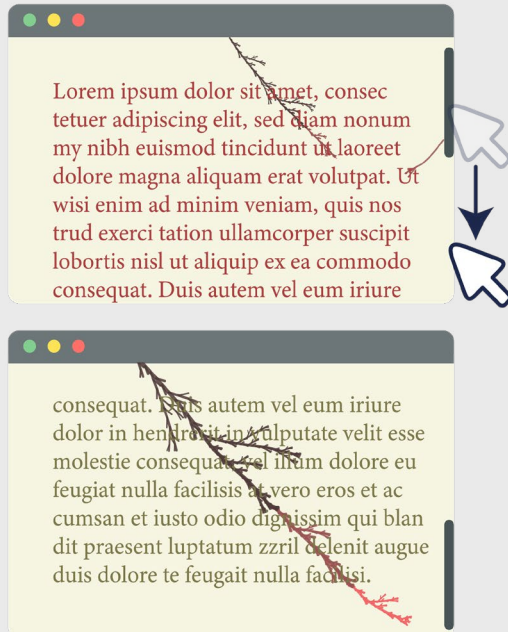
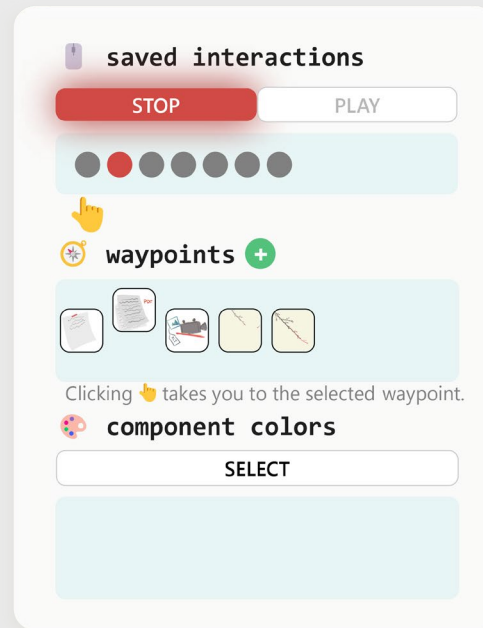


Figure 8. Scroll interactions require two waypoint/webpage states per interaction, a “start” and “end” state.



LINKING INTERACTIONS TO STATES

To link recorded interactions to stored illustration and website states, the user must first select a pre-recorded interaction sequence that serves as the trigger. Then, they can specify which illustration or webpage states the interaction will trigger. Under the hood, these associations are saved within a map for quick lookup and checked each time with click and scroll listeners.

>> Linking the Click Interaction

This example demonstrates how a click interaction (👉) can be linked to an illustration state. In this case, the click on the hyperlink of “Papers (paper PDF)” on the website is tied to a simple corresponding drawing of a PDF. When clicked, the waypoint directly jumps the background image to that location.

>> Linking the Scroll Interaction

This example demonstrates how a single page scroll interaction (⬇️) can be linked to both illustration and webpage states, specifically to the drawing of a larger tree branch and changing the text color from red to yellow. Given that this interaction moves the webpage from a before scroll position S_A to an after scroll position S_B , the waypoint parameter values (d, x, y) at any intermediary scroll position S with start (d_A, x_A, y_A) and end (d_B, x_B, y_B) waypoint states is just an interpolation:

$$(d, x, y) = (d_A, x_A, y_A) + [(S - S_A) / (S_B - S_A)] [(d_B, x_B, y_B) - (d_A, x_A, y_A)]$$

Similar interpolations are done for color values. We first convert the colors to RGB if they are formatted in hex, then compute intermediary (R, G, B) values based on start and end scroll positions and colors. This results in a smooth transition of the illustration gradually zooming in and the text changing from red to yellow as the user scrolls down on the page.

To support multiple authoring sessions, these interaction links are saved as a JSON file.

BOOKMARKLET EXPORT

As previously illustrated in Figure 1, the outputs can be exported and run on any modern web browser, depending on the particular needs of the user. If the user wants to continue editing, they can simply import the JSON file back into the web extension. Otherwise, if they want to share it with a wider audience, they can export a bookmarklet, or JavaScript-powered bookmark.

Bookmarklets are stored as `javascript: URLs` within bookmarks. To create the bookmarklet, we first prompt the user for the URLs to the SVG illustration used in the design and the exported JSON file. Then, the code bundles these two links with import code, all of the click and scroll event listeners, and the code that triggers the links between mouse interaction and SVG/DOM states. We use URLs instead of directly embedding the files because the typical character limit for bookmarklets is 8000. Typically the SVG illustration and parallax interactions are too large to directly embed. Instead, we include fetch requests to illustration URLs in the bundle to pull it before any event triggers are added. To ensure that the triggers only fire on specified instances of a webpage, the code also uses a `MutationObserver` to track DOM changes in order to remove or add the interactions as necessary.

We further minify the JavaScript code in the bundle by removing white spaces and newline characters to condense the code to 7957 characters (right). This minified code can then be directly copied into the URL or saved as a bookmark to be run later. Clicking on the bookmark would activate all the previously designed interactions and website changes. Alternatively, if the user does have access to the source code of the webpage they wish to modify, they can directly embed the bookmarklet snippet within a `<script>` in the `index.html` so it launches automatically upon page load for all visitors.

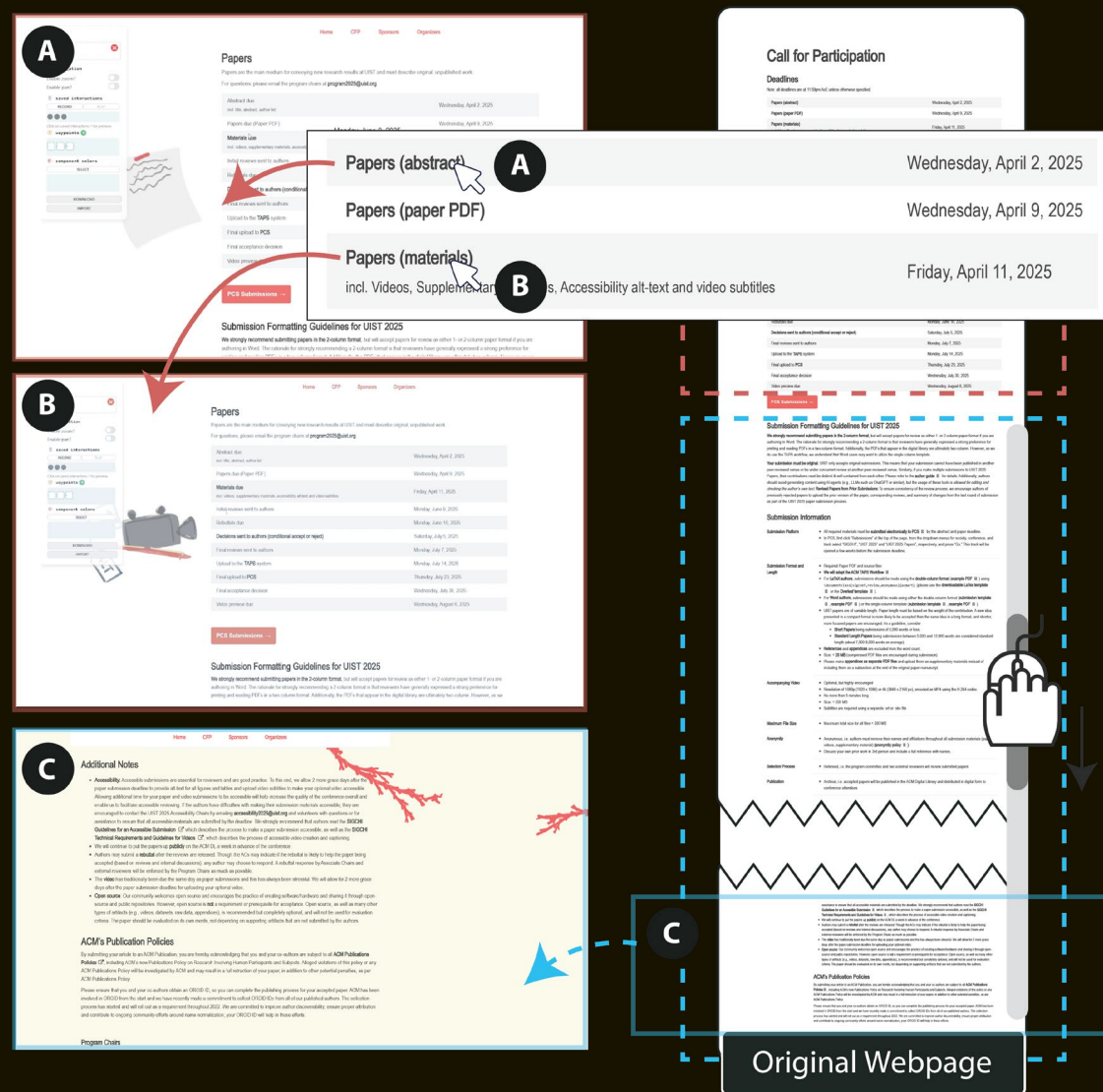


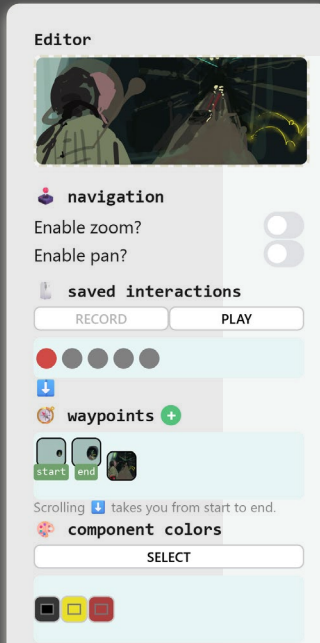
Figure 9. Example screenshots of the 2025 UIST conference page showing how (top) mouse clicks of hyperlinks can be linked to sketches and how (bottom) mouse scrolls can be linked to a blooming tree branch in the background.

GALLERY

We provide two examples that use SVGraffiti to remix the web, focusing on digital articles from *The New York Times* as sites for interpretive remix.

The first approaches remix from an additive angle, showing how an illustrator may use SVGraffiti to indirectly present an alternate opinion to the content of an article via complementary visuals. The second takes a more transformative approach, using an existing interactive article as a jumping point to construct a new narrative that reinterprets the original content.

We describe both the motivations behind and the processes involved in crafting the two experiences through demonstration. This type of evaluation by demonstration [24] can effectively reveal both what the tool can support and how users may use it.



RE-CONTEXTUALIZING FILM FESTIVAL REVIEW: "BETTER A BROKEN HEART THAN SHOT IN THE HEART"

Our first candidate for remix is a webpage that is visually very similar to the appearance of most digital articles: black text juxtaposed against a white background. This provides an ideal blank canvas to contribute a viewer's own experiences with the content presented in the article.

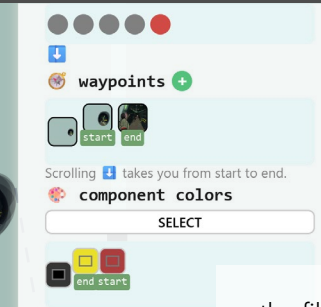
The original article, written by Stephen Holden in 1997, offers

a cynical review of Hong Kong director Wong Kar-Wai's film, "Fallen Angels." After a cursory summary of the film's plot, it describes a few stand-out moments before ultimately dismissing it, writing that *"the movie feels as weightless as the tinny pop music that keeps its restless midnight ramblers darting around the city like electronic toy figures in a gaming arcade."*⁴

In our re-contextualization, we introduce an illustration that references the film's closing scene, where two characters share a fleeting moment riding a motorbike through dark, seemingly endless tunnels, caressed by blurred neon lights. This scene, later regarded as iconic by many viewers, precisely captures the existential dread and lack of purpose produced by urban isolation, which many viewers resonated with. In this sense, Holden's criticism of

4 <https://www.nytimes.com/1997/10/07/movies/film-festival-review-better-a-broken-heart-than-shot-in-the-heart.html>

Figure 10. When scrolling through the Times article, the illustration zooms in from a small black dot to a full scene that spans the entire background. The text changes from red to neon yellow.



the film's "weightlessness," articulated using the metaphor of "electronic toy figures," is precisely the affect the film is trying to convey. Thus, by foregrounding this interpretation through SVGraffiti, our illustration reframes this perceived deficiency as an *intentional* expressive quality that visually manifests the characters' emotional states.

To craft this experience, we draw the motorbike scene and import it into the editor. We then record five separate scroll interactions, and bind them to different zoom parallax levels of the illustration. This means that as the reader scrolls through the article, the illustration becomes larger and more intimate, enveloping the viewer within the dark background.



<https://www.nytimes.com/interactive/2024/03/07/opinion/nuclear-war-prevention.html>

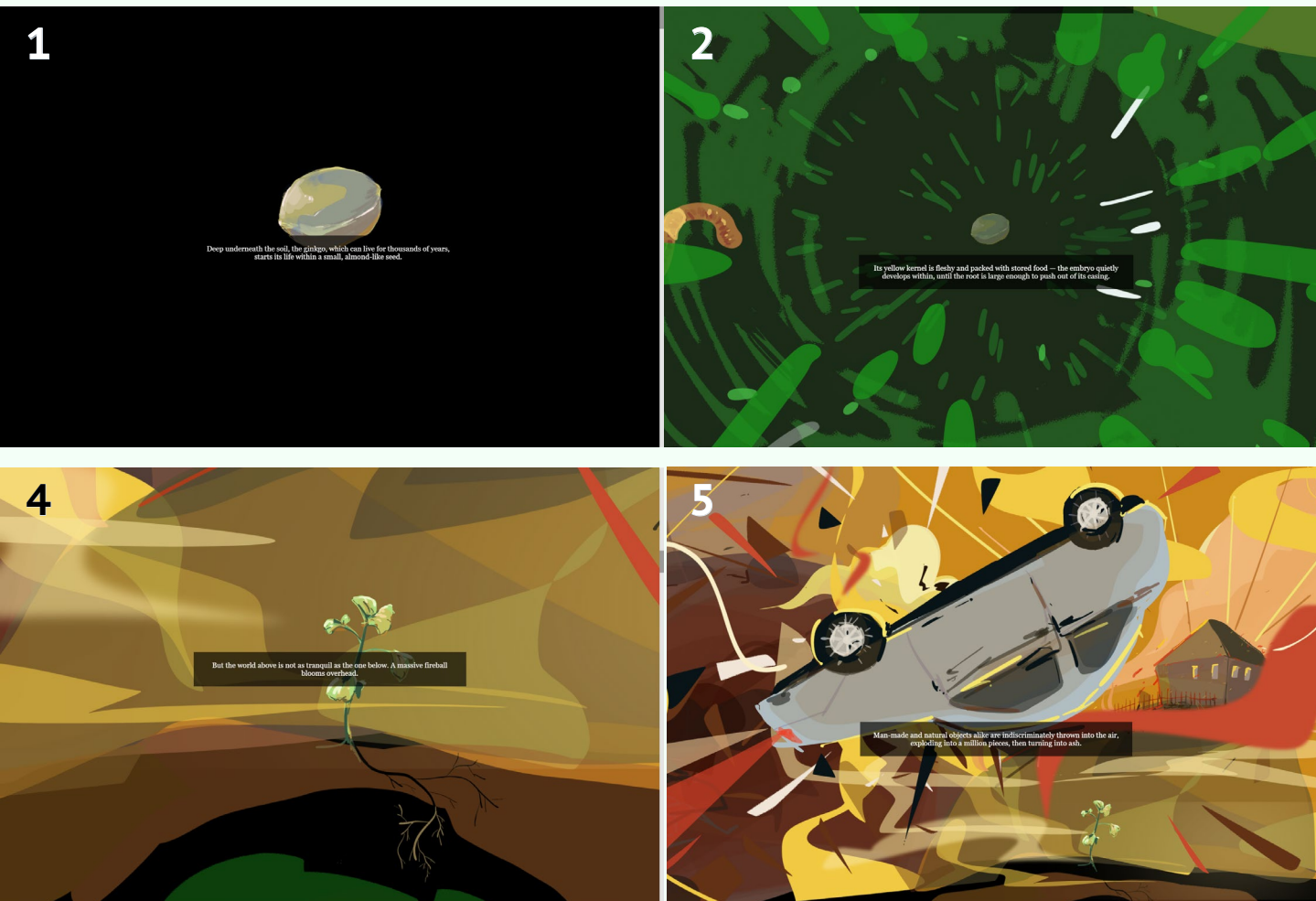
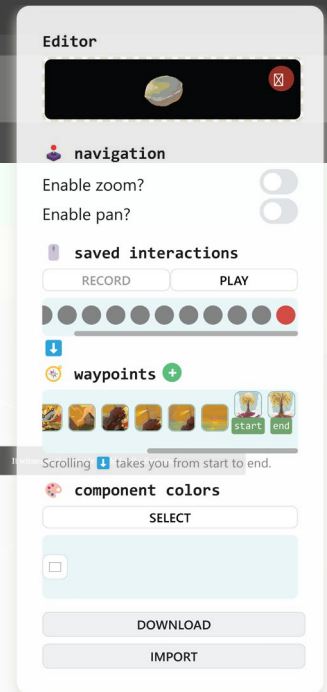


Figure 11. Panels 1-5 depict a ginkgo seed that slowly sprouts into a world of fire and carnage.

narrative [1]. We selected this article⁵ because its narrative evoked strong emotional responses and its use of parallax imagery, demonstrated through Tim McDonagh⁶'s illustrations, provided a rich inspiration for transformation. To respect the authorship of these drawings, we do not attempt to replicate them directly. Instead, we construct

⁵ <https://www.nytimes.com/interactive/2024/03/07/opinion/nuclear-war-prevention.html>

⁶ <https://www.mcdonaghillustration.com/>



RE-IMAGINING “THE BRINK”

Next, we explore an alternate scenario where SVGraffiti is used not to augment, but to reinterpret the narrative of a webpage. We draw inspiration from “The Brink” [16], a 2024 *New York Times* article that uses scrollytelling [34, 43] and parallax imagery to convey the dangers of nuclear war. This style builds on a tradition of interactive journalism exemplified by *New York Times*’ “Snow Fall,” [7], which won the Pulitzer Prize in feature writing in 2013 for its “deft integration of multimedia elements” [39]. Such works are often the result of extensive collaboration across researchers, designers, writers, and developers, and are carefully crafted to guide readers through a particular



an alternative story with accompanying text and our own illustrations to shift the meaning of the article. Specifically, we reframe the narrative from one centered on destruction to one focused on aftermath and renewal, imagining how nature may persist and reclaim the landscape. This example highlights how SVGraffiti can not only add commentary, but also reconstruct meaning using the *same interaction structure*.

The original story begins with a nuclear missile landing in the center of the screen and exploding into a flash of white. The subsequent force propels cars and buildings into the air. The scene then pans through burning piles of destruction, concluding at a hazy mushroom cloud. Our remixed panels of this can be seen annotated in sequential order. In our remix, we first replaced the imagery of the nuclear missile with that of a ginkgo tree seed, which is known for its hardiness and long lifespan. We draw analogies between the falling missile and the seed being placed into the ground; the flash of white now describes how the seed sees the outer world for the first time as it breaks through soil. There is ongoing fire and destruction outside, similar to the carnage from the original article. However, instead of ending on the grim note of the mushroom cloud, we conclude our scenes with the seed fully growing into a towering tree. This last panel suggests that regardless of the degree of destruction, there can still be hope as nature eventually reclaims the land.

To create these effects using our web extension, we drew the entire illustration first. Then, on a version of the webpage with revised text aligned to our reinterpretation, we created waypoints for each panel by adjusting how we wanted the image to sit on that page. Specifically, we wanted to match illustrated scenes to text blocks

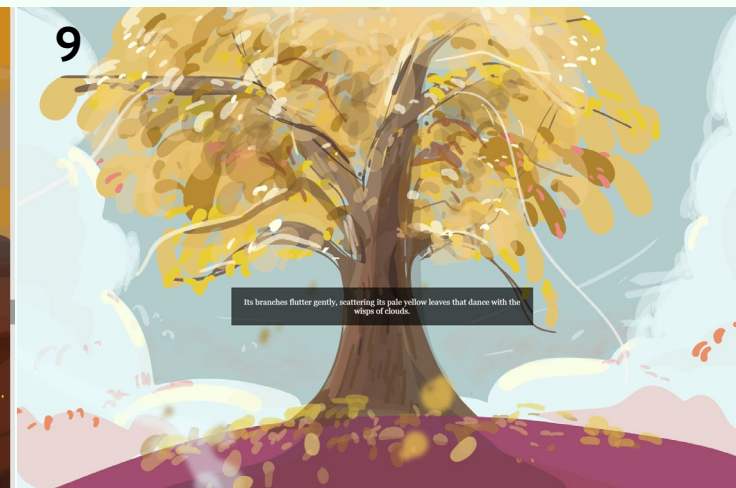
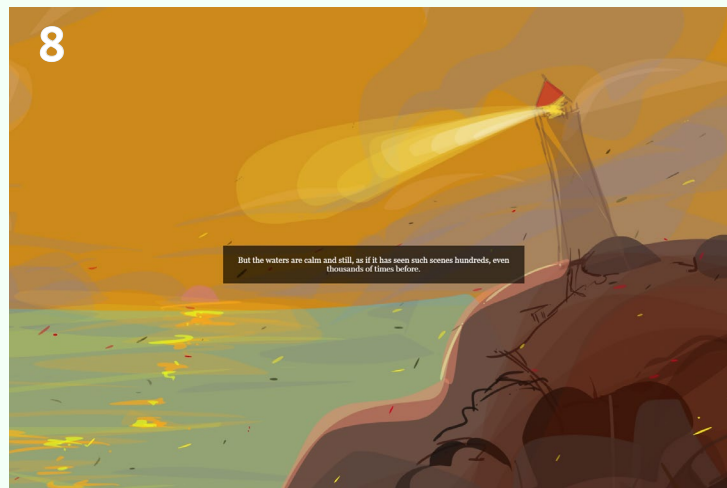
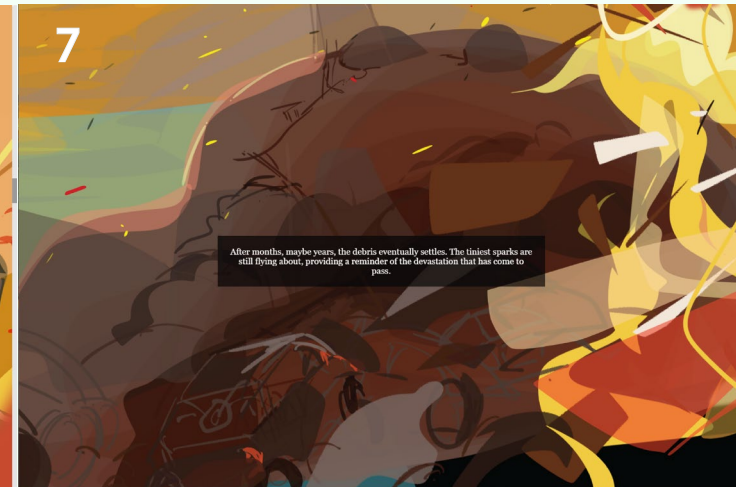


Figure 12. Panels 6-10 pans from the destruction to a lighthouse overlooking a horizon, which, upon further zooming in, shows the seed developing into a giant ginkgo tree.

that appropriately described it. This was achieved by (1) recording scroll interactions from one text block to the next and (2) linking the recorded interactions to the waypoints. The results are smooth interpolations that pan from one image panel to the next as the user scrolls down and reads each text block.

Discussion

The first gallery example showcases how users can use the web extension to contribute an alternative perspective via only additions to the page. The second example shows how the tool can also be used to completely reimagine a webpage to express a user's personal interpretation of an existing interactive article. We provide the following takeaways, derived from an autobiographical design perspective [35] from the authors' own anecdotal experiences using the tool.

Implications of Semi-Private Design

We previously drew analogies between the personalization of websites and graffiti, pointing to both of their abilities to evoke strong feelings of psychological connection [30]. However, one difference between the two is that personalization usually occurs in private settings while graffiti occurs in public ones. And like all public art that can be highly subjective, we assume not everyone will enjoy the remixes, despite how personally meaningful or engaging the narrative conveyed by the new design is. In such instances, the user simply has to turn *off* the bookmarklet. The availability of this option indicates that these digital graffiti essentially exist between public and private—in a *semi-private* space.

Understanding that users have autonomy over whether the bookmarklet is applied can change the context of authoring. For existing websites, there is less motivation to create remixes that visually “disrupt,” and more to enhance the web experience so others are more likely to use the bookmarklet. Thus, we argue that providing such opt-out options to shift outputs from the public to semi-private sphere can create a false sense of “competition” with the original web experience, leading to less potentially disruptive outputs.

Political & Ethical Dimensions of Digital Graffiti

While we draw inspiration from graffiti as a form of personal expression, physical graffiti has also historically functioned as a means of political and social

activism, challenging or questioning existent narratives. Extending this metaphor to digital spaces raises similar, though not entirely parallel, implications. Using SVGraffiti, users can overlay conflicting or potentially harmful content onto webpages, running the risk of misrepresenting or distorting the original author's intent. We consider this a double-edged sword. On one hand, web publishers typically hold the power to project their messages to a large audience. SVGraffiti enables those without an established web platform to still have a voice and contribute counter-narratives, similar to physical graffiti. On the other hand, these overlays introduce ethical concerns around authorship, consent, and misuse.

However, unlike their physical counterparts, the digital graffiti produced by SVGraffiti are not universally visible by default; they rely on the democratic process of individuals choosing to apply or share bookmarklets. This creates a more distributed, opt-in form of visibility, where the graffiti is only impactful and has reach when others deem it meaningful enough to adopt and propagate. While this does not fully prevent harmful or misleading digital graffiti from circulating, it introduces a form of collective filtering that shapes how and when such reinterpretations gain visibility.

Shaping Art Placement in Digital Spaces

The gallery examples were made with the expectation that they will be seen as part of a browsing session. While the illustrations themselves can also function standalone, a viewer will likely have a very different experience without the mouse clicks and scrolls. Prior literature has already explored this notion of “site-specific” artwork in the context of installation art [23, 45], where relocation would change the conveyed author's intent. This idea can be extended to the digital space, where how an artwork is placed on a webpage can fundamentally alter its meaning or interpretation.

In the process of creating the waypoints, we found that we made conscious choices to constrain navigation of the artwork in two ways. First, by nature of the

waypoints, only certain locations on the illustration are visible at all to a viewer. Second, these locations can only be assessed via specific mouse interactions on the website. The result is a heavily scripted experience that carries an artists' intent beyond the illustration itself to the website the work sits on. This deliberate scripting of art placement is beneficial in instances where we only want to portray a single perspective of an artwork. Even for art that does not need these constraints, we argue that there is space for future authoring systems to consider the context of how their outputs can be experienced after they leave the system.

Limitations & Future Work

SVGraffiti only supports a limited set of mouse events and color-based CSS transformations. Expanding the range of interaction types and visual modifications could broaden the expressive space of potential remixes with richer CSS/DOM state changes. SVGraffiti also assumes a single-user authorship model, when in practice, physical graffiti is frequently a collaborative task. Future work can explore how overlays involving multiple users may coexist, interact, and be negotiated on the same webpage. Lastly, beyond the current autobiographical process, future work can also involve external participants, including non-technical users, to better understand emergent practices of digital graffiti.

Conclusion

SVGraffiti is a web extension that empowers users to remix existing web experiences. By linking their dynamic illustrations to user interactions, they can externalize their interpretations directly onto webpages. These remixes reframe the web from a fixed container for content to a malleable medium that can be reinterpreted *post-hoc* through non-destructive overlays. We view this proof-of-concept as a step toward more participatory web interaction, where users not only consume content, but also actively contribute to its meaning, sense-making, and experience.

IMAGE CREDITS & PERMISSIONS

- [1] 'line drawing of man on the computer' is by Simple Line and licensed from Adobe Stock. The icons added to the illustration are licensed under CC0 1.0 from SVGRepo.
- [2] The photo of 'My God, Help Me to Survive This Deadly Love' is by Tony Webster on Flickr and is licensed under CC-BY-2.0. <https://www.flickr.com/photos/87296837@N00/15308169773>.
- [3] The photo of 'Balloon Girl' is by Miguel Discart on Flickr and is licensed under CC-BY-2.0. <https://flickr.com/photos/151639852@N07/52574400976>.
- [4] 'graffiti poster wall' is by Nazzu and licensed from Adobe Stock.
- [5] Screenshots of the UIST 2025 "Call for Participation" page are owned by ACM; this pictorial is published by ACM for the Designing Interactive Systems (DIS) Conference.
- [6] All other images not yet mentioned including diagrams, interface screenshots, and illustrations were made by the authors.

REFERENCES

- [1] Andy Adams. 2024. The New York Times Announces a New Series on Nuclear Threats. <https://outrider.org/news/new-york-times-announces-new-series-nuclear-threats>. [Accessed 19-03-2025].
- [2] Sérgio Alves, Ricardo Costa, Kyle Montague, and Tiago Guerreiro. 2024. Citizen-Led Personalization of User Interfaces: Investigating How People Customize Interfaces for Themselves and Others. *Proc. ACM Hum.-Comput. Interact.* 8, CSCW2, Article 446 (Nov. 2024), 23 pages. doi:10.1145/3686985
- [3] Mohammad Bajammal, Davood Mazinanian, and Ali Mesbah. 2018. Generating reusable web components from mockups. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering* (Montpellier, France) (ASE '18). Association for Computing Machinery, New York, NY, USA, 601–611. doi:10.1145/3238147.3238194
- [4] Banksy. 2002. Balloon Girl. Graffiti, London, UK.
- [5] Shaon Barman, Sarah Chasins, Rastislav Bodik, and Sumit Gulwani. 2016. Ringer: web automation by demonstration. *SIGPLAN Not.* 51, 10 (Oct. 2016), 748–764. doi:10.1145/3022671.2984020
- [6] Michael Bolin, Matthew Webber, Philip Rha, Tom Wilson, and Robert C. Miller. 2005. Automation and customization of rendered web pages. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (Seattle, WA, USA) (UIST '05). Association for Computing Machinery, New York, NY, USA, 163–172. doi:10.1145/1095034.1095062
- [7] John Branch. 2012. Snow Fall: The Avalanche at Tunnel Creek. <https://www.nytimes.com/projects/2012/snow-fall/index.html>
- [8] Brian Burg, Amy J. Ko, and Michael D. Ernst. 2015. Explaining Visual Changes in Web Interfaces. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) (UIST '15). Association for Computing Machinery, New York, NY, USA, 259–268. doi:10.1145/2807442.2807473
- [9] Yan Chen and Tovi Grossman. 2021. Umitation: Retargeting UI Behavior Examples for Website Design. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) (UIST '21). Association for Computing Machinery, New York, NY, USA, 922–935. doi:10.1145/3472749.3474796
- [10] Yan Chen, Sang Won Lee, and Steve Oney. 2021. CoCapture: Effectively Communicating UI Behaviors on Existing Websites by Demonstrating and Remixing. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (CHI '21). Association for Computing Machinery, New York, NY, USA, Article 416, 14 pages. doi:10.1145/3411764.3445573
- [11] Pei-Yu (Peggy) Chi, Sen-Po Hu, and Yang Li. 2018. Doppio: Tracking UI Flows and Code Changes for App Development. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (CHI '18). Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3173574.3174029
- [12] Allen Cypher and Daniel Conrad Halbert. 1993. *Watch What I Do: Programming by Demonstration*. MIT Press, Cambridge, Massachusetts.
- [13] Oscar Díaz, Cristóbal Arellano, and Maider Azanza. 2013. A language for end-user web augmentation: Caring for producers and consumers alike. *ACM Trans. Web* 7, 2, Article 9 (May 2013), 51 pages. doi:10.1145/2460383.2460388
- [14] Jorge Garza, Devon J. Merrill, and Steven Swanson. 2021. Appliance: Transforming Web Pages into Electronic Devices. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (CHI '21). Association for Computing Machinery, New York, NY, USA, Article 415, 13 pages. doi:10.1145/3411764.3445732
- [15] Siegfried Handschuh and Steffen Staab. 2002. Authoring and annotation of webpages in CREAM. In *Proceedings of the 11th International Conference on World Wide Web* (Honolulu, Hawaii, USA) (WWW '02). Association for Computing Machinery, New York, NY, USA, 462–473. doi:10.1145/511446.511506
- [16] W.J. Hennigan. 2024. At the Brink: Confronting the Risk of Nuclear War. <https://www.nytimes.com/interactive/2024/03/07/opinion/nuclear-war-prevention.html>
- [17] Joshua Hibsman and Haoqi Zhang. 2015. Unravel: Rapid Web Application Reverse Engineering via Interaction Recording, Source Tracing, and Library Detection. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) (UIST '15). Association for Computing Machinery, New York, NY, USA, 270–279. doi:10.1145/2807442.2807468
- [18] Joshua Hibsman and Haoqi Zhang. 2016. Telescope: Fine-Tuned Discovery of Interactive Web UI Feature Implementation. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) (UIST '16). Association for Computing Machinery, New York, NY, USA, 233–245. doi:10.1145/2984511.2984570
- [19] Forrest Huang, John F. Canny, and Jeffrey Nichols. 2019. Swire: Sketch-based User Interface Retrieval. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland UK) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–10. doi:10.1145/3290605.3300334
- [20] Zhicheng Huang and Jeff Huang. 2025. IRCHIVER: An Information-Centric Personal Web Archive for Revisiting

- Past Online Sensemaking Tasks. In *Proceedings of the 2025 ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR '25)*. Association for Computing Machinery, New York, NY, USA, 231–246. doi:10.1145/3698204.3716459
- [21] Hyunjin Kang and S. Shyam Sundar. 2013. Depleted egos and affirmed selves: The two faces of customization. *Computers in Human Behavior* 29, 6 (2013), 2273–2280. doi:10.1016/j.chb.2013.05.018
- [22] Tae Soo Kim, DaEun Choi, Yoonseo Choi, and Juho Kim. 2022. Stylette: Styling the Web with Natural Language. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 5, 17 pages. doi:10.1145/3491102.3501931
- [23] Helmut Leder, Benno Belke, Andries Oeberst, and Dorothee Augustin. 2004. A model of aesthetic appreciation and aesthetic judgments. *British journal of psychology* 95, 4 (2004), 489–508.
- [24] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. 2018. Evaluation Strategies for HCI Toolkit Research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (CHI '18). Association for Computing Machinery, New York, NY, USA, 1–17. doi:10.1145/3173574.3173610
- [25] Toby Jia-Jun Li, Amos Azaria, and Brad A. Myers. 2017. SUGILITE: Creating Multimodal Smartphone Automation by Demonstration. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI '17). Association for Computing Machinery, New York, NY, USA, 6038–6049. doi:10.1145/3025453.3025483
- [26] Yang Li, Xiang Cao, Katherine Everitt, Morgan Dixon, and James A. Landay. 2010. FrameWire: a tool for automatically extracting interaction logic from paper prototyping tests. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) (CHI '10). Association for Computing Machinery, New York, NY, USA, 503–512. doi:10.1145/1753326.1753401
- [27] Henry Lieberman. 2001. *Your Wish Is My Command: Programming by Example*. Academic Press, San Diego, California.
- [28] Sarah Lim, Joshua Hibschan, Haoqi Zhang, and Eleanor O'Rourke. 2018. Ply: A Visual Web Inspector for Learning from Professional Webpages. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) (UIST '18). Association for Computing Machinery, New York, NY, USA, 991–1002. doi:10.1145/3242587.3242660
- [29] Sonia Livingstone. 2007. The Challenge of Engaging Youth Online: Contrasting Producers' and Teenagers' Interpretations of Websites. *European Journal of Communication* 22, 2 (2007), 165–184. doi:10.1177/0267323107076768
- [30] Sampada Marathe and S. Shyam Sundar. 2011. What drives customization? control or identity?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) (CHI '11). Association for Computing Machinery, New York, NY, USA, 781–790. doi:10.1145/1978942.1979056
- [31] Richard G. McDaniel and Brad A. Myers. 1999. Getting more out of programming-by-demonstration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, USA) (CHI '99). Association for Computing Machinery, New York, NY, USA, 442–449. doi:10.1145/302979.303127
- [32] Raphael Menges, Steffen Staab, Christoph Schaefer, Tina Walber, and Chandan Kumar. 2025. What Did My Users Experience? Discovering Visual Stimuli on Graphical User Interfaces of the Web. *ACM Trans. Web* 19, 2, Article 11 (March 2025), 63 pages. doi:10.1145/3715881
- [33] Merriam-Webster. 2025. Definition of Graffiti. <https://www.merriam-webster.com/dictionary/graffiti>
- [34] Eric Mörth, Stefan Bruckner, and Noeska N. Smit. 2023. ScrollyVis: Interactive Visual Authoring of Guided Dynamic Narratives for Scientific Scrollytelling. *IEEE Transactions on Visualization and Computer Graphics* 29, 12 (2023), 5165–5177. doi:10.1109/TVCG.2022.3205769
- [35] Carman Neustaedter and Phoebe Sengers. 2012. Autobiographical design in HCI research: designing and learning through use-it-yourself. In *Proceedings of the Designing Interactive Systems Conference* (Newcastle Upon Tyne, United Kingdom) (DIS '12). Association for Computing Machinery, New York, NY, USA, 514–523. doi:10.1145/2317956.2318034
- [36] Stephen P. Norris, Linda M. Phillips, and Connie A. Korpan. 2003. University Students' Interpretation of Media Reports of Science and its Relationship to Background Knowledge, Interest, and Reading Difficulty. *Public Understanding of Science* 12, 2 (2003), 123–145. doi:10.1177/09636625030122001
- [37] Stephen Oney and Brad Myers. 2009. FireCrystal: Understanding interactive behaviors in dynamic web pages. In *Proceedings of the 2009 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (VLHCC '09). IEEE Computer Society, USA, 105–108. doi:10.1109/VLHCC.2009.5295287
- [38] Susan A. Phillips. 1996. Graffiti Definition: The Dictionary of Art. <https://www.graffiti.org/faq/graf.def.html>. [Accessed 01-04-2025].
- [39] The Pulitzer Prizes. 2013. John Branch of The New York Times. <https://www.pulitzer.org/winners/john-branch>. [Accessed 28-02-2025].
- [40] Alex Robinson. 2019. Sketch2code: Generating a website from a paper mockup. arXiv:1905.13750 [cs.CV] <https://arxiv.org/abs/1905.13750>
- [41] Nishant Sinha and Rezwana Karim. 2013. Compiling mockups to flexible UIs. In *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering* (Saint Petersburg, Russia) (ESEC/FSE 2013). Association for Computing Machinery, New York, NY, USA, 312–322. doi:10.1145/2491411.2491427
- [42] Kesler Tanner, Naomi Johnson, and James A. Landay. 2019. Poirot: A Web Inspector for Designers. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland UK) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3290605.3300758
- [43] Anja Tjärnhage, Ulrik Söderström, Ole Norberg, Mattias Andersson, and Thomas Mejtoft. 2023. The Impact of Scrollytelling on the Reading Experience of Long-Form Journalism. In *Proceedings of the European Conference on Cognitive Ergonomics 2023* (Swansea, United Kingdom) (ECCE '23). Association for Computing Machinery, New York, NY, USA, Article 6, 9 pages. doi:10.1145/3605655.3605683
- [44] Michael Toomim, Steven M. Drucker, Mira Dontcheva, Ali Rahimi, Blake Thomson, and James A. Landay. 2009.

- Attaching UI enhancements to websites with end users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) (CHI '09). Association for Computing Machinery, New York, NY, USA, 1859–1868. doi:10.1145/1518701.1518987
- [45] Vivian Van Saaze. 2013. *Installation Art and the Museum: Presentation and Conservation of Changing Artworks*. Amsterdam University Press, Amsterdam, Netherlands.
- [46] Dmitri Vrubel. 1990. My God, Help Me to Survive This Deadly Love. East Side Gallery, Berlin, Germany.
- [47] Jeremy Warner and Philip J. Guo. 2017. CodePilot: Scaffolding End-to-End Collaborative Software Development for Novice Programmers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI '17). Association for Computing Machinery, New York, NY, USA, 1136–1141. doi:10.1145/3025453.3025876
- [48] Jeffrey Wong and Jason I. Hong. 2007. Making mashups with marmite: towards end-user programming for the web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (CHI '07). Association for Computing Machinery, New York, NY, USA, 1435–1444. doi:10.1145/1240624.1240842
- [49] Shuhong Xiao, Yunnong Chen, Jiazhi Li, Liuqing Chen, Lingyun Sun, and Tingting Zhou. 2024. Prototype2Code: End-to-end Front-end Code Generation from UI Design Prototypes. arXiv:2405.04975 [cs.SE] <https://arxiv.org/abs/2405.04975>
- [50] Yeliz Yesilada, Robert Stevens, Simon Harper, and Carole Goble. 2007. Evaluating DANTE: Semantic transcoding for visually disabled users. *ACM Trans. Comput.- Hum. Interact.* 14, 3 (Sept. 2007), 14–es. doi:10.1145/1279700.1279704
- [51] Tongyu Zhou, Joshua Kong Yang, Vivian Hsinyueh Chan, Ji Won Chung, and Jeff Huang. 2024. Portallnk: 2.5D Visual Storytelling with SVG Parallax and Waypoint Transitions. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) (UIST '24). Association for Computing Machinery, New York, NY, USA, 16 pages. doi:10.1145/3654777. 3676376