Bachelor Project

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Web animation editor with motion tracking

Petr Kubeš

Supervisor: RNDr. Žára Ondřej
January 2021

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Kubeš Petr** Personal ID number: **474772**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Branch of study: **Computer and Information Science**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Web Animation Editor with Motion Tracking

Bachelor’s thesis title in Czech:

Editor animací se sledováním pohybu

Guidelines:

Familiarize yourself with the GIF file format as well as the general concept of 'motion tracking' within the context of a GIF animation.
Design and implement a web-based GIF animation editor. To differentiate from other commonly available tools, implement the following feature set:
- Existing GIF file format as an input and output (optionally WEBM). Loading and saving will be implemented as a client-side operation, i.e. there will be no server-side component.
- Users can add additional layers of content - text or static images; these layers can be optionally anchored to a moving parts of the main animation.
- Users can leverage a timeline UI control to adjust layer visibility.

Discuss the necessary Web API for this project. Implement the motion tracking feature in a way that does not block the main JavaScript CPU thread.
Apply a qualitative user testing session to the final product and define the set of supported web browsers.

Bibliography / sources:

[1] Nate Murray: Fullstack React, 2017 Fullstack.io, ISBN 0991344626
[2] https://developer.mozilla.org/en-US/
[3] https://www.w3.org/Graphics/GIF/spec-gif87.txt

Name and workplace of bachelor’s thesis supervisor:

RNDr. Ondřej Žára, Department of Computer Graphics and Interaction

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **18.09.2020** Deadline for bachelor thesis submission: **05.01.2021**

Assignment valid until: **19.02.2022**

RNDr. Ondřej Žára Supervisor’s signature	doc. Ing. Tomáš Svoboda, Ph.D. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

.
Date of assignment receipt

Student's signature

Acknowledgements

I would like to thank my family for motivating me to stay in school and my friends and co-workers to give me valuable feedback.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 5. January 2021

Abstract

This thesis's focus is the design, implementation, and testing of a web-based GIF animation editor. Its features include but are not limited to adding images and texts, a timeline that controls edits' visibility, and the option to motion track edits relative to the input GIF by anchoring the moving part. Users also have the option to use templates to create animations more easily and share finished GIFs.

Keywords: animation, motion tracking, react

Supervisor: RNDr. Žára Ondřej
todo

Abstrakt

Tato práce se zaměřuje na design, implementaci a testování webového editoru GIF animací. Jeho funkce obsahují mimo jiné možnost přidat obrázky a texty, časovou osu pro editaci viditelnosti přidaných prvků a sledování pohybu v rámci hlavní animace pomocí ukotvení k pohyblivé části. Uživatelé mají možnost použít šablony s předdefinovanými úpravami pro snažší tvorbu animace a sdílet výsledný GIF.

Klíčová slova: animace, sledování pohybu, react

Překlad názvu: Editor animací se sledováním pohybu

Contents

Chapter 1

Introduction

1.1 Motivation

Short animations have become a popular source of entertainment on many internet websites. Users love to share and create “memes” on websites such as Reddit and Facebook. However, the creation of GIFs with moving texts and images is currently quite difficult. Existing internet tools are not ideal because most of them support animation in a limited form or do not support it at all.

1.2 Goals

This project aims to provide a tool that will allow users to create animations with motion-tracked images and texts, which can then be downloaded, shared with others online, and exported to a template that other users could subsequently use.

Chapter 2

Analysis

2.1 Existing solutions

An internet search for “animation editor” yields only professionally focused solutions, which almost always require login and process videos on the server-side. The focus of this project is on the creation of short and most likely funny GIFs, so the following queries were inspected in the Google Ads keyword planner [8], which provided these monthly search result appearances:

Key word	Monthly searches
gif maker	450,000
gif memes	110,000
gif creator	110,000
gif editor	74,000
gif maker online	49,500
animated memes	14,800
add text to gif	12,100

Table 2.1: Search popularity of related keywords obtained from Google keywords planner.

The websites suggested by Google for the most popular queries have the following shortcomings.

Website	Add text and images	Animation	Notes
Ezgif.com	Yes	Technically yes	Allows editing of only one frame at a time, making creation of longer animations tedious
imgflip.com	Only images	No	
gifntext.com	Yes	Yes	The most similar tool. Does not support motion tracking and templates.
makeagif.com	Yes	No	
kapwing.com	Yes	Only animated effects	
animaker.com	Yes	Yes, but only predefined paths	

Table 2.2: Comparison with similar web tools.

There are also a few tools for creating GIFs and short videos for marketing purposes. However, these cannot be used since it is complicated to add animated text and images.

Other indicators of demand for a web based better GIF editor are the existence of many online communities such as `www.reddit.com/r/HighQualityGifs/`, `www.9gag.com/gif`, and Facebook groups in which users share GIF animations with added text. Usually, they track movement by using either professional tools such as Adobe After Effects or track by hand, resulting in a jerky movement.

Last but not least, there is an app called GifJif. It's only for iOS and enables matching faces to predefined GIF templates. The use case is very similar to this project and seems to be quite popular, being eighty-second in the entertainment category in AppStore despite being paid. [37]

2.2 Functionality description

2.2.1 Upload

The editing process starts by uploading an existing animation in a GIF format. This is possible either by supplying an URL address or by selecting it from the user computer's file system. The upload and consecutive decoding does not block the main thread, and the user sees a loading indicator.

2.2.2 Editor

Canvas

In the center is a canvas that displays the current animation frame. Edits - added texts and images - can be selected by clicking them, at which point all relevant controls appear. User can deselect the edit by clicking on part of the canvas with other or no edits.

Furthermore, edits can be resized by dragging corners and sides. Dragging by sides changes either width or height, whereas dragging by corners keeps the proportions. Edits can also be rotated by dragging a handler above them and moved by clicking any inner part of the edit and dragging it to the desired location. All of these operations are performed in real-time.

If an edit is animated, an animation trace is shown to give the user a better idea about the element's path.

It is possible to add white space on top of the animation to add a title.

If the user enables motion tracking, an anchor for marking the tracing target is shown.

Edit box

Edit box is a window that shows relevant settings for a particular edit. These include:

- Animation controls:
  - Buttons for motion tracking edit across single a frame or automatically throughout the whole animation
  - Overview of all positions in each frame. These can be removed either individually or all at once.
- Button for centering an edit in case the user loses track of it.
- Text controls: (Applies only to text edits)
  - Font picker
  - Color picker for text itself
  - Color picker for text border

- Input element for border width
- Background color picker
- Input element for padding size of the edit's background.

Timeline

Timeline is an element used for displaying the visibility of all added edits. Each edit can be adjusted to be visible only on a certain time span by dragging its start and end to desired frames. Users can also change the order (z-index) of each so that some are drawn above others and vice versa. It also contains UI for shortening the length of the animation as a whole.

History

Users can revert their latest actions in case of accidental edits.

2.2.3 Templates

Templates are a list of pre-edited popular GIFs whose purpose is to simplify the creation of animation. The user can then choose a template, fill in his custom texts, and export a finished animation without going through the motion tracking process.

2.2.4 User accounts & GIF storage

Since animations' main purpose is to share them with others, the project provides an option to share them immediately online. This is done by first signing into the website through Google SSO and then clicking the upload button after exporting. If the upload succeeds, an URL with the address of the animation is shown.

2.3 Tools

2.3.1 Language & Compilation

TypeScript

TypeScript is a programming language that trans-compiles into JavaScript [25]. Its main advantage is support for static typing, which can improve the program's overall stability. The author's preference is to use type-safe, functional programming languages, so other alternatives such as Kotlin/JS [15], and ReScript [22] were considered; however, these do not have as large a user-base as TypeScript [23], so potential problems would be harder to solve due to lack of resources.

TypeScript's disadvantage over plain JavaScript is the necessity to rebuild the project after every change. This was not considered an issue since in order to support older browsers with modern JavaScript features [6], a compilation is needed anyway.

Babel

This project uses Babel as a JavaScript compiler to expand the set of supported web browsers. [27] The reason why compilation is necessary is shown below. Let's consider a JavaScript arrow function. That is a feature added to JS version ES2015 and therefore is not supported by older browsers. Babel executes the following transformation:

```
// Babel Input: ES2015 arrow function
[1, 2, 3].map((n) => n + 1);

// Babel Output: ES5 equivalent
[1, 2, 3].map(function(n) {
  return n + 1;
});
```

Listing 2.1: Babel transformation

Babel can also be used for TypeScript compilation. The main shortcoming is that Babel, unlike TypeScript compiler (tsc), cannot do type-checking at compile time. Therefore a hybrid approach is used, in which types are first checked by tsc and babel generates source-code afterward. [10]

Webpack

Reusing code from multiple JavaScript files is not a trivial task. Let's consider the following example:

```
// test.js
function test() { alert("Testing") }

// main.js
test()

// index.html
<html>
    <head>
        <script src="test.js"></script>
        <script src="main.js"></script>
    </head>
    <body></body>
</html>
```

This works, but having all functions and classes accessible globally can lead to conflicts. That can cause unexpected bugs. The problem is addressed by ES6 modules, that can be used like this:

```
// test.js
export function test() { alert("Testing") }

// main.js
import { test } from './test'
test()

// index.html
<html>
    <head>
        <script type="module" src="main.js"></script>
    </head>
    <body></body>
</html>
```

ES6 `import` syntax is unfortunately supported by only 92 % of browsers [6], so a js bundler has to be used to improve the browser support. What Webpack does, is that it bundles all JS files in the project into one or just a few `bundle.js` files. Those contain all of the code of the project while handling the issue of global conflicts and using ES5 syntax. [9]

This project also uses Webpack's support for loading images and web workers, which improves readability by including both images and workers in the same folders as the code that uses them.

2.3.2 APIs

Media Recorder

MediaRecorder API provides functionality to record media streams from canvas. Example of the API usage is shown below:

```
const options = {
    mimeType: 'video/webm',
    videoBitsPerSecond: 1750000
}
const stream = canvas.captureStream()
const mediaRecorder = new MediaRecorder(stream, options)
const recordedBlobs: Blob[] = []

mediaRecorder.onstop = () => {
    getResultCallback(
        new Blob(recordedBlobs, {type: 'video/webm'})
    )
}
mediaRecorder.ondataavailable = (event: BlobEvent) => {
    recordedBlobs.push(event.data)
}

mediaRecorder.start()
setTimeout(() => mediaRecorder.stop(), 5000)
```

Listing 2.2: Media Recorder API

Its main advantage is that it allows creating WebM and mp4 videos directly from canvas and no libraries, such as FFMPEG, have to be used for the task. However, `MediaStream` object, which is obtained by calling `canvas.captureStream()`, and its `MediaStreamTrack` do not provide methods for supplying custom data, so the whole animation has to be played in the desired speed to record it.[28] This causes issues on older computers, on which the animation smoothness depends on the machine's performance capabilities.

Webworker

Web worker API is used for running scripts in the background. [1] This allows performing expensive tasks without blocking user interface. Example usage is shown below:

```
// main.js
const worker = new Worker('worker.js');
worker.postMessage("test value");
worker.onmessage = (e) => {
    console.log('Received result: ' + e.data)
}

// worker.js
onmessage = (e) => {
    const result = intensiveComputation(e.data)
    postMessage(result)
}
```

2.3.3 Libraries

React

React is a front-end JavaScript framework used for declaratively creating interactive user interfaces. Developer designs components of the app using jsx, or in this project, tsx, which is a syntax extension of JavaScript/TypeScript for creating React elements. These components can then be used within each other, and the React framework takes care of efficiently re-rendering each one only when its data change. [20]

Redux

Redux is a library for managing the state of JavaScript applications. Its main component is a "store," which contains the application's state, action dispatcher, and listeners. The state is changed by first dispatching an action processed by a pure function called reducer, that returns a newly updated state. Finally, listeners notify the UI that the state has changed. [21]

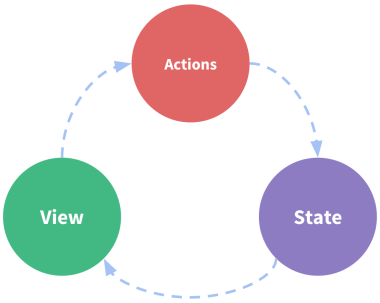


Figure 2.1: State manipulation with Redux

Its main disadvantage is a verbose interoperability with TypeScript. In order to add a new type-safe action, the following code has to be added: [26]

```
// actionTypes.ts
export const NEW_ACTION = ’NEW_ACTION’

interface NewAction {
    type: typeof NEW_ACTION,
    payload: String
}

export type ActionTypes = NewAction

// actions.ts
import { NEW_ACTION, ActionTypes }

export function newAction(message: String): ActionTypes {
  return {
    type: NEW_ACTION,
    payload: message
  }
}
```

For example, the same result would be achieved in Kotlin by a single line of code.

```
data class NewAction(val payload: String)}
```

Rambda

Rambda is a library that provides a set of pure functions to simplify functional programming in JavaScript. [18] This project uses mainly its functions for immutable array manipulation.

Styled Components

Styled Components is a library that allows creating React components with styles directly rather than inserting styles through classes or the `style` attribute. [24] This can make the code more readable, as shown below:

```
render() {
    <div className="container">
        <button className="primary-button">Click here</button>
    </div>
}
```

Listing 2.3: Using CSS classes

```
render() {
    <Container>
        <PrimaryButton>Click here</PrimaryButton>
    </Container>
}
```

Listing 2.4: Using Styled Components

It also supports passing of `props` that can dynamically change CSS properties. For example, this is how the row of edit's timeline can be dynamically updated:

```
const Timeline = styled.div<{ row: number }>‘
  grid-row: ${props => props.row};
  ...other css styles
‘

// Then in render()
<Timeline row={rowIdx}/>
```

Listing 2.5: Edit's timeline using Styled Components

React-color

A library that provides a variety of color pickers compatible with React. [19]

2.3.4 GIF

GIF is a bitmap image format supporting animations. The contents of a GIF file are described in the following table:

Block	Description
Header	This is the first block in the GIF format. It contains the GIF signature, which simply states "GIF" and a GIF version that can be "87a" or "89a".
Logical Screen Descriptor	Its purpose is to contain settings common for the whole animation. This includes Logical screen width, height, and Global Color Table options.
Global Color Table (Optional)	Images without a Local Color Table use the Global Color Table. Its capacity is up to 768 bytes resulting in 256 RGB colors.
Image Descriptor	Each image in the data stream contains such descriptor containing image's width, height, a position relative to the Logical screen, an indicator whether Local Color Table is used and its parameters, and an Interlace Flag.
Local Color Table (Optional)	Same as Global Color Table but only relevant for the following image.
Table Based Image Data	The actual data of the GIF image encoded using the LZW algorithm. Each pixel value must be in the range of the active color table.
Graphics Control Extension (Optional)	This block was added in the 89a GIF version. It specifies properties of transition between images, most notably the delay time.
Comment extension (Optional)	This extension is used for storing human-readable data, which are not part of the image. This could be used in the future to store information about the GIF being created by this tool.
Plain Text Extension (Optional)	Allows including text as part of the GIF. However, it does not support all the features (such as font setting) required by this project; therefore, it is not used.
Application Extension (Optional)	Used for including application-specific information and was not used in this project.

Table 2.3: GIF format description [2]

Omggif.js

Omggif is a small library for encoding and decoding GIFs in the GIF89a format. [30]

gif.js

The challenge with adding edits to GIF animations is that the added edits can use colors not present in the color palette of the GIF. The following solutions were considered:

- Replace the edit colors with the most similar colors from the GIF palette:
  - Pros: No need for external libraries and relatively easy implementation
  - Cons: This could result in dramatically different results in cases when the color palette is not similar enough, especially for image edits.
- Write an algorithm for quantization:
  - Pros: Lowest project size, more control over the code
  - Cons: Implementing K-means or Kohonen's neural network from scratch would not be a productive work on the project as these algorithms are already implemented in other libraries.
- Use color quantization library and encode quantized frames using Omggif.
  - Pros: Robust solution that can withstand varying colors in GIF frames and edits. Reasonable control over the export process.
  - Cons: The export process would be depended on multiple libraries.
- Use gif.js library for GIF exporting.
  - Pros: Solutions tested by thousands of users, already implemented support for parallelization through web workers. [33]
  - Cons: Code for GIF encoding would be included twice in the codebase since it is part of omggif.

The third solution was chosen to ensure the most stable export. The gif.js library uses the NeuQuant Neural Net image quantization [33] algorithm, which provides the lowest errors and faster speeds compared to other techniques such as SVM, Oct-Tree, and Median Cut. [31]

2.3.5 Services

Firebase

Firebase is a platform by Google, which provides a variety of backend services for app development. Firebase SDK can be used for easy integration. [3] Services relevant for this project are the following:

- Authentication - Firebase creates a database of users, which can be accessed using a simple admin interface. The admin can perform actions such as deleting accounts, disabling accounts, and resetting passwords. Moreover, authentication is integrated with Firestore and storage services so that Firebase Security Rules can be set to prohibit users from deleting, modifying, and reading other users' data. On the frontend, an SDK function can be invoked for signing users in using Google SSO, email, or other SSO providers.

- Cloud Firestore - Firestore is a NoSQL Document based database hosted in Google cloud that scales with demand. The Firebase SDK contains functions for asynchronously inserting and querying data.

- Storage - Firebase storage is a backend service for storing and retrieving files.

- Hosting - Firebase hosting is a fully-managed hosting service that allows developers to push production build of an application using a single command, takes care of delivering the content over SSL without any configuration, caching using CDN, and a simple DNS setup.

- Analytics - Tool for monitoring the traffic, users and optionally custom events.

Sentry.io

Sentry is a tool for monitoring crashes in an application. For each crash, a report is created, including various data points that can solve the underlying issue. [5]

Chapter 3

Implementation

3.1 Design and User interface

The most well-known video editing software - Adobe Premiere Pro and Final Cut Pro by Apple - inspired the design and user interface. Positions of the main editor elements are similar for all tools. The window with the current animation frame is in the middle. Below is a timeline with all added edits, and on top of it are controls for the animation playback. Controls for individual edits can be found on either left or right of the main screen.

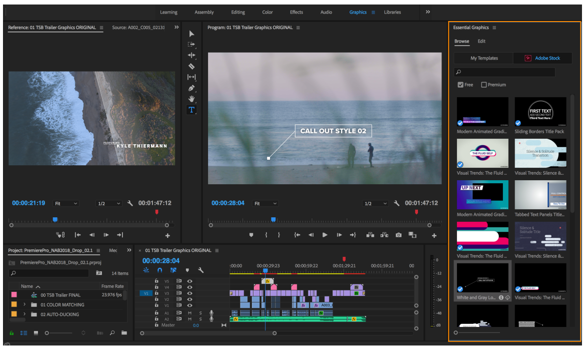
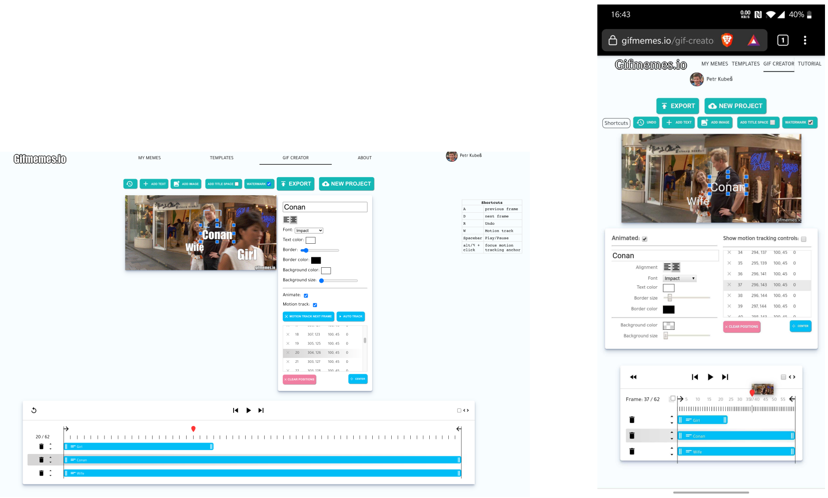


Figure 3.1: Adobe Premiere Pro [4]

Material design icons [16] were used to make the design more intuitive for users who are used to Google products. The website is responsive, but the UI was not optimized for phones; hence only creating animations from templates is usable.

Figure 3.2: Final Cut Pro [7]



(a) : Desktop **(b) :** Mobile

Figure 3.3: Design

3.2 Codebase

3.2.1 Organization

The code is organized as shown below:

```
config
├── scripts
├── public
└── src
    └── components
        ├── about
        ├── detail
        ├── editor
        └── my-memes
```

```
|   |__templates
|   |__user
|__firebase
|__redux
|   |__actions
|   |__reducers
|__styles
|__utils
```

- config, scripts - Folders containing scripts and configurations generated by the create-react-app library and were not changed for this project.

- public - Folder for publicly accessible assets. This contains files such as `index.html`, `robots.txt` and `favicon.ico`. These files are further processed during the build, which allows dynamically changing some parameters.

- components - React components organized by feature. That is, components are dependent only on components that are on the same level or deeper in the directory structure.

- firebase - Helper functions to simplify working with the Firebase SDK.

- redux - Folder for actions and reducers for the Redux library.

- styles - CSS styles that are shared across components

- utils - Various functions simplifying common tasks.

3.2.2 Upload

The GIF data is decoded using the omggif [30] library. The data is then further processed. First, transparent parts of the frame are filled with data from the previous frame. This is necessary for proper displaying of frames when a user selects them out of order. The second part of processing is saving each frame as `HTMLCanvasElement` so that the data can be put into canvas using `context.drawImage()` instead of `context.putImageData()`. The former is more performant.[17]

3.2.3 Edits

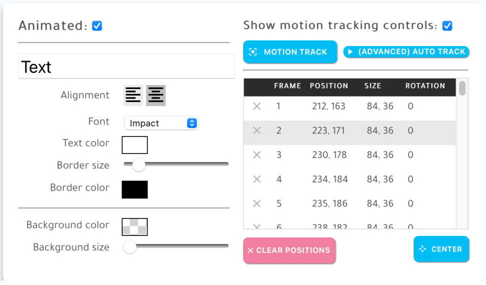
There are two types of edits - text edit and image edit. The state of all edits is stored in the Redux state in the following data structures:

```
interface Position {
    x: number,
    y: number,
    rotation: number,
    width: number,
    height: number,
}
interface Move extends Position{ frame: number }
interface MoveWithAnchor extends Move {
    anchorPosition: Position
}
type PositionState = Position | Move[] | MoveWithAnchor[]

interface Edit{
    id: number,
    moves: PositionState,
    startFrame: number,
    endFrame: number
}

interface TextEdit extends Edit {
    ...textEditOptions
}

interface ImageEdit extends Edit {
    img: string
}
```

Listing 3.1: Edits data types

Notable is the storing of images as `base64` string directly in the state. This makes reusing templates with images trivial since images do not have to be handled separately.

PositionState is one of three data types: `Position` for edits that are not animated, a list of `Moves` for animated edits without anchors and a list of `MoveWithAnchor` for motion-tracked edits. The anchor position needs to be saved for each move because having the anchor position saved separately confused some users.



Animated: ☑

Text

Alignment

Font: Impact

Text color: □

Border size

Border color: ■

Background color: ▨

Background size

Show motion tracking controls: ☑

	FRAME	POSITION	SIZE	ROTATION
×	1	212, 163	84, 36	0
×	2	223, 171	84, 36	0
×	3	230, 178	84, 36	0
×	4	234, 184	84, 36	0
×	5	235, 186	84, 36	0
×	6	238, 182	84, 36	0

Figure 3.4: Edit settings window: Text edit options on the left, list of animation positions on the right

Edit manipulation

Edits can be dragged, resized, and rotated. This can be achieved using a canvas drawing library like KonvaJs. [14] However, since image manipulation is a crucial part of this project and some features that might be useful are not supported, image manipulation was implemented from scratch.

All manipulations are Redux actions containing the x and y coordinates of the mouse. Then, the reducer computes the new size and position of the currently manipulated edit.

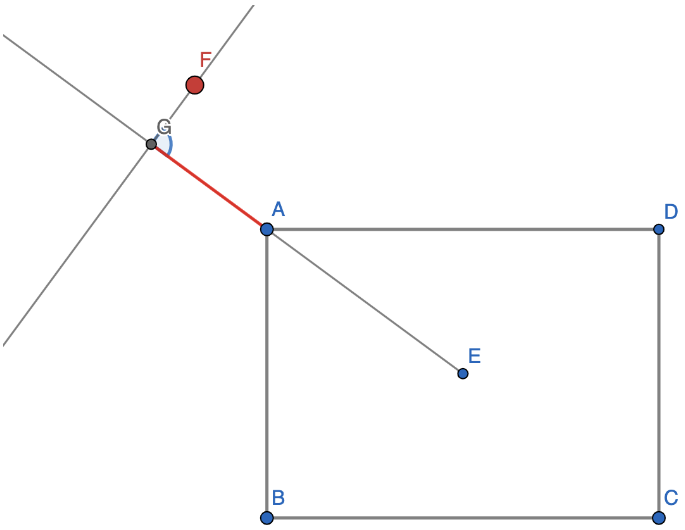


Figure 3.5: Diagonal resize. Point F is the mouse position and rectangle ABCD is the edit.

Horizontal and vertical resizing is implemented as follows:

1. The mouse vector($\vec{F}$) is rotated in the opposite direction of edits rotation around the edits center($\vec{E}$): $\vec{F_r} = M(\vec{F}-\vec{E})+\vec{E}$, where M is the rotation matrix.
2. The new width and height is two times the x and y component of the result vector: $\vec{result} = \vec{F_r} - \vec{E}$.

In order to calculate new dimension in case of diagonal resize, we need to find the vector $\vec{G}$. This is achieved using the following algorithm:

1. Set the origin of the system to center of the edit: $\vec{E} = (0, 0)$
2. Create a unit vector $\vec{u}$, such that $\vec{u} = \frac{\vec{A}}{||\vec{A}||}$, where $\vec{A}$ is the dragged corner.
3. Then $\vec{G} = (\vec{u} \cdot \vec{F})\vec{u}$

4. Finally, adjust width and height according to components of the result vector: $\vec{result} = \vec{G} - \vec{A}$

Drawing

HTMLCanvas provides an API for drawing images with chosen dimensions and locations. Rotated texts and images can be drawn by first translating canvas origin to the center of the edit, then rotating the whole context, and finally drawing the image or text.[13]

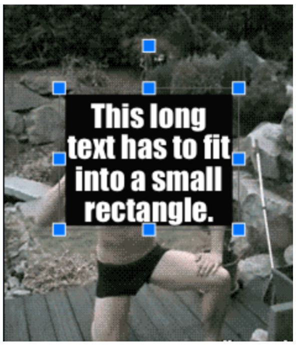
Text Fitting

Text is drawn into a rectangle which size is determined by the user. The goal is to fill as much of the user-defined space as possible, which means finding the line breaks and the biggest possible text size. Moreover, the text size is browser dependent; that is, each browser renders text with set size differently. To solve this issue, it is necessary to use canvas's function `measureText()` and find the best fit using binary search as described below:

```
let textSize = 512
let step = -textSize / 2
const words = edit.text.trim().split(" ")

while (true) {
    textSize += step
    lines = []
    // Find how many words fit in each line
    for (word in words) {
        const potentialLine = lines[lastIdx] + " " + word
        if (fits(ctx, potentialLine, editWidth)) {
            lines[lastIdx] = potentialLine // Add to line
        } else {
            lines.push(word) // Start a new line
        }
    }
    // Check, whether text fits boundaries
    if (abs(step) < 0.5 && notTooHigh(ctx, lines, height)) {
        break
    } else if (tooBig(ctx, lines, width, height)) {
        step = - abs(step / 2)
    } else if (tooSmall(ctx, lines, width, height)) {
        step = abs(step / 2)
    } else {
        break
    }
}
```

Listing 3.2: Algorithm for text fitting

The result is cached to prevent unnecessary recalculations.



This long text has to fit into a small rectangle.

Figure 3.6: Text fitting in action

3.2.4 Motion tracking

The user workflow consists of selecting a rectangular anchor and clicking the “motion track” button, which triggers a search for the closest match in the subsequent frame. This presents a challenge for a) choosing the matching measure and b) choosing an algorithm that will search the space.

Measure function

Since the optimal matching measure depends on the type of data [35], the following matching measures were tested:

- Sum of squared differences (SSD)

$$s = \sum_{(u,v)\in \boldsymbol{I}} (\boldsymbol{I_1}[u,v] - \boldsymbol{I_2}[u,v])^2$$

- Normalized Cross-Correlation NCC

$$s = \frac{\sum_{(u,v)\in \boldsymbol{I}} \boldsymbol{I_1}[u,v] \cdot \boldsymbol{I_2}[u,v]}{\sqrt{\sum_{(u,v)\in \boldsymbol{I}} \boldsymbol{I_1}[u,v]^2 \cdot \sum_{(u,v)\in \boldsymbol{I}} \boldsymbol{I_2}[u,v]^2}}$$

- Zero Mean Normalized Cross-correlation (ZNCC)

$$s = \frac{\sum_{(u,v)\in \boldsymbol{I}} (\boldsymbol{I_1}[u,v] - \overline{\boldsymbol{I_1}}) \cdot (\boldsymbol{I_2}[u,v] - \overline{\boldsymbol{I_2}})}{\sqrt{\sum_{(u,v)\in \boldsymbol{I}} (\boldsymbol{I_1}[u,v]^2 - \overline{\boldsymbol{I_1}})) \cdot \sum_{(u,v)\in \boldsymbol{I}} (\boldsymbol{I_2}[u,v]^2 - \overline{\boldsymbol{I_2}}))}}$$

Where:

- s – the difference measure
- $\boldsymbol{I}$ – the anchor

- $\boldsymbol{I_1}$ – the anchor cutout from template
- $\boldsymbol{I_2}$ – the anchor cutout from subsequent frame

The following experiment was performed to determine the ideal matching measure: Three types of animations were selected. For each, a series of 10 frames was motion-tracked five times with the help of the user. That is, for each incorrectly found match, a manual adjustment was performed to simulate real-world usage. The number of corrections necessary for each frame is presented below:

Frame/Measure	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
SSD	0	0	0	3	0	0	0	0	5	0
ZNCC	0	0	0	5	0	0	0	5	5	0
NCC	0	0	0	5	1	1	0	4	5	0

Table 3.1: Animation: Person jumping into a pool, frames: (8-18)

Frame/Measure	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
SSD	0	0	5	0	0	0	0	0	0	0
ZNCC	0	0	4	0	0	0	0	0	0	0
NCC	0	0	5	0	0	0	0	0	0	0

Table 3.2: Animation: Coffin dance, frames: (24-34)

Frame/Measure	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
SSD	0	0	0	4	0	0	0	0	5	0
ZNCC	0	5	0	0	0	0	5	5	1	2
NCC	0	3	0	2	1	0	5	5	2	5

Table 3.3: Animation: Kid knocking, frames: (10-20)

Measure	**Total number of corrections**
SSD	17
ZNCC	37
NCC	44

Table 3.4: Total number of user corrections

This experiment suggests that SSD is a more accurate matching measure while being the cheapest to compute. More rigorous testing could have been performed, but that is out of the scope of this thesis. Other techniques that could have been examined including using different color spaces for ZNCC [36], optical flow [29], or approaches based on neural networks.[32] Histogram based matching measure was also tested but turned out to be completely unsuitable for GIFs.

Search algorithm

To search the whole image and computing SSD is too expensive and takes too long even on modern hardware, making the motion tracking less pleasant to use. However, in some cases, we can observe that the score gradually improves as the anchor approaches the optimal solution.

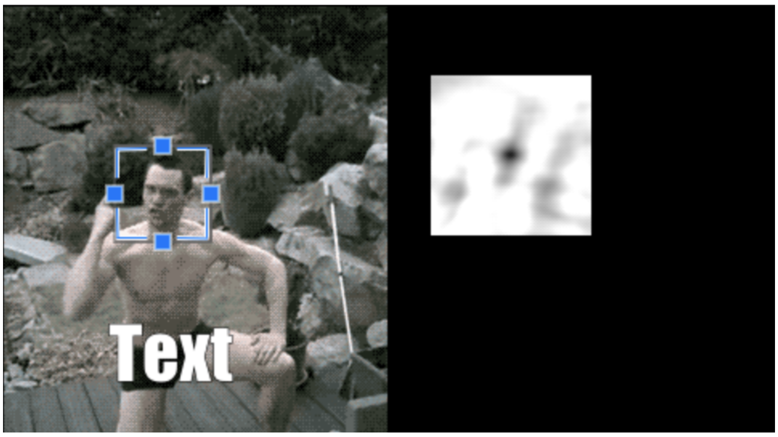


Figure 3.7: The white square is the searched area. Inside it, the darker the color, the better fit.

The gradient near the optimal solution suggests that the algorithm could be improved using a greedy search with a limit that stops the search after a certain number of non-improving positions.

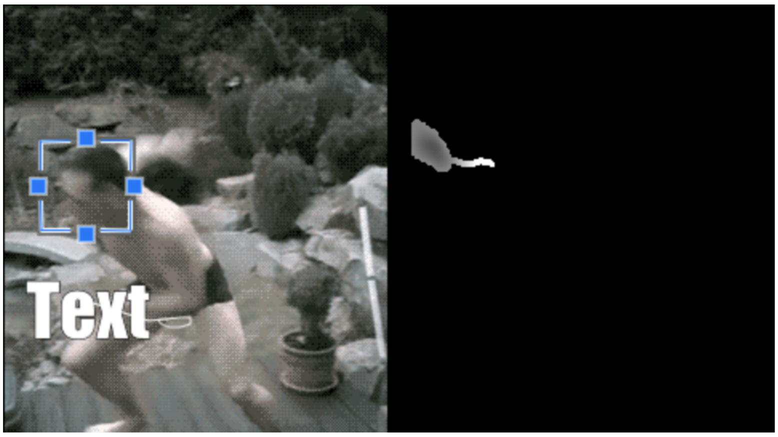


Figure 3.8: Greedy search

In some cases, this works as intended and improves the speed significantly but is susceptible to getting stuck in a local optimum. To overcome it, a grid of initial positions has been added to the greedy search queue. This proved especially effective for animations in which the positions in-between frames considerably varied.

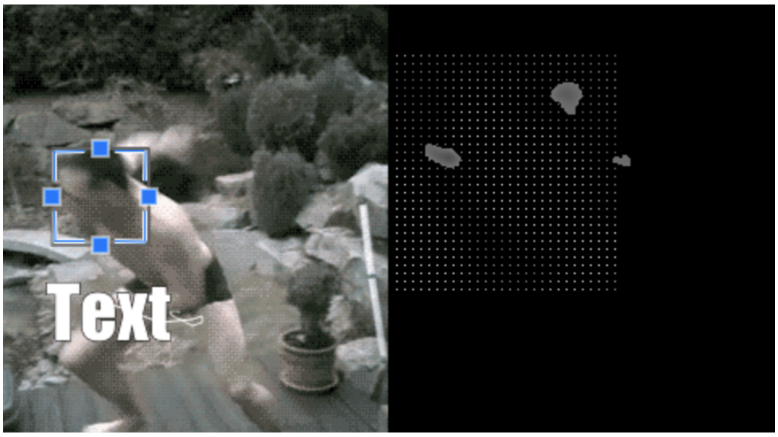


Figure 3.9: Greedy search with pre-set positions.

Additional speed improvements could have been made by parallelization and web assembly, but the current algorithm is fast enough for the project's purposes.

Auto-Tracking

It might be useful to motion-track multiple frames in a row with just a single click. The implementation is as follows:

1. When auto-tracking starts, the state is initialized:

$$n := 0$$

$$movingAvg := 0$$

2. The tracking function is called and returns a *score* (sum of squared differences).

3. If $score > movingAvg \cdot THRESHOLD$ and $n \neq 0$ then stop auto-tracking.

4. Otherwise, the state is updated and tracking continues:

$$movingAvg_{n+1} := movingAvg_n + \frac{score - movingAvg_n}{n+1}$$

$$n := n + 1$$

The *THRESHOLD* was chosen to be 1.5. In other words: If the score is worse than the previous frames by more than 50 %, stop auto-tracking.

3.2.5 History

Redux architecture allows implementing history directly from the redux state. The history reducer saves history by creating a shallow copy of the edits' state and saves it into a stack along with the current frame index. This is triggered by any action that should be saved in history, such as adding, removing, moving, and other edit manipulations. When undo is clicked, the edits' state and frame index from the top of the stack are popped and applied to the editor state.

Continuous changes of edits, including dragging in canvas, dragging of range inputs, and color changes, require a special history saving action because keeping intermediate states is not desired.

3.2.6 Timeline

The timeline is implemented using the CSS grid layout. Each of the frame indicators occupies one cell in the grid, whereas each edit's duration indicator has CSS properties `grid-column-start`, `grid-column-end`, and `grid-row-start`. This allows dynamically resizing the timeline to create space for edits spanning only a few frames.

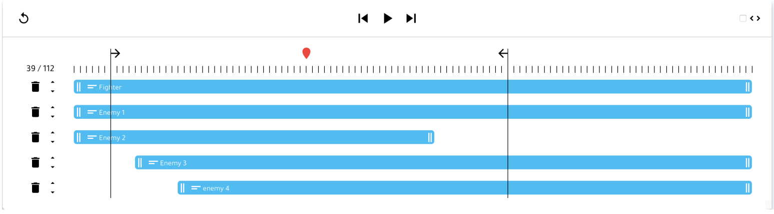


Figure 3.10: Timeline with multiple edits.

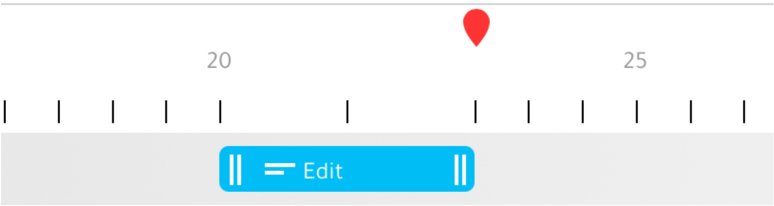


Figure 3.11: Dynamic timeline resize

The dragging of edits, current frame red pin, and video length barriers is computed as follows:

1. Total with of the timeline is measured.

2. Width of one column is approximated by dividing the total timeline width by the number of frames

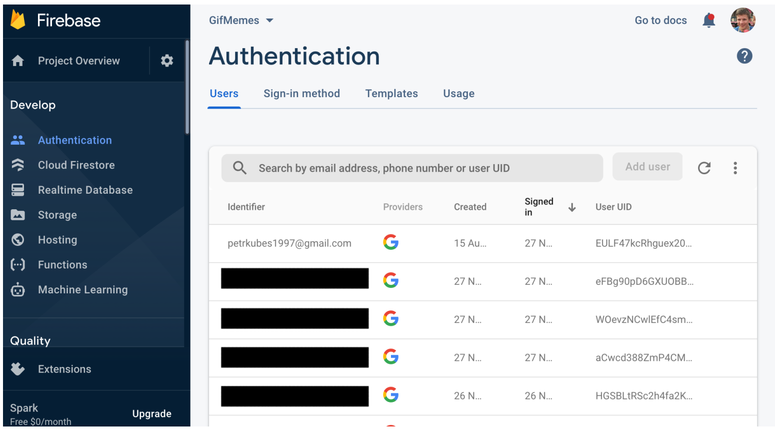
3. The X position of the cursor is measured relative to the timeline element

4. New column is calculated by dividing the cursor's X position by column width.

This approach is not perfect since each column can have different width, but it is noticeable only for unusually resized timelines. To overcome it, the width of each column would have to be measured.

3.2.7 User account

Accounts are fully managed by Firebase. The SDK provides methods and callbacks for login, logout, and information about the current user. Users' management can be accessed from the developer console.



Identifier	Providers	Created	Signed in	User UID
petrkubes1997@gmail.com	G	15 Au...	27 N...	EULF47kcRhguex20...
[redacted]	G	27 N...	27 N...	eFBg90pD6GXUOBB...
[redacted]	G	27 N...	27 N...	WOevzNCwlEfC4sm...
[redacted]	G	27 N...	27 N...	aCwcd388ZmP4CM...
[redacted]	G	26 N...	26 N...	HGSBLtRSc2h4fa2K...

Figure 3.12: User management in Firebase

3.2.8 Saving data on the backend

Templates allow users to reuse animation edits created by other users. This requires storing the original GIF, preview, and edits' state in firebase. The file storage structure is as follows: `[preview|templates|exported]/userId/data`. Preview is for `jpeg` images of previews, `templates` folder contains original unedited GIFs for templates and `exported` contains GIFs that were shared through url.

The rules are set up in such way, that only the user that created the file can delete it, while everyone else can read them.

```
rules_version = '2';
service firebase.storage {
  match /b/{bucket}/o {
    match /templates/{userId}/{templateId} {
      allow read;
      allow write: if request.auth.uid == userId;
    }
    match /previews/{userId}/{templateId} {
      allow read;
      allow write: if request.auth.uid == userId;
    }
    match /exported/{userId}/{allPaths=**} {
      allow read;
      allow write: if request.auth.uid == userId;
    }
  }
}
```

Listing 3.3: Firebase security rules

For each image, response header was set to `Cache-Control: public, max-age=604800, immutable` to prevent unnecessary db reads.

The rules for reading templates data from Firestore are set up identically.

3.2.9 App deployment

A domain `gifmemes.io` was purchased and set up with Firebase hosting service. The deployment is as simple as specifying the `index.html` in `firebase.json` and running `firebase deploy` command.

Chapter 4

Testing

4.1 Browser compatibility

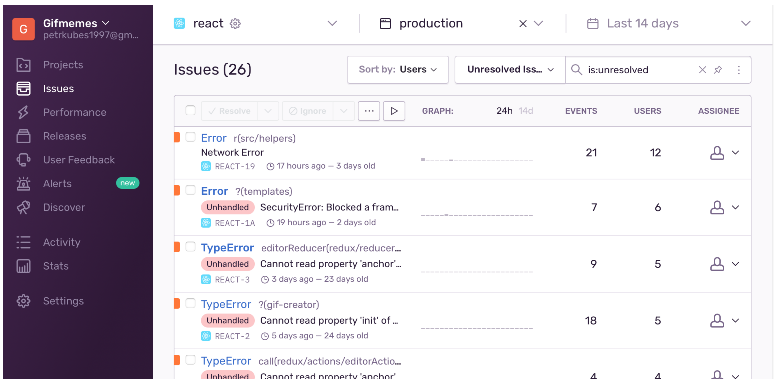
All features were tested in the most recent versions of modern browsers.

Browser	Issues
Safari, 14.0.2	MediaRecorder not supported and therefore exporting WEBMs is not possible
Chrome, 87.0.4280.88	No issue were found
Firefox, 84.0	MediaRecorder records a start delay
Opera 73.0.3856.284	No issues were found
Edge 87.0.664.66	No issues were found

Table 4.1: Browser compatibility

4.2 Sentry.io

Sentry was used to find crashes that occur in environments not available during testing or while performing unusual workflows. In the 7-day long time period from December 10th to December 16th, the website had 315 user sessions, 13 unique errors, 58 error events, happening to 31 unique users. Most of the issues were fixed.[5]



Issues (26)

	GRAPH: 24h 14d	EVENTS	USERS	ASSIGNEE
Error r(src/helpers) Network Error REACT-19 17 hours ago – 3 days old		21	12	
Error ?(templates) Unhandled SecurityError: Blocked a fram... REACT-1A 19 hours ago – 2 days old		7	6	
TypeError editorReducer(redux/reducer... Unhandled Cannot read property 'anchor'... REACT-3 3 days ago – 23 days old		9	5	
TypeError ?(gif-creator) Unhandled Cannot read property 'init' of ... REACT-2 5 days ago – 24 days old		18	5	
TypeError call(redux/actions/editorActio... Unhandled Cannot read property 'anchor'...		4	4	

Figure 4.1: Errors reported by sentry

4.3 User testing

Users who were known to like internet memes were selected and asked to edit a GIF from scratch by completing a series of tasks. Their actions were monitored through a screen sharing software.

1. Upload a GIF into the editor
2. Add a text edit
3. Animate the text edit to match the movement of a person running in the GIF.
4. Change the size of the edit in the middle of the animation.
5. Add a text edit that starts to be visible in the middle of the animation.
6. Match movement of the second edit to a second person in the animation.
7. Cut out the start of the animation.
8. Add an image edit
9. Rotate the edit
10. Put the text edit above the image edit
11. Remove the image edit
12. Revert to the state at point 7.
13. Create a new project from template and edit its texts

Participant A

Participant A is a 23-year-old student of informatics at CTU. He is known for creating memes, which he shares with his classmates and therefore was an ideal user of this tool.

The participant had problems understanding the table with animation positions and the movement of the anchor. When asked to cut out the animation's start, he intuitively did the right action but mentioned that the UI is slightly confusing. He also noted that the buttons for motion tracking were hard to find and that timeline did not look like a timeline at first sight.

He said that he will definitely use the tool and did not miss any features.

Conclusion: Thanks to participant A, parts of the UI that were expected to be problematic were proven to be so, and the following changes were implemented:

1. Frame numbers were added to every 5th frame of the timeline
2. A blur was added to unused frames after the animation duration is cut
3. Animation checkbox was moved on top of the edit box, and its size was increased
4. Animation controls were moved to the right not to block the timeline

Participant B

Participant B is a 22-year-old student of economics who also works as a backend developer. He has never created any internet memes but is an avid consumer of them and has experience editing videos.

He had no problems finding animation and motion tracking controls, suggesting that UI tweaks added after participant A were successful. While attempting to motion track text into a running person, he was tempted to press the "Auto-track" button and expected the same behavior as in Adobe After Effects, which confused him. After figuring out that motion tracking frame by frame was more suitable, he finished the tasks without any problems.

He has liked how "light-weight" the tool is and will likely use it in the future. He suggested adding more intuitive controls, such as dragging edit timelines instead of clicking arrows to change their order and double-clicking text to edit it. He would also like to see more features to allow more advanced edits and buttons to share animations on social networks directly.

Participant C

Participant C is a student of computer science who consumes memes passively. He does not have any experience with video editing software.

He managed to complete all tasks successfully, but just as participant B tried to motion track the animation with the "auto-track" button, and did not notice the motion tracking anchor. After a brief trial and error period, he figured out how motion tracking works completely and finished the task with ease. He also took a while to find controls for cutting off video length, even though he later mentioned that the UI feels very intuitive once he realized it.

Lastly, he suggested adding a drag and drop option for uploading and otherwise did not miss any features.

Conclusion: To prevent users from using the more advanced auto-tracking feature, the button was made smaller, marked as advanced, and a short description was added on hover.

Participant D

Participant D is a student of computer science who consumes memes regularly and has created a few himself.

Even though "auto-track" was newly marked as an advanced feature, he tried to use it. It took him a few trials to find that the "motion track" button worked better for his case. His reasoning for this was that the "motion track" button did not have any description, and he did not know what it does. Other than that, he has managed to finish all tasks and did not miss any features.

Conclusion: The following changes were implemented based on feedback from participants B, C, and D:

1. Motion tracking anchor position was changed to be saved with each move instead of separately
2. A short video showing how to use the "motion track" button was added

Participant E

Participant E is a student of law who creates internet memes regularly.

He had trouble finding the motion tracking button and then struggled quite a bit with motion tracking. First, he could not find the motion tracking controls. This was likely caused by the "Show motion tracking controls" checkbox being too small. Second, the anchor he created was too small, making the motion tracking algorithm jump around unpredictably. After giving him a hint about the anchor size, he has finished all other tasks without any significant issues, but finding buttons for reordering items took him a few moments.

Lastly, he mentioned that he was looking for shortcuts near the controls and might not have found the order controls without it being a task.

Conclusion: Based on feedback from participant E, the shortcuts infobox was made more visible, and some shortcuts are now displayed after hovering over buttons with the same functionality. Motion tracking checkbox was increased and set to be checked by default.

Participant F

Participant F is a student of computer science. He has never created a meme but enjoys seeing them.

He managed to motion track the text without any issues. He struggled to cut the beginning of the animation, which was caused by a progress indicator (red pin) blocking the animation length barriers.

Conclusion: Progress indicator was lifted above the video length barriers.

User testing conclusion

Participants were generally successful at finishing their tasks. Motion tracking proved to be the most problematic as expected. User's complaints were addressed by adding hints, changing labels, moving elements to more intuitive positions, changing their sizes, and minor functional changes.

A better approach to conduct user testing might have been to present a finished animation and ask the users to recreate it. A few participants have mentioned that they might not have found some features without being directly asked to use them.

Chapter 5

Conclusion

This project aimed to implement a web-based tool for editing GIFs. All requirements were implemented successfully. The editor runs only in the client's browser, performance-intensive operations are executed on a background thread, users can adjust edits using a timeline, and the app was tested thoroughly.

5.1 Usage analytics

By posting the project on Producthunt, where it received 43 upvotes [34], reddit.com, and by paying for Google Ads [12], the website gained traffic for usage testing. An `export` event is tracked every time user exports an animation.

Event name	Count	Users
Visits	803	803
GIF exports	336	104

Table 5.1: Usage analytics for period 20.12.2020 - 26.12.2020

As we can see, only about 13 % of users decide to create a GIF, but when they do, it is 3 GIFs on average. It is not tracked whether users prefer creating GIFs from templates or upload their own, but this statistic will be added in the future.

These data suggest that there are users who find this tool useful. However, 13 % leaves much room for improvement, and more UX testing needs to be performed.

5.2 Future work

SEO

The website is currently getting minimal traffic through organic search results. Adding backlinks, improving texts, and sharing the website on social media are all techniques that could be tried in the future to improve its position.

Adding API

One of how the tool could be monetized, apart from ads, would be to add an API, through which users could create bulks of customized GIFs. This might be useful for marketing purposes by for example sending each costumer a funny GIF with their name.

Mobile App

There might be users who would prefer using a mobile app instead of the web app. The app could create GIFs only from templates so that the editor would not have to be implemented.

More editor features

There is plenty of features that were not implemented and might be useful for GIF animation. Such features include:

1. Edit timeline splitting - Right now, an edit that is supposed to be visible at the start, end, but not in the middle of the animation has to either be moved to the side or split into multiple edits.
2. Transparency layer - A layer that would force all other edits below it to be transparent. This would create an illusion of edits being behind objects in an animation
3. Multi-anchor tracking - By anchoring the edit with multiple anchors, the motion tracking could also track rotation and, in the case of three or four anchors, a skew.
4. FFmpeg is a library that would allow users to upload videos in mp4, WebM, or other video formats and convert them into GIFs on the client-side. [11] This was not implemented in the current version because the library uses **SharedArrayBuffer**, which is a) supported by only 33 %

of browsers [6] and b) requires the website to be **cross-origin-isolated**, which is not compatible with Firebase Auth SDK, likely because of its use of iframe. In the future, it might be possible to implement auth using Firebase's REST API and maybe use FFmpeg to edit videos instead of GIFs.

5. Share buttons for social media - using APIs of big social networks to share GIFs directly could make the website more popular and create an organic source of backlinks.

Appendix A

Bibliography

[1] https://www.w3.org/TR/workers/.

[2] https://www.w3.org/Graphics/GIF/spec-gif89a.txt.

[3] https://firebase.google.com/.

[4] Adobe premiere pro. https://helpx.adobe.com/in/premiere-pro/user-guide.html/in/premiere-pro/using/motion-graphics-templates.ug.html.

[5] Application monitoring and error tracking software. https://sentry.io/.

[6] Can i use... support tables for html5, css3, etc. https://caniuse.com/.

[7] Change the workspace layout in final cut pro. https://support.apple.com/en-gb/guide/final-cut-pro/ver6b3626147/mac.

[8] Choose the right keywords with our research tools - google ads. https://ads.google.com/home/tools/keyword-planner/.

[9] Concepts. https://webpack.js.org/concepts/.

[10] Documentation - using babel with typescript. https://www.typescriptlang.org/docs/handbook/babel-with-typescript.html.

[11] ffmpegwasm/ffmpeg.wasm. https://github.com/ffmpegwasm/ffmpeg.wasm.

[12] Google ads - get more customers with easy online advertising. https://ads.google.com/home/.

[13] Html cnavas 2d context. https://www.w3.org/TR/2dcontext/.

[14] konvajs/konva. https://github.com/konvajs/konva.

[15] Kotlin to javascript. https://kotlinlang.org/docs/tutorials/javascript/kotlin-to-javascript/kotlin-to-javascript.html.

[16] Material design. https://material.io.

[17] Putimagedata vs drawimage. https://www.measurethat.net/Benchmarks/Show/9510/0/putimagedata-vs-drawimage.

[18] Ramda. https://ramdajs.com/.

[19] React color. https://casesandberg.github.io/react-color/.

[20] React – a javascript library for building user interfaces. https://reactjs.org/.

[21] Redux - a predictable state container for javascript apps.: Redux. https://redux.js.org/.

[22] Rescript introduction. https://rescript-lang.org/docs/manual/latest/introduction.

[23] Stack overflow developer survey 2020. https://insights.stackoverflow.com/survey/2020.

[24] Styled components. https://styled-components.com/.

[25] Typed javascript at any scale. https://www.typescriptlang.org.

[26] Usage with typescript | redux. https://redux.js.org/recipes/usage-with-typescript#a-practical-example.

[27] what is babel? https://babeljs.io/docs/en/index.html.

[28] Mediastream recording. https://www.w3.org/TR/mediastream-recording/, Dec 2020.

[29] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[30] Deanm. deanm/omggif. https://github.com/deanm/omggif.

[31] A. H. Dekker. Kohonen neural networks for optimal colour quantization. *Network: Computation in Neural Systems*, 5(3):351–367, 1994.

[32] B. Gao and M. Spratling. Robust template matching via hierarchical convolutional features from a shape biased cnn. *arXiv preprint arXiv:2007.15817*, 2020.

[33] Jnordberg. jnordberg/gif.js. https://github.com/jnordberg/gif.js.

[34] P. Kubes. Gifmemes - create gif memes with motion tracked text or images. https://www.producthunt.com/posts/gifmemes, Dec 2020.

[35] U. Rembold. *Comparison of correlation techniques*, page 86–93. IOS Press, 1995.

[36] P. Sebastian, Y. V. Voon, and R. Comley. The effect of colour space on tracking robustness. *2008 3rd IEEE Conference on Industrial Electronics and Applications*, 2008.

[37] M. Shaw. Gifjif - custom gif creator. `https://apps.apple.com/us/app/gifjif/id1078038667`, Feb 2016.

Appendix B

Attachments

```
/
├─code ...  Source code of the project
└─measure_function_test_gifs ...  Gifs on which were the matching
                                  measures compared
```