

Politechnika Częstochowska
Wydział Informatyki i Sztucznej Inteligencji

Praca dyplomowa inżynierska

Projekt i implementacja aplikacji internetowej do tworzenia prezentacji multimedialnych wykorzystując grafikę SVG

Design and implementation of a web application for creating SVG
graphic presentations

Imię i nazwisko: Jakub Truszkowski

Numer albumu: 135206

Promotor: dr hab. inż. Krystian Łapa prof. PCz

Kierunek: Informatyka

Zakres (specjalność): Inżynieria oprogramowania

Częstochowa 2024

Spis Treści

1. Cel pracy	3
2. Zakres pracy	3
1 Przegląd dostępnych rozwiązań	4
1.1 Microsoft PowerPoint	4
1.2 Open/Libre Office	4
1.3 Google Slides	4
2 Omówienie zagadnienia...	5
3 Opis wykorzystanych technologii oraz narzędzi	6
3.1 Svelte	6
3.2 SVG.js	6
3.3 svg2pdf.js	6
3.4 npm	6
3.5 Vite	6
3.6 Visual Studio Code???	6
4 Założenia projektu	7
5 Prezentacja projektu	8
5.1 Edytor	8
5.2 Przeglądarka slajdów	8
5.3 Eksport do formatu PDF	8
6 Opis wybranych rozwiązań programistycznych	9
7 Podsumowanie	10
7.1 Przydatność aplikacji	10
7.2 Perspektywy rozwoju	10
7.2.1 Wsparcie dla przeglądarek Safari	10
7.2.2 Nowe funkcje dla tekstu	10
7.2.3 Rozwinięcie animacji dodawanych do obiektu	10
8 Streszczenie	11
9 Summary	12
Bibliografia	12

1. Cel pracy

Celem pracy jest stworzenie aplikacji internetowej do tworzenia prezentacji multimedialnych przy użyciu grafiki SVG. Aplikacja internetowa powinna zapewniać prosty i zrozumiały interfejs, obsługę predefiniowanych animacji na tworzonych przez użytkownika obiektach, przeglądarkę slajdów, oraz eksport pliku prezentacji do formatu PDF.

Pobocznym celem jest sprawdzenie, czy obecny standard SVG implementowany w najpopularniejszych przeglądarkach pozwala na realne wykorzystanie tej technologii w projektach, na przykład w zastępstwie canvas API.

2. Zakres pracy

W rozdziale pierwszym pracy opisano porównanie istniejących już rozwiązań kreatorów prezentacji multimedialnych, m. in. takich jak Microsoft PowerPoint. W drugim rozdziale opisano omówienie zagadnienia stworzenia aplikacji internetowej do tworzenia prezentacji multimedialnych. Do zagadnienia należy wykonanie analizy technologii obsługiwanych przez najpopularniejsze przeglądarki, rozwiązanie problemu przechowywania danych potrzebnych do odtworzenia aplikacji i wyszczególnienie rozwiązań. W kolejnym rozdziale szczegółowo opisano technologie i narzędzia użyte w procesie tworzenia projektu inżynierskiego, takie jak Svelte, SVG.js i npm, oraz podano porównania do konkurencyjnych bibliotek i narzędzi, oraz powody ich użycia. W rozdziale czwartym opisano założenia projektu - wymagania funkcjonalne, нефункционалне, diagram przypadków użycia, diagram klas i projekt interfejsu graficznego użytkownika w programie Figma. W następnym rozdziale opisana jest prezentacja projektu, czyli działanie najważniejszych jego części, edytora prezentacji, przeglądarki slajdów i eksport prezentacji do pliku PDF. W rozdziale szóstym można znaleźć najciekawsze rozwiązania programistyczne, takie jak implementacja poruszaniem obiektem SVG bez użycia zewnętrznej biblioteki, implementacja zwiększania obiektu przez użytkownika i implementacja eksportu do pliku PDF przy użyciu biblioteki `svg2pdf.js`. W ostatnim, siódmym rozdziale znajduje się podsumowanie pracy, razem z wnioskami zdobytymi podczas tworzenia projektu, oraz subiektywnymi perspektywami rozwoju kreatora.

Rozdział 1

Przegląd dostępnych rozwiązań

Na rynku znajduje się obecnie wiele darmowych jak i płatnych rozwiązań dotyczących prezentacji multimedialnych. Najpopularniejszym z nich jest Microsoft PowerPoint, który jest rozwiązaniem płatnym, popularne są również Open/Libre Office Impress i Google Slides. Wszystkie pozwalają na stworzenie zaawansowanych prezentacji multimedialnych, posiadają kreatory animacji dla obiektów, wsparcie dla szablonów styli.

1.1 Microsoft PowerPoint

Dzieło Microsoftu jest liderem, jeżeli chodzi o światowe użycie. Ponieważ jest to najpopularniejszy program tego typu, inne programy jak Open/Libre Office Impress kopiuje go w wyglądzie interfejsu użytkownika, jak i działaniu.

Program posiada zaawansowany, jednak często trudny do zrozumienia interfejs użytkownika. Można dowolnie edytować tekst, ustawiać wielkość nawet pojedynczych liter, kolory tła czy liter. Jest też ogromna ilość przejść pomiędzy slajdami i animacji obiektów. W nowszych wersjach programu można wykonać bardzo zaawansowane prezentacje, które nieraz mogłyby przypominać render ręcznie wymodelowanych i animowanych obiektów 3d.

1.2 Open/Libre Office

1.3 Google Slides

Rozdział 2

Omówienie zagadnienia...

Rozdział 3

Opis wykorzystanych technologii oraz narzędzi

3.1 Svelte

Svelte to otwartoźródłowy frontendowy framework języka JavaScript stworzony przez Richa Harrisa w 2016 roku[1]. Cechuje się małym rozmiarem i prostotą. W porównaniu do popularnych frameworków tego języka - Reacta i Vue - nie jest on importowaną do projektu biblioteką, ale kompiluje on projektowe pliki, do plików wynikowych.

3.2 SVG.js

SVG.js to otwartoźródłowa biblioteka napisana w języku JavaScript stworzona przez Wouta Fierensa w 2012 roku[2]. Ma za zadanie ułatwić tworzenie obiektów SVG i operacje na nich. Jak można zauważyć na stronie biblioteki, rozwiązanie to jest wolniejsze[3], niż obsługa tego typu grafiki w kodzie, jednak jest ona dużo przyjemniejsza i szybsza do napisania dla programisty przez jej obiektowe podejście, a także posiada ułatwione animowanie obiektów SVG.

3.3 svg2pdf.js

Svg2pdf.js jest otwartoźródłową biblioteką Javascript, działającą także na frontendzie. Rozwija ją niemiecka firma yWorks GmbH na licencji MIT, jednak napisana została w 2013 przez Vitalya Puzrina[4]. To ona umożliwia przeniesienie prezentacji z grafiki SVG do pliku PDF gotowego do pobrania przez użytkownika. W wymaganiach do działania biblioteki potrzebna jest także instalacja biblioteki jsPDF. Wszystko odbywa się na komputerze użytkownika, bez potrzeby używania zasobów serwera.

3.4 npm

npm to menedżer pakietów dla języka JavaScript. Pozwala na łatwą instalację paczek potrzebnych do skonfigurowania projektu w językach kompilowalnych do JavaScript (TypeScript, CoffeScript itp). Wszystkie powyższe biblioteki wraz z ich wymaganiami zostały dodane do projektu właśnie za pomocą npm. Wyszukać paczki można na [tej stronie](#).

3.5 Vite

Vite jest narzędziem programistycznym do pomocy przy rozwoju aplikacji internetowych. Kreator aplikacji pozwala na wybranie frameworku JavaScript z wybranych, m. in. Svelte, od razu można dodać też obsługę języka TypeScript, Sass, czy narzędzia do analizy kodu jak ESLint, lub obsługę testów. Vite ma wbudowany serwer dla programisty, uruchamiany na jego maszynie, który jest przystosowany pod tzw. „Hot reloads”, czyli szybkie, automatyczne odświeżanie, kiedy zmieni się kod źródłowy. Umożliwia to obserwowanie zmian na żywo, bez konieczności budowania projektu, czy nawet odświeżania przeglądarki.

3.6 Visual Studio Code???

Warto tutaj go dodawać?

Rozdział 4

Założenia projektu

Rozdział 5

Prezentacja projektu

5.1 Edytor

5.2 Przeglądarka slajdów

5.3 Eksport do formatu PDF

Rozdział 6

Opis wybranych rozwiązań programistycznych

Rozdział 7

Podsumowanie

7.1 Przydatność aplikacji

Aplikacja z punktu użytkownika działa poprawnie i oferuje podstawowe możliwości tworzenia prezentacji multimedialnych takie jak wstawianie tekstu, obrazów, animacji, czy eksport prezentacji do pliku PDF. Rozwiązanie takie jako darmowy kreator prezentacji być może miałby nawet możliwość odniesienia pewnego sukcesu na rynku komercyjnym - użytkownikowi nie jest potrzebne konto, miejsce w chmurze, czy odpowiedni system operacyjny jak w przypadku rozwiązań konkurencyjnych, a przeglądarka obsługująca OPFS i SVG. Unika się też w ten sposób konieczności instalacji innego programu takiego jak PowerPoint czy LibreOffice. Aplikacja w tej chwili może być wykorzystywana z powodzeniem do prostych prezentacji.

7.2 Perspektywy rozwoju

Dużą niewiadomą dla rozwoju tego typu aplikacji, jest rozwój standardu SVG 2, który ma przynieść między innymi usprawnienia w obsłudze tekstu [5], jednak jego wprowadzenie jest mało prawdopodobne. Rozwijany od 2016 roku [6] nie doczekał się jednak wsparcia w jakiegokolwiek przeglądarce.

7.2.1 Wsparcie dla przeglądarek Safari

Obecnie cała aplikacja nie działa na przeglądarkach Safari. Powodem jest brak wsparcia przez nie API OPFS, a konkretnie metody umożliwiającej stworzenie pliku w OPFS. Rozwiązaniem mogłoby być przejście na API IndexedDB, lub stworzenie tzw. service workera, które safari już wspiera.

7.2.2 Nowe funkcje dla tekstu

Obiekt tekstowy w tej implementacji posiada tylko jeden obiekt svg, co uniemożliwia zrobienie rzeczy do których przyzwyczaili nas inne edytory, na przykład z programu Microsoft PowerPoint. Nie można na przykład uzyskać w tym samym obiekcie tekstowym innego rozmiaru czcionki, nawet dla jednej litery. Było by to możliwe poprzez zastosowanie dosyć skomplikowanego algorytmu dzielącego tekst na sekcję przy użyciu obiektu SVG o nazwie tspan. Każdy taki obiekt, będący jako dziecko obiektu text, może mieć inne właściwości, jak wielkość czcionki, kolor, czy efekty pogrubienia, pochylenia i podkreślenia. Obecnie nie można też zaznaczać tekstu kursorem myszy.

7.2.3 Rozwinięcie animacji dodawanych do obiektu

Aby jeszcze bardziej rozwinąć aplikację, należałoby dodać edytor animacji, w której użytkownik, mógłby dokładnie zdefiniować jaką animację chce uzyskać na danym obiekcie. Umożliwiłoby to tworzenie lepszych i ciekawszych prezentacji multimedialnych, co zachęciło by potencjalnych użytkowników do jego użycia.

Streszczenie

Po polsku

Summary

In english

Bibliografia

- [1] „Svelte”. [Online]. Dostępne na: <https://en.wikipedia.org/wiki/Svelte>
- [2] „SVG.js License”. [Online]. Dostępne na: <https://github.com/svgdotjs/svg.js/blob/master/LICENSE.txt>
- [3] „It's speedy - SVG.js”. [Online]. Dostępne na: <https://svgjs.dev/docs/3.2/#it-s-speedy>
- [4] „svg2pdf.js MIT License”. [Online]. Dostępne na: <https://github.com/yWorks/svg2pdf.js/blob/master/LICENSE>
- [5] „Auto-wrapped text”. [Online]. Dostępne na: <https://svgwg.org/svg2-draft/text.html#TextLayoutAuto>
- [6] „Scalable Vector Graphics (SVG) 2, W3C Candidate Recommendation”. [Online]. Dostępne na: <https://www.w3.org/TR/2016/CR-SVG2-20160915/>