

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

Praca przyjęta dnia:

Podpis promotora:

Spis Treści

Wstęp	2
Cel pracy	3
Zakres pracy	3
1 Przegląd dostępnych rozwiązań	5
1.1 Microsoft PowerPoint	5
1.2 Open/Libre Office Impress	6
1.3 Apple Keynote	6
1.4 Podsumowanie przeglądu	6
2 Omówienie zagadnienia... ..	7
3 Opis wykorzystanych technologii oraz narzędzi	8
3.1 Svelte	8
3.2 SVG.js	8
3.3 svg2pdf.js	8
3.4 npm	8
3.5 Vite	8
3.6 Visual Studio Code???	9
4 Założenia projektu	10
4.1 Wymagania funkcjonalne	10
4.2 Wymagania нефункционалне	10
4.3 Diagram przypadków użycia	10
4.4 Diagram klas	10
4.5 Projekt interfejsu	10
5 Prezentacja projektu	11
5.1 Edytor	11
5.2 Przeglądarka slajdów	14
5.3 Eksport do formatu PDF	14
6 Opis rozwiązań programistycznych	15
7 Podsumowanie	16
7.1 Przydatność aplikacji	16
7.2 Perspektywy rozwoju	16
Bibliografia	18
Spis rysunków	20
Streszczenie	21
Summary	22
Słowa kluczowe	23

Wstęp

Prezentacje multimedialne są dzisiaj nieodłączną częścią dzisiejszego świata. Zobaczymy je podczas spotkań biznesowych i spotkań zarządu firm, konferencjach, szkoleniach dla pracowników, a wszystko po to by w bardziej przystępny sposób przedstawić słuchającym zagadnienie.

Już w latach 40. Armia USA zaadaptowała grafoskopy - projektory działające na zasadzie poświetlania diapozytu, obrazu naniesionego na przeźroczysty materiał, i przenoszącego go przez soczewki i lustra na powierzchnię ściany czy ekranu projekcyjnego. Wszystko to po to, by lepiej przygotowywać swoich żołnierzy do walki w bardziej przystępny i łatwiejszy do zrozumienia sposób. W późniejszym czasie zaczęli je używać także nauczyciele akademicki, [1] a kilka lat później, rzutniki pisma zaczęto używać w amerykańskich firmach.[2]

Początkowo slajdy do tego typu projektorów tworzone najczęściej ręcznie, pisząc na specjalnej folii (ang. carbon film) metalowym rysikiem, czasami z pomocą szablonów liter, cyfr i kształtów, lub używając do tego zwykłej maszyny do pisania.[3] Można było także wkleić zdjęcia, na przykład z gazety, z pomocą kleju kauczukowego, wody i zdjęcia przyklejonego do folii.[3] Było więc to zadanie bardzo żmudne i wymagające sporej precyzji przy tworzeniu.



Rysunek 1: Grafoskop używany na Uniwersytecie North Carolina, lata 60.[4]

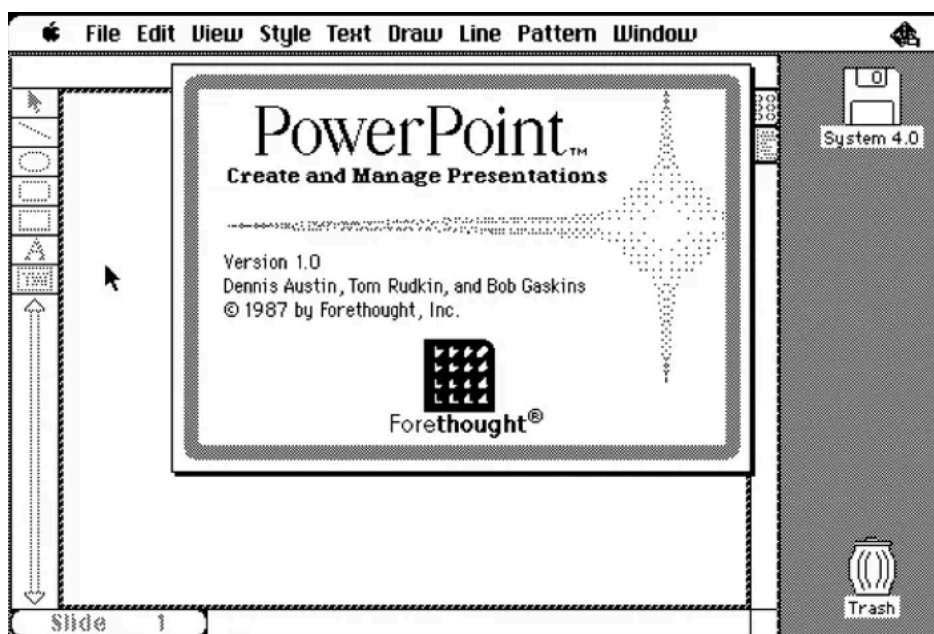
W tym samym okresie popularne były też rzutniki przeźrocze - diaskopy, najczęściej na film typu 135. Najbardziej znane na świecie są rzutniki Kodaka, z tzw. obrotową karuzelą, w której znajdują się pojedyncze kawałki taśmy filmowej. Projektor tak jak i grafoskop działa na zasadzie podświetlenia półprzeźroczystego materiału oraz wyświetlenie powstałego obrazu na ścianie przez system soczewek, jednak jest zazwyczaj od niego mniejszy.

Tworzenie slajdów do rzutników przeźroczych różni się od tworzenia slajdów do grafoskopów, ponieważ opierało się po prostu zazwyczaj na robieniu zdjęć.[5] Takie materiały były więc zazwyczaj tekstem napisanym na jakiejś powierzchni, której po prostu zrobiono zdjęcie,

to samo tyczy się kopiowania obrazków z książek czy gazet. Następnie wywoływano film diapozytywowy taki jak Kodak Ektachrome czy polski Foton, lub używano filmu natychmiastowego cięto film na pojedyncze slajdy, po czym film mógł być wyświetlany przez rzutnik.

W latach 80. szybki rozwój komputerów pozwolił na zdigitalizowanie procesu tworzenia prezentacji. Programy takie jak BRUNO i Slidemaster mogły znacząco ułatwić tworzenie slajdów, jednak przeniesienie ich na film wymagało specjalnych drukarek i wyszkolonych operatorów do obsługi programów. W 1987 PowerPoint zostaje wypuszczony na komputery Macintosh i jeszcze w tym samym roku zostaje zakupiony przez Microsoft, jednak prezentacje są dalej prezentowane głównie na dwóch wyżej wymienionych rzutnikach. Dopiero w latach 90. i na początku XXI wieku, projektory cyfrowe zaczynają zastępować grafoskopy i diaskopy.

Cyfrowa rewolucja pozwoliła na jeszcze większe rozprzestrzenienie się prezentacji multimedialnych. Znajdziemy je na każdym etapie życia: w szkołach, w pracy, oglądając przeróżne analizy, na przykład rynku mieszkaniowego, czy przy wypuszczeniu nowych produktów przez firmy.



Rysunek 2: Interfejs użytkownika w pierwszym wydaniu programu PowerPoint [6]

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.

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, niefunkcjonalne, 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 programów do tworzenia prezentacji multimedialnych. Najpopularniejszym z nich jest Microsoft PowerPoint, który jest rozwiązaniem płatnym, popularne są również Open/Libre Office Impress i Google Slides, a także Apple Keynote jeżeli chodzi o komputery typu Mac. Programy te pozwalają na stworzenie zaawansowanych prezentacji multimedialnych, posiadają kreatory animacji dla obiektów, wsparcie dla szablonów stylów.

1.1 Microsoft PowerPoint

PowerPoint został wypuszczony na rynek przez firmę Forethought Inc. w 1987 na komputery Apple Macintosh po niemal dwóch latach pracy nad oprogramowaniem.[7] Planowano wypuścić oprogramowanie także na komputery z systemem Windows w niedalekiej przyszłości.

Program ten jest liderem (dane z 2009 r.), jeżeli chodzi o światowe użycie i to pomimo istnienia darmowych alternatyw.[8] Powodem tego jest implementacja wielu przydatnych funkcji, oraz niejako stanie się standardem w tego typu zastosowaniach. Przy konwersji z formatu Open/Libre Office lub Google Slides na .pptx często występują poważne błędy, co zniechęca do użycia darmowych alternatyw, pomimo tego, że programy te są wzorowane na programie Microsoftu.

Od wersji 2007 interfejs użytkownika w całym pakiecie Office zmienił się dość mocno w porównaniu do poprzedników, co wywołało mieszane uczucia wśród użytkowników. W tym przypadku konkurencja w postaci Libre Office pozwoliła na wybranie nowszej, lub pozostawienie starszej wersji interfejsu. Program ma świetną obsługę tekstu, można zmieniać atrybuty nawet pojedynczych liter w tym samym polu tekstowym, co nie jest dostępne we wielu programach z uwagi na trudność implementacji. W oprogramowaniu znajdziemy też możliwość wstawienia do slajdu zdjęć, figur, hiperlinków, gifów, a nawet wideo. Jest też zaawansowany system szablonowy slajdów, którym użytkownik może nadać pewne właściwości, jak kolor tła, czy umiejscowienie elementów. Program ma także bardzo rozbudowaną obsługę animacji i przejść pomiędzy slajdami. Animacje są tworzone przez użytkownika dzięki interfejsowi graficznemu, w widoku slajdów, można aktywować na różne sposoby: automatycznie, czy przez kliknięcie.

Od 2017 roku PowerPoint jest też dostępny jako aplikacja internetowa w przeglądarce w pakiecie Office365, chociaż posiada pewne ograniczenia względem wersji desktopowej.

1.2 Open/Libre Office Impress

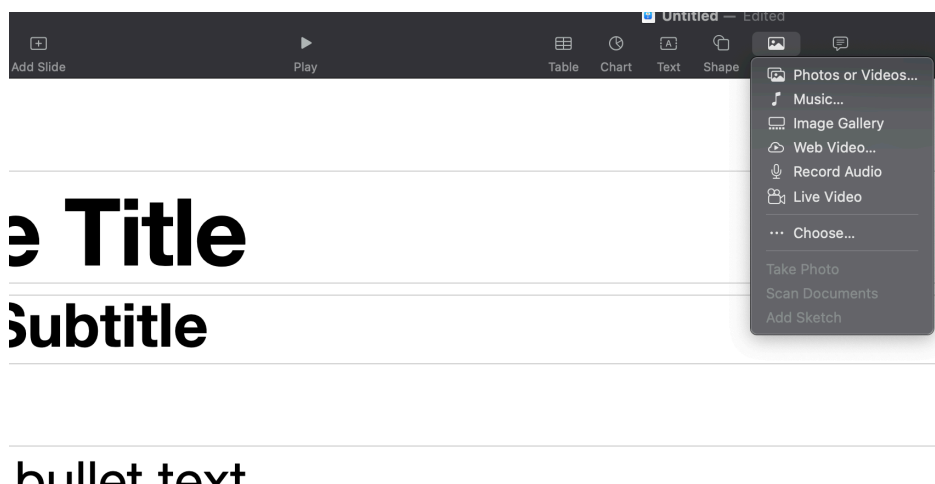
Program ten narodził się w 1999 kiedy to firma Sun wykupiła firmę Star Division, która stworzyła pakiet oprogramowania biurowego konkurencyjny dla Microsoft Office. Podobno wykupienie firmy miało być tańsze roczne koszty wykupienia Microsoft Office w całym Sun.

W 2002 wypuszczono pakiet w wersji 1.0 a z nim Open Office Impress. Program już na starcie miał wsparcie dla szablonów slajdów, oraz animacji.

1.3 Apple Keynote

Program stworzony przez Apple jako konkurencja dla PowerPoint for Mac z pierwszym wydaniem w roku 2003 na systemy OS X jako część pakietu iWork. Nie odniósł sukcesu wśród użytkowników systemu macOS, pomimo bycia dostępnym tylko na ten system. Obecnie za darmo dla każdego użytkownika systemu Apple, oferuje także eksport pliku do formatu PowerPoint (pptx).

Interfejs Apple Keynote jest na pewno prostszy, niż ten, który możemy znaleźć w Microsoft PowerPoint, lecz nie zawsze jest lepszy. Przykładem jest proste wstawienie zdjęcia do slajdu. Po kliknięciu przycisku Media, to opcja „Photos or Videos” wydaje się być tą prawidłową, ale po jej kliknięciu wyskoczą nam zdjęcia i wideo z aplikacji Photos. Aby wybrać zdjęcie, czy wideo z pliku, musimy wybrać opcję „Choose...”. Jest to bardzo nieintuicyjne, przeciętny użytkownik, najprawdopodobniej zmarnuje sporo czasu, zanim domyśli się jak ma wstawić zdjęcie do tego programu.



Rysunek 3: Menu wstawiania mediów w Apple Keynote

1.4 Podsumowanie przeglądu

Projekt aplikacji internetowej powinien przede wszystkim skupiać się na prostocie użytkowania, ponieważ nie ma szans na konkurowanie z powyższymi programami pod względem funkcjonalności, które budowały przez długie lata.

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[9], [10]. 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[11]. Ma za zadanie ułatwić tworzenie obiektów SVG i operacje na nich. Jak można zauważyć na stronie biblioteki, rozwiązanie to jest wolniejsze[12], 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[13]. 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

4.1 Wymagania funkcjonalne

4.2 Wymagania niefunkcjonalne

4.3 Diagram przypadków użycia

4.4 Diagram klas

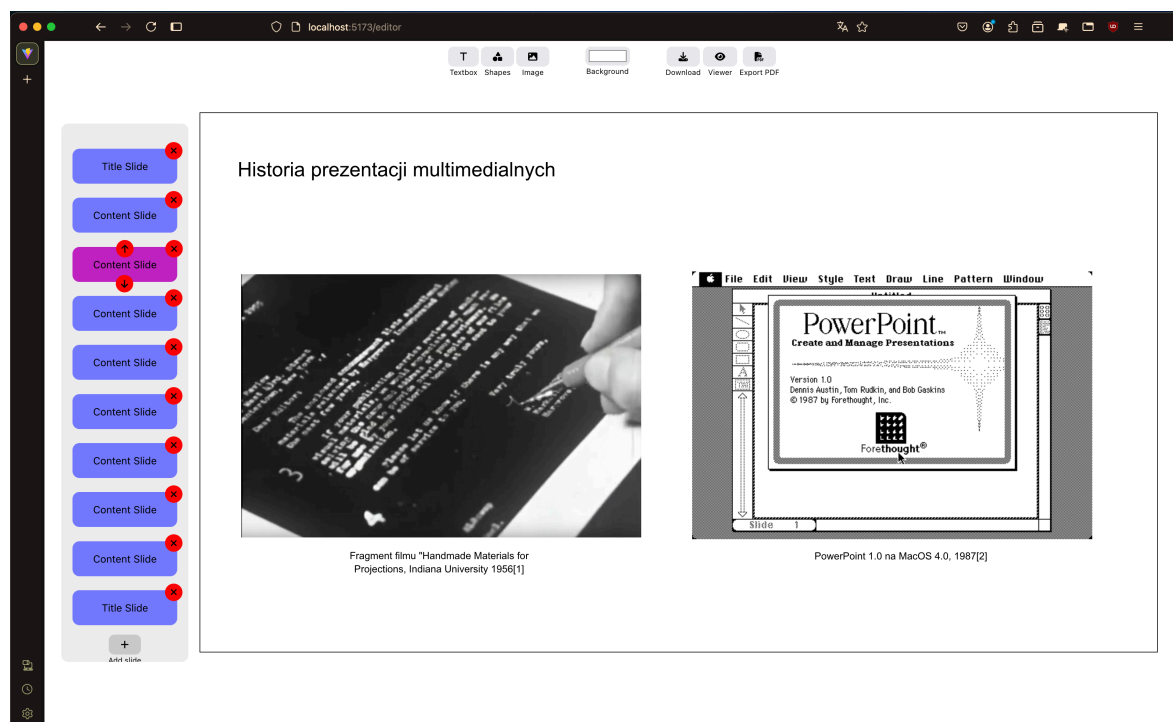
4.5 Projekt interfejsu

Rozdział 5

Prezentacja projektu

5.1 Edytor

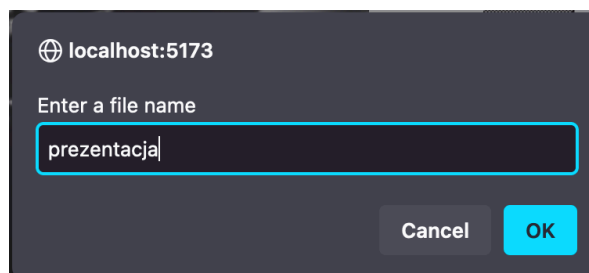
Edytor jest najważniejszą i najbardziej skomplikowaną częścią projektu, ponieważ musi on pozwolić użytkownikowi na łatwe i intuicyjne tworzenie treści w programie, oraz przetłumaczyć je na język komputera, aby dane mogły zostać zapisane do pliku i tak pobrane i odtworzone na dowolnym innym urządzeniu.



Rysunek 4: Widok edytora po wejściu do aplikacji

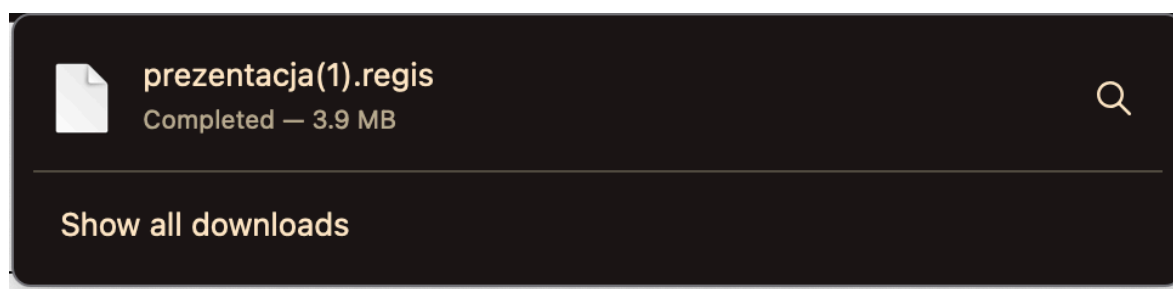
Na samej górze widoczny jest pasek zadań podzielony na trzy części: dotyczącą obiektów, slajdu i całego pliku. Zaczynając od lewej do slajdu użytkownik może dodać rzeczy niezbędne w prezentacji multimedialnej: pole tekstowe, dzięki któremu można dodać tekst do naszego slajdu, kształt SVG, predefiniowany lub swój własny (stworzony w użytkownika w zewnętrznym programie i wyeksportowany jako ścieżka SVG), oraz zdjęcia, które można załadować bezpośrednio z dysku danego komputera. Następnie oddzielony przerwą jest wybierak koloru slajdu. Należy jednak zauważyć, że kolor tła slajdów działa tylko we wbudowanej przeglądarce slajdów, ale nie są wspierane przez eksport do PDF.

W kolejnej części górnego paska dostępne są trzy kolejne opcje: opcja pobierania pliku prezentacji w z rozszerzeniem .regis, opcja przejrzenia prezentacji użytkownika we wbudowanej przeglądarce slajdów online i opcja wygenerowania i pobrania pliku PDF, który możemy odtworzyć na praktycznie każdym urządzeniu bez dostępu do internetu.



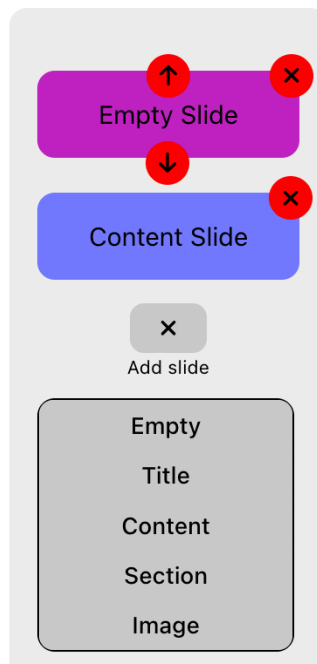
Rysunek 5: Okno nazwy pobieranego pliku

Po wybraniu opcji pobrania użytkownik zostanie poproszony o podanie nazwy pliku, który ma zostać zapisany na dysku komputera. W tym celu pojawia się okienko dialogowe, w którym należy wpisać preferowaną nazwę pliku. Po zatwierdzeniu przyciskiem „OK”, plik zostaje pobrany na dysk użytkownika. Brak wpisania nazwy, lub naciśnięcie przycisku „Cancel” spowoduje anulowanie pobierania prezentacji. Wybranie przycisku Viewer automatycznie zapisze postępy pracy w prezentacji i przeniesie użytkownika do przeglądarki prezentacji, w której będzie można wyświetlić dany projekt w tym animacje, które zostały dodane do obiektów. Ostatnia opcja również zapisze postępy pracy i wyeksportuje prezentację do pliku PDF. Możliwe jest to dzięki bibliotece `svg2pdf.js`.



Rysunek 6: Pobrany plik prezentacji

Po lewej stronie na szarym tle znajduje się panel slajdów, który pozwala na zarządzanie całą strukturą prezentacji. W tym miejscu znajdują się wszystkie slajdy w projekcie, ułożone od góry do dołu, jak w konkurencyjnych rozwiązaniach. Zaznaczony przez użytkownika slajd jest podświetlony na kolor różowy co wyróżnia go z pośród reszty, która jest w kolorze jasnogranatowym. Każdy slajd można usunąć klikając czerwony krzyżyk umieszczony w prawym górnym rogu każdego slajdu, a zaznaczony slajd można przenieść w projekcie wyżej lub niżej przyciskami ze strzałkami co odpowiada pierwszeństwu w wyświetlaniu. Ponadto każdy slajd ma nazwę w zależności od typu szablonu, który został wybrany przy jego tworzeniu. Na samym dole panelu znajduje się szary przycisk z podpisem „Add slide” służący do dodania slajdu. Po jego naciśnięciu rozwija się lista slajdów z szablonów: pusty, tytułowy, zawartości, sekcja i obrazek. Jest to dość prosty ale skuteczny system szablonów, który ułatwi użytkownikowi przystosowanie danego slajdu na swoje potrzeby, bez konieczności tworzenia wszystkiego za każdym razem od nowa. Po dodaniu slajdu, pojawia się on z nazwą szablonu (na przykład Image Slide) na pasku.



Rysunek 7: Panel slajdów

Dodając do slajdu zdjęcie z dysku za pomocą przycisku Image wyświetli się systemowe okno wyboru pliku. Dla mniej obeznanego z technologią użytkownika ważne jest, że nie da się wybrać pliku, który nie jest zdjęciem - podświetlone są tylko te z odpowiednimi rozszerzeniami, a inne formaty nie będą zaakceptowane. Zdjęcia mają najmniej opcji edycji ze wszystkich obiektów, można je tylko przenieść w przód, lub w tył, dodać do nich animacje, oraz usunąć.

Dodając kształt SVG dostępne jest więcej opcji personalizacji obiektu. Dostępna jest zmiana koloru obwódki, który zmieniamy używając próbnika kolorów wbudowanego w przeglądarkę (Chrome) lub system operacyjny (Safari, Firefox), aktualizacja jej grubości w 4 różnych wariantach od poziomu 0 (brak obwódki) do poziomu 4 (największa grubość). Tak samo możliwa jest też zmiana koloru wypełnienia kształtu, również używając tego samego próbnika dostępnego z przeglądarki. Warto wspomnieć, że próbnyki te mogą przyjmować także wartości HEX, czy RGBA. Ostatnią opcją, którą także można zmieniać jest przezroczystość obiektu od 1 do 0 (z dokładnością do 0.01), gdzie 1 oznacza brak przezroczystości a 0 oznacza brak widocznego obiektu.



Rysunek 8: Zaznaczony tekst i opcje jego edycji powyżej

Najwięcej opcji edycji ma obiekt pola tekstowego - Textbox. Poza wcześniej wymienioną zmianą koloru i przezroczystości, użytkownicy mają do dyspozycji pogrubienie tekstu,

kursywę, podkreślenie tekstu, ustawienie go do lewej, środka i prawej strony pola, wybór czcionki z kilku dostępnych w polu typu select, oraz jej wielkość.

Te podstawowe funkcje edycji są najczęściej wykorzystywane przez użytkowników i powinny w zupełności wystarczyć do stworzenia profesjonalnej i estetycznej prezentacji. Edytor zapewnia więc dużą swobodę tworzenia, a jednocześnie jest na tyle prosty, że nie potrzeba nauki, czy poradników by z niego korzystać, jak jest to często w przypadku konkurencyjnych rozwiązań.

5.2 Przeglądarka slajdów

5.3 Eksport do formatu PDF

Rozdział 6

Opis 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 [14], jednak jego wprowadzenie jest mało prawdopodobne. Rozwijany od 2016 roku [15] nie doczekał się jednak wsparcia w jakiegokolwiek przeglądarce.

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 jest już wspierane przez Safari, jednak wymagałoby to bardzo dużych zmian w systemie zapisu plików. Prawdopodobne jest też, że wkrótce safari zacznie wspierać w całości OPFS, jako że jest to dosyć nowe API.

Nowe funkcje dla tekstu

Obiekt tekstowy w tej implementacji posiada tylko jeden obiekt svg, co uniemożliwia zrobienie rzeczy do których przyzwyczailiśmy 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, pochyleń i podkreślenia. Obecnie nie można też zaznaczać tekstu kursorem myszy.

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.

Wprowadzenie systemu pozycjonowania zawartości

Na obecną chwilę obiekty w edytorze ustawiamy ręcznie, więc trudno jest na przykład ułożyć obok siebie zdjęcia, tak aby były równe. Tutaj pomógłby system pozycjonowania zawartości, który pomagałby rozszerzyć zdjęcie do tego samego punktu w którym jest zdjęcie obok.

Bibliografia

- [1] Oscar W. Richards, „The Projection of Chemical Lecture Experiments onto the Screen”, maj 1940. doi: [10.1021/ed017p210](https://doi.org/10.1021/ed017p210).
- [2] David E. Christensen, „Overhead projector use in televising geography”, październik 1964. doi: [10.1080/00221346408985098](https://doi.org/10.1080/00221346408985098).
- [3] Harvey R. Frye, „Handmade materials for projection”. Dostęp: 5 listopad 2024. [Online]. Dostępne na: <https://www.youtube.com/watch?v=tUXIYXnNMFs>
- [4] „Instructor using an overhead projector to teach a lesson”. [Online]. Dostępne na: <https://d.lib.ncsu.edu/collections/catalog/0020075#?c=&m=&cv=&xywh=-1309%2C-160%2C9639%2C5022>
- [5] Clarence M. Flaten, William C. Prigge, i Dennis W. Pett, „Photographic Slides for Instruction”. Dostęp: 5 listopad 2024. [Online]. Dostępne na: <https://www.youtube.com/watch?v=6kgeDd2BDKs>
- [6] D. Austin, „Forethought, a Silicon Valley startup, brought PowerPoint 1.0 to market in April 1987.”. [Online]. Dostępne na: <https://spectrum.ieee.org/the-improbable-origins-of-powerpoint>
- [7] Bob Gaskins, „Forethought: the First 18 Months”. Dostęp: 8 listopad 2024. [Online]. Dostępne na: <https://www.robertgaskins.com/powerpoint-history/documents/gaskins-history-of-forethought-1987-may-25.pdf>
- [8] Meinald T. Thielsch i Isabel Perabo, „Use and Evaluation of Presentation Software”, *Technical Communication* (59), s. 112–129, maj 2012, Dostęp: 8 listopad 2024. [Online]. Dostępne na: http://www.thielsch.org/download/paper/Thielsch_Perabo_2012.pdf
- [9] „Svelte”. Dostęp: 31 październik 2024. [Online]. Dostępne na: <https://en.wikipedia.org/wiki/Svelte>
- [10] „Svelte MIT License”. Dostęp: 31 październik 2024. [Online]. Dostępne na: <https://github.com/sveltejs/svelte/blob/main/LICENSE.md>
- [11] „SVG.js License”. Dostęp: 31 październik 2024. [Online]. Dostępne na: <https://github.com/svgdotjs/svg.js/blob/master/LICENSE.txt>
- [12] „It's speedy - SVG.js”. Dostęp: 31 październik 2024. [Online]. Dostępne na: <https://svgjs.dev/docs/3.2/#it-s-speedy>
- [13] „svg2pdf.js MIT License”. Dostęp: 31 październik 2024. [Online]. Dostępne na: <https://github.com/yWorks/svg2pdf.js/blob/master/LICENSE>
- [14] „Auto-wrapped text”. Dostęp: 31 październik 2024. [Online]. Dostępne na: <https://svgwg.org/svg2-draft/text.html#TextLayoutAuto>

- [15] „Scalable Vector Graphics (SVG) 2, W3C Candidate Recommendation”. Dostęp: 31 październik 2024. [Online]. Dostępne na: <https://www.w3.org/TR/2016/CR-SVG2-20160915/>

Spis rysunków

Rysunek 1: Grafoskop używany na Uniwersytecie North Carolina, lata 60.[4]	2
Rysunek 2: Interfejs użytkownika w pierwszym wydaniu programu PowerPoint [6]	3
Rysunek 3: Menu wstawiania mediów w Apple Keynote	6
Rysunek 4: Widok edytora po wejściu do aplikacji	11
Rysunek 5: Okno nazwy pobieranego pliku	12
Rysunek 6: Pobrany plik prezentacji	12
Rysunek 7: Panel slajdów	13
Rysunek 8: Zaznaczony tekst i opcje jego edycji powyżej	13

Streszczenie

Po polsku

Summary

In english

Słowa kluczowe