

**Wrocławska Wyższa Szkoła Informatyki Stosowanej we Wrocławiu**

**„Kreator Innowacyjności – wsparcie innowacyjnej przedsiębiorczości  
akademickiej”**

*Przedsiębiorczość innowacyjna – Grafika komputerowa*

**Tytuł projektu:**

**Stworzenie pomocy dydaktycznych dla studentów  
kierunków informatycznych**

**Creation of learning aids for computer science students**

**Projekt studialny**

**Autor:**

**Agata Rusoń**

**Wrocław, 2013**

## **Spis treści**

<b>1. Wstęp .....</b>	<b>3</b>
<b>2. Cel i zakres projektu.....</b>	<b>3</b>
<b>3. Część teoretyczna.....</b>	<b>4</b>
I. Etap poznawczy .....	4
i. Opis algorytmu Kruskala.....	5
ii. Opis algorytmu Prima.....	7
II. Etap koncepcyjny .....	9
<b>4. Opis tworzenia i implementacji projektu .....</b>	<b>9</b>
I. Tworzenie tablic początkowych oraz końcowych w Photoshopie .....	9
II. Tworzenie animacji we Flashu .....	11
<b>5. Wnioski .....</b>	<b>13</b>
<b>6. Możliwości rynkowego wykorzystania projektu .....</b>	<b>14</b>
<b>7. Wykaz literatury.....</b>	<b>15</b>
<b>8. Spis rysunków .....</b>	<b>16</b>

## 1. Wstęp

Projekt studialny zrealizowany w ramach warsztatów obejmuje stworzenie pomocy dydaktycznych dla studentów kierunków informatycznych. Pomoce te dotyczą algorytmów grafowych, które studenci poznają oraz implementują około drugiego roku studiów inżynierskich. Nieodłącznym elementem zapoznania się z działaniem algorytmu jest przedstawienie przez prowadzącego przykładu działania danego algorytmu dla konkretnego wybranego grafu. Przykład taki, musi być jednocześnie prosty i pełny, co oznacza, że musi obejmować wszystkie możliwe przypadki działania danego algorytmu, po to, by późniejsza implementacja, powstała na bazie tak przedstawionego przykładu, nie zawierała błędów spowodowanych nie uwzględnieniem szczególnego przypadku. Dwie animacje, które powstały w ramach tego projektu mają za zadanie pomóc studentom zrozumieć działanie algorytmów grafowych, w szczególności algorytmów poszukiwania najmniejszego drzewa rozpinającego tzw. MST (z ang. Minimum Spanning Tree). Student po zajęciach ma możliwość odtworzenia przygotowanych materiałów i odpowiedzenia sobie samodzielnie na pytania lub wątpliwości, a także w przypadku nieobecności, może w ten sposób poznać zagadnienia omawiane na zajęciach.

W projekcie użyto dwóch narzędzi, obydwa są narzędziami firmy Adobe: Photoshop, dzięki któremu stworzono plansze otwierające oraz zamykające animacje oraz Flash Professional CC, który służy do tworzenia animacji. Podczas realizacji projektu należało również wykorzystać wiedzę teoretyczną z zakresu informatyki dotyczącą działania oraz cech szczególnych przedstawianych algorytmów.

## 2. Cel i zakres projektu

Celem projektu było stworzenie pomocy dydaktycznych dla studentów kierunków informatycznych.

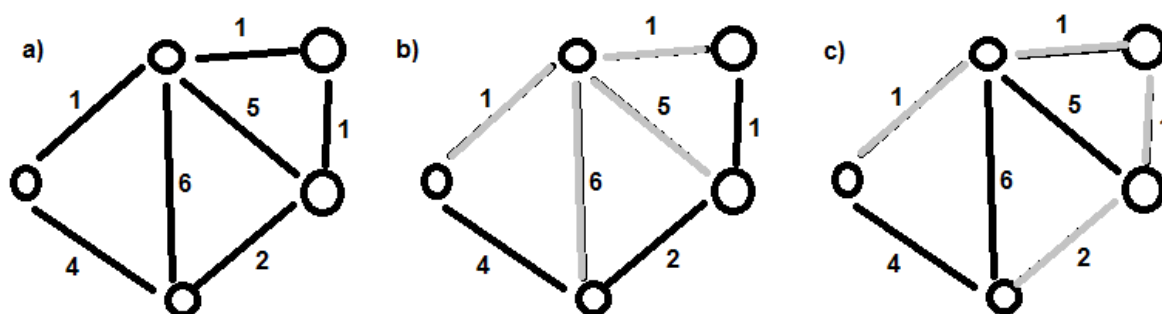
W trakcie realizacji projektu stworzono animacje pokazujące na przykładzie działanie dwóch algorytmów grafowych Kruskala oraz Prima, stworzonych do wyszukiwania minimalnego drzewa rozpinającego w spójnym grafie ważonym nieskierowanym.

### 3. Część teoretyczna

#### I. Etap poznawczy

Etap poznawczy obejmował przede wszystkim zgromadzenie jak najdokładniejszych informacji na temat przedstawianych algorytmów, co umożliwiło stworzenie pomocy naukowych bez błędów merytorycznych. Odwołano się do literatury naukowej z zakresu informatyki oraz opisów działania algorytmów znalezionych na licznych stronach internetowych.

Oba algorytmy Kruskala i Prima są algorytmami poszukiwania w spójnym grafie ważonym i nieskierowanym tzw. MST, czyli minimalnego drzewa rozpinającego (z ang. Minimum Spanning Tree). Dla dowolnego grafu spójnego (dla każdych dwóch wierzchołków istnieje ścieżka je łącząca), ważonego (każda z krawędzi ma nadaną wagę) oraz nieskierowanego (jeżeli istnieje krawędź łącząca dwa wierzchołki  $a$  z wierzchołkiem  $b$ , to istnieje także krawędź łącząca wierzchołki  $b$  z wierzchołkiem  $a$  i jest to ta sama krawędź) można znaleźć tzw. drzewa rozpinające, które są podgrafami tego grafu zawierającymi



Rysunek 1 Ilustracja graficzna drzew rozpinających a) Spójny graf ważony i nieskierowany, b) przykładowe drzewo rozpinające, na szaro zaznaczono krawędzie grafu wchodzące w skład drzewa rozpinającego, jego waga to 13, c) minimalne drzewo rozpinające – drzewo rozpinające o najmniejszej możliwej wadze równej 5.

wszystkie jego wierzchołki oraz minimalną liczbę krawędzi je łączących, przy czym podgraf ten nie może zawierać cykli. Jeżeli graf zawiera  $n$  wierzchołków to minimalna liczba krawędzi łączących te wierzchołki to  $n-1$ , zatem drzewo rozpinające grafu o  $n$  wierzchołkach będzie miało zawsze  $n-1$  krawędzi (Cormen, Leiserson, Rivest i Stein, 2012). Dla dowolnego drzewa rozpinającego można obliczyć jego wagę, która jest sumą wag krawędzi tworzących to drzewo. W każdym spójnym grafie ważonym nieskierowanym można znaleźć jedno lub więcej drzew rozpinających o pewnych wagach, to drzewo, którego waga jest najmniejsza jest

minimalnym drzewem rozpinającym. Na Rysunek 1 można zobaczyć przykładowy graf oraz dwa drzewa rozpinające tego grafu, z których jedno jest drzewem MST.

### **i. Opis algorytmu Kruskala**

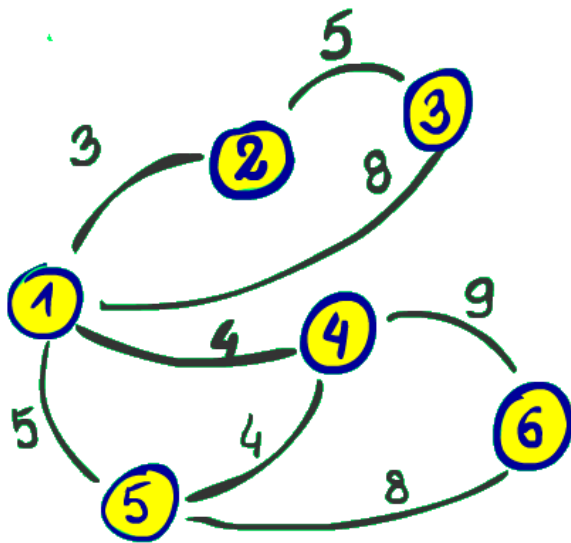
Algorytm Kruskala rozpoczyna się od posortowania niemalejąco wag wszystkich krawędzi w grafie. Sortowanie ustala kolejność rozpatrywania krawędzi grafu w każdym kolejnym kroku. Jak zostało już powiedziane w drzewie rozpinającym nie może być cykli, dlatego zachodzi potrzeba ich detekcji lub unikania w trakcie poszukiwań minimalnego drzewa rozpinającego. Efektywna metoda sprawdzania obecności cykli, użyta w przykładzie opiera się na kolorowaniu wierzchołków grafu. W kroku inicjalizacyjnym algorytmu każdy wierzchołek otrzymuje unikalny kolor. Podczas sprawdzania czy daną krawędź można dołączyć można spotkać się z czterema sytuacjami:

1. Dwa wierzchołki, które łączy badana krawędź mają różne kolory i żadne z wierzchołków nie był jeszcze rozpatrywany, w takim przypadku krawędź może zostać dołączona, ponieważ nie spowoduje to powstania cyklu. Jeden z wierzchołków przyjmuje kolor drugiego i oba stają się „już rozpatrywane”.
2. Dwa wierzchołki, które łączy badana krawędź mają różne kolory, ale jeden z nich był już rozpatrywany, a drugi jeszcze nie. W takim wypadku krawędź może zostać dołączona, nie spowoduje to powstania cyklu, ale wierzchołek jeszcze nie rozpatrywany przyjmuje kolor wierzchołka już rozpatrywanego.
3. Oba wierzchołki mają różne kolory i były już rozpatrywane. Jeżeli wierzchołek był już rozpatrywany, to znaczy, że istnieje przynajmniej jeszcze jeden wierzchołek w jego kolorze, a wszystkie wierzchołki w tym samym kolorze tworzą pewne poddrzewo. Skoro oba były już rozpatrywane i mają różne kolory to są częściami dwóch różnych poddrzew, zatem wszystkie wierzchołki jednego z poddrzew trzeba przekolorować używając koloru drugiego poddrzewa.
4. Ostatnia możliwa sytuacja jest taka, że oba wierzchołki mają ten sam kolor, w takim wypadku krawędź należy odrzucić, ponieważ jej dołączenie spowoduje powstanie cyklu.

**Przykład działania algorytmu Kruskala, który został użyty w animacji.**

Dany jest graf jak na rysunku (**Błąd! Nie można odnaleźć źródła odwołania.**) niżej zostały przedstawione kroki algorytmu.

- Algorytm Kruskala rozpoczyna się od posortowania krawędzi według niemalejących wag. W takiej kolejności krawędzie będą rozpatrywane przez algorytm. Dla grafu o  $n$  wierzchołkach, drzewo rozpinające jest pewnym podgrafem o  $n-1$  krawędziach, który nie zawiera cykli, zatem w trakcie dołączania kolejnych krawędzi należy sprawdzić czy dołączenie danej krawędzi nie spowoduje powstania cyklu. Stąd idea pokolorowania wierzchołków. Na początku algorytmu każdy z wierzchołków ma swój unikalny kolor.



**Rysunek 2 Przykładowy graf spójny, ważony i nieskierowany.**

Sposób detekcji cykli za pomocą kolorów zostanie omówiony w trakcie rozwiązywania przykładu.

- Jako pierwsza rozpatrywana jest krawędź o najmniejszej wadze, czyli krawędź 1,2 o wadze 3. Zarówno wierzchołek pierwszy jak i drugi, nie zostały jeszcze dołączone do drzewa oraz są różnego koloru, zatem po dodaniu tej krawędzi nie powstanie cykl. Dołączamy zatem krawędź 1,2 i wierzchołek pierwszy przyjmuje kolor wierzchołka drugiego lub

odwrotnie. Krawędź 1,2 jest zatem pierwszą krawędzią drzewa rozpinającego.

- Kolejną krawędzią jest 4,5 o wadze 4. Proszę zauważyć, że również krawędź 1,4 ma wagę 4. W takim wypadku kolejność rozpatrywania krawędzi jest dowolna. W tym przykładzie najpierw zostanie rozpatrzona krawędź 4,5. Wierzchołki 4 i 5 mają różne kolory oraz nie zostały jeszcze dołączone do żadnego poddrzewa, zatem sytuacja się powtarza i krawędź 4,5 zostaje dodana, a wierzchołek piąty przyjmuje kolor wierzchołka czwartego.
- Rozpatrzenia dwóch pierwszych krawędzi spowodowało powstanie dwóch poddrzew o różnych kolorach. Zobaczmy co będzie działo się dalej.

- Kolejną rozpatrywaną krawędzią jest krawędź 1,4 o wadze 4. Zarówno wierzchołek pierwszy jak i wierzchołek czwarty były już rozpatrywane, czyli należą do pewnych poddrzew, jednak wciąż mają różne kolory, w związku z tym, krawędź 1,4 może zostać dołączona, jednak jedno z poddrzew musi przyjąć kolor drugiego, dzięki czemu stają się jednym poddrzewem.
- Kolejną krawędzią jest krawędź 1,5 o wadze 5. Oba wierzchołki pierwszy i piąty były już dołączane oraz mają ten sam kolor, w takiej sytuacji dołączenie krawędzi 1,5 spowoduje powstanie cyklu, co jest niedopuszczalne, zatem krawędź tę należy odrzucić, nie będzie ona należała do końcowego drzewa rozpinającego.
- Jako, że dołączono dopiero trzy krawędzie, a dla grafu o sześciu wierzchołkach drzewo rozpinające ma 5 krawędzi, algorytm kontynuuje działanie i bada krawędź 2,3 o wadze 5. Wierzchołki mają różne kolory, zatem krawędź zostanie dołączona, jednak wierzchołek drugi należy już do pewnego poddrzewa, a wierzchołek trzeci. W takiej sytuacji, wierzchołek trzeci przyjmuje kolor wierzchołka drugiego.
- Kolejną krawędzią jest 5,6 o wadze 8. Wierzchołki mają różne kolory, ale wierzchołek piąty należy już do pewnego poddrzewa, a wierzchołek szósty nie. Sytuacja jest zatem podobna jak poprzednio i wierzchołek szósty przyjmuje kolor wierzchołka piątego, co powoduje jego dołączenie do poddrzewa.
- Algorytm kończy swoje działanie, ponieważ dołączył już  $n-1$  krawędzi, które wystarczą do połączenia wszystkich wierzchołków jednym drzewem rozpinającym. Znalezione minimalne drzewo rozpinające ma wagę 24, co stanowi sumę wag krawędzi tworzących to drzewo.

## ii. Opis algorytmu Prima

Algorytm Prima na początku wymaga wybrania jednego z wierzchołków, który nazwiemy wierzchołkiem startowym. Może to być dowolny z wierzchołków grafu, ponieważ jego wybór nie wpływa na optymalność algorytmu. Zawsze bada się tylko krawędzie, które wychodzą z wierzchołków już rozpatrzonych, a kończą się w wierzchołkach jeszcze

nierozpatrzonych. Taka zasada pozwala uniknąć cykli i dzięki niej nie ma potrzeby ich detekcji. Z badanych krawędzi wybiera się tę, która ma najmniejszą wagę i ją dołączamy do drzewa rozpinającego. Należy zauważyć, że w każdym kroku jeden z wierzchołków nierozpatrzonych staje się wierzchołkiem rozpatrzonym, a wiedząc, że drzewo rozpinające musi zawierać wszystkie wierzchołki grafu wiemy dokładnie w ilu krokach algorytm zakończy swoje działanie – jest to  $n-1$  kroków.

Poniżej przedstawiono przykład działania algorytmu Prima, ten sam, który został umieszczony w animacji. Graf wzięty do przykładu znajduje się na Rysunek 2, poniżej przedstawiono kroki algorytmu.

- Na początku wszystkie wierzchołki są nierozpatrzone. Na wierzchołek startowy wybierzemy wierzchołek szósty. Z tego wierzchołka wychodzą dwie krawędzie do wierzchołków jeszcze nie rozpatrzonych: 4,6 oraz 5,6, zaznaczone na żółto. Z tych dwóch krawędzi wybieramy tę, która ma mniejszą wagę i dołączamy ją jako pierwszą do drzewa rozpinającego zaznaczonego na czerwono.
- Kolejnym etapem jest znalezienie krawędzi, które zaczynają się w wierzchołkach już rozpatrzonych, w tym przypadku piątym i szóstym, a kończą w wierzchołkach jeszcze nie rozpatrzonych. Są to odpowiednio krawędzie 1,5, 4,5 oraz 4,6. Tym razem z trzech krawędzi wybieramy tę o najmniejszej wadze i dołączamy ją do drzewa, jest to krawędź 4,5. Jeżeli dwie krawędzie mają takie same wagi, to dołączamy dowolną z nich.
- Nasze drzewo ma już dwie krawędzie i rozpatrzyliśmy trzy wierzchołki. Procedura się powtarza. Krawędzie, które wychodzą z wierzchołków rozpatrzonych i kończą się w wierzchołkach nierozpatrzonych to 1,5 i 1,4. Wybieramy krawędź 1,4 o wadze 4 i dołączamy ją do drzewa.
- Wciąż niektóre wierzchołki nie zostały rozpatrzone, dlatego algorytm kontynuuje swoje działanie. Interesujące nas krawędzie to 1,2 oraz 1,3. Mniejszą wagę ma krawędź 1,2, dlatego dołączamy ją do drzewa.



- Pozostał już tylko jeden nierozpatrzony wierzchołek, zatem bierzemy pod uwagę krawędzie 1,3 oraz 2,3. Mniejszą wagę równą 5 ma krawędź 2,3 i to ona zostaje dołączona jako ostatnia krawędź drzewa rozpinającego.
- Wszystkie wierzchołki zostały rozpatrzone, dlatego algorytm kończy działanie, a znalezione drzewo jest minimalnym drzewem rozpinającym i ma wagę 24.

## **II. Etap koncepcyjny**

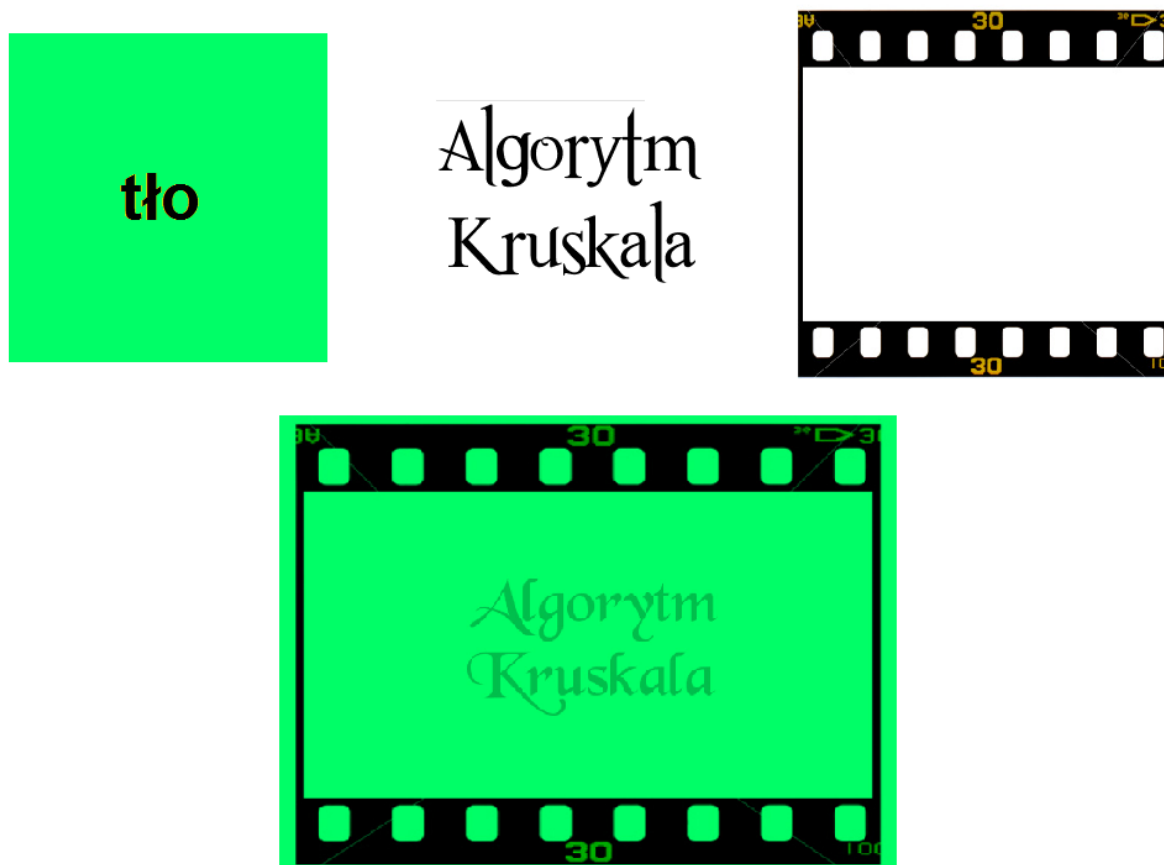
W trakcie warsztatów poznaliśmy wiele narzędzi, które służą z powodzeniem grafikom do realizacji projektów typowo graficznych. Tematyka tego projektu jednak nie jest typowo graficzna, ponieważ grafika w tym przypadku jest tylko narzędziem umożliwiającym przekazanie pewnej wiedzy, zatem podczas realizacji tego projektu celem nie było zadziwienie odbiorcy kunsztem i koncepcją, a tylko przejrzyste przedstawienie treści. Zatem użyto programu Photoshop nie po to, by pokazać jego ogromne możliwości wykorzystania w szeroko rozumianej grafice komputerowej, ale by pokazać, że autor tego projektu posiadał umiejętność pracy z nim. Głównym programem wykorzystywanym przy realizacji projektu był Adobe Flash Professional, który jest narzędziem umożliwiającym tworzenie animacji. Program ten, jego działanie oraz możliwości został przedstawiony na zajęciach i stąd pomysł jego użycia w tym przypadku. Animacja ta nie jest skomplikowana, ale dzięki dodaniu narratora staje się rzeczywiście użytecznym narzędziem, z którego mogą skorzystać studenci.

## **4. Opis tworzenia i implementacji projektu**

Do realizacji projektu użyto dwóch najnowszych wersji narzędzi firmy Adobe, czyli Photoshopa oraz Flasha Professional. Z racji tego, że programy te są drogie, pracowano na 30-dniowej wersji próbnej. Dodatkowo w trakcie realizacji projektu został użyty specjalny tablet dla rysowników, który doskonale współpracuje z oboma programami.

### **I. Tworzenie tablic początkowych oraz końcowych w Photoshopie**

Tworzenie tablic wymagało na początku ustalenia wielkości tych tablic we właściwościach dokumentu, wymiary te zostały ustalone na 1024 x 800 pikseli. Od razu ujednolicono także tło planszy na zielone, takie jak w reszcie animacji. Koncepcja wymagała wyszukiwania grafiki przedstawiającej 35 mm film do aparatu, co udało się znaleźć na stronie (wdstock, 2013). Grafika została umieszczona na osobnej warstwie, by zwiększyć możliwości manipulacji nią, jednocześnie uniezależniając jej wygląd od innych elementów. Następnie zaczęły się poszukiwania ciekawej czcionki. Bardzo wiele autorskich czcionek można znaleźć

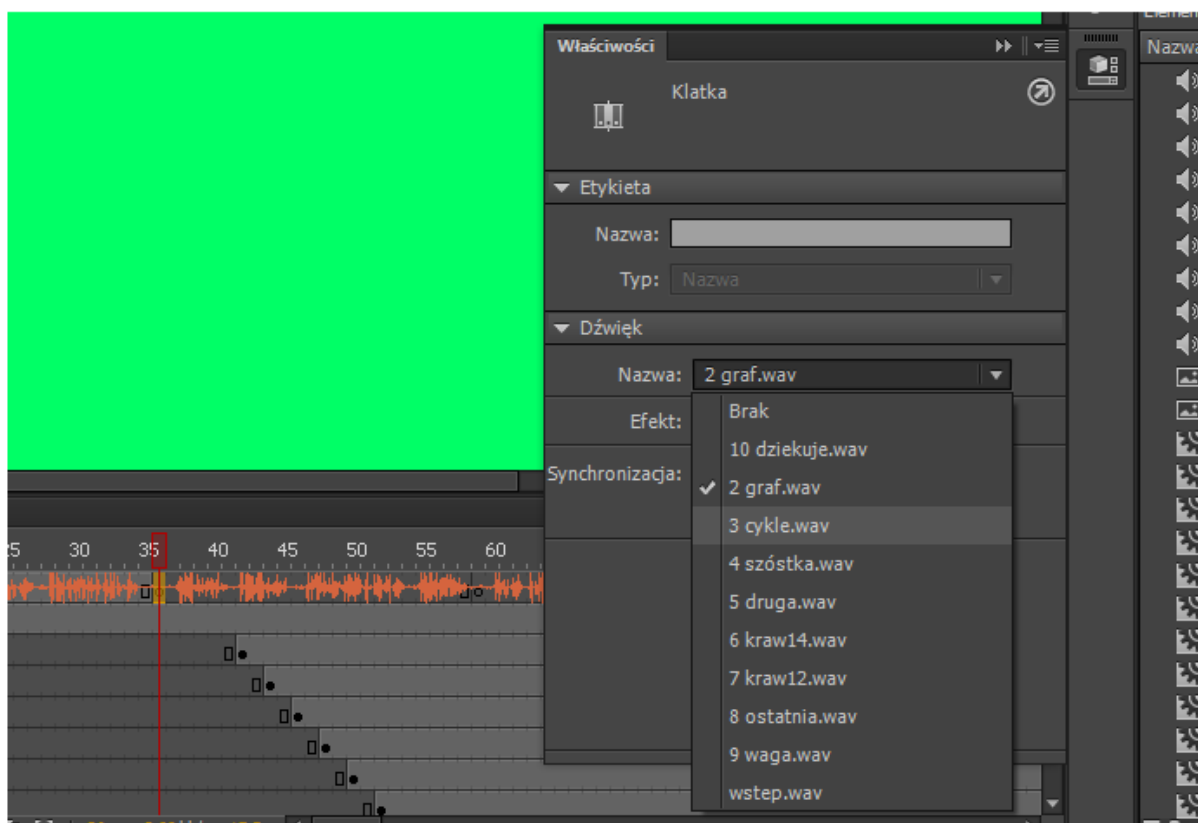


**Rysunek 3** Trzy elementy składowe projektu planszy początkowej algorytmu Kruskala oraz wykonana w Photoshopie gotowa plansza.

na stronie dafont.com, gdzie od razu jest możliwość wpisania interesującego nas tekstu i sprawdzenia czcionki na własnym przykładzie. Najciekawszą czcionką była czcionka stworzona przez Juana Casco pod nazwą Romance Fatal Serif, którą można znaleźć w (Casco, 2013). Napis również został umieszczony na osobnej warstwie. Na poniższym rysunku pokazano wszystkie elementy oraz końcowy efekt. Aby pozbyć się białego tła w grafice użyto mnożenia warstw, aby zmniejszyć kontrast między tłem a napisem zmieniono współczynnik alfa napisu na 20%, co spowodowało zwiększenie przeźroczystości.

## II. Tworzenie animacji we Flashu

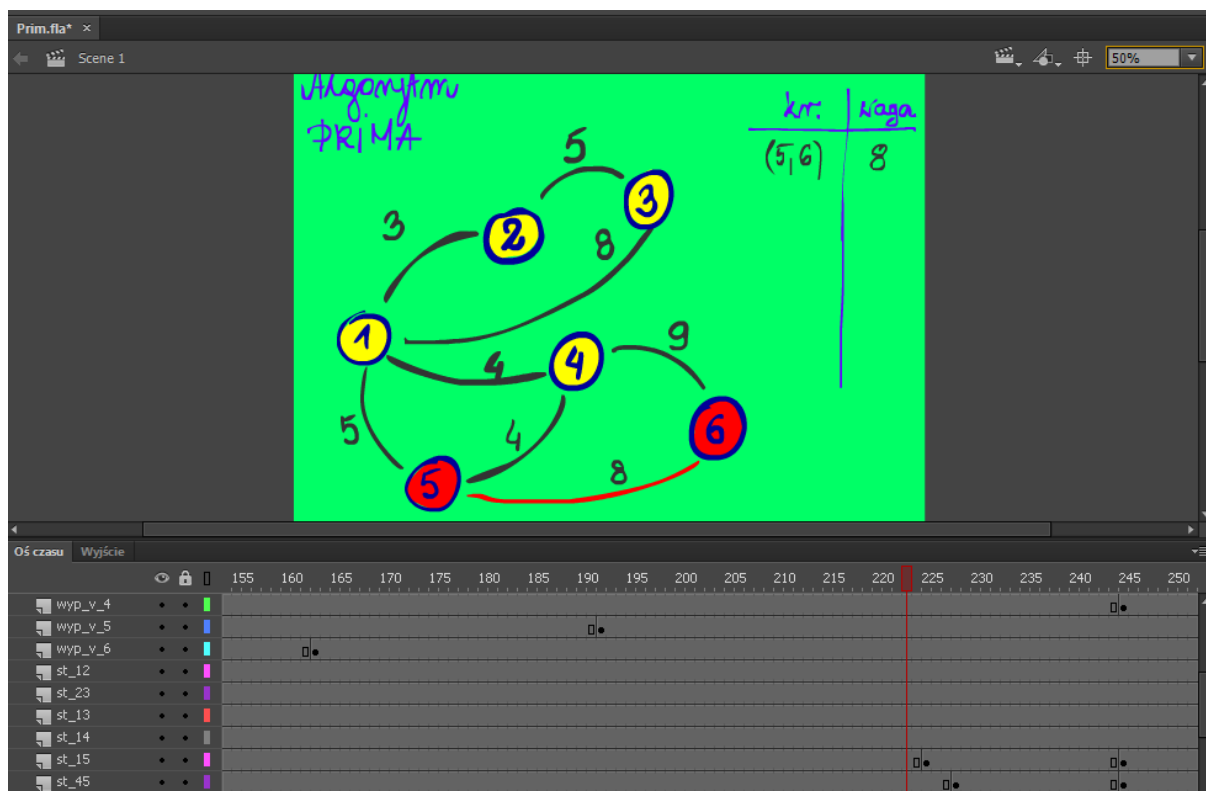
Tworzenie animacji we Flashu wymaga od projektanta bardzo dużej uwagi, a przed wszystkim znajomości pewnych mechanizmów, które rządzą tym programem oraz



Rysunek 4 Sposób dodawania dźwięku we Flashu. Dźwięk dodaje się do wybranej 36 klatki w jej właściwościach.

ograniczeń, które sprawiają, że niektóre zmiany koncepcyjne będą wymagały przebudowy całego projektu. Pierwszym najważniejszym wnioskiem z pracy z programem Adobe Flash jest to, że każdy element, który będzie w pewien sposób aktywny, powinien być na osobnej warstwie. Koncepcja tego projektu wymagała zmian kolorów wierzchołków grafu w trakcie animacji, niektóre elementy pojawiały się później/wcześniej niż inne. Te wszystkie pomysły sprawiły, że na samym początku tworzenia animacji należało utworzyć ok. 40 warstw. Na każdej z warstw umieszczono jeden element. Koncepcja zakładała, że cała animacja ma przypominać wykład przy tablicy, stąd pomysł, aby cały graf wyglądał jakby został narysowany ludzką ręką. Elementy tego grafu zostały zatem stworzone za pomocą tabletu dla rysowników z rysikiem i dzięki temu wyglądają jakby właśnie zostały narysowane flamastrem. Uzyskano zamierzony efekt. Cała animacja byłaby bezużyteczna, gdyby nie

komentarz, który został nagrany w bardzo dobrym i najważniejsze darmowym programie Audacity 2.0.3. Narratorem jest autorka projektu. Wykorzystując doświadczenia zdobyte w pracy nauczyciela udało się stworzyć syntetyczną i jasną wypowiedź, komentującą wybrany przykład. Narracja musiała być dobrze przemyślana, ponieważ do niej została dostosowana cała animacja. Dźwięk został nagrany we fragmentach i w ten sam sposób umieszczony w prezentacji. Tempo głosu i moment powiedzenia dane słowa wyznaczały moment animacji. Plik dźwiękowy z rozszerzeniem .wav (tylko takie rozszerzenie jest obsługiwane) należy



Rysunek 5 Zrzut ekranu przedstawiający fragment osi czasu dla wybranych warstw, umiejscowienie klatek kluczowych oraz wygląd jednej z ramek.

najpierw zaimportować na stół montażowy (Plik -> Importuj -> Importuj na stół montażowy...), a następnie dodać we właściwościach danej klatki tak jak pokazuje Rysunek 4. Aby nie generować niepotrzebnej liczby klatek, z racji tego, że ta animacja nie wymagała dużej rozdzielności w czasie (24 kl/s to zbyt dużo), zmieniono liczbę klatek na 2 na sekundę, co w zupełności wystarczyło. Dodanie dźwięku w jednej warstwie jest jednoznaczne z dodaniem dźwięku do całej animacji.

Po umieszczeniu dźwięku oraz stworzeniu poszczególnych elementów na różnych warstwach, należało do każdej warstwy dodać indywidualną dla danego elementu animację.

Ten etap był etapem najbardziej skomplikowanym i wymagającym. Należało prześledzić zachowanie się danego elementu w trakcie animacji/narracji i ustalić pewne klatki kluczowe, które rozgraniczały zmiany w wyglądzie obiektów. Fragment osi czasu dla wybranych warstw został przedstawiony na Rysunek 5.

Po zanimowaniu wszystkich warstw (elementów) plik został wyeksportowany, najpierw jako plik swf (Plik -> Eksportuj -> Eksportuj film...), a później jako plik filmowy mov (Plik -> Eksportuj -> Eksportuj wideo...). Rozszerzenie swf jest to rozszerzenie animacji flasha, które odtwarzane są za pomocą programu Adobe Flash Player. Pliki to są bardzo małe, żadna ze stworzonych w ramach tego projektu animacji nie miała po zapisaniu jej jako plik swf więcej niż 1 MB. Jakość wyświetlania tej animacji jest bardzo dobra, jak na tak mały plik trwający ok. 4 minut. Animacja została także zapisana jako plik z rozszerzeniem mov jako film o rozdzielczości 1024 x 800 pikseli. Ten zapis był dużo mniej efektywny pamięciowo, ponieważ 4 minutowy klip miał ok 700 MB. Obydwie animacje zostały wrzucone na portal youtube.pl jako pliki niepubliczne. Oznacza to, że dostęp do nich ma każdy, kto posiada link. Linki do poszczególnych animacji są następujące:

**Algorytm Kruskala:** <http://www.youtube.com/watch?v=UnEGd4m8-uo>

**Algorytm Prima:** <http://www.youtube.com/watch?v=fjMSreoCb7c>

## 5. Wnioski

Wnioski poznawcze z realizacji tego projektu na pewno obejmują zapoznanie się z narzędziami, które zostały użyte do jego stworzenia. Poznanie obu programów dotyczyło przede wszystkim poznania ich słabych stron, które wymuszają pewien konkretny sposób pracy z każdym z nich. Programy firmy Adobe, zarówno Photoshop, jak i Flash korzystają z tzw. warstw. W niektórych przypadkach takie podejście jest wygodne, ale w przypadku tego projektu ilość pracy włożonej w stworzenie tych warstw była niewspółmierna z rezultatami. Przy prostych animacjach, takich jak zmiana koloru obiektu, przesunięcie go przy wielu różnych elementach program bardzo się rozbudowuje i nie ma możliwości jego uproszczenia.

Wniosków praktycznych jest więcej. Tworzenie tego projektu pokazało, że relatywnie małe nakłady pracy, które należało włożyć w przygotowanie takich pomocy, mogą bardzo szybko zaowocować lepszymi ocenami z kolokwium, mniejszą liczbą popraw. Taka animacja

z odpowiednim komentarzem bardzo dobrze zastępuje pełne zajęcia i może służyć zarówno jako pomoc, dla tych, którzy na zajęciach byli i interesowali się ich przebiegiem, jak i dla tych, którzy z różnych powodów nie mogli w tych zajęciach uczestniczyć. Znajomość flasha jest przydatne jednak nie tylko w pracy dydaktycznej, ale również w pracy naukowej, ponieważ pomoce wizualne dużo bardziej docierają do osób słuchających. W przypadku pracowników naukowych, którzy specjalizują się w bardzo wąskich dziedzinach, ciężko jest wytłumaczyć, przedstawić wyniki swoich badań, ponieważ wiąże się to z pewną wiedzą w ściśle określonym temacie, którą musi posiadać słuchacz. Dzięki takim pomocom, można przedstawić dużo więcej w krótkim czasie i zainteresować nawet osoby, które wcześniej na dany temat nie wiedziały wiele.

## **6. Możliwości rynkowego wykorzystania projektu**

Oprócz oczywistych możliwości rynkowego wykorzystania projektu, czyli sprzedaży konkretnych animacji w sklepie internetowym lub w dowolny inny sposób, należy zwrócić uwagę na to, że takie pomoce mogłyby być użyteczne w bardzo wielu dziedzinach. Osoba, która tworzy taką animację, wcale nie musi się bardzo dobrze znać na temacie, w którym działa. Wystarczy, że otrzyma pewne wytyczne od osoby, która jest ekspertem w danej dziedzinie. Projekt zatem można byłoby rozszerzyć i znaleźć ekspertów w danych dziedzinach, którzy potrafiliby określić temat i podać zawartość merytoryczną takiej animacji. Oprócz informatyki, takie pomocy mogłyby powstać w biologii, chemii, mechanice i innym dowolnym temacie, pod warunkiem, że da się dany proces, zjawisko pokazać krok po kroku. Dzięki takiej współpracy można by stworzyć bazę takich pomocy i udostępniać je osobom zainteresowanym. Takie projekty, których elementem jest wiedza naukowa są projektami wysoko cenionymi zarówno pod względem społecznym, jak i finansowym, jest to element tzw. „know how”. Taki projekt wycenia się na podstawie ich przydatności, a nie ilości pracy w nie włożonej. Taki projekt, w którym łączy się różne dziedziny tworząc bazę wiedzy powstał w 2013 roku na Wydziale Mechanicznym Technologicznym Politechniki Śląskiej pod nazwą InterUdu3D. Cele, zadania i rezultaty tego projektu, dofinansowanego przez Unię Europejską w ramach programu Kapitał Ludzki można znaleźć na stronie [www.interedu3d.pl](http://www.interedu3d.pl).

## 7. Wykaz literatury

1. Adobe Fl 2. (2013, sierpień 31). <http://adobe.com/pl/>. Pobrano z lokalizacji Flash Professional: <http://www.adobe.com/pl/products/photoshop.html>
2. Adobe Fl. (2011). *Adobe Flash CS5/SC5 PL Professional. Oficjalny podręcznik*. Gliwice: Helion.
3. Adobe PS 1. (2013, sierpień 31). <http://adobe.com/pl/>. Pobrano z lokalizacji Photoshop: <http://www.adobe.com/pl/products/photoshop.html>
4. Adobe PS. (2004). *Adobe Flash CS5/SC5 PL Professional. Oficjalny podręcznik*. Gliwice: Helion.
5. Casco, J. (2013). *dafont.com*. Pobrano z lokalizacji Romance Fatal Serif: <http://www.dafont.com/romance-fatal-serif.font?text=Algorytm+Kruskala>
6. Cormen, T. H., Leiserson, E., Rivest, R. i Stein, C. (2012). *Wprowadzenie do algorytmów*. Warszawa: Wydawnictwo Naukowe PWN.
7. wdstock. (2013). *iStock*. Pobrano z lokalizacji <http://www.istockphoto.com/stock-photo-5709042-35mm-color-positive-slide-film-frame-isolated-on-white.php>

## 8. Spis rysunków

1. Rysunek 1 Ilustracja graficzna drzew rozpinających a) Spójny graf ważony i nieskierowany, b) przykładowe drzewo rozpinające, na szaro zaznaczono krawędzie grafu wchodzące w skład drzewa rozpinającego, jego waga to 13, c) minimlane drzewo rozpinające – drzewo rozpinające o najmniejszej możliwej wadze równej 5...4
2. Rysunek 2 Przykładowy graf spójny, ważony i nieskierowany..... 6
3. Rysunek 3 Trzy elementy składowe projektu planszy początkowej algorytmu Kruskala oraz wykonana w Photoshopie gotowa plansza. .... 10
4. Rysunek 4 Sposób dodawania dźwięku we Flashu. Dźwięk dodaje się do wybranej 36 klatki w jej właściwościach..... 11
5. Rysunek 5 Zrzut ekranu przedstawiający fragment osi czasu dla wybranych warstw, umiejscowienie klatek kluczowych oraz wygląd jednej z ramek..... 12