Nie tylko Power Point

Piotr Bolek
ul. Szkolna 15
05-180 Pomiechówek
Phone: (48) 22 – 785 43 39
P.Bolek@GUST.org.pl

Streszczenie

Tekst jest opisem opracowanego przez autora pakietu `pdfslides` służącego do przygotowywania prezentacji w formacie PDF. Pakiet może być wykorzystywany zarówno przez użytkowników formatu `plain` czy MeX, jak i LaTeX. Wyjściowy plik PDF może być generowany programem Adobe Acrobat Distiller, `ps2pdf` (czyli `ghostscipt`) albo bezpośrednio z TeX-a przy użyciu programu pdfTeX.

Motywacja

Każdy przygotowujący wystąpienie, prezentację, seminarium, kurs czy wykład staje przed problemem wyboru sposobu przygotowania materiałów do wystąpienia. Czasy slajdów pisanych odręcznie na przezroczystej folii odchodzą w przeszłość. Normą staje się przygotowywanie prezentacji w formie elektronicznej i to nie po to aby ją wydrukować na folii lecz po to by pokazywać ją bezpośrednio z komputera. O ile do przygotowania prezentacji foliowych istnieje wiele możliwości wykorzystania TeX-a (standardowa klasa LaTeX-owa `slides`, pakiety `foiltex`, `seminar`), to do przygotowania porządnej prezentacji elektronicznej nie istniało do tej pory łatwe w użyciu narzędzie. Narzędzie takie powinno pozwalać na wykorzystanie możliwości jakie daje prezentacja bezpośrednio z komputera: np. „odsłanianie" slajdów po kawałku, animacje, efektowne przejścia między slajdami, możliwość „skoczenia" lub powrotu do dowolnego slajdu.

Standardy w tej dziedzinie wyznacza program pewnej dużej firmy. Dlaczego jednak użytkownicy TeX-a mieliby zmieniać swoje upodobania i przyzwyczajenia, i używać programów, z którymi pracuje się zupełnie inaczej niż z ich ulubionym narzędziem. I nie jest to oczywiście tylko kwestia czyjegoś widzimisię, cała sprawa ma bardzo konkretny aspekt praktyczny. Jeśli prezentacja jest związana z publikacją, która była przygotowana w TeX-u to dlaczego użytkownik miałby wszystkie wzory wpisywać jeszcze raz, albo używać jakichś trików typu – konwersja DVI na PCX i wklejanie bitmap do prezentacji. TeX może być znakomitym narzędziem do przygotowywania multimedialnych prezentacji. Jest to poniekąd także zasługą firmy Adobe, która opracowała format PDF i udostępniła jego specyfikację oraz bezpłatną przeglądarkę. Acrobat Reader jest całkiem sympatycznym narzędziem do realizacji prezentacji bezpośrednio z komputera. Potrafi pokazywać dokument w trybie pełnoekranowym, realizuje efektowne przejścia między stronami, obsługuje odnośniki hipertekstowe. W połączeniu z TeX-em stanowi potencjalnie sensowną alternatywę dla programu wymienionego w tytule tego tekstu.

Ale przejdźmy do konkretów. Zacznę od przedstawienia założeń jakie przyjąłem przy tworzeniu pakietu do przygotowywania prezentacji w formacie PDF przy użyciu TeX-a.

Założenia

Podstawowe założenia przy implementacji pakietu były następujące:

1. Możliwość używania niezależnie od formatu i wersji TeX-a. Konkretnie chodziło o to aby można go było używać w plainie (czyli także MeX-u) oraz w LaTeX-u zarówno w „zwykłym" TeX-u jak i pdfTeX-u. Z tym wiążą się być może pewne ograniczenia – np. niezgodność z pakietem `hyperref` oraz używanie własnych, specyficznych makr do obsługi kolorów. Niezależność od formatu i wersji TeX-a nie oznacza oczywiście, że przygotowana z wykorzystaniem pakietu prezentacja może być zawsze przetworzona bez żadnych modyfikacji dowolną wersją TeX-a z dowolnym formatem. Oznacza to tylko tyle, że pakiet stanowi podstawę uwalniającą użytkowników TeX-a/LaTeX-a od konieczności wgłębiania się w tajniki PDF-a, a jednocześnie pozwala wykorzystać możliwości jakie daje taki sposób tworzenia prezentacji.
2. Ustanowienie szablonu prezentacji czyli dostosowanie wymiarów i proporcji strony do ekranu oraz definicja części stałych – w praktyce paska na dole z definiowanym przez użytkownika tekstem oraz numerem slajdu.
3. Obsługa kolorów – jednakowy we wszystkich „trybach pracy" (rozumianych w sensie formatu i wersji

TEX-a) sposób definiowania kolorów, możliwość definiowania oraz zmiany kolorów tła, tekstu i części stałych.
4. Wsparcie dla definiowania graficznych znaczników zamiast numerów stron.
5. Możliwość definiowania sposobu przejścia między stronami przez wykorzystanie mechanizmu dostępnego w formacie PDF (ang. *transitions*) oraz określania czasu wyświetlania poszczególnych stron i automatyczne przechodzenie do kolejnych stron.
6. Wsparcie do tworzenia „animacji” i „odsłanianych” stopniowo slajdów.
7. Możliwość definiowania i wykorzystywania odnośników hipertekstowych: „aktywny” spis slajdów i definiowanie przez użytkownika odnośniki między slajdami.

Opis pakietu

Uwagi wstępne. Pakiet automatycznie wykrywa wersję formatu i wersję TEX-a, w której jest wywołany. Pakiet jest w zasadzie niezależny od innych, bardziej czy mniej standardowych, stylów i pakietów. Jedynymi wymaganymi do działania dodatkowymi plikami są `colordvi.tex` w zwykłej wersji TEX-a oraz `pdfcolor.tex` w przypadku używania pdfTEX-a.

W obecnej wersji pakiet jest niezależny od LATEX-owego pakietu `color` i może być używany razem z nim, z tym że do definiowania koloru tła oraz paska na dole slajdu należy używać makr z pakietu `pdfslides`. Pakiet może być niezgodny z pakietem `hyperref` oraz makrami `tp-pdf`, ponieważ w implementacji wykorzystywane są fragmenty kodu pochodzące z tychże pakietów (bezpośrednio z `tp-pdf`, gdzie były wzięte z `hyperref`)[1]. Dlatego odradzane jest używanie ich razem, zresztą wydaje mi się, że nie ma takiej potrzeby. Jeśli ktoś uważa, że się mylę, to proszę o odzew.

Struktura pliku. Prezentacja składa się z definicji globalnych parametrów prezentacji oraz kolejnych slajdów definiowanych w zależności od formatu w różny sposób. Ogólny schemat pliku wejściowego jest następujący:

```
%%% PLAIN
\input pdfslides.sty
  % Ustawienie parametrów globalnych
  % prezentacji
\beginslides
\beginslide
  % treść slajdu pierwszego
  % + lokalne parametry dotyczące slajdu
\endslide
  % ...
  % ew. zmiana parametrów globalnych
\beginslide
  % treść slajdu n-tego
  % + lokalne parametry dotyczące slajdu
\endslide
\endslides
\bye
```

```
%%% LaTeX
  % Zwykła preambuła LaTeX-owa
\documentclass{article}% można używać
                        % opcji np. [12pt]
\usepackage{pdfslides}
  % inne pakiety np polski, color,
  % graphics itd.
\begin{document}
  % Ustawienie parametrów globalnych
  % prezentacji
\begin{slides}
\begin{slide}
  % treść slajdu pierwszego
  % + lokalne parametry dotyczące slajdu
\end{slide}
  % ...
  % ew. zmiana parametrów globalnych
\begin{slide}
  % treść slajdu n-tego
  % + lokalne parametry dotyczące slajdu
\end{slide}
\end{slides}
\end{document}
```

[1] Uwaga ta dotyczy normalnej wersji TEX-a. W przypadku pdfTEX-a niebezpieczeństwo niekorzystnych interakcji z pakietem `hyperref` jest ograniczone, a z `tp-pdf` zupełnie nie istnieje, ponieważ jest on przeznaczony dla ścieżki przetwarzania TEX – dvips – distiller.

Polecenia dla użytkownika. Poniżej są opisane wszystkie dostępne dla użytkownika polecenia zdefiniowane w pakiecie. Terminem „środowisko” będę określał środowiska w sensie LATEX-owym oraz pary makr TEX-owych `\begin`⟨*xxx*⟩ i `\end`⟨*xxx*⟩, których można używać w plainie.

- Środowisko `slides` (polecenia `\beginslides`, i `\endslides` w plainie). Środowiskiem tym powinny być objęte wszystkie slajdy wchodzące w skład prezentacji. W rzeczywistości użycie tego środowiska nie jest absolutnie konieczne (choć zwykle wskazane) – w chwili obecnej powoduje ono m.in. dodanie ostatniego slajdu z wycentrowanym definiowanym przez użytkownika napisem (typu „The End” czy „Koniec”) oraz zapamiętanie liczby stron prezentacji. W kolejnych wersjach pakietu środowisko to być może będzie spełniało dodatkowe funkcje, np. produkowało stronę tytułową – w obecnej wersji nie ma żadnego wsparcia do tworzenia strony tytułowej – w szczególności nie zaleca się używania LaTeX-owego polecenia `\maketitle`.

- Środowisko `slide` (polecenia `\beginslide`, i `\endslide` w plainie). Środowisko definiujące slajd. Powinno w zasadzie zawierać materiał mieszczący się na jednej stronie. Jeśli tekst nie zmieści się na jednej stronie, to zostanie utworzona następna. W ten sposób liczba stron w prezentacji może być większa niż liczba środowisk `slide` w pliku źródłowym. Zalecana jest jednak taka organizacja pliku źródłowego aby cały materiał umieszczony wewnątrz środowiska `slide` mieścił się na jednej stronie prezentacji. W przeciwnym przypadku działanie poleceń określających sposób przejść (ang. *transitions*) między stronami oraz czas wyświetlania strony mogą być interpretowane niewłaściwie.
- `\foottext#1` – tekst, który będzie wyświetlany w pasku na dole strony (pasek tytułowy). Tekst będzie umieszczony na środku wiersza. Można tu oczywiście podać także polecenia zmiany wielkości i kroju pisma.
- `\endtext#1` – tekst, który będzie umieszczony na ostatnim slajdzie. Slajd ten jest tworzony automatycznie tylko w przypadku użycia środowiska `slides` – nie trzeba go wtedy tworzyć ręcznie. Slajd ten nie zawiera paska z tytułem prezentacji na dole strony.
- `\footlinecolor#1` – kolor paska na dole strony. Parametr musi być wywołaniem jednego z poleceń wyboru koloru (`\rgbcolor` albo `\cmykcolor`).
- `\foottextcolor#1` – kolor tekstu w pasku na dole slajdu. Parametr musi być wywołaniem jednego z poleceń wyboru koloru (`\rgbcolor` albo `\cmykcolor`).
- `\bgcolor#1` – kolor tła slajdu. Parametr musi być wywołaniem jednego z poleceń wyboru koloru (`\rgbcolor` albo `\cmykcolor`).
- `\pagenum#1` – tekst umieszczany z prawej strony paska tytułowego (w prawym dolnym rogu slajdu). Domyślnie jest to po prostu kolejny numer slajdu (numer strony).
- `\lastpage` – makro, w którym zapamiętana jest liczba slajdów. Wartości tego makra można używać np. przy konstruowaniu „zegarka”, który będzie umieszczany zamiast numeru strony w linii tytułowej, albo do definiowania numeru strony typu „str. 3 z 10” (`\pagenum{str. \the\count0\ z~\lastpage}`). Numer strony jest zapamiętywany dwuprzebiegowo, przez zapisanie numeru ostatniej strony w zewnętrznym pliku. Numer ostatniej strony jest zapamiętywany przy zamykaniu środowiska `slides` – w przypadku nieużycia tego środowiska polecenie `\lastpage` zwraca zawsze wartość 0 (zero).
- `\startpage#1` – podanie strony, która ma być wyświetlana jako pierwsza. Może to być przydatne w czasie przygotowywania prezentacji – jeśli pracujemy np. nad slajdem 28, to po otwarciu pliku Acrobat może skoczyć od razu do tej strony. Zdefiniowanie strony startowej innej niż pierwsza może być też wskazane, jeśli chcemy mieć w naszej prezentacji spis slajdów, którego nie chcemy domyślnie pokazywać. Wtedy umieszczamy spis slajdów na pierwszej stronie, a właściwa prezentacja zaczyna się od drugiej.
- `\listofslides` – umieszczenie w dokumencie listy slajdów. Polecenie to powinno być użyte oczywiście wewnątrz środowiska `slide`. Lista slajdów zawiera aktywne odnośniki do wybranych slajdów. Slajdy nie są dodawane do spisu automatycznie.
- `\addtolos#1` – dodanie bieżącego slajdu do spisu slajdów. Parametr jest tekstem, który pojawi się w spisie (nie na slajdzie).
- `\transition#1#2...` – określenie sposobu przejścia między stronami. Pierwszy parametr określa rodzaj przejścia (jeden opisanych w [2] lub [4]), drugi czas trwania procesu przejścia (w sekundach). Niektóre rodzaje przejść wymagają podania jednego lub dwóch dodatkowych parametrów. Dopuszczalne sposoby przejść oraz ich parametry są podane w tabeli 1. Jeśli dla danej strony nie zostanie podany sposób przejścia, to domyślnie jest to przejście typu `Replace` (`/R` w terminologii PDF).

Przejście	Arg. #3	Arg. #4
`Split`	H albo V	I albo O
`Blinds`	H albo V	—
`Box`	I albo O	—
`Wipe`	0, 90, 180, 270	—
`Dissolve`	—	—
`Gliter`	0, 270, 315	—
`Replace`	—	—

Tabela 1. Parametry przejść między stronami

- `\duration#1` – określenie czasu wyświetlania bieżącego slajdu (w sekundach). Po upłynięciu podanego czasu nastąpi przejście do kolejnego slajdu.
- `\rgbcolor#1` – polecenie służące do wybrania bieżącego koloru w modelu RGB. Parametr powinien być trójką liczb z przedziału [0..1] rozdzieloną spacjami, lub przecinkami (np. `\rgbcolor{1,.3,0}`, `\rgbcolor{1. 3 0.1}`).
- `\cmykcolor#1` – j.w. z tym, że kolor w modelu CMYK w związku z czym parametr musi zawierać cztery liczby.
- `\PDFcolorbox#1#2` – umieszczenie tekstu w kolorowym pudełku. Pierwszy parametr powinien być wywołaniem jednego z poleceń wyboru koloru (`\rgbcolor` albo `\cmykcolor`), a drugi zawartością pudełka. Domyślnym kolorem tekstu w pudełku jest czarny – można to zmienić używając jednego z poleceń wyboru koloru. W przypadku LaTeX-a możliwe jest oczywiście definiowanie pudełek z wykorzystaniem poleceń udostępnianych przez pakiet `graphics`.
- `\PDFdestination#1` – zdefiniowanie miejsca, do którego można się odwołać poleceniem `\PDFhyperlink`. Parametr jest etykietą.
- `\PDFhyperlink#1#2` – utworzenie łącza hipertekstowego. Pierwszy parametr jest etykietą, która musi być zdefiniowana poleceniem `\PDFdestination`, drugi tekstem, który stanie się „aktywny”.

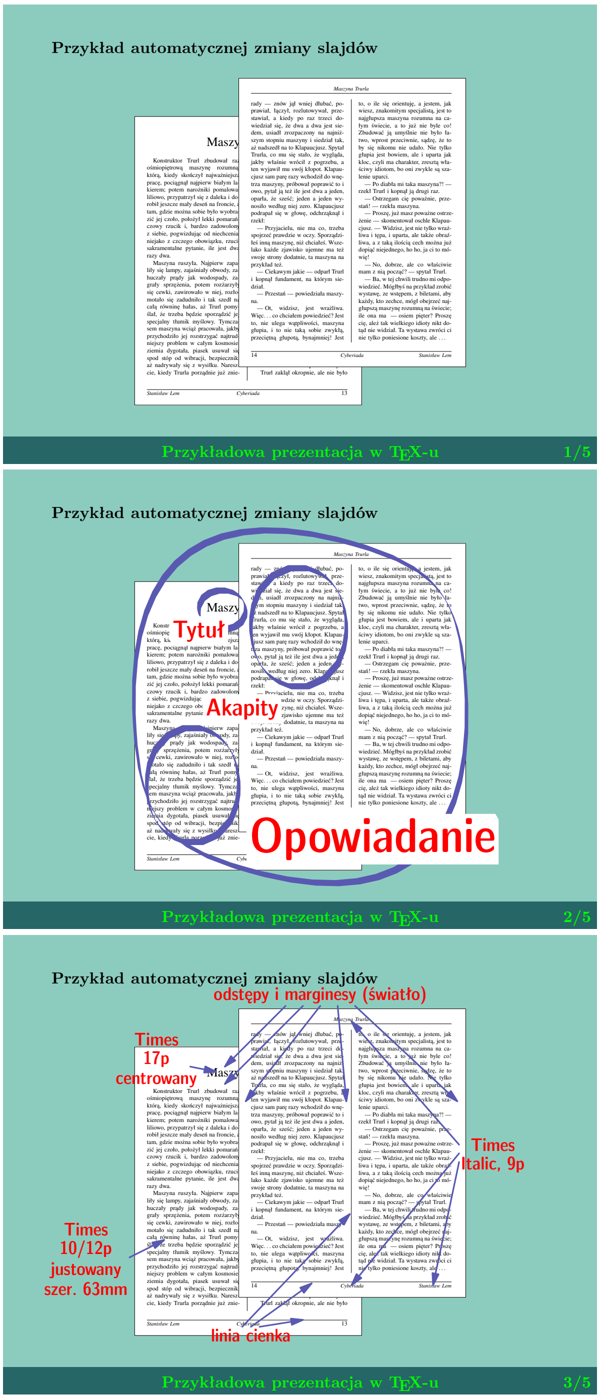
- `\eval`⟨*kod*⟩`\foreach`⟨*arg1*⟩`...`⟨*argn*⟩`\endfor` – wykonanie podanego przez użytkownika kodu dla każdego z argumentów. Kod musi kończyć się wywołaniem zdefiniowanego wcześniej makra co najmniej jednoargumentowego. Zostanie ono wywołane tyle razy ile jest parametrów (n razy). Jako ostatni (lub jedyny) parametr tego makra będą podstawiane kolejne argumenty. Konstrukcja ta pozwala np. realizować „odsłaniane” po kawałku slajdy lub proste animacje.
- `\ifpdftex` – polecenie warunkowe, które zwraca wartość „prawda” gdy używany jest pdfTEX, bez względu na format.
- `\iflatex` – polecenie warunkowe, które zwraca wartość „prawda”, jeśli pakiet jest używany w LATEX-u, bez względu na wersję TEX-a.
- `\ifpdfplain`, `\ifdviplain`, `\ifdvilatex`, `\ifpdflatex` – polecenia warunkowe określające jednocześnie używany format i wersję TEX-a.

Przykłady

Na koniec pokazane będą dwa proste przykłady prezentujące sposób użycia większości z dostępnych dla użytkownika poleceń.

Prezentacja przygotowana w plainie. W przykładzie wczytywane są trzy rysunki przygotowane z wykorzystaniem programu METAPOST. Grafikę METAPOST-ową włącza się w zwykłym TEX-u inaczej niż w pdfTEX-u. Przykład jest przygotowany w taki sposób aby można go było bez konieczności modyfikacji przetwarzać zwykłym TEX-em i pdfTEX-em.

```
 1. \input pdfslides.sty
 2. \ifpdftex
 3. \input supp-mis
 4. \input supp-pdf
 5. \def\fig#1#2{\convertMPtoPDF{#1}{#2}{#2}}
 6. \else
 7. \input epsf
 8. \def\fig#1#2{\def\epsfsize##1##2{#2##1}%
 9.   \epsfbox{#1}} \fi
10. \footlinecolor{\rgbcolor{.15,.4,.4}}
11. \foottextcolor{\rgbcolor{0,1,0}}
12. \bgcolor{\rgbcolor{.55,.8,.75}}
13. \foottext{\bf Przykładowa prezentacja
14.   w~\TeX-u}
15. \pagenum{\thepage/\lastpage}
16. \beginslides
17. \def\zmiany#1#2{
18.   \beginslide
19.   \transition{Dissolve}1
20.   {\bf Przykład automatycznej zmiany
21.     slajdów}\par
22.   \bigskip
23.   \centerline{\fig{a.#2}{#1}}
```



Rys. 1. Kolejne slajdy prezentacji w plainie

```
24.   \endslide}
25. \eval\duration3\zmiany{.38}
26.    \foreach123\endfor
27. \zmiany{.38}1
28. \endslides
29. \bye
```

Do sparametryzowania pliku wykorzystaliśmy polecenie warunkowe `\ifpdftex` (wiersze od 2 do 9).

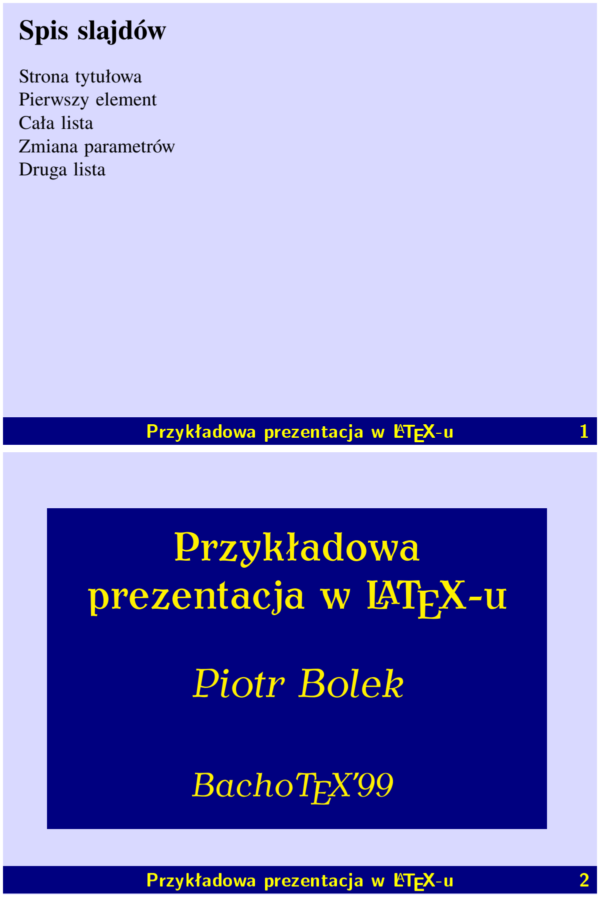
W kolejnych wierszach (10–15) parametry naszej prezentacji: kolory, tekst w pasku tytułowym oraz sposób oznaczania numeru strony. Następnie między poleceniami `\beginslides` i `\endslides` umieszczamy właściwą prezentację. W tym celu definiujemy dwuargumentowe makro `\zmiany`. Parametrem jest numer pliku graficznego (nasze pliki nazywają się `a.1`, `a.2` i `a.3`). Jako sposób przejścia między slajdami wybieramy `Dissolve`. Następnie tworzymy pętlę. W kodzie pętli ustalamy czas wyświetlania slajdów na 3 sekundy (`\duration3`) i polecamy wykonać pętlę dla trzech argumentów (1, 2 i 3). Na koniec jeszcze raz wyświetlamy bezpośrednim wywołaniem makra `\zmiany` (wiersz 27) pierwszy z sekwencji rysunków (wiersz 27). Pierwsze trzy slajdy prezentacji są pokazane na rys. 1. Czwarty slajd jest taki sam jak pierwszy (różni się jedynie numerem strony).

Przy okazji drobna uwaga. Jeśli chcemy w podobny sposób jak w tym przykładzie zsynchronizować rysunki na koljenych slajdach to musimy zadbać o to aby zawierały one pokrywające się obiekty i miały identyczny prostokąt ograniczający (ang. *bounding box*). Spełnienie takich wymagań można przy odrobinie starania zapewnić w dowolnym programie graficznym, mnie osobiście (i chyba nie jestem w tym osamotniony) najbardziej odpowiada wykorzystanie w tym celu możliwości jakie daje program METAPOST.

Prezentacja w LATEX-u. W kolejnym przykładzie widzimy jak można w różny sposób wykorzystywać udostępniane przez pakiet mechanizmy aby przygotować prezentację z „odsłanianymi” stopniowo slajdami.

```
\documentclass[12pt]{article}
\usepackage{polski}
\usepackage{qtimes}
\usepackage{pdfslides}

\begin{document}

\footlinecolor{\rgbcolor{0,0,.5}}
\foottextcolor{\rgbcolor{1,1,0}}
\def\bg{\rgbcolor{.85,.85,1}}
\bgcolor{\bg}
\foottext{\bfseries\sffamily Przykładowa
  prezentacja w \LaTeX-u}
\startpage{2}
\endtext{\fontsize{35}{40}\selectfont
  \bfseries The End}

\begin{slides}
\begin{slide}
\section*{Spis slajdów}
\listofslides
\end{slide}

\begin{slide}
\addtolos{Strona tytułowa}
\transition{Split}5HO
\null\bigskip
\fontfamily{antt}\selectfont
\PDFcolorbox{\rgbcolor{0,0,.5}}{
  \parbox{.85\hsize}{\centering
    \cmykcolor{0,0,1,0}
    \bigskip
    \Huge\bfseries Przykładowa
    prezentacja w \LaTeX-u\\[2ex]
    \itshape Piotr Bolek\\[3ex]
    \LARGE
    Bacho\TeX'99
    \bigskip
    }
  }
\end{slide}

\def\odslona#1{
  \begin{slide}
  \section*{Kolejno
    odsłaniane punkty}

  \Large
  \begin{enumerate}
  \item Pierwszy   \ifnum#1=1\bg\fi
  \item Drugi punkt\ifnum#1=2\bg\fi
  \item Itd. \ldots\ifnum#1=3\bg\fi
  \item I ostatni
  \end{enumerate}

  \transition{Glitter}{2}{315}
  \ifnum#1=4\addtolos{Cała lista}\fi
  \ifnum#1=1\transition{Split}5VI
  \addtolos{Pierwszy element}\fi
  \end{slide}}

\eval\odslona\foreach1234\endfor

\footlinecolor{\rgbcolor{.15,.4,.4}}
\foottextcolor{\rgbcolor{0,1,0}}
\bgcolor{\rgbcolor{.55,.8,.75}}

\def\odslonaII#1{
  \begin{slide}
  \section*{Zmiana parametrów i \ldots}
  \section*{Inna technika odsłaniania}

  \Large
  \begin{enumerate}
  \item Początek   \ifnum#1<2\else
  \item Ciąg dalszy\ifnum#1<3\else
  \item \ldots
  \item I w końcu koniec...  \fi\fi
```

```
79.   \end{enumerate}
80.
81.   \transition{Glitter}10
82.   \ifnum#1=3\addtolos{Druga lista}\fi
83.   \ifnum#1=1\addtolos{Zmiana
84.     parametrów}\fi
85.   \end{slide}}
86.
87. \eval\odslonaII\foreach123\endfor
88.
89. \transition{Dissolve}3
90. \end{slides}
91. \end{document}
```



Rys. 2. Pierwsze dwie strony prezentacji w LaTeX-u

W tym przykładzie widać jak można wykorzystać makra zdefiniowane w pakiecie do zamykania tekstu w kolorowe pudełka (slajd tytułowy – wiersze od 19 do 41). Tak naprawdę, to mechanizm kolorowego pudełkowania został zaimplementowany raczej z myślą o użytkownikach formatów plainowych (bo nie mają pakietu `graphics` jak LaTeX-owcy), ale mechanizm jest ogólny i można go używać także tutaj. Kolejne dwa slajdy stanowią przykład jak można zrealizować „odsłanianie” po kawałku tekstu. Każda „odsłona” jest tak naprawdę kolejną stroną. Używając jednak mechanizmu przejść uzyskujemy efekt przyrostowego wyświetlania pojedynczej strony. Zawsze używamy pokazywanego już w poprzednim przykładzie mechanizmu realizacji wielokrotnego wywoływania z różnymi argumentami zdefiniowanego wcześniej makra. W różny sposób realizowane jest jednak „zasłanianie” tekstu.

W pierwszym rozwiązaniu (wiersze od 50 do 53) stosujemy prosty chwyt polegający na tym, że piszemy tekst kolorem takim samym jak kolor tła (makro `\bg` było używane do zdefiniowania koloru tła w wierszach 10 i 11). Drugi sposób (wiersze 75–78) jest bardziej wyrafinowany, bo stosujemy tam zagnieżdżone polecenia warunkowe i nie chowamy tekstu jak słonia w jarzębinie ale rzeczywiście nie umieszczamy na stronie tego czego nie widać. Są to tylko przykłady – z pewnością każdy będzie w stanie wymyślić własne dopasowane do konkretnych potrzeb sposoby.

W obydwu odsłanianych slajdach używamy też polecenia `\addtolos`, dodając wybrane strony do spisu treści. Przykład ten jest także ilustracją wykorzystania możliwości określenia strony, od której ma być rozpoczęta prezentacja. Ponieważ na pierwszej stronie umieściliśmy spis slajdów, to musimy zacząć od drugiej (rys. 2). Pomiędzy dwoma slajdami zmieniamy także kolory (64–66). Możliwa jest oczywiście zmiana innych parametrów takich jak sposób numerowania stron czy napis w pasku na dole strony.

Zakończenie

Mam nadzieję, że zaprezentowany w tekście pakiet okaże się użyteczny i trafi w zapotrzebowanie szerokiej rzeszy użytkowników TeX-a. Jest tu i trochę ryby, dla tych, którzy właśnie ryby potrzebują i mam nadzieję, że kawałek wędki dla bardziej wymagających. Wszystkim użytkownikom pakietu życzę więc obfitych łowów i smacznego.

Literatura

[1] Adobe Systems Inc., *pdfmark Reference Manual*, Adobe Developer Support. Technical Note 5151, 24 October 1995.

[2] Adobe Systems Inc., *Portable Document Format Reference Manual*, version 1.2, 27 November 1996.

[3] Tomasz Przechlewski, *Publikacje elektroniczne: TeX i PDF*, Biuletyn GUST z. 8, 1997.

[4] Tomasz Przechlewski, *Publikacje elektroniczne: TeX i Acrobat/PDF*, Bachotek 1998.

[5] Hàn Thế Thành, Sebastian Rahtz, Hans Hagen, *The pdfTeX user manual*, 9 March 1999.