

Application for effective online presentation of lectures

Jakuš Juraj · Elektrotechnika, Informačné technológie

24.12.2012



Subject of this document is development and realisation of original application created in Adobe Flash CS4 environment in programming language ActionScript 3 for effective presentation of PowerPoint lectures with accompanying soundtrack, or subsidiary text on the web.

Application that was created is programmed in object oriented manner and therefore allows easier extension or change. This application became part of educational portal eLearn central and web site of Institute of electronics and photonics, where it is going to serve for science and technology education.

1. Introduction

Today, a lecture using PowerPoint presentation is a standard not only in education or science and technology popularization, but also of course in commercial sector. But there is a requirement for repeated availability of such a presentation in large time gap after real presentation in education and popularization. Most solutions however are extremely demanding in the amount of processed data, more precisely on hardware equipment. Normally captured lecture very often takes GBs of data space, which is difficult to transfer and store.

When looking for a way to effectively present Powerpoint lectures on the web, it is necessary to take into account the following requirements for application:

- generally applicable
- simply controlled
- undemanding for author at lecture insertion
- available
- space undemanding.

One of possible solutions is creating Flash based application for web presentation of PowerPoint lectures with accompanying soundtrack, or with subsidiary text on web [1].

2. Author's tools

Adobe Flash [2] (originally Macromedia Flash) is multimedial platform originally in possession of Macromedia and at present developed and distributed by Adobe Systems Company. From its introduction in 1996 it became popular method for adding

animations and interactivity on the web pages. Flash is commonly used for the creation of the animations, advertisement and other website components. It integrates videos to websites and earlier it was used for developing rich internet applications.

Flash can manipulate with vector and raster graphics and support bi-directional streaming of audio and video. It uses scripting language called Action Script. There are many products able to create or display Flash content including mainly Adobe Flash Player, which is freely available for most casual internet browsers, for some mobile phones (flash lite also in newer ones) and other electronic devices. Adobe Flash Professional application's purpose is to create engaging multimedial content like web applications, games, movies and content for mobile phones and other devices.

Movies, „Flash animations“ and „Flash games“ usually use files with SWF extension, traditionally called “Shockwave Flash”. SWF file can be part of a website and it is being played in independent Flash Player, or it can be exported as separate runnable application (with .exe extension in Windows or hqx in Macintosh). Flash video files use .FLV extension which has .SWF files integrated and can be played using codecs (on VLC player, QuickTime or Windows Media player with supplied external codecs).

When need for creating a way of presenting ppt lectures with minimal space requirements arose, there were no programs available for creating such applications in HTML 5 standard [5]. Adobe developed application called Edge, which offers possibility of creating simple animations and vector drawings with effects using only HTML + CSS + Javascript. Next tool available is Tumult Hype, which is runnable only on Macintosh from Apple. It's possibilities are comparable to Edge. Both tools have advantage, that they can hide code needed to make the items run behind buttons, which when pressed apply needed scripts for desired effects. By the concept these tools are similar to flash. They also have keyframes, into which are put items - photographs and others. Effects are applied on them, and the result is exported similarly as movie afterwards.

These tools by its range don't cover possibilities of flash [6] so daring claims about HTML 5 replacing flash are unjustified by now. The main disadvantage of HTML 5 is different implementation of new features in every browser. So programmer has to modify the code to get the same result in all browsers. Reason of this problem is that HTML 5 is still in state of proposal so some elements are added sooner than they are standardized. This problem [4] encountered for example Facebook, which bet on HTML 5 because of mobile devices. This choice proved later to be wrong and to get some quality standard and speed they had to make native applications for Android and iOS separately. Based on gathered facts Adobe Flash was chosen as a platform for creating application for online presentation.

3. Solution analysis

To make application generally usable without repeated exporting of source files, input data must be included in external files, with which is application able to cooperate. Ideally it should be similar to youtube player, with difference in size, because lecture texts have to be easily readable. In order to be as small as possible it won't contain videos. Instead it will use soundtrack with voice of lecturer and images of individual

presentation slides. In order to work it also needs .txt file with times of slides change. This type of presentation takes less disk space and provides all resources to master lecture materials. The input data should be easily insertable for the user to make it available for everyone.

4. Programming solution

Program is created in Flash CS4 environment in programming language ActionScript 3. This language offers class approach and this way it simplifies the whole programming. Also when the classes are implemented well, it is possible to modify them and extend their functionality later easily.

In source configuration file there is variable "easy" followed by equal sign and then times of slide change in tenth of second separated with comma (1 s = 10). Similarly looks configuration file for slides name (required) and file with slides titles (optional). Slides needs to be in „slides“ folder as is default set in program code. Numbering begins with 0 with increase step of 1. Next the application loads .JPG image of lecturer (optional) and „audio.mp3“ file (required), which is soundtrack with voice of lecturer. In play mode application loads individual slides of presentation. If one of required configuration files was missing, program declares an error and won't run.

If one of the slides is missing, application shows symbol of crossed image. If file with slides titles was incomplete, not present or didn't have enough titles, application replenish missing or if needed all slides titles with „Snímka“ and appropriate sequence number (which is also added for provided title names). Slides are loaded in background. That means, that from showing the first slide all others are being loaded sequentially. If user clicks custom slide out of loading sequence (other slide is being loaded) background loading is paused and requested slide is loaded instead.

In addition to .JPG slides application can load videos in .FLV format as well as .SWF animations. For this feature to work, it is needed to use configured name with appropriate sequence number and desired extension instead of classic .JPG. Application identifies file type and process the object appropriately. Application also accepts optional parameters in html page, which can be set at embedding flash to website. Using these parameters, 2 colors can be set, that will be used. Colors are passed in hexadecimal state ranging from 0x000000 to 0xFFFFFFFF. Due to this the number of possible color variations is $0xFFFFFFFF \wedge 2$ (Fig. 1.).

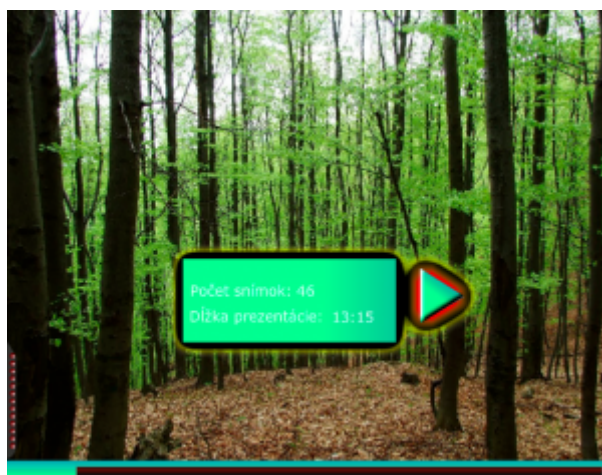


Fig. 1. Preview of SlideShow environment with changed base colors

Programming solution for text version

Program core remained unchanged, but the soundtrack was removed as well as all related control buttons. Slide swapping and side bar remained unchanged. On the bottom old panel was replaced with a new one. This new panel is extendable and when extended it contains educational texts loaded from .xml configuration file. This frame extends automatically and it can be pulled to demanded length (bounds to flash frame on website). There is a button that cancels automatic extending on the panel and frame stays extended even after mouse is away. If presented text is longer than the frame, scrollbar appears. Because this version doesn't have state bar, in case of using video control panel appears on the bottom left. If SWF animation is used it can be paused by clicking anywhere on it and it is possible to step animation using 2 arrows, that appears on the bottom right.

5.1. Graphics / program environment

Program environment (Fig. 2.) is visible whole time and the interface, that user interacts with. User judge the program mostly based on environment. That is why it is not inferior then algorithm for control components on the screen. It should be simple and simultaneously react appropriately on user mouse inputs so user can guess which items are actually control panels without need to study manual.

With these things in mind base environment with simple bar was created, as is usual to group control items. There is extendable side panel on the left, which reacts on mouse move and extends automatically. There is also lecturer photo in the upper part of this panel and remaining space is filled with scrollable list of slide titles. Whole side panel is half transparent. There is standard button for pausing / resuming presentation on the bottom followed with „next“ and „previous“ arrow buttons for jumping to the individual presentation slides. Next control item is navigation bar, which shows presentation progress. If user move mouse over it a „bubbles“ shows up, which indicate the position of slidechange. By clicking on navigation bar it is possible to jump to any loaded part of soundtrack.



Fig. 2. Preview of SlideShow environment with extended side panel with presentation content

Text version environment

Original bottom panel (Fig. 3.) was replaced by sheet of „bookmarks“, divided by functionality into logical units. These parts show and hide as needed. More specifically if slide shows up, which has no associated text, control items for expanding and switching text frame are hidden. If it is expanded and not pinned it will be hidden automatically. Tacking button has tack icon used for notice-boards, which is created by vectorizing bitmap image. Comment button shows as text „comment“ for simplicity and clarity. Remaining control items are bookmarks, which show current slide number and number of total slides, and bookmark that holds control arrows for switching to next and previous slide. These arrows take the form of triangles.

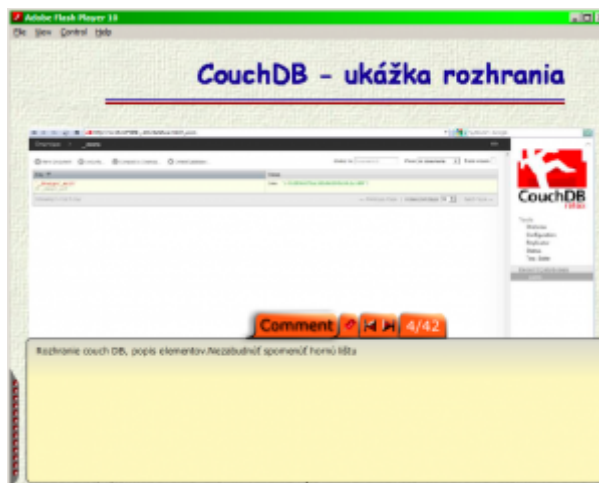


Fig. 3. Preview of Text SlideShow environment with extended bottom panel with texts

5.2. Description of functional elements

Presentation consists of several standalone parts, which trade information. When loading is complete „First info“ frame appears as first, which shows number of total slides (based on slides change time file) and total lecture time (based on soundtrack length). This panel also contains button, which starts the whole presentation.

As mentioned earlier, the bottom panel contains presentation controls. First is pause/play button, which changes based on pause or play mode. When paused it dispatches signal to presentation to stop the soundtrack. At play it dispatches signal to continue. Next to it are buttons for switching to next and previous slide, which work both in play and pause mode. Bar gets information about current track position and with total track length, ratio of these two numbers is track lapse and its position. At click it jumps to new position, dispatches information for track about it and based on this number track is readjusted.

There is side bar on the left side, which is activated at launch and extends if mouse pointer touches its corner. Lecturer photo is loaded from external source and if it is too big it is resized appropriately. Under the photo there is list of slide titles. This list is created dynamically at loading - individual items are added based on number of titles. These buttons have no listener on them as bottom bar buttons have, because it would load CPU too much. Instead there is listener bound on their parent class - list itself and at clicking it finds out which one was clicked. Also currently shown slide title is highlighted to indicate playing position. If user uses bar and clicks on slide, whose

title is outside of currently shown slides in list, the whole list is moved so currently shown slide is in the middle and user doesn't have to search where he is. Side panel also moves titles list on mouse scrolling.

Description of functional elements in text version

Application uses first info frame too, but it contains only information about total slide number, since soundtrack is not present (Fig. 4.). At start presentation shows the first slide and it won't move to next slide without user interaction. The difference when compared with the original version is that the frame number is not extracted from the file of transition times. This file is not present so a variable was added to configuration file with slide file names, which indicates number of slides. Beside these file with educational texts to slides is required and file with slide titles remains optional. After loading the files user can click on any slide. If it is not loaded loading animation appears, that indicates loading.

There is also side panel that has kept all its functionality. Bottom panel was replaced with new one, mentioned in the environment (chapter 5.1). From control items arrows for slide change have remained, but they have other placement. „Comment“ button was added, which is shown only if current slide has educational text or when it is pinned. By dragging this button it is possible for user to set size of text frame, which overlaps shown frame. That is why it can come in handy to dynamically extend frame based on slide content and what it overlaps. Frame is automatically hidden, when user move mouse outside of its area. To turn off this feature, there is tack button next to comment button. In case of SWF slide two gray arrows controlling SWF by step appears on the bottom right. By clicking on the SWF surface it is possible to pause / play animation. In case of flv video control appears on the bottom left and it is possible to pause / play the playback.

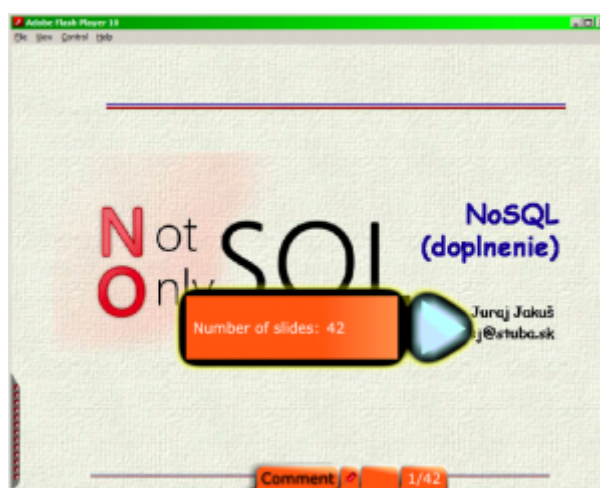


Fig. 4. Preview of SlideShow text version intro frame

5.3. Description of program solution

Flash has its own types of objects, and it is possible to assign the created class code to each of them. For the purpose of this slideshow 9 classes were created, one for every component, which are folded together. The whole slideshow is encapsulated in class SlideShow.as, which is on the bottom of all objects. Inside is connection error note, frame with first informations to start presentation, side panel, bottom control panel,

„FotoKontajner_mc“ objekt, which shows every slide and MovieClip objects, which are played at pause / play as animation in the main area.

Application start

After starting application constructors of objects are called. Configuration file with slide names is loaded as first to show the first slide immediately. After that file with times of slide transitions follows (based on that number of frames is calculated) and soundtrack. Because flash cannot determine length of mp3 file while not fully loaded, this detail is calculated as ratio of number of loaded bytes and total number of bytes combined with length in seconds of loaded part. These informations that flash provides are used for calculation. So when more data are loaded more precise the calculation is. Slide titles are loaded together with soundtrack in the background, but that doesn't prevent application start. If everything is successfully loaded start button appears to start the presentation.

Play mode

After start button is pressed the whole „prveInfo“ object is destroyed to free memory and soundtrack is started. There is listener bound to soundtrack, that tracks the time and when the time comes, which corresponds to the transition to the next slide, the listener switches the slide.

Pause mode

At clicking pause the soundtrack is paused and its tracking by listener is turned off. All control buttons remain active.

Control items possibilities

Main control item is lapse bar, which shows current soundtrack position and allows to jump to any place in whole presentation. Bar is updated with 2 functions „updateBar“ and „updateBar2“. UpdateBar is used in play mode and it is being updated based on soundtrack position. UpdateBar2 is called in pause mode and position information is obtained from slide number, for which it is possible to obtain time of change or by clicking on bar getting soundtrack position with ratio of clicking distance and bar length. Left and right arrows cause immediate jump to next/previous slide.

Side panel is independent component of slideshow with static extend animation. It is activated by moving mouse over it. Also second MovieClip object is activated, that is above whole presentation and causes panel to shrink back if mouse is positioned over it. There is summary slide list with its titles and position numbers on this panel. Titles react as buttons and are linked with the same position changing function as lapse bar. Automatic movement of titles occur based on current position in presentation trough playback if user don't manipulate with side panel.

Error handling

In case of error in configuration file or at file loading there is MovieClip object with invisible property. If error appears, this property is set to true to warn user about

server error and impossibility to play presentation.

Tweener class

On „fade in“ effect of slide showing was used freely available flash class Tweener, which has built in functions for smooth picture appearance. This class is also used for extending animation of side panel. Previously this was made with static animation, but with the help of this class it is much easier to program it.

Browsers compatibility

Application is tested on the most popular browsers. Since flash is a plugin, there shouldn't have been any problems. But that is not true. In Opera slides won't load if diacritic is in the name of path.

Software solution of text version

In addition to the previous version a new class was added that controls extendable text field, which also holds control items. Extending animation is solved through ENTER_FRAME event. Previously mentioned Tweener was hardly used because of interruptions and other problems.

5. Programming solution of Slideshow modifier

Besides the basic application, a flash application was developed (Fig. 5.) for adding interactive elements to existing presentations. To achieve this, the structure of slides class in the original application had to be reprogrammed to be able to keep the arbitrary amount of added objects. To define elements xml configuration file was introduced. As manual adding of elements position would be very difficult, there's an application just for this purpose, which outputs a configuration file that is downloaded by original presentation. If loading xml file fails, presentation goes on undisturbed.

Environment is very simple for comfortable work. It also can load existing configuration file for its next editing or change and save it again. To insert objects there are icons present, each represents one selected object. Since flash is normally forbidden to access file structure of local user, communication channel actionscript - javascript - java was used for saving [3]. Java has methods that can save the file, and the first use dispatches window to indicate the security risk that is necessary to allow.

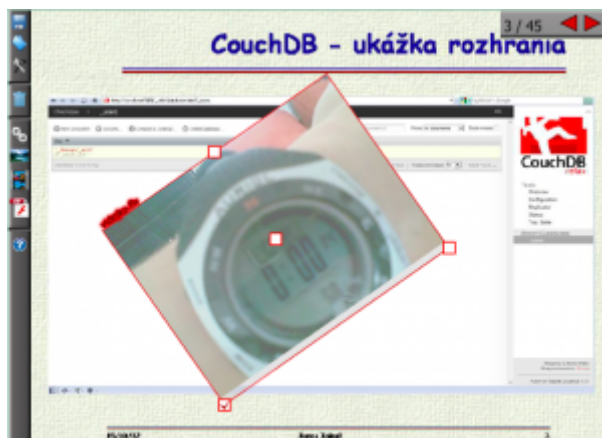


Fig. 5. Preview of SlideShow modifier with added video which is selected to show manipulating options

6.1 Modifier environment

Environment consists of main plane, where current slide of presentation is shown. It is possible to switch between slides in the panel located in upper right corner. It is possible to move 5 slides at once if ctrl is pressed at clicking on panel button. The left side is dedicated to the bar where are located icons for saving and loading, inserting elements into your presentation, button to remove selected element and the “help” button to explain the functionality of the application. The icons were downloaded as a bitmaps from free available resources, or made specifically to testify as much as possible about their function. At object insertion, the bar is graphically overlaid with semitransparent frame that notes that the icons are inactive until the completion of the operation. “Help” is a series of 6 images with photos and text descriptions of the elements.

6.2 Description of functional elements in Modifier

Modifier load files from file structure. Therefore, if file is put on slide, it is loaded and displayed, which provides immediate information about the appearance of the application. It is desirable to copy Modifier in the same folder as Slideshow and run it from there, so paths to files being loaded matched mutually. If the user enters the wrong path to the file, warning icon will appear on the image indicating that the file could not be loaded, which is an error indicator. After start, the user moves to the desired image, where he wants to put content. After that it is needed to click the appropriate icon based on type of object user wants to insert. An information rectangle is shown near mouse pointer that indicates type of object being inserted. By clicking and dragging mouse over the area a square for object is being drawn (it is possible to change it later) which represents object boundaries. After creating the square a box runs up in which user has to enter the path to file (or URL address of webpage at link insertion).

Once entered the object is added. Then you can add another object, delete or edit an existing object. To delete or edit object, it is needed to select it first. That is possible by holding ctrl and clicking on the object. By clicking on the trash icon the selected object is deleted. Editing object is possible using white rectangles at its borders. With its help it is possible to rotate, move or change size of the object. A rectangle on the bottom left is checkbox indicating whether aspect ratio should be preserved at changing size. By clicking on the textfield in the upper side of object boundary it is possible to change path to file instantly, or change address in case of URL address. To deselect object user has to click anywhere outside objects boundaries, at pressed ctrl key. After work is finished it is possible to save created configuration file by clicking on the diskette icon.

6.3 Modifier programming solution

For elements insertion a class “RectCreat” was used, which cares for rectangle creation at inserting the beginning. After boundary is created it is replaced with

“TransformMatrix”, which handles editing operations of existing elements. Since editing represents the same operations this class is used for all elements. Javascript is used for saving, which can communicate with actionscript without problems. But for this connection to work it is necessary to set permission. After warning window is dispatched about script access (it is dispatched at attempt to save) it is needed to click on “settings” and in window that appears click to “advanced” tab and then click „Trusted Location Settings...” button, where it is necessary to add path to folder with modifier SWF file.

After that follows security warning from java. If checkbox to not show this warning again is clicked and “ok” follows modifier works properly and security warnings will not be dispatched again. JApplet class from “jnlp” package for access to functions is used in java, which are denied to applet by default. Since this application is not created for online use, it is quite large compared to presentation parts (4-5 times bigger).

6. Possible further developments

Program core was from the beginning programmed by chaining all classes together. Particularly loading external configuration files, presentation slides and controls are programmed this way. But flash offers a better approach. Similarly as java or messages of mfc framework in visual C++ flash has ability to dispatch custom user events, which can be dispatched by every component. These events can even hold custom data, information, pointers and other needed structures. Chaining with events is also more clear from the view of object oriented design and enables even easier component changing, since these are not directly linked – they only have listeners and handlers of particular events, which can but don’t have to be caught. Also this model allows better error handling, because classes are better encapsulated and separated. The mentioned approach facilitates extendability, what is a needed property for the future.

Challenge to change, although not so significant from the functionality point of view, is adaptation of control items in text version so that at it would be also changed at setting input colors with parameters. The problem is the specific shape and mask problems with this element.

In programming solution of modifier it is worth to considerate using FLEX. It has built in class for saving files locally, so security warnings from flash and java would be eliminated. This solution was not used in first place, because of lack of knowledge about FLEX and its usage. External language configuration is also worthy of consideration. Up to this day only English variation was needed and in this particular case the source texts were manually changed to English. Loading external files slows down start time at the beginning so adding additional configuration files is time consuming. Possible workaround is to include language variations in code itself and then simply pass parameter to choose language. This feature may be added in the future.

7. Conclusion

Flash application for web presentation of ppt lectures with soundtrack by experts from

the field of science and research was developed. Moreover text version of the same application was created, which is able to show text content for every slide. To support variability application for adding interactive content was created as well. This player will become a part of educational portal eLearn central and webpage of the Institute of electronics and photonics at FEI STU Bratislava, where it will serve for the purpose of education and popularization of science and technology. It has been already used for the presentation of selected problems of microelectronics.

8. Acknowledgments

The present work has been developed at the Institute of Electronics and Photonics, FEI STU Bratislava with the help of projects VEGA 1/0507/09 and VEGA 1/0712/12. The authors would like to thank to doc. Ing. Jiří Hrbáček, PhD for guidance in ActionScript.

9. Literature references

1. Hrbáček, J. Flash 1 – tvorba inteligentní grafiky – multimediální učebnice. Brno: MSD, spol. s.r.o Brno, 2007. 89 s. ISBN 978-80-7392-000-5
2. ActionScript reference pages, [online] Publikované 22.04.2012, [citované 10.2.2007], Dostupné z <http://www.scienceworld.cz/sw.nsf/ID/4AE29E09E5310910C1256E970049227B?OpenDocument&cast=1>
3. Mullins Evan, Actionscript (as3) Javascript Communication | Call Flash to and from javascript. [online] Publikované 15.12.2010, [citované 20.4.2012], Dostupné z <http://circlecube.com/2010/12/actionscript-as3-javascript-call-flash-to-and-from-javascript/>
4. Matt Rosoff. Don't abandon HTML5 just because Mark Zuckerberg hates it [online] Publikované 11.9.2012, [citované 26.9.2012], Dostupné z http://www.citeworld.com/development/20652/mark-zuckerberg-hates-html5-but-you-dont-have-to?source=ifwincpt_cite_zuck
5. Stephen Shankland. HTML vs. Flash: Can a turf war be avoided? [online] Publikované 3.2.2010, [citované 28.9.2012], Dostupné z http://news.cnet.com/8301-30685_3-20000037-264.html
6. Peter Wayner. HTML5 vs. Flash: The case for Flash. [online] Publikované 2.6.2010 [citované 28.9.2012], Dostupné z <http://www.infoworld.com/d/application-development/html5-vs-flash-the-case-flash-721>

Coauthor of this paper is Ľubica Stuchlíková, Institute of Electronics and Photonics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava

