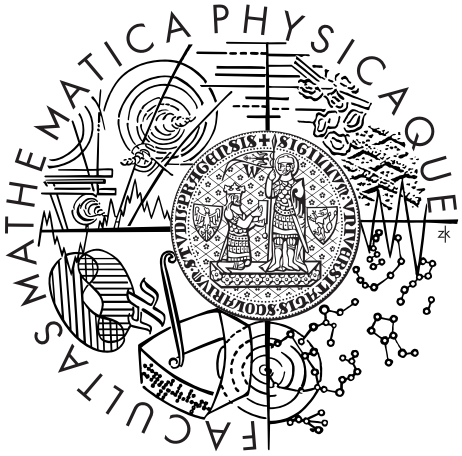
Charles University in Prague
Faculty of Mathematics and Physics

BACHELOR THESIS



Šimon Rozsíval

Vector Screencast

Department of Distributed and Dependable Systems

Supervisor of the bachelor thesis: Mgr. Martin Děcký
Study programme: Computer science
Specialization: Programming and software systems

Prague 2015

Dedication.

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In ..... date ..... signature of the author

Název práce: Vektorový screencast

Autor: Šimon Rozsíval

Katedra: Katedra distribuovaných a spolehlivých systémů

Vedoucí bakalářské práce: Mgr. Martin Děcký

Abstrakt: Cílem bakalářské práce je vytvořit software pro záznam a přehrávání výukových videí pro potřeby Khanovy školy. Na rozdíl od běžných videí nejsou obrazová data uložena ve formě bitmap, ale jako vektory, což umožní snížit datovou náročnost a vykreslit obraz ostře při libovolně velkém rozlišení obrazovky uživatele. Přehrávač videa i nástroj pro nahrávání běží ve webovém prohlížeči. Součástí práce je také návrh a implementace vhodného formátu pro uchovávání obrazových a zvukových dat a implementace v softwarové architektuře klient/server.

Klíčová slova: screencast, vektory, video, on-line

Title: Vector Screencast

Author: Šimon Rozsíval

Department: Department of Distributed and Dependable Systems

Supervisor: Mgr. Martin Děcký

Abstract: The goal of this bachelor thesis is to create a software for recording and playback of educational videos for Khanova škola (Czech clone of Khan Academy). Contrary to common videos the visual data is not stored as a sequence of bitmaps, but as vectors. This allows to reduce the data bandwidth and playback sharp images in any target resolution. The player and also the tool for recording the videos runs in a web browser. The thesis also focuses on designing and implementing a suitable file format for storing the visual and audio data and implementing the software according to the client/server paradigm.

Keywords: screencast, vector, video, on-line

Contents

Introduction **2**

1 Distance education **3**
1.1 Current systems . . . 3
1.1.1 Coursera . . . 3
1.1.2 Youtube.com . . . 4
1.1.3 Moodle . . . 4
1.1.4 Educreations and ShowMe . . . 4
1.1.5 Khan Academy . . . 5

2 The Vector Video project **6**
2.1 Video recording tool requirements . . . 6
2.2 Video player requirements . . . 6
2.3 Goal of this thesis . . . 7

3 Analysis **8**
3.1 Technical requirements . . . 8
3.2 Available techonologies . . . 8
3.2.1 Java applets . . . 8
3.2.2 Adobe Flash . . . 9
3.2.3 Microsoft Silverlight . . . 9
3.2.4 HTML5 . . . 9
3.3 Conclusion . . . 10
3.4 Possible issues and known limitations . . . 10

4 Vector Screencast file format **11**
4.1 Audio . . . 11
4.2 Video . . . 11
4.2.1 Binary format . . . 11
4.2.2 Text format . . . 11
4.3 Conclusion . . . 11

5 Implementation **12**
5.1 ECMAScript . . . 12
5.1.1 TypeScript . . . 12
5.2 Event driven programming . . . 13
5.3 HTML5 . . . 13
5.3.1 Rendering graphics using HTML5 . . . 14
5.3.2 Audio capturing, processing and upload . . . 15
5.3.3 AJAX . . . 15
5.3.4 High Resolution Timer . . . 15
5.3.5 Input from pointing devices . . . 16
5.4 Drawing curves . . . 17
5.4.1 DynaDraw algorithm . . . 17

6 Integration into other applications **20**
6.1 Video player 20
6.2 Video recording 20

7 Testing **21**
7.1 Comparint the tool to youtube 21

8 Users' documentation **22**

9 Programmers' documentation **23**

Conclusion **24**

Bibliography **25**

List of Tables **27**

List of Abbreviations **28**

Attachments **29**

Introduction

Each and every person on earth explores the world from the day of and continues to learn new things all his life. Education in the so-called developed world is essential for later employment.

@todo

Khan Academy

Khan Academy is an olnine tool providing free access to instructional videos and practice exercises covering various subjects including math, history, programming, economics, and more.

@todo more info

Vector Screencast project

A year ago two members of K̈hanova škola, the czech brach of the Khan Academy, with an idea of improving the current technical solution of recording and displayling videos on their website. Their idea was not to capture classical bitmap-based videos, but rather a vector-based animation that could be scaled to any display resolution or edited in the future to improve quality of their videos, as some of videos recorded only a few years ago don't look nice or are't even well readable on large displays. The other extreme are small displays of tablets and smartphones where the downscaled letters are too small to read and can't be zoomed well in most video players.

One of the other reasons for this type of solution was the possible decrease of the size of data transfered over the Internet, as most of the image does not change between each two frames and very often the image doesn't change at all.

Thesis structure

@todo

1. Distance education

Distance education is not only a phenomenon of the few last decades, but can be traced at least back to the 18th century, when Caleb Phillipps posted an advertisment called "Teacher of the New Method of Short Hand" in Boston Gazette, saying "Persons in the Country desirous to Learn this Art, may by having the several Lessons sent weekly to them, be as perfectly instructed as those that live in Boston." [1]

With the development of the Internet and its general accessibility in the developed world, providing distance education has become much easier and has spread widely. In some countries tuition rates are high and young people take loans. This topic is covered in a fitting way by John Oliver in his show [2]. The flexibility and low cost of distatnce education over the Internet gives people, who wouldn't be otherwise able to attend a traditional university, an opportunity to gain knowledge and train skills from their homes [3] spending much less money or even for free.

Students in the deveopling world are also taking the advantage of educational content available on the Internet. Several of the top U.S. universities, like Harvard, Stanford or MIT, put some of their materials on so called MOOCs (Massive Open Online Course) like Coursea, edX or Udacity. This content is then available to anyone with a computer and Internet connection and knowledge of the language. An great example is *Kepler* - non profit university project in Rwanda [5]. The goal of this project is to "provide an American-accredited degree, a world class education, and a clear path to good jobs for thousands of students for around $1,000 tuition per year." [6]

A concept of teaching often refered to as *Flipped Classroom* uses distance education. The idea is to let students study lecture materials at home at their own pace and then apply the things at school the next day by doing activities to illustrate the concepts. The teacher can then help them or explain details here in person. These materials are often in the form of videos either from an open source, like the Khan Academy, or created by the teacher himself.

1.1 Current systems

In the next few paragraphs I will try to pick some of current distance education services and tools available on the Internet. This list is not complete and is only meant to give the reader a notion of available technologies and their paradigms, on which the project of Vector Screencast is based.

1.1.1 Coursera

The mission of *Coursera* is to "provide universal access to the world's best education." [9] Anyone can, for free, go through materials published by universities and other organizations aimed at education.

Courses at Coursera consist mainly of video lectures commonly with a transcript and a presentation document attached to. These videos can be viewed directly in web browser on demand or downloaded to user's computer. After

studying the materials, students can submit assignments' solutions and take quizes and receive a Verified Certificate for the accomplishment of the course. These certificates are not free.

Most of the courses are in English and only a few of the courses are also translated into other languages. It is not possible for everyone to publish his materials through Coursera.

1.1.2 Youtube.com

Youtube.com [10] is not an educational service by design. Youtube allows people to create their own channels and upload their video content to share it with other Internet users for free.

Youtube was launched in 2005 and has became one of the most frequently visited websites on the Internet according to Alexa Internet [11]. Uploading video to Youtube is free, advertisment is displayed to the user while watching videos though.

The ease of making original-created videos available and the wide audience makes Youtube a perfect place for all individuals and organizations, who want to share their ideas or any video materials. Many educational channels can be found here, for example *Numberphile* [12], *Veritasium* [13], and *Khan Academy* [14].

Youtube videos can be viewed only online in a web browser or a specialized mobile app. There are only unofficial tools for downloading these videos.

The form and content of the video is practicaly unlimited, as long as does not violate the terms of the service. The maximum file size of a video is 128 GB in size and 11 hours in length. To upload larger or longer videos, user must split them into several parts. Video can have a text description which might contain the transcription of the video content and any subtitles can be attached to a video. Youtube also downscales videos to multiple resolutions so they can be viewed with a low speed Internet connection.

1.1.3 Moodle

Moodle [7] is an open-source project used by millions of users [8] providing a robust tool for creating custom learning materials and providing them to students. The source code of Moodle can be downloaded and deployed it any server. Teachers then publish their materials and resources for students or for assigning homework.

1.1.4 Educreations and ShowMe

Educreations is a service for creating and sharing educational videos similar to Khan Academy videos. The idea is that teachers create a their own videos by drawing onto a digital whiteboard and then share these videos with their classes.

Educreations is designed for an iPad device, but can be used also as a web application from any web browser, that supports Adobe Flash Player. Web application supports playing and recording and video player can be easily embeded into any website. Content of the video is scaled appropriately to the output screen and is drawn smoothly.

Figure 1.1: Khan Academy lesson

There is a free variant, which is limited. For more storage capacity and advanced tools, users must pay monthly fees. The software is proprietary and is is the file format of the video.

ShowMe [1] is a very similar service. An iPad app is available for recording and viewing videos. Recorded videos can then be played in a web browser, but not recorded. Pricing is very similar to Educreations.

Both ShowMe and Educreations of these services target mainly on tablets. The advantage of tablets is their touchscreen, which can be used for drawing in a natural way with fingers or a special stylus.

1.1.5 Khan Academy

A similar service to Coursera is *Khan Academy*. Khan Academy originated in 2003 when it's founder, Salman Khan, began tutoring his cousin over an instant messenger via drawing pictures with a computer mouse. Salman then started to record these videos and put them on his Youtube channel, so someone could watch them later. This channel became the base of Khan Academy.

Khan Academy became known and has grown a lot, but the style of Khan Academy videos remained the same. A person draws lines and diagrams using a bitmap editor on his computer and talks about the subject aloud while recording his computer screen and recording his voice using a microphone. These videos are then uploaded to Youtube and embedded on the Khan Academy website [14]. Apart form the video lectures, the website also contains exercises and quizes to encourage students in learning. The pace of the lesson depends on the student. He can pause the videos or watch them multiple times before continuing with the lesson 1.1.

Most of the videos are recorded in English, but many of the videos are translated into other languages - by replacing the audio track with a different one or with subtitles.

One of the projects working on the localization of Khan Academy videos is a czech branch called Khanova Škola[15].

[1] http://www.showme.com/

2. The Vector Video project

In the spring of 2014 the people behind "Khanova Škola" — the Czech branch of the Khan Academy — were looking for a person to develop an experimental video technology based on vector graphics. Their idea was to record raw data of user's input as he creates a Khan Academy style video and later draw the video scaled to match user's device's resolution and achieving maximum quality of the output. Thanks to the sparse nature of vectors in comparison to bitmaps, the data consumption might also be reduced or at least similar. This animation would also be linked to an audio track of authors voice commentary forming a complex Khan Academy style video suitable for educational purposes.

2.1 Video recording tool requirements

User, who wants to create a new video, should enter website with his web browser and without installing any additional software start recording a video using his mouse, touchscreen or digital drawing pen and a microphone. This video should then be uploaded to the server.

If the user uses a digital pressure-sensitive stylus, then the applied pressure during cursor movement should be recorded too. The more pressure the user applies, the thicker the line will be in that point. Using this specific hardware might require additional drivers or specific software installed.

User should be able to choose from different brush sizes, brush colors and select background color. User should be able to erase certain parts of the canvas or the whole canvas at once. User should be able to pause and continue recording at any time.

User should have direct visual feedback of the drawn data on the canvas so he sees exactly the same output as the viewer will when he is playing the video later. All the raw data collected from the user should be stored, including the data that has no effect for video playback, like recording cursor movement while recording is paused. This will allow the user to simulate the process of video recording in the future and use it for further post-processing. Recording of this redundant information should be optional.

2.2 Video player requirements

Any user should be able to play vector video on-line in all major modern web browsers without any special software or plugin installed, including mobile browsers.

Video should be scaled appropriately to the size of the player and device's screen resolution. The lines should have the same shape as the author has intended.

User can pause and continue with the playback of the video at any time. User can skip to any point of the video either forward or backward.

If the author of the video had recorded his voice, it must be played along the video. Audio must be synchronized with the video whenever user plays or pauses the video or when he skips to a different point on the time line.

User interface should be intuitive and easy to use either on a desktop computer using mouse and keyboard, but also with touchscreens.

2.3 Goal of this thesis

The result of this thesis should be an open-source library suitable for extending any web application with the abilities of recording and playing Khan Academy style videos in modern web browsers. An appropriate vector-based format should be chosen or defined to store video data.

The library should be easily adjustable and configurable for different purposes. User interface should be fully translatable to any language.

3. Analysis

3.1 Technical requirements

The overall project consists of two separate tools — the recorder and the player. Each of these tools behaves differently and will be used by different people. While the video player will be used by general audience, the recorder will be used by a much narrower group of content creators.

The recording tool will capture the movement of a virtual chalk and lines drawn on a virtual blackboard, as well as voice of the author using a microphone. The recorder data will be then sent to the server, where it should be stored. The video player will receive previousely recorded data and display the movement of the chalk and lines created by the author while playing the voice comment. It is required that the video player will placed in a web page and it must run well in a all modern web browsers without requiring installation of any non-standard plugins or codecs and thus making it easy for a potential viewer to watch contents of the video.

For the purposes of recording, the tool must be able to capture the input from a microphone, track mouse movement and left mouse button state, collect information from Wacom graphics tablet devices and draw lines on screen. What the creator sees should be the same as what the end user will watch later.

3.2 Available techonologies

Web is a huge and fast growing environment. In only a few years, it has become a universal place for exchanging and presenting information. This put the web in the focus of many software companies and organizations and as a result, many different technologies for developing rich interactive applications (RIA) have been created. Some of them have already faded into obscurity, other are just emerging. One of the main limitations in the selection of the right technology for developing web application is their compatibility with operating systems and web browsers.

3.2.1 Java applets

Java applets are used for creating interactive applications withing web browser. Java applets meet all the specified technical requirements of both video player and recording tool.

Java applets are written in any language, that can be compiled into bytecode, this bytecode is then downloaded to the web browser and then run using Java Virtual Machine (JVM). This means that to be able to run a Java applet, user needs to have JVM installed on the device and an installed and allowed Java plugin in user's browser. This isn't a problem for desktop systems as Java is open source, but there is no support for mobile operating systems such as iOS and Android [].

3.2.2 Adobe Flash

Adobe Flash is a multimedia and software platform used for creating vector graphics, animations and games. Flash has all required features: vector graphics manipulation, working with XML, mouse input capturing, microphone input, and audio streaming [].

To view Flash animations or to execute Flash applications, Adobe Flash Player is needed. Adobe Flash Player is available and being developed for all major operating systems, although that is not true for mobile platforms. There was never any support for Apple iOS [] and in 2012 deveolpment of Flash for Android was discontinued []. Using Adobe Flash would mean to exclude most users of tablets and smartphones [], which is a large disadvantage of this technology.

3.2.3 Microsoft Silverlight

Microsoft Silverlight [] a development tool for creating web applications. It is based on the .NET Framework and it is similar to Java applets and Adobe Flash. It was Microsoft's attempt to compete Adobe Flash, but wasn't well adopted.

Silverlight comes as a plugin for web browsers. It is free, altough the list of supported browsers is even smaller [] than the one of Adobe Flash consisting of exclusively desktop operating systems. Development of Silverlight was also discontinued by Microsoft in 2012 and the combination of these facts makes it unsuitable for this project.

3.2.4 HTML5

HTML5 is the fifth revision of Hypertext Markup Language (HTML) standard of World Wide Web Consortium (W3C). It has been given the Recommendation status in the end of 2014 and all crutial aspects of both tools can be implemented using the proposed standard. Tracking mouse was long supported even in older specifications of HTML and ECMA Script (familiarly known rather as Java Script). Wacom provides a plugin and "Wacom WebPlugin Feel™ Multi-Touch API" [http://www.wacomeng.com/web/WebPluginTouchAPI.htm] for web browsers providing access to precise data from graphical tablets from this manufacturer. Vector graphics are supported through the Scalable Vector Graphics (SVG) format [] and it can be manipulated through Document Object Model (DOM) API [], as well as any other XML content. MediaStream API [] enables access to audio input from user's microphone. The "¡audio¿" tag can be used to stream audio files and play them in the web browser.

While HTML5 is a new technology, many of the most important features have already been implemented in some web browsers, like Google Chrome and Mozilla Firefox. More specifically all the features needed to implement video player are supported in the latest versions of all major web browsers. The only catch might be a disagreement on supported audio file formats among the developers of web browsers. This can be overcome by converting the audio into the most widely used formats and providing them to the browser, which will then choose the one it supports.

The features needed by the recording tool, like the MediaStream API, are more specific and the number of browsers supporting these features is smaller,

but this won't be an issue for content creators, who can easily install a supported web browser on all desktop platforms and even some mobile ones. Also a fast development in this area is expected and these features will be most likely implemented in all major browsers soon.

3.3 Conclusion

The bottleneck of most of the technologies is their support in mobile devices. These devices don't allow some of the above mentioned technologies to run inside them. As the popularity and market share of mobile devices grows, supporting them is a high priority. This eventually leads to only one option, and that is HTML5. All major web browsers have the features needed to create the video player, including the versions of browsers for mobile devices.

3.4 Possible issues and known limitations

Web browsers are developed by several companies and

Mobile OS browsers limitations - audio recording.

Audio recording - large ammount of data - uncompressed - which approach of upload to choose? The most simple - create wav in browser and upload it via multipart form. The more complicated approach - continuous stream using WebSockets or WebRTC - need of specific web server process implementation.

4. Vector Screencast file format

What are the possibilities to encode the video.

4.1 Audio

Audio - standard Mp3 or Ogg Vorbis - I am not possibly able to come with something better.

4.2 Video

4.2.1 Binary format

Video - wrap into Matroska or WebM? What are the pros and cons?

4.2.2 Text format

This is only about the video.

SVG

- opensource standard for vector graphics. Needs to be prerendered - I can't just specify sequence of points with pressures - I need to specify exactely the quadrilaterals - rendering method might change in the future. No support for drawing-lines-like animation.

Custom XML-based format

Raw data in some sturctured file format, based on SVG or somehow wrapping SVG.

4.3 Conclusion

Which one I have chosen and why. Internal storage (MKV/WebM), immediate (XML)?

5. Implementation

@todo How the source was designed and maintained. Mention that the development of the code can be found on Github?

5.1 ECMAScript

ECMAScript [1] is a standardized scripting language widely used in website development. Latest approved edition of ECMAScript is ECMAScript 5.1, which is implemented in most major web browsers, and is commonly called JavaScript [2] [3].

JavaScript is a dynamic programming language [4], which combines multiple aspects of imperative, functional, and object-oriented programming.

Functions are so called first-class citizens. This means they can be stored in variables, passed as function parameters, returned as results of functions, and included in data structures []5.

Object oriented programming (OOP) [6] in JavaScript differs from class-oriented OOP in the way inheritance is implemented. While in class-oriented languages, for example in C++ or C#, inheritance is achieved by declaring classes of objects. In JavaScript, object prototypes are used to implement this functionality. @todo

JavaScript doesn't provide any mean of static type checking. All types are created during runtime and they are also checked only during runtime. This lack of static checking during compilation might lead to rarely occurring errors and it is important to take this in mind while writing JavaScript code. A good practice is to write documentation comments[7], where the types are stated.

JavaScript is an interpreted language, some implementations also use a Just-in-time compilation (JIT) [8] for better performance.

Another very important aspect of ECMAScript which affects it's performace is the absence of direct control over memory usage – a garbage collector frees unused .

5.1.1 TypeScript

TypeScript is a typed superset of JavaScript that compiles to plain JavaScript[] [9]. It is an open-source project developed by Microsoft. It is, as it's name suggests, is a strongly typed programming language compatible with JavaScript.

TypeScript extends capabilities of JavaScript by static type checking in the time of compilation, which helps finding errors in source code, and speeds up the

[1] http://www.ecmascript.org/
[2] https://developer.mozilla.org/en/docs/Web/JavaScript
[3] https://msdn.microsoft.com/cs-cz/library/d1et7k7c(v=vs.94).aspx
[4] http://en.wikipedia.org/wiki/Dynamic_programming_language
[5] http://mitpress.mit.edu/sicp/full-text/book/book-Z-H-12.html#call_footnote_Temp_121
[6] http://en.wikipedia.org/wiki/Object-oriented_programming
[7] Commonly used syntax in JavaScript projects is JSDoc (http://usejsdoc.org/)
[8] For example Google V8 engine used in Google Chrome. See https://code.google.com/p/v8/
[9] http://www.typescriptlang.org/

process of coding. TypeScript also introduces class-based object oriented programming to JavaScript. It includes concepts of interfaces and polymorphism, which makes programs written in TypeScript more understandable to programmers familiar with other object-oriented programming languages like Java, C# or C++.

TypeScript is transcompiled into regular JavaScript, and therefore doesn't bring any new functionality. The reason for choosing TypeScript is the clarity of code and more convenient development process for the programmer.

5.2 Event driven programming

The idea behind event driven programming is to break direct references between objects and to communicate with *events* instead of calling object methods directly. The advantage of the this approach is loose coupling [10] – features might be added or removed without breaking the core of the application.

Each object handles only it's own task without knowing anything about the other objects it's collaborating with. When an object completes it's task, it publishes the result with a specific event. Objects subscribed for this event are notified and are given the outcomes of the previous task.

This mechanism is sometimes called the *Event Aggregator* design pattern []. It is implemented through the *VideoEvents*[11] static class, which makes it a simple singleton object. This class provides an interface for registering and triggering callbacks for specified events. Callbacks executed by a triggered event are called asynchronously. It is worth mentioning that web browsers execute all scripts in a single thread.

```
VideoEvents.on(VideoEventType.Message, function(message) {
console.log("received message:", message);
});

VideoEvents.on(VideoEventType.Message, function(message) {
console.log("received message backwards:", message.split("").reverse().join("")
});

VideoEvents.trigger(VideoEventType.Message, "Hello world.");
```

Events triggered and expected by each object are described in the programmer's documentation and can be used to extend the behavior of the player or the recording tool without modifying the original source code.

5.3 HTML5

What is HTML5, what parts are needed. Compatibility of these technologies in browsers. A chart of people using a compatible browser? Playing should be possible in the vast majority of browsers today and the prognosis is good.

[10] http://en.wikipedia.org/wiki/Coupling_(computer_programming)

[11] Implementation can be found in */public/js/app/Helpers/VideoEvents.ts* file

Web workers

Multi-threading in JavaScript. Async methods and web workers.

Working with XML data

The XMLDocument object.

5.3.1 Rendering graphics using HTML5

Displaying text and static visual content is the main purpose of HTML and Cascade Style Sheets (CSS) and it is widely used this way across the web. Creating a complex polygon or curve would be very hard and would involve various tricks [12] or would be even impossible.

The new HTML specification takes this in mind and brings ways of creating more rich and dynamic content within a web page. There are two technologies that should be taken into account - Canvas 2D Context and SVG.

Canvas 2D Context The Canvas element provides scripts with a resolution-dependent bitmap canvas, which can be used for rendering graphs, game graphics, or other visual images on the fly [] [13].

Using canvas seems appropriate for this project. Canvas could be created with respect to user's resolution and web browser window size. All elements can be scaled to fit this viewport. This will make them look sharp and there won't be any artifacts, noise and blur caused by interpolation which would be caused by scaling normal bitmap video.

The problem with Canvas might occur when user resizes his window or enters full-screen after the canvas is initialized. Canvas contains a bitmap image consisting of graphical primitives drawn onto it. All content must be redrawn so it remains sharp.

Scalable Vector Graphics (SVG) Scalable Vector Graphics (SVG)[14] is an XML (Extensible Markup Language) based file format designed for describing two-dimensional vector images. It is an open format developed and maintained by the W3C SVG Working Group [] . Current W3C recommendation is SVG 1.1 (Second Edition).

A valid SVG document must have an *svg* root element with specific namespace attributes and specified *width* and *height* attributes. An example of an empty, but valid, SVG document might look as shown:

```
<?xml version="1.0"?>
<svg version="1.1"
        width="470"
        height="100"
        xmlns="http://www.w3.org/2000/svg"
        xmlns:xlink="http://www.w3.org/1999/xlink"
```

[12] http://nicolasgallagher.com/pure-css-gui-icons/
[13] http://www.w3.org/TR/2010/WD-html5-20100624/the-canvas-element.html
[14] http://www.w3.org/TR/SVG/

```
        xmlns:ev="http://www.w3.org/2001/xml-events">

</svg>
```

The specification of SVG introduces several graphical primitives, that can be used to compose complex shapes. Each primitive is represented by an XML element and a set of attributes. The list of primitives is long, though we will need only two of them – the *circle* and *path*.

A circle needs three basic properties – *cx* and *cy* (coordinates of it's center) and it's radius *r*. Other attributes, like *fill* or *stroke*, can be used to specify the appearance of the primitive. For example a black dot with it's center at point

$$50, 50$$

and radius of 50 would be represented by this XML element:

```
<circle cx="50" cy="50" r="50" fill="black" />
```

5.3.2 Audio capturing, processing and upload

HTML5 provides only one way to access microphone data at the moment and it is through the *getUserMedia API* [15] []. *navigator.getUserMedia* function prompts the user to for permission to use their audio input (this function is also used to access webcam stream in other applications). The *navigator.getUserMedia* function is well documented on Mozilla Developer Network (MDN) [16] website.

If user's device has a connected microphone and user gives his permission to use his audio input, then a *MediaStream* [17] object is provided by the browser and from this time on audio can be captured. Error callback with *MediaStreamError*[18] instance is called otherwise.

Once the *MediaStream* is obtained,

5.3.3 AJAX

5.3.4 High Resolution Timer

To make the video look as good as possible, we need to store as precise data as possible. The *Date.now()* function[19] returns the number of milliseconds elapsed since 1 January 1970 00:00:00 UTC. The millisecond accuracy might seem enough, but modern browsers provide even more accurate data via the *High Resolution Time* [20] via the *window.performance.now()* function[21] [22] with the accuracy of

[15] getUserMedia: http://www.w3.org/TR/mediacapture-streams/#dom-mediadevices-getusermedia

[16] MDN: https://developer.mozilla.org/en-US/docs/Web/API/Navigator/getUserMedia

[17] MediaStream: http://www.w3.org/TR/mediacapture-streams/#idl-def-MediaStream

[18] MediaStreamError: http://www.w3.org/TR/mediacapture-streams/#idl-def-MediaStreamError

[19] https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date/now

[20] https://dvcs.w3.org/hg/webperf/raw-file/tip/specs/HighResolutionTime/Overview.html

[21] https://developer.mozilla.org/en-US/docs/Web/API/Performance/now

[22] http://updates.html5rocks.com/2012/08/When-milliseconds-are-not-enough-performance-now

microseconds. The *window.performance.now()* function doesn't provide data related to current time, but the milliseconds elapsed since page was loaded as a floating point number. This makes it more suitable for animation purposes.

Timing functionality is wrapped in *VideoTimer* class[23] with a method for getting the number of milliseconds elapsed since last timer reset with the best precision provided by the web browser.

5.3.5 Input from pointing devices

Detecting mouse movement and the state of it's buttons is a very common task in web development. Users navigate through web pages mainly by clicking on hypertext links with their computer mouse. Therefore mouse events are well specified and work across all desktop web browsers and desktop platforms.

Unfortunately, the situation among other pointing devices other than computer mice is much less uniform. With the boom of smartphones and tablets, touchscreens are very common. Also computer graphics tablets are used by artists and many people use them when creating a Khan Academy style video.

Wacom Webplugin pen API

The Wacom Webplugin pen API (WebPAPI) is a browser plugin interface for pen data access from all Wacom consumer and professional tablets []24.

Unfortunatelly support for this plugin was discontinued by Chromium and Google Chrome [25] and therefore it should not be relied on.

Touch Events API

Touch Events API is an API for handling touch input from touch screens. The standard is proposed by Apple and is implemented across many platforms and in many mobile web browsers [26] [].

This API supports multiple touches at once, but this feature is not needed and neither implemented in this project. Unfortunately, this API provides no touch pressure information.

Pointer Events API

Pointer Events API is an open API created by Microsoft. It's purpose is to unify the way mouse events, touch screen events, stylus and other (i.e. Kinect) similar ways into one API. This technology is implemented in Internet Explorer and will be also present in the final version of the Microsoft Edge browser. Firefox implements this API, but it is so far accessible only if a specific hidden flag is enabled. Google has announced the intent to also implement this functionality in upcoming releases of Google Chrome across all platforms.

This API also provides pressure information for pointing devices, that support this feature, including Wacom graphics tablets.

[23] implementation can be found in */public/js/app/Helpers/VideoTimer.ts* file
[24] http://www.wacomeng.com/web/WebPluginReleaseNotes.htm
[25] http://blog.chromium.org/2013/09/saying-goodbye-to-our-old-friend-npapi.htm
[26] http://caniuse.com/#feat=touch

5.4 Drawing curves

Khan Academy videos are known for it's consistent and simple style. A person draws on a virtual canvas (evoking a school blackboard) with a brush (or possibly a chalk) of a round shape.

Tutor has a pointing device, typically a computer mouse or digital pen, and it's position on the canvas is marked with a moving cursor. When the tutor clicks, a dot is marked on canvas in the current position of the cursor. Tutor can produce a line (typically a curve) following this cursor when he presses a mouse button or increases digital pen pressure and while moving the cursor. The curve ends when he releases the button or pen pressure. The color and size of the dot or line corresponds to the current settings.

At the time of recording, mouse coordinates relative to the drawing board are captured along with current pressure of the digital pointing device (mouse, graphical tablet pen). This data is then used to draw a curve with variable thickness at the moment of recording as visual feedback for the person recording and the same process is done every time the video is replayed.

The outcome of the rendering phase should be the same every time so the intention of the creator is preserved. On the other hand, the rendering algorithm might be improved (i.e. by making the lines smoother) in the future and the video could be rendered using to this algorithm without any editing, while the information in the video will remain untouched.

Rendering at the time of playback gives us the opportunity to adjust the outcome to the environment of the end user. This means that the result can be sharp on every display resolution without the need of having many versions of the same video for each resolution.

5.4.1 DynaDraw algorithm

Paul Haeberli has created a simple algorithm called *"DynaDraw"* in 1989, which is suitable for calligraphy. Brush is modeled as a physical object with it's mass, velocity and friction coefficient [27] []. Mouse movement is interpreted as a way of exerting force on the brush – the faster you move the brush, the greater the force applied on the brush is. Acceleration is then calculated according to Newton's second law of motion considering brush's mass and velocity of the brush is derived with respect to the amount of brush's friction. This velocity is then applied and brush is moved. The trace brush should leave behind is then drawn onto the blackboard.

The advantage of this algorithm is it's simplicity and the possibilities of configuring the brush with different values of mass and friction (the author of the algorithm refers to this constant also as a drag, which better fits the purpose of slowing down the brush).

Heavier brushes move slowly, but the path they leave behind is much smoother, as the hand shaking is eliminated by composition of forces in different directions.

Light brushes move faster and are often very close to the cursor during the movement. When the cursor stops abruptly, light brushes with little friction tend

[27] DynaDraw: http://www.graficaobscura.com/dyna/index.html

to keep moving past the cursor and wrap around it. This produces little curls at the end of lines.

This approximation of cursor movement with appropriate constants selected improves quality of user's input and leads to nice results event when the user is using an ordinary mouse instead of a digital drawing pen.

Brush movement simulation

One step of simulation applies force on the brush according to current mouse position and thus moves it in the direction of the pointer. This process of applying force must be done periodically, at the frequency of 60 Hz in ideal case[28]. Implementation of this simulation is not identical to the original *DynaDraw* algorithm, but all of it's key aspects are preserved. Pseudocode 1 describes the algorithm used in this theses.

The main difference between the original implementation and the one used in Vector Video is brush width calculation. The original algorithm calculates the width of the line in a specific point by measuring it's velocity – the faster the brush is moving, the thinner the drawn line is. This width dynamics gave the algorithm it's name.

In our implementation, this effect is implemented, but is much more subtle. Brush dynamic in this implementation relies mainly on the pressure of a digital pen on a graphics tablet. Since the exact value of pressure in the point of current brush's location is not always known, the value is linearly interpolated between the value of pressure in the previous position of the brush and the value of pressure in the current mouse position according to distance from each of these points.

As this simulation isn't deterministic, this process cannot be reconstructed afterwards and when the video is being played, only the already computed values of points along the path must be used. The precise value of pressure is therefore redundant for later playback of the video and

Rendering of one line segment

Line consists of many segments that are drawn after every simulation step, which causes brush to move. There are several ways to draw the segment, the most straightforward is to draw a simple quadrilateral between two points . as shown in Figure 5.1 and Pseudocode 2.

[28]HTML5 provides requestAnimationFrame function, which is intended for animations and which targets 60 frames per second

Pseudocode 1 One step of brush movement simulation

```
function OneStep(M, Δt)                    ▷ M - mouse position, Δt - elapsed time
    if M ≠ ∅ then
        brushMoved ← apply(M⃗, Δt)
        if brushMoved = true then
            DrawSegment
        end if
    end if
end function
function apply(M, Δt)
    F⃗ ← M − P                              ▷ P - current position of the brush
    a⃗ = F⃗/m                               ▷ m - mass of the brush
    if ‖a⃗‖ ≤ C_a then                      ▷ C_a - minimum acceleration constant
        return false;
    end if
    v⃗ ← v⃗ + a⃗
    if ‖v⃗‖ > C_v then                      ▷ C_v - minimum velocity constant
        P ← P + μΔtv⃗                       ▷ μ - coefficient of friction
        return true
    end if
    return false
end function
```

Figure 5.1: Drawn segment - quadrilateral

Pseudocode 2 Draw one segment of a line

```
function DrawSegment
    n⃗ ← (−v⃗_y, v⃗_x) / ‖v⃗‖
    w ← CurrentBrushPressure · b           ▷ b - brush size
    L ← P − wn⃗
    R ← P + wn⃗
    BeginPath
    MoveTo(L′)                             ▷ L′ and R′ - previousely drawn point
    LineTo(R′)
    LineTo(R)
    LineTo(L)
    ClosePath
    Fill(c)                                ▷ c - current brush color
    L′ ← L
    R′ ← R
end function
```

@todo: splines, bezier curves, curved segment

6. Integration into other applications

How to integrate the player and the recorder into another web application.

6.1 Video player

The frontend. And a short comment on how to deal with large data transfers - CDN like Amazon S3, Google App Platform, Microsoft Azure...

6.2 Video recording

The frontend. And the server program. A short comment on how to implement the the website backend - insert the data into a database.

7. Testing

7.1 Comparint the tool to youtube

- File size
- Subjective quality comparison - test subjects.
- Preloading speed - how long does it take for the video to start playing?
- Mobile devices and tablets - viewing on small display - 4", 5", 7".

8. Users' documentation

Users' documentation with screenshot similar to the one I have already done.

9. Programmers' documentation

Programmers' documentation with screenshot similar to the one I have already done.

Conclusion

Bibliography

[1] Bower, Beverly L. Hardy, Kimberly P. *From correspondence to cyberspace: Changes and challenges in distance education* [online]. New Directions for Community Colleges, Volume 2004, Issue 128, pages 5–12.

[2] John Oliver: Student Debt, HBO [online] 9/7/2015, available at https://www.youtube.com/watch?v=P8pjd1QEA0c

[3] Bill Gates: Online, All Students Sit in the Front Row [online], posted on November 18, 2014, seen on 3/12/2015, available at http://www.gatesnotes.com/Education/Colleges-Without-Walls-Arizona

[4] Jessica Leber: In the Developing World, MOOCs Start to Get Real [online], MIT Technology Review, published on March 15, 2013, available at http://www.technologyreview.com/news/512256/in-the-developing-world-moocs-start-to-get-real/

[5] Kepler [online], http://kepler.org/

[6] Generation Rwanda [online], http://www.generationrwanda.org/

[7] Moodle, https://moodle.org

[8] Moodle statistics, https://moodle.net/stats/

[9] Coursera [online], 3/13/2015, available at https://www.coursera.org/about/

[10] Youtube.com, https://www.youtube.com

[11] Alexa Internet, http://www.alexa.com/topsites

[12] Numberphile Youtube channel, https://www.youtube.com/user/numberphile

[13] Veritasium Youtube channel, https://www.youtube.com/user/veritasium

[14] Khan Academy Youtube channel, https://www.youtube.com/user/khanacademy

[15] Khanova škola, https://khanovaskola.cz

[16] Oracle, Java.com, How do I get Java for Mobile device? [online] viewed April 16, 2015 http://www.java.com/en/download/faq/java_mobile.xml

[17]

[18] Steve Jobs, Apple.com, Thoughts on Flash, published April 2010, viewed April 16, 2015 [online] http://www.apple.com/hotnews/thoughts-on-flash

[19] Adobe Blog, An Update on Flash Player and Android, published June 28, 2012, viewed April 16, 2015 [online] http://blogs.adobe.com/flashplayer/2012/06/flash-player-and-android-update.html

[20] Bosomworth Danyl, SmartInsights.com, Mobile Marketing Statistics 2015, published January 15, 2015, viewed April 16, 2015 [online] http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/

[21] Microsoft Silverlight, http://www.microsoft.com/silverlight/

[22] Microsoft Silverlight, Get Microsoft Silverlight, viewed April 16, 2015 [online]

List of Tables

List of Abbreviations

Attachments