

Czech University of Life Sciences Prague
Faculty of Economics and Management
Department of Information Technologies



Diploma Thesis

The Usage of HTML5 Technology in Term of Web Accesibility

Bc. Michal Bouša

Diploma Thesis Supervisor: Ing. Petr Benda, Ph.D.

Course: Informatics

© 2013 Prague

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE
Department of Information Technologies
Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Bouša Michal

Informatics

Thesis title

The usage of HTML 5 technology in term of web accessibility

Objectives of thesis

The first diploma thesis's goal will be to concisely describe HTML5 and CSS3.

The second goal will be to show potential advantages of these technologies from the perspective of Web Accessibility.

The third goal will be to create accessible web pages with multimedia content which is based on the use of HTML5 and CSS3. This goal also contains accessibility testing.

Methodology

The methodology of this study is based on analysis and synthesis of technical information resources dealing with selected issues. Acquired knowledge will be applied to the creation and testing of accessible website established on the basis of HTML5 technology.

Schedule for processing

January to September 2012 - Preparation and studying technical information sources

October 2012 - Processing review issues according to Information Resources

November-December 2012 - Creation of the practical part

January 2013 - Conclusion and completion of the work

The proposed extent of the thesis

60 - 80 pages

Keywords

HTML5, CSS3, Accessibility of Web Pages, Handicapped User,

Recommended information sources

ŠPINAR, David. Tvoříme přístupné webové stránky. Brno: ZONER software s.r.o., 2004. ISBN 80-86815-11-0.

LUBBERS Peter, Brian ALBERS, Frank SALIM. Pro HTML5 Programming. New York: Apress Media LLC, 2011, 352 Pages. ISBN: 978-1-4302-3864-5.

HOGAN, Brian P. HTML5 a CSS3 : výukový kurz webového vývoje. Brno: Computer Press, 2011, 272 pages. ISBN 978-80-251-3576-1.

CEDERHOLM, Dan. CSS3 for Web Designers. New York: A Book Apart, 2010, 120 pages. ISBN 978-0-9844425-2-2.

KEITH, Jeremy. HTML5 for Web Designers. New York: A Book Apart, 2010, 85 pages. ISBN 978-0-9844425-0-8.

PFEIFFER, Silvia. HTML5 - audio a video, kompletní průvodce. Brno: ZONER software, a.s., 2011. ISBN 978-80-7413-147-9.

O CONNOR, Joshue. Pro HTML5 Accessibility. New York: Apress Media LLC, 2012, 386 Pages. ISBN13: 978-1-4302-4194-2

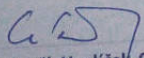
BOWERS, Michael, Dyonisios SYNODINOS, Victor SUMMER. Pro HTML5 and CSS3 design patterns New York: Apress Media LLC, 2012, 504 Pages. ISBN 978-1-4302-3780-8.

The Diploma Thesis Supervisor

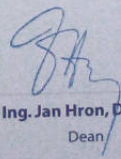
Benda Petr, Ing.

Last date for the submission

March 2013


doc. Ing. Zdeněk Havlíček, CSc.
Head of the Department




prof. Ing. Jan Hron, DrSc., dr.h.c.
Dean

Prague January 15, 2013

Declaration

I declare that I have worked on my diploma thesis titled „The Usage of HTML5 Technology in Term of Web Accesibility“ by myself and I have used only the sources mentioned at the end of the thesis.

In Prague 29th November 2013

.....

Bc. Michal Bouša

Acknowledgement

I would like to thank to Ing. Petr Benda, Ph.D. for his advices and supervising of my diploma thesis and I would like to thank my family for the valuable support.

The Usage of HTML5 Technology in Term of Web Accessibility

Annotation

The topic of this thesis is the use and describe benefits of HyperText Markup Language 5 (HTML5) from the perspective of web accessibility. The theoretical part is devoted to HTML5, CSS3 and other issues related to web accessibility. This section also provides an overview of the new HTML5 elements and properties.

The practical part is focused on creating accessible web pages with multimedia content by using new elements. This part will include accessibility testing.

Keywords

HTML5, CSS3, Accessibility of Web Pages, Handicapped User

Využití technologie HTML5 z hlediska přístupnosti webových stránek

Anotace

Tématem této diplomové práce je využití a výhody technologie HyperText Markup Language 5 (HTML5) z hlediska přístupnosti webových stránek. Teoretická část je věnována HTML5, CSS3 a dalším tématům souvisejícím s přístupností. Tato část přináší i přehled nových elementů a vlastností HTML5.

Praktická část je věnována tvorbě přístupných webových stránek s multimediálním obsahem, které využívají nové elementy. Součástí praktické části bude i testování přístupnosti.

Klíčová slova

HTML5, CSS3, Přístupnost webových stránek, Hendikepovaný uživatel

Contents

1	Introduction	1
2	Objectives and Methodology	2
2.1	Objectives	2
2.2	Methodology	2
3	Web Accessibility	4
3.1	Handicapped Users of the Internet	5
3.1.1	Users with Ophtamical Impairments	5
3.1.2	Users with Hearing Impairment	6
3.1.3	Users with Physical Impairment	6
3.1.4	Cognitive Disorders and Learning Disabilities	6
3.2	Assistive Technology	7
3.2.1	Software tools	7
3.2.2	Hardware tools	7
3.3	Methodologies for Creating Accessible Websites	8
3.3.1	Web Content Accessibility Guidelines 2.0	8
3.3.2	Section 508	9
3.3.3	Bliend Friendly	9
4	HTML5 and CSS3	10
4.1	HTML5	10
4.2	New Elements	11

4.2.1	Web Page Structure	11
4.2.2	Another New Tags	13
4.3	HTML5 and Multimedia	14
4.3.1	Audio	14
4.3.2	Video	15
4.3.2.1	Element track	17
4.3.3	Canvas	18
4.4	CSS3	19
4.5	JavaScript	22
4.6	WAI-ARIA	23
5	Benefits of HTML5 and CSS3	24
6	Practical Realization	26
6.1	Movie Player	29
6.1.1	Creation of Interface	32
6.2	Styling of pages	35
6.3	Audio Player	36
6.4	Testing of Accessibility	37
6.4.1	Testing Tools	37
6.4.1.1	User Testing	39
6.4.1.2	Test Results	39
7	Conclusion	39
	Bibliography	42
	List of Figures	i
	List of Source Codes	iii
	Content of DVD - Appendix A	I

Acronyms - Appendix B

II

Screenshots - Appendix C

III

1 Introduction

Internet has become an integral part of daily life. We use it for sharing ideas, reading news, fast communicating, presentation our creations, banking services, shopping etc. Every user of the Internet is different and they also have different needs. They have a dissimilar level of knowledge, diverse output devices and software but also different health status.

Internet has also become a place of great competition among sellers. Who wants to be successful must try to provide products or services to each possible customers and users. Accessibility, usability and attractive design have become important part of web development. Accessible web pages have some benefits. They usually provide better usability and UX. They also fullfil requirements for accessibility from governments in the form and provide equal opportunities for people with some disabilities or unusual output devices.

Significant proportion of Internet content is made up of multimedia contents. People can watch videoclips, news, recorded lectures and even full movies. They can also listen to music, play online games and use other interactive services possible. A considerable number of these pages have inaccessible content.

Accessibility of multimedia content is still sometimes problematic. One of causes is use of technologies that are under control individual companies. Users have to install their plugins and output devices not always supports these technologies. This diploma thesis is mainly focused on new HTML5 technology and its potential benefits for web accessibility.

2 Objectives and Methodology

2.1 Objectives

The first and second objective will be theoretical analyzed. The first objective is to describe HTML5 and CSS3. We will also point out how they are able to change the possibilities of web development.

The second objective will be to define potential advantages of these technologies from the perspective of web accessibility.

The last task is to convert theory into functional practice. We will develop accessible web pages which will be based on HTML5, CSS3 and acquired knowledge. The output will be web pages with audio and video content. In this part we will also do tests of accessibility.

2.2 Methodology

The methodology of this study is based on analysis and synthesis of technical information resources dealing with selected issues. Acquired knowledge will be applied to the creation and testing of accesible website established on the basis of HTML5 technology.

We will use professional literature, articles from professional servers and online specifications of used technologies for theoretical parts. In chapter 3 we will describe terms like web accessibility, handicapped user and assitive technology. There will be also briefly mentioned the methodologies of creating accesible web pages.

We will describe HTML5 and CSS3 in chapter 4. We will introduce new elements for clearer structure of a document and we will focus on the elements that allow work with

2. Objectives and Methodology

multimedia files. This chapter will include some practical examples of the HTML5 and CSS3 technology and their description. Small space will be devoted to the topic WAI-ARIA.

In chapter 5 we will characterize advantages and benefits of these technologies ifrom the perspective of accessibility and availability of website content.

In the sixth chapter titled Practical Part we will develop website with video and audio players. We will use acquired knowledge to create alternative website for Sintel. It is short computer animated film made by the Blender Institute. In this chapter we will do some test of accessibility with special online tools and toolbars. We will also present outputs of simple user testing which will be done by handicapped users.

The last chapter will contain the conclusions of diploma thesis. We will summarize outputs and suggest possible extension of this work.

3 Web Accessibility

There are several definitions of (web) accessibility. *“The usability of a product, service, environment or facility by people with the widest range of capabilities.”* (ISO TC 16071, 2003).

[1] Another possible definition can be *“Web accessibility means that people with disabilities can use the Web. More specifically, Web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web. Web accessibility also benefits others, including older people with changing abilities due to aging.”* (WAI Introduction to Accessibility, 2010). [1]

There are some benefits of accessible web pages:

- Services or products can be offered for more customers.
- Accessible web pages usually provide better usability and UX.
- Accessible web pages fulfill requirements for accessibility from governments in the form of laws (public administration etc.)
- Accessible web pages provide equal opportunities for people with disabilities by removing barriers to communication and interaction. [1]

3.1 Handicapped Users of the Internet

Handicapped users of the Internet are wider group than expected users with ophtamical impairments. „*Most of the time when people think of Web accessibility, they think of blind people! You could be forgiven for thinking this was the case—because blind users are often the most vocal group online. However, Web accessibility is not solely about blind computer users. When you read a lot of discussion online about both subjects, don't fall into the trap of associating Web accessibility with only blind people and screen readers. The idea that “If it works with a screen reader, it is accessible” is only partially true.*“ [1]. It is obvious that this group contains users with hearing impairment, physical impairment, cognitive disorders or learning disabilities.

3.1.1 Users with Ophtamical Impairments

They have a major problem because internet is primarily a visual medium.

Examples of vision impairments:

- Glaucoma
- Macular Degeneration
- Retinopathy
- Detached Retina



Figure 3.1: Garden seen by someone with macular degeneration [33]

3.1.2 Users with Hearing Impairment

In this case, they are people with temporary or permanent hearing impairment or users without reproducing device. A typical problem is the podcast without a text transcript or video without subtitles.

3.1.3 Users with Physical Impairment

Users with temporary or permanent physical injuries may have difficulties to control the websites. They usually use some special devices (trackball, speical keyboards etc.).

3.1.4 Cognitive Disorders and Learning Disabilities

Users suffer from learning difficulties or problems with concentration. They have problems to perform certain mental tasks as:

3. Web Accessibility

- read the text;
- concentrate;
- solve mathematical tasks;
- perceive visual information.

These users usually don't use any assistive technology.

3.2 Assistive Technology

Handicaped users use assistive technology for browsing web pages and using web applications. Here is one possible defenition: *“A term used to describe all of the tools, products, and devices, from the simplest to the most complex, that can make a particular function easier or possible to perform.”* US National Multiple Sclerosis Society. Assistive technology can be divided into software and hardware tools.

3.2.1 Software tools

Software tools are represented by screen readers, screen magnifier etc. Screen readers (JAWS, VoiceOver etc.) read aloud content of web page and indicate semantic meaning of HTML tags (link, list, heading etc.). Screen magnifiers are useful for zooming of web page content.

3.2.2 Hardware tools

Typical hardware tool is braille display. It's electronic device with small plastic or metal pins that move up and down to display the braille characters. User can read content of the web page.



Figure 3.2: Braille display [32]

3.3 Methodologies for Creating Accessible Websites

Methodologies consist guidelines and instructions for making web pages accessible for the larger number of users.

3.3.1 Web Content Accessibility Guidelines 2.0

Web Content Accessibility Guidelines (WCAG) 2.0 is list of recommendations and instructions how to make web pages accessible for a larger number of users. WCAG 2.0 is organized around four simple principles:

Principle 1: Content must be Perceivable. This principle refers to all content, including multimedia, video and audio:

- Provide a text alternative for all non-text content.
- Provide synchronized alternatives for multimedia (such as captioned video, audio descriptions, and so forth).
- Information and structure must be separate from presentation.
- Make it easy to distinguish foreground information from the background (for example,

3. Web Accessibility

have good color contrast).

Principle 2: Interface elements must be Operable:

- All functionality must be operable via the keyboard.
- Users must be able to control limits on their reading or interaction.
- Provide mechanisms for users to find content, orientate themselves, and navigate through it.
- Users must be able to avoid content that can cause seizures due to photosensitivity.
- Help users avoid mistakes, and make it easier to correct mistakes when they do occur.

Principle 3: Content and controls must be Understandable:

- Make text content readable and understandable.
- Make the placement and functionality of content predictable.

Principle 4: Content should be robust enough to work with current and future technologies:

- Support compatibility with current and future user agents.
- Ensure that content is accessible, or provide accessible alternatives (from WAI Introduction to WCAG, 2010).

3.3.2 Section 508

American rules was created in 2000. „*Section 508 requires that Federal agencies’ electronic and information technology is accessible to people with disabilities.*“ [1] Rules are not only for websites of offices, but they are also for telecommunication devices, applications and hardware etc.

3.3.3 Blind Friendly

Blind Friendly Web is a Czech methodology for creating accessible web. It was created by Mgr. Radek Pavlíček. Methodology is available from <http://blindfriendly.cz/>.

4 HTML5 and CSS3

„HTML5 and CSS3 are more than just two new standards proposed by the World Wide Web Consortium (W3C) and its working groups. They are the next iteration of technologies you use every day, and they’re here to help you build better modern web applications.“ [4]

4.1 HTML5

HTML5 will be the new standard for HTML. Most of HTML 4.01 is used in HTML5.

„HTML5 isn’t a completely new language created from scratch. It’s an evolutionary rather than revolutionary change in the ongoing story of markup. If you are currently creating websites with any version of HTML, you’re already using HTML5.“ [2]. Unfortunately, some of the tags are not supported yet. There are many sites allowing test of support HTML5. For example current state is available from <http://html5test.com/>.



Figure 4.1: Logos of HTML5 and CSS3 [31]

4.2 New Elements

HTML5 includes new elements for adding media content, better page structure, drawing graphics, better form handling, etc. Some elements has been simplified for clearer html code.

Source code 4.1: HTML doctypes

```
<!--Here is doctype for XHTML 1.0 Strict:-->

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!--Here is doctype for HTML5:-->

<!DOCTYPE html>
```

Second example is link to a CSS file:

Source code 4.2: Link to CSS

```
<!--Here's the old way:-->

<link rel="stylesheet" type="text/css" href="style.css">

<!--Here is simplier version in HTML5:-->

<link rel="stylesheet" href="style.css">
```

4.2.1 Web Page Structure

HTML5 brings group of new tags which has been created for clearer web page structure. They could substitute non semantic tag like `div` or `span` by semantic elements. New semantic tags:

- `<header>` - represents a group of introductory or navigational aids. [1]
- `<footer>` - represents footer of page. It can contains information about page, author etc.
- `<section>` - represents of web pages sections. It shows the document structure.

4. HTML5 and CSS3

- `<aside>` - represents a piece of content that is only slightly related to the rest of the page. [1]
- `<nav>` - represents a section of the document intended for navigation. [1]
- `<article>` - represents piece of content which is independent of the rest of page.
- `<figure>` - represents some media content, video, image, graph, code sample etc.
- `<figcaption>` - it provides caption for `<figure>`.

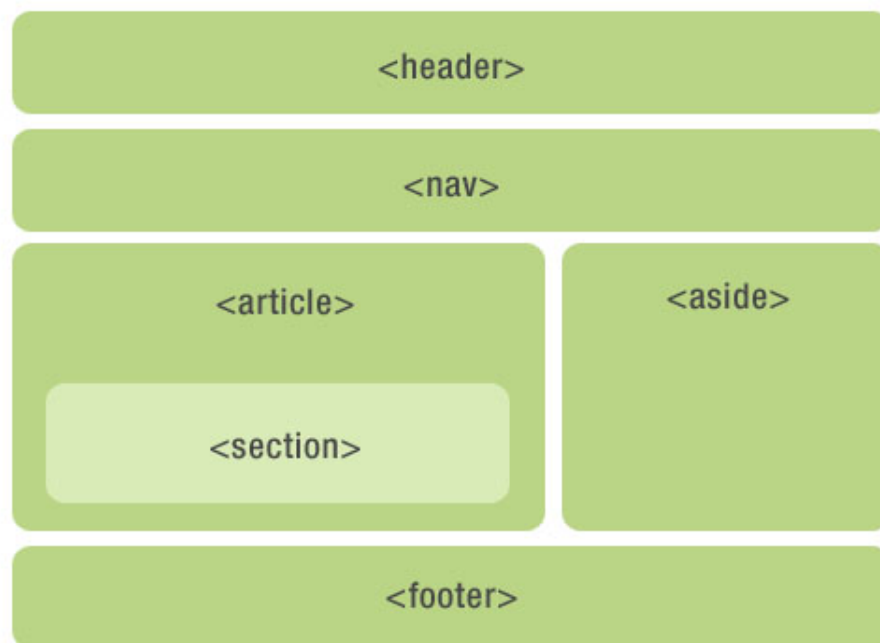


Figure 4.2: Example of possible structure of web page with new HTML5 tags [30]

New semantic elements weren't created just to replace `div` elements because `div` may still be useful for page formatting by CSS.

4.2.2 Another New Tags

HTML5 doesn't bring only structure tags. It provides new inline elements but some of them aren't fully supported yet.

- `<time>` - represents time, date and time duration data
- `<bdi>` - tag isolates a part of text that might be formatted in a different direction from other text outside it
- `<command>` - defines a command that user can invoke
- `<details>` - gives the option to show or hide additional informations
- `<summary>` - is visible heading for the `<details>` tag
- `<mark>` - is for highlighting part of text
- `<meter>` - is used for indicating a scalar measurement within a known range, or a fractional value [5]
- `<progress>` - represents the progress of a task
- `<ruby>` - is used for specifying Ruby annotations, which is used in East Asian typography [6]
- `<rt>`, `<rp>` - it's used for ruby annotations
- `<wbr>` - specifies a line break opportunity

Some old tags have been removed from HTML5: `<basefont>`, `<big>`, `<center>`, `<dir>`, ``, `<frame>`, etc.

4.3 HTML5 and Multimedia

Multimedia content like video or audio is important and popular part of the Internet. Users usually use YouTube, Vimeo and other multimedia players based on third-party technologies such as Flash or Silverlight. *„Plug-in is an accurate term for these technologies—they help to fill the holes on the web. They make it relatively easy to get games, films, and music online. But these technologies are not open. They are not created by the community. They are under the control of individual companies.“* [2] Users can't use the website without the required technology. This is one of the reason why HTML5 introduces new elements for embedding audio, video and interactive content into a page.

4.3.1 Audio

The element `<audio>` is used for embedding an audio file. It can be used for easy spreading music and podcasts etc. Unfortunately browsers support different audio formats. It's necessary to have an audio files in multiple formats.

Browser	.aac	.mp3	.ogg	.webm
Chrome 12+	X	X	X	X
Firefox 3.6+			X	X
IE9+	X	X		
Opera 10.6+			X	X
Safari 5+	X	X		

Figure 4.3: Audio formats and browsers support [29]

4. HTML5 and CSS3

Example of usage:

Source code 4.3: Element audio

```
<audio controls preload>

  <source src="example.mp3" type="audio/mpeg">

  <source src="example.ogg" type="audio/ogg">

  <source src="example.wav" type="audio/wav">

  Download <a href="example.ogg">audio example</a>.

</audio>
```

Element `<audio>` has several attributes. Attribute `controls` gives users control over playback like play, volume or pause. Next attribute is `autoplay`. Audio will start playing as soon as it is ready but it's better to remove this attribute. `Autoplay` may break some of the accessibility rules (for example 3.2.2 rule of WCAG 2.0). Attribute `loop` plays audio files in a loop and attribute `src` specifies location of audio files. The remaining attributes are `muted` and `preload`. `Muted` specifies that the audio output should be muted. Attribute `preload` starts loading audio content when the page loads.



Figure 4.4: Example of HTML5 audioplayer (Mozilla Firefox 23) (Author)

4.3.2 Video

The element `video` works similarly like the `audio` element. It is supported by Internet Explorer 9 and 10, Firefox, Chrome, Safari and Opera. Internet Explorer 8 and its earlier versions don't support the `<video>` element. There is the same problem with the formats like `<audio>` element.

4. HTML5 and CSS3

Browser	MP4	WebM	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	NO Update: Firefox 21 running on Windows 7, Windows 8, Windows Vista, and Android now supports MP4	YES	YES
Safari	YES	NO	NO
Opera	NO	YES	YES

Figure 4.5: Video Formats and Browsers Support [25]

Example of usage:

Source code 4.4: Element video

```
<video controls width="800" height="600" poster="example.jpg">

  <source src="example.ogv" type="audio/ogg">

  <source src="example.mp4" type="audio/mpeg">

  <source src="example.webm" type="audio/webm">

  Download <a href="example.ogv">video example</a>.

</video>
```

Attribute `controls` gives users control over playback. Attributes `width` and `height` define dimensions of player in web browser and `poster` adds chosen representative image. Other attributes are the same as in `<audio>` element.

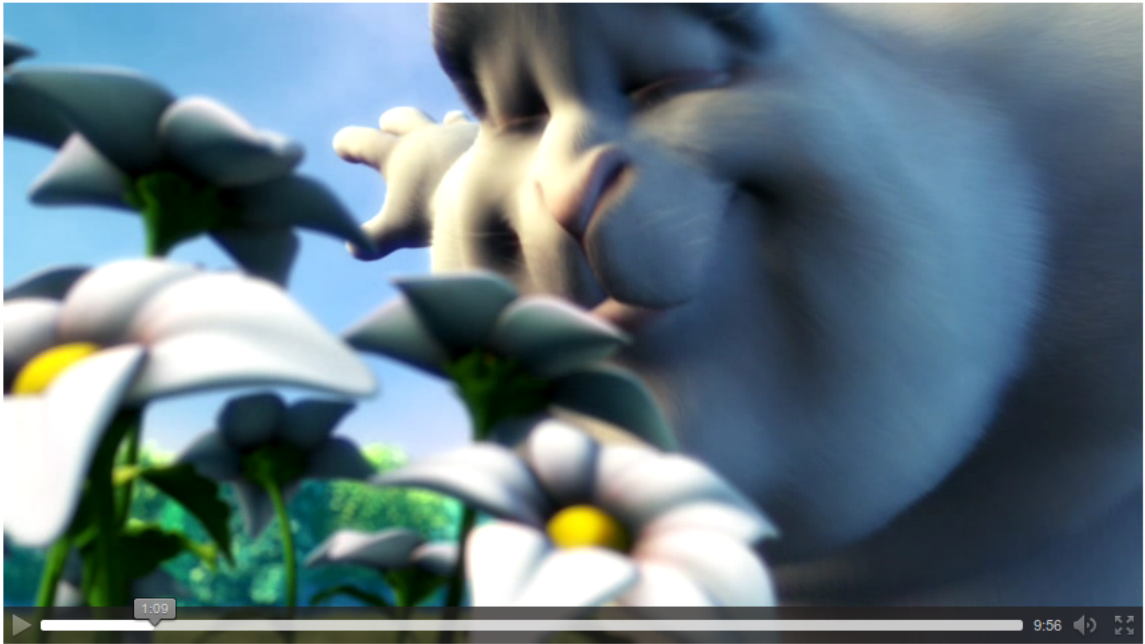


Figure 4.6: Example of HTML5 videoplayer (Mozilla Firefox 23) (Author)

4.3.2.1 Element track

We use `<track>` to specify subtitles, caption files or other files containing text that should be visible when the media is playing. [26] Unfortunately only Internet Explorer 10, Opera, and Chrome support `<track>` tag. We can modify text by CSS so we can separate characters by color etc. Ruby tags are used for asiag languages. Attribute `track` define type of text.

Possible values of kind attribute:

- **subtitles** - translation of the dialogue
- **captions** - translation of the dialogue, sound effects, relevant musical cues, and other relevant audio information [35]
- **descriptions** - textual descriptions of the video component of the media resource [35]
- **chapters** - chapter titles, intended to be used for navigating the media resource [35]
- **metadata**



Figure 4.7: Videoplayer with arabian subtitles (Chromium 30) (Author)

4.3.3 Canvas

The `<canvas>` element is an environment for creating dynamic images [2]. Content is created and performed by JavaScript. It's useful tool for creating web games, graphs, animations and other interactive content. *„However, from an accessibility perspective (to make life easier for everyone), don't use `<canvas>` for anything more than pretty pictures. At the time of this writing, it cannot be considered accessible.“* [1]

Example of usage:

Source code 4.5: Element canvas

```
<canvas width="640" height="480">
  

  Sorry, your browser doesn't support canvas element.
  

</canvas>
```

4.4 CSS3

CSS3 is third version of CSS. It's compatible with earlier versions of CSS. It still under development. It started in 2005 and its end is expected in 2015. Major modern browsers support many of the new properties.

Current state is available from: http://www.w3schools.com/cssref/css3_browsersupport.asp.

CSS3 brings more text effects, gradients, animations, transformations of objects, multiple columns and new background or borders properties, etc.

Example of usage CSS3:

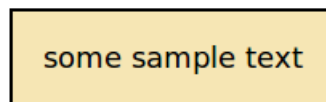


Figure 4.8: Simple button created by older version of CSS (Author)

Gradients are smooth transitions between two or more colors.

Source code 4.6: Defining of a linear gradient

```
.button { linear-gradient(to bottom, #f6e6b4 0%, #ed9017 100%); }
```



Figure 4.9: Button with gradient, (Author)

4. HTML5 and CSS3

Rounded corners are popular element of web design. Designers usually used background images for creating rounded corners. CSS3 brings property `border-radius`.

Source code 4.7: Defining of rounded corners

```
.button { border-radius:25px; }
```

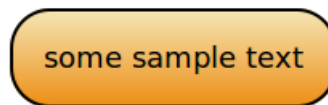


Figure 4.10: Button with rounded corners (Author)

Last step is adding special font. Before CSS3, web designers had to use fonts that were already installed on the user's computer.[3] They usually used images for logo or decorative lettering but now they can choose from a large number of paid or free fonts. Fonts are defined in `@font-face` rule. Web browsers support different formats of fonts.





	OTF/TTF	SVG	WOFF	EOT
	IE9		IE9	IE5
	FF3.5	FF3.5	FF3.6	
	Chrome 4	Chrome 0.3	Chrome 5	
	Safari 3.1	Safari 3.1	Safari 5.1	
	Opera 10	Opera 9	Opera 11.01	
	iOS 4.2	iOS 1		
	2.2			

Figure 4.11: The major web browsers support different formats of web font [28]

4. HTML5 and CSS3

Source code 4.8: Definition and use custom font

```
@font-face {  
  
    font-family: 'desyrelregular';  
  
    src: url('desyrel-webfont.eot');  
  
    src: url('desyrel-webfont.eot?#iefix') format('embedded-opentype'),  
  
    url('desyrel-webfont.woff') format('woff'),  
  
    url('desyrel-webfont.ttf') format('truetype'),  
  
    url('desyrel-webfont.svg#desyrelregular') format('svg');  
  
    font-weight: normal;  
  
    font-style: normal; }  
  
body { font-family: 'desyrelregular', sans-serif; }
```



Figure 4.12: Final version of button, (Author)

This examples shows that designers can create website templates without images or third-party technologies. It's also useful for users. They need not to use images and third-party plugins.

4.5 JavaScript

JavaScript has been known as *"very dirty word in accessibility circles"*. [1] As we can see on graph most of the screen reader users use JavaScript. This implies that we can develop multimedia web sites using JavaScript. We can use it for developing buttons of the video player.

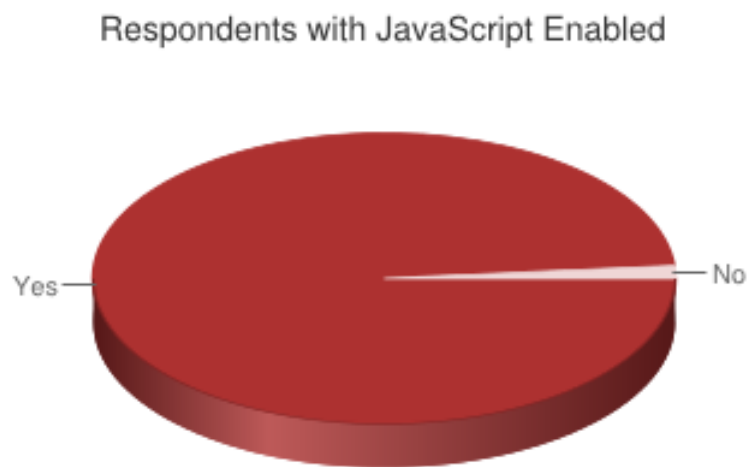


Figure 4.13: Respondents with JavaScript Enabled [27]

"Most of the accessibility problems that arise from the use of JavaScript come about because it has been used incorrectly, or because there is a general lack of awareness about the impact your code can have on the user experience for people with disabilities in particular. It is largely understandable how this happens, because the developers might just be unaware that how they are actually applying scripting is even a problem" [3]

4.6 WAI-ARIA

WAI-ARIA brings `role` attribute and offers the ability to assign web page areas information about what is their role. WAI-ARIA is supported in most current versions of browsers and assistive technologies. Examples of values of the attribute `role`:

- `navigation` - area of the page which contains the navigation links or navigation
- `main` - main content of page
- `search` - search form
- `contentinfo` - footer - area of the page with information about copyright, contact, etc.
- `complementary` - area of the page with additional content
- `banner` - area of the page that is used to identify the site (site name or logo)
- `article` - page area which gives meaning itself independently of the rest of the document (blog post etc)
- `application` - area of application

Source code 4.9: WAI-ARIA roles

```
<header role="banner">

  <h1>Heading 1</h1>

  <nav role="navigation">

    <ul>

      <li><a href="index.html">News</a></li>

      <li><a href="ourservices">Our Services</a></li>

      <li><a href="faq.html">F.A.Q.</a></li>

      <li><a href="contact.html">Contact</a></li>

    </ul>

  </nav>

</header>
```

5. Benefits of HTML5 and CSS3

```
</ul>
```

```
</nav>
```

```
</header>
```

5 Benefits of HTML5 and CSS3

"The range of web-enabled devices is enormous and getting broader by the day." [8] As we can see people own a large number of devices that have a web browser: a desktop, a laptop, a tablet, a smartphone, an ebook reader, a games console, a smart TV and portable games console etc. Applications, games and multimedia players are an integral part of the Internet. Many of them are based on the technology of third parties.

We can't expect that every browser have installed third-party plugins or they even support these technologies so we can use HTML5 and JavaScript for replace it. *"It is in direct competition with proprietary technologies like Flash and Silverlight. But instead of requiring a plug-in, the rich media elements in HTML5 are native to the browser."* [2] We can create keyboard-accessible content with audio and video.

HTML5 also offers new semantic elements for a clearer definition of different parts of a web page. *"The new structural elements in HTML5 will be very useful to assistive technology. Instead of creating "skip navigation" links, all we need to do is use the nav element correctly. This will allow screen reader users to skip past navigation without us having to provide an explicit link. That's the plan, anyway. For now, we must make do with the technologies that browsers and screen readers support today."* [2] For this reason we can add WAI-ARIA roles into the code and our document structure will be fully accessible for present and future assistive technology.

The old way how to create website was combination bitmap graphics, HTML and CSS code. As mentioned users utilize a large range of devices with various screen dimensions. We can develop adaptable layouts by using the @media rule, relative units or flexible images. Usage new CSS3 properties also provides better usability and accessibility. Using images

6. Practical Realization

isn't essential for creating boxes with rounded corners, text effects, gradients, animations and other visual elements. It is also possible to use our own or external fonts for creating for example logo which makes the text accessible and permits its further processing by assistive technology or another software.

As we can see HTML5 and CSS3 bring several benefits in terms of web accessibility and UX but on the other hand some problems can be found. The biggest pitfall is browsers support or standardization of elements and technologies. We aren't able to use some elements just because of the fact that support and standards are missing or they aren't supported.

6 Practical Realization

We create alternative web page for Sintel movie (complete website and source codes are located on the enclosed CD). *"Sintel is an independently produced short film, initiated by the Blender Foundation as a means to further improve and validate the free/open source 3D creation suite Blender. With initial funding provided by 1000s of donations via the internet community, it has again proven to be a viable development model for both open 3D technology as for independent animation film."* zdroj:<http://www.sintel.org/about/> Official page provides links for download or link to YouTube.

We start with index page. We define basic structure of page using new structural elements and doctype. We use `<header>`, `<nav>`, `<article>`, `<header>` and `<footer>` tags.

Source code 6.1: Basic html document

```
<!DOCTYPE HTML>

<html lang="en">
<head>
<meta charset="utf-8">
  <title>Sintel, the Durian Open Movie Project</title>
</head>
<body>

  <header>
    <h1>SINTEL</h1>
    <nav>
      <ul>
        <li><a href="index.html">Movie&nbsp;Player</a></li>
        <li><a href="soundtrack.html">Soundtrack</a></li>
        <li><a href="gallery.html">Gallery</a></li>
        <li><a href="about.html">About&nbsp;Sintel</a></li>
      </ul>
    </nav>
  </header>

```

6. Practical Realization

```
        </ul>
      </nav>
    </header>

<article>
  <h1>Movie Player</h1>
  <video> </video>
</article>

  <footer>
    <p>&copy;&nbsp;&copy;copyright Blender Foundation |
    <a href="http://www.sintel.org/">www.sintel.org</a></p>
  </footer>

</body>
</html>
```

These tags have semantic meaning resulting from their names. However some assistive technologie don't support them so we should add WAI-ARIA roles. They provide useful information about purpose of specific areas of web pages.

Source code 6.2: HTML document with WAI-ARIA-roles

```
<!DOCTYPE HTML>

<html lang="en">
<head>
<meta charset="utf-8">
  <title>Sintel, the Durian Open Movie Project</title>
</head>
<body>

  <header role="banner">
    <h1>SINTEL</h1>
    <nav role="navigation">
      <ul>
        <li><a href="index.html">Movie&nbsp;Player</a></li>
        <li><a href="soundtrack.html">Soundtrack</a></li>
        <li><a href="gallery.html">Gallery</a></li>
        <li><a href="about.html">About&nbsp;Sintel</a></li>
      </ul>
    </nav>
```

6. Practical Realization

```
</header>

<article role="main">
  <h1>Movie Player</h1>
  <video> </video>
</article>

<footer role="contentinfo">
  <p>&copy;&nbsp;&copy;right Blender Foundation |
  <a href="http://www.sintel.org/">www.sintel.org</a></p>
</footer>

</body>
</html>
```

6.1 Movie Player

We add tag `<video>` to our page and we define some attributes like `width`, `height` for dimensions of video player and `poster` for decoration. Attribute `preload` loads video when the page loads. We must write some alternative text for older browsers which don't support `<video>` element. We also add link for download a movie in MP4 format.

Source code 6.3: Video player with defined attributes

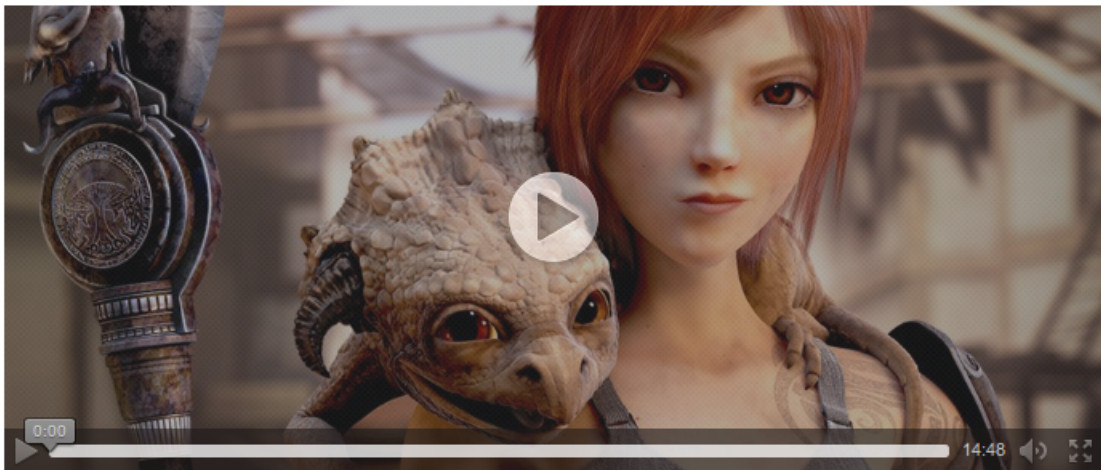
```
<video width="686" height="292" preload controls poster="img/poster.png" >
  <source src="video/sintel.webm" type="video/webm" />
  <source src="video/sintel.mp4" type="video/mpeg" />
  <p>Oops! Your browser doesn't support <code>video</code> element.
  You can download <a href="video/sintel.mp4">Sintel movie</a>.
</p>
</video>
```

The result (Figure 6.1) is displayed in Mozilla Firefox 25. It looks nice but some browsers have video and audio players which are not controllable via keyboard. They also have a low color contrast between controls and background. These are reasons why we create our own controls.

SINTEL

- [Movie Player](#)
- [Soundtrack](#)
- [Gallery](#)
- [About Sintel](#)

Movie Player



© copyright Blender Foundation | www.sintel.org

Figure 6.1: Videoplayer in Mozilla Firefox 25 (Author)

6. Practical Realization

We use JavaScript for developing player buttons (complete source codes are located on the enclosed CD). The HTML5 DOM has methods, properties, and events for the `<audio>` and `<video>` elements so we can manipulate with `<audio>` and `<video>` elements using JS.

Source code 6.4: Button element

```
<button id="play-pause" type="button" role="button"
title="Press to play or pause movie">
  Play / Pause
</button>
```

JavaScript seeks a competent element by its ID. We add also WAI-ARIA `role` and `title` with description of function to HTML code.

Source code 6.5: Example Play/Pause button

```
// Play or Pause button

window.addEventListener('DOMContentLoaded', function () {
  'use strict';
  var playPause = document.getElementById('play-pause'),
      video = document.getElementById('sintel_video');
  playPause.addEventListener('click', function () {
    if (video.paused) {
      video.play();
    } else {
      video.pause();
    }
  }, false);
}, false);
```

We add timer for showing time of song or video. It also indicates the running track.

Source code 6.6: HTML5/JS Timer

```
<!-- Timer is crated by Rob Crowther - http://www.boogdesign.com/ -->

<video id="sintel_video" width="686" height="292" preload
poster="img/poster.png"
ontimeupdate="document.getElementById('tracktime').
innerHTML = Math.floor(this.currentTime) + ' / ' +
Math.floor(this.duration);">
```

6. Practical Realization

```
<source src="video/sintel.webm" type="video/webm" />
<source src="video/sintel.mp4" type="video/mpeg" />
<p>Oops! Your browser doesn't support <code>video</code> element.
You can download <a href="video/sintel.mp4">Sintel movie</a>.
</p>
</video>

<p id="timer">Time: <span id="tracktime">0 / 0 </span> seconds</p>
```

6.1.1 Creation of Interface

Default browser buttons are bland so we style them by CSS. For making our buttons more attractive and usable we use iconic font Font Awesome (<http://fontawesome.io/>). We can choose the classic symbols for play, pause etc. Fonts have many advantages over bitmap images. For example: they can be changed by CSS, they are not in conflict with screen readers and they can be infinitely magnified.

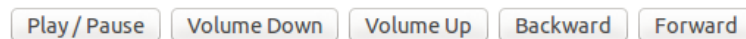


Figure 6.2: Default buttons (Author)

Source code 6.7: Font Awesome and our stylesheet applied to the button

```
<head>
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/font-awesome.min.css">
</head>

<button id="play1" role="button" title="Press to play or pause song">
<i class="fa fa-play fa-lg"></i> Play / <i class="fa fa-pause fa-lg"></i> Pause
</button>
```

Source code 6.8: CSS - Button element

```
button{
```

6. Practical Realization

```
border-style: solid;
border-width: 2px;
border-color: #000000;
border-radius: 5px;
padding: 3px;
margin: 2px;
}

button:focus, button:hover {
border-style: solid;
border-width: 2px;
border-color: #98002e;
border-radius: 5px;
color: #98002e;
}
```



Figure 6.3: Modified buttons (Author)

We would like create our website more attractive. We change design of player. We add rounded corners and white background for better contrast.

Source code 6.9: CSS3 - Movie Player style

```
#video_player {
width: 730px;
height: 430px;
border-radius: 15px;
border: 2px solid #000000;
position: relative;
left: 10px;
background: #ffffff;
}

#sintel_video {
color:#999;
margin: 20px;
border: 30px;
```

6. Practical Realization

```
}

#player_controls {
color:#999;
margin: 10px;
border:10px;
border-radius:15px;
float: inherit;
position: relative;
}
```

Source code 6.10: HTML5 - Movie Player

```
<div id="video_player">
  <video id="sintel_video" width="686" height="292" preload poster="img/poster.png">
    <source src="video/sintel.webm" type="video/webm" />
    <source src="video/sintel.mp4" type="video/mpeg" />
    <p>Oops! Your browser doesn't support <code>video</code>
    element. You can download <a href="video/sintel.mp4">Sintel movie</a>.
  </p>
</video>
</div>
```

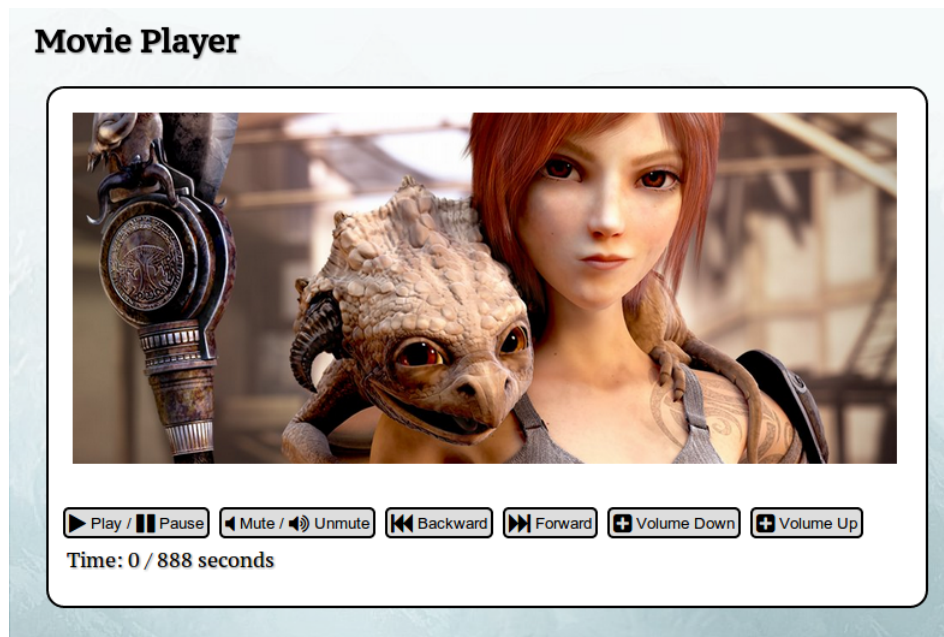



Figure 6.4: Movie Player - final version (Author)

6.2 Styling of pages

We would like to make logo of page same like original logo. For this reason we use the external fonts and we use image only for decoration. This procedure ensure that our logo will be accessible. We also apply these fonts for remaining content. Selected fonts are from Google Fonts service. CSS3 property `text-shadow` makes our text more stronger against the background.

Source code 6.11: HTML5 - link to Google Fonts service

```
<link href='http://fonts.googleapis.com/css?family=Eagle+Lake|PT+Serif+Caption'  
rel='stylesheet' type='text/css'>
```

Source code 6.12: CSS3 - applying fonts

```
body {  
font-family: 'PT Serif Caption', serif;  
color: #000000;  
font-size: 100%;  
text-shadow: 1px 1px 2px #999;  
}
```

6. Practical Realization

```
header h1 {
font-family: 'Eagle Lake', cursive;
letter-spacing: 12px;
}
```

For background we use official artwork image. We make background of our content slightly transparent by CSS property `background-color`.

Source code 6.13: CSS - Transparent background

```
#opacity{
width: 790px;
height: 100%;
background-color: rgba(256, 256, 256, 0.3);
}
```

6.3 Audio Player

We make audip player similarly as video player. We choose correct audio format and add attribute `preload`.

Source code 6.14: CSS3 - applying fonts

```
<audio id="1" preload ontimeupdate="document.getElementById
('tracktime').innerHTML = Math.floor(this.currentTime) + ' / '
+ Math.floor(this.duration);">
  <source src="audio/1-Snow_Fight.mp3" type="audio/mpeg">
  <source src="audio/1-Snow_Fight.ogg" type="audio/ogg">
  <p>Oops! Your browser doesn[U+FFFD]t support <code>audio</code> element.
  You can download <a href="audio/1-Snow_Fight.mp3">Snow Fight.mp3</a>.</p>
</audio>

<script>
  window.addEventListener('DOMContentLoaded', function () {
    'use strict';
    var playPause = document.getElementById('play1'),
        video = document.getElementById('1');
    playPause.addEventListener('click', function () {
      if (video.paused) {
        video.play();
```

6. Practical Realization

```
        } else {
            video.pause();
        }
    }, false);
}, false);
</script>

<button id="play1" role="button" title="Press to play or pause song">
<i class="fa fa-play fa-lg"></i> Play / <i class="fa fa-pause fa-lg"></i> Pause
</button>
<button onclick="document.getElementById('1').volume-= 0.2" role="button"
title="Press to Volume Down">
<i class="fa fa fa-minus-square fa-lg"></i> Volume Down
</button>
<button onclick="document.getElementById('1').volume+= 0.2" role="button"
title="Press to Volume Up">
<i class="fa fa fa-plus-square fa-lg"></i> Volume Up
</button>
<button onclick="document.getElementById('1').currentTime-= 20" role="button"
title="Skip backward 20 seconds">
<i class="fa fa-fast-backward fa-lg"></i> Backward
</button>
<button onclick="document.getElementById('1').currentTime+= 20" role="button"
title="Skip forward 20 seconds">
<i class="fa fa-fast-forward fa-lg"></i> Forward
</button>

<p>Time: <span id="tracktime" role="timer">0 / 0</span> seconds </p>
```

6.4 Testing of Accessibility

Sintel pages are finished. Now we should do a test of accessibility. Even a significant errors can be detected using simple tests. Firstly we will try to get to each control element or link on the page using the keyboard. As we can see see in the picture below our controls are accessible for keyboards. We also test color contrast between background and text, buttons and links. We choose Color Contrast Checker from WebAIM server.

We also try open our site in various mobile devices and web browsers. In most cases created site works. But we can find some problems with the type of multimedia file or some

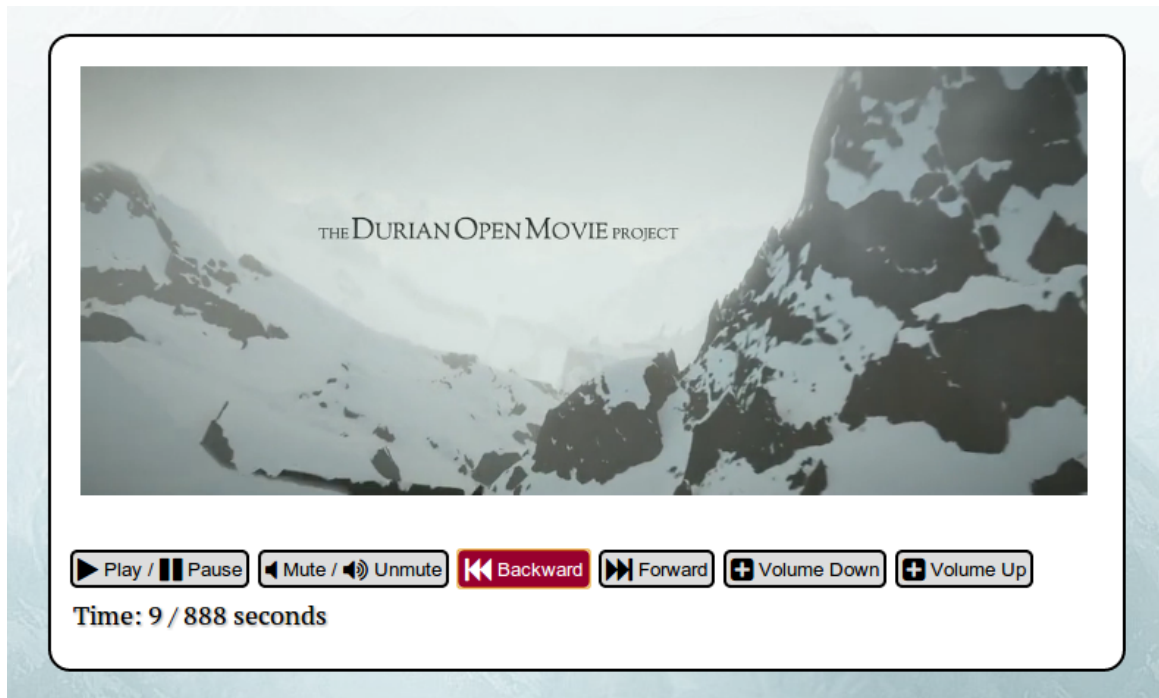


Figure 6.5: Play/Pause button is accessible (Author)

elements. For example Firefox doesn't support `<track>` element or some browsers require a specific type of codec.

6.4.1 Testing Tools

Firstly we test validity of code. We can use online validators like validator.w3.org or validator.webylon.info. Our test was successful and no errors were found.

Toolbar WAVE is a quick tool for basic accessibility test. It can shows missing alternative text for images, missing form labels, table structure, script elements etc. We could not find any errors.

There are other useful toolbars for browsers that can be used for test. For example Firebug, Web Develeopr or Fangs which simulates an output from a screen reader.

6. Practical Realization

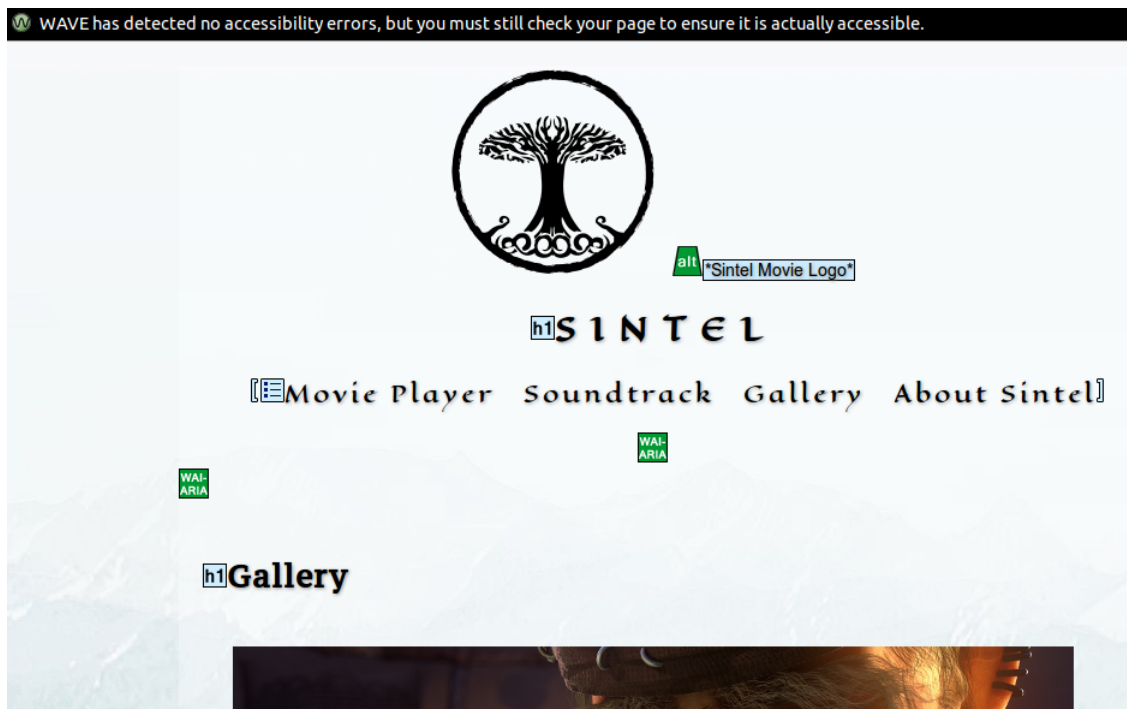


Figure 6.6: Output from Wave (Author)

6.4.1.1 User Testing

We can say that accessibility and usability tests with users is an effective way to detect errors and possible shortcomings which makes our website difficult to use.

We prepare some simple test scenarios for our potential testers. We sent tests and web page link by e-mail.

1. Example of testcase:

1. Go to "test URL".
2. Start the video.
3. Sound off.
4. Turn on sound.
5. Use the "forward" and "backward" buttons.

7. Conclusion

2. Example of testcase:

1. Go to "test URL".
2. Start the third song.
3. Mute sound to complete silence and then amplify it.

6.4.1.2 Test Results

Most of sent tests passed. Users had no problem to use controls or go to a specific page. They have some problem with number of controls in Sountrack page and Movie Player page. Based on results we reduced them from six to four in Soundtrack page and from seven to six in Movie Player page. We also added stronger **focus/hover** effect for better usability.

7 Conclusion

This work was focused on developing accessible Web sites using HTML5. In chapter 3 was described term web accessibility. We depicted who is a handicapped user and what is an assistive technology. There were also mentioned the methodologies of creating accessible web pages. We showed that disabled users are people with different needs and difficulties.

In the fourth chapter we paid attention on HTML5 and CSS3. We presented new structure tags like `nav`, `header` etc. which built clearer structure of content. We also describe other new tags and other innovations. A large space was devoted to HTML5 and its relationship to multimedia content. We displayed the way how to use new tags audio and video. At the end of this chapter CSS3 and WAI-ARIA was briefly introduced.

In chapter 5 we characterized some advantages and benefits of these technologies from the perspective of accessibility website content. We mentioned the independence of the third-party plugins, keyboard-accessible content or better structure of HTML5 document. CSS3 also brings some benefits like creating of decorative elements only using CSS technology.

Chapter 6 represented utilization of theoretical knowledge in web page formation. We described the various steps of making a site based on HTML5 and CSS3. We started with HTML document and sequentially added more functionality and content. Our resulting web is based only on HTML5, CSS3 and JavaScript.

Last part of this chapter was concentrated on testing of accessibility. We showed some methods how we can do this tests. We used some online applications and tools like WAVE or online controller validity of code. We also made user testing with impaired volunteers were involved into process of testing. Proposals and suggestions were used for repairing or improving accessibility.

7. Conclusion

Due to the results of testing we can say that created websites are accessible. We also found some problems. Some browsers have very strict requirements on the file format and codec so we need to have a big number of test devices. We also found that some of the browser have problems displaying subtitles. Some users had a problem with the number of control buttons and they would prefer a more classic appearance of control. However it is obvious that these technologies have great potential to replace third party technologies and we can assume that these technologies will be used more frequent.

Bibliography

- [1] O CONNOR, Joshue. **Pro HTML5 Accessibility**. 1st edition. New Yourk: Apress. 2012. 386 pages. ISBN13: 978-1-4302-4194-2
- [2] KEITH, Jeremy. **HTML5 for Web Designers**. 1st edition. New Yourk: A Book Apart. 2010. 96 pages. ISBN 978-0-9844425-0-8
- [3] **CSS3 Fonts** [online]. [cited 2013-10-10]
<http://www.w3schools.com/css/css3_fonts.asp>
- [4] HOGAN, Brian P. **HTML5 and CSS3 (2nd edition): Level Up with Today's Web Technologies**. 2nd edition. The Pragmatic Programmers 2013. 300 pages. ISBN: 978-1-93778-559-8
- [5] **HTML <meter> Tag** [online]. [cited 2013-10-11]
<http://www.w3schools.com/tags/tag_meter.asp>
- [6] **HTML 5 <ruby> Tag** [online]. [cited 2013-11-11]
<http://www.quackit.com/html_5/tags/html_ruby_tag.cfm>
- [7] THATCHER, Jim. **Web Accessibility - Section 508** [online]. [cited 2013-9-10]
<<http://jimthatcher.com/webcourse1.htm>>
- [8] MURPHY, Christopher. CLARK, Richard. STUDHOLME, Oli. MANIAN, Divya **Pro HTML5 and CSS3 Design Patterns**. 1st edition. New Yourk: Apress. 2012. 624 pages. SBN13: 978-1-4302-2874-5

- [9] BOWERS, Michael. SYNODINOS, Dionysios. SUMNER, Victor. **Beginning HTML5 and CSS3 The Web Evolved**. 1st edition. New Yourk: Apress. 2012. 532 pages. ISBN13: 978-1-4302-3780-8
- [10] CEDERHOLM, Dan. **CSS3 for Web Designers**. 1st edition. New Yourk: A Book Apart. 2010. 133 pages. ISBN 978-0-9844425-2-2
- [11] GASSTON, Peter **Multi-Device Web Development with HTML5, CSS3, and JavaScript**. 1st edition. San Francisco: No Starch Press. 2013. 264 pages. ISBN: 978-1-59327-487-0
- [12] ŠPINAR, David **Tvoříme přístupné webové stránky**. 1st edition. Brno: Zoner Press, 2004. 306 pages. ISBN 80-86815-11-0
- [13] PAVLÍČEK, Radek. **Přístupnost HTML5: Textové alternativy obrázků (2/2) – praxe** [online]. 2012. [cited 2013-11-20]
<<http://www.zdrojak.cz/clanky/pristupnost-html5-textove-alternativy-obrazku-2-2-praxe/>>
- [14] MALÝ, Martin. **Vyhněte se nejobvyklejším chybám v HTML5** [online]. 2011. [cited 2013-11-21]
<<http://www.zdrojak.cz/clanky/vyhnete-se-nejobvyklejsim-chybam-v-html5>>
- [15] PAVLÍČEK, Radek. **Přístupnost RIA – strukturování dokumentu a přístupnost z klávesnice** [online]. 2009. [cit. 2013-09-29]
<<http://www.zdrojak.cz/clanky/pristupnost-ria-strukturovani-dokumentu-a-pristupnost-z-klavesnice>>
- [16] KEITH, Jeremy. **Responsive Web Design**. 1st edition. New Yourk: A Book Apart. 2010. 157 pages. ISBN 978-0-9844425-7-7

Bibliography

- [17] THOMSON, Terrill. **Creating Your Own Accessible HTML5 Media Player** [online]. 2010. [cit. 2013-11-08]
<<http://terrillthompson.com/blog/32>>
- [18] THOMSON, Terrill. **Using HTML5 audio and video** [online]. [cit. 2013-11-10]
<https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Using_HTML5_audio_and_video>
- [19] LEADBETTER, Tom. **Video Subtitling and WebVTT** [online]. 2011. [cit. 2013-11-18]
<<http://html5doctor.com/video-subtitling-and-webvtt/>>
- [20] THOMSON, Terrill. **Comparison of Browsers on HTML5 Video Accessibility** [online]. 2013. [cit. 2013-11-08]
<http://terrillthompson.com/blog/366>
- [21] **HTML5 Video Events and API** [online]. 2013 [cit. 2013-11-17]
<http://www.w3.org/2010/05/video/mediaevents.html>
- [22] BOAS, Mark. **HTML5 Audio — The State of Play** [online]. 2011. [cit. 2013-09-18]
<<http://html5doctor.com/html5-audio-the-state-of-play/>>
- [23] FAULKNER, Steve. **HTML5 Accessibility Chops: notes on using ARIA** [online]. 2012. [cit. 2013-10-28]
<<http://blog.paciellogroup.com/2012/06/html5-accessibility-chops-using-aria-notes/>>
- [24] DUTTON, Sam. **Getting started with the HTML5 track element** [online]. 2013. [cit. 2013-11-22]
<http://www.html5rocks.com/en/tutorials/track/basics/?redirect_from_locale=cs>

Bibliography

- [25] **HTML <video> Tag** [online]. [cited 2013-11-11]
<http://www.w3schools.com/tags/tag_video.asp>
- [26] **HTML5 New Elements** [online]. [cited 2013-11-15]
<http://www.w3schools.com/html/html5_new_elements.asp>
- [27] **HTML <track> Tag** [online]. [cited 2013-10-17]
<http://www.w3schools.com/tags/tag_track.asp>
- [28] **Screen Reader User Survey 4 Results** [online]. 2012. [cited 2013-10-27]
<<http://webaim.org/projects/screenreadersurvey4/>>
- [29] VEEN, Greg. **Introduction to web typography and @font-face** [online]. 2011 [cit. 2013-11-18]
<<http://www.adobe.com/devnet/html5/articles/web-typography-and-css-font-face.html>>
- [30] LEWIS, Emily. **Native Audio with HTML5** [online]. [cit. 2013-11-09]
<<http://msdn.microsoft.com/en-us/hh550090.aspx>>
- [31] GOULD, David. **HTML5 and SEO: New Strategies for Optimizing Code** [online]. 2011. [cit. 2013-11-19]
<<http://www.verticalmeasures.com/search-optimization/html5-and-seo-new-strategies-for-optimizing-code/>>
- [32] TAHIRA. **CREATING RESPONSIVE MENU USING HTML5 AND CSS WITH MEDIA QUERY** [online].2013 [cit. 2013-10-20]
<<http://wemblo.com/blog/creating-responsive-menu-using-html5-and-css-with-media-query/>>
- [33] **Braille Displays - HumanWare US** [online]. [cit. 2013-11-09]
<http://www.humanware.com/en-new_zealand/products/blindness/braille_displays/_details/id_229/brailiant_bi_32_new_generation.html>

Bibliography

- [34] **Preventing Eye Macular Degeneration Steps** [online]. [cit. 2013-11-01]
<<http://www.howtoimproveeyesightwithouturgery.com/preventing-eye-macular-degeneration-steps/>>>
- [35] PFEIFFER, Silvia. **The Definitive Guide to HTML5 Video**. 1st edition. New York: Apress. 2010. 337 pages. ISBN-13: 978-1-4302-3091-2

List of Figures

3.1	Garden seen by someone with macular degeneration	6
3.2	Braille display	8
4.1	Logos of HTML5 and CSS3	10
4.2	Example of possible structure of web page with new HTML5 tags	12
4.3	Audio formats and browsers support	14
4.4	Example of HTML5 audioplayer (Mozilla Firefox 23)	15
4.5	Video Formats and Browsers Support	16
4.6	Example of HTML5 videoplayer (Mozilla Firefox 23)	17
4.7	Videoplayer with arabian subtitles (Chromium30)	18
4.8	Simple button created by older version of CSS	19
4.9	Button with gradient	19
4.10	Button with rounded corners	20
4.11	The major web browsers support different formats of web font	20
4.12	Final version of button	21
4.13	Respondents with JavaScript Enabled	22
6.1	Videoplayer in Mozila Firefox 25	30
6.2	Default buttons	32
6.3	Modified buttons	33
6.4	Movie Player - final version	34
6.5	Play/Pause button is accessible	38
6.6	Output from Wave	38
C.1	Screenshot of Movie Player page	III

List of Figures

C.2 Screenshot of Soundtrack Page	IV
C.3 Screenshot of Gallery Page	V
C.4 Screenshot of About Sintel Page	VI

List of Source Codes

4.1	HTML doctypes	11
4.2	Link to CSS	11
4.3	Element audio	15
4.4	Element video	16
4.5	Element canvas	18
4.6	Defining of a linear gradient	19
4.7	Defining of rounded corners	20
4.8	Definition and use custom font	21
4.9	WAI-ARIA roles	23
6.1	Basic html document	27
6.2	HTML document with WAI-ARIA-roles	28
6.3	Video player with defined attributes	29
6.4	Button element	31
6.5	Example Play/Pause button	31
6.6	HTML5/JS Timer	31
6.7	Font Awesome and out stylesheet applied to the button	32
6.8	CSS - Button element	32
6.9	CSS3 - Movie Player style	33
6.10	HTML5 - Movie Player	34
6.11	HTML5 - link to Google Fonts service	35
6.12	CSS3 - applying fonts	35
6.13	CSS - Transparent background	35

List of Source Codes

6.14 CSS3 - applying fonts 36

Content of DVD - Appendix A

```
/sintel
├── about.html
├── audio
│   ├── 1-Snow_Fight.mp3
│   ├── 1-Snow_Fight.ogg
│   ├── 2-Finding_Scales-Chicken_Run.mp3
│   ├── 2-Finding_Scales-Chicken_Run.ogg
│   ├── 3-The_Ziggurat.mp3
│   ├── 3-The_Ziggurat.ogg
│   ├── 4-Expedition.mp3
│   ├── 4-Expedition.ogg
│   ├── 5-Dragon_Blood_Tree.mp3
│   ├── 5-Dragon_Blood_Tree.ogg
│   ├── 6-Cave_Fight-Lament.mp3
│   ├── 6-Cave_Fight-Lament.ogg
│   ├── 7-I_Move_On-Sintels_Song.mp3
│   ├── 7-I_Move_On-Sintels_Song.ogg
│   ├── 8-Circling_Dragons.mp3
│   ├── 8-Circling_Dragons.ogg
│   ├── 9_Trailer_Music.mp3
│   └── 9_Trailer_Music.ogg
├── css
│   ├── font-awesome.min.css
│   └── style.css
├── fonts
│   ├── FontAwesome.otf
│   ├── fontawesome-webfont.eot
│   ├── fontawesome-webfont.svg
│   ├── fontawesome-webfont.ttf
│   └── fontawesome-webfont.woff
├── gallery.html
├── img
│   ├── background.png
│   ├── logo_sintel.png
│   ├── poster.png
│   ├── screen_sin1.png
│   ├── screen_sin2.png
│   ├── screen_sin3.png
│   └── screen_sin4.png
├── index.html
├── js
│   └── video_player.js
├── soundtrack.html
├── video
│   ├── sintel.mp4
│   └── sintel.webm
```

Acronyms - Appendix B

API - Application Programming Interface

CSS3 - Cascading Style Sheets 3

JS - JavaScript

HTML5 - HyperText Markup Language 5

UX - User Experience

WAI-ARIA - Web Accessibility Initiative - Accessible Rich Internet Applications

WCAG 2.0 - Web Content Accessibility Guidelines 2.0

W3C - World Wide Web Consortium

Screenshots - Appendix C

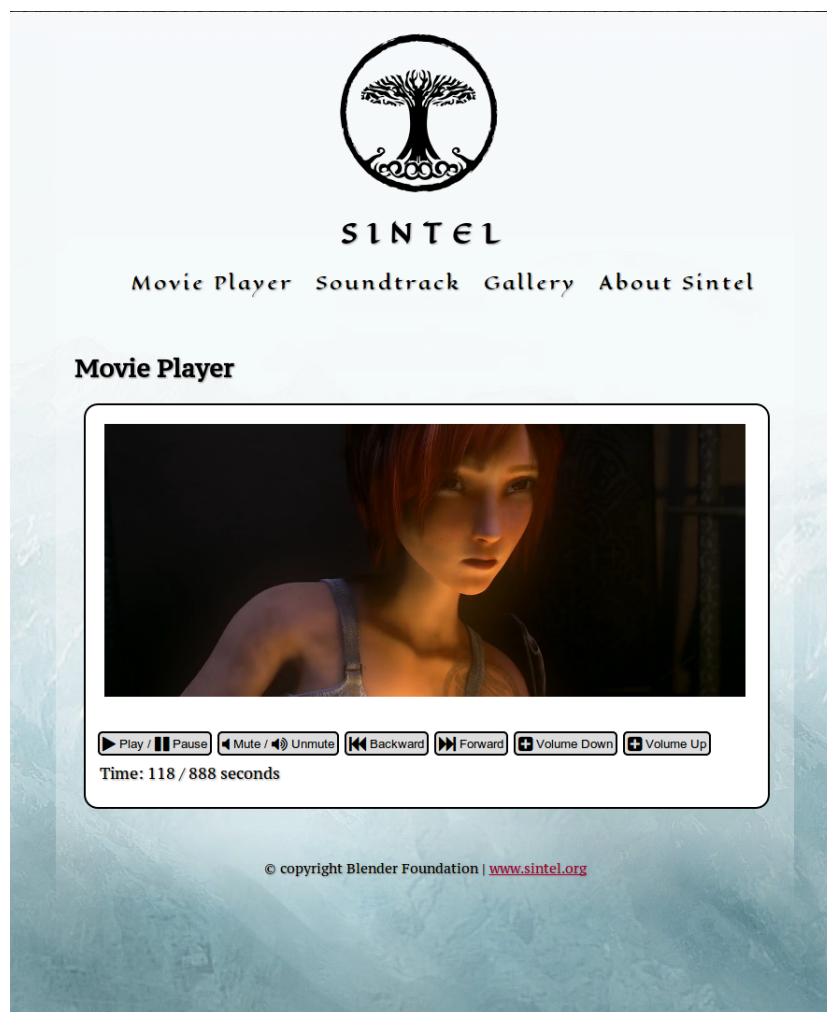


Figure C.1: Screenshot of Movie Player page

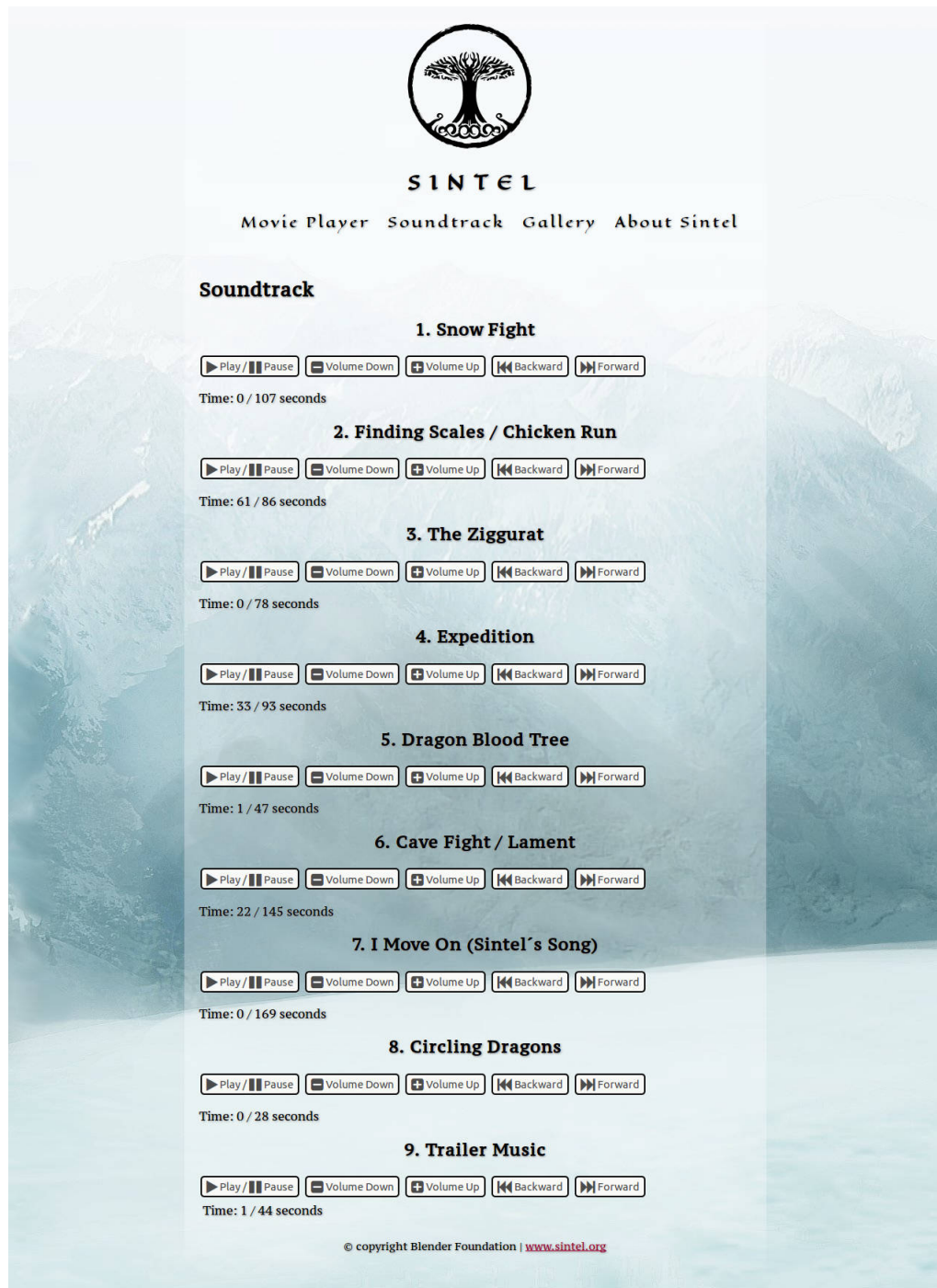


Figure C.2: Screenshot of Soundtrack Page

Screenshots - Appendix C

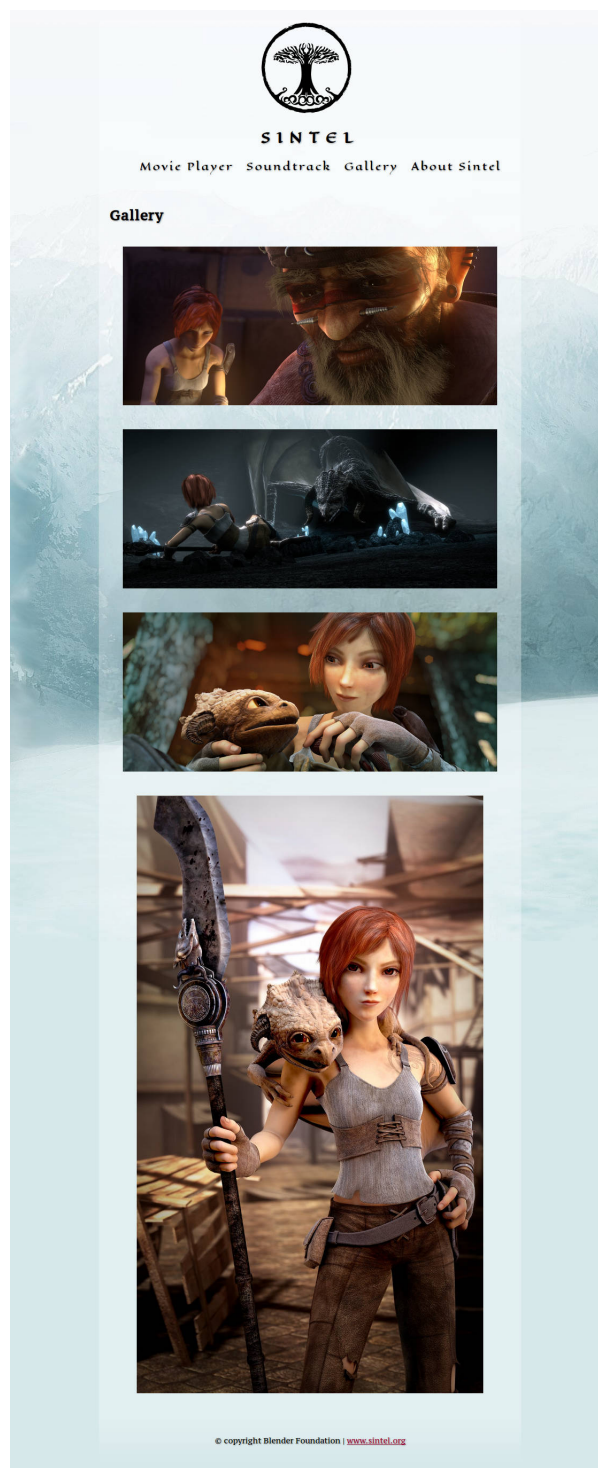


Figure C.3: Screenshot of Gallery page

Screenshots - Appendix C

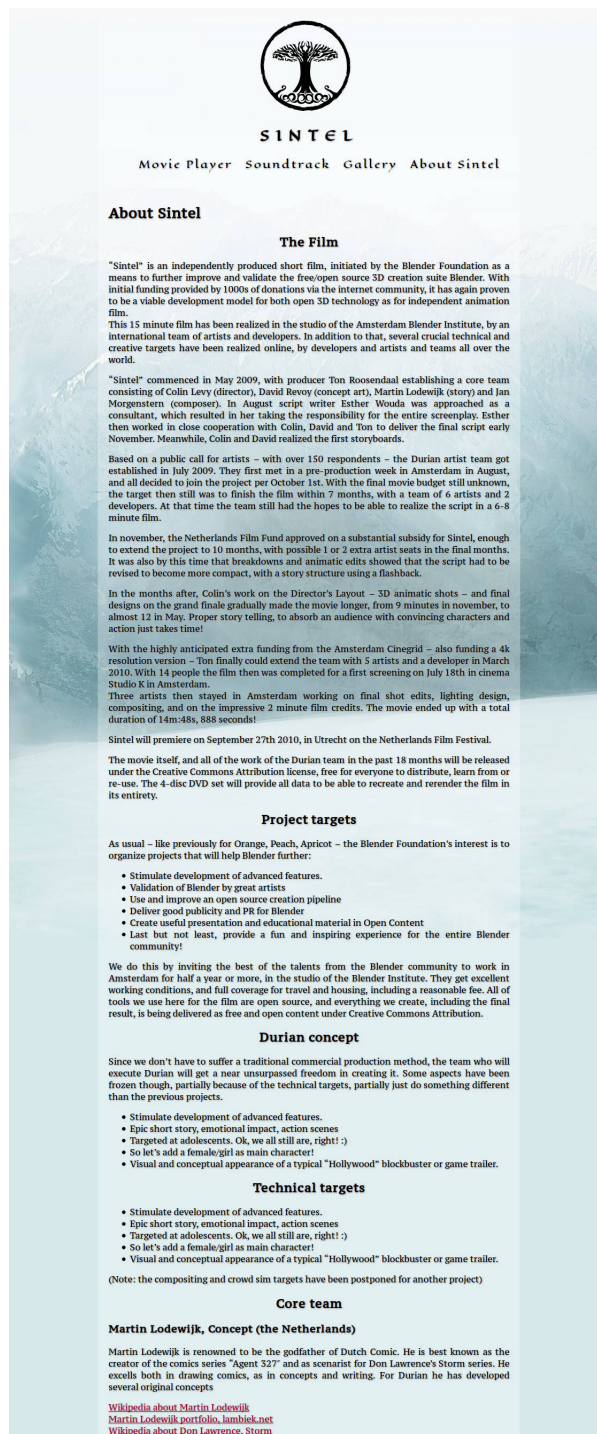
The image is a screenshot of a web page titled "About Sintel". At the top center is a circular logo featuring a tree with roots and branches, with the word "SINTEL" written below it. Below the logo, there are navigation links: "Movie Player", "Soundtrack", "Gallery", and "About Sintel". The main heading is "About Sintel". Underneath, there is a sub-heading "The Film". The text describes the film's origin, mentioning the Blender Foundation and the Amsterdam Blender Institute. It details the production process, including the involvement of various artists and the challenges faced. The text is organized into several paragraphs, each starting with a bolded first letter. There are also several bullet points under the "Project targets" and "Durian concept" sections. At the bottom, there is a "Core team" section mentioning Martin Lodewijk and Don Lawrence Storm, with links to their Wikipedia pages and portfolios. The background of the page is a light blue, abstract, textured image.

Figure C.4: Screenshot of About Sintel page