

Animation of Objects on the Website by Application of CSS3 Language

Vladimir Simovic, Matija Varga, Robert Svetlacic

Abstract—Scientific work analytically explores and demonstrates techniques that can animate objects and geometric characters using CSS3 language by applying proper formatting and positioning of elements. This paper presents examples of optimum application of the CSS3 descriptive language when generating general web animations (e.g., billiards and movement of geometric characters, etc.). The paper presents analytically, the optimal development and animation design with the frames within which the animated objects are. The originally developed content is based on the upgrading of existing CSS3 descriptive language animations with more complex syntax and project-oriented work. The purpose of the developed animations is to provide an overview of the interactive features of CSS3 descriptive language design for computer games and the animation of important analytical data based on the web view. It has been analytically demonstrated that CSS3 as a descriptive language allows inserting of various multimedia elements into websites for public and internal sites.

Keywords—Animation recording, web page graphics, HTML5 forms, Cascading Style Sheets 3 - CSS3, man-computer interaction, KML animation presenting format, GML, Google Earth Professional.

I. INTRODUCTION

ANIMATION is the term used in graphics, multimedia and other fields and features a wide range of uses [3]. In order for a spectator to be impressed, the visual effects created by animations are desirable. In most cases, after having seen the visualized content, an average user wonders what the ways are in which the web page animations have been rendered, i.e. how the visual effects are produced or, shortly, how the virtual reality is attained. Computer animation (a large and an important field of computer graphics) started to develop simultaneously with computer graphics. Although animation (as the procedure of creating the illusion of movement) was known a long time before modern computers came into being, it is only with these computers that it has started flourishing in a way. Nowadays, the animations in web pages can be rendered by means of CSS3 descriptive web language that can be used within a separate file or written/implemented within HTML5 descriptive language used for structuring and presenting web page content [1].

Vladimir Simovic is with the Rectorate, University North, Koprivnica, Trg dr. Zarka Dolinara 1, HR-48000 Koprivnica, Croatia, EU (phone: +385-98-262271, e-mail: vladimir.simovic@unin.hr).

Matija Varga is Ph.D. lecturer at the University of Zagreb, Faculty of Teacher Education, Savska 77, 10000 Zagreb, Croatia (e-mail: maavarga@gmail.com).

Robert Svetlacic is the Ph.D. Candidate at the Faculty of Tourism and Hospitality Management, University of Rijeka, Primorska 42, 51410 Opatija, Croatia (e-mail: rsvetlacic@gmail.com).

II. RESEARCH APPROACH

The awareness of animation as an important factor of the contemporary multimedia technologies is in turn important for web page visitors (in: tourism, business, gaming, etc.) [3]. Animation can attract a reader and this is why the research dealt with the detection and the importance of the role of animation in contemporary media, i.e. in the largest and the most powerful media of today, the Internet, as well as with the analysis of CSS3 descriptive language used for rendering animations. Google Earth Professional as a tool is based on various web technologies that include better resolution of photographs, the possibility of recording animations, map printing options and a series of advanced animation options. The research techniques that were used to make the respective conclusions of the paper were as follows: observing the web page designing CSS3 descriptive language, the analysis of CSS3 descriptive language, the analysis of possible modifications of CSS3 descriptive language and the influence of these modifications on the changes in web page content.

III. DESCRIPTIVE LANGUAGES FOR WEB ANIMATIONS

The paper mostly analyses CSS3 descriptive language because Google Earth Professional offers the options for recording animations [1]. Google Earth Professional as a web-based tool includes better resolution of photographs, the possibility of recording animations, map printing options and a series of advanced options. Google Earth is structured by Keyhole descriptive language used in geographic information systems; it is a Keyhole company product and its basis is structured in the same manner as the basis of Google Maps. The full name behind the KML abbreviations is Keyhole Markup Language. KML is the format of geographic data records and it is used for presenting 2D and 3D maps, locations, polygons, 3D models and for presenting the animation of relations among the data. KML provides for a direct connection to GPS (Global Positioning System) [2], [4]. KML format is mentioned in the paper because it encourages a wider and an increased interoperability (compatibility) and because it is complementary with the majority of the existing standards (for example, with OGC, Open Geospatial Consortium that originated as far back as in 1994 and for the purpose of defining the open systems of geographic information) [4]. OGC was inspired by XML (Extensible Markup Language) and it resulted in the creation of GML (Geography Markup Language). XML formats GML and KML belong to the WFS (Web Feature Service) domain of OGC web services [5]. KML and GML were developed from XML descriptive language used for creating web databases.

Moreover, by means of GML and KML descriptive languages and some supporting languages, it is possible to animate the most convenient route from one point to another (from the standpoint to a destination) within a map on the web. The destinations may be hospitals, police stations, cultural monuments or the best beaches etc. and it is frequently used in (online) tourist information systems that are in turn parts of geographic information systems used for finding the most convenient and/or the most economic route (Fig. 6) [4].

IV. CSS3 ANIMATION EXAMPLES

It is a common knowledge that CSS3 (Cascading Style Sheets 3) was the latest descriptive language that appear after the other two languages used in the DHTML (Dynamic Hypertext Markup Language) environment. CSS3 descriptive language, as the latest version of CSS (or, rather, its latest generation), has brought a galore of good features that have significantly facilitated the making of web pages and animations on web pages [1]. CSS3 consists of a set of rules and styles that define how particular HTML5 elements will be presented. CSS3 manages both the HTML5 standard elements and elements and all marks used within SVG (Scalable Vector Graphics). SVG is used for defining the web graphics. The <svg> tag as an element is a “container” for SVG graphics [5]. The advantages provided by CSS3 are as follows: considerably higher possibilities of determining the HTML5 properties, the possibility of separating the content from a presentation, and finally, the looks of a complete website can be changed with deployment of only one CSS3 file. CSS3 descriptive language can be divided in modules, selectors, box models, backgrounds and borders, textual effects, 2D/3D transformations, animations, the look with multiple columns and user interface.

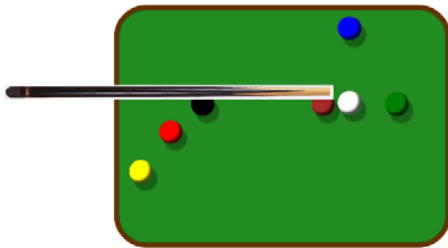


Fig. 1 Animation “Billiards” in CSS3 descriptive language

Fig. 1 shows the animation in CSS3 descriptive language, “Billiards”. The balls move in accordance with the trajectory defined in CSS3 descriptive language (down) within a rounded-corners brown frame shaped by border-radius. Here is a source code example (developed by the authors):

```
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.0
Transitional//>
<html><head>
<meta charset="utf-8">
<link rel="stylesheet" href="ball.css" type="text/css">
<style>
.aaa{
```

```
display: block;
width: 50px;
height: 50px;
border-radius: 70px;
background-color: yellow;
box-shadow: inset -6px -6px 5px rgba(0,0,0,.7), 10px 15px 3px
rgba(0,0,0,.3) ;
position: absolute;
-webkit-animation: moveX 3.3s linear 0s infinite alternate, moveY
2.3s linear 0s infinite alternate;
-moz-animation: moveX 3.3s linear 0s infinite alternate, moveY
2.3s linear 0s infinite alternate;
-o-animation: moveX 3.3s linear 0s infinite alternate, moveY 2.3s
linear 0s infinite alternate;
animation: moveX 3.3s linear 0s infinite alternate, moveY 2.3s
linear 0s infinite alternate; }
@-webkit-keyframesmoveX {
from {left: 0; } to {left: 640px; }}
@-moz-keyframesmoveX {
from {left: 0; } to {left: 640px; }}
@-o-keyframesmoveX {
from {left: 0; } to {left: 640px; }}
@keyframesmoveX {
from {left: 0; } to {left: 640px; }}
@-webkit-keyframesmoveY {
from {top: 0; } to {top: 440px; }}
@-moz-keyframesmoveY {
from {top: 0; } to {top: 440px; }}
@-o-keyframesmoveY {
from {top: 0; } to {top: 440px; }}
@keyframesmoveY {
from {top: 0; } to {top: 440px; }}
.container {
background-color: #228B22;
margin: 0 auto;
width: 700px;
height: 500px;
position: relative;
box-shadow: inset 0 0 3px #000;
border-radius: 70px;
border: 10px solid #663300;
overflow: hidden;}
.bbb{
display: block;
width: 50px;
height: 50px;
border-radius: 70px;
background-color: blue;
box-shadow: inset -6px -6px 5px rgba(0,0,0,.7), 10px 15px 3px
rgba(0,0,0,.3) ;
position: absolute;
-webkit-animation: moveX 2.3s linear 0s infinite alternate, moveY
3.3s linear 0s infinite alternate;
-moz-animation: moveX 2.3s linear 0s infinite alternate, moveY
3.3s linear 0s infinite alternate;
-o-animation: moveX 2.3s linear 0s infinite alternate, moveY 3.3s
linear 0s infinite alternate;
animation: moveX 2.3s linear 0s infinite alternate, moveY 3.3s
linear 0s infinite alternate; }
@-webkit-keyframesmoveX {
from {left: 0; } to {left: 640px; }}
@-moz-keyframesmoveX {
from {left: 0; } to {left: 640px; }}
@-o-keyframesmoveX {
from {left: 0; } to {left: 640px; }}
@keyframesmoveX {
```

```

from {left: 0; } to {left: 640px; }}
@-webkit-keyframesmoveY {
from {top: 0; } to {top: 440px; }}
@-moz-keyframesmoveY {
from {top: 0; } to {top: 440px; }}
@-o-keyframesmoveY {
from {top: 0; } to {top: 440px; }}
@keyframesmoveY {
from {top: 0; } to {top: 440px; }}
.ccc{
display: block;
width: 50px;
height: 50px;
border-radius: 70px;
background-color: red;
box-shadow: inset -6px -6px 5px rgba(0,0,0,.7), 10px 15px 3px
rgba(0,0,0,.3);
position: absolute;
-webkit-animation: moveX 5.3s linear 0s infinite alternate, moveY
4.3s linear 0s infinite alternate;
-moz-animation: moveX 5.3s linear 0s infinite alternate, moveY
4.3s linear 0s infinite alternate;
-o-animation: moveX 5.3s linear 0s infinite alternate, moveY 4.3s
linear 0s infinite alternate;
animation: moveX 5.3s linear 0s infinite alternate, moveY 4.3s
linear 0s infinite alternate; }
@-webkit-keyframesmoveX {
from {left: 0; } to {left: 640px; }}
@-moz-keyframesmoveX {
from {left: 0; } to {left: 640px; }}
@-o-keyframesmoveX {
from {left: 0; } to {left: 640px; }}
@keyframesmoveX {
from {left: 0; } to {left: 640px; }}
@-webkit-keyframesmoveY {
from {top: 0; } to {top: 440px; }}
@-moz-keyframesmoveY {
from {top: 0; } to {top: 440px; }}
@-o-keyframesmoveY {
from {top: 0; } to {top: 440px; }}
@keyframesmoveY {
from {top: 0; } to {top: 440px; }}
.ddd{
display: block;
width: 50px;
height: 50px;
border-radius: 70px;
background-color: white;
box-shadow: inset -6px -6px 5px rgba(0,0,0,.7), 10px 15px 3px
rgba(0,0,0,.3);
position: absolute;
-webkit-animation: moveX 2.3s linear 0s infinite alternate, moveY
1.8s linear 0s infinite alternate;
-moz-animation: moveX 2.3s linear 0s infinite alternate, moveY
1.8s linear 0s infinite alternate;
-o-animation: moveX 2.3s linear 0s infinite alternate, moveY 1.8s
linear 0s infinite alternate;
animation: moveX 2.3s linear 0s infinite alternate, moveY 1.8s
linear 0s infinite alternate; }
@-webkit-keyframesmoveX {
from {left: 0px; } to {left: 640px; }}
@-moz-keyframesmoveX {
from {left: 0px; } to {left: 640px; }}
@-o-keyframesmoveX {
from {left: 0px; } to {left: 640px; }}
@keyframesmoveX {

```

```

from {left: 0px; } to {left: 640px; }}
@-webkit-keyframesmoveY {
from {top: 0px; } to {top: 440px; }}
@-moz-keyframesmoveY {
from {top: 0px; } to {top: 440px; }}
@-o-keyframesmoveY {
from {top: 0px; } to {top: 440px; }}
@keyframesmoveY {
from {top: 0px; } to {top: 440px; }}
.eee{
display: block;
width: 50px;
height: 50px;
border-radius: 70px;
background-color: black;
box-shadow: inset -6px -6px 5px rgba(0,0,0,.7), 10px 15px 3px
rgba(0,0,0,.3);
position: absolute;
-webkit-animation: moveX 2s linear 0s infinite alternate, moveY
1.2s linear 0s infinite alternate;
-moz-animation: moveX 2s linear 0s infinite alternate, moveY 1.2s
linear 0s infinite alternate;
-o-animation: moveX 2s linear 0s infinite alternate, moveY 1.2s
linear 0s infinite alternate;
animation: moveX 2s linear 0s infinite alternate, moveY 1.2s
linear 0s infinite alternate; }
@-webkit-keyframesmoveX {
from {left: 0px; } to {left: 640px; }}
@-moz-keyframesmoveX {
from {left: 0px; } to {left: 640px; }}
@-o-keyframesmoveX {
from {left: 0px; } to {left: 640px; }}
@keyframesmoveX {
from {left: 0px; } to {left: 640px; }}
@-webkit-keyframesmoveY {
from {top: 0px; } to {top: 440px; }}
@-moz-keyframesmoveY {
from {top: 0px; } to {top: 440px; }}
@-o-keyframesmoveY {
from {top: 0px; } to {top: 440px; }}
@keyframesmoveY {
from {top: 0px; } to {top: 440px; }}
</style>
</head>
<body>
<div id="header"></div>
<div class="main">
<br/>
<div class="container">
<div class="aaa">
</div>
<div class="bbb">
</div>
<div class="ccc">
</div>
<div class="ddd">
</div>
<div class="eee">
</div>
<div class="fff">
</div>
<div class="iii">
</div>
</div>
</body>

```

</html>

CSS3 starts from the <style> tag and ends with the </style> tag within HTML5 descriptive language. By means of CSS3, the movement of objects in various directions has been provided for and the movements of the objects take place within some self-shaping round-cornered frames that can in turn be of various colours. The roundness of corners is attained via the CSS3 feature of border-radius (for example, border-radius: 70px). The displaying of the elements is attained via the features like inline, block, none. In our example, the block feature was used (for example: display: block). The defining of colours of objects is attained via the background-colour feature: yellow. Further, the balls have a shadow within them thanks to the box-shadow feature: inset. The width of the object is attained via the width feature set at 50px, whereas the height of the object is attained via the height feature set at 50px. The position is in our example set at absolute. The absolute position changes the position of the objects in relation with the given sequence of appearance in web pages. An element appears in relation with the upper left position of the closest parental element. The parental element is the closest superior element whose position feature is set at any value except static. The value of the position static feature ensures the given sequence of appearance of elements in a web page. If such an element does not exist, the position is defined in relation with the upper left corner of the browser. Other positions that can be defined are relative and fixed [2]. The fixed position fixes the element and it will not move when the content of the object is moved. The relative position places the element normally. The position of the div element is relative. In the end of the animation-displaying descriptive language, there are divs and divs classes. The <div> tag as an element is a block structural element. Within the <div> element, there may be almost all html elements embedded. The three basic features of the <div> elements are style (used to define the CSS3 features), class (defines the name of the class for access to a CSS or JavaScript element) and ID (the unique identifier for access to a CSS or JavaScript element) [2]. The command moveY 3.3s determines the speed of movements of elements within a frame.

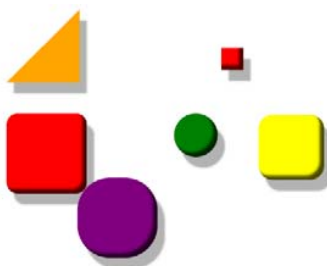


Fig. 2 Animation "Movements of geometric shapes" in CSS3 descriptive language

Fig. 2 shows animation in CSS3 descriptive language, "Movements of geometric shapes" in web sites and based on the descriptive language (that is itself partly described below).

```

*G:\Animacije\Animacija_mojet.html - Notepad++
Datoteka Uredi Traži Prilaz Format Jezik sintaksa Postavke Alati Makro Pločeni Priključni Prozor 7
Dokumenti
Animacija_mojet.html
1 <DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//>
2 <html><head>
3 <meta charset="utf-8">
4 <link rel="stylesheet" href="ball.css" type="text/css">
5 <style>
6 .kocka {
7   display: block;
8   width: 60px;
9   height: 60px;
10  border-radius: 70%;
11  background-color: green;
12  box-shadow: inset -6px 5px rgba(0,0,0,.7), 10px 15px 3px rgba(0,0,0,.3);
13  position: absolute;
14  -webkit-animation: moveX 2.7s linear 0s infinite alternate, moveY 2.7s linear 0s infinite alternate;
15  -ms-animation: moveX 2.7s linear 0s infinite alternate, moveY 2.7s linear 0s infinite alternate;
16  -o-animation: moveX 2.7s linear 0s infinite alternate, moveY 2.7s linear 0s infinite alternate;
17  animation: moveX 2.7s linear 0s infinite alternate, moveY 2.7s linear 0s infinite alternate;
18 }
19 .kda {
20   display: block;
21   width: 90px;
22   height: 90px;
23   border-radius: 20px;
24   background-color: yellow;
25   box-shadow: inset -6px -6px 5px rgba(0,0,0,.7), 10px 15px 3px rgba(0,0,0,.3);
26   position: absolute;
27   -webkit-animation: moveX 1.7s linear 0s infinite alternate, moveY 2.7s linear 0s infinite alternate;
28   -ms-animation: moveX 1.7s linear 0s infinite alternate, moveY 2.7s linear 0s infinite alternate;
29   -o-animation: moveX 1.7s linear 0s infinite alternate, moveY 2.7s linear 0s infinite alternate;
30   animation: moveX 1.7s linear 0s infinite alternate, moveY 2.7s linear 0s infinite alternate;
31 }
32 .lopta {
33   display: block;
34   width: 110px;
35   height: 110px;
36   border-radius: 15px;
37   background-color: red;
38   box-shadow: inset -6px 5px rgba(0,0,0,.7), 10px 15px 3px rgba(0,0,0,.3);
39   position: absolute;
40   -webkit-animation: moveX 3.7s linear 0s infinite alternate, moveY 2.7s linear 0s infinite alternate;
41   -ms-animation: moveX 3.7s linear 0s infinite alternate, moveY 2.7s linear 0s infinite alternate;
42   -o-animation: moveX 3.7s linear 0s infinite alternate, moveY 2.7s linear 0s infinite alternate;
43   animation: moveX 3.7s linear 0s infinite alternate, moveY 2.7s linear 0s infinite alternate;
44 }
45 .aaa {
46   display: block;
47   -o-animation: moveX 4.7s linear 0s infinite alternate, moveY 5.7s linear 0s infinite alternate;
48   animation: moveX 4.7s linear 0s infinite alternate, moveY 5.7s linear 0s infinite alternate;
49 }
50 .lopta {
51   display: block;
52   width: 0;
53   height: 0;
54   border-top-width: 100px;
55   border-top-style: solid;
56   border-top-color: transparent;
57   border-right-width: 100px;
58   border-right-style: solid;
59   border-right-color: orange;
60   box-shadow: inset -6px -6px 5px rgba(0,0,0,.7), 10px 15px 3px rgba(0,0,0,.3);
61   position: absolute;
62   -webkit-animation: moveX 3.7s linear 0s infinite alternate, moveY 5.7s linear 0s infinite alternate;
63   -ms-animation: moveX 3.7s linear 0s infinite alternate, moveY 5.7s linear 0s infinite alternate;
64   -o-animation: moveX 3.7s linear 0s infinite alternate, moveY 5.7s linear 0s infinite alternate;
65   animation: moveX 3.7s linear 0s infinite alternate, moveY 5.7s linear 0s infinite alternate;
66 }
67 @-webkit-keyframes moveX {
68   from (left: 0%; ) to (left: 90%; )
69 }
70 @-ms-keyframes moveX {
71   from (left: 0%; ) to (left: 90%; )
72 }
73 @-o-keyframes moveX {
74   from (left: 0%; ) to (left: 90%; )
75 }
76 @keyframes moveX {
77   from (left: 0%; ) to (left: 90%; )
78 }
79 @-webkit-keyframes moveY {
80   from (top: 0%; ) to (top: 90%; )
81 }
82 @-ms-keyframes moveY {
83   from (top: 0%; ) to (top: 90%; )
84 }
85 @-o-keyframes moveY {
86   from (top: 0%; ) to (top: 90%; )
87 }
88 @keyframes moveY {
89   from (top: 0%; ) to (top: 90%; )
90 }
91 </style>
92 </head><body>
93 <div class="lopta"></div>
94 <div>
95 <div class="kocka"></div>
96 <div class="kda"></div>
97 <div class="aaa"></div>
98 <div class="lopta"></div>
99 <div class="lopta"></div>
100 </body></html>

```

Fig. 3 Listing of codes required for the Fig. 2, i.e. animated objects (geometric shapes)

Fig. 3 presents codes required for the previous figure, i.e. animated objects (geometric shapes) and the movements of the geometric shapes in various directions and at various speeds. The animations rendered by CSS3 descriptive language can replace animated figures rendered in Flash or JavaScript in web pages. Animation consists of two components, namely the style describing the CSS3 animation and a series of *key frames* determining the initial and the final state of animation. The blocks of commands for CSS3 descriptive language (movement of geometric shapes) are as follows:

"Triangle"

```
.loptx {
    display: block;
    width: 0;
    height: 0;
    border-top-width: 100px;
    border-top-style: solid;
    border-top-color: transparent;
    border-right-width: 100px;
    border-right-style: solid;
    border-right-color: orange;
    box-shadow: inset -6px -6px 5px rgba(0,0,0,.7), 10px 15px 3px
    rgba(0,0,0,.3);
    position: absolute;
    -webkit-animation: moveX 3.7s linear 0s infinitealternate, moveY
    5.7s linear 0s infinitealternate;
    -moz-animation: moveX 3.7s linear 0s infinitealternate, moveY
    5.7s linear 0s infinitealternate;
    -o-animation: moveX 3.7s linear 0s infinitealternate, moveY 5.7s
    linear 0s infinitealternate;
    animation: moveX 3.7s linear 0s infinitealternate, moveY 5.7s
    linear 0s infinitealternate; }
```

“Ball” (circle)

```
.lopta {
    display: block;
    width: 60px;
    height: 60px;
    border-radius: 70%;
    background-color: green;
    box-shadow: inset -6px -6px 5px rgba(0,0,0,.7), 10px 15px 3px
    rgba(0,0,0,.3); }
```

```
position: absolute;
    -webkit-animation: moveX 2.7s linear 0s infinitealternate, moveY
    2.7s linear 0s infinitealternate;
    -moz-animation: moveX 2.7s linear 0s infinitealternate, moveY
    2.7s linear 0s infinitealternate;
    -o-animation: moveX 2.7s linear 0s infinitealternate, moveY 2.7s
    linear 0s infinitealternate;
    animation: moveX 2.7s linear 0s infinitealternate, moveY 2.7s
    linear 0s infinitealternate; }
```

„Square“ with rounded corners

```
.kda {
    display: block;
    width: 90px;
    height: 90px;
    border-radius: 20px;
    background-color: yellow;
    box-shadow: inset -6px -6px 5px rgba(0,0,0,.7), 10px 15px 3px
    rgba(0,0,0,.3); }
```

Fig. 4 shows the animation of *divs* (frame) and figure within the `<style></style>` tag. The frame is stretched and then shrunk again, which is defined within HTML (see figures above). Besides all what has been said, it is worth mentioning that, in order for the sequence of CSS3 animation to be created, an element is tagged with the animation feature and/or its sub features. That adjusts the duration and the cue of the implementation of a part of animation. Animation is applied to the element whose animation-name attribute value is the feature referring to valid `@keyframes` rule (see Figs. 3 and 4).

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style>
5 #myDIV {
6     height: 200px;
7     background-color: lightblue;
8     -webkit-animation: mymove 5s infinite; /* Chrome, Safari, Opera */
9     animation: mymove 5s infinite;
10 }
11
12 @-webkit-keyframes mymove {
13     50% {margin: 50px;}
14 }
15
16 /* Standard syntax */
17 @keyframes mymove {
18     50% {margin: 50px;}
19 }
20 </style>
21 </head>
22 <body>
23
24 <div id="myDIV">
25 <center><b>Tekst.</b></center>
26 
27 </div>
28
29 </body>
30 </html>
```

Fig. 4 Animation of *divs* (frame) and figure

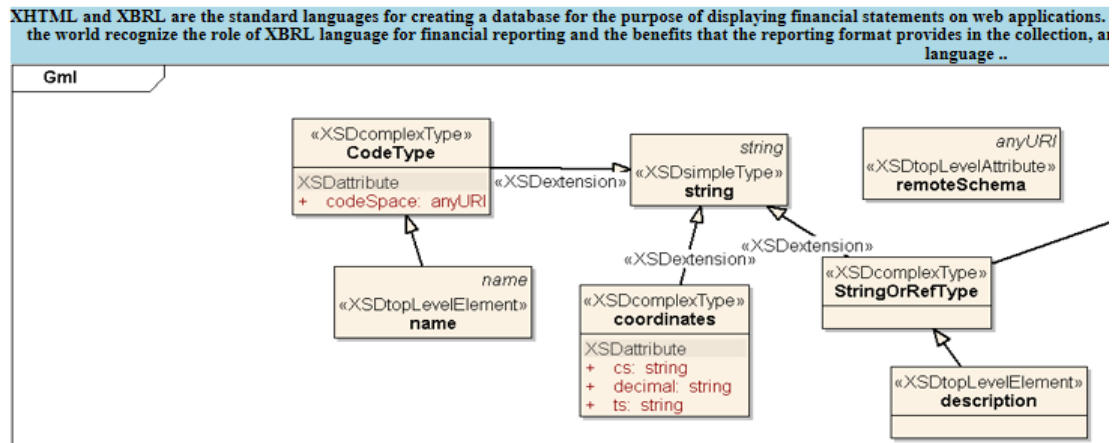


Fig. 5 Animation rendered with the descriptive language shown in Fig. 4

The animation finishes by utilizing the combination of animation-duration, animation-iteration-count and animation-fill-mode features. The sub features of the animation attribute that can be used are animation-delay, animation-direction, animation-duration, animation-iteration-count, animation-name, animation-play-state, animation-timing-function and animation-fill-mode. The @keyframes rule consists of the keyword, @keyframes, - the identifier of the rule that gives animation the name – and a set of rules in curly brackets that define the animation (Figs. 3 and 4) [1]. Fig. 5 shows the animation rendered with the structure of the descriptive language shown in Fig. 4. The frame grows larger and then smaller within a given time. The figure not only adjusts to the frame area but leaves it as well.



Fig. 6 An example of displaying the route from the standpoint to a destination in a web page

Fig. 6 gives an example of displaying the route from the standpoint to a destination in a web page. The standpoint is at the roundabout and the (animated) orange lines dynamically lead to the destination. On the basis of the animated map, two possible routes to the chosen destination are offered.

V. CONCLUSION

A comparatively new feature of CSS3 descriptive language is the transform attribute by means of which the elements can

be scaled, moved without using positioning and rotated [2]. Until deployment of CSS3, animations could be rendered by JavaScript (which is a script language whose syntax is similar to C language and movement is most frequently performed by client in the web browser), whereas it is possible now to render an animation impeccably in CSS3 within HTML5 code or in the form of a separate file. Moreover, for the purpose of rendering animations on web, CSS3 can be combined with other languages (the descriptive languages used for creating the databases) and web formats so that animations are as real as possible and as useful as possible accordingly.

The interest in viewing and using the optimized animation has been growing, especially because animation has increasingly been coming closer to the professional realm of a creative occupation whose performers have been, up to now, TV stations and various audiovisual arts that did not include the use of web technologies. As the web animations rendered in CSS3 descriptive language are to keep on developing, the paper has shown the possibility of animating individual objects within the appropriate rounded and not-rounded frames.

The users and developers of animations use animations for learning (especially if animations are easily accessed) because they agree that animations on web are important for improving knowledge. The use of animations may as well make the teaching process more attractive. Animation in the teaching process has a long-term perspective as well as in other fields (in simulations, namely). Animations in the field of education, ranging from primary schools to universities, can make students more interested in the content of learning or in recapitulations at the end of a class. Animation is an upgrade to the existing classic learning contents whereas the visual effects in the field of the technical sciences – computer science, that is – cause the most impressive (personal) experiences in spectators. Finally, the paper analytically explaining as many meanings as possible of some commands, elements and features within the utilized descriptive language used for rendering CSS3 animations. Next step of analytical scientific research is comparative study of complex animation

models with CSS3 solutions.

REFERENCES

- [1] CSS3 Animations. URL: http://www.zemris.fer.hr/predmeti/irg/Zavrzni/13_Komar/sadrzaj_css.html. (accessed 7/6/2017).
- [2] CSS Positioning. URL: <https://www.mojwebdizajn.net/skriptni-jezici/vodic/css/css-positioning.aspx>. (accessed 9/6/2017).
- [3] Fezile Özdamli, The experiences of teachers candidates in developing instructional multimedia materials in project based learning, Procedia – Social and Behavioral Sciences, Volume 15, 2011, Pages 3810-3820.
- [4] Geoinformation systems. Internet GIS. URL: https://www.fer.unizg.hr/_download/repository/GIS_06.pdf. (accessed 9/6/2017).
- [5] W3SCHOOLS. https://www.w3schools.com/cssref/pr_class_display.asp (accessed 8/6/2017).