

Vedran Harmadi¹
Marin Kaluža²
Dijana Liverić³

Stručni rad
UDK 004.928:004.73852

USPOREDBA TEHNOLOGIJA PRIKAZA ANIMACIJA U RAZLIČITIM WEB-PREGLEDNICIMA⁴

SAŽETAK

U radu se provodi analiza često korištenih tehnologija za prikaz animacija u web-preglednicima. Flash je jedna od tehnologija koja je značajno pridonijela prikazu i mogućnostima izrade web-animacija, te se u počecima najviše koristila za njihovu izradu. Kroz godine pojavile su se mnoge nove tehnologije, od kojih se posebice može izdvojiti HTML5 canvas, koja omogućuje prikaz animacija bez uporabe dodataka, te mnoge nove mogućnosti njihove izrade. Ovim radom želi se pokazati koja od promatranih tehnologija ostvaruje najbolje performanse prilikom izvođenja animacija u različitim web-preglednicima. U radu se ispituju performanse izvođenja Flash, HTML5 canvas, SVG i JavaScript animacija unutar web-preglednika Firefox, Safari, Opera i Chrome. Broj prikazanih okvira u sekundi, vrijeme učitavanja okvira, opterećenje procesora, korištenost memorije i veličina datoteke su performanse koje su mjerene kod izvođenja animacija unutar različitih web-preglednika. Radom su objašnjene promatrane tehnologije za izradu web-animacija. Prikazan je izvorni kod mjernog instrumenta testiranja. Izvršeno je testiranje izvođenja animacija po promatranim tehnologijama.

Ključne riječi: performanse, Flash, HTML5 canvas, SVG, JavaScript

1. UVOD

Često se za način prikaza sadržaja kroz web-tehnologije koriste dvodimenzionalne animacije (web-animacije). Postoje razne tehnologije izgradnje takvih animacija.

Za izgradnju web-animacija u velikoj mjeri koristi se Flash tehnologija, koja je donijela mnogo novih mogućnosti za prezentaciju sadržaja na internetu. Flash tehnologija nastala je kako bi se tekst, slika i zvuk lakše strukturirali i prezentirali putem web-preglednika.

Osnovne web-tehnologije Skalarna vektorska grafika (engl. *Scalable Vector Graphics*, SVG), Javascript i HTML5 također omogućuju prikaz animacija i mogu se koristiti za izgradnju web-animacija.

Koja tehnologija omogućuje najbolje performanse izvođenja web-animacija? U radu se ispituju performanse izvođenja animacija Flash, SVG, JavaScript i HTML5 canvas unutar različitih web-preglednika. Testirani su parametri koji utječu na performanse izvođenja web-animacije, a to su:

¹ Bacc. ing. comp. E-mail: vedranharmadi@gmail.com

² Mr. sc., viši predavač, Veleučilište u Rijeci, Vukovarska 58, Rijeka, Hrvatska. E-mail: mkaluza@veleri.hr

³ Asistent, Veleučilište u Rijeci, Vukovarska 58, Rijeka, Hrvatska. E-mail: dliveric@veleri.hr

⁴ Datum primitka rada: 19. 2. 2014.; datum prihvatanja rada: 5. 5. 2014.

opterećenje procesora, korištenost memorije, veličina datoteke, broj prikazanih okvira u sekundi, vrijeme učitavanja okvira.

Opisane su pojedine tehnologije koje su primijenjene za izradu različitih tipova animacija. Opisani su tipovi testiranih animacija i postupak njihove izrade. Prikazana je funkcionalnost aplikacije koja je izgrađena za potrebe mjerenja.

Web-animacije, odnosno računalne animacije koje se izvršavaju direktno u web-pregledniku pogodnije su (brže, manje opterećuju resurse računala) od animacija koje koriste posrednički softver. To će se pokazati mjerenjem performansi izgrađenim web-animacijama u prije navedenim tehnologijama. Analizom 4 tehnologije za izradu web-animacija pokazat će se koja od njih daje najbolje rezultate po mjerenim parametrima.

2. OPIS TEHNOLOGIJA ZA IZRADU WEB-ANIMACIJA

Za izradu web-animacija postoje mnogobrojne tehnologije koje se danas koriste. Svaka od tih tehnologija ima svoje područje gdje se i zašto se koristi, kao i svoje prednosti i nedostatke. U nastavku će biti prikazane tehnologije koje su u ovom radu primijenjene za izradu web-animacija i nad kojima se provodilo testiranje.

2.1 Adobe Flash

U svojim počecima Flash je bio usmjeren i koristio se za izradu animacija i jednostavnih web-aplikacija koje su uključivale neke jednostavne igre, reklame i slično. Danas Flash predstavlja standard multimedijskog prikaza na internetu, koji dizajneri koriste za izradu različitih web-sadržaja (Carnet Cert, 2009).

Adobe Flash je multimedijaska tehnologija koja se već duže vrijeme koristi za izradu animacija, reklama, web-navigacije i raznih drugih elemenata web-stranica, za integraciju videa i/ili zvuka u web-stranici, a u posljednje vrijeme za izradu multimedijски bogatih web-aplikacija, poput igara. Isto tako Flash može manipulirati vektorskom i raster grafikom, te podržava dvosmjerno emitiranje zvuka i videa (Kovačević, 2009).

Za izradu i manipulaciju objektima animacije Flash koristi ActionScript skriptni jezik koji je sličan jezgri JavaScript programskog jezika, koji omogućava dodavanje kompleksnih interakcija, kontrolu *playbacka* i podatkovni prikaz. ActionScript, kao i ostali skriptni jezici, slijedi svoja pravila sintakse i ključnih riječi, koristi vlastite stvorene objekte i funkcije te dopušta kreiranje novih objekata i funkcija (Kovačević, 2009).

Adobe Flash tehnologija svojom je pojavom značajno pridonijela bogatstvu multimedijskog sadržaja na internetu, te omogućila dizajnerima više mogućnosti za kreativni prikaz. S obzirom na to da Flash tehnologija ima podršku u većini „desktop“ operacijskih sustava, na svim sustavima jednako se prikazuje, odnosno ima jednako ponašanje.

Nedostatak Flash tehnologije jest da za prikazivanje njegovog sadržaja korisnici moraju imati instaliranu jednu od verzija Flash čitača. Isto je tako velik nedostatak veličina datoteke u Flash

formatu te njezino dugo učitavanje kroz *web-preglednik*. Flash slabije podržavaju mobilni *web-preglednici* (Brekalo, Horvatić, 2013).

2.2 SVG

SVG (engl. *Scalable Vector Graphics*) jezik je koji opisuje dvodimenzionalnu vektorsku grafiku unutar XML dokumenta. SVG crta grafičke oblike koristeći objekte kojima je moguće posebno promijeniti vrijednost pojedinog atributa. SVG je neovisan o rezoluciji ekrana, što znači da zumiranjem ili promjenom veličine objekata slika neće gubiti na kvaliteti (W3C, 2002).

W3C definira specifikaciju SVG-a, a format zapisa bazira se na već standardiziranom XML formatu. SVG se prema XML-u odnosi kao i XHTML prema XML-u, tj. definirana su pravila i sintaksa jezika koja u potpunosti prate XML specifikaciju, odnosno baziraju se na njoj.

SVG specifikacija je otvoreni standard koji se razvija od 1999. godine. Današnji moderni *web-preglednici* omogućuju osnovnu podršku za SVG, te se uz JavaScript biblioteke danas češće koriste.

SVG sadržaj može biti interaktivan i dinamičan. Animacije se mogu definirati postavljanjem svojstava SVG elemenata ili JavaScript programskim kodom (W3C, 2011).

SVG za sada omogućuje prikaz samo 2D elemenata i transformacija nad njima, no uz pomoć CSS3 standarda moguće je raditi i 3D transformacije. SVG sadrži unaprijed definirane elemente koji predstavljaju određene oblike. Svaki od elemenata je objekt s vlastitim atributima čije se vrijednosti zadaju prilikom crtanja (Pap, Pavazza, 2012).

SVG je jezik koji je primarno namijenjen za primjenu na internetu i koji je prihvaćen u gotovo svim internetskim preglednicima. Osim vektorske grafike SVG podržava prikaz rasterskih slika te pretraživanje datoteka, kao i mogućnost indeksiranja sadržaja internetskim tražilicama. S obzirom na vektorski format, idealan je za prikaz modernih računalnih ikona.

Najčešći nedostatak je iscrtavanje SVG datoteke kompleksnog vektorskog sadržaja koje može potrajati vremenski dugo te pritom usporiti rad internetskih preglednika (Baotić, 2009).

2.3 HTML5 Canvas

HTML5 je sljedeća generacija HTML-a koja podržava nove značajke potrebne modernim *web-aplikacijama*. On ujedno standardizira mnoge značajke *web-platforame* koje *web-programeri* već godinama koriste, ali još nisu službeno odobrene ili dokumentirane pri standardizacijskoj komisiji. HTML5 je kolekcija pojedinačnih značajki (crtaća površina, video ili geolokacija). HTML5 definira crtaću površinu, odnosno element canvas kao rastersku površinu za crtanje ovisnu o rezoluciju koja se može koristiti za iscrtavanje grafova, grafike igara i drugih slika tijekom izvršavanja (W3C, 2013). Crtaća površina je pravokutnik na stranici po kojem je moguće pomoću JavaScript programskog jezika nacrtati razne oblike. HTML5 definira skup funkcija za crtanje oblika, definiranje staza, izradu gradijenata i primjenu transformacija (Pilgrim, 2010).

Canvas je element koji je izvorno podržan unutar *web-preglednika* i dio je objektnog modela dokumenta (engl. *Document Object Model, DOM*) (Hickson, 2014). Canvas

predstavlja revolucionarnu mogućnost dinamičkog generiranja sofisticirane grafike na klijentskoj strani, odnosno *web*-pregledniku. Razvio ga je Apple za implementaciju Dashboard *widgeta* i poboljšanje grafike u Safari pregledniku (Smith, 2005). Za razliku od nekih drugih tehnologija, kao što je primjerice Flash, za upotrebu canvas elementa nisu potrebne dodatne biblioteke, već je potreban HTML5 kompatibilan *web*-preglednik i uređivač teksta (Williams, 2011).

S obzirom na korištenje određene značajke HTML5 standarda potrebno je ispitati njezinu podršku, odnosno provjeriti sadrži li pojedini internetski preglednik podršku za korištenje određene značajke (Pilgrim, 2010). Svaki objekt koji se nalazi unutar canvas elementa strogo je definiran brojem i pozicijom piksela, što znači da je canvas ovisan o rezoluciji. Canvas se koristi za izradu zahtjevnijih grafičkih aplikacija, npr. igara, jer omogućuje brzo iscrtavanje složenijih scena (Pilgrim, 2010).

Canvas element omogućuje izradu naprednih grafičkih aplikacija bez potrebe za priključcima. Ova tehnologija dobiva sve veću podršku u svim modernim *web*-preglednicima, pri čemu se smanjuje potreba za priključcima (Sucan, 2009).

2.4 JavaScript

JavaScript je kompaktan skriptni jezik za razvoj internetskih aplikacija klijent-server. Programski kod upisuje se direktno u HTML stranicu i omogućuje izradu dinamičnih *web*-stranica. Postoje različite JavaScript biblioteke, pri čemu se njezina funkcionalnost može generalno podijeliti u tri osnovne kategorije (Deković, 2008):

- DOM (engl. *Document Object Model*) pristup i manipulacija sadržajem *web*-stranice
- jednostavnije i brže programiranje zbog već ugrađenih funkcionalnosti
- proširivanje funkcionalnosti stranica pomoću gotovih komponenti.

Različiti internetski preglednici mogu različito interpretirati JavaScript naredbe. Neki preglednici imaju podršku za metode i događaje koji nisu u skladu s *web*-standardima. Ako se skripte s takvim nestandardnim metodama izvode na preglednicima koji poštuju standarde, neće se moći izvršiti.

JavaScript skriptni jezik razvila je tvrtka Netscape čiji je *Web*-preglednik Netscape Navigator s udjelom od čak 90 %, bio dominantan na tržištu devedesetih godina prošlog stoljeća (Čačić, 2012).

JavaScript skriptni jezik omogućuje izmjenu sadržaja *web*-stranice bez njezinog ponovnog učitavanja, te time čini *web*-stranicu dinamičnom (Flanagan, 1997).

Kao i ostali skriptni jezici, JavaScript je brz jer se čita red po red. Provođenje JavaScript naredbi započinje odmah, jer JavaScript za svoje izvođenje ne zahtijeva vezu s poslužiteljem.

3. IZRADA I PRIKAZ WEB-ANIMACIJA

Pojam animacije podrazumijeva brzo prikazivanje niza 2D ili 3D slika modela postavljenih tako da stvaraju privid pokreta. Ostvaruje se prikazom nacrtanih objekata u različitim položajima u svakom okviru, pa izgleda kao da se objekt miče kad se okviri izmjenjuju određenom brzinom. Animacija je moguća zbog tromosti oka. Objekt koji je oko vidjelo ostaje zapamćen još nekoliko trenutaka nakon gledanja.

Neki tipovi animacija su:

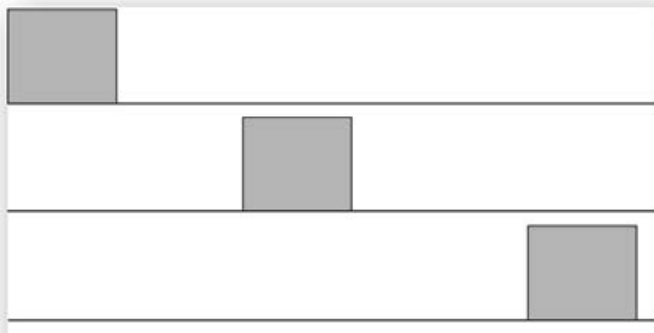
- animacija po stazi - objekt se pomiče po zadanoj stazi i pritom ne mijenja svoj oblik, već samo položaj
- animacija s različitim kadrovima - animacija počinje prvim ključnim kadrom i završava posljednjim ključnim kadrom koji mogu, ali i ne moraju biti jednaki. Između prvog i posljednjeg kadra prelazi se postupno, crtanjem odgovarajućeg broja međukoraka po zadanom putu.
- animacija preobražavanjem - računalni program generira međuslike, pa se jedna slika pretvara u drugu.

Za svaki prethodno navedeni tip animacije izrađene su identične animacije, animacije s istim tijekom izvođenja, pomoću Flash, SVG, HTML5 canvas i JavaScript tehnologija. Skriptni jezici koji su primijenjeni kod izrade animacija su JavaScript i ActionScript, pri čemu je JavaScript primijenjen za izradu SVG, HTML5 canvas i JavaScript animacija, dok je ActionScript primijenjen za izradu Flash animacija.

3.1 Animacija po stazi

Animacija po stazi koncipirana je tako da se oblik kvadrata kreće po određenoj stazi. Oblik kvadrata definiran je vrijednostima visinom i širinom 200 piksela, dok je staza po kojoj se oblik kvadrata kreće definirana vrijednostima visinom 20 piksela i širinom 1200 piksela. Vrijednost pomaka kvadrata po stazi povećava se svakih 5 milisekundi za 1 piksel. Na slici 1 prikazana je izrađena animacija po stazi s različitim vrijednostima pomacima, odnosno pozicijama oblika kvadrata.

Slika 1. Prikaz animacije po stazi

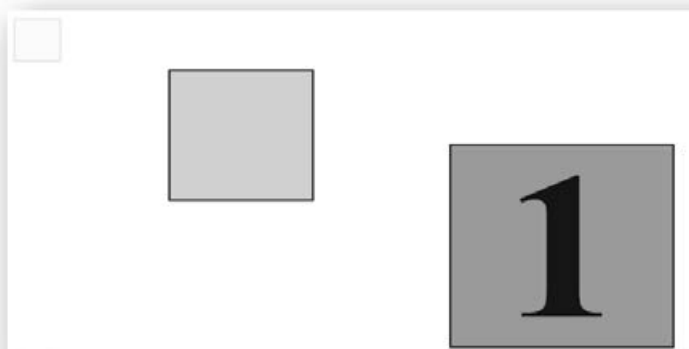


Izvor: obrada autora

3.2 Animacija s različitim kadrovima

Animacija s različitim kadrovima provodi se tako da se tijekom izvođenja animacije vrši promjena vrijednosti položaja, dimenzije, boje i vidljivosti oblika kvadrata. Oblik kvadrata definiran je vrijednostima, visinom i širinom 50 piksela. Tijekom prvog dijela izvođenja animacije, odnosno prvih 2500 milisekundi vrši se promjena vrijednosti položaja, dimenzije i vidljivosti kvadrata, dok se u drugom dijelu, odnosno drugih 2500 milisekundi, vrši promjena vrijednosti vidljivosti broja oblika kvadrata. Na slici 2 prikazana je izrađena animacija s različitim kadrovima.

Slika 2. Animacija s različitim kadrovima



Izvor: obrada autora

3.3 Animacija preobražavanjem

Animacija preobražavanjem koncipirana je tako da se tijekom izvođenja animacije oblik kvadrata pretvori u oblik kruga. Oblik kvadrata definiran je vrijednostima visinom i širinom 200 piksela. Tijekom izvođenja animacije vrši se promjena vrijednosti boje, rotacije i zaobljenosti ruba oblika kvadrata. Na slici 3 prikazana je izrađena animacija preobražavanjem s različitim prikazom promjena oblika kvadrata.

Slika 3. Animacija preobražavanjem



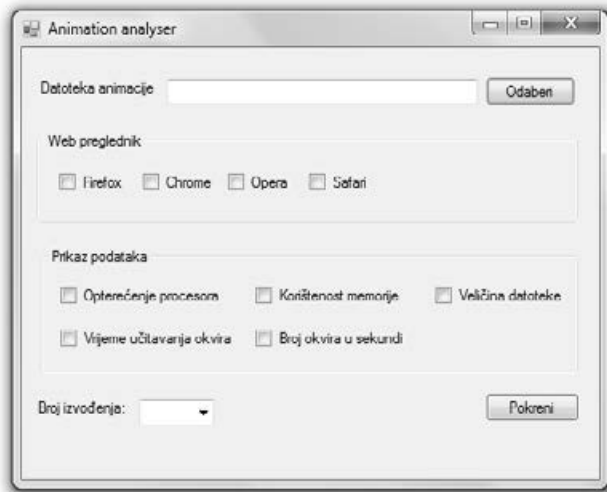
Izvor: obrada autora

4. MJERNI INSTRUMENTI TESTIRANJA

Za pokretanje animacija u različitim *web*-preglednicima te prikupljanje i obradu mjernih podataka izrađena je desktop aplikacija pomoću alata Microsoft Visual Studio. Aplikacija za testiranje omogućuje:

- odabir datoteke animacije nad kojom se želi provesti testiranje
- odabir *web*-preglednika unutar kojeg će se izvoditi animacija
- odabir broja ponavljanja izvođenja odabrane animacije
- bilježenje (engl. *log*) podataka koji se prikupljaju tijekom izvođenja animacije.
- Podaci koji se prikupljaju tijekom izvođenja animacije su:
- opterećenje procesora
- korištenost memorije
- veličina datoteke
- broj prikazanih okvira u sekundi
- vrijeme učitavanja okvira.

Slika 4. Prikaz korisničkog sučelja aplikacije testiranja animacija



Izvor: obrada autora

U svakoj sekundi izvođenja određene animacije vrši se prikupljanje gornjih podataka. Nakon završetka izvođenja animacije unutar određenog *web*-preglednika pojedinačno se izračunava aritmetička sredina svih prikupljenih mjernih podataka koji se nakon izračuna upisuju u datoteku. Na slici 4 prikazano je korisničko sučelje aplikacije za testiranje animacija.

4.1 Mjerenje opterećenja procesora

Prikupljanje mjernih podataka testiranja animacije izvršeno je primjenom različitih klasa i metoda. Mjerenje opterećenja procesora izvršeno je primjenom PerformanceCounter klase koja omogućuje praćenje i prikupljanje različitih vrsta podataka o procesima u realnom vremenu. Spomenuta PerformanceCounter klasa dio je Microsoft .NET biblioteke klasa. Nad procesom, odnosno *web-preglednikom* unutar kojeg se izvodi određena animacija, tijekom svake sekunde izvođenja izvršeno je mjerenje, odnosno prikupljanje podataka o opterećenju procesora. Vrijednost opterećenja procesora mjerena je u postotnom obliku. U nastavku je prikazan programski kod mjerenja opterećenja procesora za određeni *web-preglednik*:

```
double cpuProcesa = 0;
string nazivProcesa = 'firefox';

PerformanceCounter mjeracCpuProcesa = new PerformanceCounter();

mjeracCpuProcesa.CategoryName = 'Process';
mjeracCpuProcesa.CounterName = '% Processor Time';
mjeracCpuProcesa.InstanceName = nazivProcesa;

cpuProcesa += mjeracCpuProcesa.NextValue();

mjeracCpuProcesa.Close();
mjeracCpuProcesa.Dispose();
```

4.2 Mjerenje korištenosti memorije

Kao i za mjerenje opterećenja procesora, PerformanceCounter klasa primijenjena je kod mjerenja korištenosti memorije. Nad procesom, odnosno *web-preglednikom* unutar kojeg se izvodi određena animacija, tijekom svake sekunde izvođenja izvršeno je mjerenje, odnosno prikupljanje podataka o korištenju memorije. Vrijednost korištenosti memorije mjerena je u kilobajtima. U nastavku je prikazan programski kod mjerenja korištenosti memorije za određeni *web-preglednik*:

```
double memorijaProcesa = 0;
string nazivProcesa = 'firefox';

PerformanceCounter mjeracMemorijeProcesa = new PerformanceCounter();
mjeracMemorijeProcesa.CategoryName = 'Process';
mjeracMemorijeProcesa.CounterName = 'Working Set - Private';
mjeracCpuProcesa.InstanceName = nazivProcesa;

memorijaProcesa += Convert.ToInt32(mjeracMemorijeProcesa.NextValue()) / (int)(1024);

mjeracMemorijeProcesa.Close();
mjeracMemorijeProcesa.Dispose();
```


4.3 Mjerenje veličine datoteke

Mjerenje veličine datoteke animacije izvršeno je primjenom `FileInfo` klase koja omogućuje prikupljanje različitih vrsta podataka o datotekama. Spomenuta `FileInfo` klasa dio je Microsoft .NET biblioteke klasa. Vrijednost veličine datoteke animacije mjerena je u kilobajtima. U nastavku je prikazan programski kod mjerenja veličine datoteke određene animacije:

```
String nazivDatoteke = fileDialogAnimacija.FileName;  
  
FileInfo infoDatoteke = new FileInfo(@nazivDatoteke);  
String velicinaDatotekeAnimacije = Math.Ceiling((double)infoDatoteke.Length / 1024).ToString();
```

4.4 Mjerenje broja prikazanih okvira u sekundi

Mjerenje broja prikazanih okvira u sekundi (engl. *Frames per second*, FPS) kod SVG, JavaScript i HTML5 canvas animacija izvršeno je primjenom JavaScript `Stats` klase, dok je isto mjerenje kod Flash animacija izvršeno primjenom ActionScript `Stats` klase.

Nad procesom, odnosno web-preglednikom unutar kojeg se izvodi određena animacija, tijekom svake sekunde izvođenja izvršeno je mjerenje, odnosno prikupljanje podataka o broju prikazanih okvira u sekundi. Veća vrijednost broja prikazanih okvira u sekundi predstavlja brže izvođenje animacije. U nastavku je prikazan programski kod mjerenja broja prikazanih okvira u sekundi:

```
var stats = new Stats();  
var fps = 0;  
  
setInterval( function () {  
    stats.begin();  
    fps += stats.getFPS();  
    stats.end();  
}, 1000 );
```

4.5 Mjerenje vremena učitavanja okvira

Kao i za mjerenje vrijednosti broja prikazanih okvira u sekundi, JavaScript i ActionScript `Stats` klase primijenjene su kod mjerenja vremena učitavanja okvira. Nad procesom, odnosno web-preglednikom unutar kojeg se izvodi određena animacija, tijekom svake sekunde izvođenja izvršeno je mjerenje, odnosno prikupljanje podataka o vremenu učitavanja okvira. Manja vrijednost vremena učitavanja okvira, koja je mjerena u milisekundama, predstavlja brže izvođenje animacije. U nastavku je prikazan programski kod mjerenja vrijednosti vremena učitavanja okvira.

```
var stats = new Stats();  
var msOkvira = 0;  
  
setInterval( function () {  
    stats.begin();  
    fps += stats.getMS();  
    stats.end();  
}, 1000 );
```

5. TESTIRANJE ANIMACIJA I USPOREDBA REZULTATA

Prema statistici korištenosti preglednika vidljivo je da su Firefox i Google Chrome najviše korišteni preglednici (W3Schools, 2014; Buckler, 2014). Vjerojatno će se u skoroj budućnosti, povećati dosadašnja podržanost tehnologija za izradu animacija u svim *web*-preglednicima.

Svi tipovi animacija (poglavlje 3) izrađeni po opisanim tehnologijama (poglavlje 2) izvodili su se unutar Firefox, Chrome, Opera i Safari *web*-preglednika.

Izvođenje animacija nije izvršeno unutar *web*-preglednika Internet Explorer jer se programske procedure za mjerenje broja prikazanih okvira u sekundi i vremena učitavanja okvira nisu mogle izvesti.

Trajanje animacije ograničeno je na 5 sekundi izvođenja.

Za testiranje i mjerenje izvođenja animacija korišteni su prethodno navedeni preglednici i izrađeni softver. Mjerenje se provodilo tako da je prilikom provođenja testova na *web*-pregledniku jedino testni *web*-preglednik bio aktivan. Svi ostali aplikacijski programi korisnika bili su ugašeni.

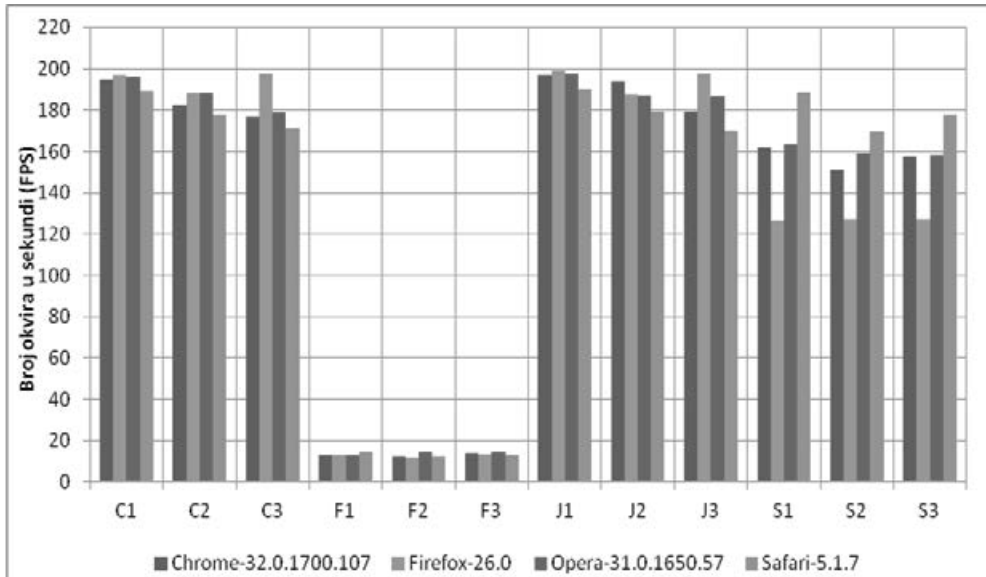
Svaka animacija unutar određenog *web*-preglednika izvršavala se 100 puta. Tijekom cijelog izvođenja svake pojedine animacije izvršeno je prikupljanje mjernih podataka. Nakon završetka izvođenja animacije, pojedinačno je nad svim prikupljenim mjernim podacima izvršen izračun aritmetičke sredine.

Testiranje animacija izvršeno je na računalnoj konfiguraciji sljedeće specifikacije:

- Operacijski sustav: Windows 7 (32-bit)
- Procesor: Pentium Dual-Core Processor, 2.1GHz
- Memorija: 4 GB DDR2
- Grafička kartica: Intel Graphics Media Accelerator 4500M
- Tvrdi disk: 320 GB.

Svaka tehnologija izrade i tip animacije grafički se unutar grafikona prikazuju u obliku kratica koje su definirane s dva znaka. Prvi znak kratice označava prvo slovo tehnologije izrade, dok drugi znak, odnosno broj, označava tip animacije. Animacija po stazi definirana je brojem 1, animacija s različitim kadrovima brojem 2, dok je animacije preobražavanjem definirana brojem 3. Primjerice, animacija po stazi izrađena Flash tehnologijom prikazuje se kraticom F1, dok se isto tako, SVG animacija s različitim kadrovima, prikazuje kraticom S2. Animacija preobražavanjem izrađena Canvas tehnologijom prikazuje se kraticom C3.

Grafikon 1. Broj prikazanih okvira u sekundi



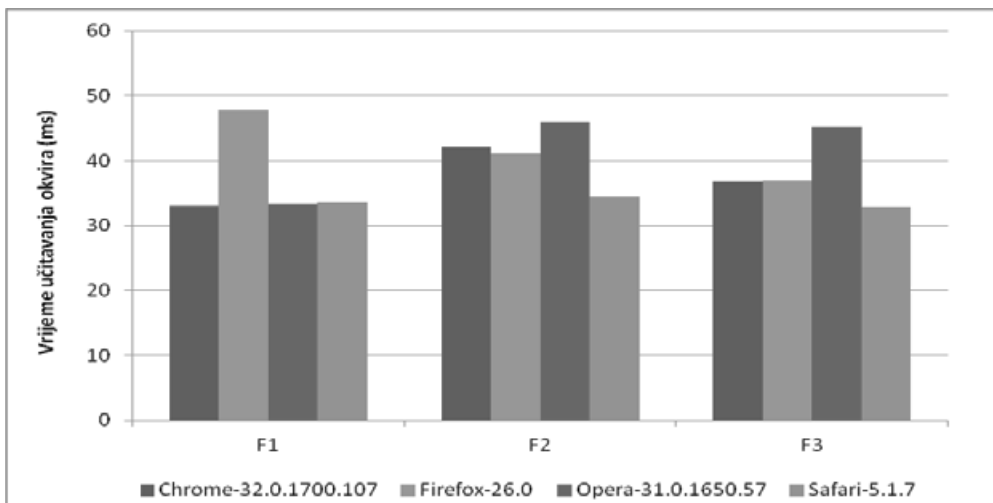
Izvor: obrada autora

Iz grafikona 1 može se vidjeti da su sve tri HTML5 canvas animacije ostvarile rezultate slične JavaScript animacijama. Ukupno u svim web-preglednicima ostvaruju najveću vrijednost broja prikazanih okvira u sekundi. JavaScript animacije ostvarile su u svim web-preglednicima slične vrijednosti broja prikazanih okvira u sekundi. SVG animacije ostvarile su nešto lošije rezultate od Javascript animacija. Suprotno tome, najsporiji prikaz svih tipova animacija ostvarile su animacije izrađene Flash tehnologijom. Flash animacije u usporedbi s HTML5 canvas animacijama unutar svih web-preglednika ostvarile su 14 puta manju vrijednost broja prikazanih okvira u sekundi.

Promatrajući pojedinačno web-preglednike kod HTML5 canvas i Javascript animacija, vidljivo je da se pomoću Safari preglednika prikazuje najmanji broj okvira u sekundi. Suprotno tomu, promatrajući samo SVG tehnologiju, pomoću Safari preglednika prikazuje se najviše okvira u sekundi. Firefox je kod HTML5 canvas i Javascript animacija uglavnom najbrži web-preglednik, dok je kod SVG animacija značajnije najsporiji web-preglednik.

Promatrajući pojedinačno vrste animacija vidi se da se animacije po stazi (1) prikazuju s uglavnom najviše okvira u sekundi, a animacije preobražavanjem (3) prikazuju se s uglavnom najmanje okvira po pojedinom pregledniku i pojedinoj tehnologiji.

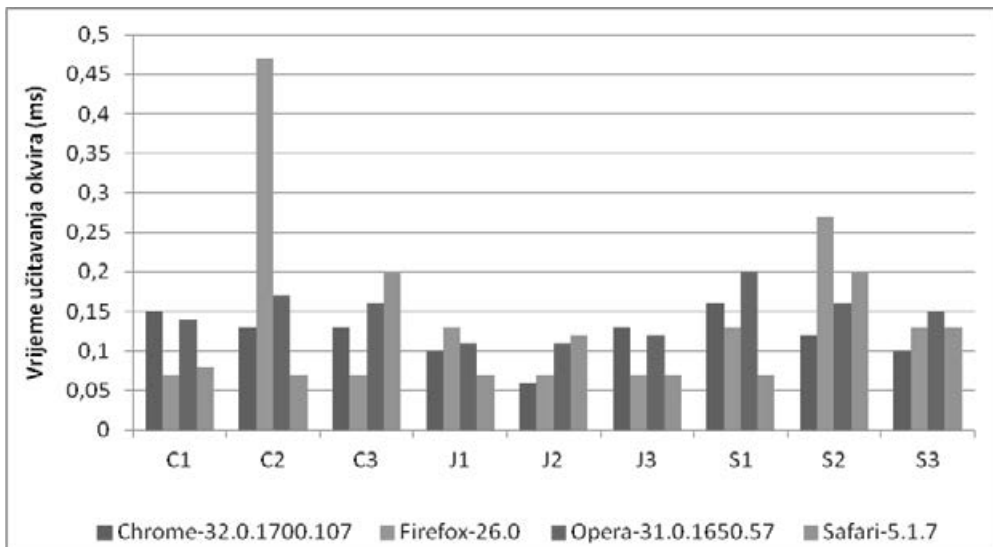
Grafikon 2. Vrijeme učitavanja okvira kod Flash animacija



Izvor: obrada autora

Iz grafikona 2 može se vidjeti da su sve tri Flash animacije ostvarile visoke vrijednosti vremena učitavanja okvira, zbog kojih je ostvaren znatno sporiji prikaz u odnosu na vrijeme učitavanja animacija koje su izgrađene drugim tehnologijama (grafikon 3). Flash animacije sporije su se prikazivale unutar svih web-preglednika u odnosu na animacije s ostalim tehnologijama, pri čemu je najsporije prikazivanje zabilježeno kod prikazivanja animacije po stazi unutar Firefox web-preglednika. Suprotno tome, ukupno gledano, najbrži prikaz svih Flash animacija ostvaren je prikazivanjem unutar Safari web-preglednika.

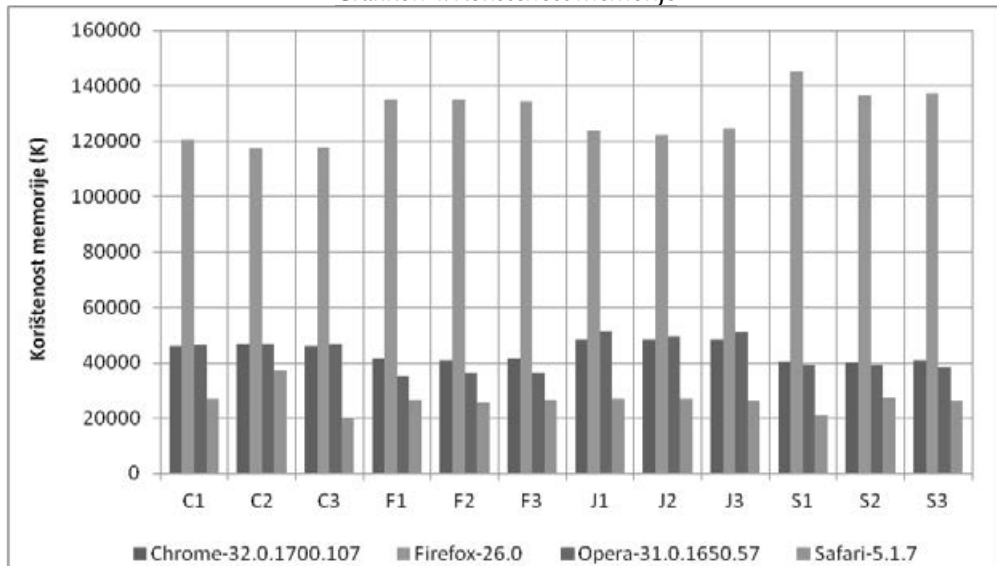
Grafikon 3. Vrijeme učitavanja okvira kod HTML5 canvas, JavaScript i SVG animacija



Izvor: obrada autora

Iz grafikona 3 može se vidjeti da je vrijeme učitavanja okvira kod svih triju HTML5 canvas, JavaScript i SVG animacija vrlo slično. S obzirom na male vrijednosti vremena učitavanja okvira ostvaren je brzi prikaz animacija unutar svih web-preglednika. Najveće vrijeme učitavanja okvira animacije zabilježeno je kod prikaza HTML5 canvas animacije s različitim kadrovima unutar Firefox web-preglednika.

Grafikon 4. Korištenost memorije



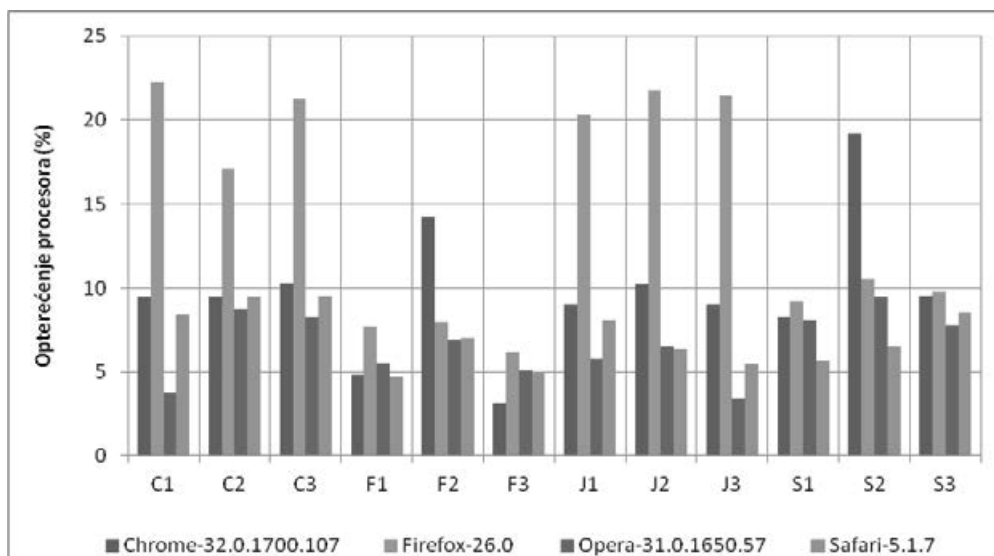
Izvor: obrada autora

Na grafikonu 4 može se vidjeti da svi tipovi animacija svih tehnologija po pojedinom web-pregledniku koriste približno jednaku količinu memorije. Razlika koja se može uočiti jest da je tijekom izvođenja svih triju animacija svih tehnologija unutar Firefox web-preglednika vrijednost korištenosti memorije znatno veća, odnosno da dostiže vrijednosti od čak 144 MB. Najmanja vrijednost korištenosti memorije tijekom izvođenja svih triju animacija svih tehnologija zabilježena je kod Safari web-preglednika. Preglednici Chrome i Opera ostvaruju uglavnom slične rezultate, ali troše više memorije od Safari preglednika.

HTML5 canvas i Javascript animacije na Chrome i Opera preglednicima troše više memorije od Flash i SVG animacija. Suprotno tomu HTML5 canvas i Javascript animacije na Firefox web-pregledniku troše manje memorije od Flash i SVG animacija.

Promatrajući tipove animacija po pojedinoj tehnologiji i pregledniku uočava se da ne postoji značajna razlika u korištenosti memorije.

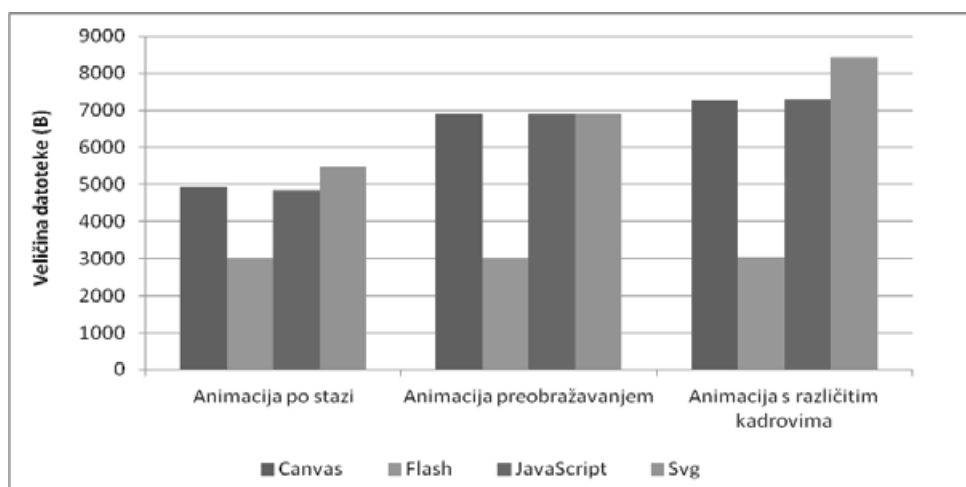
Grafikon 5. Opterećenje procesora



Izvor: obrada autora

Iz grafikona 5 može se vidjeti da su sve tri animacije Canvas, Flash, JavaScript i SVG po *web*-pregledniku ostvarile uglavnom slične vrijednosti opterećenja procesora. Najveća vrijednost, odnosno vrijednost od 22 posto opterećenja procesora zabilježena je kod izvođenja HTML5 canvas animacije po stazi unutar Firefox *web*-preglednika. Ukupno gledano, kod svih Flash animacija, opterećenje procesora najveće je kod njihovog izvođenja unutar Firefox *web*-preglednika. HTML5 canvas i JavaScript animacije s Firefox *web*-preglednikom značajnije opterećuju procesor (cca 20 %), u odnosu na ostale *web*-preglednike. Opera *web*-preglednik uglavnom najmanje opterećuje procesor kod HTML5 canvas i Javascript animacija.

Grafikon 6. Veličina datoteka svih triju animacija



Izvor: obrada autora

Grafikon 6 prikazuje veličinu datoteke u bajtovima svih triju animacija, na svim testiranim tehnologijama. Može se vidjeti da su veličine datoteka kod svih Flash animacija jednake i značajnije manje od animacija izrađenih pomoću ostalih tehnologija. Kod HTML5 canvas, JavaScript i SVG animacija vidljiva je sličnost u veličinama animacija po pojedinim tipovima animacija. Najveća veličina datoteke animacije, u odnosu na ostale, zabilježena je kod SVG animacije s različitim kadrovima. Vidljivo je rastući trend veličine datoteka iste tehnologije za različite tipove animacija, od najmanje animacije po stazi, do najveće animacije s različitim kadrovima.

6. DISKUSIJA

Iz navedenog istraživanja i dobivenih rezultata testiranja može se zaključiti da korištenje Flash tehnologije za izradu *web*-animacija nije najbolje rješenje. Za istu namjenu, značajno bolji rezultati postignuti su korištenjem HTML5 Canvas, SVG i JavaScript tehnologija, koje su za svoje prikazivanje animacija slično opterećivale procesor i koristile radnu memoriju. Velika razlika vidljiva je u brzini prikazivanja, gdje se Flash animacija prikazivala do 14 puta sporije od animacija izgrađenih u ostalim tehnologijama. Razlog tomu jest razlika između renderiranja sadržaja, pri čemu Flash tehnologija za svoj prikaz zahtijeva dodatak (engl. *plugin*), dok se kod ostalih tehnologija sadržaj renderira izvorno.

Brzina prikazivanja animacija različitih tehnologija potvrđuje se vremenom potrebnim za učitavanje okvira. Prema tome, vrijednost učitavanja okvira izravno utječe na vrijednost brzine prikazivanja animacije. Tako je, zbog visoke vrijednosti vremena potrebnog za učitavanje okvira, uočen znatno sporiji prikaz Flash animacija.

Golemi nedostatak Flash tehnologije za izradu *web*-animacija jest u tome što je za svoj znatno sporiji prikaz svih tipova animacija slično zahtijevala performanse računala, kao i animacije izgrađene ostalim tehnologijama, kod kojih je ostvaren znatno brži prikaz, u nekim slučajevima čak 14 puta brži.

Što se tiče samih *web*-preglednika, iz rezultata je vidljivo da su tijekom testiranja svih izrađenih animacija, Chrome, Safari i Opera *web*-preglednici, za njihovo izvođenje, najmanje opterećivali procesor i koristili memoriju. Suprotno tomu, najveća vrijednost opterećenja procesora i korištenosti memorije zabilježena je kod izvođenja svih animacija unutar Firefox *web*-preglednika.

Animacije svih tehnologija podjednako su se prikazivale unutar svih *web*-preglednika. To je pokazatelj da sve promatrane tehnologije za izradu *web*-animacija imaju zadovoljavajuću podršku njihovog izvođenja kod svih *web*-preglednika.

7. ZAKLJUČAK

Sveopća prisutnost informacijsko-komunikacijskih tehnologija (ICT) i potrebe korisnika za upotrebom i obradom multimedijских sadržaja zahtijevaju razvoj i uvođenje novih tehnologija koje će omogućiti brži i jednostavniji pristup sadržajima i koje će poboljšati performanse sustava. Svaka nova tehnologija mora proći dug vremenski period da bi se dokazala kao dobar nov standard.

Na temelju brzine prikazivanja animacije, odnosno vrijednosti broja prikazanih okvira u sekundi, može se uvidjeti da se Flash tehnologija ne može smatrati najpogodnijom tehnologijom za izradu *web*-animacija. Činjenica koja potvrđuje sporost prikaza Flash animacija na *web*-pregledniku jest vrijeme koje je potrebno za učitavanje okvira animacije.

Prema dobivenim rezultatima testiranja animacija svih tehnologija, može se zaključiti da su HTML5 canvas, JavaScript i SVG pogodne tehnologije za izradu animacija na internetu. Za izgradnju animacije u navedenim tehnologijama potreban je uređivač teksta, dok je za izgradnju Flash animacija potrebna specijalizirana aplikacija. Flash zahtijeva programski dodatak (*plugin*) za izvođenje u pregledniku, a HTML5 canvas, Javascript i SVG izvršavaju se direktno u pregledniku.

S obzirom na to da su HTML5 canvas i Javascript animacije ostvarile najveći broj prikazanih okvira u sekundi, te neznčajnu razliku u korištenosti memorije i opterećenju procesora u odnosu na ostale tehnologije, mogu se smatrati najpogodnijim tehnologijama za izradu animacija na internetu. Činjenica koja to potvrđuje jest nativnost HTML5 tehnologije kojom je omogućeno da se objekti ne ugrađuju u *web*-dokument, već da se tretiraju kao izvorni. Takav pristup omogućuje niz prednosti kao što su: nezahtjevnost za dodacima, brzina, navigacijske kontrole u sklopu *web*-preglednika, upravljanje tipkovnicom.

Isto tako, izbor HTML5 kao najpogodnije tehnologije za izradu animacija potvrđuje njezina pozicija najboljeg rješenja za razvoj animacija koje zadovoljavaju zahtjev – može se izvoditi na svim platformama.

LITERATURA

- Baotić, G. (2009) *Najčešće korišteni grafički formati*, Završni rad, Zagreb: Fakultet elektrotehnike i računarstva, <http://www.baotic.net/graficki-formati-na-webu/vektorski/svg/> (pristupano 13. 2. 2014.)
- Brekalo, S., Horvatić, S. (2013) *Usporedba značajki web animacija kreiranih u SVG, HTML5 Canvas i Flash tehnologijama*, Zbornik radova Međimurskog veleučilišta u Čakovcu, Vol. 4 No. 2, <http://hrcak.srce.hr/113470>
- Buckler, C. (2014) *Browser Trends January 2014: a Yearly Review*, <http://www.sitepoint.com/browser-trends-january-2014-yearly-review/> (pristupano 17. 2. 2014.)
- Carnet Cert (2009) *Ranjivosti Adobe Flash tehnologije*, <http://www.cert.hr/sites/default/files/CCERT-PUBDOC-2009-05-264.pdf> (pristupano 18. 2. 2014.)
- Čačić, B. (2012) *Razvoj naprednih korisničkih sučelja u web 2.0 okruženju*, Grafički fakultet, Zagreb, http://eprints.grf.hr/930/1/DB104_%C4%8Ca%C4%8Di%C4%87_Marko.pdf (pristupano 29. 4. 2014.)
- Deković, T. (2008) *JavaScript biblioteke*, <http://tomislavdekovic.iz.hr/sheshnjak/clanci/javascriptBiblioteke.html> (pristupano 13. 2. 2014.)
- Flanagan, D. (1997) *Javascript: The Definitive Guide*, O'Reilly & Associates, <http://docstore.mik.ua/oreilly/web/jscript/index.html> (pristupano 30. 4. 2014)
- Hickson, I. (2014) *HTML Living standard*, <http://www.whatwg.org/specs/web-apps/current-work/multipage/> (pristupano 30. 4. 2014)
- Kovačević, D. (2009) *Animirani 3D likovi u Flash tehnologiji*, Diplomski rad, Fakultet elektrotehnike i računarstva, Zagreb, <http://hotlab.fer.hr/images/50009024/diplomski-Kovacevic.pdf> (pristupano 13. 2. 2014.)
- Pap, K., Pavazza, S. (2012) *The Alternative Way of Creating Infographics Using SVG Technology*, Grafički fakultet, Zagreb, <http://hrcak.srce.hr/file/128170> (pristupano 29. 4. 2014.)
- Pilgrim, M. (2010) *HTML5 Spreman za upotrebu*, Zagreb: Dobar plan
- Smith, D. (2005) *Dashboard Widgets for MAC OS X Tiger: Visual QuickStart Guide*, Peachpit Press, Berkeley, CA, 2005)
- Sucan, M. (2009) *HTML5 canvas – the basics*, <http://dev.opera.com/articles/view/html-5-canvas-the-basics/> (pristupano 18. 2. 2014.)
- W3C (2002) *Scalable Vector Graphics (SVG) 1.2*, World Wide Web Consortium, <http://www.w3.org/TR/2002/WD-SVG12-20021115/> (pristupano 30. 4. 2014)
- W3C (2011) *Scalable Vector Graphics (SVG) 1.1*, World Wide Web Consortium, <http://www.w3.org/TR/SVG/> (pristupano 30. 4. 2014)
- W3C (2013) *HTML Canvas 2D Context*, World Wide Web Consortium, <http://www.w3.org/TR/2dcontext/> (pristupano 30. 4. 2014)
- W3Schools (2014) *Browser Statistics*, http://www.w3schools.com/browsers/browsers_stats.asp (pristupano, 29. 4. 2014.)
- Williams, J. (2011) *Learning HTML5 Game Programming: A Hands-on Guide to Building Online Games Using Canvas SVG and WebGL*, Addison-Wesley Professional, 2011.

Vedran Harmadi¹

Marin Kaluža²

Dijana Liverić³

Professional paper

UDC 004.928:004.73852

COMPARISON OF TECHNOLOGIES ENABLING ANIMATION DISPLAY IN DIFFERENT WEB BROWSERS⁴

ABSTRACT

This paper presents the analysis of commonly used technologies for building and execution of animation in different web browsers. Flash is a technology that has significantly contributed to the possibilities of creating a web animation, and certainly to the development of other technologies. Over the years, a number of technologies have been developed, like HTML5 canvas, which brings us a huge amount of improvements over the old document-oriented web. The main goal of this paper is to present which technology achieves the best performance in animation execution in different web browsers. The paper analyzes execution performances of Flash, HTML5 canvas, SVG, and JavaScript animations in Firefox, Safari, Opera, and Chrome web browsers. Measurement was performed on a frame frequency - frames per second (FPS), loading time, CPU usage, file size and RAM usage. The paper also presents the source code for tests that were performed on the observed animations for measuring their performances.

Key words: performance, Flash, HTML5 canvas, SVG, JavaScript

¹ Bacc. ing. comp. E-mail: vedranharmadi@gmail.com

² MSc, Senior Lecturer, Polytechnic of Rijeka, Vukovarska 58, Rijeka, Croatia. E-mail: mkaluza@veleri.hr

³ Assistant, Polytechnic of Rijeka, Vukovarska 58, Rijeka, Croatia. E-mail: dliveric@veleri.hr

⁴ Received: 19. 2. 2014.; accepted: 5. 5. 2014.