



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Deivydas Pocius

**Didelio formato nuotraukų manipuliavimo naršyklėje realizavimas ir
algoritmų tyrimas**

Baigiamasis magistro projektas

Vadovas

Doc. dr. Šarūnas Packevičius

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

**Didelio formato nuotraukų manipuliavimo naršyklėje realizavimas ir
algoritmų tyrimas**

Baigiamasis magistro projektas
621E16001 Programų sistemų inžinerija (kodas M4046N21)

Vadovas

Doc. dr. Šarūnas Packevičius

Recenzentas

Doc. dr. Tomas Blažauskas

Projektą atliko

Deivydas Pocius

KAUNAS, 2017



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos

(Fakultetas)

Deividas Pocius

(Studento vardas, pavardė)

Programų sistemų inžinerija (kodas M4046N21)

(Studijų programos pavadinimas, kodas)

„Didelio formato nuotraukų manipuliavimo naršyklėje realizavimas ir algoritmų tyrimas“

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 ____ m. _____ d.
Kaunas

Patvirtinu, kad mano, **Deivido Pociaus**, baigiamasis projektas tema „Didelio formato nuotraukų manipuliavimo naršyklėje realizavimas ir algoritmų tyrimas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Deivydas Pocius. Large-scale images manipulation in browser - implementation and research of algorithms LARGE-SCALE IMAGES MANIPULATION IN BROWSER – IMPLEMENTATION AND RESEARCH OF ALGORITHMS: *Master's thesis in* / supervisor assoc. prof. Šarūnas Packedvičius. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: Program software systems

Key words: *printing, html5, browser editor, photo processing browser, speed*

Kaunas, 2017. 05 p.

SUMMARY

Large format photo printing market is constantly growing. Companies which are offering large-scale printing usually only let you upload your images file - PDF, "Adobe Photoshop", "Adobe InDesing" and in other formats in their websites.

HTML5 standard introduced new technologies that created more possibilities for browsers and more companies are investing in this area. That's why we can assume that this technology has enough functionality for creating browser editor, which is able to process large-format pictures.

Therefore, the main job is to create a browser editor, which is able to manipulate large-format photographs and do an experiment that would compare the existing browser technologies and their suitability for large-format images manipulation. The paper suggested checking main HTML5, Canvas, WebGL and SVG technology. SVG as the source of experiment was rejected because the use of the same images display and manipulation technique as HTML5. However, some of the technologies are complex and requires libraries that would speed up creation. So, libraries that were chosen: HTML5 - JQuery and GSAP. Meanwhile WebGL - ThreeJS library and Canvas - it's standart implementation.

Effectiveness of the proposed libraries/technologies has been proven experimentally. The experiment showed that the Canvas with a photo, which dimensions were up to 10000px x 10000px showed good results (without delays), but with larger size photos for Canvas it was hard to handle - with 12000px x 12000px dimensional picture average delay was big - 6 s. However, this technology is widely used because it is versatile enough - has a perfect compatibility between browsers. WebGL showed the best results - even with 14000px x 14000px size image delay was the lowest - 0.2s. However, this technology is oriented more to 3D applications and only 70 percent of computers support it - also if your computer support it but does not have a video card - it may be that it can support only pictures up to 8192px x 8192px dimensions. This reduces the proportion of potential customers. Meanwhile, GSAP with 14000px x 14000px dimension size image delayed by 2s only - which makes it the second best and also - more universal than WebGL. JQuery with 14000px x

14000px dimension size image delayed more than 4 s. So if you focus on large-format printing and the wide range of consumers - it is the best to use GSAP.

Deividas Pocius. DIDELIO FORMATO NUOTRAUKŲ MANIPULIAVIMO NARŠYKLĖJE REALIZAVIMAS IR ALGORITMŲ TYRIMAS. Magistro baigiamasis projektas / vadovas doc. dr. Šarūnas Packedvičius; Kauno technologijos universitetas, Informatikos fakultetas.

Mokslų kryptis ir sritis: Programų sistemų inžinerija

Reikšminiai žodžiai: *spausdinimas, html5, naršyklės rengyklė, nuotraukų apdorojimas naršyklėje, greitis*

Kaunas, 2017. 05 p.

SANTRAUKA

Didelio formato nuotraukų spausdinimo rinka nuolat auga. Įmonės siūlančios plataus formato nuotraukų spausdinimą savo svetainėse vartotojui dažniausiai siūlo tik įkelti nuotraukų failą – „PDF“, „Adobe Photoshop“, „Adobe InDesign“ ir kitais formatais.

HTML5 standarto pristatytos technologijos praplėtė galimybes naršyklių programėlių ir įmonės vis daugiau investuoja į šią sritį. Todėl galima daryti prielaidą, kad šios technologijos turi pakankamai funkcionalumo, kad būtų galima sukurti naršyklės rengyklę, kuri sugebėtų apdoroti plataus formato nuotraukas. Todėl šio darbo pagrindinis tikslas yra sukurti naršyklės rengyklę, kuri sugebėtų manipuluoti plataus formato nuotraukomis, padaryti eksperimentą, kuriame palyginti naršyklėje veikiančias technologijas ir jų tinkamumą plataus formato nuotraukų manipuliacijai.

Darbe siūloma tikrinti įprastą HTML5, Canvas, WebGL, SVG technologijas. SVG kaip eksperimento šaltinis buvo atmestas, kadangi naudoja ta patį paveiksliukų atvaizdavimo ir manipuliavimo būdą kaip ir HTML5. Tačiau, kai kurios technologijos sudėtingos, todėl reikia bibliotekų, kad pagreintinti kūrimą. Todėl buvo pasirinkta HTML5 - JQuery, GSAP bibliotekos. Tuo tarpu WebGL - ThreeJS biblioteka, o Canvas nebuvo naudojama biblioteka.

Pasiūlytų bibliotekų efektyvumas buvo įrodomas eksperimentu. Eksperimentas parodė, kad Canvas su nuotraukomis, kurių matmenys iki 10000px x 10000px susidoroja puikiai, o su didesnių matmenų nuotraukomis sunkiai susidoroja - su 12000px x 12000px matmenų nuotrauka vėlavimas vidutiniškai 6s. Tačiau yra plačiai naudojamas, nes yra pakankamai universalus - turi gerą suderinamumą tarp naršyklių. WebGL geriausia, kadangi vėlavimas su 14000px x 14000px buvo mažiausias - 0.2s, todėl netrukde darbu. Tačiau ši technologija orientuota į 3D programas, 70 proc. kompiuterių palaiko ją ir tai - jeigu kompiuteris neturi vaizdo plokštės gali būti, kad bus palaikomas tik 8192px x 8192px matmenų dydis. Todėl sumažėja vartotojų dalis. Tuo tarpu GSAP su 14000px x 14000px nuotrauka vėlavo iki 2s, o JQuery iki 4s. Todėl jei orientuojamasi į plataus formato nuotraukas ir į platų vartotojų spektrą geriausia būtų GSAP naudoti.

TURINYS

Paveikslėlių sąrašas.....	8
Lentelių sąrašas.....	9
1. Įvadas.....	10
2. Analitinė dalis.....	12
2.1. Poreikis.....	12
2.3. Nagrinėjamos problemos.....	17
2.4. Darbo tikslas.....	21
2.5. Uždaviniai.....	21
2.6. Laukiami rezultatai.....	21
2.7. Naujumas.....	21
2.8. Esami sprendimai.....	21
2.9. Siūlomas metodas tyrimui.....	26
3. Projektinė dalis.....	28
3.1. Sistemos paskirtis.....	28
3.2. Sistemos funkcijos ir reikalavimai sistemai.....	28
4. Tyrimo dalis.....	45
5. Eksperimento dalis.....	49
5.2. Paveiksliais manipuliavimo naršyklėje technologijų eksperimentas.....	50
5.3. Eksperimento aplinka.....	51
5.4. Naudojamos plataus formato nuotraukos.....	51
5.7. Eksperimento eiga ir rezultatai.....	53
5.8. Išvados.....	60
6. Išvados.....	62
7. Literatūra.....	63
8. Terminų ir santrumpų žodynas.....	67
9. Priedai.....	68

PAVEIKSLĖLIŲ SĄRAŠAS

1 pav. Pagrindinių spausdinimo paslaugų rinkos segmentų pasiskirstymas (2012)	13
2 pav. Canvas elemente nupieštas apskritimas	15
3 pav. WebGL palaikomumas kompiuteriuose	16
4 pav. WebGL palaikomumas išmaniuose įrenginiuose – mobiliuosiuose telefonuose	17
5 pav. Naršyklės pateiktas “SVG” elemento atvaizdas	17
6 pav. Bendras architektūros pateikimo vaizdas	29
7 pav. Plataus formato nuotraukų kūrimo įrankio panaudojimo atvejų diagrama	30
8 pav. Plataus formato nuotraukų kūrimo įrankio paketų diagrama	30
9 pav. MainCore paketo klasių diagrama	31
10 pav. ExportManager paketo klasių diagrama	32
11 pav. AlbumManager paketo klasių diagrama	33
12 pav. Orders paketo klasių diagrama	34
13 pav. DocumentsManager paketo klasių diagrama	35
14 pav. DocumentsManager paketo – DocumentModel paveldėjimo schema	35
15 pav. Sistemos išdėstymo schema – PDF failo kūrimo kontekste	36
16 pav. Pagrindinis sistemos langas – rodomi albumai, dokumentai	38
17 pav. Dokumento kūrimo langas – matoma navigacija, dokumentas	38
18 pav. Užsakymo forma	39
19 pav. Jasmine vienetų testų rezultato langas	42
20 pav. Užpildymo algoritmo optimizacijos – greičių palyginimas tarp naršyklių	48
21 pav. Bibliotekų palyginimas greičio parametru	50
22 pav. Paieškos rezultatų kiekio palyginimas	50
23 pav. Pasirinktos testuoti technologijas pavyzdinė nuotrauka	52
24 pav. Pirmo kompiuterio CPU (%) eksperimento rezultatai	55
25 pav. Pirmo kompiuterio GPU (MB) eksperimento rezultatai	56
26 pav. Pirmo kompiuterio vėlavimo (sekundėmis) eksperimento rezultatai	56
27 pav. Antro kompiuterio CPU (%) eksperimento rezultatai	59
28 pav. Antro kompiuterio GPU (MB) eksperimento rezultatai	59
29 pav. Antro kompiuterio vėlavimo (sekundėmis) eksperimento rezultatai	60

LENTELIŲ SĄRAŠAS

2 lentelė. Naršyklėje veikiančių rengyklų skirtų tik pardavimui palyginimas.....	23
3 lentelė. Įmonės turinčios savo sukurtą rengyklę ir ją yra pritaikiusios savo reikmėms palyginimas	25
4 lentelė. Dokumento kūrimo testavimo rezultatai	39
5 lentelė. Paveiksluko manipuliavimo dokumente testavimo rezultatai	39
6 lentelė. PDF failo įkėlimo į užsakymą testavimo rezultatai.....	40
7 lentelė. Testuotini metodai vartotojo sąsajos	41
8 lentelė. Testuotini metodai serverio pusės	41
8 lentelė. Naršyklėje veikiančių technologijų palyginimas	45
9 lentelė. Naršyklėje veikiančių bibliotekų palyginimas	46
10 lentelė. Naršyklėje veikiančių technologijų tyrimo rezultatai	49
11 lentelė. Testavimui naudotų nuotraukų charakteristikos.....	52
12 lentelė. 1 kompiuterio HTML5 jQuery UI bibliotekos eksperimento rezultatai.....	53
13 lentelė. 1 kompiuterio GSAP bibliotekos eksperimento rezultatai	54
14 lentelė. 1 kompiuterio Canvas technologijos eksperimento rezultatai.....	54
15 lentelė. 1 kompiuterio WebGL technologijos eksperimento rezultatai.....	55
16 lentelė. 2 kompiuterio HTML5 jQuery UI bibliotekos eksperimento rezultatai.....	57
17 lentelė. 2 kompiuterio GSAP bibliotekos eksperimento rezultatai	57
18 lentelė. 2 kompiuterio Canvas technologijos eksperimento rezultatai.....	58
19 lentelė. 2 kompiuterio WebGL technologijos eksperimento rezultatai.....	58

1. ĮVADAS

Šiuo metu ypač didėja skaitmeninių spausdinimo paslaugų pardavimai. Žmonės vis daugiau fotografuoja ir jau padaro daugiau kaip 1.8 milijardo nuotraukų kiekvieną dieną [1]. Anot vieno iš interneto kūrėjų („Vint Cerf“) –, jeigu norite išsaugoti nuotrauką – geriausia jas išspausdinti, kadangi išspausdintos nuotraukos gali laikyti daugiau kaip 100 metų, o nuotraukos laikomos skaitmeninėje laikmenoje gali ir po 10 metų būti nebuskaitomos [2]. Šis teiginys patvirtina ir rinkos tendencijas – žmonės vis mažiau pasikliauja skaitmeninėmis talpyklomis savo nuotraukoms ir renkasi spausdinimo paslaugas [1].

Pagal pasaulinės įmonės „Smithers Pira“, kuri užsiima pakavimu, popieriaus gamyba ir spausdinimu, rinkos tyrimo – skaitmeninio spausdinimo industrija augs nuo 131.5 milijardo dolerių 2013 metais iki 187.7 milijardo dolerių 2018 metais dėl naujų rašalinių technologijų taikymo [3]. Šiuo metu daugiausia didelio formato nuotrauką spausdina reklamuotojai - jie sudaro didžiausią 31 proc. rinkos dalį.

Įmonės siūlančios plataus formato nuotraukų spausdinimą savo svetainėse vartotojui dažniausiai siūlo tik įkelti nuotraukų failą – „PDF“, „Adobe Photoshop“, „Adobe InDesign“ ir kitais formatais [4] [5]. Tokio formato ir dydžio failai dažniausiai kuriami profesionalia programine įranga „Adobe Photoshop“, „Adobe InDesign“, kuria naudotis nėra paprasta ir palyginti su naršyklių programomis (rengyklėmis) yra gerokai brangiau [1].

Atsižvelgiant į rinką – HTML5 standarto pristatytos technologijos praplėtė naršyklių programėlių galimybes ir įmonės vis daugiau investuoja į šią sritį [8]. Todėl matant didėjančią rinką ir naršyklės rengyklių populiarumo didėjimą lyginant su darbatalio programomis buvo nuspręsta kurti savo naršyklės rengyklę orientuotą į plataus formato nuotraukų manipuliavimą naršyklėje – ir PDF formatu eksportavimą. O HTML5 standarto pristatytos technologijos teikia daug vilčių. Todėl šio darbo pagrindinis tikslas yra sukurti naršyklės rengyklę, kuri gebėtų manipuluoti plataus formato nuotraukomis, padaryti eksperimentą, kuriame palyginti naršyklėje veikiančias technologijas ir jų tinkamumą plataus formato nuotraukų manipuliacijai.

Darbe siūloma tikrinti įprasta HTML5, Canvas, WebGL, SVG technologijas. SVG kaip eksperimento šaltinis buvo atmestas, kadangi naudoja ta patį paveikslukų atvaizdavimo ir manipuliavimo būdą kaip ir HTML5. Tačiau, kai kurios technologijos sudėtingos, todėl reikia bibliotekų, kad pagreintinti kūrimą. Todėl buvo pasirinkta HTML5 - JQuery, GSAP bibliotekos. Tuo tarpu WebGL - ThreeJS biblioteka, o Canvas nebuvo naudojama biblioteka. Pasiūlytų bibliotekų efektyvumas buvo įrodomas eksperimentu.

Šis realizuotas įrankis padės sukurti PDF failą, kurio matmenų dydis iki 14000px x 14000px. Toks ir nuotraukų dydis didžiausias parinktas, kurį bus galima manipuluoti. Jis skirtas neprofesionaliam naudojimui bus.

Darbo analitinėje dalyje apžvelgiama rinka, esminiai principai naršyklės technologijų, kaip jos veikia ir pavyzdžiai. Pateikiami palyginimai esamų produktų iš dviejų pusių: tų, kurie kuria programinę įrangą ir patys spausdina ir tų, kurie kuria programinę įrangą spausdinimo įmonėms (patys nespausdina). Apibrėžiamos kylančios problemos ir analizuojami sprendimai. Projekto metu realizuojamas įrankis. Jame realizuojamos pasirinktos technologijos ir aprašoma architektūra bei testavimo etapas. Tyrimo metu atliekama technologijų apžvalga ir pasirinktų bibliotekų. Taip pat aptariami optimizacijos galimybės. Eksperimento metu palyginamos technologijos ir bibliotekos kiekybiškai praktiškai testuojant ir vertinant kaip buvo susidorota su kiekviena pasirinkta nuotrauka.

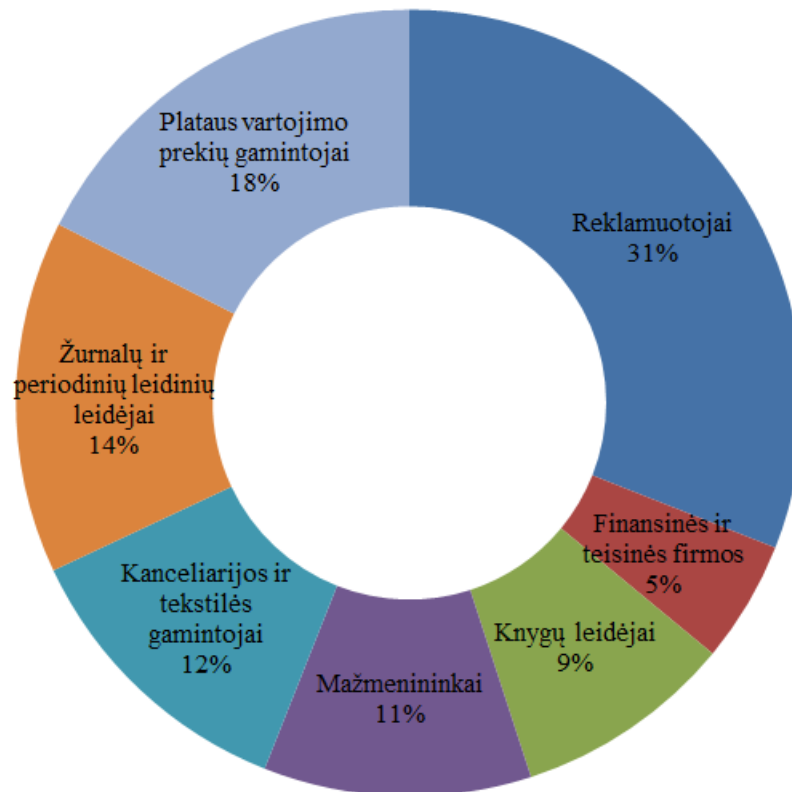
2. ANALITINĖ DALIS

2.1. Poreikis

Šiuo metu ypač didėja skaitmeninių spausdinimo paslaugų pardavimai. Žmonės vis daugiau fotografuoja ir jau padaro daugiau kaip 1.8 milijardo nuotraukų kiekvieną dieną [1]. Anot vieno iš interneto kūrėjų („Vint Cerf“) –, jeigu norite išsaugoti nuotraukas – geriausia jas išspausdinti, kadangi išspausdintos nuotraukos gali laikyti daugiau kaip 100 metų, o nuotraukos laikomos skaitmeninėje laikmenoje gali ir po 10 metų būti nebenuskaitomos [2]. Šis teiginys patvirtina ir rinkos tendencijas – žmonės vis mažiau pasikliauja skaitmeninėmis talpyklomis savo nuotraukoms ir renkasi spausdinimo paslaugas [1].

Pagal pasaulinės įmonės „Smithers Pira“, kuri užsiima pakavimu, popieriaus gamyba ir spausdinimu, rinkos tyrimo – skaitmeninio spausdinimo industrija augs nuo 131.5 milijardo dolerių 2013 metais iki 187.7 milijardo dolerių 2018 metais dėl naujų rašalinių technologijų taikymo [3]. Vadinasi rinka nuolat auga šioje srityje.

Pagrindinių spausdinimo paslaugų rinkos segmentų pasiskirstymas (2012)



Viso: \$76.6 milijardai

Šaltinis: WWW.IBISWORLD.COM

1 pav. Pagrindinių spausdinimo paslaugų rinkos segmentų pasiskirstymas (2012)

Šiuo metu daugiausia didelio formato nuotraukas spausdina reklamuotojai - jie sudaro didžiausią 31 proc. rinkos dalį, kurią matome 1 pav.. Taip pat svarbi didelio formato paveikslukų rinkos dalis yra plataus vartojimo prekių gamintojai, kurie sudaro 18 proc., nors suprantama, kad šioje srityje didelių paveikslukų spausdinimas žymios dalies gali nesudaryti.

Įmonės siūlančios plataus formato nuotraukų spausdinimą savo svetainėse vartotojui dažniausiai siūlo tik įkelti nuotraukų failą – „PDF“, „Adobe Photoshop“, „Adobe InDesign“ ir kitais formatais [4] [5]. Tokio formato ir dydžio failai dažniausiai kuriami profesionalia programine įranga „Adobe Photoshop“, „Adobe InDesign“, kuria naudotis nėra paprasta ir palyginti su naršyklių programomis (rengyklėmis) yra gerokai brangiau [1].

Atsižvelgiant į rinką – HTML5 standarto pristatytos technologijos praplėtė naršyklių programėlių galimybes ir įmonės vis daugiau investuoja į šią sritį [6].

Pristatyta net keletas technologijų, kurios skirtingai apdoroja ir manipuliuoja nuotraukomis. Tačiau vis dar atlikta mažai tyrimų, akcentuojant plataus formato nuotraukų apdorojimą naršyklėje, panaudojant skirtingas technologijas.

Kaip yra pastebima, rinka nuolat auga ir šiuo metu reklamuotojai turi įsigyti profesionalią programinę įrangą, kuri nemažai kainuoja ir reikia nemažai pastangų norint išmokyti ja naudotis ir sukurti plataus formato nuotraukas. Tačiau tokioje rinkoje visada atsiranda jaunų įmonių, kurioms ne visada reikia profesionalaus įrankio nuotraukoms sukurti. Todėl yra daroma prielaida, kad naršyklėje veikiantis kūrimo įrankis padėtų tokioms įmonėms sukurti neprofesionalias plataus formato nuotraukas, skirtas spausdinimui, taip sutaupant laiko ir piniginių lėšų.

2.2. Naršyklių programos

Kai 2014 metais pasaulinis saityno konsorciumas (W3C) pristatė HTML5 standartą, jis labai greitai tapo populiarus dėl savo funkcionalumo [7]. Būtent HTML5 pristatytos technologijos praplėtė naršyklių programėlių galimybes [6]. Pagrindinės pristatytos technologijos susijusios su paveiksliukų manipuliavimu:

- HTML5 įprastas dokumentas su CSS3 ir JavaScript. Jis tapo lyginant su ankstesnėmis versijomis ir taip greitesnis – funkcionalumas labiau optimizuotas. Ir taip pat aparatūrinės įrangos pagreitinimo galimybė ypač pagreitina sistemos darbą, kadangi leidžia naudoti GPU galią [8].
- HTML5 „Canvas“ elementas. Šie „Canvas“ elementai gali būti pridėti į puslapį ir naudojant „JavaScript“ ant šių elementų gali būti „piešiami“ paveiksliukai ir įvairios formos. Šioje vietoje kiekvienoje naršyklėje, kuri atitinka HTML5 standartus bus „piešiama“ lygiai taip pat, kadangi „JavaScript“ „pieš“ ant šių elementų. Taip pat manipuliavimas bus toks pats. Formos kris tik ten kur ir programuotojas parašys kode. Pavyzdžiui, žaidime plyta kris būtent ten, kur ji ir turi kristi. Dėl tam tikrų skirtumų naršyklėse, ji nekris ant kairės juostos jei ji turi kristi ant dešinės juostos. Taip pat šis elementas priverčia apmąstyti tam tikrus projektavimo sprendimus. Kaip kad pavyzdžiui, pereinant tarp skirtingų įrenginių – naršyklė automatiškai nepritaikys prie skirtingo dydžio. Programuotojas pats turi tai nuspręsti. Kai kurios įmonės (pvz. „Taopix“) jau pradėjo naudoti šį elementą kurdamos savo naršyklių programas. Tačiau paveiksliukas yra sumažinamas dažniausiai iki 4000px x 4000px dydžio, kuris yra per mažas plataus formato spausdinimui [9].
- SVG (angl. „*Scalable Vector Graphics*“). SVG elementai skiriasi savo valdymu ir pateikimu nuo „Canvas“. Naršyklė pati „piešia“ formas ir paveiksliukus pagal SVG elementą. Kai „JavaScript“ piešia „Canvas“ elemento viduje – SVG turi savo atskirus elementus, kurie yra pateikti dokumente ir naršyklė supranta, kaip „piešti“ ar „perpiešti“ visą langą. Todėl mažinant ir pereinant tarp įrenginių, automatiškai vyksta puslapio pasikeitimai.

- „WebGL“. Susijusi su Canvas technologija ir panašiai piešiama – naudojant „JavaScript“. Tačiau WebGL technologija leidžia pasiekti OpenGL technologiją, kuri leidžia manipuluoti ir nupiešti paveikslukus net ir 3D, panaudojant daugiau GPU galios. Todėl ši technologija labai padidina piešimo greitį.

Kita technologija naudojama naršyklių redaktorių kūrimui yra Flash. Tačiau jau ir pats „Adobe“ kūrėjas pripažįsta, kad ši technologija yra atgyvenusi [10]. Tačiau ji vis dar yra naudojama. Plačiai naudojama „Chili“ dokumentų rengyklė vis dar naudoja šią technologiją [11].

2.2.1. „Canvas“ ir „WebGL“ technologijos

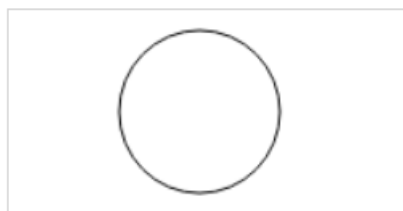
Abiem technologijom piešti grafiką yra pasitelkiama JavaScript programavimo kalba. Į dokumentą yra pridedami Canvas elementai, kurie yra tik grafikos sudėtiniai rodiniai, kuri piešiama pasitelkus JavaScript. Pavyzdžiui, pats elementas („Canvas“), kurį reikia pridėti į HTML5 dokumentą gali atrodyti taip:

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">
</canvas>
```

Pridėję šį kodą, dokumente matysime tuščią stačiakampį. Tačiau norint priešt šiaame elemente, jau reikia naudoti JavaScript. Pavyzdžiui nupiešti apskritimą [12]:

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();
```

Įvykdę kodą, gautume apskritimą, kurį matome 2 pav.



2 pav. Canvas elemente nupieštas apskritimas

Pateiktame pavyzdyje naudojama Canvas technologija. WebGL naudojimo pavyzdys būtų [13]:

```
var gl; // A global variable for the WebGL context

function start() {
  var canvas = document.getElementById('myCanvas');
```

```

// Initialize the GL context
gl = initWebGL(canvas);

// Only continue if WebGL is available and working
if (!gl) {
    return;
}

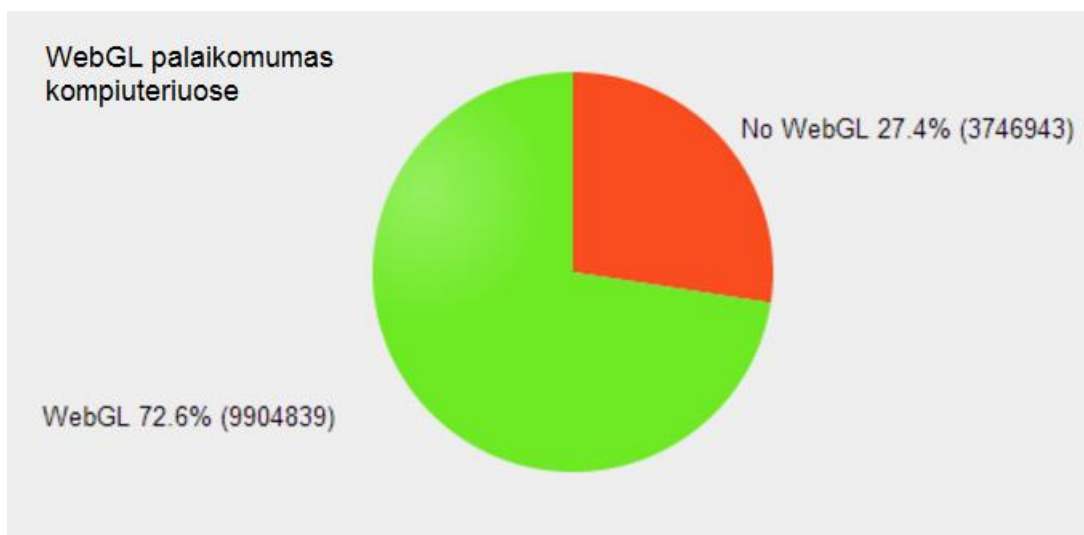
// Set clear color to black, fully opaque
gl.clearColor(0.0, 0.0, 0.0, 1.0);
// Enable depth testing
gl.enable(gl.DEPTH_TEST);
// Near things obscure far things
gl.depthFunc(gl.LEQUAL);
// Clear the color as well as the depth buffer.
gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
}
function initWebGL(canvas) {
    gl = null;

    // Try to grab the standard context. If it fails, fallback to experimental.
    gl = canvas.getContext('webgl') || canvas.getContext('experimental-webgl');

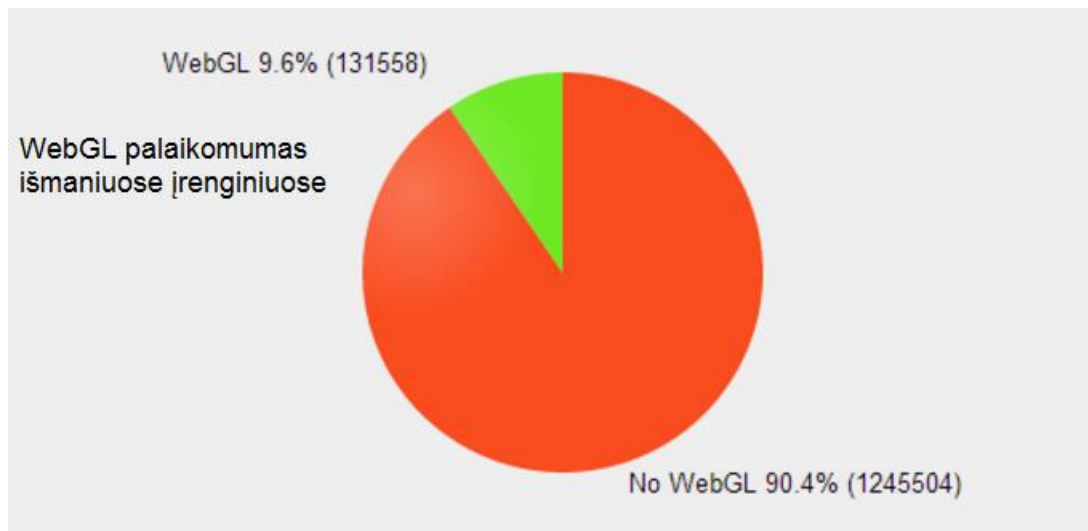
    // If we don't have a GL context, give up now
    if (!gl) {
        alert('Unable to initialize WebGL. Your browser may not support it.');
```

WebGL atveju jis tampa „įjungtas“ ir toliau piešimui rašomas kodas yra panašus ar toks pats kaip ir Canvas atveju.

Canvas technologija yra palaikoma ir mobiliuose įrenginiuose, ir kompiuteriuose [14]. Tačiau WebGL situacija kitokia.



3 pav. WebGL palaikomumas kompiuteriuose



4 pav. WebGL palaikomumas išmaniuose įrenginiuose – mobiliuosiuose telefonuose

Šiuo metu WebGL yra palaikoma 72.6 procentų kompiuterių kaip matome 3 pav. Ir tik 9.6 procentuose mobiliuose įrenginiuose kaip matome 4 pav.

2.2.2. SVG technologija

Ji apibūdina vektorine grafika „XML“ formatu, kuris pateikiamas HTML5 dokumente ir naršyklė kodą atvaizduoja. Pavyzdinis kodas apskritimui:

```

<html>
<body>
<h1>My first SVG</h1>
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-
width="4" fill="yellow" />
</svg>
</body>
</html>

```

Įvykdę pateiktą kodą gausime apskritimą pateiktą 5 pav.

My first SVG



5 pav. Naršyklės pateiktas „SVG“ elemento atvaizdas

2.3. Nagrinėjamos problemos

2.3.1. Pasirinkimas technologijos kuriant nuotraukų redaktorių

Šiuo metu naršyklėje veikiančių nuotraukų rengyklių yra sukurta su HTML5 (div), Canvas, SVG ir Flash technologijomis. Nebuvo rasta naudojančios WebGL technologijos. Kadangi SVG, Canvas ir WebGL technologijos yra palyginus jaunos, todėl yra mažai pavyzdžių ir gairių, kurios padėtų nuspręsti kurią pasirinkti kuriant naršyklės programą.

2.3.2. Plataus formato nuotraukų apdorojimas ir manipuliavimas naršyklėje

Naršyklė veikia kliento kompiuteryje. Todėl spausdinimu užsiimančios įmonės ir kuriančios naršyklėje veikiančią nuotraukų rengyklę vartotojo įkeltas nuotraukas dažniausiai sumažina iki 4000px x 4000px [15]. Su tokiomis matmenimis kokybiškai neišspausdinsi didelio paveiksluko – teks didinti rezoliuciją ir todėl kokybė labai nukentės ir matysis pikseliai [16]. Visgi, tai yra geras sprendimas orientuojantis į platų vartotojų ratą, kadangi paveikslukų apdorojimo greitis ir manipuliavimas ypač priklauso nuo kompiuterio, kuriuo naudojasi vartotojas. Norint padidinti greitį ir dydį paveikslukų galima naudoti WebGL technologiją. Tačiau šiuo metu WebGL palaikomumas kompiuteriuose nėra didelis (72proc.), o išmaniuose telefonuose visai mažas (apytiksliai 9 proc.) kaip matėme 3 pav. ir 4 pav..

Šiuo metu naršyklės padeda manipuliuojant paveikslukais. Jau kiekviena naršyklė turi pasirinkimą naudoti aparatūrinės įrangos pagreitinimą ir šis pasirinkimas pagal nutylėjimą būna įjungtas [17]. Šis pagreitinimas dažniausiai reiškia GPU išteklių naudojimą. Ypač Canvas technologija tokiu būdu pagreitinama. Taip pat ir HTML5 stiliaus aprašymo kalba “CSS3” galima užrašyti stilių taip, kad būtų naudojamas GPU, o ne CPU kiek galima labiau. Pavyzdžiui, užrašant stiliaus kalba, kad vienas judėtų naudodamas GPU, kitas CPU, o kitą panaudojant JQuery (labiausiai naudojamos technologijos) [8]:

```
<div id="gpu" class="box">G</div>
<div id="cpu" class="box">C</div>
<div id="jquery" class="box">J</div>
<script type="text/javascript">
var $box = $('#jquery');

function goToRight() {
    $box.animate({ left:200 }, 3000, 'linear', goToLeft);
}
function goToLeft() {
    $box.animate({ left: 0 }, 3000, 'linear', goToRight);
}
```

```

    goToRight();
  </script>
<style>
.box {
  -webkit-animation: 3s infinite linear alternate;
  background-color: #9cf;
  border: solid 1px #99f;
  box-shadow:
    3px 3px 3px rgba(0,0,0,.2),
    0 0 10px rgba(0,0,0,.2) inset;
  height: 50px;
  line-height: 50px;
  margin: 4px;
  position: relative;
  text-align: center;
  width: 50px;
}

#gpu {
  -webkit-animation-name: move-by-transform;
}
@-webkit-keyframes move-by-transform {
  0% { -webkit-transform:translateX(0) }
  100% { -webkit-transform:translateX(200px) }
}

#cpu {
  -webkit-animation-name: move-by-left;
}
@-webkit-keyframes move-by-left {
  0% { left:0 }
  100% { left:200px; }
}

```

```
}  
</style>
```

Pateiktame kode esminės dalys yra šios:

- `-webkit-animation: 3s infinite linear alternate;` - ši kodo dalis priverčia judėti elementą 3 sekundžių ilgumu, kad vartotojas matytų judėjimą;
- `-webkit-transform:translateX(0)` – „-webkit-transform“ sako, kad elementui gali būti atliekami 2D arba 3D transformacijos. Šios transformacijos naudoja GPU jeigu naršyklėje yra įjungtas aparatūrinės įrangos pagreitinimo pasirinkimas. „Translate“ metodas elementą pastumia į kitą vietą. „translateX“ – x-o ašyje pastumia kiek nurodyta.
- `left:0` – kad šis CSS3 stiliaus aprašas suveiktų reikia, kad elemento aprašyme taip pat būtų nurodytas kitas „position“ aprašas, o ne „static“ (pagal nutylėjimą). Šiuo atveju yra elemente nurodytas „position: relative;“. Vadinasi nuo esamos pozicijos elementas bus patrauktas kiek nurodyta „left“ stiliaus aprašo – šiuo atveju 0px, tačiau keisis iki 200px. Šis aprašas naudoja CPU galią – kadangi tai standartinis aprašas kaip elemento poziciją keisti.
- `$box.animate({ left: 0 }, 3000, 'linear', goToRight);` - jQuery iškviečiama „animate“ funkcija, kuri „linear“ (paslankiai) stiliumi vykdo elemento stiliaus pakeitimą („left“:0), todėl jQuery keičia elemento stilių, o šis tuo tarpu naudoja CPU ir naršyklė parodo pasikeitimus.

Aprašytos technologijos yra plačiai naudojamos ir pakankamai standartinės. Tačiau kaip teigia Jack Doyle apie naršyklinių nuotraukų manipuliavimo technologijas autorius - jQuery biblioteka paveiksliukų manipuliacijų greitį sumažina lyginant su JavaScript kalba [8]. Taip pat lyginant su CSS3 transform (GPU) animacijomis JQuery yra lėtesnis. Tačiau, pasak Jack Doyle, JavaScript pagrindu sukurta GSAP TweenMax paveiksliukų manipuliavimo biblioteka yra greitesnė ir už CSS3 manipuliacijas. Pats autorius nenurodo greičių dėl naršyklių skirtumų ir mini, kad geriau pačiam pasitestuoti pagal tai į ką orientuota sistema.

Plataus formato nuotraukoms greitis yra viena iš esminių dalių, kadangi apdorojimo greitis nulems FPS (angl. “*Frames per second*”) – kadrų per sekundę skaičių, kuris savo ruožtu rodo ar vartotojas matys vėlavimą (angl. “*Lag*”), o vėlavimo reikia išvengti.

Yra manoma, kad sistema veiks tinkamai tik galingesniuose kompiuteriuose. Kompiuteriai nuolat tobulėja ir galingi kompiuteriai tampa vis pigesni. Todėl galima daryti prielaidą, kad sukurta sistema kuo toliau tuo labiau taps prieinama didesniai vartotojų daliai.

2.4. Darbo tikslas

Išsiaiškinti su kokiais paveikslukų dydžiais naršyklės programos yra pajėgios dirbti. Kurios naršyklinės technologijos tam yra tinkamiausios, jų stiprybes ir silpnybes. Taip pat naudojamo kompiuterio įtaką paveikslukų apdorojimo greičiams.

2.5. Uždaviniai

Ištirti technologijas, kurios geba manipuluoti paveikslukais naršyklėje:

- WebGL
- SVG
- Canvas
- HTML5 div

Iš šių perspektyvų:

- FPS greitis
- CPU panaudojimas
- GPU panaudojimas
- Vėlavimas keičiant dydį

Keičiant:

- Nuotraukų dydžius
- Kompiuterį su kuriuo daromas tyrimas

2.6. Laukiami rezultatai

Atlikus tyrimą su Canvas, WebGL, SVG ir HTML5 (div) technologijomis - pamatyti jų rodikliai ir išsiaiškinta technologija, kuri leidžia tinkamiausiai dirbti su plataus formato nuotraukomis.

2.7. Naujumas

Nebuvo rastas tyrimas ar rengyklė, kuris naršyklėje orientuotųsi į plataus formato nuotraukų manipuliavimą siekiant gautą variantą spausdinti. Ir taip pat apytiksliai gairių nusakančių kokie dydžiai apriboja naršyklių pajėgumus.

2.8. Esami sprendimai

Šiuo metu yra nemažai sukurtų skirtingų naršyklių rengyklių skirtų nuotraukų manipuliavimui. Buvo išrinkti populiareesnės rengyklės su skirtingomis technologijomis, kurie nagrinėjami atskiruose skyriuose. Rasti sprendimai skirstomi į dvi kategorijas:

- Įmonės kuriančios programinę įrangą tik pardavimui.

- Įmonės turinčios savo sukurtą rengyklę ir jį yra pritaikiusios savo reikmėms

2.8.1. Įmonės kuriančios programinę įrangą tik pardavimui

Nebuvo rasta daug tokių. Kadangi rinka tik dar auga tokių rengyklių nėra daug.

2.8.1.1. Taopix

Sukurta 2007 ir šiuo metu „Taopix“ pardavinėja savo sukurtą naršyklėje veikiančią nuotraukų manipuliavimo programinę įrangą, kuria naudojasi daugiau kaip 50 šalių [18].

Programa sukurta su Canvas technologija. Sistema orientuojasi į fotoknygų, kalendorių, sveikinimų kortelių ir ant sienos kabinamų nuotraukų (angl. “wall art”) kūrimą. Produkto aprašyme nebuvo rasti spaudinių dydžiai. Tačiau iš to kas aprašyta ir į ką orientuojasi sistema – galima daryti prielaidą, kad spaudinių dydžiai nėra skirti plataus formato spaudiniams.

Sukurta Jungtinėje Karalystėje. Kaina varijuoja nuo: £11,750 iki £40,000 priklauso nuo to kiek svetainių užsakoma ir koks funkcionalumas parenkamas. Taip pat apmokėjimus pritaiko jeigu perkama už Jungtinės Karalystės ribų. Siūlo iki galo paruoštą produktą (angl. „End-to-end“), kuriuo iškart naudotis galima.

Taopix privalumai:

- Canvas technologija;
- Nuo – iki produktas su e-parduotuve.

Taopix trūkumai:

- Kaina;
- Pritaikyti tik kai kuriems formatams – fotoknygos, kalendorių, sveikinimų.

2.8.1.2. Chili

Sukurta 2010 metais Belgijoje. Chili vadinama dokumentų rengykle naršyklėje, kadangi ši sistema gali būti plačiai pritaikyta naudoti [11]. Tačiau ji reikalauja integravimo ir įrašymo į savo kompiuterius ar įmonės užsakymo, kuri padarys tai. Turi tik rengyklę nuotraukoms ir nuo kliento priklauso kaip jis bus panaudotas – pardavimams, savo reikmėms ar kitaip.

Sistema sukurta su Flash. Yra dirbama ties HTML5 versija, tačiau ji nėra pilnai baigta. Nėra nurodyti dydžiai, tačiau Flash sistema dėl savo lėtumo labai apriboja plataus formato paveiksliukų manipuliavimą.

Kaina - £35,000 vienkartinis mokestis. Tačiau 20 proc. kiekvienais metais reikia sumokėti dėl kasmetinio palaikymo.

Chili privalumai:

- Platus pritaikymas įvairiems formatams.

Chili trūkumai:

- Kaina

- Naudojama Flash technologija
- Tik dokumento redagavimo sistema – nėra e-parduotuvės integruotos

2.8.1.3. Pixami photo site

Sukurta 2000-aisiais Amerikoje. Jie taip pat kaip ir Taopix siūlo sistemą nuo – iki. E-parduotuvę, kurioje integruota naršyklės programa skirta kurti ir redaguoti nuotraukas (nenurodoma technologija, tačiau sakoma HTML5). Minima, kad galima kurti fotoknygas, korteles, kalendorius, puodukų dizainius ir kitka [19]. Siūlo savo serveriuose laikomą arba licenzijuotą, tačiau tik “dideliems” vartotojams skirtą programinę įrangą.

Kaina - £5000 – £13000 vienkartinis mokestis ir plus 5 – 7 proc. metinis mokestis nuo pardavimų skaičiaus.

Pixami privalumai:

- HTML5 technologija
- Palyginti nedidelė kaina
- Nuo – iki su e-parduotuve.

Pixami trūkumai:

- Nėra pavyzdžių, detalių aprašymų kokia sistema

2.8.1.4. Bendras palyginimas produktų

1 lentelė. Naršyklėje veikiančių rengyklų skirtų tik pardavimui palyginimas

Palyginimo kriterijai	Taopix	Chili	Pixami	Kuriama
Kaina	£11,750 iki £40,000	£35,000 vienkartinis plus 20proc. kasmetinis	£5000 – £13000 vienkartinis mokestis ir plus 5 – 7 proc. metinis mokestis nuo pardavimų	Nemokama
Rengyklės technologija	HTML5 Canvas	Flash	HTML5 (konkrečiai nenurodyta kuri)	Pagal tyrimo rezultatus. Numanoma WebGL.
Kliento kodas	Taopix serveryje	Lokaliai kliento serveryje	Pixami serveryje. “Dideliem” klientam – jų serveriuose	Lokaliai kliento
Ar viskas viename?	+ su e-parduotuve	- tik dokumentų	+ su e-parduotuve	- tik rengykl

Palyginimo kriterijai	Taopix	Chili	Pixami	Kuriama
		redaktorius		ė
Dydžio pasirinkimai	Pritaikyta fotoknygai, kalendoriui, kitiems formatams. Apie platų formatą nerašoma	Neribojama. Flash riboja.	Pritaikyta fotoknygai, kalendoriui, kitiems formatams. Apie platų formatą nerašoma	Neribojama. Ribojama apdorojimas ir manipuliavimas paveiksluko atsižvelgiant į WebGL [20]. Canvas, SVG technologijų apribojimus.
Turi pavyzdžių, aprašymų	+	+	-	-

Chili yra brangiausias ir atsilieka nuo konkurentų technologijų naudojimu, tačiau laimi ties savo rengyklės funkcionalumu kaip yra matoma 1 lentelėje. Taopix yra pakankamai pigus, turi gerą aprašymą ir yra su e-parduotuve, todėl iš šių variantų šis variantas vienas geriausių. Tačiau kuriamas ir siūlomas variantas ir eksperimentas su WebGL – literatūroje ši technologija jau plačiai aprašyta ir taipogi jos apribojimai. Todėl yra daroma prielaida, kad ji jau pakankamai subrendusi tokiom užduotim.

2.8.2. Įmonės turinčios savo sukurtą rengyklę ir ją yra pritaikiusios savo reikmėms

Tokių įmonių yra nemažai. Net Lietuvos rinkoje yra tokių rengyklių [21]. Tačiau visos funkcionalumu panašios, todėl išrinktos naršyklių programos, kurios skiriasi technologijomis panaudotomis jas kuriant.

2.8.2.1. Shutterfly

1999 metais įmonė pradėjo savo veiklą teikdama fotoknygos kūrimo ir spausdinimo paslaugas ir toliau plėtėsi [15]. Siūlo plataus formato nuotraukų spausdinimą šiais dydžiais: 12 x 36 ", 16 x 20 ", 20 x 30 ".

Naudoja HTML5 Canvas pagrindu sukurtą naršyklės rengyklę. Turi daug įvairių funkcijų – maketo pasirinkimą, fono spalvos keitimą, importavimą nuotraukų iš socialinių tinklų ir kt.

Shutterfly privalumai:

- HTML5 Canvas nuotraukų rengyklė.
- Didelis dydžių pasirinkimas.

Shutterfly trūkumai:

- Spaudiniai orientuoti į platų vartotojų ratą ir tik keli plataus formato spaudiniai.

2.8.2.2.MixBook

2006 įkurta ši įmonė šiuo metu siūlo susikurti ir įsigyti fotoknygas, foto korteles, kalendorius, metų knygas, kompanijų knygas ir kitus spaudinius. Duoda platų maketų pasirinkimą, fono, lipdukų. Tačiau naudoja Flash pagrindu sukurtą rengyklę nuotraukoms redaguoti. Plataus formato nuotraukas siūlo: 14 x 11 ", 16 x 20 ", 30 x 20 ", 36 x 24 ".

MixBook privalumai:

- Platus spaudinių pasirinkimas.

MixBook trūkumai:

- Flash technologija.
- Spaudiniai orientuoti į platų vartotojų ratą ir tik keli plataus formato spaudiniai.

2.8.2.3.Snapfish

Turi daugiau kaip 15 metų patirtį spausdinimo rinkoje, vartotojams siūlo foto knygas, foto korteles, kitus spaudinius [22]. Taip pat standartiškai siūlo maketus, fono spalvas, lipdukus. Taip pat vartotojams siūlo savo naršyklės programą skirtą redaguoti nuotraukas. Programa sukurta su SVG ir veikia pakankamai greitai. Siūlo vartotojams plataus formato spaudinius: 11 x 14 ", 12 x 18 ", 16 x 20 ", 20 x 30 ".

Snapfish privalumai:

- SVG technologija panaudota;
- Plataus formato spaudinių pasirinkimas.

Snapfish trūkumai:

- Spaudiniai orientuoti į platų vartotojų ratą ir tik keli plataus formato spaudiniai.

2.8.2.4.Bendras palyginimas produktų

2 lentelė. Įmonės turinčios savo sukurtą rengyklę ir ją yra pritaikiusios savo reikmėms palyginimas

Palyginimo kriterijai	Shutterfly	MixBook	Snapfish
Rengyklės technologija	Canvas	Flash	SVG
Plataus formatų pasirinkimas	11 x 14 ", 16 x 20 ", 20 x 30 ".	14 x 11 ", 16 x 20 ", 30 x 20 ", 36 x 24 ".	11 x 14 ", 12 x 18 ", 16 x 20 ", 20 x 30 ".

Palyginus produktus matome, kad šiuo metu plataus formato spaudinius siūlo visi analizuoti produktai. Tačiau šie dydžiai pritaikyti plataus vartotojų ratui, todėl reklamuotojai taip plačiai jais nesinaudoja. Kiek buvo ieškota naršyklių rengyklių skirtų nuotraukų manipuliavimui ir jų pardavimui tai didžioji dalis sukurta su Flash ir Canvas technologijomis. Beveik visos įmonės kuria

savo programinę įrangą su Canvas technologija. Tai pakankamai suprantama, kadangi Canvas technologija ypač paveikia aparatūros pagreitinimo įjungimas ir yra labai geras suderinamumas tarp naršyklių.

2.9. Siūlomas metodas tyrimui

Norint sukurti naršyklės rengyklę orientuotą į plataus formato nuotraukų kūrimą pirmiausia reikia išsiaiškinti kokias naršyklė teikia galimybes tam. Kaip buvo minėta prieš tai esančiuose skyreliuose naujasis HTML5 standartas pristatė technologijas tinkamas tam – Canvas, WebGL, SVG ir HTML5 įprastinis. WebGL ypač teikia daug vilčių, kadangi su OpenGL susieja, kuri leidžia naudoti GPU galią. Tačiau čia iškyla problema, kokiais būdais nustatyti, kuri technologija tinkamiausia tam ir iš kokių pusių. Tam bus nagrinėjami keli parametrai: FPS, CPU, GPU. Šie parametrai nagrinėjami detaliau atskirame skyrelyje.

2.9.1. FPS, CPU, GPU parametrai

FPS (angl. „*Frame rates per second*“) – kadrų skaičius per sekundę. Bendrai, norint išvengti trūkčiojančio judesio (angl. „*jerky motion*“) rekomenduojamas minimalus FPS dydis yra apie 30. Turinyje, kuriame keičiasi greit vaizdai geriausia yra 60 FPS [23]. Saityno našume, kadras parodo darbą, kurį naršyklė turi padaryti, kad atnaujintų ir perpieštų ekraną [24]. Naršyklėje FPS nuolat keičiasi ir ji nebūna visada 60 FPS, tačiau šiuo atveju svarbu, kad manipuliuojant nuotraukomis nebūtų FPS mažesnis kaip 30 dėl galimų trūkčiojimų. Žinoma, trūkčiojimų išvengti visiškai nepavyks dėl naršyklės vidinių problemų.

Bendra taisyklė yra ta, kad kuo didesnis kadrų skaičius per sekundę tuo didesnis CPU panaudojimas. Iš to išeina, kad perkeliant piešimo darbą į GPU bus sutaupomas CPU ir tokiu būdu FPS bus aukštesnis.

CPU (angl. „*Central processing unit*“) – centrinis apdorojimo blokas. Šis blokas apdoroja visas instrukcijas, kurias gauna iš techninės įrangos ir programinės veikiančios kompiuteryje. Ši dalis vadinama kompiuterio „smegenimis“, kuri daro visus apskaičiavimus [25]. Kaip ir minėta, nuo CPU galingumo ir greičio priklauso kaip greit bus įvykdytos užduotys ir vartotojui parodomas rezultatas.

GPU (angl. „*Graphical processing unit*“) – grafinis apdorojimo blokas. Šis blokas naudojamas pagreitininti 2D ir 3D paveiksliukų kūrimą [26]. GPU gali būti integruotas į kompiuterį, kas reiškia jie yra pridėti į CPU ar į pagrindinę plokštę ir naudoja CPU galią. Taip pat jie gali būti paskirti (angl. „*dedicated*“) kas reiškia, kad yra atskira techninės įrangos dalis vadinama vaizdo korta (angl. „*video card*“). Turinti atskirą procesorių (vaizdo kortą), GPU leidžia kompiuterio CPU resursus panaudoti kitiems svarbiems darbams ir pagreitinį apdorojimą.

2.10. Išvados

1. Šiuo metu didėja skaitmeninių paslaugų pardavimai ir rinka - 2018 metais manoma, kad pasieks 187.7 milijardo dolerių. Tačiau įmonės siūliančios plataus formato nuotraukų spausdinimą vis dar siūlo tik įkelti nuotraukų failą - PDF ar kitu formatu. Kitos įmonės siūlo ir naršyklės programą, kuri leidžia redaguoti/keisti nuotraukas taip pasiekdamos didesnę vartotojų ratą, tačiau yra apribojusios nuotraukų dydžius. Todėl daroma prielaida, kad tokioje didelėje rinkoje plataus formato nuotraukų naršyklės rengyklė padės pasiekti didesnę vartotojų ir suteikti jiems galimybę greičiau ir pigiau susikurti plataus formato nuotraukas.
2. HTML5 pristatytos technologijos praplėtė galimybes naršyklių rengyklėms. Todėl didžioji dalis įmonių pradėjo kurti programas sukurtas su Canvas technologija dėl suderinamumo tarp naršyklių ir aparatūrinės įrangos pagreitinimo. Vis dėlto, WebGL technologija teikia daugiau vilčių dėl savo GPU panaudojimo.
3. Pagrindinės nagrinėjamos problemos yra technologijos pasirinkimas kuriant naršyklėje veikiančią nuotraukų rengyklę orientuotą į spaudinius ir plataus formato nuotraukų apdorojimą ir manipuliavimą tokioje programoje.
4. Įmonės kuriančios programinę įrangą pardavimui yra užsidėjusios pakankamai aukštas kainas. Vis dėlto, tokia programinė įranga neša nemažai pelno ir žmonės vis daugiau ja naudojami. Tačiau didžioji dalis yra pritaikiusios programinę įrangą tik tam tikrai sričiai formatų - fotoknygoms, atvirutėms, metų knygoms, kalendoriams. Tačiau dideliai reklamuotojų daliai mažai skiria dėmesio.
5. Įmonės turinčios savo programinę įrangą ir ja naudojami savo reikmėms jau teikia keletą plataus formato spaudinių. Vis dėlto, tie formatų pasirinkimas pritaikytas plačiai vartotojų daliai, o reklamuotojai dažnai naudojami specifiniais formatais, kurie nėra siūlomi.
6. Tyrimui atlikti buvo pasirinkta patikrinti naujų naršyklių technologijų savybes iš FPS, CPU ir GPU panaudojimo perspektyvos.

3. PROJEKTINĖ DALIS

3.1. Sistemos paskirtis

Šios sistemos pagrindinis tikslas yra suteikti galimybę susikurti plataus formato nuotrauką panaudojant naršyklės rengyklę. Šiuo įrankiu galėtų naudotis vartotojai, kurie norėtų susikurti nuotraukas skirtas spausdinimui dideliu formatu. Tokiu būdu būtų sutaupyta laikas, kurį reikia įrašant programinę įrangą su kuria galima kurti plataus formato nuotraukas. Taip pat būtų sutaupyta lėšos, kadangi plataus formato nuotraukoms kurti programinė įranga dažnai būna nepigi ir skirta profesionalams. Tačiau nevisada reikia profesionalaus įrankio tam. Šis įrankis ir būtų neprofesionalams skirtas.

Kiti šio projekto tikslai yra išnagrinėti paveikslukų manipuliavimo technologijas pagal tam tikrus parametrus (FPS, CPU, GPU) bei nuspręsti, kurią realizuoti įgyvendinamoje sistemoje.

3.2. Sistemos funkcijos ir reikalavimai sistemai

Sistemos funkcijos:

- Albumo sukūrimas – spaudinių failų laikykla.
- Kurti nuotraukas pasinaudojant nuotraukų rengykle.
- Kainos apskaičiavimas.
- Užsakymas.

Funkciniai reikalavimai sistemai:

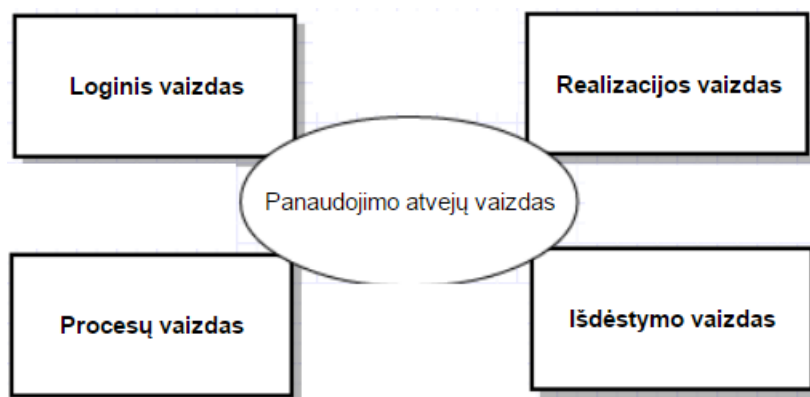
- Leidžiama pasirinkti dokumento dydį.
- Leidžiama pridėti/trinti/pasirinkti puslapį.
- Leidžiama iš anksto sukurtą maketą pasirinkti.
- Leidžiama eksportuoti į PDF formatą.
- Leidžiama pridėti/trinti/panaudoti/manipuliuoti paveiksluku.
- Leidžiama pridėti/keisti/ tekstą.
- Apskaičiuoti kainą jeigu yra daugiau kaip 8 puslapiai.
- Vartotojui parodomi informaciniai pranešimai apie klaidas.

Nefunkciniai reikalavimai sistemai:

- Įrankiui taikoma GNU GPL licenzija.
- Kuriant įrankį atsižvelgti į praplėtimo galimybes.
- Įrankis turi veikti įvairiose OS.

3.3. Architektūra

Šiame dokumente sistemos architektūra pateikiama šiais vaizdais: panaudojimu atvejų, procesų vaizdu, dinaminiu ir išdėstymo vaizdu. Šių vaizdų išdėstymas pateikiamas 6 pav. Patys vaizdai kurti panaudojus unifikuoatą modeliavimo kalbą (UML).



6 pav. Bendras architektūros pateikimo vaizdas

- Panaudojimo atvejų vaizdas – panaudos atvejų diagrama
- Loginis vaizdas – paketai ir klasių diagramos
- Procesų vaizdas – būsenos ir veiklų diagramos
- Išdėstymo vaizdas – išdėstymo diagrama

3.3.1. Architektūros tikslai ir apribojimai

Reikalavimai ir apribojimai, kurie įtakoja architektūrinius sprendimus:

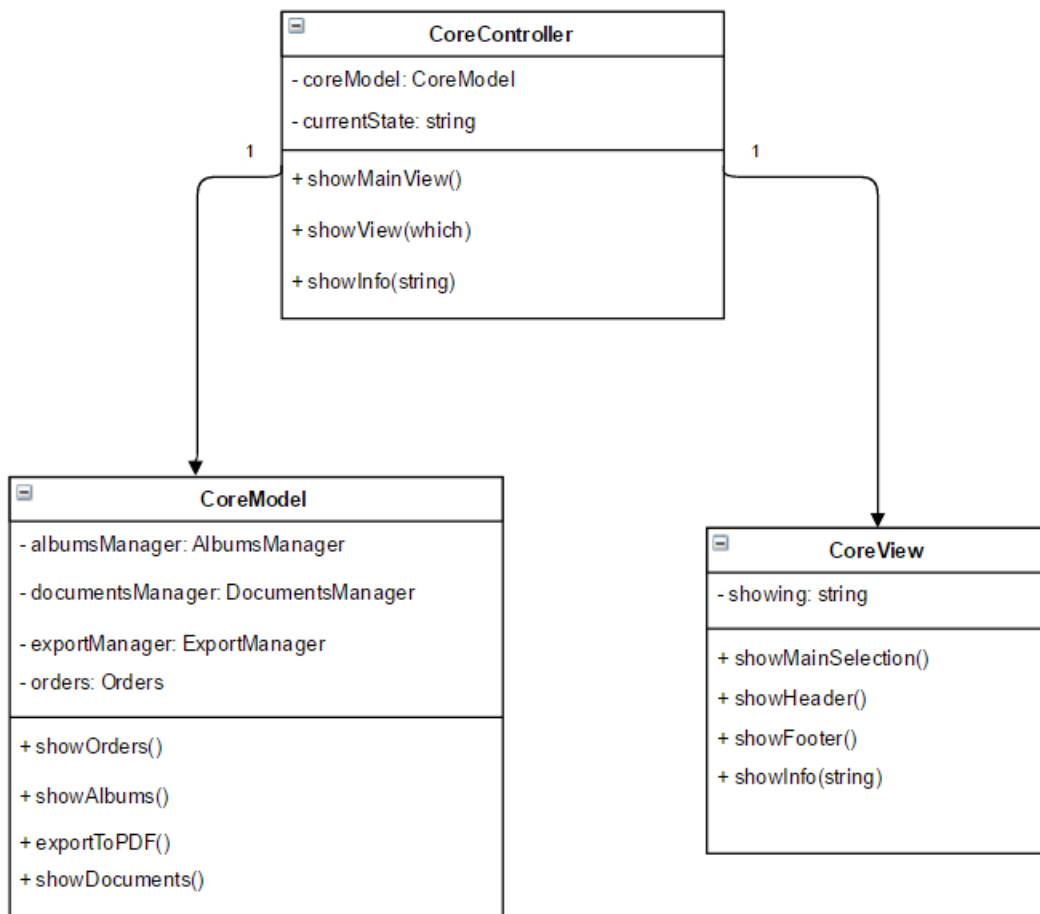
- Kuriant įrankį atsižvelgti į praplėtimo galimybes.
- Įrankis turi veikti įvairiose OS.

3.3.2. Panaudojimo atvejų vaizdas

Kuriamo įrankio panaudojimo atvejai pateikti 7 pav.

MainCore paketas

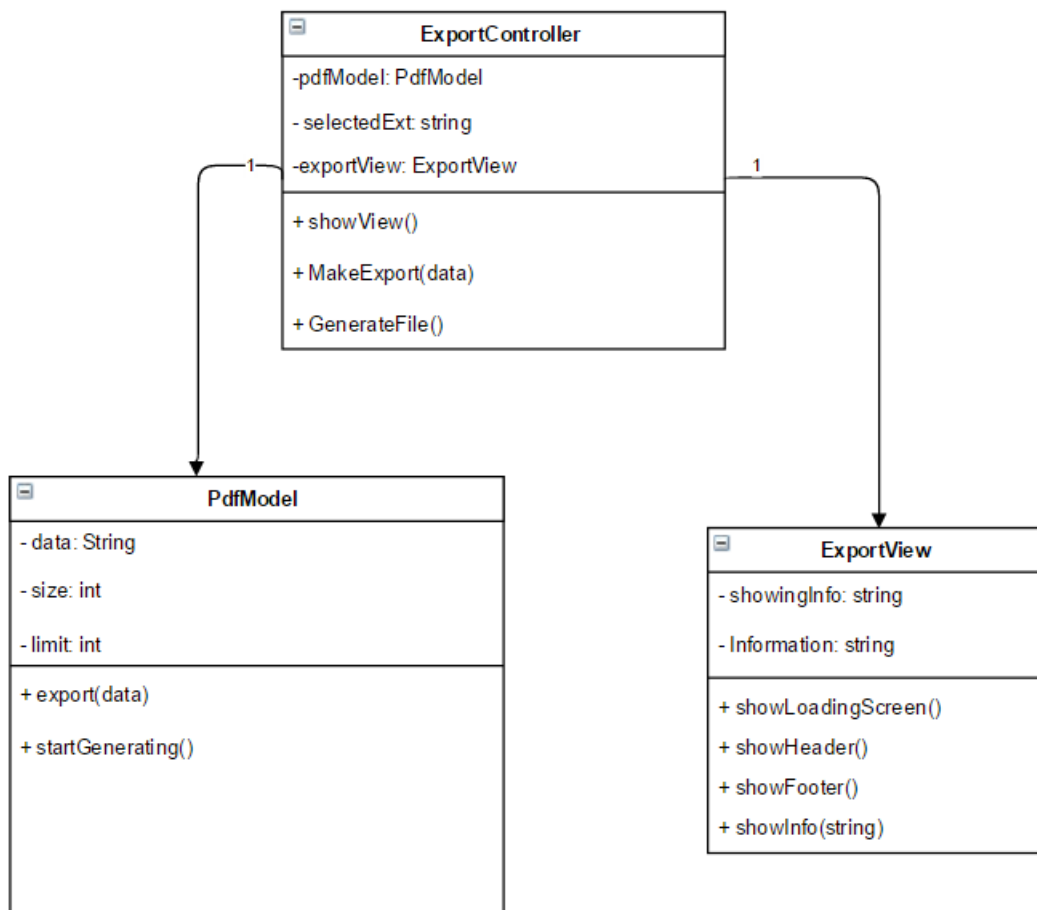
Šis paketas yra pagrindinis, kuris valdys ir kvies kitų paketų metodus ir funkcijas. Šio paketo klasių diagrama pateikiama 9 pav.



9 pav. MainCore paketo klasių diagrama

ExportManager paketas

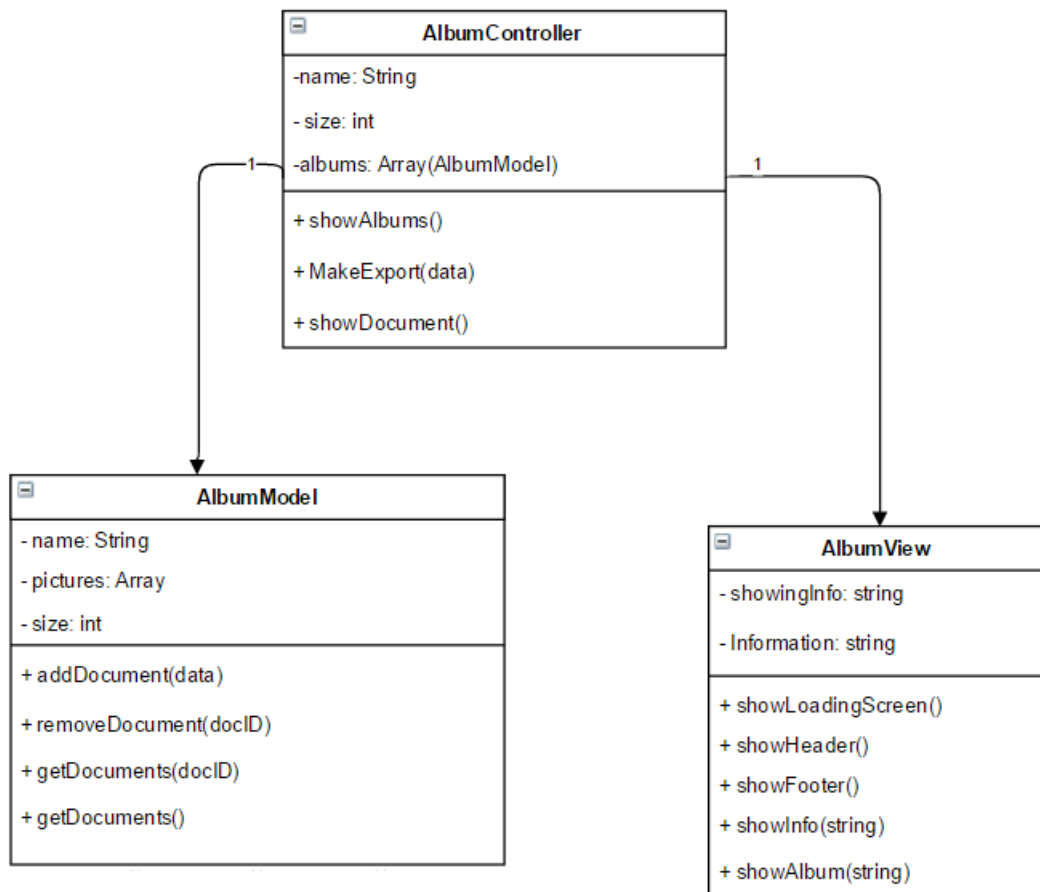
Šis paketas yra atsakingas už eksportavimą failų. Jame naudojama MVC modelis, kad būtų lengva pridėti daugiau eskportavimo formatų. Šiuo metu pridėtas tik vienas eksportavimo modelis – PDF. Ši diagrama pateikiama 10 pav.



10 pav. ExportManager paketo klasių diagrama

AlbumsManager paketas

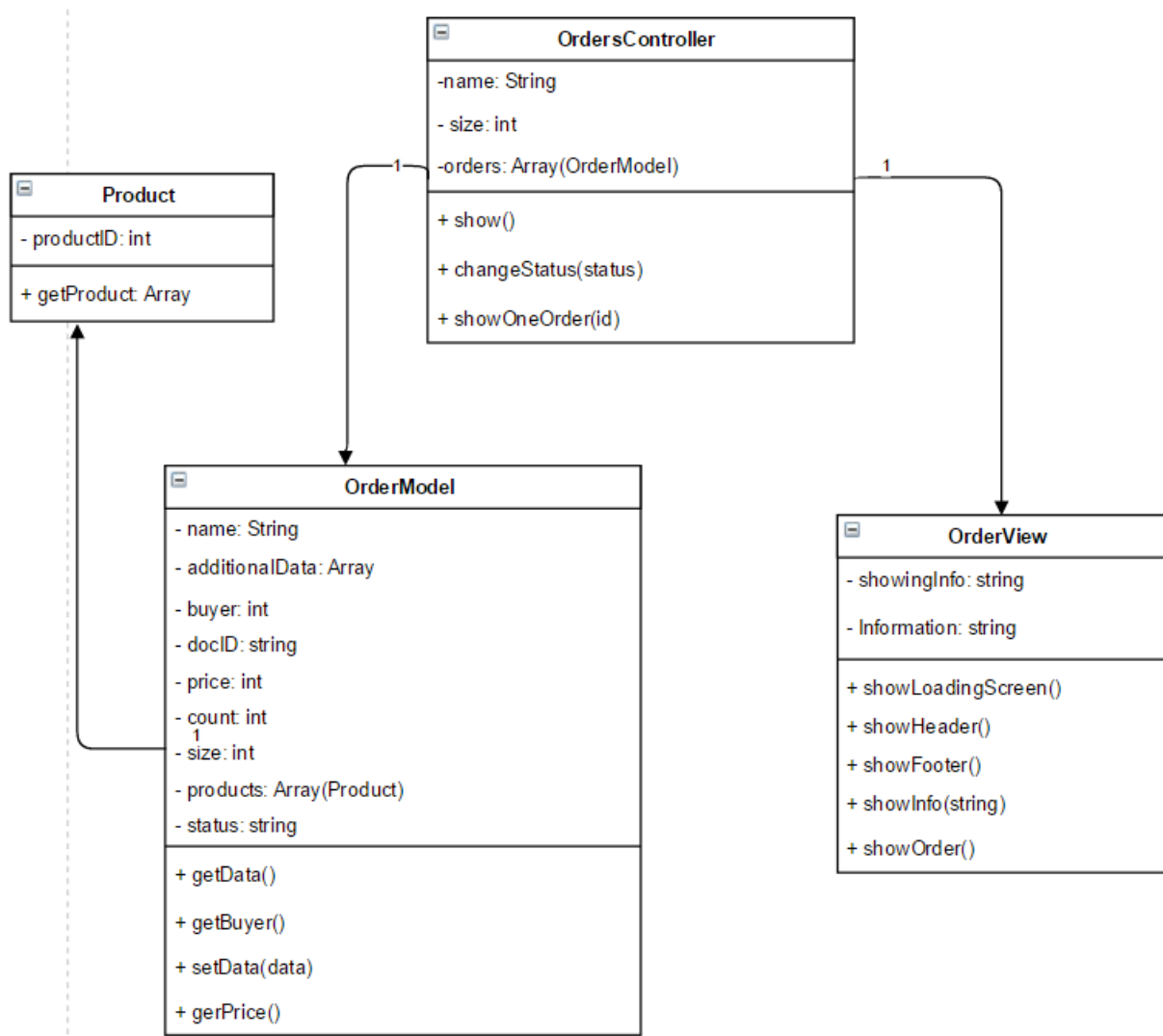
Šis paketas atsakingas už albumų rodymą ir manipuliavimą jais. Šio paketo pagalba yra pridedami nauji albumai ir dokumentai į juos. Ši diagrama pateikiama 11 pav.



11 pav. AlbumManager paketo klasių diagrama

Orders paketas

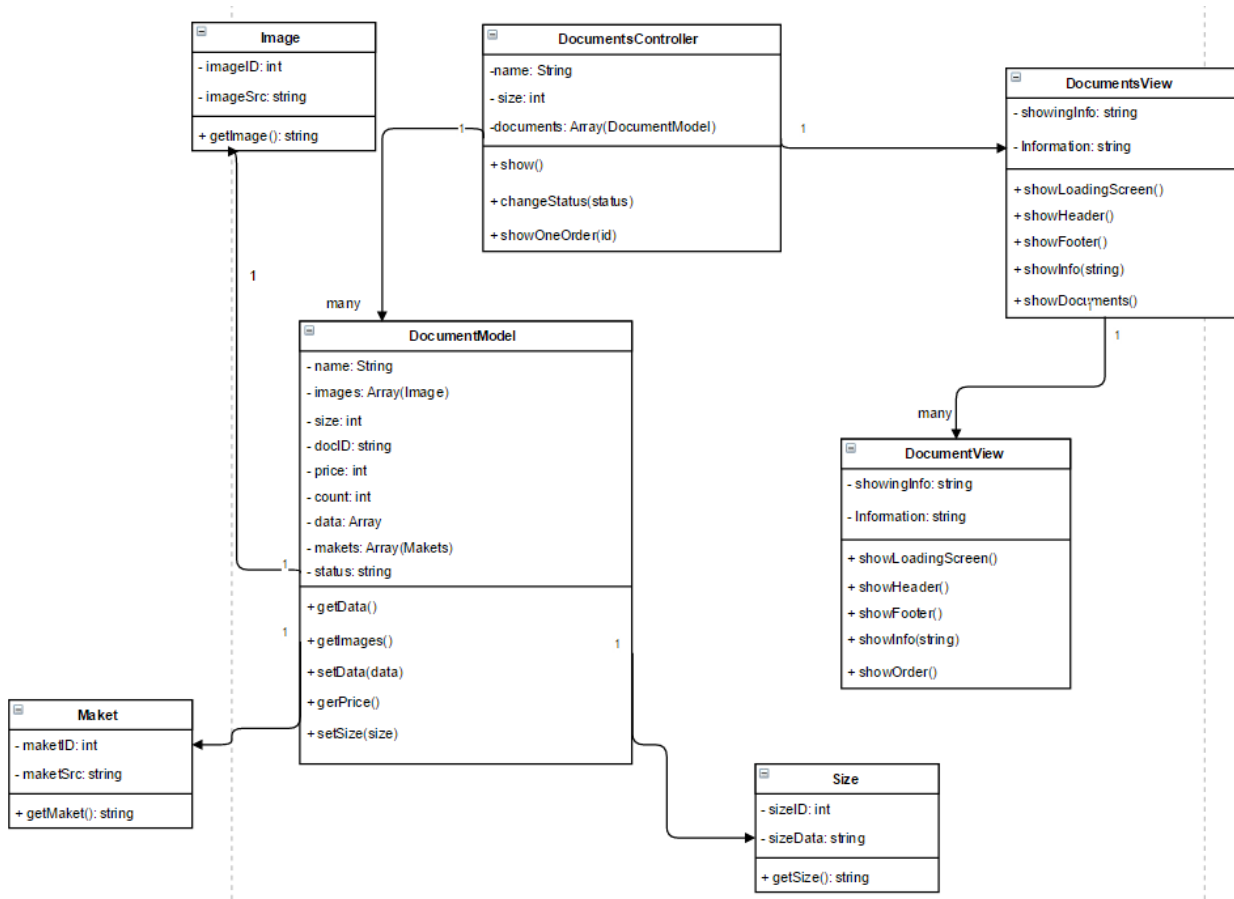
Šis paketas skirtas užsakymų valdymui – sukūrimui, atvaizdavimui ir manipuliacijoms su jomis. Preliminari klasių diagrama pavaizduota 12 pav.



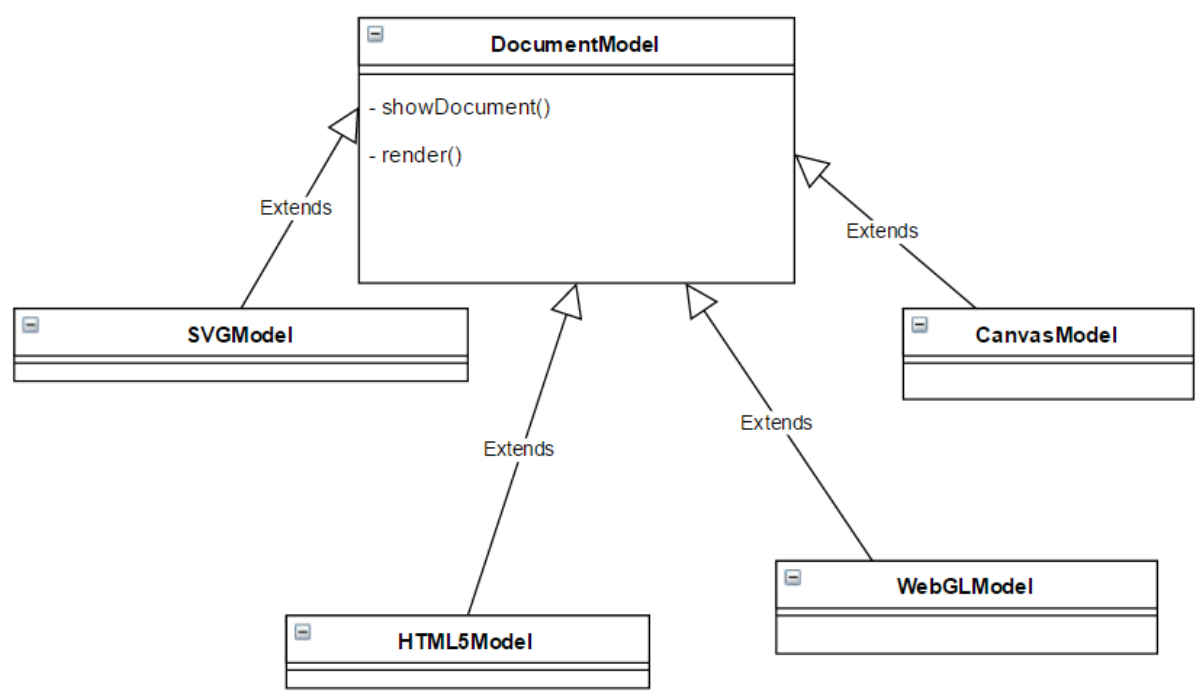
12 pav. Orders paketo klasių diagrama

DocumentsManager paketas

Šis paketas skirtas dokumentams kurti, jais manipuliuoti ir po truputį plėsti jo funkcionalumą. Taip pat paskirti maketus, leisti pasirinkti dokumento dydį, pridėti paveikslukų į dokumentą. Preliminari klasių diagrama pavaizduota 13 pav. Taip pat DocumentModel praplečią skirtingos technologijos kaip kad matoma 14 pav.



13 pav. DocumentsManager paketo klasių diagrama



14 pav. DocumentsManager paketo – DocumentModel paveldėjimo schema

- Atsižvelgiant į tai, kad sistema sukurta naršyklėje – vartotojas iškart pamatys atnaujinimus.
- Pernešamumas - šiuo metu naudojama architektūra leidžia pernešti sistemą į kitas sistemas. Naudojama “NodeJS” technologija, kuri veikia įvairiose operacinėse sistemose.
- Programos patikimumą padidina tai, kad žemo lygio duomenų apdorojimą atlieka duomenų bazių valdymo sistema. Dėl to supaprastėja programos realizacija ir padidėja patikimumas.

3.4. Sistemos testavimas

Šiame skyriuje pateikiamas apibendrintas testavimo planas, aprašomos testavimo procedūros ir pateikiami testavimo rezultatai.

3.4.1. Testavimo planas

Testavimo planas yra skirtas aprašyti testavimo strategiją, bei testavimo metodus, kurie bus naudojami testuoti programinę įrangą. Metodai aprašomi abstrakčiai, nepateikiamos detalios metodų realizavimo procedūros. Metodų realizavimo procedūros yra pateikiamos skyriuje „Testavimo procedūra“.

3.4.1.1. Testuojama programinė įranga

Testuojama plataus formato kūrimo programinė įranga, kurios pagrindinės funkcijos yra pasinaudojus naršyklės rengykle yra sukurti PDF formato nuotraukų failą ir leisti pateikti užsakymą dėl šio failo spausdinimo.

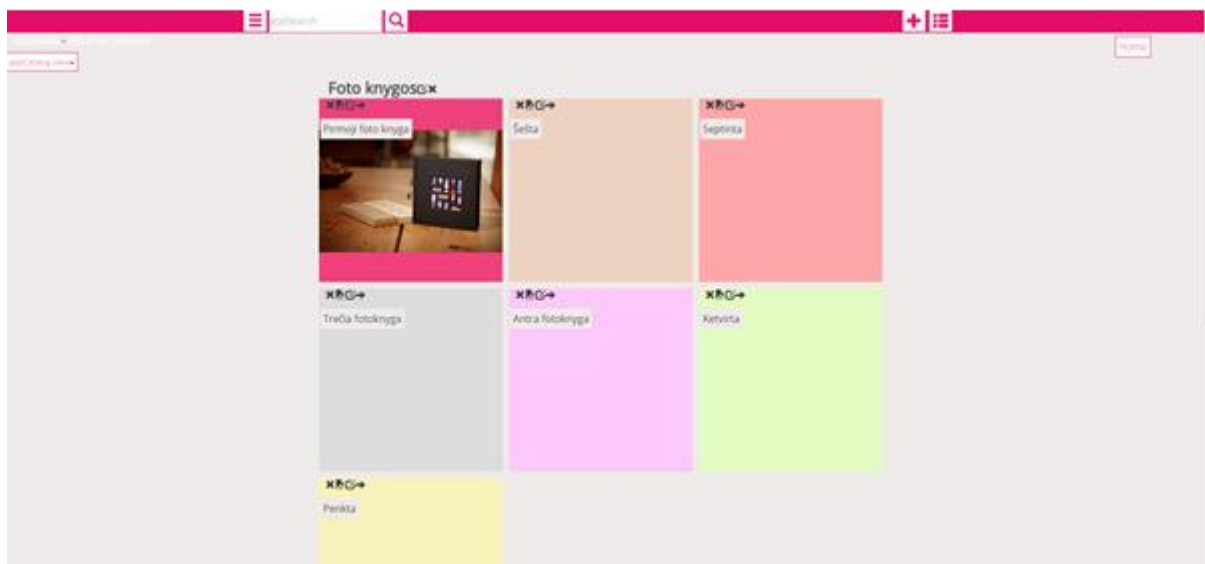
3.4.1.2. Sąsajos

Sąsajos testavimas buvo atliekamas dviem būdais:

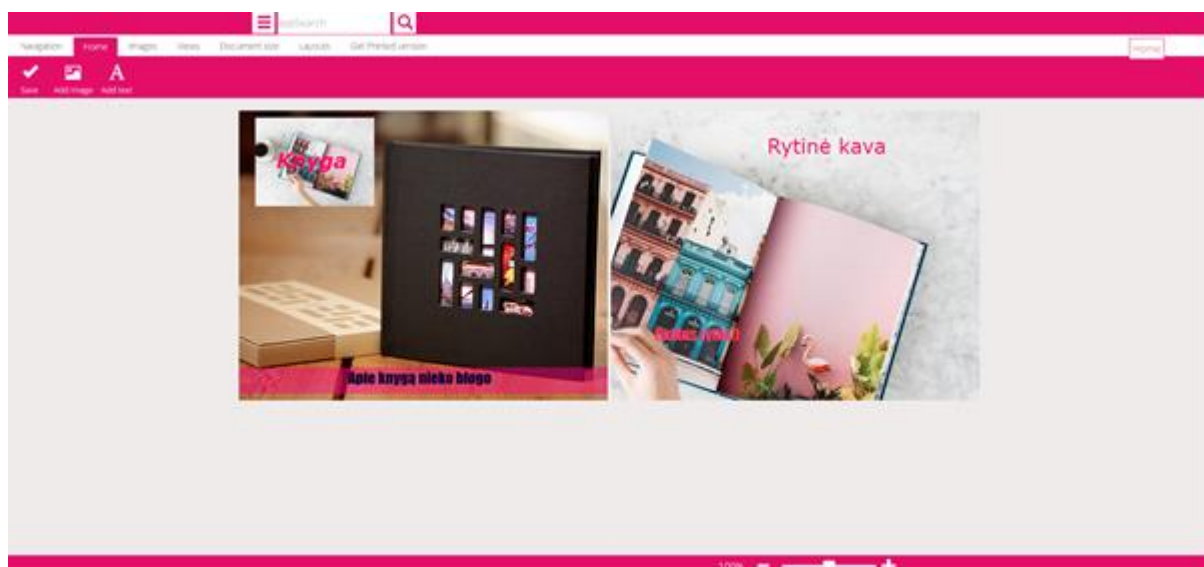
1. Testuotojas pagal sąsajos testavimo scenarijus testuos programą ir rezultatus žymės lentelėse.

Testuotojas mato pagrindinius sistemos langus:

- Albumo langą – rodomi dokumentai ir albumai - 16 pav.
- Dokumento kūrimo langą – matome dokumentą ir jo keitimo galimybes - 17 pav.
- Užsakymo langas – matome formą su tam tikrais pasirinkimais ir įkeltu PDF failu - 18 pav.



16 pav. Pagrindinis sistemos langas – rodomi albumai, dokumentai



17 pav. Dokumento kūrimo langas – matoma navigacija, dokumentas

Select product (pdf)
 Download PDF - Users photobook
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
 Upload other PDF file

Page number
 16

Physical / feed paper
 250 g. Standard cover sheet

Cover stock
 250 g. Standard cover sheet

Select the desired binding
 Fals (folding, min 8 pages)

Protective Film on Cover
 Blank pp-foil

Document size
 14x11

Number	Price	Shipping	VAT	Total
1	120.00	1.12	0.00	29.60
11	1320.00	12.32	0.00	325.60
26	3120.00	29.12	0.00	769.60
51	6120.00	57.12	0.00	1509.60
76	9120.00	85.12	0.00	2249.60
101	12120.00	113.12	0.00	2989.60
151	18120.00	169.12	0.00	4469.60
201	24120.00	225.12	0.00	5949.60
251	30120.00	281.12	0.00	7429.60
301	36120.00	337.12	0.00	8909.60
351	42120.00	393.12	0.00	10389.60
451	54120.00	505.12	0.00	13349.60
501	60120.00	561.12	0.00	14829.60
601	72120.00	673.12	0.00	17789.60
701	84120.00	785.12	0.00	20749.60
801	96120.00	897.12	0.00	23709.60
901	108120.00	1009.12	0.00	26669.60
1001	120120.00	1121.12	0.00	29629.60

Select shipping
 Sent no later than after 24 hours
 Sent no later than after 48 hours
 Recalling the shop - 24 hour
 Recalling the shop - 48 hours

Total sum	
Price	120.00
Shipping	1.12
VAT	0.00
Total	29.60

Add to cart

18 pav. Užsakymo forma

3 lentelė. Dokumento kūrimo testavimo rezultatai

Testas	Laukiamas rezultatas	Rezultatas
Pasirenkamas pridėti naują dokumentą	Išskviečiama kūrimo forma	Pavyko
Nesuvedami duomenys ir spaudžiama “kurti dokumentą”	Išmetama klaida, kad nenurodyti duomenys	Pavyko
Užpildomi duomenys ir spaudžiama “kurti dokumentą”	Dokumentas sukurtas ir parodomas	Pavyko

4 lentelė. Paveiksluko manipuliavimo dokumente testavimo rezultatai

Testas	Laukiamas rezultatas	Rezultatas
Pasirenkamas paveikslukas iš kompiuterio ir spaudžiamas įkėlimo mygtukas	Vartotojui parodomas įkėlimo langas, kuriame rodoma būseną įkėlimo ir įkėlus parodomas paveikslukas	Pavyko
Pasirenkamas paveikslukas iš	Paveikslukas atsiranda	Pavyko

sistemos ir įkeliamas į svetainę	dokumente ir juo leidžiama manipuluoti – didinti/mažinti, perkelti	
Pasirenkamas paveikslukas ir spaudžiama “ištrinti”	Paveikslukas ištrintas iš dokumento ir nėra matomas	Pavyko

5 lentelė. PDF failo įkėlimo į užsakymą testavimo rezultatai

Testas	Laukiamas rezultatas	Rezultatas
Pasirenkamas PDF failas iš kompiuterio ir spaudžiama įkelti	Vartotojui parodomas informacinis langas apie įkėlimo būseną. Įkeltas failas išanalizuojamas ir vartotojas pamato laukelių reikšmes pakeistas – puslapių skaičiaus, dydžiaus dokumento ir kainos.	Pavyko
Pasirenkamas PDF failo vienas iš puslapių	Vartotojui parodomas sugeneruotas puslapio paveikslukas	Pavyko

2. Tikslingai atsirenkami elementai, kuriems parašomi automatiniai sąsajos veiksmų scenarijai pasitelkus Selenium įrankį.

Tokio testo pavyzdys:

```

WebElement element = driver.findElement(By.name("userName"));
WebElement element2 = driver.findElement(By.name("userPassword"));
element.sendKeys("testName");
element.sendKeys("testPassword");
String userName = driver.findElement(By.name("userName")).getText();
String userPassword = driver.findElement(By.name("userPassword")).getText();
Assert.assertEquals(userName, "testName");
Assert.assertEquals(userPassword, "testPassword");

```

3.4.2. Vientų testavimas

Tokiems testams bus taikomas baltos dėžės testavimo metodas, kai testuotojas gali disponuoti programos kodu. Pirmiausiai bus išsiaiškinta, kokius duomenis reikia paduoti į atitinkamą programos komponentą ir bus nustatomi laukiami rezultatai arba leistinos ir draudžiamos išėjimo duomenų reikšmės.

Testiniai atvejai yra gaunami iš programos struktūros. Vienetų testavimas bus naudojamas atskiroms klasėms testuoti. Jos bus testuojamos sukuriant objektus, perduodant įėjimo duomenis, stebint gautus rezultatus ir lyginant juos su laukiamais rezultatais.

Toliau pateikiami numatomi testuoti klasių metodai, įvedami pradiniai duomenys bei nurodomas laukiamas rezultatas. Yra paminėti preliminarūs metodai, kurie bus testuojami. Vartotojo sąsajos dalies testuoti metodai parodyti 6 lentelėje, o serverio dalies 7 lentelėje.

6 lentelė. Testuoti metodai vartotojo sąsajos

Metodas	Įeinantys duomenys	Rezultatas kurio tikimasi	Testavimo rezultatas
getSelectedViewsId	-	Id puslapio, kuris įjungtas arba tuščias stringas.	Klaidos nėra.
photobook_get_size_of_element_def	-	12x12	Klaidos nėra.
photobook_get_cm_of_document_size_def	-	30.48cm*30.48cm.	Klaidos nėra.

7 lentelė. Testuoti metodai serverio pusės

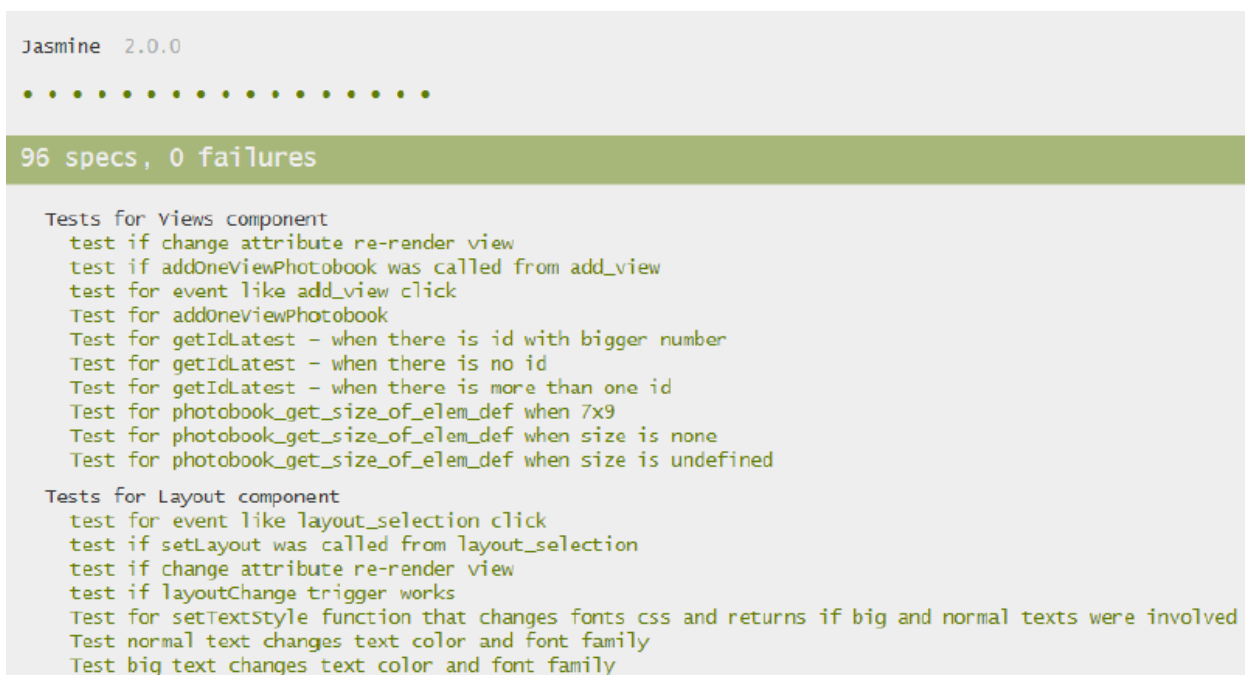
Metodas	Įeinantys duomenys	Rezultatas kurio tikimasi	Testavimo rezultatas
createPhotobook	vardas	Fotoknygos id	Klaidos nėra.
addViewToPhotobook	-	Success/False	Klaidos nėra.
createAlbum	vardas	Išsaugotas albumo id	Klaidos nėra.

Šie testai buvo rašomi pasitelkus Jasmine Javascript testų rašymo biblioteką. Pavyzdinis testo kodas pateikiamas žemiau:

```
describe("Login", function() {
  it("calls the Login() function", function() {
    var fakeLogin = new Login();
    spyOn(fakeLogin, "checkIfUserExists");
    fakeLogin.tryLogin("testName","testPassword");
  });
});
```

```
    expect(fakeLogin.checkIfUserExists).toHaveBeenCalled();  
  });  
});
```

Atlikus šiuos testus visos surastos klaidos (apytiksliai 21) pataisytos ir sėkmingų testų rezultatas pateikiamas 19 pav.



```
Jasmine 2.0.0  
.....  
96 specs, 0 failures  
  
Tests for Views component  
  test if change attribute re-render view  
  test if addOneViewPhotobook was called from add_view  
  test for event like add_view click  
  Test for addOneViewPhotobook  
  Test for getIdLatest - when there is id with bigger number  
  Test for getIdLatest - when there is no id  
  Test for getIdLatest - when there is more than one id  
  Test for photobook_get_size_of_elem_def when 7x9  
  Test for photobook_get_size_of_elem_def when size is none  
  Test for photobook_get_size_of_elem_def when size is undefined  
  
Tests for Layout component  
  test for event like layout_selection click  
  test if setLayout was called from layout_selection  
  test if change attribute re-render view  
  test if layoutChange trigger works  
  Test for setTextstyle function that changes fonts css and returns if big and normal texts were involved  
  Test normal text changes text color and font family  
  Test big text changes text color and font family
```

19 pav. Jasmine vienetų testų rezultato langas

3.4.3. Integracinis testavimas

Naudojant vienetų testavimą bus patikrinti individualūs programinės įrangos komponentai. Šie komponentai yra apjungiami ir testuojami naudojant integracinį testavimą. Integraciniame testavime dalyvaus vartotojo pusės kodo ir serverio pusės integracija. Vartotojo sąsaja yra atsakinga už bendradarbiavimą su serveriu, todėl reikės ištestuoti ar užklauskos kviečiant serverio pusę bei skaitant jos išduotą informaciją yra tinkamai suprantamos, ar tinkamai sulaukiama asinchroninių procesų pabaigos.

3.4.4. Priėmimo testavimas

Šis testavimas bus atliekamas kaip juodos dėžės testas. Pagrindinė užduotis patikrinti, kaip sistema atitinka specifikaciją, funkcinius bei nefunkcinius reikalavimus. Taip pat, šio testavimo metu

bus tikrinama ar pateikiant duomenis jie yra tinkamai apdorojami ir atvaizduojami grafinėje vartotojo sąsajoje.

Testavimas atliekamas peržiūrint kiekvieną programinės įrangos panaudojimo atvejį. Aptikus neatitikimus vartotojo norams bus tikrinama ar funkcijos atitinka specifikaciją. Esant neatitikimams bus registruojamas sistemos pakeitimas.

Buvo užsakovui demonstruojama programinė įranga ir peržiūrimi panaudojimo atvejai. Panaudojimo atvejų peržiūros rezultatai aprašyti žemiau.

Panaudojimo atvejis: Matyti užsakymus

- Patikrintas funkcionalumas. PA veikia teisingai.

Panaudojimo atvejis: Pateikti užsakymą

- Patikrintas funkcionalumas. PA veikia teisingai.

Panaudojimo atvejis: Sukurti albumą

- Patikrintas funkcionalumas. PA veikia teisingai.

Panaudojimo atvejis: Sukurti dokumentą

- Patikrintas funkcionalumas. PA veikia teisingai.

Panaudojimo atvejis: Modifikuoti dokumentą

- Patikrintas funkcionalumas. PA veikia teisingai.

3.5. Išvados

1. Sąsajos testavimas leido ištestuoti bendrą sistemos veikimą iš vartotojo požiūrio. Testuotojas gavo scenarijus, kuriuos tikrino rankiniu būdu. Rezultatai puikūs - viskas veikė. Tačiau ateitį sistema bus keičiama, todėl buvo nuspręsta pasinaudojus Selenium įrankiu parašyti automatinių testų, kurie padės patestuoti labiausiai naudojamas vietas ir greičiau atrasti klaidas.
2. Vienetų testavimas padėjo atrasti 21 klaidą ir jas ištaisyti kūrimo stadijoje. Tokiu būdu sutaupant daug kūrimo ir keitimo laiko. Naudojant Jasmine įrankį buvo parašyta 96 vienetų testai, kurie toliau padės atrasti klaidas keičiant kodą.
3. Integracinis testavimas buvo daromas apjungiant komponentus. Vartotojo ir serverio sąsajos komponentų apjungimas leido atrasti klaidas tarp šių skirtingų technologijų.
4. Priėmimo testavimas vyko tikrinant panaudojimo atvejus funkcionalumo požiūriu. Tikrinta specifikacija ir sistemos veikimas jos atžvilgiu. Visi panaudojimo atvejai veikė tinkamai.

5. Sistema ištestuota iš įvairių pusių ir galima daryti prielaidą, kad likusios klaidos gali būti nežymios ir netrukdyti naudojimuisi sistema.

4. TYRIMO DALIS

Analitinėje dalyje buvo bendrai aptartos naršyklės rengyklų technologijos, kurios leidžia manipuluoti paveikslukais. Šioje dalyje bus tiriamos jų metrikos ir parametrai kiekvienų technologijų.

4.1. Naršyklėje veikiančių technologijų palyginimas

Analitinėje dalyje buvo bendrai pažiūrėta į kiekvieną iš technologijų. Šiame tyrime buvo šios technologijos apibendrintos ir palygintos. Šios technologijos turi savo plusus ir minusus, kuriuos matome 8 lentelėje. Flash technologija iš anksto atmetama dėl jos atgyvenimo, tačiau dėl bendro supratimo – ji palyginama, nes ji vis dar naudojama.

8 lentelė. Naršyklėje veikiančių technologijų palyginimas

Palyginimo kriterijai	Canvas	WebGL	SVG	Flash	HTML5 (div) JS/CSS3
Atvaizdavimas	JavaScript	JavaScript	Naršyklė atsižvelgiant į pateiktą SVG kodą	Flash įskiepis	Naršyklė atvaizduoja
Preliminariai spėjamas greitis	Su aparatūros pagreitinimu – greitas. Įprastai – vidutinis.	Greitas	Su aparatūros pagreitinimu – greitas. Įprastai – vidutinis.	Lėtas	Vidutinis. Naudojama CPU. Tačiau tam tikri būdai leidžia GPU naudoti ir padaryti greitu.
Palaikomumas naršyklų	Visos naujausios pilnai palaiko [12].	70proc. kompiuterių palaiko ir apytiksliai 9 proc. mobilių įrenginių.	Visos naujausios pilnai palaiko [27].	Palaiko visos. Tačiau technologija pripažinta atgyvenusia [10].	Visos naršyklės [28].

4.2. Bibliotekų pasirinkimas

Praktiškai kuriant programinę įrangą dažnai tenka ieškoti bibliotekų, kurios pagreitintų kūrimo procesą. Buvo atsirinktos vienos iš populiariausių bibliotekų ir taip pat pagal aprašymą viena greičiausių ir šios bibliotekos parodytos 9 lentelėje. Buvo išskirtos kategorijos pagal ką bus parenkama biblioteka. Šios kategorijos yra šios:

- Javascript pagrindu – GSAP TweenMax pagal aprašymus greičiausia, nes jis buvo kuriamas orientuojantis į animacijų kūrimą [29].
- WebGL pagrindu – ThreeJS pagal aprašymus daugiausia funkcionalumo. Žinant, kad programuojant WebGL reikia daug laiko.

Canvas ir JQueryUI dėl savo populiarumo buvo parinktos.

9 lentelė. Naršyklėje veikiančių bibliotekų palyginimas

Palyginimo kriterijai	JQuery UI	GSAP TweenMax	ThreeJS	Canvas standartinis
Tarpinė biblioteka	+ (jQuery)	-	-	-
Technologija	HTML5 div	HTML5 div	WebGL	Canvas
Papildoma informacija		Integravimas su JQuery biblioteka [29]	WebGL – piešia ant Canvas elementų, todėl turi panašių savybių.	Pikseliai pasiekiami per Canvas sistemą. Suderinamas tarp naršyklių. [30]

4.3. Nuotraukos manipuliacijos pasirinkimas eksperimentui

Nuotraukas naršyklėje galima manipuluoti dauguma būdų. Pagrindinės yra šios pagal Taopix [9]:

- Keisti jos aukštį ir plotį;
- Keisti jos poziciją dokumente;
- Kirpti nuotrauką;
- Suteikti tam tikrų išvaizdos savybių:
 - Uždėti šešėlį;
 - Pakeisti nuotraukos orientaciją;
 - Suteikti tam tikrus kontūrus;
 - Uždėti nuotraukai rėmelį;

Atliekant eksperimentą, kuri iš bibliotekų yra greičiausia teks išsirinkti vieną iš nuotraukų manipuliavimo būdų ir su ja patikrinti greičius. Peržvelgiant kiekvieną iš pagrindinių būdų geriausias variantas yra padaryti eksperimentą su nuotraukos aukščio ir pločio keitimu. Kodėl?

- Pozicijos keitimas reikalauja tik paveiksluko perpiešimo kitoje vietoje – nėra pats paveikslukas įtakojamas;

- Kirpti nuotrauką – kirpimui tektų ieškotis dar bibliotekų, kurios padėtų tai padaryti, todėl eksperimentui tektų daugiau laiko skirti.
- Suteikimas tam tikrų išvaizdos savybių dažniausiai būna greitas procesas.

4.4. Piešimo optimizavimo algoritmo efektyvumas

Pats paveiksliuko piešimo/kokybės optimizavimas gali būti atliekamas arba serverio pusėje, arba klientinėje pusėje. Kadangi serverio pusę šiuo atveju atmetama, kadangi vartotojas nori realiu metu manipuluoti paveiksliuku, todėl bus kalbama apie klientinę pusę.

Klientinėje pusėje norint optimizuoti paveiksliuką – geriausia gauti to paveiksliuko kiekvieno pikselio koordinatės ir su jomis atlikti operacijas. Kiek ieškota – atrodo tik su Canvas elementais galima taip gauti pikselių koordinates [31].

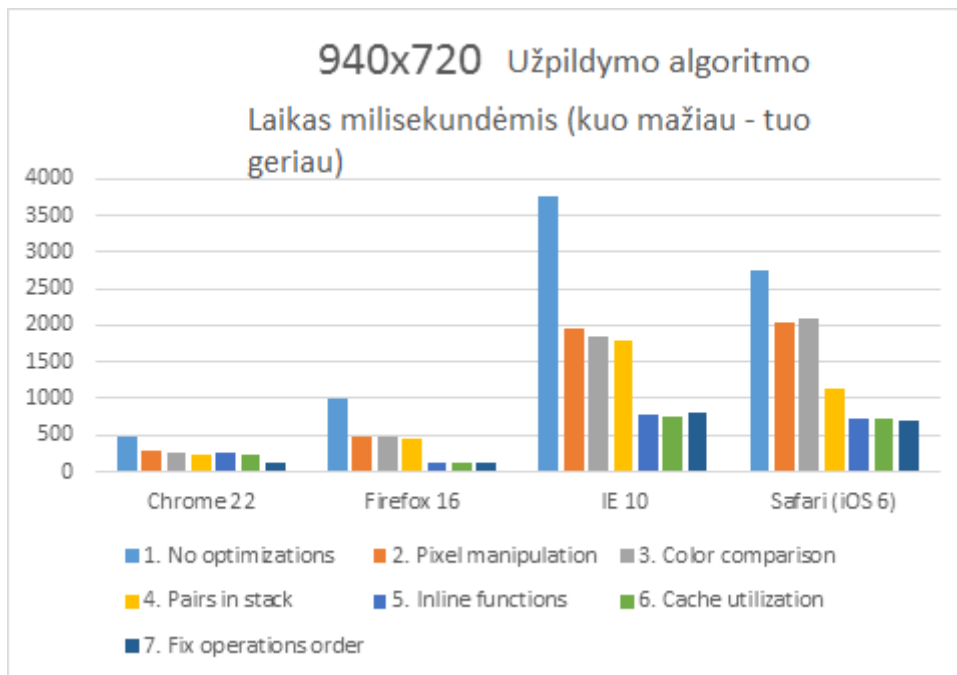
Užpildymo (angl. “*Flood fill*”) algoritmo optimizavimo greičio tyrimas

Šis algoritmas apskaičiuoja susijusias gijas daugiamačiame masyve. Algoritmas naudojamas daugiausia piešimo programose spalvos užpildymui. Bendras algoritmas C++ kodo [32].

```
//Recursive 4-way floodfill, crashes if recursion stack is full
void floodFill4(int x, int y, int newColor, int oldColor)
{
    if(x >= 0 && x < w && y >= 0 && y < h && screenBuffer[y][x][y] == oldColor &&
screenBuffer[x] != newColor)
    {
        screenBuffer[y][x] = newColor; //set color before starting recursion

        floodFill4(x + 1, y, newColor, oldColor);
        floodFill4(x - 1, y, newColor, oldColor);
        floodFill4(x, y + 1, newColor, oldColor);
        floodFill4(x, y - 1, newColor, oldColor);
    }
}
```

Šį eksperimentą buvo atlikę Victor Haydin ir Yuriy Guts [30]. Jie sukūrė Canvas pagrindu veikiančią redaktorių, kuriame buvo galima piešti ir užpildyti laukus spalva. Todėl šį užpildymo algoritmą Javascript kalba parašė. Po to šį kodą optimizavo ir palyginimo naršyklės pagal greitį. Rezultatai pateikti 20 pav. paveiksliuke. Kaip matome iš rezultatų – labiausiai optimizacijos reikia IE 10 ir Safari naršyklėms. Galima tik daryti prielaidą, kad šios naršyklės pačios savyje atlieka mažai optimizacijų dėl greičio. Tuo tarpu Chrome labai mažai atsiliko greičio atžvilgiu nuo pradinės kodo versijos.



20 pav. Užpildymo algoritmo optimizacijos – greičių palyginimas tarp naršyklių

4.5. Tyrimo išvados

1. Atliktas tyrimas, kurio metu buvo palygintos naršyklėje veikiančios technologijos nuotraukų manipuliavimui. Taip pat išskirtos bibliotekos ir aptartos jų savybės.
2. Kaip buvo išsiaiškinta - Flash technologija oficialiai laikoma atgyvenusia. Tačiau naujai pristatytos HTML5 technologijos atrodo, kad yra pasirengusios ją pakeisti. Aparatūros pagreitinimas veikia kiekvienoje iš jų. Kiekvieną iš šių technologijų naujausios naršyklės palaiko, tačiau WebGL palaikomumas priklauso nuo kompiuterio - šiuo metu apytiksliai 70 procentų kompiuterių palaiko šią technologiją.
3. Bibliotekos buvo pasirinktos pagal technologiją. Canvas sukurti programos paprasta, todėl ji nereikalauja bibliotekos, o WebGL norint sutaupyti laiko - geriau rinktis biblioteką. Taip pat ir su HTML5 div atveju - buvo rinkta bibliotekas. JQuery buvo pasirinkta dėl populiarumo ir GSAP pagal tai, kad ji buvo sukurta orientuojantis į animacijų kūrimą.
4. Taip pat buvo aptartos nuotraukų manipuliacijos. Pasirinktas eksperimento darymo būdas - keičiant aukštį ir plotį, kadangi tokiu būdu ištestuojamas vienas iš kritinių atvejų.
5. Atlikus optimizavimo tyrimą buvo atrasta, kad Canvas technologija leidžia tai atlikti gaudama kiekvieno pikselio duomenis. Tačiau buvo pamatyta, kad Chrome naršyklė savyje turi daug optimizacijų. Todėl jei orientuojamasi į šią naršyklę - optimizacijos šiuo atveju neatlieka didelį vaidmenį. Labiausiai optimizacijos paveikė IE 10 ir Safari naršykles.

5. EKSPERIMENTO DALIS

Šioje dalyje yra atliekama projektinėje dalyje sukurtos programinės įrangos eksperimentai. Projektinėje dalyje buvo realizuotos HTML5 standarto pristatytos technologijos, kurios leidžia manipuluoti paveikslukais. Šioje dalyje pateikiama šių technologijų eksperimentai bei išvados.

Programinės įrangos realizacijos metu buvo realizuotos keturios naršyklėje veikiančios paveikslukų manipuliavimo technologijos:

- HTML5 (div) – jQuery UI pagrindu [33] [34].
- HTML5 (div) – Javascript pagrindu sukurta GSAP TweenMax biblioteka [35].
- Canvas – pasinaudojant standartiniu funkcionalumu [36].
- WebGL – pasinaudojant ThreeJS biblioteka [37].
- SVG – pasinaudojant jQuery keičiamas aukštis, todėl eksperimento su šia sistema nebus atliekama – pats SVG savo viduje naudoja įprastą nuotraukos atributą ir nuotrauka yra užkrauna kaip ir „HTML5 div“ atveju.

5.1. Preliminarus greičio nustatymo eksperimentas

Testavimas nepalankiausiomis sąlygomis buvo atliktas (angl. “*Stress testing the performance*”) naudojantis GSAP sukurtu įrankiu [29]. Canvas stresiniam testavimui buvo naudojama Scott Porter sukurtu įrankiu [38].

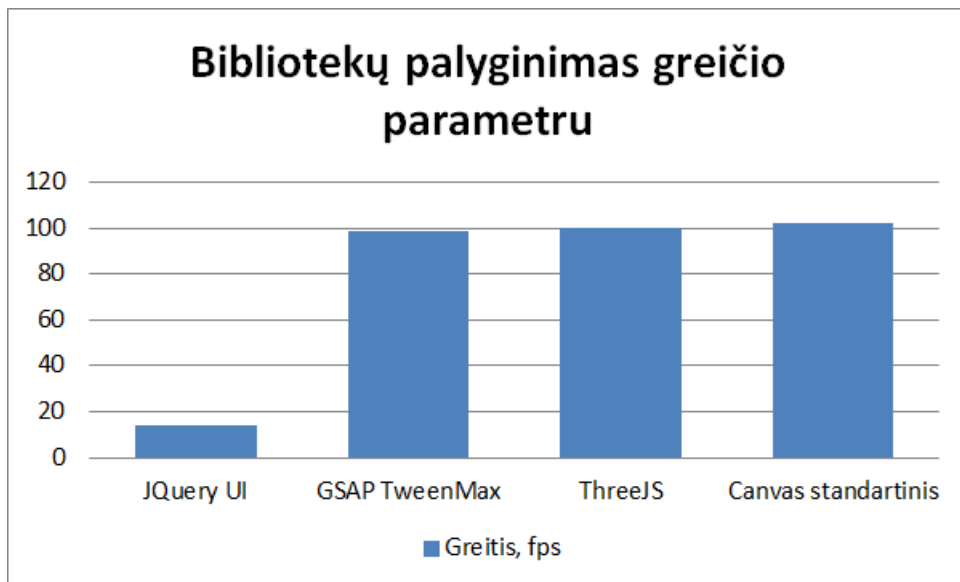
Testavimui naudoto kompiuterio techniniai parametrai: CPU – Intel Core 2 branduolių (2.20GHz), RAM – 12GB, Windows 10 64bit. Integruota vaizdo plokštė – 512MB.

Testai buvo atlikti naudojantis Chrome 57.0.2987.133 versija, kadangi didžiausia dalis – apie 56 procentai vartotojų naudojami Chrome naršykle [39].

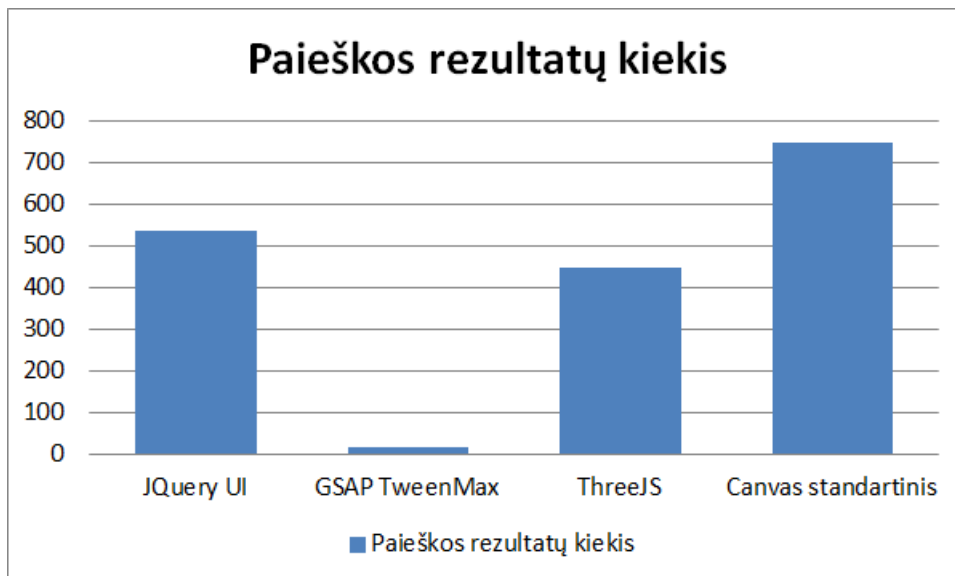
10 lentelė. Naršyklėje veikiančių technologijų tyrimo rezultatai

Palyginimo kriterijai	jQuery UI	GSAP TweenMax	ThreeJS	Canvas standartinis
Vidutinis greitis stresinio testavimo	14 fps	99 fps	100fps	102 fps
Paieškos rezultatų kiekis ¹	Apie 536 000 rezult.	Apie 16 600 rezult.	Apie 447 000 rezult.	Apie 746 000 rezult.

¹ Pavyzdžių kiekis – įrašius į Google paieškos laukelį “biblioteka image resize”.



21 pav. Bibliotekų palyginimas greičio parametru



22 pav. Paieškos rezultatų kiekio palyginimas

Canvas šiuo atveju labiausiai laimi – jo pavyzdžių daugiausia (virš 700 tūkst.) ir stresinis testavimas parodė, kad jis greičiausias 102 fps kaip kad matome 10 lentelėje ir 21 pav.- 22 pav.. Nors TweenMax parašytas vien su „javascript“ ir yra palyginus su jQuery daug greitesnis, tačiau jis turi mažai pavyzdžių, todėl bus sunku išspręsti kilusias problemas. ThreeJS kaip ir Canvas yra greitas, tačiau buvo sukurtas 3D animacijoms kurti, todėl daroma prielaida, kad daugiausia pavyzdžių susijusių su 3D, todėl kūrimas su šia sistema gali būti problematiškas.

5.2. Paveiksliais manipuliavimo naršyklėje technologijų eksperimentas

Buvo atliekamas eksperimentas, kurio tikslas išsiaiškinti, kuri iš realizuotų technologijų yra tinkamiausia manipuluoti plataus formato nuotraukomis.

Pagrindiniai eksperimento uždaviniai yra: ištestuoti pasirinktas plataus formato nuotraukas pasinaudojant projekto metu sukurta naršyklės rengykle; atlikti rezultatų analizę bei pateikti palyginimus ir išvadas.

Technologijų vertinimo kriterijai:

- **FPS** – kadrų per sekundę skaičius. Skaičiuojama plataus formato nuotraukos dydžio keitimo ir kitų manipuliacijų greitis – ar vartotojas mato vėlavimus.
- **CPU** – pagrindinio procesoriaus išnaudojimo skaičius. Skaičiuojama kiek procesoriaus pajėgumų reikalauja manipuliacijos su nuotraukomis.
- **GPU** – grafinio procesoriaus išnaudojimo skaičius. Skaičiuojama kiek grafinio procesoriaus resursų išnaudoja manipuliacijos su nuotraukomis.
- **Parodymo laikas** – laikas per kurį vartotojas pamato paveiksluką.

5.3. Eksperimento aplinka

Bandymai buvo atliekami su dviem kompiuteriais pasinaudojus trimis naršyklėmis.

Kompiuterių techniniai parametrai:

- Pirmo kompiuterio techniniai parametrai: CPU – Intel Core 2 branduolių (2.20GHz), RAM – 12GB, Windows 10 64bit. Integruota vaizdo plokštė – 512MB.
- Antro kompiuterio techniniai parametrai: CPU – Intel 2 branduolių, RAM – 8GB, Windows 10 64bit. AMD Radeon HD 6570 vaizdo plokštė.

Naršyklė:

- Chrome - versija 57.0.2987.133 (64-bit).

5.4. Naudojamos plataus formato nuotraukos

Testavimui buvo naudotos skirtingo dydžio plataus formato nuotraukos, kurios buvo gautos iš įvairių šaltinių. Nuotraukos matmenims apskaičiuoti naudojama Jim Beecher sukurta skaičiuoklė [40] ir taip pat sritis, kuriose tokios nuotraukos dažniausiai spausdinamos (pagal tai DPI keičiama) [41]. Orientuotasi į plataus formato dydžius. 36 x 24 colių – MixBook siūlo. Spausdinti tokio dydžio nuotrauką reiktų 9000px x 6000px naudojant 250 dpi (angl. „Dots per inch“). BrantInStore gali išspausdinti nuotrauką iki 126'' x 63'' colių [42]. Kokybiškai išspausdinti tokią nuotrauką reiktų minimum 12600px x 6300px nuotraukos pasirenkant 100 dpi. Tačiau dėl testavimo tikslumo bus imamos vienodos pikselių matmenys. Taip pat nuotraukos dydžiai testuojami sukuriant paveiksluką su „Paint“ programa ir didinant nuotraukos matmenis. Kadangi dažniausiai fotoaparatai sukuria „JPG“ tipo nuotraukas, todėl ir šis tipas bus naudojamas. Pavyzdinė nuotrauka parodyta 23 pav.



23 pav. Pasirinktos testuoti technologijas pavyzdinė nuotrauka

Naudojamų nuotraukų charakteristikos pateikiamos 11 lentelėje.

11 lentelė. Testavimui naudotų nuotraukų charakteristikos

Matmenys, px	Spausdinimo dydis	Tipas	Failo dydis
8000 x 8000	33 x 33 colių, 250 dpi	JPG	829 KB
10000 x 10000	40 x 40 colių, 250 dpi	JPG	1.19 MB
12000 x 12000	120 x 120 colių, 100 dpi	JPG	1.17 MB
14000 x 14000	140 x 140 colių, 100 dpi	JPG	2.83 MB

5.5. Naudojami įrankiai

Testuojama Chrome naršyklėje, kadangi kaip buvo minėta tyrime – ji populiariausia.

- CPU naudojimui skaičiuoti buvo naudojamas Windows “Task Manager” įrankis. Taip pat Chrome naršyklės „atlikimo įrankis“ (angl. „*Performance tool*“) ir įskiepis „Processor Monitor“.
- GPU naudojimui apskaičiuoti buvo naudojamas “FPS Meter” Chrome naršyklės įrankis.
- FPS naudojimui skaičiuoti buvo naudojama “JavaScript Performance Monitor” biblioteka [43].
- Parodymo laikui buvo Chrome naršyklės “atlikimo įrankis”.

5.6. Eksperimento scenarijus

Eksperimentas buvo atliekamas pagal žemiau pateikiamą scenarijų. Jo metu buvo lyginami projekto metu realizuotos technologijos testuojant kaip jos leidžia manipuluoti plataus formato nuotraukomis.

Scenarijus:

1. Pasirenkama plataus formato nuotrauka.
2. Pasirenkama analizuojama naršyklės technologija.
3. Keliami pasirinkta nuotrauka ir skaičiuojamas atvaizdavimo laikas.
4. Skaičiuojama žemiausi FPS, aukščiausi CPU ir GPU reikšmių vidurkiai atliekant 5 kartus šiuos veiksmus:
 - a. Nustatoma 100px plotis ir aukštis. Nuotrauka išplečiama iki 500px horizontaliai ir vertikalčiai.
5. Keičiama plataus formato nuotrauka į kitą ir procesas kartojamas nuo 2 etapo.

5.7. Eksperimento eiga ir rezultatai

Eksperimento metu nuotraukų manipuliavimo technologijos buvo testuojamos naudojant aukščiau aprašytą tyrimo scenarijų. Tačiau keičiant kompiuterius su kuriais atliekamas eksperimentas. Eksperimentas kiekvienu atveju buvo vykdomas su įjungta aparatūrinės įrangos pagreitinimo funkcija.

1. Rezultatai naudojant pirmą kompiuterį – su integruota vaizdo plokšte

Naudojant pirmą kompiuterį – rezultatai pateikiami 12 - 15 lentelėse.

HTML5 JQueryUI bibliotekos eksperimento rezultatai pateikiami 12 lentelėje. Vėlavimas prasideda naudojant 8000px x 8000px nuotrauką- vėlavimas mažas – 0.4 s ir iš penkių kartų keliskart suvėluoja. GPU, keičiant nuotrauką, ryškiai nesikeičia – pasiima daugiausia 9MB. CPU šokinėja – tačiau matosi, kad jos yra panaudojama nemažai– vidutiniškai apie 20 procentų. Vėlavimas ryškus jau nuo 2-os (10000px x 10000px) nuotraukos, kai vėlavimas pakyla virš 2 s su kiekviena kita nuotrauka.

12 lentelė.1 kompiuterio HTML5 jQuery UI bibliotekos eksperimento rezultatai

Nuotrauka, px x px	FPS	CPU, %	GPU, MB	RAM, GB	Užkrovimo laikas, s	Vėlavimas, s
8000 x 8000	1-43 - 50	4-18	5.3 - 9	5.1 – 5.5	Atsiuntimas: 0.191 s Piešimas: 0.9 s	0.4
10000 x 10000	2-30-50	3-25	5.3 - 9	5 – 6.1	Atsiuntimas: 0.730 s Piešimas: 1.8 s	2
12000 x 12000	1-11-50	4 - 34	5.3 - 9	4.7 – 6.2	Atsiuntimas: 1.17 s	4.73

Nuotrauka, px x px	FPS	CPU, %	GPU, MB	RAM, GB	Užkrovimo laikas, s	Vėlavimas, s
					Piešimas: 2.1 s	
14000 x 14000	1-21-50	6 - 26	5.3 – 9.5	3.9 - 5.8	Atsiuntimas: 3.93 s Piešimas: 2.11 s	4.3

HTML5 GSAP bibliotekos eksperimento rezultatai pateikiami 13 lentelėje. Kaip matome iš eksperimento rezultatų – ši biblioteka greita ir vėlavimas mažas. Tik nuo 3-ios nuotraukos (12000px x 12000px) pradeda jaustis vėlavimas – 1.6 s. O ties 4-a nuotrauka ryškiai vėlavimas nepadidėjo – vidutiniškai 2 sekundės buvo.

13 lentelė. 1 kompiuterio GSAP bibliotekos eksperimento rezultatai

Nuotrauka, px x px	FPS	CPU, %	GPU, MB	RAM, GB	Užkrovimo laikas, s	Vėlavimas, s
8000 x 8000	48 - 50	4-24	2.4 – 6.8	5.5 – nesikeitė	Atsiuntimas: 0.255 s Piešimas: 1.1 s	0
10000 x 10000	41-50	3-23	2.6 – 6.8	4.9 – nesikeitė	Atsiuntimas: 0.38 Piešimas: 1.8	0
12000 x 12000	7-27-50	4 - 20	3 – 6.8	5.1 – 6.6	Atsiuntimas: 0.5 s Piešimas: 2.1 s	1.6
14000 x 14000	5-39-50	12 - 29	2.8 - 6.8	5.5-7	Atsiuntimas: 1.4 s Piešimas: 2.2 s	2

Canvas eksperimento rezultatai pateikiami 14 lentelėje. Vėlavimas ypač pasijaučia nuo 3 nuotraukos (12000px x 12000px) ir jis didelis – 8 s. Ir didėjant nuotraukom – jis tik didėja.

14 lentelė. 1 kompiuterio Canvas technologijos eksperimento rezultatai

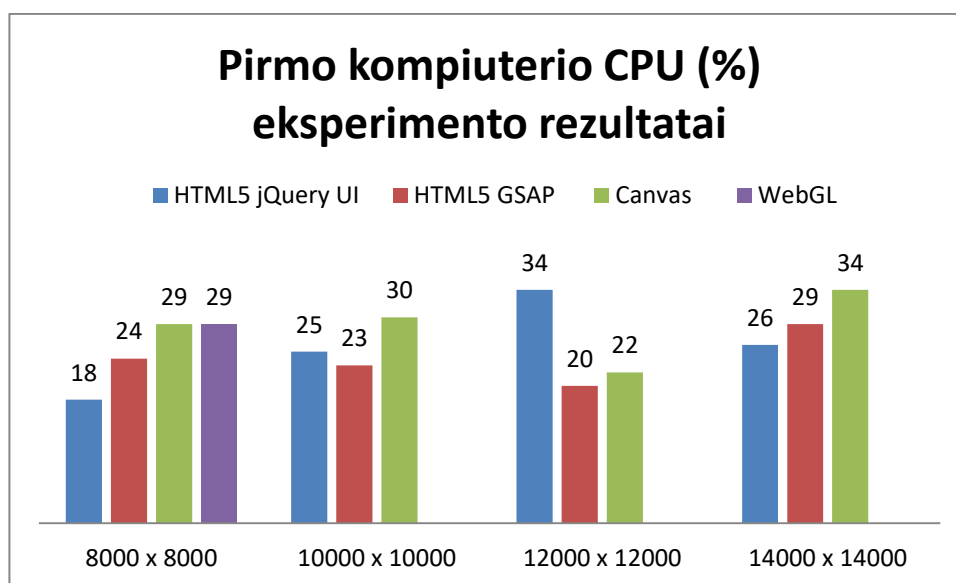
Nuotrauka, px x px	FPS	CPU, %	GPU, MB	RAM, GB	Užkrovimo laikas, s	Vėlavimas, s
8000 x 8000	39-50	11-29	4.6 – 5.1	5.1 – nesikeitė	Atsiuntimas: 0.122 s Piešimas: 0.4 s	0 s
10000 x 10000	41 - 50	11 - 30	4.6 - 6	5.1 – nesikeitė	Atsiuntimas: 0.353 s Piešimas: 1.15 s	0 s
12000 x 12000	1-4-50	6-22	4.6 - 6	5.3 - 6.7	Atsiuntimas: 1.1 s Piešimas: 1.3 s	8 s
14000 x 14000	1-20-50	6-34	5.3 - 6.8	5.3 - 6.7	Atsiuntimas: 1.33 s Piešimas: 1.7 s	11 s

Kompiuteris, kuris naudojamas testavimui palaiko 8192 WebGL technologijos MAX_TEXTURE_SIZE reikšmę, todėl “ThreeJs” bibliotekos sistema paveiksluką sumažino į šį dydį (8192px x 8192px) [20]. Todėl buvo testuojama tik su žemiausiu paveiksluku. Kaip ir buvo tikėtasi – jokio vėlavimo, FPS šokinėja tarp nepastebimo lygio akiai.

15 lentelė. 1 kompiuterio WebGL technologijos eksperimento rezultatai

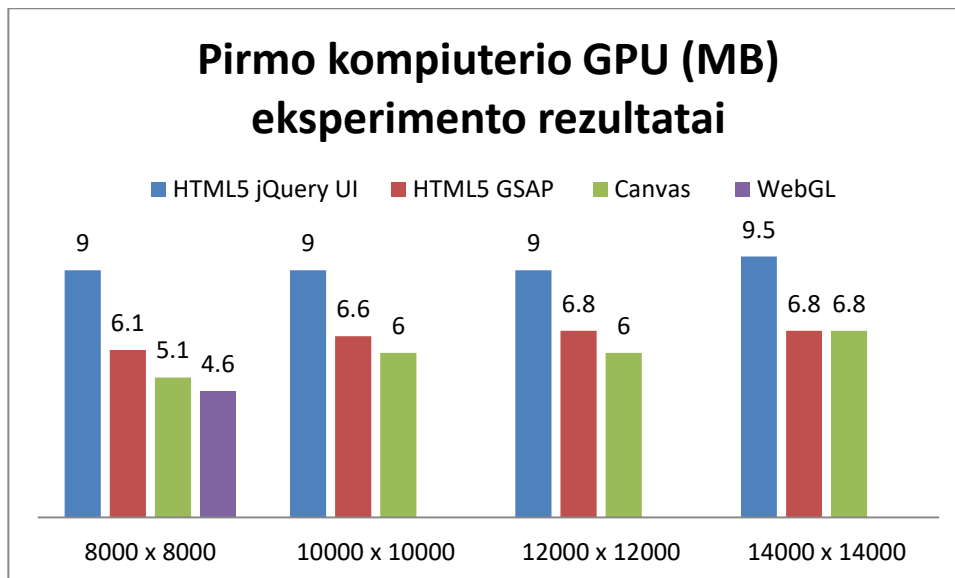
Nuotrauka, px x px	FPS	CPU, %	GPU, MB	RAM, GB	Užkrovimo laikas, s	Vėlavimas, s
8000 x 8000	39-50	11 - 29	2.1 - nesikeitė	5.7 – nesikeitė keičiant	Atsiuntimas: 1.30 s Piešimas: 2.7 s	0 s

CPU eksperimento rezultatai pateikti 24 pav. Kaip matome iš diagramos – CPU naudojimas pasiskirsto netolygiai. Canvas ir JQueryUI vidutiniškai naudoja daugiausia CPU galios – vidutiniškai iki 25 procentų pakyla. Tuo tarpu GSAP vidutiniškai mažiausiai – 22 procentus.



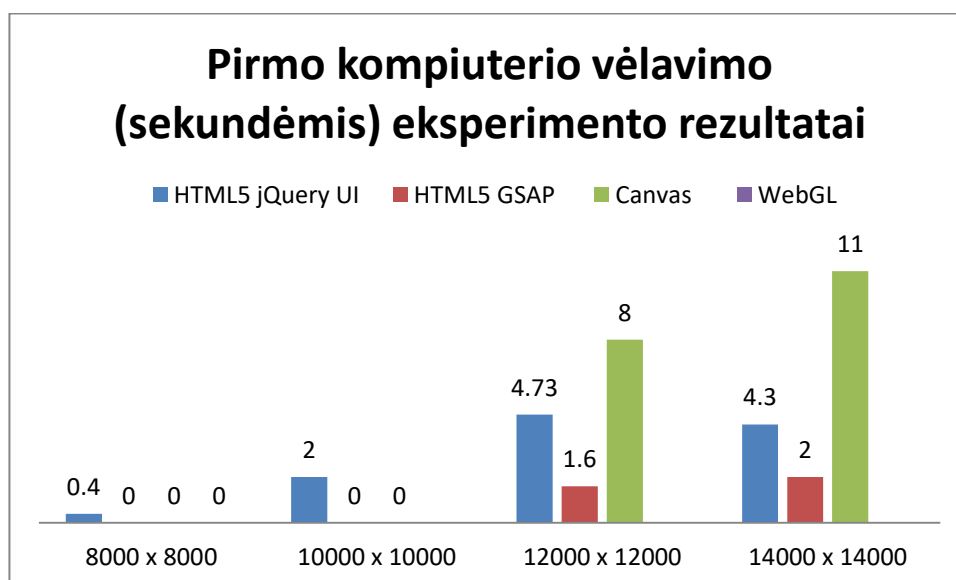
24 pav. Pirmo kompiuterio CPU (%) eksperimento rezultatai

GPU eksperimento rezultatai pateikti 25 pav. Mažiausiai GPU galios išnaudoja Canvas – vidutiniškai 6MB, tačiau GSAP biblioteka nedaug atsilieka – vidutiniškai 6.2MB išnaudoja. Tuo tarpu JQueryUI GPU daugiausia naudoja – 8.5MB.



25 pav. Pirmo kompiuterio GPU (MB) eksperimento rezultatai

Vėlavimo eksperimento rezultatai pateikti 26 pav. JQueryUI jau turi vėlavimų ir su 8000px x 8000px matmenų paveikslu – kaip matėme ji daugiausia GPU ir CPU galios naudoja, tačiau nesugeba tokio dydžio paveikslukų greit apdoroti, o su didesniais paveikslukais vėlavimo tik didėja – iki 4.5 sekundės su 14000px x 14000px matmenų paveikslu. Geriausiai atrodo GSAP biblioteka – vėlavimų šioje bibliotekoje mažiausiai net ir su 14000px x 14000px paveikslu – tik 2 s. Tuo tarpu Canvas iki 10000px x 10000px matmenų paveikslukais susidoroja puikiai – nebuvo matyti vėlavimo, tačiau su didesniais vėlavimas ypač ryškus pasimatė – jau su 12000px x 12000px matmenų paveikslu vėlavimas vidutiniškai buvo 8 s.



26 pav. Pirmo kompiuterio vėlavimo (sekundėmis) eksperimento rezultatai

2. Rezultatai naudojant antrą kompiuterį – su AMD Radeon vaizdo plokšte

Naudojant antrą kompiuterį – rezultatai pateikiami 16 - 19 lentelėse.

HTML5 JQueryUI bibliotekos eksperimento rezultatai pateikiami 16 lentelėje. Vėlavimas prasideda naudojant 10000px x 10000px nuotrauką- vėlavimas mažas – 0.4 s ir iš penkių kartų keliskart suvėluoja. GPU, keičiant nuotrauką, ryškiai nesikeičia – pasiima daugiausia 7MB. CPU šokinėja – tačiau matosi, kad jos yra panaudojama nemažai– vidutiniškai apie 30 procentų. Vėlavimai nėra ryškūs - tik nuo 12000px x 12000px matmenų paveiksluku jau vėlavimas pasijunta – virš 1 sekundės pakyla.

16 lentelė. 2 kompiuterio HTML5 jQuery UI bibliotekos eksperimento rezultatai

Nuotrauka, px x px	FPS	CPU, %	GPU, MB	RAM, GB	Užkrovimo laikas, s	Vėlavimas, s
8000 x 8000	52-60	4-20	4.3 – 6.3	3.9	Atsiuntimas: 0.047 s Piešimas: 0.2 s	0
10000 x 10000	1-40-60	3-31	3.3 – 5.3	4.6 4.2	Atsiuntimas: 0.359 s Piešimas: 1.2 s	0.4
12000 x 12000	3-11-60	3-33	3.5 – 5.3	4.3 3.3	Atsiuntimas: 1.17 s Piešimas: 0.8 s	1.04
14000 x 14000	1-24-60	4-37	4.3 – 7	4.6 3.9	Atsiuntimas: 1.23 s Piešimas: 1.01 s	1.5

HTML5 GSAP bibliotekos eksperimento rezultatai pateikiami 17 lentelėje. Kaip matome iš tyrimo rezultatų – ši biblioteka greita ir vėlavimas mažas ir su šiuo kompiuteriu. Vėlavimai maži ir jaučiasi gerai tik su 14000px x 14000px matmenų nuotrauka.

17 lentelė. 2 kompiuterio GSAP bibliotekos eksperimento rezultatai

Nuotrauka, px x px	FPS	CPU, %	GPU, MB	RAM, GB	Užkrovimo laikas, s	Vėlavimas, s
8000 x 8000	57 - 60	16-22	2.5 – 4.3	3.9 – nesikeitė	Atsiuntimas: 0.053 s Piešimas: 0.3 s	0
10000 x 10000	55-60	17-25	2.5 – 4.3	4.0 – nesikeitė	Atsiuntimas: 0.88 Piešimas: 0.4	0
12000 x 12000	3-24-60	4 - 30	2.5 – 6.8	3 – 4.1	Atsiuntimas: 1.25 s Piešimas: 0.5 s	0.7
14000 x 14000	1--60	2- 33	3.3-4.3	3-4.0	Atsiuntimas: 3.8 s Piešimas: 1.1 s	1.3

Canvas eksperimento rezultatai pateikiami 18 lentelėje. Vėlavimas ypač pasijaučia nuo 3 nuotraukos (12000px x 12000px) ir jis didelis – 6 s. Ir nuotraukom didėjant – vėlavimai taip pat ypač didėja.

18 lentelė. 2 kompiuterio Canvas technologijos eksperimento rezultatai

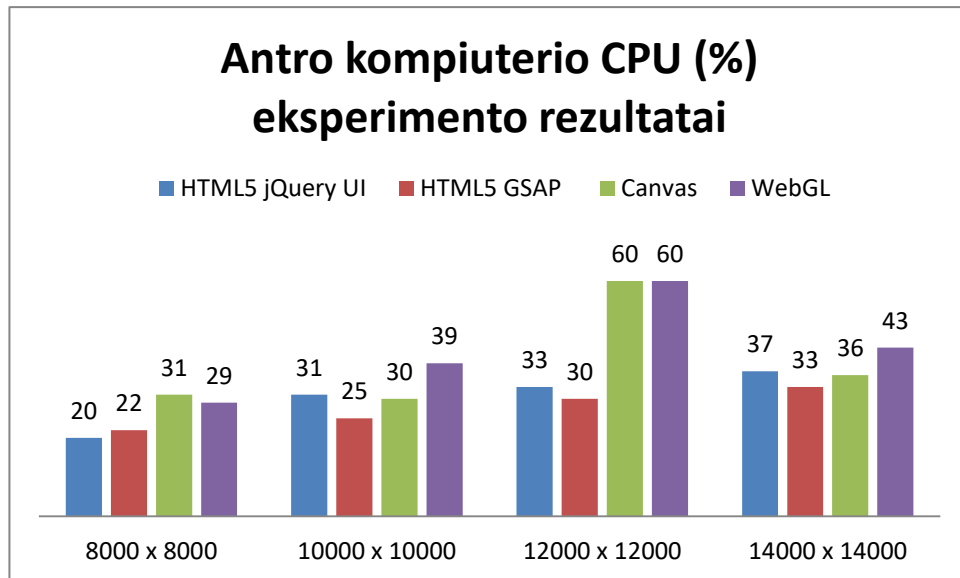
Nuotrauka, px x px	FPS	CPU, %	GPU, MB	RAM, GB	Užkrovimo laikas, s	Vėlavimas, s
8000 x 8000	60	4-31	3.6 – 5	3.3 – nesikeitė	Atsiuntimas: 0.082 s Piešimas: 0.3 s	0 s
10000 x 10000	60	4 - 30	3.6 - 5	3.7 – nesikeitė	Atsiuntimas: 0.81 s Piešimas: 0.4 s	0 s
12000 x 12000	1-60	3-60	3.6 – 5.1	4.3 - 3.4	Atsiuntimas: 1.1 s Piešimas: 1.2 s	6 s
14000 x 14000	5-60	6-36	3.6 - 5.1	4.7 - 3.2	Atsiuntimas: 3.9 s Piešimas: 1.5 s	12 s

Kompiuteris, kuris naudojamas testavimui palaiko 16384 WebGL technologijos MAX_TEXTURE_SIZE dydį. WebGL bibliotekos eksperimento rezultatai pateikiami 19 lentelėje. Vėlavimo praktiškai nesijaučia – 0.2 sekundės beveik nesijaučia, todėl dirbti su šia sistema patogiu. Tačiau GPU – vidutiniškai 8.3 MB, tačiau tik pradžioje ypač pakyla, o po to sumažėja ir keičiantis dydžiui paveiksluko nesikeičia ir būna gan mažas - iki 2.3MB. Vis dėlto, yra išnaudojama didesnė dalis RAM ir CPU lyginant su kitomis sistemomis – net iki 60 procentų pakyla, o RAM iki 6.4GB.

19 lentelė. 2 kompiuterio WebGL technologijos eksperimento rezultatai

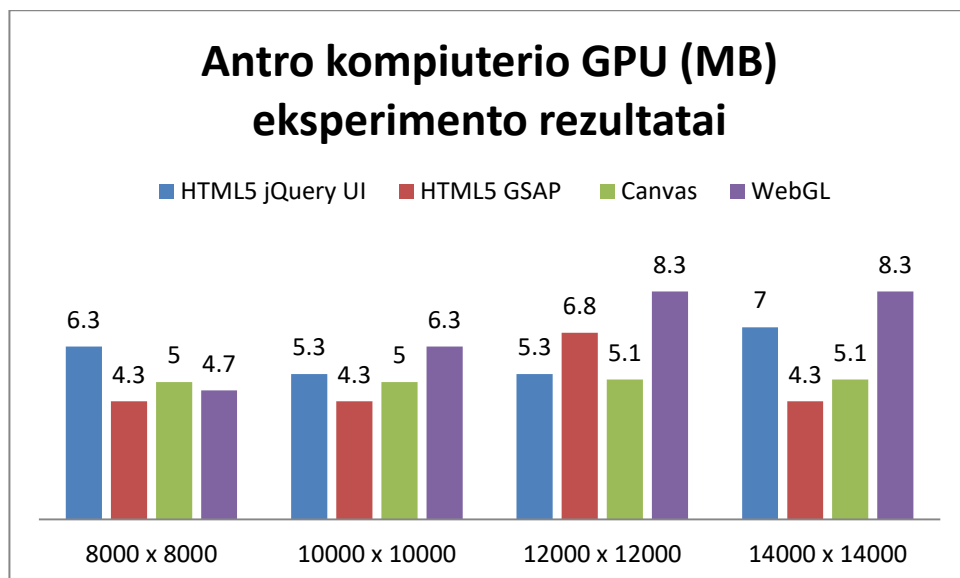
Nuotrauka, px x px	FPS	CPU, %	GPU, MB	RAM, GB	Užkrovimo laikas, s	Vėlavimas, s
8000 x 8000	39-60	11 - 29	4.7	2.5 – nesikeitė keičiant	Atsiuntimas: 1.30 s Piešimas: 2.7 s	0 s
10000 x 10000	54-60	4 - 39	6.3	4.5 – nesikeitė	Atsiuntimas: 0.81 s Piešimas: 0.4 s	0 s
12000 x 12000	1-60	3-60	8.3	4.3 - 3.4	Atsiuntimas: 1.1 s Piešimas: 1.2 s	0.1 s
14000 x 14000	52-60	10-43	8.3	6.4	Atsiuntimas: 3.9 s Piešimas: 1.5 s	0.2 s

Antro kompiuterio CPU eksperimento rezultatų suvestinė pateikiama 27 pav. Canvas ir WebGL technologijos CPU išnaudoja daugiausia – net iki 60 procentų pakyla nors vidutiniškai pakyla iki 39 procentų. Tuo tarpu GSAP biblioteka atrodo CPU išnaudoja mažiausiai – vidutiniškai 27 procentus, o jQuery šiek tiek daugiau – vidutiniškai 30 procentų.



27 pav. Antro kompiuterio CPU (%) eksperimento rezultatai

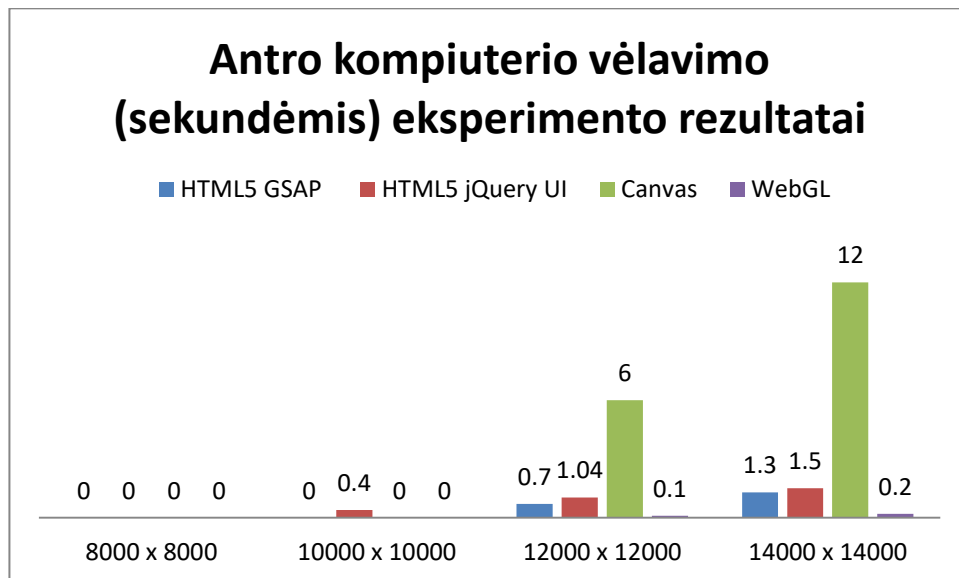
Antro kompiuterio GPU eksperimento rezultatų suvestinė pateikta 28 pav. Kas keista – WebGL naudoja mažiausiai GPU galios, tačiau ji išnaudoja daugiau kitų resursų kaip kad CPU. Tuo tarpu daugiausia JQuery ir GSAP išnaudoja GPU galios – pakyla iki 7 MB, o vidutiniškai abi naudoja apie 5.5 MB. Tuo tarpu Canvas – keičiasi nežymiai – išlieka apie 5 MB vidutiniškai.



28 pav. Antro kompiuterio GPU (MB) eksperimento rezultatai

Antro kompiuterio vėlavimo laiko eksperimento rezultatų suvestinė pateikta 29 pav. Matome, kad Canvas technologija ir šiame kompiuteryje pasirodė pakankamai prastai – kai nuotrauka didesnė kaip 12000px x 12000px matmenys vėlavimas labai jaučiasi – vidutiniškai 6 sekundžių, o su

14000px x 14000px matmenų nuotrauka vėlavimas dvigubas – vidutiniškai 12 sekundžių. Tuo tarpu geriausiai pasirodė WebGL - vėlavimo visai nesijaučia su didelėmis nuotraukomis – 0.2 sekundės tik kas visai nepasimato. Tuo tarpu JQuery ir GSAP pasirodė panašiai. JQuery su 14000px x 14000px suvėlavo 1.5 sekundės, o GSAP 1.3 sekundės., tačiau JQuery pasijuto vėlavimas jau su 10000px x 10000px matmenų nuotrauka, todėl GSAP šiuo atveju prieš JQuery laimi.



29 pav. Antro kompiuterio vėlavimo (sekundėmis) eksperimento rezultatai

5.8. Išvados

1. Atlikti keli eksperimentai, kurie iš skirtingų pusių įvertino naršyklėje veikiančių technologijų greitį. Svarbiausia - buvo įvertinta kaip skirtingos bibliotekos/technologijos geba susidoroti su didelio formato nuotraukomis.
2. Pirmas eksperimentas preliminariai nustatė greičius bibliotekų. JQuery prasčiausiai pasirodė - su 14fps greičiu, tuo tarpu likusios trys - GSAP, ThreeJS ir Canvas panašiai pasirodė - apie 100fps kiekviena. Taipogi, svarbi dalis ta, kad paieškos rezultatų daugiausia buvo atrasta su Canvas - 746 tūkst., o JQuery atsilieka nedaug - 536 tūkst. buvo atrasta. Tuo tarpu ThreeJS irgi daug - 447 tūkst, o GSAP mažiausiai - tik 16.6 tūkst. Iš to seka išvada, kad Canvas technologija ypač naudojama ir turi daug pavyzdžių.
3. Antru eksperimentu buvo detaliau įvertintos bibliotekos ir technologijos.
 - 3.1. Eksperimentas atliktas su dviem kompiuteriais.
 - 3.2. Pirmo kompiuterio, su integruota vaizdo plokšte, eksperimentas parodė, kad:
 - 3.2.1. Canvas technologija sunkiai susitvarko su nuotraukomis, kurių matmenys didesni kaip 12000px x 12000px. Su tokiomis ypač jaučiasi vėlavimai nors daug GPU neišnaudoja. Tuo tarpu JQuery jau su 8000px x 8000px jaučiasi vėlavimai ir tie vėlavimai didėja nors ir netaip ženkliai kaip su Canvas. Tuo tarpu WebGL kompiuteris pavežė dalinai - didesnes nuotraukas sumažindavo iki

8192px x 8192px matmenų. Ir WebGL su šia nuotrauka pasirodė panašiai kaip ir Canvas - jokio vėlavimo. GSAP biblioteka pasirodė stulbinamai gerai - net ir su 14000px x 14000px nuotrauka - vėlavimas tik 2 sekundės kas darbui netrukdo labai, o GPU ir CPU išnaudojo mažiau nei kitos.

3.3. Antro kompiuterio, su AMD Radeon vaizdo plokšte, eksperimentas parodė, kad:

3.3.1. WebGL technologijos nuotraukos dydis buvo palaikomas iki 16384px. Ir ši technologija pasirodė geriausiai - CPU išnaudojo daugiausiai - iki 60 procentų, GPU daugiausia - vidutiniškai 8.3 MB išnaudojo ir vėlavimas nelabai jautėsi. Su didžiausia nuotrauka tik 0.2 sekundės kas visai netrukde darbui. Canvas pasirodė prasčiausiai - jau nuo 12000px x 12000px matmenų nuotraukos vėlavimas buvo žymus - 6 sekundės, GPU išnaudojo tik 5MB, o CPU gan daug - vidutiniškai 40 procentų. Tuo tarpu GSAP antras buvo pagal geriausiai - tik 1.3 sekundės vėlavimas su didžiausia testuota nuotrauka. O JQuery - truputi nuo GSAP atsiliko - 1.5 sekundės su didžiausia nuotrauka vėlavimas.

6. IŠVADOS

1. Šiuo metu didėja skaitmeninių paslaugų pardavimai ir rinka - 2018 metais manoma, kad pasieks 187.7 milijardo dolerių. Tačiau įmonės siūliančios plataus formato nuotraukų spausdinimą vis dar siūlo tik įkelti nuotraukų failą - PDF ar kitu formatu.

2. HTML5 pristatytos technologijos praplėtė galimybes naršyklės rengyklių. Todėl didžioji dalis įmonių pradėjo kurti programas sukurtas su Canvas technologija dėl suderinamumo tarp naršyklių ir aparatūrinės įrangos pagreitinimo. Vis dėlto, WebGL technologija teikia daugiau vilčių dėl savo GPU panaudojimo nors tik apie 70 procentų kompiuterių palaiko šią technologiją.

3. Projekto metu sukurta programinė įranga leidžianti sukurti PDF formato nuotraukų failą naudojantis naršyklės rengykle.

4. Eksperimentu buvo įvertinta kaip skirtingos bibliotekos/technologijos geba susidoroti su didelio formato nuotraukomis:

- Canvas
 - Iki 10000px x 10000px matmenų nuotraukomis - vėlavimas 0 s.
 - Nuo 12000px x 12000px matmenų nuotraukomis vėlavimas didelis - vidutiniškai 7 s.
- HTML5 JQueryUI
 - Iki 10000px x 10000px matmenų nuotraukomis - vidutiniškai vėlavimas 0.6 s.
 - Nuo 12000px x 12000px matmenų nuotraukomis vėlavimas didelis - vidutiniškai 3.5 s.
- HTML5 GSAP
 - Iki 10000px x 10000px matmenų nuotraukomis - vidutiniškai vėlavimas 0 s.
 - Nuo 12000px x 12000px matmenų nuotraukomis vėlavimas didelis - vidutiniškai 1.6 s.
- HTML5 WebGL
 - Iki 10000px x 10000px matmenų nuotraukomis - vidutiniškai vėlavimas 0 s.
 - Nuo 12000px x 12000px matmenų nuotraukomis vėlavimas didelis - vidutiniškai 0.1 s.

7. LITERATŪRA

- [1] P. a. C. P. Scott W. Hardy, "Direporter," Direporter. Prieiga per internetą: <<http://www.direporter.com/state-of-the-industry/digital-imaging-reporter-s-state-industry-2015>>. [Žiūrėta 2017 04 02].
- [2] D. Haueter, "To protect your most important photos, you need to print them," InfoBlog, 27 02 2015.. Prieiga per internetą: <<http://blog.infotrends.com/?p=18256>>. [Žiūrėta 02 04 2017].
- [3] S. Hill, "Digital print market to reach \$187.7 billion by 2018," SmithersPira, 12 12 2013. Prieiga per internetą: <<http://www.smitherspira.com/news/2013/june/digital-printing-trends-market-analysis-to-2018>>. [Žiūrėta 02 04 2017].
- [4] "Brant InStore FILE UPLOAD FORM," Brant InStore ., Prieiga per internetą: <<http://brantinstore.com/upload-a-file/>>. [Žiūrėta 05 04 2017].
- [5] "Fotofabrikas - Nestandartinio formato nuotraukos įkėlimas," Fotofabrikas. Prieiga per internetą: <<http://www.fotofabrikas.lt/puslapis-2422299.html>>. [Žiūrėta 06 04 2017].
- [6] M. L. Andy Gryc, "Why HTML5 Is Becoming the HMI Technology of Choice," *QNX Software systems*, p. 8, 01 01 2012.
- [7] I. D. Agency, "HTML5 POPULARITY," INCORE. Prieiga per internetą: <<http://www.incore.com/Fortune500HTML5/>>. [Žiūrėta 06 04 2017].
- [8] J. Doyle, "Myth Busting: CSS Animations vs. JavaScript," css-tricks, 13 01 2014. Prieiga per internetą: <<https://css-tricks.com/myth-busting-css-animations-vs-javascript/>>. [Žiūrėta 08 04 2017].
- [9] Taopix, "Taopix," Taopix. Prieiga per internetą: <<http://taopix.com/>>. [Accessed 06 04 2017].
- [10] M. Murphy, "Adobe is finally (tacitly) admitting that Flash is obsolete," QUARTZ, 02 12 2016. Prieiga per internetą: <<https://qz.com/563235/adobe-is-finally-tacitly-admitting-that-flash-is-obsolete/>>. [Žiūrėta 06 04 2017].
- [11] Chili, "Chili," Chili. Prieiga per internetą: <<https://www.chili-publish.com/start>>. [Žiūrėta 06 04 2017].
- [12] S. F. a. J. Fulton, "HTML5 Canvas Basics of Canvas," in *HTML5 Canvas. Native Interactivity and Animation for the Web*, Sebastopol, O'Reilly Media, 2013, pp. 20-50.
- [13] C. Félizard, "Mozilla WebGL API," Mozilla, 23 02 2017. Prieiga per internetą: <https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial/Getting_started_with_WebGL>. [Žiūrėta 07 04 2017].

- [14] S. F. a. J. Fulton, HTML5 Canvas. Native Interactivity and Animation for the Web, Sebastopol: O'Reilly Media, 2013, p. 726.
- [15] "Shutterfly Large Format Prints," Shutterfly. Prieiga per internetą: <<https://www.shutterfly.com/prints/large-format-prints>>. [Žiūrėta 05 04 2017].
- [16] R. Hirsch, Light and Lens. Photography in the Digital Age, Burlington: Elsevier Inc., 2008.
- [17] H. Klein, "Impact of GPU Acceleration on Browser CPU Usage," Helge Klein, 10 12 2014. Prieiga per internetą: <<https://helgeklein.com/blog/2014/12/impact-gpu-acceleration-browser-cpu-usage/>>. [Žiūrėta 08 04 2017].
- [18] Taopix, "Why choose Taopix?," Taopix. Prieiga per internetą: <<http://taopix.com/why-choose-taopix>>. [Žiūrėta 09 04 2017].
- [19] "Pixami - Bringing stories to life," Pixami. Prieiga per internetą: <<http://www.pixami.com/>>. [Žiūrėta 09 04 2017].
- [20] WebGL, "WebGL - MAX_TEXTURE_SIZE," WebGL, Prieiga per internetą: <https://webglstats.com/webgl/parameter/MAX_TEXTURE_SIZE>. [Žiūrėta 12 04 2017].
- [21] ZoomBook, "ZoomBook - kurti naują knygą," ZoomBook, Prieiga per internetą: <<https://www.zoombook.lt/shelf/photobook/>>. [Žiūrėta 09 04 2017].
- [22] Snapfish, "Snapfish - about us," Snapfish , Prieiga per internetą: <<https://www.snapfish.com/photo-gift/about-us>>. [Žiūrėta 11 04 2017].
- [23] Microsoft, "Understanding Frames Per Second (FPS)," Microsoft, 06 08 2008 . [Online]. Prieiga per internetą: <<https://support.microsoft.com/en-us/help/269068/understanding-frames-per-second-fps>>. [Žiūrėta 12 04 2017].
- [24] L. Žitnik, "Frame rate," Mozilla, 02 09 2015. Prieiga per internetą: <https://developer.mozilla.org/en-US/docs/Tools/Performance/Frame_rate>. [Žiūrėta 12 04 2017].
- [25] J. Wiley, "Electronic Structure Calculations on Graphics Processing Units," in *Parallel Computing on Central Processing Units*, West Sussex, Sons Ltd., 2016, pp. 7-10.
- [26] J. Wiley, "Electronic Structure Calculations on Graphics Processing Units," in *GPU's Hardware to Software*, West Sussex, Sons Ltd., 2016, pp. 23-30.
- [27] K. Cagle, HTML5 Graphics with SVG & CSS3, New York: O'Reilly Media, 2012.
- [28] B. A. F. S. Peter Lubbers, Pro HTML5 Programming, New York: Springer Science Business Media, 2010.

- [29] GreenSock, "GreenSock - HTML5 Animation Speed Test," GreenSock, Prieiga per internetą: <<https://greensock.com/js/speed.html>>. [Žiūrėta 28 04 2017].
- [30] V. Haydin, "HTML5 Canvas: Performance and Optimization," Elekslabs Prieiga per internetą: <<http://elekslabs.com/2012/11/html5-canvas-performance-and.html>>. [Žiūrėta 07 05 2017].
- [31] B. Nickel, "How to get a pixel's x,y coordinate color from an image?," StackOverFlow, 5 07 2012. Prieiga per internetą: <<http://stackoverflow.com/questions/8751020/how-to-get-a-pixels-x-y-coordinate-color-from-an-image>>. [Žiūrėta 06 05 2017].
- [32] L. Vandevenne, "Lode's Computer Graphics Tutorial," lodev, 01 01 2004. Prieiga per internetą: <<http://lodev.org/cgtutor/floodfill.html>>. [Žiūrėta 07 05 2017].
- [33] C. Lindley, "jQuery Cookbook," in *User Interfaces with JQuery UI - Draggable*, Sebastopol, O'Reilly, 2010, pp. 315-327.
- [34] C. Lindley, "jQuery Cookbook," in *User Interfaces with JQuery UI - Resizable*, Sebastopol, O'Reilly, 2010, pp. 315-327.
- [35] J. Marzullo, "Image Resize with GSAP Draggable + basic resizing TEST," Codepen, Prieiga per internetą: <<http://codepen.io/jonathan/pen/LEqLbO>>. [Žiūrėta 30 04 2017].
- [36] E. Drowel, "HTML5 Canvas Drag, Drop, and Resize Images," HTML5Canvas tutorials, Prieiga per internetą: <<http://www.html5canvastutorials.com/labs/html5-canvas-drag-and-drop-resize-and-invert-images/>>. [Žiūrėta 23 04 2017].
- [37] ThreeJS, "ThreeJS," ThreeJS, Prieiga per internetą: <<https://threejs.org/>>. [Žiūrėta 23 04 2017].
- [38] S. Porter, "Canvas Performance Test," Smashcat, 11 01 2011. Prieiga per internetą: <http://www.smashcat.org/av/canvas_test/>. [Žiūrėta 28 04 2017].
- [39] NetMarkteShare, "Desktop Browser Version Market Share," Net Applications, 01 03 2017. Prieiga per internetą: <<https://www.netmarketshare.com/browser-market-share.aspx?qprid=2&qpcustomd=0>>. [Žiūrėta 28 04 2017].
- [40] J. Beecher, "Photokaboom - Print Size to Pixels Calculator," Photokaboom, Prieiga per internetą: <http://www.photokaboom.com/photography/learn/printing/1_calculators.htm#Pixels>. [Žiūrėta 28 03 2017].
- [41] R. Sheppard, EPSON Complete guide to DigitalPrinting, New York: Lark Books, 2008.
- [42] BRANTInStore, "BRANTInStore," BRANTInStore, Prieiga per internetą: <<http://brantinstore.com/upload-a-file/>>. [Žiūrėta 28 03 2017].
- [43] Mr.doob, "JavaScript Performance Monitor," mrdoob, Prieiga per internetą:

- <<https://github.com/mrdoob/stats.js/>>. [Žiūrėta 20 04 2017].
- [44] "Fotofabrikas - nestandartinė nuotrauka," Fotofabrikas, Prieiga per internetą: <<http://www.fotofabrikas.lt/items/Nestandartinio-formato-nuotraukos/Asmenines-nuotraukos/16d.1-Nestandartine-nuotrauka.html>>. [Žiūrėta 06 04 2017].
- [45] "Fotor," Fotor, Prieiga per internetą: <<http://www.fotor.com/>>. [Žiūrėta 28 03 2017].
- [46] "Convert CM to Inches," Prieiga per internetą: <<http://www.inches-to-cm.com/>>. [Žiūrėta 28 03 2017].
- [47] Tizen, "HTML5 Canvas," Tizen, 01 07 2013. Prieiga per internetą: <https://developer.tizen.org/dev-guide/2.4/org.tizen.guides/html/web/w3c/graphics/canvas_w.htm>. [Žiūrėta 08 04 2017].
- [48] MixBook, "MixBook - print sizes," MixBook, Prieiga per internetą: <https://www.mixbook.com/home-decor/poster-prints?utm_referrer=https%3A%2F%2Fwww.mixbook.com%2Fphoto-prints>. [Žiūrėta 11 04 2017].
- [49] C. Elvidge, "Night Lights 2012 - Flat map," NASA, 18 04 2012. Prieiga per internetą: <<https://earthobservatory.nasa.gov/NaturalHazards/view.php?id=79765>>. [Žiūrėta 20 04 2017].
- [50] yongsup, "Resize Svg Image Using JQuery," StackOverflow, Prieiga per internetą: <<http://stackoverflow.com/questions/36612215/resize-svg-image-using-jquery>>. [Žiūrėta 23 04 2017].
- [51] J. Spiess, "Accumulation Along the Edge: Final Submissions," 24 03 2013. Prieiga per internetą: <<https://johnspiess.wordpress.com/2013/05/24/accumulation-along-the-edge-final-submissions/>>. [Žiūrėta 28 04 2017].

8. TERMINŲ IR SANTRUMPŲ ŽODYNAS

1. **Didelio formato nuotrauka** – nuotrauka, kurios matmenys didesnės kaip 3543px x 5315px, kadangi “fotofabrikas” nurodo šį dydį kaip pradinį tašką plataus formato nuotraukų (30x45 cm [44], o su 3543px x 5315px ir 300 rezoliucija kokybiškai tokia nuotrauka išspausdinama [45]).
2. **PDF (angl. “Portable Document Format”)** - yra atviro standarto formatas, skirtas elektroniniam dvimačiam dokumentui atvaizduoti.
3. **Adobe Photoshop** – Adobe įmonės sukurta programinė įranga, kuri leidžia profesionaliai kurti/keisti/manipuliuoti nuotraukas.
4. **Adobe InDesign** – Adobe įmonės sukurta programinė įranga leidžianti kurti spausdinimui naujienlaiškius ir kitus spaudinius.
5. **HTML5 (angl. “Hypertext markup language”)** – yra hiperteksto žymėjimo kalba, kuria aprašomi internetiniai puslapiai.
6. **HTML5 (div)** – “div” tai vienas iš pagrindinių HTML5 elementų. Kadangi WebGL ir Canvas norint atvaizduoti reikia “canvas” elementų, tuo tarpu “div” elementas įprastam HTML5 naudojamas.
7. **Canvas** – elementas, kuris leidžia pasitelkus JavaScript piešti paveikslukus, tekstą ir kt.
8. **WebGL (angl. “Web Graphics Library”)** – yra JavaScript API, kuris leidžia piešti 3D grafiką kiekvienoje naršyklėje nenaudojant įskiepių.
9. **OpenGL (angl. “Open Graphics Library”)** – yra kompiuterių industrijoje standartinis API, kuris leidžia apibrėžti 2-D ir 3-D grafinius paveikslukus.
10. **SVG (angl. “Scalable vector graphics”)** – naudojamas aprašyti vektorinę grafiką XML formate.
11. **CSS3 (angl. “Cascading Style Sheets”)** – stiliaus kalba skirta aprašyti HTML5 elementus.
12. **JavaScript** – programavimo kalba. Šiame kontekste turima omeny, kad HTML5 klientinės pusės kalba.
13. **CPU (angl. “Central processing unit”)** – centrinis apdorojimo blokas. Šis blokas apdoroja visas instrukcijas, kurias gauna iš techninės įrangos ir programinės veikiančios kompiuteryje.
14. **GPU (angl. “Graphical processing unit”)** – grafinis apdorojimo blokas. Šis blokas naudojamas pagreitinti 2D ir 3D paveikslukų kūrimą.
15. **RAM (angl. “Rapid access memory”)** - kompiuterio operatyvioji atmintinė.
16. **XML (angl. “Extensible Markup Language”)** - yra bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba.

17. **FPS** (angl. “*Frames per second*”) – kadrų skaičius per sekundę. Bendrai, norint išvengti trūksčiojančio judesio (angl. “*jerky motion*”) rekomenduojamas minimalus FPS dydis yra apie 30. Turinyje, kuriame keičiasi greit vaizdai geriausia yra 60 FPS [23].
18. **Vėlavimas** (angl. “*Lag*”) – pastebimas delsimas tarp vartotojo veiksmų ir sistemos atsako (perpiešimo).
19. **Trūkčiojantis judesys** (angl. “*Jerky motion*”) – pastebimas netolygus perpiešimas sistemos vaizdo atliekant tam tikrą veiksmą.
20. **UML** (angl. “*Unified modeling language*”) – vieninga modeliavimo kalba. Modeliavimo ir specifikacijų kūrimo kalba, skirta specifikuoti, atvaizduoti ir konstruoti objektiškai orientuotų programų dokumentus.
21. **API** (angl. “*Application programming interface*”) – Aplikacijų programavimo sąsaja. Tai sąsaja, kurią suteikia kompiuterinė sistema, biblioteka ar programa tam, kad programuotojas per kitą programą galėtų pasiekti jos funkcionalumą ar apsikeistų su ja duomenimis.
22. **DPI** (angl. “*Dots per inch*”) – taškai per colį. Yra matavimas erdvinio spausdinimo taškų tankio. Iš esmės tai skaičius individualių taškų, kurie gali būti išdėstyti per tam tikrą 1 colio (2.54 cm) erdvę [16].

9. PRIEDAI