



Web-sivujen vektorianimaatiot

Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutus

Syksy 2023

Oskari Saarela

TIIVISTELMÄ

Opinnäytetyön tavoitteena oli kertoa mitä on vektorigrafiikka ja selvittää tehokkain tapa SVG-animaatioiden tekemistä varten. Opinnäytetyössä vertaillaan yhtä animointialustaa sekä kolmea eri ohjelmointikieltä toisiinsa selvittääkseen animaatio mahdollisuudet, rajoitukset sekä oppimiskäyrän jokaisella tavalla.

Opinnäytetyön teoriaosuudessa kerrotaan mitä on skaalautuva vektorigrafiikka ja SVG-animaatio tavoista. Käytännönsosassa tehdään animaatioalustalla sekä kolmella eri ohjelmointikielillä SVG-animaatio, joissa suunnitellaan käyttötarkoitus, graafinen suunnittelu kuralle sekä animaatiot. Luotu SVG-tiedosto on logo, mainos tai muu visuaalinen elementti suunniteltuna web-sivulle.

Jokaisen käytännönsosuden tavasta kerrotaan henkilökohtaiset kokemukset sekä huomautukset hyödyt SVG-animaatioita varten. Lopullisessa vertailussa selviää tehokkain tapa SVG-animaatioiden tekemiseen työksi sekä sama vaihtoehto ohjelmointikielen käytössä. Lopullisissa tuloksissa tehokkaimmaksi tavaksi päädyin SVGator-animointialustaan sekä ohjelmointikielissä ECMAScript on tehokkain. SVGator-animointialusta on SVG-tiedostoja työtä tekeväälle nopein ja helpoin tapa luoda kyseisiä tiedostoja.

Avainsanat SVG, vektorigrafiikka, animaatiot

Sivut 29 sivua ja liitteitä 2 sivua

ABSTRACT

The goal of the thesis was to explain what vector graphics are and to find out the most efficient way to make SVG animations. In the thesis, one animation platform and three different programming languages are compared to each other in order to find out the animation possibilities, limitations and the learning curve in each way.

The theory part of the thesis explains what scalable vector graphics and SVG animation are. In the practical part, an SVG animation is made using an animation platform and three different programming languages, where the purpose of use, graphic design for the image and animations are planned. The generated SVG file is a logo, advertisement or other visual element designed for a web page.

Personal experiences and noticed benefits for SVG animations are explained about the method of each practical part. The final comparison reveals the most efficient way to make SVG animations work, as well as the same option when using a programming language. In the final results, I ended up with the SVGator animation platform as the most effective method and in programming languages ECMAScript is the most effective. The SVGator animation platform is the fastest and easiest way for those working with SVG files to create those files.

Keywords SVG, vector graphic, animations

Pages 29 pages and appendices 2 pages

Sanasto

Vektorigrafiikka	Tietokonegrafiikka, jonka ominaisuudet perustuvat kuvan koordinaatteihin.
SVG	Kuvatiedostotyyppi, lyhenne sanoista scalable vectorgraphics
HTML	Web-merkintäkieli, jolla vaikutetaan sivun elementtien asetteluun ja järjestykseen.
Ohjelmointikieli	Tietokoneella kirjoitettava kieli jolla ohjelmoidaan sivuja, sovelluksia ja muuta.
Web-standardi	Web-pohjaisen sisällön luominen ja tulkitseminen.
World Wide Web Consortium (W3C)	Ylläpitää ja kehittää web-standardeja.
Animaatiot	Kuvan liikehdintä ja muuttuminen ajan kuluessa.
ECMAScript	Ohjelmointikieli joka on yleiseen tarkoitukseen luotu scriptauskieli.
CSS	Ohjelmointikieli web-sivun tyyllittelyä varten.
SMIL SVG	Ohjelmointikieli luotu multimediatekijöitä varten.

Sisällys

1	Johdanto.....	1
2	Skaalautuva vektorigrafiikka	2
2.1	Alkuperä.....	2
2.1.1	Vektorigrafiikka	2
2.1.2	XML.....	3
2.2	SVG-kuvan piirtämisen idea	3
2.3	SVG-animaatiot	4
3	SVG-tiedostot ohjelmointikielillä	8
3.1	Visual Studio Code.....	8
3.2	Ohjelmointikieli tavat	8
4	SVG web-sivuilla.....	9
4.1	Visuaalinen markkinointi	9
4.2	Visuaalinen viestintä	10
5	Kehitysprojektin kuvaus	11
6	SVGator.....	12
6.1	SVGator animointialustan työkalut	12
6.1.1	Canvas ja piirtotyökalut	12
6.1.2	Elementtien muokkaus	14
6.1.3	Animaattori.....	15
6.2	Animointialustan toteutus	15
7	Toteutukset mahdollistamaan vertailu	17
7.1	CSS	17
7.2	ECMAScript.....	20
7.3	SMIL SVG.....	22
8	Toteutusten vertailu	25
8.1	Animointialusta verrattaessa ohjelmointikieleen ja graafiseen alustaan.....	25
8.2	Ohjelmointikielten vertailu	26
9	Yhteenveto.....	27
	Lähteet.....	28

Kuvat, taulukot, ohjelmakoodit ja liitteet

Kuvat

Kuva 1	SVG:n piirtomallin idea.....	3
Kuva 2	Tekstin ja visuaalisuuden erot havaitessa web-sivun sisältö	10
Kuva 3	Canvaksen hallinta ja piirtotyökalut.....	13
Kuva 4	Elementin sijainnin, koon ja värien muokkaus tarkasti.....	14
Kuva 5	Animaattorin hallinta.....	15
Kuva 6	SVGator SVG-tiedosto PNG-kuvana esikatseluna.....	16
Kuva 7	CSS-ohjelmointikielen SVG-tiedosto PNG-kuvana esikatseluna.....	18
Kuva 8	SVG-tiedoston sisältö CSS-ohjelmointikieli tavalla sisältäen kaiken graafisen sisällön sekä elementtien nimeäminen.....	18
Kuva 9	SVG-tiedoston sisältö CSS-ohjelmointikieli tavalla sisältäen CSS-tyylittelyn animaatioita varten.	19
Kuva 10	ECMAScript-ohjelmointikielen SVG-tiedosto PNG-kuvana esikatseluna.....	20
Kuva 11	SVG-tiedoston sisältö ECMAScript-ohjelmointikieli tavalla sisältäen jakamisikonin sekä animaation linkit.....	21
Kuva 12	SVG-tiedoston sisältö ECMAScript-ohjelmointikieli tavalla sisältäen scriptin, joka mahdollistaa interaktiivisen animaation.....	21
Kuva 13	SMIL SVG-kuvauskielen SVG-tiedosto PNG-kuvana esikatseluna.....	23
Kuva 14	SVG-tiedoston sisältö SMIL-kuvauskieli tavalla sisältäen postilaatikon rungon sekä kirjeen ja sen animoinnin.	23
Kuva 15	SVG-tiedoston sisältö SMIL-kuvauskieli tavalla sisältäen postilaatikon sivustan sekä lipun ja sen animoinnin.	24

Taulukot

Taulukko 1	Erilaisia SVG-elementtejä ja niiden toiminnot.....	4
Taulukko 2	Erilaisia komentoja animaatioiden tekemiseen sekä niiden kuvaukset.....	7

Ohjelmakoodit

Ohjelmakoodi 1	CSS-ohjelmointikielen käyttö SVG-tiedostossa.....	6
Ohjelmakoodi 2	ECMAScript-ohjelmointikielen käyttö SVG-tiedostossa.....	6
Ohjelmakoodi 2	SMIL SVG-ohjelmointikielen käyttö SVG-tiedostossa.....	7

Liitteet

Liite 1	Aineistonhallintasuunnitelma
Liite 2	SVG-tiedostojen esikatselu linkkien kautta

1 Johdanto

Skaalautuvia vektorigrafiikka-animaatiokuvia eli SVG-kuvia hyödynnetään nykyaikana logoina, ohjelmistojen ja pelien kuvakkeina, animaatiosarjoissa sekä web-sivun yleisilmeen kohentamiseen ja mainostamisen keinona. SVG-kuvien suosio on kasvanut viime vuosina paljon ja niitä hyödynnetään web-sivuilla visuaalisina elementteinä.

Opinnäytetyössä luodaan skaalautuvia vektorigrafiikka-animaatioita kuville, joita voi hyödyntää mainoksina, logoina tai muihin visuaalisiin tarkoituksiin. Tärkeässä osassa on myös vertailla eri ohjelmointikielten tapoja valmiiseen alustaan animaatioiden tekemistä varten sekä tuoda esille eri tapojen hyödyt. Opinnäytetyö käsittelee teoriaa SVG:stä, vektorigrafiikan merkityksestä sekä animaatioiden hyödyistä. Työn tutkimuksellinen osuus käsittelee eri ohjelmointikielten käyttämisen sekä valmiin alustan, jota voi käyttää ilman ohjelmointiosaamista animaatioiden tekemiseen.

Ohjelmointikieli-tyylit ovat CSS ja ECMAScript sekä valittu animointialusta SVGator. Animointialusta SVGatorin valinta perustui yleisestä suosiosta, hakukoneen ehdotuksista sekä alusta on ilmainen ja nopeasti käyttöön otettava. Alusta tarjoaa seitsemän päivän ilmaisen kokeilujakson, jonka takia SVGator soveltuu paremmin aloitus alustaksi. Työssä on myös tarkoitus tutkia pystyykö SMIL SVG:tä hyödyntämään nykyaikana tehokkaasti.

Alla opinnäytetyössä käsiteltävät tutkimuskysymykset:

- o Mitä on vektorigrafiikka?
- o Mitä eri tekniikoita voi hyödyntää animaatioiden tekemisessä?
- o Miten vektorianimaatioita voidaan hyödyntää web-sivuilla?
- o Mikä on hyvä valinta ohjelmointikieleksi vektorianimaatioiden tekemiseen?

2 Skaalautuva vektorigrafiikka

Skaalautuva vektorigrafiikka eli SVG on web-pohjainen tiedostotyyppi, joka luo kuvia hyödyntäen viivoja, tekstejä, muotoja sekä värejä. SVG on monipuolinen tiedostotyyppi, joka tarjoaa vähemmän tallennuskokoa, mahdollisuudet animaatioihin, tarkemmin nähtävät kuvat sekä mahdollisuuden tallentaa tietoa kuvaan. SVG:n tarkkuus perustuu siihen, että kuvan sisältämät muodot, viivat, tekstit ja värit luetaan kuvan eri kohdista itsenäisinä objekteina, jonka seurauksena kuvan kokoa voi muuttaa ilman, että tarkkuus heikkenee.

2.1 Alkuperä

SVG-tiedoston kehitys aloitettiin vuonna 1999, kun World Wide Web Consortium (W3C) halusi kehittäjien luoda uuden vektorigrafiikka formaatin. SVG:n ensimmäinen versio SVG 1.0 tuli julki vuonna 2001 ja sen jälkeen päivitetyn versio SVG 1.1 vuonna 2003. Nykyään SVG:n web-standardina toimii SVG 2 vuodesta 2018 lähtien, joka omaa piirteitä edellisestä versiosta sekä mahdollisti yhteensopivuuden uusien web-standardien kanssa. SVG:lle on myös määritelty kaksi erilaista mobiiliprofiilia: SVG Basic ja SVG Tiny, joiden perusidea on helpottaa erilaisten SVG-sovellusten toteuttamista mobiililaitteisiin. SVG:n suosio alkoi nousta vasta vuonna 2017, kun ihmiset alkoivat nähdä SVG:n hyödyt nykyaikaisissa web-selaimissa. SVG-tiedostot ovat nyt laajasti käytettyjä web-sivuilla, koska useimmat selaimet ja piirto-sovellukset hallitsevat paremmin vektorityyppi-tiedostot. (MDN Web Docs, 2023, SVG Introduction, Flavors of SVG)

2.1.1 Vektorigrafiikka

Vektorigrafiikka on tietokoneella luotua grafiikkaa, jonka kuvan sisältämät eri objektit, kuten muodot ja värit luetaan kuvassa matemaattisina lausekkeina. Matemaattisten lausekkeiden tiedot voivat sisältää objektin sijainnin, koon ja värin. Vektorigrafiikka mahdollistaa kuvan skaalauksen eli koon muuttamisen rajattomasti ilman, että kuvan tarkkuus kärsii. (Nykänen, 2007, s. 14)

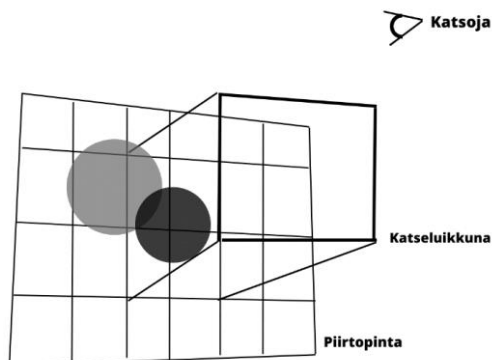
2.1.2 XML

Extensible Markup Language (XML) on merkintäkieli, joka tallentaa tiedon rakenteen ja sen merkityksen. XML-koodit jakavat tiedot eri osiin sekä määrittävät eri osien tunnisteet. SVG hyödyntää pohjanaan XML-tekstiformaattia, jonka avulla voidaan luoda kuvaan muotoja, tekstejä ja värejä. SVG luo jokaiselle objektille oman tunnisteen, johon saa tallennettua tietoa. (Nykänen, 2007, s. 14)

2.2 SVG-kuvan piirtämisen idea

SVG-kuvan piirtäminen ohjelmoinnin avulla, perustuu muotojen lisäämiseen päällekkäin, jolloin saataisiin haluttu lopputulos. Erilaisten viivojen asettelu loisi kuvioita ja muotoja, kuten tavallisen piirtämisen ajatuskin on. Muodot esitetään kaksiulotteisina koordinaatiston avulla, jotka tulevat esille taustalle. Kuvassa 1 näkyy SVG-tiedoston piirtämisen idea.

Kuva 1. SVG:n piirtomallin idea (Nykänen, 2007, s. 15)



SVG:ssä paperia vastaa **piirtopinta** (canvas) ja taulunkehystä **katseluikkuna** ja siihen keskeisesti liittyvä (suhteellinen) **koordinaatisto** (viewport). Kun mukaan lisätään mahdollisuus ryhmitellä kuvioita ja suorittaa erilaisia kuvankäsittelyoperaatioita loputtamattomasti "apupaperia" hyödyntäen, on mielikuva kohtuullisen osuva. (Nykänen, 2007, s. 15)

2.3 SVG-animaatiot

SVG-animaatio tarkoittaa kuvan muutosta ajan perusteella. Animaatiot voivat muuttaa kuvan sijaintia, muotoa tai kokoa sekä piirto-ominaisuuksia. Animaation ominaisuuksiin voidaan vaikuttaa ajastamalla tai animaatio voi reagoida tapahtumiin. SVG:n skaalautuvuus mahdollistaa animaatioiden sujuvan muutoksen, että jokaisen animaatio kohtauksen tarkkuus pysyy samana.

SVG-animaatioiden saavuttamiseksi on kolme eri tapaa kirjoittaa animaation toiminto, jotka ovat CSS, ECMAScript ja SMIL. SVG:llä animoitaviin elementteihin kuuluu erilaisia muotoja, käyttötapoja sekä tekstejä. Yleisimmät SVG-elementit näkyvät taulukosta 1, jossa on elementin nimi ja sen toiminto.

Taulukko 1. Erilaisia SVG-elementtejä ja niiden toiminnot (SVG element reference, 2022)

Elementti	Toiminto
<g>	Elementtien ryhmittely yhdeksi elementiksi.
<defs>	Graafisen objektin tallentaminen myöhemmäksi, Käytetään <use> elementillä.
<image>	Kuvien käyttö SVG-tiedoston sisällä.
<switch>	Arvioi tarvittavat toiminnot, laajennukset ja kielen haluttuun elementtiin järjestyksessä ja hahmottaa ensimmäisen, joka on totta ohjelmakoodissa.
<path>	SVG:n voimakkain elementti, jolla mahdollista viivojen, käyrien tai kaarien luomisen pisteiden avulla.
<rect>	Luo suorakolmio muodon.
<circle>	Luo ympyrä muodon.

<code><ellipse></code>	Luo soikio muodon.
<code><line></code>	Luo viiva muodon.
<code><polyline></code>	Luo murtoviivoja pisteiden perusteella.
<code><polygon></code>	Luo monikulmio muodon.
<code><text></code>	Luo graafisen elementin, joka sisältää tekstiä.
<code><clipPath></code>	Mahdollistaa polun rajaamista varten.
<code><mask></code>	Objekti sen sisällä muodostetaan osaksi taustaa.
<code><a></code>	Luo hyperlinkin web-sivuja, tiedostoja tai teidosto-sijainteja varten.
<code><foreignObject></code>	Hyödyntää muuta XML-merkintäkieltä, yleisimmin HTML.

CSS-ohjelmointikielen käyttö perustuu tyyllittelyn asettamiseen, jossa voidaan asettaa haluttu elementti ja mitä muutoksia sille tulisi tehdä. Ohjelmakooditekstissä 1 asetetaan CSS-tyyllittely sekä keyframesilla haluttu elementti, joka on rot. Animaatio tapahtuu sulkeiden sisällä, joka tässä tapauksessa on, että elementti pyörii 360 astetta yhden sekunnin ajan sekä animaatio on loputon.

Ohjelmakoodi 1. CSS-ohjelmointikielen käyttö SVG-tiedostossa.

```
<style type="text/css">
  @keyframes rot {
    from { transform: rotate(0deg); }
    to { transform: rotate(360deg); }
  }
  .rot { animation: rot 1s linear infinite; }
```

ECMAScript-ohjelmointikieli tavalla SVG-animaatiot voi muuttaa elementit erilaisiksi muuttujiksi sekä interaktiivisiksi käyttäjän klikkausten tai web-sivun muiden elementtien perusteella. Ohjelmakooditekstissä 2 tehdään samanlainen animaatio kuin CSS-tavalla, mutta ECMAScripti mahdollistaa enemmän asetuksia elementin hallintaan.

Ohjelmakoodi 2. ECMAScript-ohjelmointikielen käyttö SVG-tiedostossa.

```
<script type="text/ecmascript">
function rotate(evt) {
  var object = evt.target.ownerDocument.getElementById(rot);
  setInterval(function () {
    var now = new Date();
    var milliseconds = now.getTime() % 1000;
    var degrees = milliseconds * 0.36; // 360 degrees in 1000 ms
    object.setAttribute('transform', 'rotate(' + degrees + ')');
  }, 20);
}
</script>
```

SMIL-kuvauskieli kirjoitetaan valitsemalla haluttu animaatio tyyli ja sen jälkeen kaikki asetukset siihen liittyen. Valitun animaation sisälle valitaan haluttu elementti, animaation tyyppi, asteet sekä ajastus. Ohjelmakooditekstissä 3 animaatio on samanlainen kuin CSS ja ECMAScriptin animaatio. Ohjelmakoodin jälkeen taulukossa on SMIL-kuvauskielen eri animaatioiden komennot.

Ohjelmakoodi 3. SMIL SVG-ohjelmointikielen käyttö SVG-tiedostossa.

```
<animateTransform
  attributeName="transform"
  attributeType="XML"
  type="rotate"
  from="0"
  to="360"
  begin="0s"
  dur="1s"
  repeatCount="indefinite"/>
```

(Wikipedia, 2010, SVG_animation. Examples)

Taulukko 2. Erilaisia komentoja animaatioiden tekemiseen sekä niiden kuvaukset (Nykänen, 2007, s. 148)

Komento	Kuvaus
animate	Objektin yksittäisen valitun, lähinnä numeerisen ominaisuuden animointi
Set	Objektin yksittäisen ominaisuuden asettaminen; mahdollistaa paloittain etenevät animaatiot
animateMotion	Objektin liikuttaminen polkua pitkin, joka asetetaan sijaintipisteinä, jotka tulee kulkea.
animateColor	Objektin väriominaisuuksien animointi
animateTransform	Objektin geometristen ominaisuuksien animointi

3 SVG-tiedostot ohjelmointikielillä

Opinnäytetyössä käytännönosaa varten hyödynnetään Visual Studio Codea (VS Code) ja sen eri ohjelmointikielitäpoja SVG-tiedostojen luomista varten. VS Code:lla hyödynnetään kolmea eri ohjelmointikieltä, jotka ovat CSS, ECMAScript ja SMIL SVG. Näillä eri tavoilla luodaan animoitu SVG-kuva tiedosto, joka on logo, ikoni, web-sivun tausta tai mainostus sekvenssiä.

3.1 Visual Studio Code

VS Code on kevyt ja voimakas lähdekoodieditori, jolla voidaan ohjelmoida eri ohjelmointikielillä mitä vain pystyy. VS Code:lle on sisäänrakennettuna sopivuus JavaScript, TypeScript ja Node.js ohjelmointiin sekä itse lähdekoodi editorin voi ladata tietokoneen työpöydälle sovellukseksi Windowsille, MacOS:lle ja Linuxille. VS Codea varten on ladattava laajennus SVG:tä varten, joka mahdollistaa SVG-tiedostojen luomisen. VS Code sisältää laajennuksia ja työkaluja myös, jos haluaa muuttaa SVG-tiedostoja toisiksi tiedostoiksi tai SVG-kuvien laajempaan esikatseluun editorissa. (Visual Studio Code, 2023)

3.2 Ohjelmointikieli tavat

CSS-ohjelmointikieli on myös samantapainen kuin XML-merkintäkieli, mutta se on suunniteltu web-sivun elementtien tyyllittelyyn, kuten asettelu, värit ja miltä elementit tulevat näyttämään sivulla.

ECMAScript-ohjelmointikieli on yleiseen tarkoitukseen luotu scripti-kieli, jonka on luonut Javascript. Sen luomisen tarkoitus on varmistaa eri web-sivujen toimivuus eri web-selaimissa. SVG:ssä voidaan hyödyntää ECMAScriptiä, joka mahdollistaa muuttujien käyttämisen elementteinä sekä mahdollistaa interaktiiviset animaatiot SVG-tiedostossa.

SMIL SVG-kuvauskieli on SVG-tiedostoja varten luotu ohjelmointikieli, joka on animaatioita varten. Ohjelmointikielen tekniikka on vanha, jonka hyödyntämistä tehokkaasti tutkitaan opinnäytetyössä nykyaikana.

4 SVG web-sivuilla

SVG-tiedostojen suurimmat käyttökohteet web-sivuja varten ovat yritysten logot, animoidut SVG-kuvat sekä kaaviot tai muut graafiset kuvakkeet. SVG-kuvilla parannetaan yritysten web-sivujen visuaalista ilmettä sekä kuvataan yritysten toimintamallia, jonka tarkoituksena on mainostaa sivuston kävijöille. SVG-tiedostojen skaalautuvuus sekä vähäinen tiedoston koko mahdollistavat monen tiedoston hyödyntämisen sivustolla, joka on visuaaliselta laadulta asiakasta miellyttävä. SVG-tiedostojen käyttö web-sivuilla parantaa myös näkyvyyttä hakukoneissa, koska pystyt tallentamaan kuvan tietoihin avainsanoja kuvaan tai web-sivuun liittyen. Nykyaikana SVG-tiedoston yhteensopivuus web-selaimien kanssa onnistuu kaikkien, paitsi muutamien vanhempien selain versioiden kanssa, kuten IE 6-8 sekä Android 2.1–2.3.

4.1 Visuaalinen markkinointi

Visuaalisuus web-sivustoilla on iso osa markkinointia uusille asiakkaille. Markkinoinnin perinteinen tarkoitus on vaikuttaa tunteisiin, johon voidaan vaikuttaa erilaisilla visuaalisilla elementeillä, kuten väreillä ja kuvilla sekä muilla visuaalisilla keinoilla. Markkinoinnilla pyritään tehdä tuotteesta tai palvelusta mahdollisimman haluttu, jonka visuaalisuus tuo parhaiten esille. Visuaalisuuden tarkoitus on antaa hyvä ensivaikutelma, joka mahdollisesti saa asiakkaan kiinnostumaan yrityksen palveluista tai tuotteista. (Huttunen, 2020, Visuaalinen markkinointi)

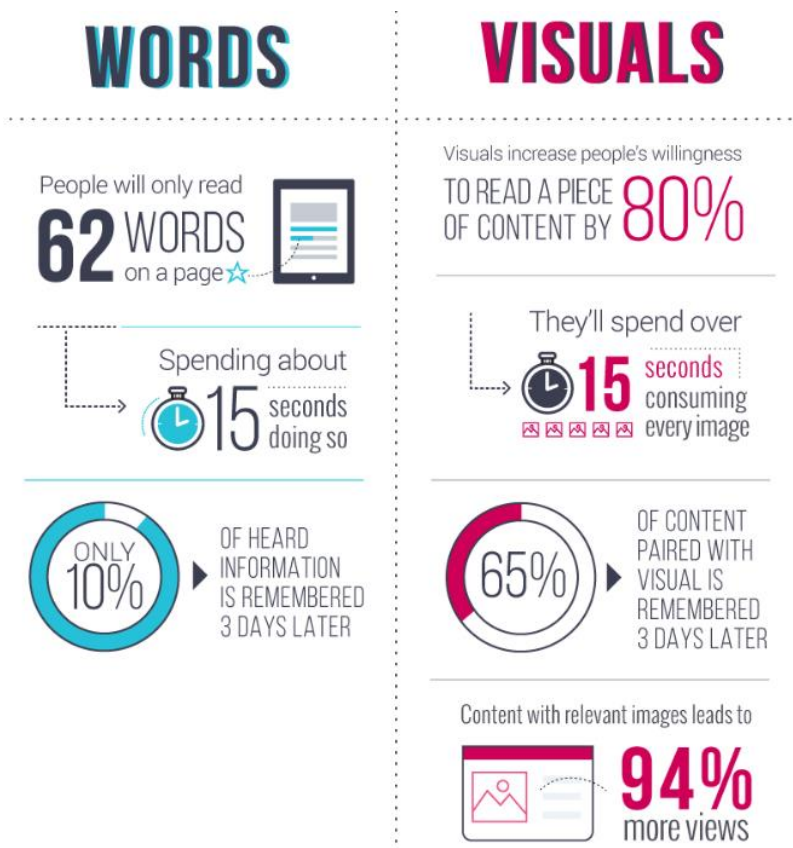
Tutkimusten mukaan ihminen päättää 0,05 sekunnissa jääkö hän päätymilleen nettisivuille. Tuossa ajassa emme ehdi rekisteröidä kuin visuaalisen kokonaisuuden eli ainoa tapa vaikuttaa ensivaikutelmaan on visuaalisuus. Jos potentiaalinen asiakas päätyy hakukoneen kautta sivuillesi, ja ne ovat hyvin sekavat tai eivät anna ammattimaista kuvaa, voit hetkessä menettää tämän hänet kokonaan. (Huttunen, 2020, Visuaalinen markkinointi, Ensivaikutelma muodostuu visuaalisuudesta)

4.2 Visuaalinen viestintä

SVG-tiedostoja pystyy hyödyntämään myös visuaalisen viestinnän keinoin, jonka tarkoituksena markkinoinnista eroten on informoida haluttua kohderyhmää. Viestintä on asiallista ja faktapohjaista informaatiota, joka pyrkii kertomaan vastaanottajalle tietoa. Viestinnällä pyritään edesauttamaan organisaation strategisten tavoitteiden toteuttamista sekä visuaalisudella pyritään välttämään turhia kuvia, jolloin sisältö olisi asiallista. (Nelimarkka, 2017, Visuaalinen viestintä)

Kuvalla voi olla konkreettinenkin rooli siinä, miten viesti vastaanotetaan. Tekstissä esiintuotu, vaikeasti hahmotettava fakta voi avautua ja konkretisoitua kuvassa tai graafissa. Toisaalta taas kuvituskuvat, abstraktimmatkin, ovat keino luoda haluttua brändimielikuvaa ja mahdollisesti toimia näkyvyyden herättäjänä tai aikaansaajana. (Nelimarkka, 2017, Visuaalinen viestintä, Kuvat viesti jo itsessään)

Kuva 2. Tekstin ja visuaalisuuden erot havaitessa web-sivun sisältö (Chandra, 2020, Why Marketers Need a Visual Influence Strategy, Why Visuals Matter)



5 Kehitysprojektin kuvaus

Opinnäytetyön käytännön osan toteutin kehitysprojektina, jonka menetelmänä on Scrum-projektinhallinta tapa. Jokainen käytännön osa on tehty viikon sprinteissä koostuen suunnittelusta, graafisesta-suunnittelusta ja animointien implementoinnista.

Jokaista käytännön osan varten on suunniteltu käyttötarkoitus sekä mitä animaatioita tai hyötyjä pitäisi esitellä. Ohjelmointikielien SVG-tiedostojen graafiseen suunnitteluun on hyödynnetty Figma graafista sivustoa, jolla kuvat elementit on piirretty, joka kostuu muutamasta erillisestä elementistä animointia varten. Jokaisella tavalla on suunniteltu kuvan tarkoitus, joka voisi olla logo, mainos tai web-sivulle ikoni. Graafisessa-suunnittelussa pyritään luomaan kuva, joka koostuu erillisistä elementeistä, joita voisi animoida. Tarkoituksena on myös tehdä kuvasta tarpeeksi yksinkertainen, että henkilö näkee mitä kuvassa on.

Animaatioiden suunnittelussa tarkoituksena on hyödyntää, jokaisen tavan vahvuuksia sekä mitä tapa mahdollistaa. SVGator-animointialustalla on tarkoitus tutkia alustan tarjontaa sekä luoda kaikista vaativin SVG-tiedosto graafisten ominaisuuksien ja animaatioiden kannalta. CSS- ja SMIL SVG-ohjelmointikielillä pyritään luomaan mainos tai logo, jolla pyritään esittelemään ohjelmointikielten animaatioita sekä ECMAScriptin tarkoituksena esitellä interaktiivisuutta käyttäjän kanssa.

6 SVGator

SVGator on animointialusta, joka tarjoaa SVG-animaatioita varten työkaluohjelmiston. Alusta on ilmainen rajallisilla ominaisuuksilla ja sivusto tarjoaa myös LITE- ja PRO-versiot maksullisilla suunnitelmissa. Maksulliset suunnitelmat tarjoavat mahdollisuuden tehdä enemmän SVG-animaatioita, valmiita ikoneita, ajastetut pidemmät animaatiot, kehittyneempi animaattori, lisää tallennustilaa sekä muita tiedostomuotoja videoiden tekemiseen. SVGator on suunniteltu helpottamaan ja nopeuttamaan SVG-animaatioiden luomista alustan kehitetyllä animaattorilla sekä erilaisilla graafisilla työkaluilla. (SVGator, 2023)

6.1 SVGator animointialustan työkalut

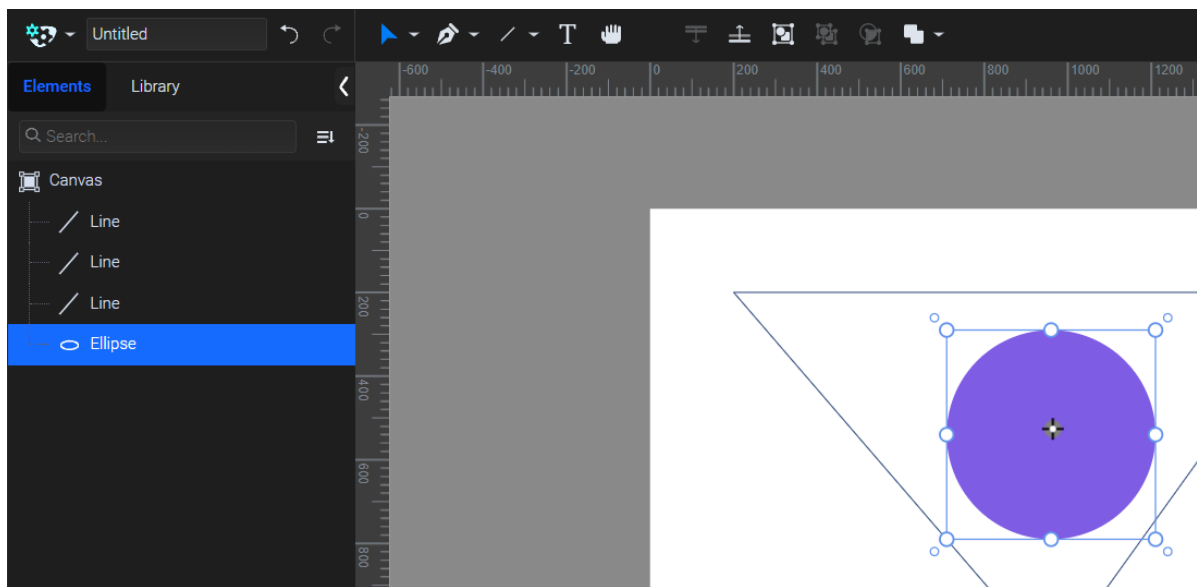
SVGator tarjoaa monta graafista työkalua itse suunniteltuihin piirroksiin sekä animaattorin, jolla voi animoida sen. SVGator tarjoaa myös valmiita kuvia sekä animoituja kuvia erilaisina ikoneina, jotka sisältyvät vain PRO-versioon. SVGator sisältää työkaluja, kuten canvaksen (piirtopinta) hallinta, erilaiset piirtotyökalut, elementtien (muodot ja piirrokset) muokkaus sekä animaattorin.

6.1.1 Canvas ja piirtotyökalut

SVGatorilla projektin luodessa aukeaa sivu, jossa pääset näkemään canvaksen ja sivun ympärillä erilaisia työkaluja. Sivun keskiössä on canvas ja tehdyt elementit tulevat siihen nähtäville, joita pystyy hallitsemaan sivun vasemman laidan hallintatyökalun avulla. Hallintatyökalun avulla pystyt valitsemaan yksittäisiä elementtejä ja oikea klikkauksella kopioimaan, poistamaan, animoida, kääntämään ja siirtää elementtiä eri kohtaan canvasta. Canvaksen laidoilla on asteita 200 välein, jotka auttavat tekemään poikittaisia viivoja tarkasti sekä canvaksella on automaattinen viiva joka kertoo keskikohdan muotoja sijoittaessa.

Piirtotyökalut löytyvät hallintatyökalun vierestä sivun ylälaidasta, joihin kuuluu muutostyökalu, piirtämistyökalu sekä muotojen ja tekstin lisäämiseen painikkeet. Piirtämistyökalulla voi piirtää suoria viivoja, jonka jälkeen voi tehdä kaareutuvia viivoja sekä tarjolla on myös kynätyökalu, jolla voi piirtää kokonaisvaltaisesti mitä vain osaa. Muotoja tarjolla on viiva, kolmio, ympyrä, neliö ja monikulmio. Yläpalkissa on vaihtoehtoja myös canvaksen elementtien kiertämiseen asteina, asetteluun sekä jos haluaa yhdistää eri elementtejä yhdeksi elementiksi. Kuvassa 3, joka visualisoi canvaksen elementtien hallinnan sekä yläpalkista löytyvät piirtotyökalut.

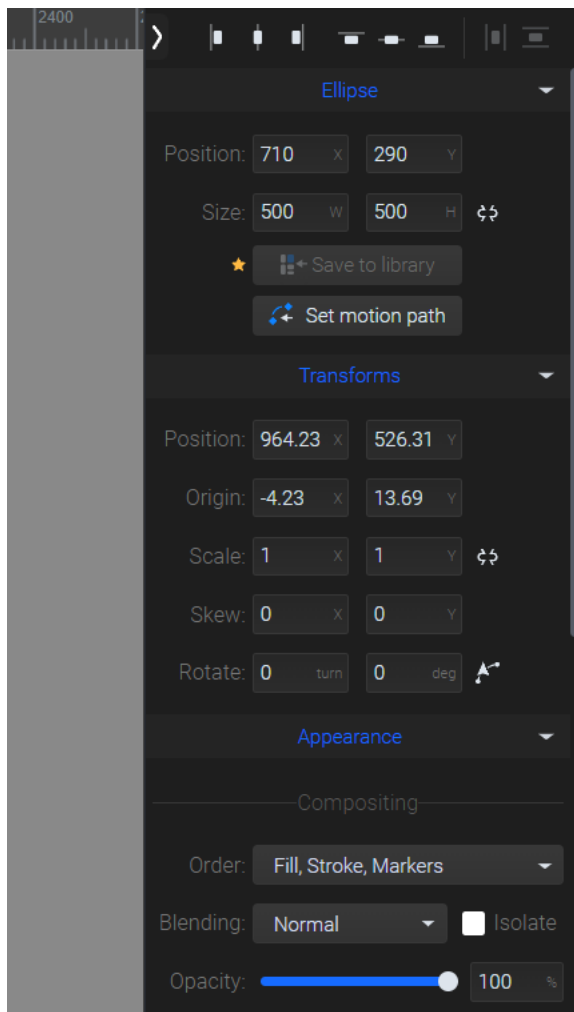
Kuva 3. Canvaksen hallinta ja piirtotyökalut (SVGator, 1.2.2023)



6.1.2 Elementtien muokkaus

Yksittäisen elementin muokkaus tulee näkyviin sivun oikealle puolelle, kun valitset elementin canvakselta. Pystyt muokkaamaan sieltä elementin sijaintia, kokoa sekä väri asetuksia. Elementin sijainnin ja koon tarkemmat asteet voi asettaa kirjottamalla, jos haluaa elementtien olevan symmetrisesti canvaksella. Kuvassa 4 näkyy yksittäisen elementin asetukset, kuten sijainti, koko ja ulkonäkö.

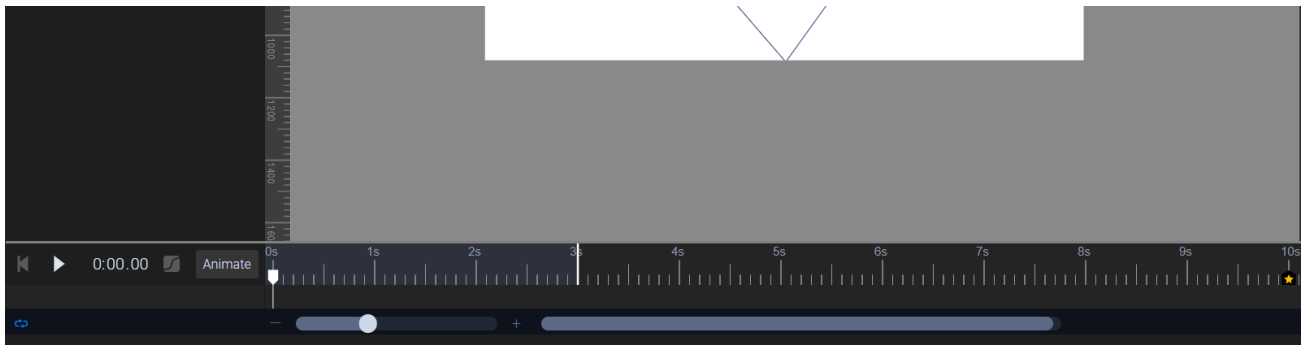
Kuva 4. Elementin sijainnin, koon ja värien muokkaus tarkasti (SVGator, 1.2.2023)



6.1.3 Animaattori

Sivun alhaalta löytyy Animaattori, kun valitset halutun elementin voit animoida sen sijaintia toiseen sijaintiin, koon muokkausta sekä elementin kiertoa. Sekunti mittarille asetetaan kesto ja animate painikkeella haluttu animaatio toiminto. Ilmaisen version kesto on enintään 10 sekuntia, joka riittää hyvin logo tai mainos animaatioihin. Alapuoolella kuva, jossa animaation ajastuksen hallinta sekä Animate painike valitun animaation toteuttamiseen elementille.

Kuva 5. Animaattorin hallinta (SVGator, 1.2.2023)



6.2 Animointialustan toteutus

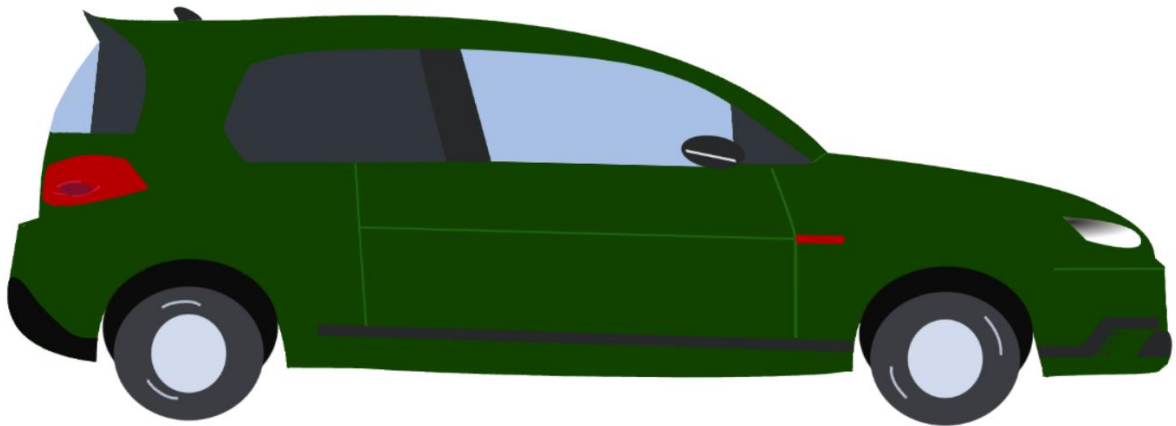
Alustan toteutusta tehdessä tärkeimmässä osassa oli keskittyä animaation graafiseen suunnitteluun sekä suunnittelin käyttökohdetta animoidulle SVG-kuvalle. Toteutuksena suunnittelin animoidun auton, jota voi käyttää mainoksena yrityksille, jotka liittyvät autoihin. Toteutuksen SVG-kuva on luotu käytettäväksi Janakkalaiselle yritykselle Fiksupesulle, mainostuksen visuaaliseksi elementiksi, jota tultaisiin hyödyntämään auton pesumainosten kanssa.

Toteutuksen hahmottelussa mallina käytettiin netistä löytyviä erilaisia auto piirroksia. Keskeisimmässä osassa oli hyödyntää animaatioalusta SVGatorin graafisia-työkaluja sekä erilaisia animaatio tekniikoita. Animoidun SVG-kuvan rakenne on tehty viivoilla sekä piirrostyökalulla. Piirrostyökalun piirrettyjä muotoja muokkasin piirroksen jälkeen näkyvien pisteiden avulla.

Animaatioissakin oli kahdella eri tapaa hyödynnetty viiva piirroksia, kuten auton renkaissa viivat liikkuvat sijainnista seuraavaan sijaintiin ajan kuluessa luoden renkaiden pyörimisen sekä auton sivuikkunassa vaaleansininen viiva, joka ajan mittaan liikkuu sivulle ja kasvaa, jonka tarkoituksena oli luoda heijastusefekti. Animaattorilla oli myös nostettu auton rungon sijaintia sekä käännetty auton keulaa ylöspäin ajan kuluessa, joka edistää animaation liikettä ihan kuin auto ajaisi.

SVGatorin graafiset-työkalut tarjosivat paljon erilaisia vaihtoehtoja sekä vain mielikuvitus rajanaan piirtämisen ja muotojen käytössä. Toteutusta tehdessä, kuitenkin paljon pikkutarkkaa silmää tarvitaan sekä taitoa piirtämisestä. Toteutuksen animaation voi esikatsella opinnäytetyön toisen liitteen linkin kautta.

Kuva 6. SVGator SVG-tiedosto PNG-kuvana esikatseluna.



7 Toteutukset mahdollistamaan vertailu

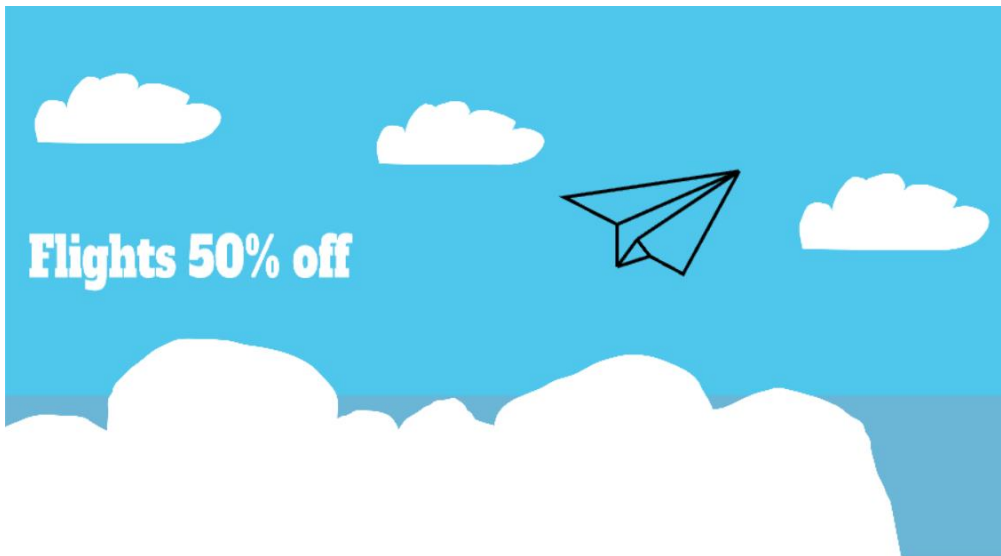
Toteutuksissa SVG-animaatioita tehdään kahdella eri ohjelmointikieli tavalla CSS ja ECMAScript sekä kuvauskieli SMIL, joilla selvitetään tapojen hyödyt ja mahdollisuudet toisiinsa verrattuna. Kyseisiä toteutuksia verrataan myös animaatioalusta SVGatorin toteutukseen. SVG-tiedoston vektorigrafiikan kirjoittaminen itse on erittäin haastavaa, jos yrittää rakentaa kuvaa, jossa on paljon eri muotoja yhdistettynä, siksi toteutuksissa on käytetty Figman graafista-työkalua kuvien suunnittelussa, jonka jälkeen ne on tuotu VS Codeen lisä muokkauksia varten sekä tuoda kuviin animoinnit.

7.1 CSS

CSS-ohjelmointikielen avulla päätin suunnitella mainostaustan, jossa olisi tarpeeksi yksinkertaiset animaatiot, koska CSS on huomaamani perusteella varsin rajoitettu animaatioiden tekemiseen. Animaatioiden hallinta CSS-tavalla tekemäni perusteella oli rajoitettu animaatiot tekemään yksi animointi mahdollisuus kerrallaan. CSS-tavan rajoitukset huomioon ottaen päätin suunnitella simppelein mainoksen, jota voisi hyödyntää web-sivuilla.

Mainos on lentoyhtiö teemainen, joka mainostaa alennetuista hinnoista. CSS-tyylittelyn takia halusin yhdistää animaatioon tekstin, joka johti mainokseen. Graafinen suunnittelu mainosta varten on suunniteltu Figma-työkalun avulla, jonka aloitin luomalla kahden eri sävyn taivaan taustoilla sekä piirtämällä pilviä. Pääelementtinä paperilennokki, joka kuvastaisi lentokonetta simppeleinä tapana sekä mainostusteksti. Toteutuksen animaation voi esikatsella opinnäytetyön toisen liitteen linkin kautta.

Kuva 7. CSS-ohjelmointikielen SVG-tiedosto PNG-kuvana esikatseluna.



Kuva 8. SVG-tiedoston sisältö CSS-ohjelmointikieli tavalla sisältäen kaiken graafisen sisällön sekä elementtien nimeäminen.

```

Flight ad-banner.svg
D: > Opinnäytetyö > SVG > CSS > Flight ad-banner.svg > {} svg
1 <svg width="910" height="501" viewBox="0 0 910 501" fill="none" xmlns="http://www.w3.org/2000/svg">
2 <rect width="910" height="501" fill="#f5f5f5"/>
3 <rect x="-23.7" y="23" width="931" height="497" transform="matrix(1,0,0,1,4,0)" fill="#6bb6d6"/>
4 <rect x="-9.3" y="-0.3" width="921" height="355" transform="matrix(1,0,0,1,4,0)" fill="#4fc6eb"/>
5
6 <g>
7 <path id="clouds" d="M 145.7 128.3 C 107.4 127.8 25.9 128 25.7 126.6 C 25.5 125.6 23.2 116.2 23.2 114.3 C 23.2 110.6 2
8
9 <path id="clouds" d="M 836.5 224.2 C 797.3 223.7 715 223.9 714.7 222.4 C 714.6 221.3 712.2 211 712.2 208.9 C 712.2 204
10
11 <path id="clouds" d="M 441.2 147.4 C 406.8 146.9 333.6 147.1 333.4 145.8 C 333.3 145 331.2 136.4 331.2 134.7 C 331.2 1
12 </g>
13 <path id="plane" d="M 661.5 153 L 504 176.5 L 551.5 201 M 661.5 153 L 611 245.5 L 581 229.5 M 661.5 153 L 569.5 214.5
14
15 <path id="text" d="M 22 248.6 V 240.3 H 23.7 V 217.5 H 22 V 209.2 H 43.8 V 222.1 H 37.3 V 217.5 H 34.2 V 225.9 H 39.5
16
17 <path transform="matrix(1,0,0,1,-1.7,-3.1)" d="M -2.4 393.4 C -0.2 388.7 -5.9 385.8 -1.1 388.9 C 0.6 388.4 8 384.9 9.7
18 </svg>

```

SVG-tiedoston ohjelmakoodi riveillä 1–18 on kaikki graafinen sisältö ja niiden vektorigraafiset pisteet sekä asetukset, kuten väri. Riveillä 6–12 on g-merkillä ryhmitelty taustan pilvet yhdeksi sekä annettu jokaiselle id-nimeksi "clouds" animaatiota varten. 12 ja 15 rivillä on nimetty myös id:llä paperilennokki "plane" ja teksti "text".

Kuva 9. SVG-tiedoston sisältö CSS-ohjelmointikieli tavalla sisältäen CSS-tyylittelyn animaatioita varten.

```
Flight ad-banner.svg
D: > Opinnäytetyö > SVG > CSS > Flight ad-banner.svg > ...
19 <style>
20 #clouds {
21   animation: sky 20s infinite;
22 }
23 @keyframes sky {
24   from {
25     transform: translateX(50%)
26   }
27 }
28 #plane {
29   animation: flight 15s;
30 }
31 @keyframes flight {
32   from {
33     transform: translateX(-63%) translateY(40%)
34   }
35 }
36 #text {
37   animation: appear 15s;
38 }
39 @keyframes appear {
40   from {
41     transform: translateY(33%)
42   }
43 }
```

CSS-ohjelmointikieli tavalla animaatiot kuuluvat "style" eli tyylittely elementin sisälle toisin kuin muilla tavoilla animaatiot ovat SVG-elementin sisällä. Ensimmäiseksi animoidessa valitaan haluttu elementti, joka aiemmin oli id:llä nimetty sekä animaation nimi ja ajan hallinta. Animaation aktivointi ja nimeäminen tapahtuu keyframes komennolla, jonka sisälle tulee elementin animoitu muutos. Esim. paperilennokki, joka oli nimetty id:llä plane, jonka sisällä on animaatio "flight" ja kesto 15 sekuntia. Flight animaatio, jonka animaation tyyppi transform liikuttaa elementtiä ja asetukset on elementin lähtö pisteenä liikkua -63 prosentista x-akselilla ja 40 prosenttia y-akselilla, joka mahdollistaa paperilennokin siirtymisen kuvan alavasemmalta keskelle kuvaa.

7.2 ECMAScript

ECMAScript-ohjelmointikielen avulla oli tarkoitus suunnitella SVG-kuva, joka olisi interaktiivinen käyttäjän kanssa. Kyseisellä tavalla pyrin keskittymään enemmän interaktiiviseen toimintoon, kuin itse eri animaatioiden hyödyntämiseen. ECMAScript-ohjelmointikielillä SVG-tiedoston kirjoittaminen oli myös haastavampaa, joten viikon sprintissä en suunnitellut ehtivän tekemään monimutkaisempaa SVG-kuvaa.

SVG-tiedoston käyttötarkoitukseksi suunnittelin jakamisikoniksi, joka voisi olla Web-sivuilla visuaalisempi elementti tekstin, kuvan tai videon alapuolella, jonka avulla jakaisi kyseisen sisällön muihin nettisivuille. SVG-tiedostoon ei kuitenkaan ole tehty linkin jakamisominaisuutta, johon tarvittaisiin sisältö ja alusta, jossa sisältö olisi alun perin. Toteutuksen animaation voi esikatsella opinnäytetyön toisen liitteen linkin kautta.

Kuva 10. ECMAScript-ohjelmointikielen SVG-tiedosto PNG-kuvana esikatseluna.



Kuva 11. SVG-tiedoston sisältö ECMAScript-ohjelmointikieli tavalla sisältäen jakamisikonin sekä animaation linkit.

```

ShareButton.svg X
D: > Opinnäytetyö > SVG > ECMAScript > ShareButton.svg
1 <svg width="513" height="250" viewBox="0 0 513 250" fill="none" xmlns="http://www.w3.org/2000/svg">
2
3 <g id="Facebook">
4 <a href="https://www.facebook.com/">
5 <path fill="#039be5" d="M 204.6 60.7 c 194.1 60.7 185.6 69.2 185.6 79.7 c 185.6 90.2 194.1 98.7 204.6 98.7 c 215.1 98.7
6 <path fill="#ffffff" d="M 206.2 82.7 h 4.9 l 0.8 -5 h -5.7 v -2.7 c 0 -2.1 0.7 -3.9 2.6 -3.9 h 3.1 v -4.4 c -0.5 -0.1
7 </a>
8 </g>
9
10 <g id="Youtube">
11 <a href="https://www.youtube.com/">
12 <path transform="matrix(0.1,0,0,0.1,-185.9,126)" style="fill: #f61c0d" d="M 4680 -654.2 H 4410.5 C 4357.6 -654.2 4314.8
13 </a>
14 </g>
15
16 <a href="https://twitter.com/">
17 <path id="Twitter" fill="#03a9f4" d="M 347.7 66.2 c -1.4 0.7 -2.9 1.1 -4.6 1.4 c 1.6 -1.3 -2.7 3.6 -4.7 c -1.5 0.9 -3.1
18 </a>
19
20 <g id="Share" onclick="share_click(event)">
21 <rect stroke-width="2" stroke="black" x="169.5" y="45.5" width="195" height="68" rx="34" id="Share" fill="#455386"/>
22 <path d="M 234 86 c 235.1 86 235.9 85.7 236.6 85.2 c 237.2 84.7 237.6 83.9 237.6 82.9 c 237.6 82 237.3 81.4 236.8 80.8 c
23 </g>

```

SVG-tiedoston koodin riveillä 1–23 on kaikkien linkki ikonien sekä itse Share painikkeen graafiset pisteet sekä asetukset. Share painikkeessa on myös siirtymiskohta, jonka käynnistää onclick kohta rivillä 20.

Kuva 12. SVG-tiedoston sisältö ECMAScript-ohjelmointikieli tavalla sisältäen scriptin, joka mahdollistaa interaktiivisen animaation.

```

ShareButton.svg X
D: > Opinnäytetyö > SVG > ECMAScript > ShareButton.svg
24
25 <script type="application/ecmascript"> <![CDATA[
26     function share_click(event) {
27         object1 = document.getElementById('Facebook');
28         object2 = document.getElementById('Youtube');
29         object3 = document.getElementById('Twitter');
30         window.requestAnimationFrame(translate);
31     }
32     function translate() {
33         object1.setAttribute('transform', 'translate(0, 63)');
34         object2.setAttribute('transform', 'translate(0, 67)');
35         object3.setAttribute('transform', 'translate(0, 65)');
36     }
37 >>> </script>
38 </svg>

```

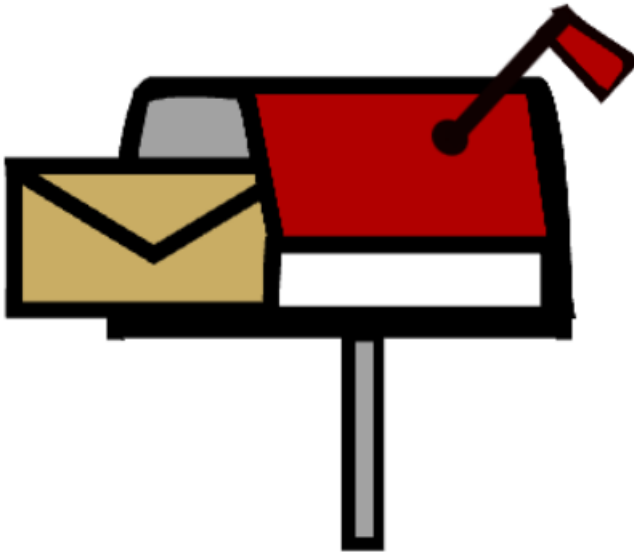
SVG-tiedoston koodissa riveillä 25–38 on loput SVG-tiedostosta, joka sisältää ECMAScriptin scripti osan. Scriptin tyyppi valitaan alussa sekä luodaan kaksi funktiota klikkausta ja animaatiota varten. Ensimmäinen funktio etsii halutut elementit id:n perusteella ja asettaa nimetyiksi objekteiksi sekä valmistelee animaatiota varten. Toinen funktio on animaatio, joka siirtää kaikki objektit alaspäin 63–67 astetta. Kyseiset linkki ikonit ovat alun perin asetettu Share painikkeen alle.

7.3 SMIL SVG

SMIL SVG on erittäin hyvä keino luoda SVG-tiedostoja ja varsin helppo oppia. Animaatio mahdollisuudet, joita SMIL tarjoaa, on siihen asti riittävät, kun ei tarvitse tehdä interaktiivisia animaatioita, joten se sopii hyvin logojen, mainosten tai ikonien tekemiseen. SMIL-kuvauskieli tavalla suunnitellin animaation, joka on animoitu ikoni postitus mainokseen liittyen. Itse kuva oli suunniteltu kolmen eri elementin välille, jotka Figman avulla oli rakennettu. Eri elementit ovat postilaatikonrunko ja sisääntulo aukko, postilaatikon sivusta sekä lippu ja viimeisenä itse kirje. Jokainen elementti oli siirretty yksitellen VS Codeen, jonka jälkeen ne yhdistettiin sekä laitettiin oikeaan järjestykseen. Järjestys ja elementtien yhdistäminen oli tärkeässä osassa, jotta voisin luoda efektin, että kirje näyttää menevän postilaatikon sisälle.

Animaatioissa hyödynsin SMIL-komentoa `animateTransform` kahdella eri tapaa, joiden tyypit olivat kirjeen liikuttaminen eteenpäin 3,5 sekunnin, jonka jälkeen palaisi takasin aloitussijaintiin sekä lipun kääntäminen 3,5 sekunnin ajan. Lippu on aloitussijainnissa postilaatikon sivussa, joka merkitsee tyhjää postilaatikkoon, mutta kääntyy ylöspäin, kun kirje on postilaatikkoon menossa. Itse animaatio on suunniteltu loputtomaksi animaatioksi, mutta sitä voisi hyödyntää myös yhden liikkeen animaatioksi. Toteutuksen animaation voi esikatsella opinnäytetyön toisen liitteen linkin kautta.

Kuva 13. SMIL SVG-kuvauskielen SVG-tiedosto PNG-kuvana esikatseluna.



Kuva 14. SVG-tiedoston sisältö SMIL kuvauskieli tavalla sisältäen postilaatikon rungon sekä kirjeen ja sen animoinnin.

```

MailBox.svg
D: > Opinnäytetyö > SVG > MailBox.svg > {} svg > {} g
1 <svg width="219" height="217" viewBox="0 -100 620 320" fill="none" xmlns="http://www.w3.org/2000/svg">
2 <path d="M 335.5 93 C 335.9 8.7 323 -16.8 316.5 -19" stroke="black" stroke-width="8"/>
3 <path d="M 258.9 97.9 C 259.7 3.5 270.6 -21.4 275.9 -22.1" stroke="black" stroke-width="8"/>
4 <path d="M 469.9 97.9 C 469.9 9.1 461.8 -19.1 457.7 -22.1 H 274.9" stroke="black" stroke-width="8"/>
5 <rect x="31.9" y="381.2" width="8" height="210" transform="matrix(0,-1,1,0,-119,129)" fill="black"/>
6 <rect x="365.4" y="97.2" width="20" height="100" fill="black"/>
7 <rect transform="matrix(1,0,0,1,0.6,-1.6)" x="371.5" y="100.2" width="8" height="92.1" fill="#a2a2a2"/>
8 <path fill="#a2a2a2" d="M 333.7 89.3 H 261.4 L 264.7 35.8 L 266.2 21.3 L 268.2 7.8 L 270.7 -4.7 L 272.7 -14.2
9 C 287.2 -19.7 311.1 -16.2 318.7 -16.2 C 326.3 -16.2 332.4 55.1 333.7 89.3 Z"/><path stroke-width="8" stroke="black"
10 d="M 259.6 86.1 C 311.6 86.1 330.3 86.1 332.1 86.1 C 330.8 51.9 324.7 -19.4 317.1 -19.4 C 309.5 -19.4 285.6 -22.9 271.1 -17.4
11 L 269.1 -7.9 L 266.6 4.6 L 264.6 18.1 L 263.1 32.6 L 259.6 89.1"/>
12
13 <g class="Letter">
14
15 <path fill="#c9ad64" d="M 103.9 13.7 C 104.1 15.8 104.3 42.6 104 81.3 H 235.8 V 13.7 H 103.9 Z"/>
16 <animateTransform
17   attributeName="transform"
18   attributeType="XML"
19   type="translate"
20   dur="3.5s"
21   values="15,0;+185,0"
22   repeatCount="indefinite"
23 />
24 <path stroke-width="8" stroke="black" d="M 106.4 87.5 C 106.8 46.7 106.6 18.2 106.4 16 H 238.3 V 83.7 H 102.8"/>
25 <path stroke-width="8" stroke="black" d="M 109.8 19.8 C 156.2 48.2 171.8 57.5 172.3 57.8 L 235.3 19.8"/>
26 </g>

```

SVG-tiedoston koodissa 1–11 riveillä sisältää postilaatikon rungon vektorigrafiikka pisteet sekä niiden asetukset kuten koon, värit ja viivojen paksuudet. Rivit 13–26 on g-merkillä, joka on group eli elementit on ryhmitelty yhdeksi elementiksi, joka on nimetty Letter. Kyseinen kohta on kirjeen vektorigraafisia pisteitä varten sekä kyseisessä ryhmittelyssä animateTransformilla animoidaan kirje liikkumaan edestakaisin. Kirje liikkuu asteesta 15 oikealle päin 185 asteeseen asti 3,5 sekunnin ajan ja palaa takaisin.

Kuva 15. SVG-tiedoston sisältö SMIL kuvauskieli tavalla sisältäen postilaatikon sivustan sekä lipun ja sen animoinnin.

```

MailBox.svg X
D: > Opinnäytetyö > SVG > MailBox.svg > {} svg
27
28 <path d="M 399.6 58.3 H 336 L 320 -21.7 L 305.6 -21.7 H 455.6 L 470.6 82.3 Z" fill="#b10000" stroke="black" stroke-width="8"/>
29 <path d="M 332 79.9 V 86 H 463 V 53 H 333 Z" fill="white" stroke="black" stroke-width="8"/>
30
31 <g class="Flag">
32
33 <path stroke-width="8" stroke="#100000" d="M 448.1 -152.2 C 455.4 -148.3 488.2 -156.2 489.6 -155.4 V -154.5 V -136.5 L 472.6 -138
34 L 448.1 -140" transform="matrix(0,1,-1,0,345.1,-468.3)"/>
35 <animateTransform
36 attributeName="transform"
37 attributeType="XML"
38 type="rotate"
39 dur="3.5s"
40 from="0 420 -10"
41 to="-90 455 -2.1"
42 repeatCount="indefinite"
43 />
44 <path stroke-linecap="square" transform="matrix(0,1,-1,0,292.8,-438.9)"
45 d="M 424.2 -198.9 C 426 -198.9 425.7 -198.9 427.6 -198.6 C 454 -202.9 456.9 -203.3 457.1 -202.9
46 C 458.7 -196.5 456.7 -209 458.7 -193.7 M 457.1 -193.2 C 457 -193.1 457.2 -193.2 457.1 -193.2 Z M 457.1 -193.2
47 C 456.9 -193.1 456.6 -193.3 456.5 -193.2 C 426.6 -195.4 425.6 -195.5 424.7 -195.5 C 420 -195.5 434 -195.5 438.7 -195.5
48 C 443.5 -195.5 448.4 -195.7 453.2 -195.7 C 454.4 -199.7 451.4 -197.6 449.9 -197.6" stroke="#b10000" stroke-width="6"/>
49 <ellipse fill="#100101" cx="394.2" cy="43.2" rx="9" ry="8" transform="matrix(0,1,-1,0,460.5,-410.7)" />
50 <rect fill="#0f0101" transform="matrix(0,1,-1,0,424.6,-446.2)" x="426.2" y="-73.7" width="9" height="83" rx="4.5"/>
51 </g>
52
53 </svg>

```

SVG-tiedoston koodissa 28–29 riveillä on postilaatikon sivustan vektorigraafiset pisteet ja asetukset. Lipun animointia varten on ryhmitelty kaikki elementit yhdeksi elementiksi, joka tässä on Flag. Lipun animateTransform kohdassa käännetään lippua asetetuista asteista toisiin asteisiin 3,5 sekunnin ajan.

8 Toteutusten vertailu

Toteutukset mahdollistavat vertailun animaatioalustan ja ohjelmointikielten välillä sekä vertailun eri ohjelmointikielten välillä. Vertailussa huomioidaan oma henkilökohtainen kokemus, hyödyt, rajoitukset sekä oppimiskäyrä tavoilla. Jokaisella tavalla ei ole ennen opinnäytetyötä luotu SVG-tiedostoja ja ohjelmointikielistä on perustaidot ennestään opittuna, paitsi SMIL SVG, jonka tekniikka on opinnäytetyön aikana tutkittu. Animointialusta SVGatoria verrataan yleisesti kolmeen eri ohjelmointikieleen sekä vertailu ohjelmointikielten välillä.

8.1 Animointialusta verrattaessa ohjelmointikieleen ja graafiseen alustaan

SVGator-animointialustan graafinen työkalu ja animaattori mahdollistavat keinot haastaviin graafisiin piirroksiin sekä animaatioihin, joihin ohjelmointikielillä se vaatii aikaa sekä taitoa. SVGatorin etu on siksi säästää aikaa ja ei tarvitse opetella mahdollisia ohjelmointikielen kirjoittamistaitoja tai eri tapojen rajoituksia. SVGator on ilmainen, jossa pystyy tehdä vain kolme SVG:tä kuukaudessa, mutta maksullinen versio tarjoaa rajattomasti tiedostojen tuonteja sekä valmiiden ikonin käytön. Nykyaikana syystäkin on luotu erilaisia alustoja, joiden tarkoitus on nopeuttaa ja helpottaa prosesseja.

SVG-tiedostojen tekeminen työksi on kannattava valmiin animointialustan avulla, jotta säästät aikaa ja työtaakkaa, alustan maksamisesta huolimatta, koska se olisi kuitenkin työtä varten hankittu ohjelmisto. SVG-tiedostojen luominen ohjelmointikielen sekä graafisen alustan avulla on enemmänkin harrastuksena tai opiskelun keino, jotta ymmärtää miten tiedosto on rakennettu sekä SVG-animaatioiden perusteet.

8.2 Ohjelmointikielten vertailu

Ohjelmointikielten vertailussa nousee esiin kaksi pääpointtia, jotka ovat itse ohjelmointikielen oppimiskäyrä sekä eri tapojen hyödyt tai rajoitukset. Kaikista eniten hyötyjä tarjoaa ECMAScript:llä SVG-tiedostojen luominen, mutta se on myös haastavin kirjoittaa. ECMAScriptin hyödyt ovat interaktiiviset animaatiot käyttäjän ja tiedoston välillä sekä mahdollisuudet tehdä erittäin edistyneitä animaatioita. ECMAScript on kuitenkin Javascriptin kaltainen ohjelmointikieli tapa, joka tarvitsee ymmärrystä olosuhteiden sekä loogisen ohjelmoinnin perusteista hyödyntäen ECMAScriptiä tehokkaasti.

CSS-ohjelmointikieli, joka oli oman kokemuksen perusteella keskiverto muihin verrattuna oppia, tarjosi vähäiset mahdollisuudet eri animaatioiden hallitsemiseen sekä käyttämiseen. Ohjelmointikielen animaatioiden hyödyntäminen mainoksissa tai logoissa on kuitenkin riittävä. CSS-ohjelmointikieli oli omasta mielestäni heikoin tapa tuoda esille tehokkaasti animaatiot SVG-tiedostoissa, mutta tapa on kuitenkin suosittu web-sivujen tyyllittely taustan takia ja ohjelmointikieli ei tarvitse paljoa osaamista luodakseen SVG-tiedostoja.

SMIL SVG-kuvauskieli oli ohjelmointikielenä helppo oppia sekä rakenteeltaan miellyttävän kirjoittaa. Kirjoittamistyylinä oli luettelotapa, jonka avulla valittiin animaatio sekä mahdolliset asetukset sille. Ohjelmointikielillä ei ole mahdollista tehdä interaktiivisia animaatioita, kuten ECMAScript, mutta animaatioiden taso oli hyvä. SMIL SVG on nykyaikana vielä hyödynnettävä tapa, mutta ei ole ohjelmointikieli, johon kannattaa tukeutua pelkästään. Tulevaisuudessa selainten ja vaatimusten edetessä SMIL SVG tulee jäämään kehityksessä jälkeen, jos ohjelmointikieltä ei tulla päivittämään.

9 Yhteenveto

SVG on nykyaikana ja tulevaisuudessa varteenotettava tiedostomuoto, joka tarjoaa visuaalisia kuvia sekä animaatioita web-sivuille ja muihin tarkoitukseen. Vektorigrafiikkaan pohjautuva tiedostomuoto on hyödyllinen tapa nykyajan web-sivujen visuaalisuudessa, tarjoten skaalautuvuuden sekä animaatioiden hyödyntämisen. SVG-animaatiot ovat myös suosittu keino vaikuttaa web-sivujen käyttäjiin mainostuksen ja markkinoinnin kannalta. Opinnäytetyön tarkoituksena oli selvittää tehokkain tapa SVG-animaatioiden luomisen kannalta sekä huomata mahdolliset rajoitukset.

Oman kokemuksen, hyötyjen ja rajoitusten perusteella suosituimmaksi tavaksi päädyin SVGator-animointialustaan. SVGatorin tarjoamat työkalut, animaatiot sekä itse alustan helppokäyttöisyys nostivat sen arvon kärkisijalle, mutta se on juuri animointialustan tarkoituskin helpottaa luomisprosessia ja nopeuttaa työskentelyä. Ohjelmointikielet ovat, kuitenkin tärkeässä osassa SVG-tiedostojen luomista historian tai oppimisen kannalta. Ohjelmointikielistä paras SVG-tiedostojen luomiseen on ECMAScript, joka tarjoaa Java-ohjelmointimahdollisuuksien hyödyntämisen animaatioissa tai interaktiivisuudessa. CSS ja SMIL SVG ovat tutkimisen kannalta myös hyödyllisiä tapoja sekä jos omaa CSS-ohjelmointikielystä kokemusta web-sivujen suunnittelijana, on tavan hyödyntäminen riittävä yksinkertaisia mainoksia tai logojen animointeja varten.

Opinnäytetyötä tehdessä oppi paljon vektorigrafiikasta sekä itselle melko tuntemattomasta SVG-tiedostomuodosta. Jokaisella ohjelmointikieli tavalla sekä animointialustalla oli miellyttävää luoda SVG-tiedostoja sekä huomata eri tapojen hyödyt, rajoitukset ja tekniikkatyölin. Aion opinnäytetyön jälkeenkin enemmän huomioida SVG-tiedostoja sekä mahdollisesti luoda itse lisää logoja, mainoksia tai web-sivun suunnittelun kannalta visuaalisia elementtejä tulevaisuudessa.

Lähteet

Adobe. (n.d.) Learn About SVG Files

<https://www.adobe.com/au/creativecloud/file-types/image/vector/svg-file.html>

Chandra,A. (2020). Why Marketers Need a Visual Influence Strategy

<https://hub.uberflip.com/content-marketing/why-marketers-need-a-visual-influence-strategy>

Huttunen,K. (2020). Visuaalinen markkinointi - keino vaikuttaa ostopäätöksiin

<https://www.zoner.fi/digitaalinen-markkinointi/visuaalinen-markkinointi/>

Kilpeläinen,J. (2020). Markkinoinnin ja viestinnän eroa kunnioitettava

<https://viestijat.fi/markkinointi-ja-viestinta-pidettava-erillisina/>

MDN Web Docks. (2022). SVG element reference.

<https://developer.mozilla.org/en-US/docs/Web/SVG/Element>

MDN Web Docks. (2023). SVG Introduction.

<https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Introduction>

Nelimarkka,L. (2017). Visuaalinen viestintä: Mitä käyttämäsi kuvat viestivät organisaatiostasi?

<https://www.viestintapalvelut.fi/blogi/strateginen-visuaalinen-viestinta>

Nykänen,O (2007). SVG : skaalautuva vektorigrafiikka

<https://hamk.finna.fi/Record/vanaicat.91654>

Styling And Animating SVGs With CSS. (2014)

<https://www.smashingmagazine.com/2014/11/styling-and-animating-svg-with-css/>

SVGator. (2023)

<https://www.svgator.com/>

Visual Studio Code. (2023)

<https://visualstudio.microsoft.com/>

What is ECMAScript? (n.d.)

<https://www.educative.io/answers/what-is-ecmascript>

Wikipedia. SVG_animation. (2010)

https://en.wikipedia.org/wiki/SVG_animation

Liite 1. Aineistonhallintasuunitelma

Opinnäytetyön aineisto koostui suurelta osalta lähteiden tiedosta sekä itse suunnitelluista ja rakennetuista SVG-tiedostoista. Kaikki opinnäytetyöhön liittyvä, kuten muistiinpanot, kuvat ja SVG-tiedostot säilytetään paikallisesti tekijän omalla tietokoneella sekä SVG-tiedostot jokaisella eri tavalla on jaossa tekijän HAMK-tilin OneDrivessä sekä yhdessä julkisessa nettisivussa, joiden näkemiseen tarvitsee linkin.

Opinnäytetyössä käytetyt kuvat ovat lähteistä tai omalla tietokoneella otettuja kuvakaappauksia, jotka ovat merkitty lähteiden mukaan. Taulukot ja niiden tiedot ovat itse tehtyjä Wordin avulla, mutta tieto on lähde peräistä.

Aineisto säilytetään vähintään yksi vuosi opinnäytetyön hyväksymispäivästä alkaen. Aineisto ei sisällä mitään mikä ei ole julkisesti saatavilla tai henkilötietoja sisältävää materiaalia, jonka takia aineisto ei sisällä mitään hävitettävää.

Liite 2. SVG-tiedostojen esikatselu linkkien kautta

SVGatorilla tehty SVG-tiedosto, joka on animoitu auto.

[CarAnimationFinal.svg](#)

CSS-ohjelmointikieli tavalla tehty SVG-tiedosto, joka on lentoyhtiöille mainos.

[Flight ad-banner.svg](#)

ECMAScript-ohjelmointikieli tavalla tehty SVG-tiedosto, joka on jako painike ikoni.

<https://codebeautify.org/svg-viewer/y23b493ee>

SMIL SVG-ohjelmointikieli tavalla tehty SVG-tiedosto, joka on animoitu postilaatikko.

[MailBox.svg](#)