

Opinnäytetyö (AMK)

Tietotekniikan koulutusohjelma

Mediatekniikka

2013

Toni Uusvuo

INTERAKTIIVISEN SELAINPOHJAISEN SOVELLUKSEN SUUNNITTELU JA TOTEUTUS



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma | Mediatekniikka

2013 | 51

Yliopettaja, FT Mika Luimula

Toni Uusvuo

INTERAKTIIVISEN SELAINPOHJAISEN SOVELLUKSEN SUUNNITTELU JA TOTEUTUS

Opinnäytetyön tarkoituksena oli tutkia käyttöliittymäsuunnittelun eri vaiheita sekä sen tuomia haasteita ja etuja. Lisäksi tutkittiin interaktiivisen www-sovelluksen luomiseen suunnattuja työkaluja ja niissä olevia sovelluskehittäjää helpottavia ominaisuuksia.

Käyttöliittymäsuunnittelun vaiheita on monia, joista parhaaksi todettiin vuokaavion käyttö. Vuokaaviolla saatiin hyvä käsitys sovelluksen kokonaiskuvasta sekä siinä olevien elementtien sijoitus sovelluksen sisällä voitiin suunnitella vuokaaviota hyväksikäyttäen.

Opinnäytetyössä käsiteltiin eri tapoja käyttöliittymäsuunnitteluun sekä sen testaamiseen. Työssä esiteltiin myös tapoja käyttää hyödyksi sovellusten kehitystyökaluja suunnittelijan työn helpottamiseksi ja vertailtiin eri kehitystyökalujen ominaisuuksia toisiinsa. Opinnäytetyössä vertaillaan kolmea eri kehitystyökalua, joista tähän sovellukseen parhaiten soveltui Adobe Edge Animate. Yritys piti tärkeänä käyttäjäkuntaa, joten sovelluksen piti toimia selaimessa ilman ylimääräisiä laajennuksia.

Opinnäytetyön tuloksena luotiin Turku Energia Oy:lle interaktiivinen www-sovellus, joka kertoo yrityksen yleistietoa sekä tietoa yrityksen omistamista rakennuksista, työuramahdollisuuksista sekä informaatiota sähkön ja lämmön tuotannosta. Sovelluksen luomiseen käytettiin Adobe Edge Animate –ohjelmistoa sekä Javascript-, jQuery- ja HTML5- ohjelmointikieliä. Sovelluksessa olevat kuvat luotiin Autodesk 3ds Max –ohjelmistolla ja muokattiin Adobe Photoshop –kuvamuokkausohjelmalla. Sovellus testattiin yrityksen sisäisesti kolmella käyttäjällä, joiden mielipiteet vaikuttivat sovelluksen kokonaisuuteen.

ASIASANAT:

käyttöliittymäsuunnittelu, Adobe Edge Animate, Javascript, verkkosovellus

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology | Digital Media

2013 | 51

Instructor: Principal lecturer, Ph. D. Mika Luimula

Toni Uusvuo

DESIGNING AND CREATING A BROWSER-BASED INTERACTIVE APPLICATION

The purpose of this thesis was to examine the different stages of user interface design and the benefits and challenges it brings. Additionally, the purpose was to examine the web application-oriented tools and the developer-friendly features among them.

Designing a user interface has many different routes and stages, and for this work the best option was to use a flow chart. The flow chart gives a good overview of the application and makes designing the application hierarchy easier.

This thesis introduces different ways to design an application. Thesis also compares the most popular application design tools on the market. Applications were compared according to their key features and compiling capabilities. Three different design tools were compared, and the best tool in this case was the Adobe Edge Animate. Turku Energia Ltd, which commissioned this thesis, stressed the importance of the user base, so the application had to work in a web browser without any additional plug-ins.

This thesis resulted in the creation of an interactive web application for Turku Energia Ltd. The application gives the user general information about the company, as well as information about the buildings owned by the company, career opportunities and information about electricity and heat production and distribution.

The application was made using the Adobe Edge Animate software and Javascript, jQuery and HTML5 programming languages. The buildings were created using Autodesk 3ds Max 3d modeling software and the images were modified using Adobe Photoshop picture editing software. The application was tested inside the company with three different users and their opinions were taken into account while creating the application.

KEYWORDS:

user interface design, Adobe Edge Animate, Javascript, web application

SISÄLTÖ

KÄYTETYT LYHENTEET JA SANASTO	7
1 JOHDANTO	8
2 INTERAKTIIVISTEN SOVELLUSTEN KEHITYSTYÖKALUT	10
2.1 Adobe Flash	10
2.2 Unity 3D	12
2.3 Adobe Edge Animate	13
2.4 Kehitystyökalujen vertailu	14
3 KÄYTTÖLIITTYMÄSUUNNITTELU	16
3.1 Käyttöliittymän suunnittelutavat	16
3.1.1 Toiminnallisten vaatimusten kerääminen	17
3.1.2 Käyttäjänalyysi	17
3.1.3 Prototyypitys	18
3.2 Käyttöliittymän testaus	19
4 ESITTELYSOVELLUKSEN SUUNNITTELU	21
4.1 Tavoite	21
4.2 Kilpailija-analyysi	22
4.2.1 Energiavisa	23
4.2.2 Together we can do it	24
4.2.3 Interactive House	25
4.3 Sovelluksen sisällön hahmottaminen	27
5 ESITTELYSOVELLUKSEN TOTEUTUS	29
5.1 Kuvahakemiston luonti	29
5.2 Sovelluksen taustakuva sekä elementtien sijoitus	30
5.3 Informaatioikkunoiden luonti	33
5.4 Navigointipalkin ponnahdusikkuna	33
5.5 Rakennusten ponnahdusikkuna	39
5.6 Videoiden ja tekstien lisääminen	40
5.7 Sähkö- ja lämpöverkon näyttäminen	42
5.8 Animaatioiden luominen	43

6 JATKOKEHITYSMAHDOLLISUUDET	45
6.1.1 Vierityspalkki	45
6.1.2 Sovelluksen hallinta- ja muokkausohjelma	46
6.1.3 Kokoruutuun siirtyminen	47
7 YHTEENVETO	48
LÄHTEET	50

KUVAT

Kuva 1. Adobe Flashin käyttöliittymä.	11
Kuva 2. Adobe Edge Animaten käyttöliittymä.	13
Kuva 3. Vasemmalla sekuntikellon toiminnallisuuslista ja oikealla kuvakaappaus valmiista sovelluksesta.	17
Kuva 4. Energiavisan pelinäköymä.	23
Kuva 5. YIT:n yritystoimintaa esittelevän sovelluksen näköymä.	24
Kuva 6. Interactive House -sovelluksen käyttöliittymä.	26
Kuva 7. Kuvakaappaus Adobe Edgen kirjastosta.	29
Kuva 8. Turku Energia -sovelluksen taustakuva.	30
Kuva 9. Turku Energia -sovelluksen navigointipalkki.	31
Kuva 10. Turku Energia -sovelluksen aloitusnäköymä.	31
Kuva 11. Toimitalo -rakennuskuvan ominaisuuksien ikkuna.	32
Kuva 12. Turku Energia -sovelluksen ponnahdusikkunan tekstilaatikko ja sen ominaisuudet.	36
Kuva 13. Turku Energia -sovelluksen ponnahdusikkuna.	37
Kuva 14. Sulkupainike. Vasemmalla sulkupainike normaaliasennossa ja oikealla alas painettuna.	37
Kuva 15. Turku Energia -sovelluksen sähkö- ja lämpöverkko taustakuvan päällä näytettynä.	42
Kuva 16. Adobe Edge Animaten aikajana.	44

KUVIOT

Kuvio 1. Esimerkki pelin alkuvalikon vuokaaviosta.	18
Kuvio 2. Sovelluksen objektien kategoriointi vuokaaviossa.	27
Kuvio 3. Sovelluksen ponnahdusikkunoiden tiedonhaku.	28

TAULUKOT

Taulukko 1. Kehitystyökalujen vertailu.	14
---	----

KOODIT

Koodi 1. Kuvan vaihtaminen kun hiiren kursori on kuvan päällä.	33
Koodi 2. Ponnahdusikkunan otsikon luonti.	35
Koodi 3. Sulkupainikkeen kuvan vaihtaminen ja toiminnon suorittaminen.	38
Koodi 4. Videopotuksen luominen.	39
Koodi 5. Tekstien ja otsikoiden alustaminen.	40
Koodi 6. Tekstien, otsikoiden, videolinkin ja kuvien sijoittaminen ponnahdusikkunaan.	41
Koodi 7. Lämpö- ja sähköverkon näyttäminen ja piilottaminen.	43

KÄYTETYT LYHENTEET JA SANASTO

2D	Kaksiulotteinen maailma, jossa otetaan huomioon pituus- ja leveysasteet.
3D	Kolmiulotteinen maailma, jossa otetaan huomioon pituus-, leveys- ja syvyysasteet.
.NET	Microsoftin kehittämä ohjelmistokomponenttikirjasto.
Actionscript3	Adobe Flashin käyttämä oliopohjainen skriptauskieli. Käytetään myös joissain tietokantaohjelmissa.
Dreambroker	Verkossa toimiva videonhallintaohjelmisto.
HTML5	HyperText Markup Language. HTML5 on HTML-merkintäkielen uusin versio.
Javascript	Pääasiassa verkkoympäristössä käytettävä komentosarjakieli.
jQuery	Avoimen lähdekoodin Javascriptkirjasto. Soveltuu hyvin toimintojen käsittelyyn ja animaatioiden tekemiseen.
WYSIWYG	What you see is what you get. Viittaa ohjelmistoihin, joissa sisältö näyttää muokattaessa samalta kuin lopputulos.

1 JOHDANTO

Internet on täynnä sovelluksia, joista osa voi olla ominaisuuksiltaan hyvinkin samanlaisia. Tämä johtuu osaksi siitä, että suosituimmat sovelluskehitystyökalut sisältävät paljon valmiita komponentteja, joita sovelluskehittäjä voi käyttää. Mikäli valmiita muiden tekemiä komponentteja käytetään, täytyy sovelluksen erottua muista samoilla työkaluilla tehtävistä sovelluksista jollain muulla tavalla. Yksi hyvä tapa erottua muista on tehdä hyvin suunniteltu ja yksinkertainen käyttöliittymä.

Käyttöliittymä voi tehdä keskivertosovelluksesta hyvän ja hyvästä sovelluksesta huonon. Vaikka otettaisiin vertailuun kaksi samoja ominaisuuksia sisältävää sovellusta, toinen niistä on usein käyttäjien mielestä parempi. Tämä johtuu usein käyttöliittymästä ja varsinkin sen suunnittelusta. Suunnittelussa täytyy ottaa huomioon kohderyhmä ja sovelluksen käyttötarkoitus. Käyttöliittymän suunnittelussa ei voida sanoa oikeaa ja väärää tapaa. Tyylejä on monia, mutta muutama erilainen suunnittelutapa on kuitenkin noussut suosituksi. Tällaisia suunnittelutapoja ovat esimerkiksi prototypitys, toiminnallisten vaatimusten kerääminen, monijaksoinen testaus, käyttäjäanalyysin hyödyntäminen sekä erilaisten kaavioiden luominen sovellusta havainnoimaan. Opinnäytetyö käsittelee käyttöliittymäsuunnittelun peruseriaatteita, tutkitaan mikä tekee hyvän käyttöliittymän perehdytään käyttöliittymiin liittyviä yleisimpiä suunnittelu- ja testaustapoja.

Tässä opinnäytetyössä käsitellään myös internetpohjaisten interaktiivisten sovellusten suosituimpia kehitystyökaluja sekä niiden hyviä ja huonoja puolia. Vertailuun valittiin Adobe Edge Animate, Unity 3D sekä Adobe Flash –ohjelmistot. Näille työkaluille yhteistä on se, että jokaisella voidaan suunnitella käyttöliittymiä valmiilla komponenteilla, jokainen auttaa sovelluskehittäjää koodin tuottamisessa sekä jokainen on vakiinnuttanut tai vakiinnuttaa parhaillaan paikkaansa yhtenä suosituimmista kehitystyökaluista.

Lisäksi opinnäytetyössä esitellään esimerkksiovellus, joka luotiin Turku Energia Oy:lle. Sovelluksen tarkoituksena on esitellä Turku Energian tuotantolaitoksia sekä kertoa sähkön ja lämmön tuotannosta. Lisäksi sovelluksen tarkoituksena on tutustuttaa käyttäjä sähkölaitoksen toimintaan sekä sen tarjoamiin uramahdollisuuksiin.

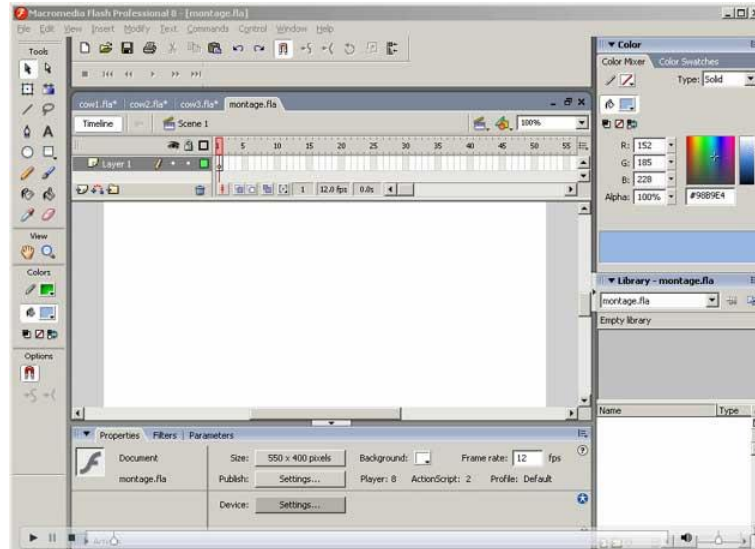
2 INTERAKTIIVISTEN SOVELLUSTEN KEHITYSTYÖKALUT

Iso osa nykyinternetiä perustuu interaktiivisiin sovelluksiin. Internetkauppojen ostoskorit, verkkopankkien käyttö, sivuilla nähtävät mainokset ja vaikkapa videoistosisivustojen videot sisältävät käyttöliittymän.

Kun sovellukset, ohjelmistot ja pelit monimutkaistuvat, on varsinkin pienempien yritysten hyvä käyttää apunaan sellaisia kehitystyökaluja, jotka suorittavat osan koodista valmiiksi. Tämä auttaa varsinkin sellaisia sovelluskehittäjiä, jotka joutuvat tekemään montaa eri asiaa, eikä välttämättä pysty erikoistumaan tietylle asialle. Tällaisia ohjelmia ilmestyy markkinoille jopa ilmaisversioina, joten niiden opettelu ja käyttö saattaa tehostaa työnopeutta huomattavasti.

2.1 Adobe Flash

Flash-tekniikka on ollut viime vuosien ajan suuressa suosiossa ohjelmoijien ja käyttäjien keskuudessa internetpelejä ja –mainoksia luotaessa. Yksi iso syy tähän on Adobe Flash -ohjelman helppokäyttöisyys, joka perustuu pitkälti WYSIWYG-periaatteeseen (eng. What You See Is What You Get, ”se mitä näet, sen saat”). Tällaisella käyttöliittymällä ei tarvitse jokaista ympyrää ja tekstilaatikkoa erikseen koodata tiettyyn paikkaan, vaan sen saa kätevästi piirrettyä kanvaasille, ja ohjelma teki siihen liittyvän koodin. [1]



Kuva 1. Adobe Flashin käyttöliittymä.

Kuvasta 1 huomataan, että Adobe Flash -ohjelman käyttöliittymä muistuttaa paljon muita Adoben sovelluksia. Kuvassa vasemmalla ovat piirtotyökalut, kuvan ylälaudassa aikajana ja keskellä kanvaasi, jonne sovellus luodaan.

Toinen Flashin vahvuuksista on aikajanan käyttö. Aikajanalla voidaan luoda animaatioita objekteille sekä käsitellä siirtymävaiheita sovelluksen sisällä animoituna tai ilman. Tämä voisi käsittää esimerkiksi ponnahtusikkunoiden avaus- ja lopetusanimaatiot tai peleissä kentältä toiseen siirtymisen. Animaatioilla voidaan myös suorittaa 2D-kuvien animointia, kuten pelihahmon liikettä tai pelin taustakuvan liikettä.

Applen suositut tuotteet kuten iPod, iPhone ja iPad eivät kuitenkaan tue flash-tekniikkaa, joten iso osa mobiilikäyttäjistä ei pääse koskaan näkemään työtä, joka flash-sovelluksiin laitetaan. Tämän lisäksi useimmissa Android-sovelluksissa voidaan flash-animaatiot poistaa erikseen käytöstä internetsivuja selattaessa, jotta sivujen selaaminen on nopeampaa kuin sivuissa, joiden laidassa pyörii esimerkiksi flashanimaatioilla toteutettuja mainoksia. Tämä saattaa aiheuttaa suuren käyttäjäkunnan puutteen varsinkin mobiilisovellusten kehityksessä.

2.2 Unity 3D

Unity 3d –ohjelmalla voidaan luoda pelejä, videoita, animaatioita, käyttöliittymiä ja prototyyppejä. Kaiken tämän takana on käyttöliittymä, joka toimii suurimmaksi osaksi ns. raahaa ja pudota –menetelmällä. Unityn peruskäyttö on ilmaista, mutta muille kuin Windows- tai Mac- alustoille julkaisuun vaaditaan maksulliset lisenssit.

Unityllä luodaan komentosarjoja, joita voi tehdä joko C#-, Javascript- tai Boo-ohjelmointikielillä. Kaikki kolme kieltä toimivat avoimen lähdekoodin .NET alustalla. [2]

Unityllä voidaan luoda 2d- tai 3d-maisemia ja se sisältää paljon esikoodattuja toimintoja. Näihin lukeutuvat mm. painovoima, kosketustunnistus sekä lukemattomia määriä ennalta tehtyjä kontrolli-, maisema- ja efektimoduuleja. Näiden avulla voidaan esimerkiksi nopeasti demonstroida sovelluksen toimintaa suorittamatta itse paljoakaan koodin kirjoittamista. [3]

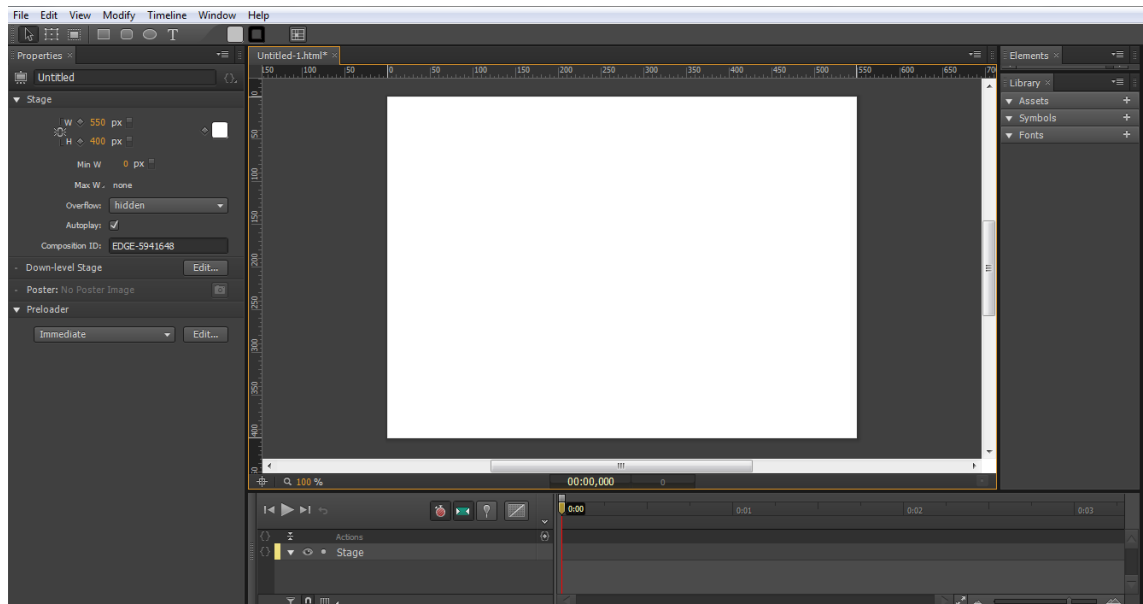
Yksi Unityn elementtejä on objektien tai kuvien tuonti projektiin leikkaa-liimaa periaatteella. Ohjelmaan on siis mahdollista tuoda piirretty kuva tai mallinnettu 3d-objekti kopioimalla se suoraan projektikansioon. Tästä aiheutuu se, että Unity päivittää tiedoston automaattisesti siihen kansioon, jonne se on laitettu. Unity tunnistaa animaatiot ja kuvat, ja ne voidaan liittää projektissa oleviin objekteihin esimerkiksi taustakuvina kai objektien kuvakartoiksi. [3]

Unityn suuria ominaisuuksia on sen mahdollisuus käyttää samaa koodia monelle eri alustalle. Unityllä voidaan luoda Windows-, Mac-, Windows phone-, Android- ja selainsovelluksia, kaikki samalla koodilla.

2.3 Adobe Edge Animate

Adobe Edge Animate on Adoben uusin sovellus, joka käyttää Javascript-ohjelmointikieltä. Se sisältää samantapaisen käyttöliittymän kuin Adobe Flash objektien piirtämisestä aikajanalla käyttöön. Koska Edgessä ei käytetä flashtekniikkaa, toimii sovellus natiivisti kaikilla uusimmilla selaimilla. [4]

Edgeen voidaan tuoda erikseen tehtyjä kuvia ja niitä voidaan animoida aikajanalla joko muuttamaan muotoaan, liikkumaan tai ilmestymään muualle. Javascript tarjoaa oliopohjaisen komentosarjakielen, joka antaa sille voimakkaan lähtökohdan esimerkiksi pelisuunnitteluun. [4]



Kuva 2. Adobe Edgen käyttöliittymä.

Kuvasta 2 huomataan, miten samankaltainen sovellus on esimerkiksi kuvan 1 Adobe Flashin käyttöliittymän kanssa. Kuvassa näkyvät moduulit ovat liikuteltavia, joten esimerkiksi kuvassa vasemmalla oleva *properties*-ikkuna voidaan halutessaan raahata keskellä olevan kanvaasin oikealle puolelle. Kuvan alalaidassa animaatiojana, jossa voidaan tehdä animaatioita ja muokata objektien ominaisuuksia ajan muuttujana.

Adobe Edgen eduksi lasketaan myös se, että Adobe on perunut uusimman Flashpohjaisen kehitystyökalun perusteenaan keskittyä enemmän muihin projekteihin [5]. Tästä voidaan päätellä, että Flashpohjaisten sovellusten määrä tulee tulevaisuudessa laskemaan kehittäjien siirtyessä muihin työkaluihin.

Voisikin sanoa, että kaikki sovellukset, jotka nykyään tehdään flashpohjaisena, kannattaa tulevaisuudessa hoitaa Edge Animatella. Pelkästään potentiaalisen käyttäjäryhmän lisääminen Applen mobiilimarkkinoilta tekee siitä kannattavaa. Lisäksi javascriptpohjaiset sovellukset ovat oletusarvoltaan nopeampia verrattuna flashsovelluksiin [4].

2.4 Kehitystyökalujen vertailu

Kaikilla esitellyillä sovelluksilla on hyviä sekä huonoja puolia. Mietittäessä, mikä työkaluista sopii parhaiten Turku Energialle luotavaan verkkosovellukseen, arvostetaan eniten työkalun mahdollisuutta palvella mahdollisimman laajaa käyttäjäkuntaa ilman ylimääräisiä liitännäisiä. Taulukosta 1 nähdään Kehitystyökalujen vertailua ominaisuuksien sekä alustatukien kanssa.

Taulukko 1. Kehitystyökalujen vertailu.

	Flash	Unity 3d	Edge Animate
Hinta			
Maksullinen	Kyllä	Kyllä	Ei
Maksuton	Ei	Kyllä	Kyllä
Alustatuki			
Selain	Vaatii liitännäisen	Vaatii liitännäisen	Kyllä
Android	Kyllä	Kyllä	Ei (toimii selaimen kautta)
iOS	Ei	Kyllä (maksullinen)	Ei (toimii selaimen kautta)

(jatkuu)

Taulukko 1 (jatkuu).

	Flash	Unity 3d	Edge Animate
Windows Phone	Ei	Kyllä (maksullinen)	Ei (toimii selaimen kautta)
Windows	Kyllä	Kyllä	Kyllä
Mac	Kyllä	Kyllä	Kyllä
Ominaisuudet			
Animointi	Kyllä	Kyllä	Kyllä
Testaus	Kyllä	Kyllä	Kyllä
Ohjelmointikieli	Actionscript	C#, Boo, Javascript	Javascript
Kuvien tuonti	Kyllä	Kyllä	Kyllä
Kehitystuki	Ei	Kyllä	Kyllä

Taulukon 1 mukaan Unity toimittaa laajimman skaalan ominaisuuksia sekä alustatukea. Flash puolestaan on muihin työkaluihin verrattuna heikko, eikä se tarjoa maksutonta työkalua sovellusten kehittämiseen. Edge Animate eroaa muista työkaluista siten, että se ei suoraan tarjoa sovelluksen kääntämistä millekään mobiilialustalle. Edgen pääpaino onkin verkkototeutuksilla, ja sen tuottamat sovellukset toimivat selaimen kautta mobiilialustoillakin.

Turku Energian sovellusta suunniteltaessa päädyttiin käyttämään Adobe Edge Animatea, sillä se tarjoaa suurimman potentiaalisen sovelluksen käyttäjäryhmän. Ratkaisevina tekijöinä olivat myös ne, että Edgen käyttö on ilmaista, se edellyttää sovelluskehittäjältä vain Adoben Cloud- palveluun rekisteröitymisen eikä se tarvitse ylimääräistä liitännäistä toimiakseen kaikilla suosituimpien selaimien uusimilla versioilla.

3 KÄYTTÖLIITTYMÄSUUNNITTELU

Käyttöliittymällä tarkoitetaan tietokoneiden, sovellusten, koneiden, mobiililaitteiden tai internetsivustojen käyttäjän ja laitteen välistä interaktiivisuutta. Yksinkertaisimmillaan käyttöliittymä voisi olla oven kahva, jota pyöräytettäessä oven lukko avautuu. Enemmän toimintoja sisältävä sovellus voisi olla vaikka Adobe Photoshop- ohjelma, jossa käyttäjä luo kuvia piirtämällä erilaisia muotoja kanvaasille käyttöliittymän kautta. Mitä enemmän toimintoja käyttöliittymä tarjoaa, sitä monimutkaisempaa sen käyttäminen todennäköisesti on.

Käyttöliittymäsuunnittelu on sovelluksen yksi tärkeimmistä vaiheista, sillä liian vaikea käyttöliittymä johtaa usein turhautumiseen ja jopa sovelluksen käytön lopettamiseen. Käyttöliittymää suunniteltaessa pitää ottaa huomioon asioita, joita sovelluksen käyttäjä ei välttämättä huomioi. Tällaisia asioita voivat olla esimerkiksi värien käyttö, oikopolkujen käyttö, valikon sijainti ja monimutkaisuus sekä ohjesivun laajuus ja helppokäyttöisyys. [6]

Suunnittelussa pitää ottaa myös huomioon käyttäjäkunta. Mikäli kehitellään sovellusta värisokeiden ihmisten verkkosivuille, täytyy ottaa huomion käyttäjäkunnan rajoitteet värien suhteen. Samoin suunniteltaessa sovellusta tietyille ikäluokalle, kuten vanhuksille tai pienille lapsille, on hyvä tehdä käyttöliittymästä yksinkertaisempi kuin suunniteltaessa vaikkapa kehitystyökalua insinööreille. [6] [17]

3.1 Käyttöliittymän suunnittelutavat

Käyttöliittymän suunnittelussa voidaan käyttää erilaisia lähestymistapoja riippuen siitä, mihin käyttöön sovellus asetetaan. Suunnittelussa ei ole oikeaa eikä väärää tapaa, ja lopputulos ratkaisee. Tässä työssä esitellään yleisimpiä käyttöliittymän suunnitteluun käytettyjä metodeja.

3.1.1 Toiminnallisten vaatimusten kerääminen

Yksi suosituista suunnittelutavoista on aloittaa sovellukselle tärkeiden ominaisuuksien listaamisella. Listaan kerätään kaikki ominaisuudet, joita sovelluksella voi tehdä. Tämä auttaa ymmärtämään sovelluksen kokoa ja kokonaisuutta sekä ymmärtämään määrää, joka käyttäjän pitää opetella oppiakseen sovelluksen tehokkaan käytön. [7]

Listan avulla voidaan suunnitella käyttöliittymän muotoa ja eri asioiden loogista sijaintia sovelluksen sisällä. Kuvasta 2 nähdään esimerkkilista sekuntikellon toiminnallisuudesta.



Kuva 3. Vasemmalla sekuntikellon toiminnallisuuslista ja oikealla kuvakaappaus valmiista sovelluksesta.

Kuvasta 3 voidaan todeta sovelluksen olevan melko pieni. Kaikkien listattavat ominaisuudet voidaan näyttää käyttöliittymässä, eikä piilotettuja ominaisuuksia tarvita. Käyttäjän on näin ollen helppo ymmärtää sovelluksen tarkoitus ja opetella sen käyttö tehokkaasti.

3.1.2 Käyttäjäänalyysi

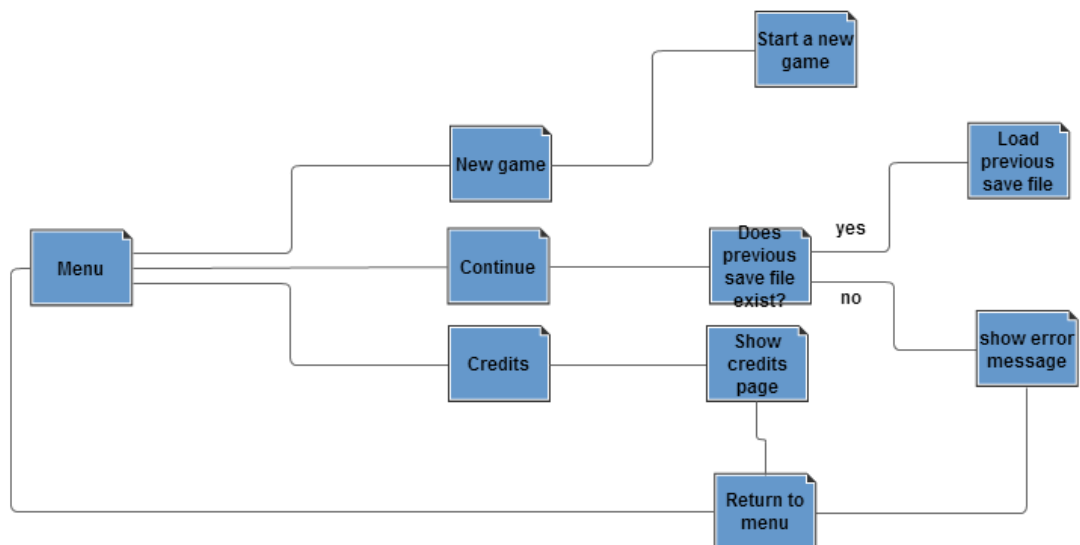
Käyttäjäänalyysissä mietitään potentiaalisia käyttäjiä. Analyysin tekemisessä voidaan jopa kysyä kysymyksiä itse käyttäjiltä ja kerätä sitä kautta tietoa. Mikäli tämä ei ole mahdollista, voidaan myös itse kehittää kysymyksiä joihin mietitään mahdollisimman todenmukainen ja perusteltu vastaus.[8][15] Kysymykset voivat olla esimerkiksi seuraavanlaisia: [15]

- Mitä käyttäjä haluaisi, että sovellus tekee?
- Miten sovellus sopii käyttäjän muuhun työpäivään tai miten sovellus sopii yhteen muiden käyttäjän käyttämien sovellusten kanssa?
- Millainen on potentiaalisten käyttäjien tekninen tausta ja osaaminen?
- Millainen graafinen ulkoasu miellyttää potentiaalista käyttäjäkuntaa?
- Miten sovellus eroaa muista samankaltaisista sovelluksista tai mitä uutta sovellus tarjoaa sille tarkoitetulla alalla?

Tämänkaltaiset kysymykset voivat auttaa suunnittelijaa ymmärtämään kohdekäyttäjien tarpeen. Vastausten perusteella voidaan alkaa suunnittelemaan ohjelman prototyyppiä.

3.1.3 Prototyypitys

Tehtäessä sovelluksia, joissa on paljon ominaisuuksia, voi olla vaikeaa hahmottaa, kuinka iso sovelluksesta oikeasti tulee. Tätä helpottamaan voidaan luoda vuokaavioita, joista selviää kytkökset tiettyjen ominaisuuksien välillä. Vuokaavio auttaa myös selvittämään, mitä kaikkea sovellus oikeasti tekee ja mihin tarkoitukseen se vastaa (Kuvio 1). [8] [15]



Kuvio 1. Esimerkki pelin alkuvalikon vuokaaviosta.

Kuviosta 1 nähdään esimerkkivuokaavio pelin alkuvalikosta. Valikossa on kolme vaihtoehtoa: *New game*, *Continue* ja *Credits*. Valittaessa *New game* peli aloittaa uuden pelin. Valittaessa *Continue* ohjelma tarkastaa onko aikaisempaa tallennusta olemassa. Mikäli aiempi tallennus löytyy, ohjelma lataa aikaisemman pelitallennuksen. Mikäli aiempaa tallennusta ei löydy, peli antaa virheilmoituksen ja palaa alkuvalikkoon. Valittaessa *Credits* ohjelma näyttää pelin krediittisivuston, jonka jälkeen se siirtyy jälleen alkuvalikkoon.

Prototyypitystä voi myös tehdä paperilla ja kynällä. Valikot voidaan leikata palasiksi ja koko sovellusta voidaan tarkastella yksinkertaisella metodilla, joka ei vaadi sovelluskehittäjiltä graafisen käyttöliittymän luomista. [8]

3.2 Käyttöliittymän testaus

Käyttöliittymä on tärkeä testata, jotta käyttäjät saavat mahdollisimman viimeistellyn sovelluksen itselleen. Käyttöliittymätestauksessa voidaan saada monenlaista informaatiota, kuten miltä sovellus näyttää, miltä sen käyttäminen tuntuu ja miten hyvin se vastaa käyttäjän odotuksia.

Yrityksen sisäinen testaus on tärkeää, jotta mahdollisimman monet viat saadaan korjattua. Varsinkin pelisovelluksissa yritykset suosivat kaksitasoista testausta, alfa- ja betatestausta. [16]

Alfatestauksessa sovellus on vielä hyvin raaka ulkonäöltään, mutta sen toiminnallisuus on valmis. Tässä testausvaiheessa yrityksen omat työntekijät ja varsinaiset kehittäjät etsivät sovelluksesta mahdollisimman paljon virheitä, jotka korjataan ennen seuraavaa testausta. Kun löydetyt virheet on korjattu, voidaan alfatestaus suorittaa uudestaan uusien virheiden löytämiseksi. Testauksen uusiminen on tärkeää aina siinä vaiheessa, kun sovellus saa uusia ominaisuuksia. [8] [16]

Alfatestauksen jälkeen suoritetaan yleensä betatestaus. Betatestaus voidaan suorittaa joko suljetusti vain kutsuille henkilöille tai avoimesti, jolloin kuka tahansa voi liittyä testiryhmään. Testikäyttäjät lataavat tai saavat muuten

sovelluksen käyttöönsä, käyttävät sitä tilanteesta riippuen joko kotonaan omalla ajalla tai kutsutussa paikassa valvottuna. [8] [16]

Betatestauksen aikana ei käyttäjälle yleensä anneta suoria tehtäviä, vaan käyttäjän tekemiset tallennetaan ja informaatiota verrataan muihin käyttäjiin. Tällaista tallennettua informaatiota voi olla esimerkiksi sovelluksen käyttöaika, mitä asioita käyttäjä sovelluksella teki, kuinka nopeasti käyttäjä sai suoritettua tietyn tehtävän tai missä vaiheessa käyttäjä lopetti sovelluksen käytön. Tämänkaltaiset asiat antavat sovellus- ja pelinkehittäjille erittäin tärkeää informaatiota sovelluksen käyttäjästä. Mikäli iso osa käyttäjistä lopettaa pelin pelaamisen tietyssä kohtaa, voidaan miettiä miksi näin on ja mitä voidaan tehdä kyseisen kohdan parantamiseksi.[8] [16]

Kun käyttäjät testaavat sovellusta, voidaan myös käyttää niin sanottua ääneenpuhumisprotokollaa. Tätä tekniikkaa käytettäessä testaaja puhuu ääneen kokemuksistaan, tuntemuksistaan ja muista asioista käyttäessään sovellusta. Testaajalle voidaan luoda tarkat ohjeet siitä, miten hän toimii sovellusta käyttäessään. Tänä aikana testaaja puhuu ääneen, miltä kunkin tehtävän suorittaminen tuntuu ja miten sovellus antaa testaajalle palautetta. [9]

Testaajaa voidaan myös kuvata ja tietokoneesta voidaan tallentaa esimerkiksi hiiren osoittimen sijainti testauksen eri vaiheissa. Tämä auttaa sovelluskehittäjiä ymmärtämään, miten loppukäyttäjät käyttävät sovellusta ja tehdä siihen tarvittavia muutoksia, mikäli siihen löytyy tarvetta. [9]

4 ESITTELYSOVELLUKSEN SUUNNITTELU

Turku Energia Oy on Suomen suurimpia energiayhtiöitä. Sen toimintaan kuuluu sähkön ja lämmön tuotanto, myynti sekä jakelu ja niihin liittyvät palvelut. Tuotteita ovat sähkö, lämpö, höyry ja kaukojäähdytys. [10]

Turku Energia esittää itsensä julkisuuteen nuorekkaana ja vastuullisena energiayhtiönä, joka kantaa vastuunsa ympäristöstä. Turku Energian ohjaaviin arvoihin kuuluu mm. asiakaslähtöisyys, kannattavuus, toimintavarmuus, ympäristöystävällisyys sekä yhteiskunnallinen toimivuus. [10]

Nykytilanteessa Turku Energia mainostaa itseään työnantajana mm. kiertämällä kouluja, järjestämällä tapaamisia Turku Energian tiloissa sekä esittelemällä yritystä messuilla sekä mediassa. Varsinkin koululaisille esitettävässä materiaalissa oli kuitenkin puutteita. Materiaali itsessään on vanhentunutta ja niiden tehokas hyödyntäminen oli tehty turhan hankalaksi. Erityisesti parannettavaa oli kohderyhmäkohtaisissa materiaaleissa sekä niiden esittelytavassa. Vanhentunut materiaali sisälsi noin 40 sivun PowerPoint –esityksen, joka käsitteli tietoa mm. yrityksen historiasta, organisaation eri osastoista sekä tietoa kodin sähkönkulutuksesta ja sähkön tuottamiseen liittyvistä toimenpiteistä.

4.1 Tavoite

Tämän sovelluksen tarkoitus oli päivittää esittelymateriaali kohderyhmille sopivaksi yhdeksi sovellukseksi, josta voidaan esittelytilanteen mukaan valita sopivat esittelymateriaalit. Sovellusta varten luodaan käyttöliittymä, jotta sitä voidaan tehokkaasti näyttää myös messuilla sekä muissa esityksissä.

Sovelluksen tarkoituksena on myös esitellä Turku Energian tärkeimmät rakennukset, antaa tietoa sähkön jakelusta sekä kertoa Turku Energian eri työuramahdollisuuksista. Sovelluksessa on mahdollista katsoa videoita, kuvia

sekä tekstiä, jotka kaikki antavat oman osuutensa informaatiosta. Sovellukseen laitettiin seuraavat kohteet:

- Toimitalo
- Naantalin voimalaitos
- Lämpöpumppu
- Kaasukello
- Myllyahde
- Orikedon biopolttolaitos
- Lämpötuotanto (yleisesti)
- Sähkölinjat (yleisesti).

Sovellus rakennetaan karttanäkymälle niin, että klikattavat rakennukset sijoittuvat kartalla oikeille paikoilleen. Lisäksi sovellukseen lisätään navigointipalkki, jonne lisätään muita tietolähteitä, joita ei voida kartalle asettaa. Nämä tietolähteet sisältävät linkit *historia*, *yritys*, *energia-ala* sekä *info*. Tämän lisäksi navigointipalkkiin sijoitetaan kaksi painiketta, joita painamalla kartalle piirretään summittainen ketju sähkönsiirrosta tuotantolaitokselta kotitalouteen sekä kaukolämmön ja kaukojäähdytyksen toimituksesta tuotantolaitokselta kotitalouteen. Molemmat piirrokset käyttävät ketjussaan hyväksi kartassa jo olevia rakennuskohteita.

4.2 Kilpailija-analyysi

Osalla suomalaisista yrityksistä on jo olemassa sovelluksia, jotka joko opastavat käyttäjää tietyissä asioissa tai esittelevät omaa yritystään ja tuotantoaan. Tällaisten sovellusten tarkoitus on yleensä tuoda yritystä esille ihmisten keskuudessa. Lisäksi erilaiset visailut ja kisat tuottavat kävijöitä internetsivuille ja voivat sitä kautta toimittaa jopa uusia asiakkaita.

4.2.1 Energiavisa

Helsingin Energian tuottama energiavisapeli on Haluatko Miljonääriksi -visailun muotoon tehty peli, jossa pelaajalle esitetään kysymys ja neljä vastausvaihtoehtoa. Pelaajan täytyy vastata kysymykseen tietyn ajan sisällä tai muuten peli päättyy. Mikäli pelaaja vastaa oikein, saa hän pisteitä ja peli jatkaa seuraavaan kysymykseen. Mikäli pelaaja vastaa väärin, poistetaan häneltä pisteitä ja peli päättyy. [12]

Pelaajalla on myös käytössään kolme apupainiketta, joita voi käyttää yhden pelisession aikana kerran. Poista kaksi -vaihtoehto poistaa vastausvaihtoehdoista kaksi väärää vastausta, Vihje- vaihtoehto antaa pelaajalle vihjeen, josta voi päätellä oikean vastauksen ja Skippaa- vaihtoehdossa peli vaihtaa kysymyksen ja vastausvaihtoehdot (Kuva 4). [12]



Kuva 4. Energiavisian pelinäköymä.

Kaikki energiavisassa olevat kysymykset ovat sähkönkulutukseen tai – tuottamiseen liittyviä. Pelin tarkoituksena on opastaa käyttäjiä

sähkönkulutukseen liittyvistä asioista sekä antaa vinkkejä siitä, miten oman kodin energiatehokkuutta voi parantaa.

Pelityyppiset ratkaisut tuovat usein sovellukseen käyttöikä, sillä jotkut pelaajat käyttävät sitä useamman kerran ja palaavat sivuille myöhemmin. Lisäarvoa voidaan lisätä, jos parhaille pelaajille tarjotaan esimerkiksi kuukausittain vaihtuva palkinto. Tämä antaa pelaajille syyn palata takaisin ja yrittää rikkoa aikaisemmin luotuja ennätyksiä.

Tämäntyyppinen peli vaatii kuitenkin muutoksia tasaisin väliajoin, jotta käyttäjät eivät opettele kysymyksiä ulkoa ja siten saavuta parhaita mahdollisia pistemääriä. Toinen vaihtoehto olisi luoda niin laaja kysymysmäärä, ettei niitä voi muistaa ulkoa. Kysymyspelit kuitenkin vanhenevat, ellei niihin tuoteta uutta sisältöä.

4.2.2 Together we can do it

Together we can do it on YIT:n tuotannosta kertova sovellus. Sovellukseen on sekoitettu videokuvaa, valokuvia sekä piirrettyä kuvaa. YIT on jakanut tehtävänsä kuuteen eri prosessivaiheeseen, joita voidaan sovelluksessa tutkia yksi kerrallaan. [13]



Kuva 5. YIT:n yritystoimintaa esittelevän sovelluksen näkymä.

Kuvassa 5 nähdään alkuvideon jälkeinen sovellusnäkyvä, jossa YIT:n toimintaperiaate on jaettu kuuteen eri osa-alueeseen. Näitä alueita voidaan klikata, jonka jälkeen avautuu uusi näkymä keskittyen pelkästään valittuun osa-alueeseen. Uudessa näkymässä käyttäjälle annetaan lyhyt lause sekä videollinen selitys kyseisestä prosessivaiheesta. [13]

Sovellus käyttää hyvin paljon videota ja vaatiikin käyttäjää pitämään ääniä päällä toistolaitteessaan. Teksti sovelluksen sisällä on hyvin pienimuotoista ja koostuu lähinnä yksittäisistä pienistä lauseista. Tämä toimii hyvin suljetussa ympäristössä esimerkiksi kotona, mutta ei sovellu täysipainoisesti tiloihin, jossa voi jo valmiiksi olla taustahälinää. Tämänkaltaiset paikat voivat olla erilaisia messuja tai tapahtumia, joissa halutaan esitellä yritystä potentiaalisille uusille asiakkaille.

Sovellusta luodessa ei ole myöskään ole täysipainoisesti otettu huomioon kaikkia eri käyttäjäryhmiä, erityisesti kuuroja. Koska sovellus käyttää paljon videosiittelyä ja puhetta, ei kuurot tai huonokuuloiset saa sovelluksesta täysipainoista informaatiota, vaan joutuvat tyytymään jokaisessa projektiesittelykohdassa yhteen tai kahteen lauseeseen.

4.2.3 Interactive House

Interactive House on Florida Power and Light -yrityksen luoma 3D-mallinnukseen perustuva sovellus, jossa käyttäjä tekee virtuaalisen kierroksen kuvitteelliseen omakotitaloon. Taloon sekä sen ulkopuolelle on lisätty pisteitä, joita klikkaamalla saa lisätietoa kyseisen kohteen energiankulutuksesta sekä vinkkejä energiankulutuksen vähentämiseen. [18] Kuvassa 6 sovelluksen käyttöliittymä.



Kuva 6. Interactive House –sovelluksen käyttöliittymä.

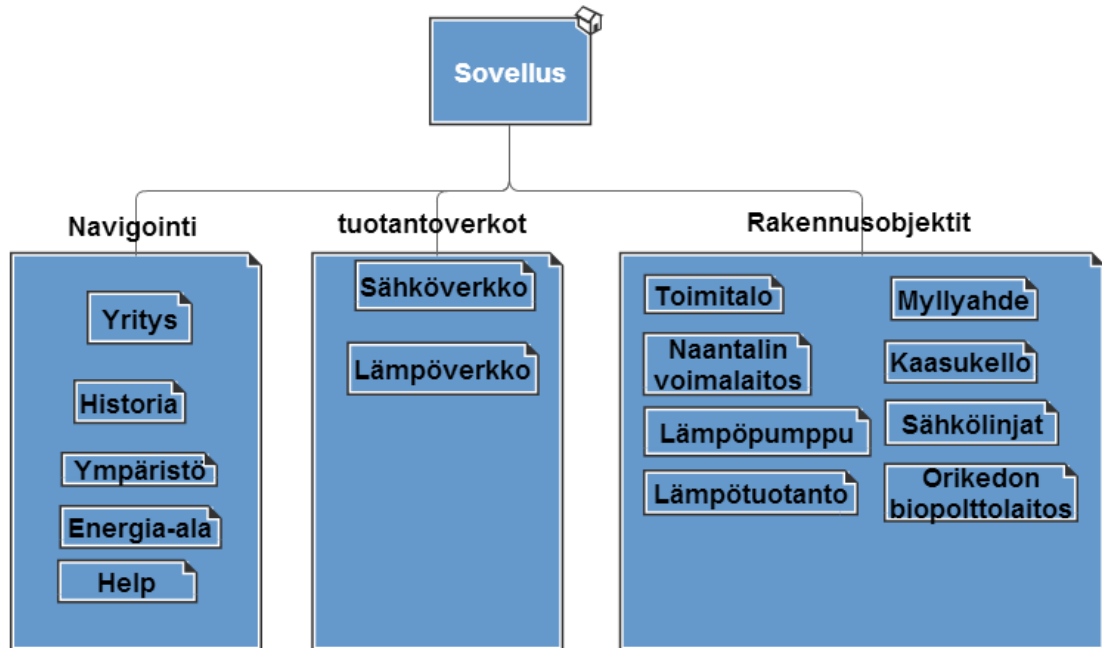
Kuvan 6 yläaidassa olevalla navigointipalkilla voidaan sovelluksessa siirtyä talon eri huoneisiin, kuten olo- tai makuuhuoneeseen tai keittiöön. Jokaisessa näkymässä on kuvaan lisätty pisteitä, joita painamalla avautuu ponnahdusikkuna, joka kertoo kyseisen kohtaan liittyviä energiansäästövinkkejä. [18]

Vaikka sovellus ei esittele yrityksen toimintaa tai houkuttele käyttäjiä palkinnoilla, antaa se kuitenkin paljon jälleenkäyttöarvoa kaikille niille, jotka ovat kiinnostuneita oman kodin sähkölaskun pienentämisestä tai ympäristön suojelemisesta. Sovelluksessa kerrottavat asiat ovat myös ajankohtaisia ja hyvin suunniteltuja, jotka koskevat lähes kaikkia asukkaita. Tämä nostaa potentiaalisen käyttäjäryhmän erittäin korkeaksi.

Sovellus voi kuitenkin olla raskasta käyttää varsinkin heikommilla tietokonetehoilla, joten sen käyttöä voisikin suositella vain kotikoneilla, eikä esimerkiksi mobiililaitteilla.

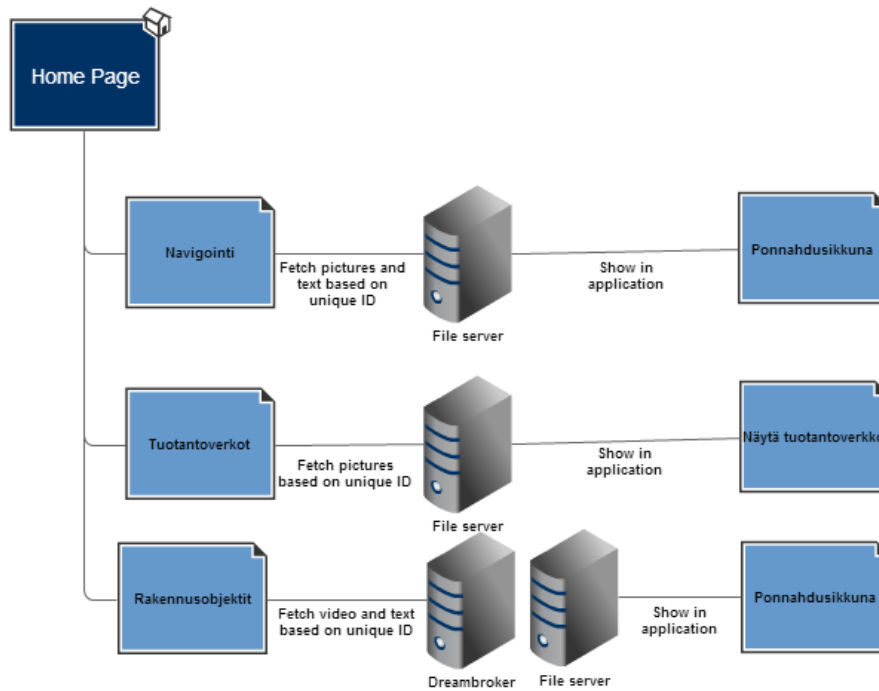
4.3 Sovelluksen sisällön hahmottaminen

Sovelluksen sisällä olevat objektit jaettiin kolmeen eri kategoriaan: navigointiin, tuotantoverkkoihin sekä rakennusobjekteihin (kuvio 2). Näin voidaan jokaiselle kategorialle suunnitella oma tyyli kertoa informaatiota ponnahdusikkunoiden muodossa.



Kuvio 2. Sovelluksen objektien kategoriointi vuokaaviossa.

Kuvion 2 mukaan jaetut objektit voidaan koodausvaiheessa asettaa toimimaan omilla kategorioillaan, jotta niistä kertovat informaatiot saadaan aukeamaan oikeassa ponnahdusikkunassa (kuvio 3).



Kuvio 3. Sovelluksen ponnahdusikkunoiden tiedonhaku.

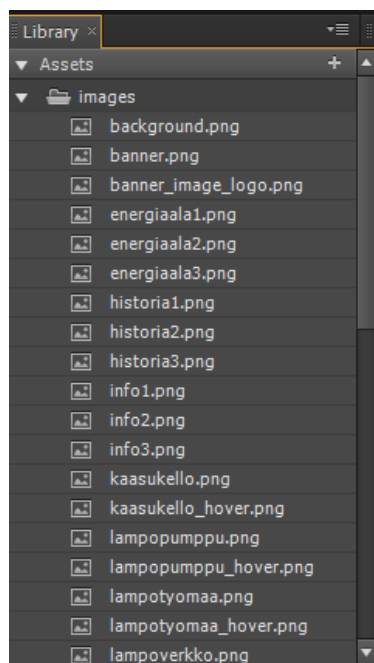
Kuvion 3 mukaan Navigointi- sekä Tuotantoverkot -kategoriat tarvitsevat vain tiedostoserveriä, sillä ne eivät sisällä videomateriaalia. Koska Turku Energialla on jo käytössä Dreambroker-ohjelmisto, täytyy sovelluksessa käytettävät videot sisällyttää erillisille serverille.

5 ESITTELYSOVELLUKSEN TOTEUTUS

Sovelluksen piti toimia lähes kaikilla selaimilla ilman ylimääräisiä liitännäisiä, jotta sen käyttäjäpotentiaali saadaan mahdollisimman korkeaksi. Koska HTML5 –merkintäkieli ja Javascript- ohjelmointikieli toimivat natiivisti kaikkien suosituimpien selaimien uusimmissa versioissa[11], luotiin sovellus Adobe Edge Animate –ohjelmalla ja sen toteutukseen käytettiin html5, javascript- ja jQuery-ohjelmointikieliä. Lisäksi rakennusten luomiseen käytettiin 3ds max- 3d mallinnusohjelmaa. Kaikkiin kuvankäsittelyihin käytettiin Adobe Photoshop-ohjelmaa.

5.1 Kuvahakemiston luonti

Ensin luotiin kuvat, joita sovelluksessa käytetään. Tähän sisältyi kaikki navigointipalkin kuvat, toimipisteet sekä sähkönsiirtolinjat sekä lämpö- ja jäähdytysputket. Nämä lisättiin sovelluksen kansiorakenteeseen joka noudettiin Edgen kirjastoon sovelluksessa käytettäväksi kuvan 7 mukaisesti.

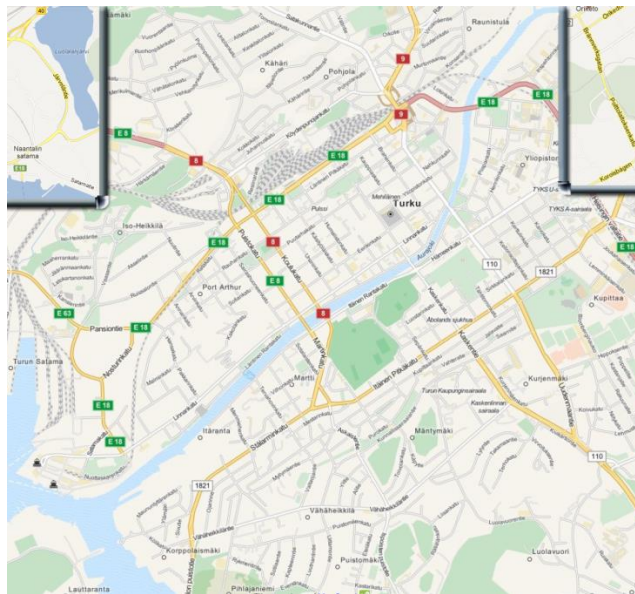


Kuva 7. Kuvakaappaus Adobe Edgen kirjastosta.

Kuvan 7 kirjaston kuvat on nimetty sisältönsä mukaan, ja ne haetaan koodissa aina nimensä mukaan. Tämä tarkoittaa sitä, että jos tulevaisuudessa haluaa vaihtaa sovelluksessa käytettäviä kuvia, riittää vain kansiorakenteeseen uuden kuvan lisääminen sillä nimellä, johon kuvan haluaa laittaa.

5.2 Sovelluksen taustakuva sekä elementtien sijoitus

Koska sovelluksessa käsitellään rakennuksia, jotka ovat kohtuullisen lyhyen matkan päässä toisistaan, tuntui luontevalta asettaa sovelluksen taustakuvaksi Turun kartta. Sovellus kuitenkin sisältää kaksi kohdetta Turun alueen ulkopuolelta, joten karttaan asetettiin reunat kuvan 8 mukaisesti. Lisäksi reunojen sisälle asetettiin kunkin alueen kartat niiltä kohdilta, jossa kohde oikeasesti sijaitsee.



Kuva 8. Turku Energia- sovelluksen taustakuva.

Kuvan 8 yläkulmissa näkyvät reunat tehtiin Photoshopilla, ja ne pyrittiin tekemään putkien näköiseksi teemaan sopivaksi. Kartta otettiin kuvakaappauksella karttasovelluksesta, jonka materiaali on vapaasti käytettävissä. Koska sovelluksen karttarive on alueena varsin pieni, ei Google Maps –sovellusliittymää koettu tarpeelliseksi. Vasemmanpuoleisen kehiksen

sisällä on kartta naantalin voimalaitoksen alueelta sekä oikeanpuoleisessa kartta orikedon biopolttolaitoksen alueelta. Keskellä näkyvä iso kartta-alue on Turun keskusta-aluetta lähimaastoineen.

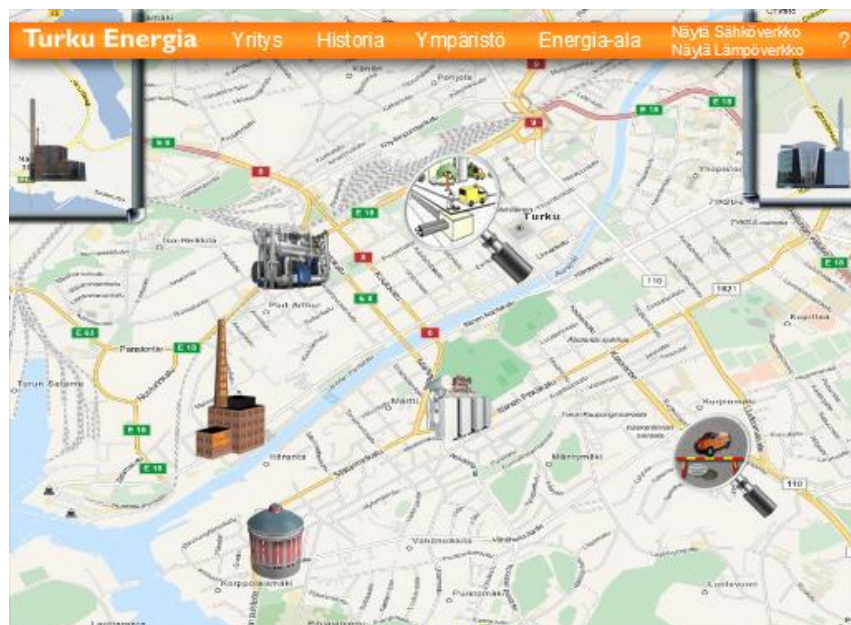
Navigointipalkki sijoitettiin ohjelmassa yläreunaan, koska se on käyttöliittymäsuunnittelun kannalta tutuin vaihtoehto. Näin ollen Turku Energian logo on näkyvillä vasemmassa yläreunassa ja mikäli käyttäjä tarvitsee apua sovelluksen käytöstä, infopainike sijaitsee sovelluksen oikeassa yläreunassa, kuten kuvasta 9 voidaan huomata.



Kuva 9. Turku Energia -sovelluksen navigointipalkki.

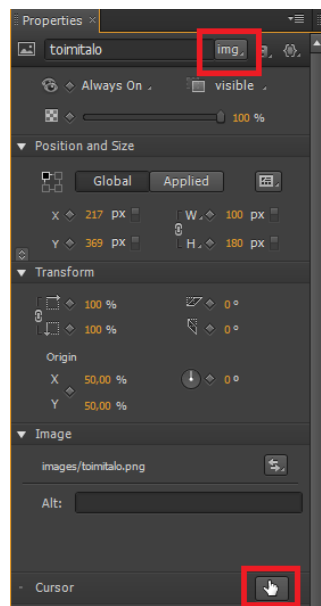
Kuvan 9 Turku Energia -teksti on sijoitettu kuvaan eikä ole erikseen painike. Muut kuvassa näkyvät tekstit ovat painikkeita, joista avautuu ponnahdusikkuna lisäinfoineen.

Tämän jälkeen rakennukset sijoitettiin niille paikoilleen kuin ne kartassakin kuuluvat kuvan 10 mukaisesti.



Kuva 10. Turku Energia- sovelluksen aloitusnäkyvä.

Kuvan 10 mukaisen rakennusten sijoittelun jälkeen sovellus on aloitusnäkymltään valmis. Kun käyttäjä vie hiiren painikkeen objektin päälle, on kuitenkin tarpeellista ilmoittaa käyttäjälle, että objektia kannattaa klikata. Tätä varten täytyy jokainen kuva asettaa kuvatunnisteihin sekä muuttaa hiiren kursori erilaiseksi kuvan ominaisuuksista kuvan 11 mukaisesti.



Kuva 11. Toimitalo- rakennuskuvan ominaisuudet- ikkuna.

Kuvan 11 ylempi punainen suorakulmio osoittaa paikan, jossa Edgessä muutetaan objektin tyyppi. Tässä tilanteessa tyyppiä asetetaan *img*, joka tarkoittaa kuvaa. Näin ollen sitä voidaan myöhemmin koodissa kutsua kuvanimellä. Alempi punainen suorakulmio näyttää, minkälaiseksi kursori muuttuu, kun se viedään kyseisen objektin päälle.

Edellisten toimenpiteiden lisäksi tehdään toiset versiot kaikista kuvista niin, että niiden ympärille laitetaan Turku Energian värien mukainen oranssi halo. Kuva vaihdetaan halolliseen kuvaan aina, kun käyttäjä vie hiiren kursorin kuvan päälle ja vaihdetaan takaisin alkuperäiseen kuvaan, kun kursori poistuu kuvan päältä. Jokaiseen kuvaan lisätään tämä ominaisuus lisäämällä kuviin koodin 1 mukaisesti.

Koodi 1. Kuvan vaihtaminen kun hiiren kursori on kuvan päällä.

```

Symbol.bindElementAction(compId, symbolName, "${_kaasukello}", "mouseover",
function(sym, e) {
$(this.lookupSelector("kaasukello")).attr("src","images/kaasukello_hover.png").animate({
"up": "-5", "opacity": 1}, 500);

});

Symbol.bindElementAction(compId, symbolName, "${_kaasukello}", "mouseout", function(sym,
e) {

// kun hiiren kursori lähtee kuvan päältä, näytetään kuva missä ei ole haloa

$(this.lookupSelector("kaasukello")).attr("src","images/kaasukello.png").animate({
"down": "5", "opacity": 1}, 500);

});

```

Koodi 1 asetetaan jokaiseen kuvaan, johon haloefekti halutaan käyttöön. Lisäksi koodi lisää hyvin pienimuotoisen animaation, jossa kuva liikkuu 5 pikseliä ylös ja alas, jotta käyttäjä vieläkin paremmin huomaisi, että kohdetta kannattaa klikata hiiren painikkeella.

5.3 Informaatioikkunoiden luonti

Ohjelmaa varten piti luoda kaksi erilaista aukeavaa ikkunaa: toinen navigointipalkin linkeille ja toinen rakennuksista aukeaville linkeille, koska niiden sisältö on erilainen. Navigointipalkin linkeille ei tarvittu erikseen videota, mutta aukavassa ikkunassa näytetään tekstin lisäksi kolme kuvaa. Rakennuksista aukeavalle ikkunaan laitettiin tekstin viereen upotettu videosoitin Dreambroker-ohjelmasta, jossa Turku Energia säilyttää omia videoita.

5.4 Navigointipalkin ponnahdusikkuna

Navigointipalkissa on viisi painiketta, josta ponnahdusikkunan pitää aueta; Yritys, Historia, Ympäristö, Energia-ala sekä ?-merkki, joka kuvastaa inforuutua.

Jokaisesta painikkeesta avautuu samanlainen ponnahtusikkuna, mutta eri teksteillä ja kuvilla. Ensin luotiin Edgellä suorakulmio sovelluksen keskelle ja nimettiin se *popup_info* nimiseksi. Tämän jälkeen se muutettiin symboliksi painamalla siitä hiiren oikealla painikkeella ja valitsemalla *convert to symbol*. Tämän jälkeen se piti avata, jotta symbolin sisälle voidaan asettaa tekstit sekä kuvat. Symboli avataan painamalla sitä hiiren oikealla painikkeella ja valitsemalla *edit symbol popup_info*.

Symboli tarvitsee taustakuva, otsikon, tekstikentän, kuvapaikat sekä ponnahtusikkunan sulkupainikkeen.

Taustakuva

Taustakuvaksi tehtiin Photoshopilla kuva, joka on Turku Energian oranssin värinen josta keskeltä lähtee pyöreä valkoinen gradientti. Tämä pitää hyvin yllä Turku Energian värimaailmaa olematta kuitenkaan yksitoikkoinen. Se ei myöskään ole liian valkoinen, jotta siihen myöhemmin liitettävä teksti pysyy edelleen luettavana.

Otsikko

Otsikkoa varten piirrettiin Edgen tekstityökalulla symbolin ylälaitaansuorakulmio siten, että sulkupainikkeelle jäi vielä tilaa oikeasta yläkulmasta. Otsikkotekstiobjekti nimettiin *info_otsake* nimellä ja sille annettiin täyteteksti "otsake". Kirjainkooksi valittiin 35 pikseliä, ja tasaus asetettiin keskelle. Otsikon koodi luodaan koodin 2 mukaisesti.

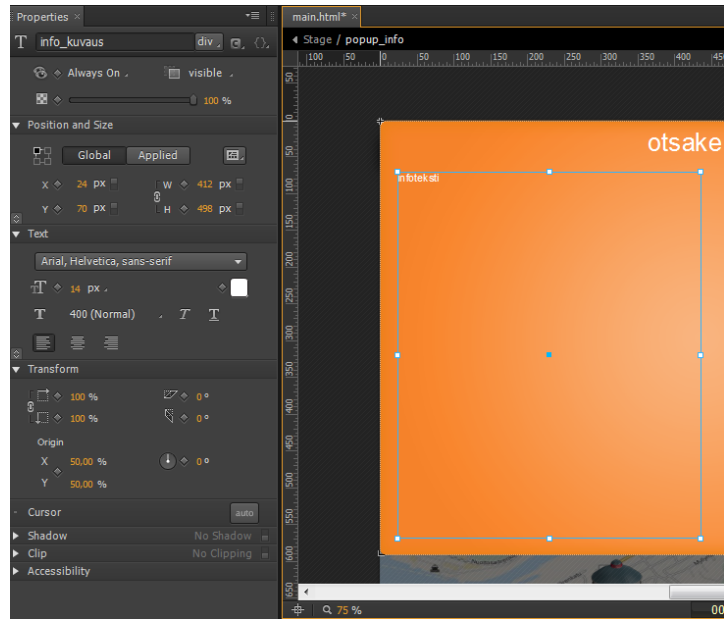
Koodi 2. Ponnahdusikkunan otsikon luonti.

```
"popup_info": {
  version: "1.0.0",
  minimumCompatibleVersion: "0.1.7",
  build: "1.0.1.204",
  baseState: "Base State",
  initialState: "Base State",
  gpuAccelerate: true,
  resizeInstances: false,
  content: {
    dom: [{
      type: 'text',
      rect: [24,9,780,38], //kohde alkaa pisteestä [24,9] ja on [780,38] pitkä ja leveä
      id: 'info_otsake',
      text: 'otsake',
      align: 'center',
      font: ['Arial, Helvetica, sans-
serif',35,'rgba(255,255,255,1)','normal','none','normal']
    }]
  }
}
```

Koodissa 2 luodaan *popup_info* objektille tekstikenttä, jonka id-tunnus on *info_otsake*. Teksti myös keskitetään, jotta se on aina ponnahdusikkunan keskiosassa.

Tekstikenttä

Tekstikenttä tehtiin samalla tavalla kuin otsikko, mutta pienillä muutoksilla. Tekstityökälulla vedettiin suorakulmio ponnahdusikkunan vasemmalle puoliskolle, jotta myöhemmin asetettava teksti asettuu ikkunan vasemmalle puolelle. Tekstin kirjasinkooksi valittiin 14 pikseliä ja tasaus vasemmalle. Tekstikentän nimeksi annettiin *info_kuvaus* ja sen täytetekstiksi laitettiin *infoteksti*. Kuvassa 12 kuvakaappaus tekstikohdasta ponnahdusikkunalla.

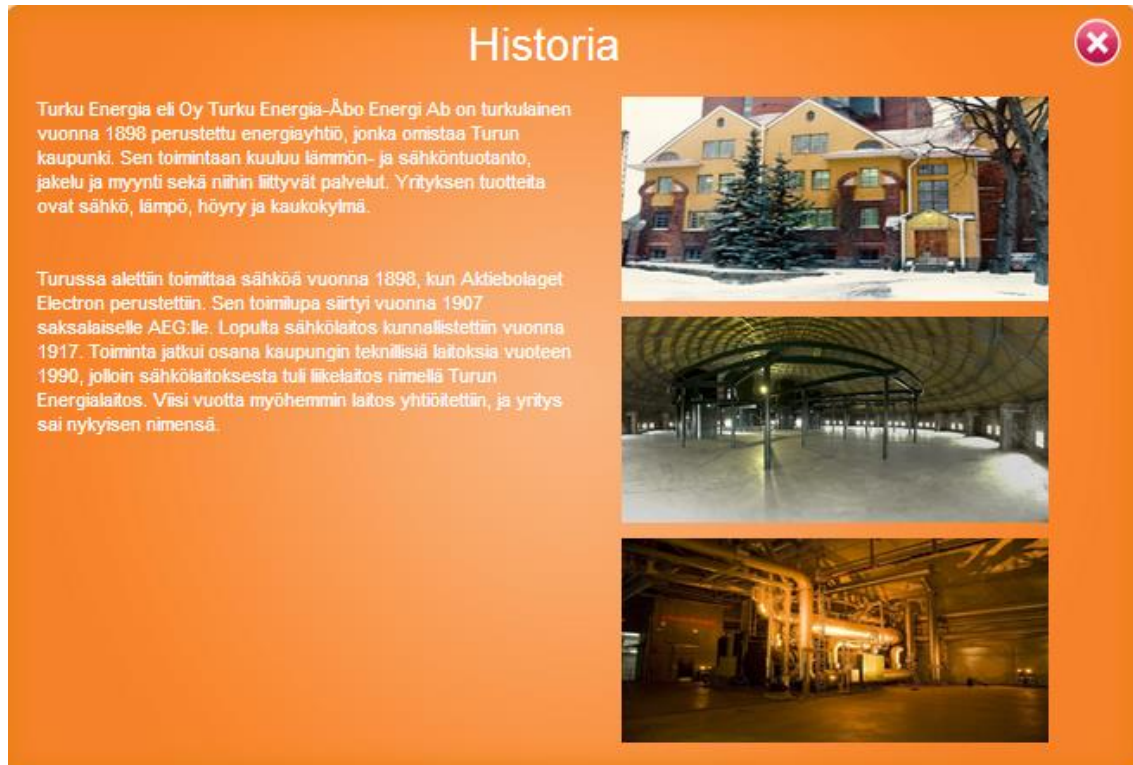


Kuva 12. Turku Energia- sovelluksen ponnahdusikkunan tekstilaatikko ja sen ominaisuudet.

Kuvan 12 mukaan tekstikenttä ylettyy ponnahdusikkunan puoleenväliin asti rajaten siihen myöhemmin kirjoitettavan tekstin. Tekstilaatikko asettaa myös rajoituksia sen suhteen, kuinka pitkä siihen kirjoitettava teksti voi maksimissaan olla.

Kuvapaikat

Kuvapaikat sijoittuvat ponnahdusikkunan oikealle puolelle. Niitä varten varattiin kolme paikkaa, johon sijoitettiin paikkamerkkeinä valkoiset kuvapaikat. Kuvassa 13 valmis Historia-lehden ponnahdusikkuna.



Kuva 13. Turku Energia- sovelluksen ponnahdusikkuna.

Kuvan 13 oikealla laidalla olevien kuvalaatikoiden koko on 329x158 pikseliä ja ne on sijoitettu allekain. Niissä voidaan näyttää aina silloiseen aiheeseen liittyvää informaatiota, esimerkiksi sertifikaatteja tai yleiskuvia.

Sulkupainike

Sulkupainikkeen tarkoitus on sulkea ponnahdusikkuna, jotta voidaan palata takaisin alkuperäiseen sovellusnäkymään. Sulkupainiketta varten piirrettiin kaksi eri versiota, punainen rastipainike sekä sama kuva muokattuna niin, että se näyttää alaspainetulta kuvan 14 mukaisesti.



Kuva 14. Sulkupainike. Vasemmalla sulkupainike normaaliasennossa ja oikealla alaspainettuna.

Kuvan 14 mukaisesti sulkupainike alaspainettuna antaa käyttäjälle välitöntä palautetta hiiren painalluksen onnistumisesta. Kuva sijoitettiin ponnahtusikkunan oikeaan ylänurkkaan, koska se tuntuu käyttäjän mielestä luonnolliselta [7].

Sulkupainike pitää ohjelmoida niin, että käyttäjän painaessa sitä sen kuva muuttuu alaspainetuksi painikkeeksi, mutta ponnahtusikkuna ei vielä sulkeudu. Vasta kun käyttäjä irroittaa hiiren painikkeen sulkupainikkeen päällä, ponnahtusikkuna sulkeutuu. Tämä antaa käyttäjälle mahdollisuuden peruuttaa oma toimintansa, jos ei jostain syystä halukaan sulkea avoinna olevaa ikkuna. Näin ollen koodissa täytyy käyttää javascriptin *mouseDown*, *mouseUp* sekä *Click* –komentoja koodin 3 mukaisesti.

Koodi 3. Sulkupainikkeen kuvan vaihtaminen ja toiminnon suorittaminen.

```
Symbol.bindElementAction(compId, symbolName, "${_rasti}", "mousedown", function(sym, e)
{
    // kun nappulaa painetaan, sen kuva muutetaan alaspainetuksi
    $(this.lookupSelector("rasti")).attr("src","images/rasti_down.png");
});
Symbol.bindElementAction(compId, symbolName, "${_rasti}", "mouseup", function(sym,
e) {
    // kun painike irroitetaan nappulasta, muutetaan kuva ylöspainetuksi
    $(this.lookupSelector("rasti")).attr("src","images/rasti_up.png");
});
Symbol.bindElementAction(compId, symbolName, "${_rasti}", "click", function(sym,
e) {
    // kun ikkuna suljetaan, kutsutaan aikajanalta uusi animaatio
    // ikkunan sulkemiseksi. soitetaan animaatio, joka on nimetty
    // "popup_lopetus" nimiseksi
    sym.getComposition().getStage().play("popup_lopetus");
});
```

Koodissa 3 on kolme eri funktiota. Mikäli sulkupainike on koskematon, kuvana näytetään normaalia sulkupainikekuvaa. Mikäli hiiren painike on alaspainettuna, kuva vaihtuu alaspainetuksi. Mikäli sulkupainiketta klikataan, haetaan aikajanalta myöhemmin tehtävä animaatio, jossa ponnahtusikkuna suljetaan.

5.5 Rakennusten ponnahtusikkuna

Koska rakennusten (kuva 10) ponnahtusikkuna on samanlainen kuin navigointipalkin ponnahtusikkunan yhtä eroa lukuunottamatta, käsitellään tässä vain upotetun videon luomista.

Rakennusten ponnahtusikkunaan tulee navigointipalkin ponnahtusikkunaan verraten kolmen kuvan sijasta yksi video ikkunan oikeaan laitaan. Tätä varten tarvitaan paikka, josta videot haetaan. Turku Energia käyttää omissa videoissaan DreamBroker –ohjelmistoa, joten videot täytyy ladata kyseisen ohjelmiston upotettua videosoitinta käyttäen. Videoputus tehtiin seuraavanlaisella koodilla:

Koodi 4. Videoputuksen luominen.

```
Symbol.bindSymbolAction(compId, symbolName, "creationComplete", function(sym, e) {

    var videolinkki = sym.getParameter("videolinkki");
    youtubevid = $("<iframe/>");
    sym.$("vcontainer").append(youtubevid);
    youtubevid.attr('type', 'text/html');
    youtubevid.attr('width', '404'); //450
    youtubevid.attr('height', '304'); //350
    youtubevid.attr('src', 'http://www.dreambroker.fi/w/6f7075b92'); // url/Video_Id
    youtubevid.attr('frameborder', '1'); // 1 | 0
    youtubevid.attr('allowfullscreen', '1'); // 1 | 0
    youtubevid.attr('id', '6f7075b92'); });
```

Koodissa 4 videota varten luotiin oma *iframe*, jolla voidaan koota kokonaan eri sivuston komponentteja ja liittää ne nykyiseen sivustoon omana elementtinään. Elementille määriteltiin leveys- ja korkeusarvot pikseleinä, videon oikea sijainti URL-osoitteena sekä oma uniikki id-tunnus, jolla video voidaan tunnistaa. Lisäksi asetettiin kaksi lisäominaisuutta. Videon voi halutessaan asettaa kokoruututilaan, jotta sitä on helpompi seurata. Tämän lisäksi videon ympärillä näytetään pieni reunusta, jotta se erottuisi paremmin taustasta.

Videoputuksen myötä täytyy sulkupainikkeelle lisätä yksi rivi koodia, joka estää videon pyörimisen taustalla vaikka ponnahtusikkuna suljettaisiinkin. Helpoin tapa on ladata videon URL-osoite uudestaan, jolloin videon nykyinen

pyöriminen pysähtyy ja video siirtyy alkutilaansa odottamaan, että käyttäjä asettaa sen päälle.

5.6 Videoiden ja tekstien lisääminen

Ponnahdusikkunoista aukeavaa tekstiä ja otsikkoja varten täytyy koodiin lisätä muuttujat, jotka lisätään aiemmin tehtyihin tekstilaatikoihin. Samoin videolinkkien URL-osoitteet täytyy tallentaa ohjelmaan, jotta voidaan toistaa oikea video oikealla hetkellä. Tekstit on hyvä ladata jo ohjelman käynnistyessä, jotta ne eivät lisää latausaikaa kesken ohjelman käytön. Näin ollen ne sijoitettiin *compositionReady* solukkaan, jolloin ne latautuvat heti kun ohjelma käynnistetään. Koodissa 5 esimerkki navigointipalkin sekä Toimitalon muuttujien alustamisesta.

Koodi 5. Tekstien ja otsikoiden alustaminen.

```
Symbol.bindElementAction(compId, symbolName, "document", "compositionReady", function(sym, e) {

    // insert code to be run when the composition is fully loaded here

    sahkoverkko_boolean = 0; //älä muuta
    lampoverkko_boolean = 0; //älä muuta
    // #Muuttuja# Näitä asetuksia muuttamalla voit määritellä yläbannerin linkeistä ja "?"
    // kohdassa aukeavan laatikon tekstejä

    popup_info_kysymysmerkki_otsake = "Tervetuloa!";
    popup_info_yritys_otsake = "Turku Energia pähkinänkuoressa";
    popup_info_historia_otsake = "Historia";
    popup_info_ymparisto_otsake = "Ympäristö";
    popup_info_energiaala_otsake = "Energia-ala";

    popup_info_kysymysmerkki_kuvaus = "Tähän tulee tietoa siitä miten ohjelmaa käytetään.";

    popup_info_yritys_kuvaus = "Tähän tulee nippelitietoa Turku Energiasta.";
    popup_info_historia_kuvaus = "Tähän tulee turku energian historiaa.";
    popup_info_ymparisto_kuvaus = "Tähän tulee turku energian ympäristöasiat.";
    popup_info_energiaala_kuvaus = "Tässä kerrotaan turku energiasta energia-
alana/hyvinä työpaikkana.";

    // #Muuttuja# Näitä asetuksia muuttamalla voit määritellä otsikot

    toimitalo_otsikko = "Toimitalo";

    toimitalo_URL = "http://www.dreambroker.fi/w/6f7075b92";
    toimitalo_kuvaus = "<p>Lorem Ipsum.</p>";

});
```


Koodissa 5 on hyvä huomioida, että kuvauksia kirjoitettaessa pitää teksti sijoittaa HTML-tunnisteisiin, jotta se formatoidaan oikein. Esimerkiksi pelkkä enter-painikkeen painaminen ei tee tekstiin rivinvaihtoa, vaan se pitää suorittaa HTML-kielen tunnisteella `
`.

Tällä hetkellä tekstien vaihtaminen vaatii koodissa tehtävää muutosta, mutta tätä asiaa käsitellään tarkemmin opinnäytetyön Jatkokehitysmahdollisuudet-osiossa.

Tekstit piti vielä saada ponnahdusikkunassa vaihtumaan sen mukaan, mitä kohdetta käyttäjä on painanut. Näin ollen rakennuskuvien sekä navigointipalkin linkkeihin lisättiin kuhunkin erikseen koodi, joka vaihtaa tekstit sekä kuvat ja videon halutuksi kokonaisuudeksi koodin 6 mukaan.

Koodi 6. Tekstien, otsikoiden, videolinkin ja kuvien sijoittaminen ponnahdusikkunaan.

```
Symbol.bindElementAction(compId, symbolName, "${_toimitalo}", "click", function(sym, e)
{
// vaihtaa toimitalon popup-ikkunan otsikon halutunlaiseksi
sym.getSymbol("popup").$(".otsake").html(toimitalo_otsikko);

//vaihtaa toimitalon popup-ikkunan tekstiosion halutunlaiseksi
sym.getSymbol("popup").$(".kuvaus").html(toimitalo_kuvaus);

// vaihtaa toimitalon videolinkin oikeaksi
youtubevid.attr('src', toimitalo_URL);
});

Symbol.bindElementAction(compId, symbolName, "${_btn_historia}", "click", function(sym,
e) {

//otsakkeen ja kuvauksen vaihto oikeanlaiseksi

sym.getSymbol("popup_info").$(".info_otsake").html(popup_info_historia_otsake);
sym.getSymbol("popup_info").$(".info_kuvaus").html(popup_info_historia_kuvaus);

//kuvien vaihto oikeiksi
sym.getSymbol("popup_info").$(".kuvapaikka1").attr("src", "images/historia1.png");
sym.getSymbol("popup_info").$(".kuvapaikka2").attr("src", "images/historia2.png");
sym.getSymbol("popup_info").$(".kuvapaikka3").attr("src", "images/historia3.png");

});
```

Koodi 6 hakee kyseisen ponnahdusikkunan otsake- ja kuvaustekstikentät ja asettaa niiden arvoksi aiemmin luodut muuttujat. Lisäksi rakennuskuvissa vaihdetaan avattavan videon URL-osoite sekä navigointipalkin ponnahdusikkunasta vaihdetaan näytettävät kuvat oikeanlaiseksi.

5.7 Sähkö- ja lämpöverkon näyttäminen

Sähkö- ja lämpöverkkoa varten luotiin Photoshopilla kaksi kuvaa, jotka sulautuvat sovelluksen perusnäkympään. Kuvat voidaan aukaista tai piilottaa navigointipalkissa olevista painikkeista kuvan 15 mukaan.



Kuva 15. Turku Energia -sovelluksen sähkö- ja lämpöverkko taustakuvan päällä näytettynä.

Kuvasta 15 huomataan, että sekä sähkö- että lämpöverkkokuvat käyttävät valmiina olevia rakennuksia hyödykseen ja pyrkivät suurpiirteisesti kuvaamaan sitä ketjua, jonka sähkö ja lämpö käyvät ennenkuin ne pääsevät kotikäyttöön. Verkoissa on myös tekstiä auttamassa käyttäjää ymmärtämään, mistä on kyse. Lämpöverkon punainen viiva kuvastaa kaukolämpöä ja sininen viiva kaukojäähdytystä. Sähköverkossa oranssi viiva kuvaa sähkön kulkua, ja sen päällä olevat tekstit kertovat, missä eri vaiheissa sähkön jännitettä muutetaan ennen kotipistorasiaan päättymistä.

Kuvat ovat oletusarvoisesti piilotettuja. Navigointipalkissa on oikeassa laidassa kaksi painiketta, *Näytä Sähköverkko* ja *Näytä Lämpöverkko*. Näitä painikkeita klikkaamalla käyttäjä aukaisee kyseisen kuvan taustakuvan päälle. Käyttäjä voi

piilottaa kuvan klikkaamalla uudestaan samaa painiketta. Tätä varten Painikkeen nimi pitää muuttaa jokaisen klikkauksen jälkeen jo *Näytä Sähköverkko*-nimiseksi tai *Piilota Sähköverkko*-nimiseksi. Lisäksi koodissa pitää tietää, onko kuva jo näkyvillä. Tätä varten luotiin kokonaislukumuuttujat *lampoverkko_boolean* ja *sahkoverkko_boolean* ja niiden arvoksi asetettiin 0. Kuvien vaihtaminen ja tarkastelu kirjoitettiin *if-else* -koodirakenteella koodin 7 mukaan.

Koodi 7. Lämpö- ja sähköverkon näyttäminen ja piilottaminen.

```
Symbol.bindElementAction(compId, symbolName, "${_btn_lampoverkko}", "click", function(sym, e) {

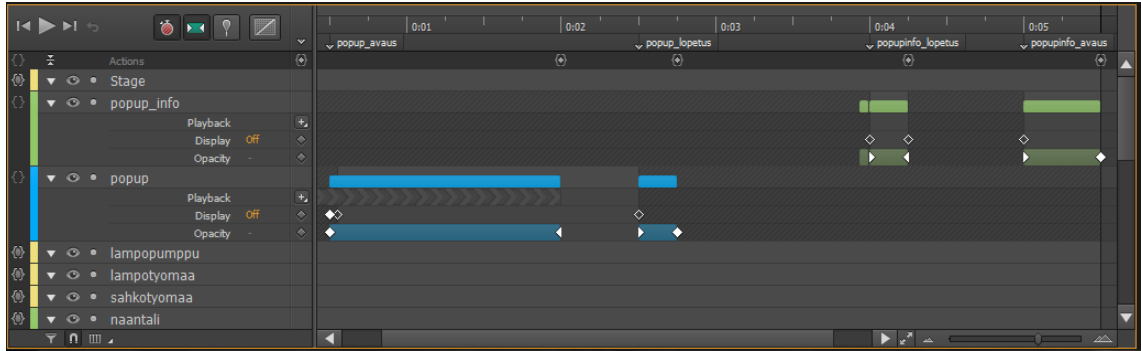
    if (lampoverkko_boolean == 0) {
        sym.$("btn_lampoverkko").html("Piilota Lämpöverkko");
        lampoverkko_boolean = 1;
        sym.$("lampoverkko").show();
    }
    else if (lampoverkko_boolean == 1) {
        sym.$("btn_lampoverkko").html("Näytä Lämpöverkko");
        lampoverkko_boolean = 0;
        sym.$("lampoverkko").hide();
    }
});
```

Koodi 7 tarkistaa hiiren painikkeen klikkauksen jälkeen onko kuva näkyvillä. Jos ei ole, se asettaa sen näkyville ja muuttaa painikkeen tekstiksi *Piilota Lämpöverkko*. Lisäksi se muuttaa booleanmuuttujan arvoon 1, jotta ohjelma tietää kuvan olevan esillä.

Mikäli kuva oli jo esillä painiketta klikattaessa, ohjelma muuttaa tekstin nimeksi *Näytä Lämpöverkko*, muuttaa booleanmuuttujan arvoon 0 sekä piilottaa kuvan näkyvistä.

5.8 Animaatioiden luominen

Sovellusta varten luotiin 4 animaatiota: Molempien ponnausikkunoiden avaus- ja sulkemisanimaatiot. Tämä toteutettiin lisäämällä aikajanaan neljään eri kohtaan merkit, jotka nimettiin nimillä *popup_avaus*, *popup_lopetus*, *popupinfo_avaus* ja *popupinfo_lopetus*. Näitä merkkejä voidaan koodissa kutsua, kun halutaan soittaa tietty animaatio esimerkiksi ponnausikkunan sulkupainiketta painaessa (kuva 16).



Kuva 16. Adobe Edge Animaten aikajana.

Kuvasta 16 huomataan sen ylälaidalla neljä merkkiä, joita kutsutaan koodissa, kun sitä vastaava toimenpide halutaan suorittaa. Kuvan pisin sininen viiva kuvaa rakennusten ponnahtusikkunan avausanimaatiota. Kun käyttäjä painaa hiiren painikkeella rakennusta, kutsuu koodi animaatiota alkamaan kohdasta *popup_avaus* koodilla `sym.play("popup_avaus");`. Näin ollen sovellus alkaa suorittamaan toimintoja alkaen kohdasta *popup_avaus* ja jatkaa niin kauan kunnes ohjelmalle muuta kerrotaan. Ponnahtusikkuna on oletusarvoisesti piilotettu, jotta se ei ole turhaan peittämässä näkyvyyttä. Kun animaatio käynnistyy, se poistaa kohteen läpinäkyvyyttä noin 2 s:n ajan. Tämän jälkeen sovellus tulee aikajanelle kohtaan, johon on kirjoitettu koodi `sym.stop();`. Tämä pysäyttää aikajanan liikkumisen eteenpäin, joten sovellus jättää ponnahtusikkunan päälle, kunnes se käyttäjän toimesta suljetaan.

Toisen ponnahtusikkunan avaaminen toimii samalla periaatteella ja näkyy kuvan 16 oikeassa laidassa. Kuvan 16 kaksi lyhyempää animaatioviivaa kuvaavat ponnahtusikkunoiden sulkemista ja ovat toimintaperiaatteiltaan samanlaisia kuin avustoimenpide. Tällä kertaa kohteisiin kuitenkin lisätään läpinäkyvyyttä, jotta se kuvastaisi realistisesti ponnahtusikkunan sulkemista.

6 JATKOKEHITYSMAHDOLLISUUDET

Vaikka sovellus on nykyisellään käyttökelpoinen, ei siihen ehditty kuitenkaan aikamääreiden puitteissa asettaa kaikkia haluttuja ominaisuuksia. Näitä ominaisuuksia lisäämällä sovelluksesta saadaan helppokäyttöisempi, päivitettävämpi sekä selkeämmän näköinen.

Alla esiteltyt kehitykset ovat kuitenkin sovelluksen eliniän kannalta tarpeellisia.

6.1.1 Vierityspalkki

Tällä hetkellä ponnahdusikkunan tekstilaatikon koko on rajallinen, eikä siihen välttämättä mahdu koko haluttu teksti. Tämä voi aiheuttaa kaksi epähaluttua tilannetta. Toinen niistä on linkin lisääminen, josta selviää lisää informaatiota kyseisestä aiheesta. Tämä kuitenkin pakottaa käyttäjän pois sovelluksesta, joten se ei ole suositeltava vaihtoehto.

Toinen vaihtoehto on tekstin karsiminen vain ydinaiheisiin. Tämä vaihtoehto on lähestulkoon pakollinen nykytilanteessa, mutta rajaa käyttäjälle annettavaa informaatiota. Näin ollen käyttäjä saa informaatiota haluamastaan asiasta, mutta mahdollisesti liian vähän.

Korjaus näihin ongelmiin on vierityspalkin lisääminen tekstikenttään. Tämä mahdollistaa pidempien tekstien ja kokonaisten kappaleiden käytön sovelluksessa. Vierityspalkki ei muuta sovelluksen ulkoasua eikä vaadi ylimääräisiä käyttöliittymämuutoksia. Sitä ei kuitenkaan ehditty ohjelmoimaan sovellukseen aikamääreiden puitteissa.

6.1.2 Sovelluksen hallinta- ja muokkausohjelma

Yksi sovellusten käyttöikä nostava ominaisuus on sen muokattavuus. Mikäli sovellusta ei voi muokata tai mikäli siihen ei voida lisätä tai siitä voidaan poistaa ominaisuuksia on sen käyttö hyvin rajattua.

Turku Energia- sovelluksessa kuvien vaihtaminen on äärettömän helppoa. Jos halutaan vaihtaa tietty kuva joko kartalla tai ponnahtusikkunalla, tarvitsee vaihdettavan kuvan tilalle vaihtaa uusi kuva *img*- kansioon sillä nimellä, jolla vaihdettava kuva on. Koska sovelluksen koodissa kuva haetaan sen nimellä, ei koodiin tarvitse koskea ollenkaan.

Tekstien, otsikoiden sekä videolinkkien kohdalla tilanne on eri. Sovellusta suunniteltaessa tarkoituksena oli luoda erillinen tekstitiedosto, josta sovellus hakee tarvittavan tekstin. Näin ollen voitaisiin tekstejä muokata tekstitiedostosta käsin, eikä sovelluksen koodiin tarvitsisi mennä. Javascript ei kuitenkaan suoraan salli tekstitiedostojen käsittelyä tietoturvan vuoksi. tekstitiedostojen käsittely on mahdollista muilla keinoilla, mutta sitä ei saatu toimimaan aikamääreiden puitteissa.

Jatkokehityksessä järkevintä olisi luoda oma hallintaohjelma, jossa on oma käyttöliittymä tekstien, kuvien, otsikoiden ja videolinkkien muokkaukseen. Tämä voitaisiin toteuttaa luomalla internetsivusto, jonne vaaditaan käyttäjätunnus ja salasana. Tämän käyttöliittymän alle voitaisiin lisätä kaikki tekstimuuttajat, joita voidaan muokata. Näin ollen ei tarvitsisi koskea edes tekstitiedostoihin, vaan kaikki voitaisiin hallita käyttöliittymällä. Sivustolla voitaisiin jopa hallita kuvia, mikäli sinne lisättäisiin kuvien lähetysmahdollisuus.

Tämänkaltaisen moduulin luominen vaatii kuitenkin niin paljon aikaa, ettei sitä ehditty aikamääreiden puitteissa valmistaa.

6.1.3 Kokoruutuun siirtyminen

Sovellus on tällä hetkellä suunniteltu tilastojen mukaan, jossa suurin osa käyttäjistä omistaa näytön, joka pystyy toistamaan 1366x768 pikseliä. Tämä vastaa noin 15,6 tuuman kannettavaa tietokonetta. Sovelluksen nykyinen koko on 1022x750 pikseliä. Tämä tarkoittaa sitä, että kun selaimen omat vierityspalkit ja muut palkit lasketaan pois, sovellus on noin koko ruudun kokoinen. Tämä toimii hyvin kaikilla niillä käyttäjillä, joilla on vähintään haluttu tietokoneen ruudun koko.

Parempi vaihtoehto olisi kuitenkin ollut skaalautuvat vektoripohjaiset kuvat, jonka ansiosta sovellus voisi oletusarvoltaan olla pienempi, mutta sen saisi halutessaan kokoruudun kokoiseksi. Mikäli kuvat olisivat vektorikuvia, voitaisiin niiden kokoa muuttaa menettämättä kuitenkaan kuvanlaatua. Tämä antaisi lisää mahdollisuuksia sovelluksen käyttöön tulevaisuudessa.

7 YHTEENVETO

Tämän opinnäytetyön tarkoituksena oli luoda Turku Energia Oy:lle interaktiivinen sovellus, jota voidaan käyttää niin esittelymateriaalina kuin toimia omana sovelluksenaan messuilla tai koululaisille. Toteutukseen valittiin Adobe Edge Animate -sovelluskehitysohjelma, joka takasi korkeimman potentiaalisen käyttäjämäärän sen natiivin selaintoimivuuden vuoksi.

Ohjelman avulla luotiin sovellus, joka esittää Turku Energian tärkeimmät rakennukset, tietoa sähkö- sekä lämpöverkon tuotannosta ja siirrosta sekä informaatiota energia-alasta sekä Turku Energian mahdollisista uravaihtoehtoista. Nämä asiat sisällytettiin sovellukseen tekstin, videon sekä 3D-mallinnettujen rakennusten avulla. Lisäksi sovellukseen sisällytettiin navigointipalkki, joka sisältää tekstipainikkeita. Näihin painikkeisiin sisällytettiin informaatiota, jota ei muuten karttanäkymässä olisi voitu esittää.

Rakennusten sekä navigointipalkkien toteutukseen tarvittiin Adobe Photoshop- sekä Autodesk 3ds max -ohjelmia. Lisäksi käytettiin Turku Energian omistamaa kuvamateriaalia täydentämään sähkö- ja lämpöverkkojen visuaalista antia.

Koko sovellus kirjoitettiin Javascript- ja HTML5 ohjelmointikielillä käyttäen apuna jQuery-kirjastoa.

Ennen sovelluksen luomista piti kuitenkin miettiä mahdollista käyttäjäryhmää sekä suunnitella käyttöliittymä niin, että se on sekä käytettävä että helppolukuinenkin. Koska sovellus on pääasiassa tarkoitettu yläasteikäisille lapsille sekä kaikille yritysmessuvieraille, on mahdollinen käyttäjäkunta niin laaja, että sovellus piti suunnitella sopimaan kaikille käyttäjille. Ratkaisussa päädyttiin luomaan vuokaavioita sovelluksen koon määrittämiseksi ja käyttöliittymän toteutuksen helpottamiseksi. Ratkaisun todettiin olevan hyvä ja erittäin käytännöllinen, mutta työtä helpottamiseksi oltaisiin voitu käyttää myös muita tekniikoita käyttöliittymän parantamiseksi. Sovellukselle voidaan myöhemmin toteuttaa vielä kunnollinen käyttäjätestaus ennen sen julkistamista. Tämä

tarkoittaa noin 5 henkilön haastattelemista ja tiedon keräämistä sovelluksen käytettävyydestä samalla, kun kohdehenkilöt käyttävät sovellusta. Koska sovellusta on jo testattu kehittäjän ja siinä mukana olevien ohjaajien toimesta, olisi suotavaa, että käyttäjätestaukseen kuuluvat henkilöt olisivat kokonaan projektin ulkopuolisia henkilöitä. Näin saataisiin mahdollisimman tarkka kuva siitä, miltä sovellus tuntuu projektin ulkopuoliselle henkilölle.

Sovelluksen jatkokehityksestä on kuitenkin sovittu, sillä aikataulullisista syistä sovellukseen tulevia tekstejä ei ole ehditty laatia eikä videoita voitu kuvata siihen varatusta ajasta huolimatta. Tämä johtui siitä, että projektiin osallistuneiden henkilöiden kanssa ei saatu työajan puitteissa lomista ja ilmastollisista syistä sovittua videointiaikoja.

Lisäksi jatkokehitystä ajatellen sovelluksesta voidaan tehdä ruutukokoon skaalautuva, jotta sitä voidaan käyttää minkäkokoisella näytöllä tahansa. Sovelluksen tämänhetkiset kuvat eivät kuitenkaan ole vektorigrafiikkaa ja näin ollen menettäisivät kuvatarkkuuttaan isommaksi skaalattaessa. Muita jatkokehitykseen liittyviä toimenpiteitä ovat hallintamoduulin luominen sekä vierityspalkin luominen tekstilaatikoihin. Nämä tuovat sovellukselle lisää käyttöikää nykyiseen verrattuna.

LÄHTEET

- [1] Adobe. "Components learning guide for flash". [www-dokumentti]. Saatavilla: http://www.adobe.com/devnet/flash/learning_guide/components.html (luettu 13.4.2013)
- [2] Unity. "Unity scripting". [www-dokumentti] Saatavilla: <http://unity3d.com/unity/workflow/scripting> (luettu 27.4.2013)
- [3] Unity. "Reference manual". [www-dokumentti] Saatavilla: <http://docs.unity3d.com/Documentation/Components/index.html> (luettu 22.4.2013)
- [4] Adobe Edge Animate: The Missing Manual, Chris Grover. ISBN10:1-4493-4225-6. Viitattu 20.4.2013
- [5] www-dokumentti. viitattu 28.4.2013. Saatavilla: http://www.gamasutra.com/view/news/191112/Unity_drops_Flash_support_says_Adobe_is_not_firmly_committed.php
- [6] Automatically Generating Personalized User Interfaces, Krzysztof Z. Gajos. Viitattu 25.4.2013 Saatavilla: <http://www.eecs.harvard.edu/~kgajos/papers/2008/kgajos-dissertation.pdf>
- [7] Designing the User Interface: Strategies for Effective Human-Computer Interaction, B. Shneiderman. ISBN-10: 0321537351. Viitattu 25.4.2013
- [8] "Alpha, Beta and Pilot testing". [www-dokumentti] Saatavilla: <http://www4.nau.edu/azregions/Testing/beta.htm> (luettu 25.4.2013).
- [9] Clayton Lewis & John Rieman, "Task centered User Interface Design: A Practical Introduction". [www-dokumentti] Saatavilla: http://grouplab.cpsc.ucalgary.ca/saul/hci_topics/tcsd-book/chap-1_v-1.html (luettu 28.4.2013).
- [10] Turku Energia Oy. "Energia syntyy yhteistyöstä". [www-dokumentti] Saatavilla: <http://www.turkuenergia.fi/index.php?page=8a15ff40a98fb4587578c4bc2f4b5> (luettu 28.4.2013).
- [11] Info Graphics mania. "HTML5 browser support cheat sheet". [www-dokumentti] Saatavilla: <http://infographicsmania.com/html5-browser-support-cheat-sheet/> (luettu 6.5.2013).
- [12] Helsingin Energia. "Energiavisa". [www-dokumentti] Saatavilla: <http://www.helen.fi/energiansaasto/energiavisa.html> (luettu 6.5.2013).
- [13] YIT. "Together". [www-dokumentti] Saatavilla: http://www.yit.fi/yit_fi/uudelleenohjaukset/together (luettu 7.5.2013).
- [14] Gamasutra. "Adobe Vs. Unity: The future of 3D web games". [www-dokumentti] Saatavilla: http://www.gamasutra.com/view/feature/134625/adobe_vs_unity_the_future_of_3d_.php?print=1 (luettu 7.5.2013).
- [15] Guide to Advanced Software Testing, Anne Mette Johanssen Hass, eISBN: 9781596932869. Viitattu 24.5.2013
- [16] Game Testing All in One, Boespflug, Koenig Schultz, Charles P. eISBN: 9781592003730 . Viitattu 25.5.2013

[17] "What Problems Do People With Disabilities Have? And Why?" [www-dokumentti] Saatavilla: http://trace.wisc.edu/docs/software_guidelines/software.pcs/disabil.htm (luettu 26.5.2013)

[18] Florida Power and Light, "Interactive House". [www-dokumentti] Saatavilla: http://fpl.com/residential/energy_saving/resources_tips/house.shtml# (luettu 26.5.2013)