



KÄYTTÖLIITTYMÄN TOTEUTUS FLASH LITE SOVELLUKSESSA

**Case: Painokuntoon kännykän
opastuksella**

Esa Saksola

**Opinnäytetyö
Toukokuu 2009**

Kulttuuriala



**JYVÄSKYLÄN
AMMATTIKORKEAKOULU**

| | | |
|---|---|--------------------------|
| Tekijä(t) SAKSOLA, Esa | Julkaisun laji Opinnäytetyö | |
| | Sivumäärä 40 <input type="checkbox"/> | Julkaisun kieli Suomi |
| | Luottamuksellisuus Salainen _____ saakka | |
| Työn nimi KÄYTTÖLIITTYMÄSUUNNITTELU FLASH LITE -SOVELLUKSESSA Case: Painokuntoon kännykän opastuksella | | |
| Koulutusohjelma Viestinnän koulutusohjelma | | |
| Työn ohjaaja(t) Martti Heikkinen | | |
| Toimeksiantaja(t) Jyväskylän ammattiopisto | | |
| <p>Tiivistelmä</p> <p>Opinnäytetyössä tehtiin toimeksiantona kännykässä toimiva ohjeistus GTO 52 – painokoneen painokuntoonlaitosta. Sovellus oli Flash Lite-sovellus, joka tehtiin Adode Flash CS3-ohjelmalla. Työssä tutkittiin, mitä vaatimuksia ja rajoituksia puhelin, Flash Lite sekä käyttäjäryhmä asettavat käyttöliittymän toteuttamiselle.</p> <p>Flash Lite on Symbian 60-pohjaisille kännyköille optimoitu versio Adoben Flash Playeristä. Flash Lite -sovelluksen toteuttaminen ei varsinaisesti eroa tavallisen Flash-sovelluksen tekemisestä, mutta tekijän täytyy huomioida yhteensopivuus ActionScript-koodin ja Flash Liten versioiden kanssa saadakseen sovellus toimimaan haluamassaan puhelinmerkissä ja –mallissa.</p> <p>Käyttöliittymä ei ole kääre, johon sovellus kääritään. Käyttöliittymä suunnitellaan käyttäjää, ei sovellusta varten. Suunnittelussa tulee huomioida käyttäjän tarpeet. Käyttöliittymäsuunnittelulla pyritään hyvään käytettävyyteen. Eräitä käytettävyyden määreitä ovat ymmärrettävyys, vaivattomuus, kattavuus ja esteettinen miellyttävyys. Näihin päämääriin päästään noudattamalla heuristisia sääntöjä sommittelun, navigoinnin ja luettavuuden suhteen.</p> <p>Flash Liten soveltamismahdollisuuksia on tutkittu vain vähän Jyväskylän ammattikorkeakoulussa, vaikka käyttöliittymäsuunnittelua ja Adobe Flashia voidaan opiskella siellä.</p> | | |
| Avainsanat (asiasanat) Käyttöliittymäsuunnittelu, käytettävyys, Adobe Flash, Flash Lite, ActionScript | | |
| Muut tiedot Mukana CD-ROM, jolta löytyy Flash Lite-sovellus ja siihen liittyvät videotiedostot. | | |

| | | |
|---|--|---------------------|
| Author(s) SAKSOLA, Esa | Type of Publication Bachelor's Thesis | |
| | Pages 40 | Language Finnish |
| | Confidential <input type="checkbox"/> Until _____ | |
| Title USER INTERFACE DESIGN FOR FLASH LITE APPLICATION Case: Ready to press guide in mobile phone | | |
| Degree Programme Degree Programme in Media Design | | |
| Tutor(s) Martti Heikkinen | | |
| Assigned by Jyväskylä Educational Consortium | | |
| Abstract <p>The purpose of the thesis was to make a guide for how to prepare the GTO 52 offset press for printing. The guide was a Flash Lite application for the mobile phone made with Adobe Flash. The aim of the study was to find what restrictions and limitations the target group, the choice of platform and software would place on designing the user interface.</p> <p>Flash Lite is mobile phone optimized version of Flash Player. It includes software for Symbian 60 based mobile phones. The process of application composition does not significantly differ from that of a regular Flash application. One has to keep in mind the compatibility of the levels of ActionScript with the different versions of Flash Lite to ensure the functionality of the application in various models of mobile phones.</p> <p>The user interface is not a wrapping for the application. The interface is designed for the user, not for the application. Designing the interface should be based on the needs of the user. The aim of user interface design is good usability. Usability can be described with adjectives like understandable, effortless, comprehensive and aesthetically pleasant. These objectives can be reached by applying heuristic rules to the layout, navigation and readability.</p> <p>The potential of Flash Lite applications is very little studied at JAMK, even though courses for both Adobe Flash and user interface design are available there.</p> | | |
| Keywords User interface design, usability, Adobe Flash, Flash Lite ActionScript | | |
| Miscellaneous Includes a CD-ROM with Flash Lite application and related video files. | | |

SISÄLTÖ

| | | |
|----------|--|----------|
| 1 | JOHDANTO..... | 4 |
| 1.1 | Työn kuvaus..... | 4 |
| 1.2 | Toimeksiantajan kuvaus | 4 |
| 1.2 | Opinnäytetyön rajaus ja tavoitteet | 5 |
| 2 | MATKAPUHELIN JULKAUSUALUSTANA | 6 |
| 2.1 | Näyttö | 6 |
| 2.2 | Näppäimistö | 6 |
| 2.3 | Muisti..... | 7 |
| 2.4 | Käyttöjärjestelmä | 7 |
| 3 | ADOBE FLASH | 8 |
| 3.1 | Historia | 8 |
| 3.2 | Sovelluksen kuvaus..... | 8 |

| | |
|---|-----------|
| 3.3 Työpohjat | 9 |
| 3.4 Symbolit..... | 10 |
| 3.5 ActionScript | 11 |
| 3.6 Flash Lite..... | 12 |
| | |
| 4 KÄYTETTÄVYYS | 13 |
| | |
| 4.1 Määritelmiä..... | 13 |
| 4.2 Käyttöliittymän suunnittelu | 14 |
| 4.2.1 Käsikirjoitus | 14 |
| 4.2.2 Käyttäjärhmä suunnittelun lähtökohtana..... | 16 |
| 4.2.3 Typografia | 17 |
| 4.2.4 Navigointi | 18 |
| 4.2.5 Ruutusuunnittelu | 19 |
| | |
| 5 KÄYTTÖLIITTYMÄN TOTEUTUS | 21 |
| | |
| 5.1 Käsikirjoitus..... | 21 |
| 5.2 Videon käyttö..... | 21 |
| 5.2.1 Videokamerasta kännykän näytölle..... | 21 |
| 5.2.2 Videon tuonti sovellukseen | 22 |

| | |
|---|-----------|
| 5.3 Ruutusommittelu | 23 |
| 5.4 Navigointi | 26 |
| 5.5 Painikkeet | 27 |
| 5.6 ActionScript koodaus | 28 |
| | |
| 6 POHDINTA..... | 30 |
| | |
| LÄHTEET | 33 |
| | |
| LIITTEET | 35 |
| | |
| Liite 1. Sovelluksen käsikirjoitus | 35 |

1 JOHDANTO

1.1 Työn kuvaus

Idea matkapuhelimessa toimivasta ohjeistuksesta on lähtöisin jo kolmen vuoden takaisesta työharjoittelusta. Tuolloin keskusteltiin vain yksittäisestä GTO 52 -painokoneen säätökohdasta ja sen havainnollistamisesta. Myöhemmässä vaiheessa, kun keskusteltiin, millaiselle sovellukselle todellisuudessa olisi käyttöä, ketkä sitä käyttäisivät ja mihin tarkoitukseen, päätettiin toteuttaa kokonainen GTO 52 -painokoneen painokuntoon laitto. Painokuntoon laitto on koko painotuotantoprosessin ensimmäinen ja yksi monista vaiheista, joka painotuotantoassistentin tulee hallita.

Flash Lite -sovellus toteutettiin sähköisen muistilistan muodossa. Suunnitteluvaiheessa painotuotannon opetushenkilöstön kanssa totesimme, että varsinaisen opetusmateriaalin toteuttaminen ei käyttötarkoitusta huomioiden, olisi järkevää. Sen sijaan itsenäiseen käyttöön tarkoitettu tukimateriaali, jonka käyttö ei olisi aikaan tai paikkaan sidottua ja joka liikkuisi opiskelijan matkapuhelimen mukana, olisi jopa tarpeellinen.

Sovellus toteutettiin Adobe Flash CS3 -ohjelmalla. Jotta sovellus toimisi matkapuhelimessa, se tallennettiin Flash Lite 1.1 muotoon, jota tukevat kaikki Symbian Series S60 -pohjaiset matkapuhelimet, joita valmistaa mm. Nokia ja Sony-Ericson. (Manninen & Marttila 2006, 448.)

1.2 Toimeksiantajan kuvaus

Jyväskylän ammattiopistossa opiskelee vuosittaan n. 4000 opiskelijaa eri aloilla. Koulutusta toteutetaan 25 eri ammattitutkinnossa kuudessa eri toimipisteessä. Painoviestintä kuuluu Tekniikan ja liikenteen koulutusalaan ja tutkintonimike on media-assistentti. Painoviestintä jakaantuu painotekniikan sekä ulkoasuntoteuttajan koulutusohjelmiin. Painotekniikan koulutusohjelmasta valmistuvien painotuotantoassis-

tenttien työpaikkoja ovat mm. kirja-, sanomalehti-, aikakauslehti- ja digipainot sekä sitomot. (Hakijan opas 09, 2008.)

Painoviestintäosaston laitekantaan kuuluu sekä digitaalisia että perinteisiä offset-painokoneita. Digitaalisia koneita on kolme kappaletta. Suurin, Xeikon on rotaatiokone, jolla painettavien töiden maksimileveys on 48 cm. Työn pituutta ei painokoneen ominaisuuksien vuoksi ole rajoitettu. Kooksi muuta konetta ovat mustavalko- sekä värikone, joiden maksimi arkkikoko on B3.

Offset koneita on yhteensä viisi kappaletta. Heidelberg GTO 52 –koneita, joita varten sovellus on tehty, on neljä kappaletta, joista kaksi on yksiväri- ja kaksi kaksivärikoneita. Näissä koneissa maksimi arkkikoko on B3. Suurin koneista on Heidelberg Speedmaster 62. Kone on kaksivärinen ja sen maksimi arkkikoko on B2.

1.3 Opinnäytetyön rajaus ja tavoitteet

Flash Litea on tarkasteltu ainakin JAMK:ssa ohjelmointimahdollisuuksien näkökulmasta (Salmi & Tokkari, 2005) ja digitaalisen dokumentaation näkökulmasta (Pasanen, 2006). Käyttöliittymään ja sen suunnitteluun, toteutukseen tai uudistukseen liittyviä opinnäytetöitä on tehty useita. Työn käytännön osuudessa tehtyä Flash Lite-sovellusta lähestyttiin käyttöliittymäsuunnittelun näkökulmasta. Nämä kaksi asiaa yhdistettynä antavat uudenlaisen lähestymissuunnan. Lisäksi puhelimien ominaisuuksien kehitys on edennyt huikeasti.

Tutkimus toteutettiin case-luontoisesti Flash Lite -sovelluksen kautta. Tavoitteena oli tutkia, millaisia rajoituksia ja vaatimuksia julkaisualusta, sovellus, käyttäjä, käyttötarkoitus ja -paikka asettavat käyttöliittymän toteutuksen suhteen. Tavoitteena oli myös käytettävyydeltään hyvä sovellus, jolloin oli tarpeen etsiä ja miettiä, millaisia teknisiä tai visuaalisia ratkaisuja se vaatisi.

Yksi suurimmista haasteista pöytäkoneelle suunniteltaviin sovelluksiin verrattuna on puhelimen näytön koko. Pienelle näytölle ei saada mahtumaan paljoa informaatiota. Koon lisäksi käytettävyyteen vaikuttaa eri puhelinmallien näppäinten sijoittelu. Ra-

joittavia tekijöitä ovat myös puhelimessa käytettävän muistin suuruus sekä prosessoriteho. (Manninen & Marttila 2006, 449.)

2 MATKAPUHELIN JULKAISUALUSTANA

2.1 Näyttö

Ensimmäiset värilliset näytöt tulivat matkapuhelimiin vuonna 2001, jolloin ne alkoivat nopeasti syrjäyttää perinteisiä mustavalkonäyttöjä. Nämä STN-LCD -näytöt pystyivät toistamaan enintään 256 eri väriä. Vuotta myöhemmin ilmestyi TFT-LCD -näytöt, joiden värintoistokyky ylsi jo 260 000 eri sävyyn, jolloin tämä näyttötyyppi alkoi vallata markkinoita myös muiden laitteiden, kuten videokameroiden ja digikameroiden näyttönä (Byun ja muut 2003). Nykisin yleisesti puhelimissa käytetyt TFT-LCD -näytöt ovat kehittyneempiä ja kykenevät jopa 16 miljoonaa eri värisävyä (Nokia, 2009).

Näyttöjen koot vaihtelevat jonkin verran puhelimen valmistajasta ja mallista riippuen. Tyypillisiä näyttökokoja ovat 176x208, 320x208 ja 320x240 pikseliä (Manninen & Marttila 2006, 449). Nykyaikaisten Symbian s60 -matkapuhelimien, joita valmistavat mm. Nokia ja Sony-Ericson, kooksi näyttää vakiintuneen 320x240 pikseliä. Näytön lävistäjä saattaa tosin vaihdella 2:sta 2,8 tuumaan. Näytön mittasuhte on 4:3, joka on sama kuin kännykässä olevan kameran tai videokameran tuottaman kuvan suhde. Sama kuvasuhde on myös perinteisissä kuvaputkitelevisioissa sekä useimmissa tietokoneen monitoreissa, sekä kuvaputkellisissa että litteissä näyttöissä.

2.2 Näppäimistö

Kaikissa Symbian -alustalla toimivissa matkapuhelimeissa on samanlainen näppäimistön asettelu ja järjestys. Numeronäppäimet juoksevat kolmessa sarakkeessa ja neljässä rivissä yhdestä yhdeksään, ja nolla on alimman rivin keskimäinen painike. Niissä on neljään suuntaan toimiva ristiohjain, jonka keskellä on valintapainike. Useimmiten ristiohjaimen molemmilla puolilla on niin kutsutut softkey-painikkeet,

joiden toiminta muuttuu puhelimesta käytettävien ohjelmien mukaan. Näille näppäimelle voi käyttäjä määrittää myös omia haluamiaan toimintoja.

Antti Wiion (2004, 170-71) näkemyksiä perinteisistä käyttöliittymän ratkaisumalleista voidaan soveltaa myös puhelimen näppäinten sijoitteluun. Hänen mukaansa totuista malleista poikkeaminen aiheuttaa ei-toivottuja tilanteita. Koska voidaan olettaa, että puhelimen käyttö kuuluu jokaisen ihmisen perustaitoihin, käyttäjän ei pitäisi tarvita opetella uudestaan painikkeiden sijainteja ja toimintoja vaihdettaessa puhelinta toiseen.

2.3 Muisti

Vertailtaessa eri valmistajien tuotteita voidaan todeta, että nykyaikaisissa matkapuhelimeissa muistikapasiteetti on kasvanut nopeasti viime vuosien aikana. Tällä hetkellä yleisimpien puhelinten sisäinen muistikapasiteetti vaihtelee yleensä 10 megatavun ja 10 gigatavun välillä puhelimen valmistajasta tai mallista riippuen.

Monessa puhelimesta on tarjolla laajennusmahdollisuus muistikorttien avulla. Nokian (Nokia, 2009) ja Samsungin (Samsung, 2009) malleissa, joissa tämä mahdollisuus on, se tapahtuu MicroSD-kortilla, jonka koko vaihtelee 2–16 gigatavuun. Sony-Ericsonilla (Sony-Ericson, 2009) memorystickMicrolla, jonka koko on 1–16 gigatavua.

2.4 Käyttöjärjestelmä

Yhteiseksi käyttöjärjestelmästandardiksi kehitetty Symbian OS on pohjana koko kommunikaatioteollisuudelle. Järjestelmä on avoin ympäristö, jolle kuka tahansa ohjelmisto- tai laitevalmistaja voi kehittää omia sovelluksia. Symbian käyttöjärjestelmällä toimivat nykyiset Fujitsu, LG, Motorola, Nokia, Samsung, Sharp ja Sony-Ericson. (Symbian Ltd, 2009.)

Tarpeet yhteiselle käyttöjärjestelmälle olivat ilmeiset. Yritysten kustannukset olivat pienempiä, kun kaikki kehittivät yhteistä alustaa, jolloin kehitystyöhön uhrattavat

resurssit pienenivät. Samalla riskit pienenivät, kun kukaan ei ollut kehittämässä toisten kanssa kilpailevaa järjestelmää. Sovelluksia kehitettäessä samaa pohjaa voitiin käyttää usealle eri valmistajan laitteelle. Laitekohtaiset muutokset sovellukselle olivat graafisen käyttöliittymän sovittaminen kulloistenkin hallintalaitteiden, kuten näppäinten tai mahdollisen kosketusnäytön sijaan. (Symbian Ltd 2009.)

3 ADOBE FLASH

3.1 Historia

Alkuperäisen Flash-formaatin kehitti Macromedia vuonna 1996, jolloin julkaistiin Flash 1.0 -versio. Vuonna 2002 julkaistiin seuraava merkittävä päivitys, Flash 5.0, joka oli edeltäjiään vakaampi versio. Sen ulkoasua oli lisäksi yhtenäistetty muiden Macromedian ohjelmien mukaisesti. 2004 julkaistiin Flash MX2004, jota seurasi vuonna 2005 julkaistu Flash 8. Samoihin aikoihin Adobe kiinnostui ohjelmasta ja myöhemmin osti sen itselleen. Vuonna 2007 julkaistiin Adobe Flash osana Adoben CS3 ohjelmistopakettia. (Paananen 2008, 7.)

3.2 Sovelluksen kuvaus

Paananen (2008, 7) kuvailee Flashin olevan maailman käytetyin multimediaformaatti. Flash -elokuva voi sisältää tekstiä, ääntä, animaatiota ja videokuvaa, pelkästään yhtä edellä mainituista tai kaikkia niitä yhdessä. Elokuva voi olla yksi kronologisesti etenevä esitys. Esitys voi olla rakennettu myös siten, että käyttäjä voi navigoida siinä kohdasta toiseen haluamassaan järjestyksessä. Flash -elokuva voidaan rakentaa myös pelin muotoon. Tällä hetkellä yksi tunnetuimpia Flashiä hyödyntäviä www-sivuja on YouTube. Muita käyttäjä Flashille on www-sivuilla näkyvien mainosten toteuttaminen, käyttöliittymien toteuttaminen sovelluksiin sekä vastaavanlaiset presentaatiot, joita esimerkiksi Microsoftin Power Point -ohjelmalla toteutetaan.

Stage

Stage on suomeksi uomeksi lava tai näyttämö on se alue, jolle Flash-elokuva kooste-

taan. Objekteja voidaan sijoittaa stagen ulkopuolelle, mutta elokuvasta näkyvä osuus on ainoastaan se, mitä dokumentin asetuksiin on määritelty.

Timeline

Paananen (2008, 14) kertoo timelinen eli aikajanan olevan elementti, joka tuo Flashiin uuden ulottuvuuden tavallisiin grafiikkaohjelmiin verrattuna. Sen avulla voidaan elokuvaan luoda halutulla tavalla etenevä sisältö. Flash-elokuva voi koostua yhdestä ainoasta aikajanasta. Se voi kuitenkin yhden aikajanan sijaan sisältää useita aikajanoja, jotka erotetaan toisistaan sceneiksi, eli kohtauksiksi. Normaalissa toistossa elokuva siirtyy edellisen kohtauksen lopusta seuraavana järjestyksessä olevan kohtauksen alkuun. Flash-dokumentin tekijä voi itse vaikuttaa siihen, missä järjestyksessä kohtaukset esitetään. Tekijä voi myös antaa katsojalle mahdollisuuden navigoida elokuvassa ja katsoa kohtaukset läpi halutussa järjestyksessä.

Frame ja keyframe

Manninen ja Marttila (2006, 16) kertovat aikajanan sisältävän kahdenlaisia ruutuja. Keyframeksi eli avainruuduksi kutsutaan ruutuja, joiden sisällä olevassa elementissä tapahtuu muutos. Muutos voi heidän mukaansa olla siirtymä paikasta toiseen, väristä toiseen tai muodosta toiseen. Paanasen (2008, 14) mukaan liikeanimaatio on nimeltään motion tween ja muotoa muuttava animaatio on shape tween. Kaikki liike- ja muotoanimaatiot alkavat ja päättyvät aina avainruutuun. Avainruutujen väliset ruudut ovat tavallisia frameja eli ruutuja.

3.3 Työpohjat

Uutta dokumenttia luotaessa Adobe Flash CS3:ssa on valmiita esiasetuksia luotavan Flash dokumentin käyttötarkoituksen mukaan. Niissä on valmiiksi määriteltyinä muun muassa dokumentin mitat leveys- ja korkeussuuntaan, framerate eli kuinka monta ruutua sekunnissa dokumentti toistaa sekä muita julkaisuasetuksia. Näitä valmiita asetuksia kutsutaan työpohjiksi. (Adobe 2007.)

Työpohjissa on valmiit asetukset esimerkiksi standardikokoisille www-mainosbannereille ja maakohtaisia asetuksia eri puhelinmalleille. Valittavana on

myös työpohja kuvaesitykselle tai tietovisalle. Käytössä on myös yleisasetuksia suoraan puhelimen näyttökoon, tuettavan ActionScriptin tason tai Flash Lite Playerin mukaan silloin kun ei tehdä tiettyyn puhelinmalliin rajattua sovellusta. (Adobe 2007.)

3.4 Symbolit

Paananen (2008, 15-16) kuvailee symbolit Flash-elokuvan näyttelijöiksi. Elokuvasssa jokin elementti saattaa esiintyä usein, jolloin sen piirtäminen aina uudestaan ei ole järkevää. Tällöin elementti kannattaa tehdä symboliksi, jolloin se voidaan ottaa haluttaessa käyttöön aina uudestaan tai sijoittaa useita kappaleita näkyviin samaan aikaan. Symboleja voidaan muokata uudelleen, jolloin kaikki siihen tehdyt muutokset päivittyvät automaattisesti sen kopioihin.

Kaikki luodut symbolit tallentuvat automaattisesti dokumentin omaan kirjastoon. Mannisen ja Marttilan (2006, 16) mukaan, kun objekti tuodaan stagelle, jossa näkyvä elokuva koostetaan, siitä tulee alkuperäisen symbolin instanssi eli ilmentymä. Stagella olevat instanssit puolestaan ovat omia objektejaan, joiden ominaisuudet eivät ole sidoksissa muihin instansseihin tai alkuperäiseen symboliin. Tämän vuoksi instansseja voidaan muokata rajallisesti. Niiden mittasuhteita voidaan vääristää, kokoa voidaan skaalata, samoin peittävyyttä ja väriä voidaan muuntaa. Alkuperäisiä viivoja ei kuitenkaan voi käsitellä. Symbolit jaetaan kolmeen luokkaan niiden ominaisuuksien ja funktioiden mukaan.

Graafinen symboli

Paananen (2008, 18) kuvailee graafista symbolia elokuvan perusrakennemateriaaliksi ja vertaa sitä tiilitalon tiileen. Graafiseksi symboliksi tehdään yleensä usein toistuvia elementtejä, joista rakennetaan suurempia kokonaisuuksia. Graafinen symboli voidaan animoida liikkumaan omalla aikajanallaan, mutta se voidaan animoida movie clip-symboliksi, jolloin sen liike ei ole enää sidoksissa pääaikajanaan.

MovieClip -symboli

Manninen ja Marttila (2006, 188) pitävät MovieClip-symbolia koko Flash-

sovelluksen perustana. Yksi MovieClip voi koostua useammasta sisäkkäisestä MovieClipistä joista, kullakin on oma itsenäinen aikajana. MovieClip voidaan toistaa esityksen aikajanalla, mutta koska sen ei tarvitse olla sidoksissa esityksen aikajanaan, sitä voidaan toistaa esityksen ollessa pysäytettynä. MovieClip-symboleja voidaan ohjata symbolin ulkopuolelta.

Painike -symboli

Manninen ja Marttila (2006, 146) kuvailevat painiketta yksinkertaistetuksi versioksi MovieClipistä. Painikkeella on automaattisesti käytössä hitarea-aikajana, joka kuvaa painikkeen aktiivista aluetta. Oletuksena hitarea on painikkeen ääriiviojen mukainen, mutta se voidaan myös määrittää erikseen painikkeen koosta tai muodosta riippumattomaksi.

Tyypillinen painike rakentuu neljästä vaiheesta, joille voidaan määrittää erilaiset ulkoasut. Niitä ovat: Up, eli normaalitilassa oleva painike; Over, eli osoittimen ollessa hiiren päällä; Down, eli osoittimen ollessa painettuna painikkeen päällä; Hit, eli painikkeen aktiivinen alue, joka määrittää painikkeen aktiivisen alueen (Paananen 2008, 18). Jokainen eri vaihe voi sisältää graafisia symboleja, Movieclip-symbolileja tai ääntä.

Bhargal (2001, 132) puhuu näkymättömän painikkeen luomisesta. Tällöin painikkeelle ei määritellä minkäänlaista Up-, Over- tai Down -näkyä. Näkymättömään painikkeeseen määritellään ainoastaan Hit-alue, jota hän kuvaa hot spotiksi. Näkymättömiä painikkeita voidaan sijoittaa painikkeelta näyttävien graafisten symbolien päälle.

3.5 ActionScript

Ellei muuta ole määritelty, Flash-elokuva toistaa itseään alusta loppuun päättymättömällä toistolla. Flash-elokuvaan saadaan aikaan vuorovaikutteisuutta ActionScript-koodilla. ActionScript on alun perin Macromedian ja nykyisin Adoben omistaman Flash-ohjelman ohjelmointikieli. Yksinkertaisimmillaan koodia voidaan käyttää elokuvan toiston pysäyttämiseen, mutta pidemmällä komentosarjoilla voidaan toteuttaa animaatioita ja grafiikkaa ilman piirtotyökaluja. (Paananen 2008, 14.)

ActionScriptiä voidaan Paanasen (2008, 19-20) mukaan sijoittaa kahteen paikkaan. Sitä voidaan kirjoittaa instansseihin ohjaamaan niiden toimintaa, kuten esimerkiksi tiettyä painiketta painaessa pysäytetään Flash-elokuvan toisto. Koodia voidaan myös kirjoittaa avainruutuun, jolloin siihen koodatut toiminnot ajetaan heti, kun elokuvan toisto osuu sen kohdalle.

Mannisen ja Marttilan (2006, 21) näkemysten mukaan koodit tulisi kirjoittaa esityksen ensimmäisiin avainruutuihin, jolloin ne suoritetaan heti esityksen alussa ja tärkeimmät toiminnot saadaan heti käyttöön. ActionScript-koodi voi sijaita millaisessa avainruudussa tahansa. Esityksen tekijän kannalta koodi on selkeämpää jos sille tehdään aikajanelle kokonaan oma tasonsa. Koodi voidaan jakaa vielä useammalle tasolle esimerkiksi siten, että esityksen toistoa käsittelevä koodit ovat omalla tasollaan ja elokuvaleikkeitä ja painikkeita koskevat koodit omalla tasollaan.

3.6 Flash Lite

Kaikki Adobe Flash -ohjelmalla toteutetut sovellukset ja animaatiot tarvitsevat Flash Player-ohjelmaa, jotta niitä voidaan toistaa. Matkapuhelimessa toimivaa Flash Playeriä kutsutaan Flash Lite:ksi. Flash Lite on nimensä mukaan se on kevennetty versio alkuperäisestä ohjelmasta, ja siitä johtuen ohjelmointimahdollisuudet sekä tuki muille medioille ovat paljon suppeammat. (Adobe, 2009.)

Flash Lite 1.x

Flash Lite 1.x oli ensimmäinen Flash Lite versio. Sen pohjana hyödynnettiin Flash Player 4-versiossa käytettävää ActionScript 1:tä joitakin pelkästään matkapuhelimelle tarkoitettuja skriptejä, joita normaali Flash Player ei voi toteuttaa. Kollektiivisesti näistä käytetään nimeä Flash Lite ActionScript 1.x. (Adobe, 2007.)

1.x -versio tukee sekä staattisia että dynaamista tekstiä sekä tekstin syöttöä. Kirjasindataa voidaan myös sisällyttää dokumenttiin. Audioformaateista tuettuja ovat MIDI-äännet sekä Flash audio, joka on Flashin natiivi audioformaatti. 1.x -versioon voidaan hakea sovelluksen ulkopuolelta dataa internetistä sekä muita flash sovelluksia. Sovellus voi myös kommunikoida itse puhelimen kanssa, jolloin sille

vellus voi myös kommunikoida itse puhelimen kanssa, jolloin sille voidaan esimerkiksi antaa käskyjä soittaa tai lähettää tekstiviesti haluttuun numeroon. (Emt.)

Flash Lite 2.x

2.x versio pohjautuu Flash Player 7-versioon ja ActionScript 2:een, jota oli laajennettu matkapuhelimelle tarkoitetuilla skripteillä. Tämä oli ensimmäinen versio, jossa dokumenttiin voitiin sisällyttää videokuva. Videon hallinta onnistui sovelluksen kautta, mutta videon toiston hoiti kuitenkin puhelimen oma videon toisto-ohjelma. Sama ominaisuus on pätee muihinkin puhelimen tukemiin kuva ja audioformaatteihin. (Emt.)

Flash Lite 3.x

Flash Lite 3.x pohjautuu Flash Palyer 8-versioon ja ActionScript 3:een, jota oli laajennettu matkapuhelimelle tarkoitetuilla skripteillä. Tämä versio pystyi ensimmäisenä toistamaan itsenäisesti siihen liitettyä videokuva. Uutena ominaisuutena oli muun muassa kyky tulkita html-koodia. (Emt)

4 KÄYTETTÄVYYS

4.1 Määritelmiä

Wiion (2004, 28-31) muukaan termiä käytettävyys ei voida määritellä aukottomasti. Hän lähestyy asiaa käyttäjäystävällisen sovelluksen näkökulmasta ja antaa sille neljä seuraavaa termiä:

- **Ymmärrettävässä** sovelluksessa käyttäjä voi helposti päätellä, kuinka haluttuun lopputulokseen voidaan päästä. Käyttäjä voi helposti päätellä sovelluksen toimintaperiaatteen ja sisäistää sen nopeasti. Ymmärrettävyys ei koske pelkästään sovelluksen toimintaa tai sen logiikkaa. Informatiivinen sisältö on myös merkittävässä osassa. Sovelluksen tekstisisältö tulisi kirjoittaa sellaiseen muotoon, jotta sen kohdetyhmä ymmärtää sitä. Selkeinkin käyttölogiik-

ka menettää merkityksensä, jos käyttäjä ei ymmärrä sovelluksen sisältöä. Ymmärrettävyyttä voidaan näin ollen pitää myös yksilökohtaisena asiana.

- **Vaivattomassa** sovelluksessa käyttäjä pystyy yksinkertaisella tavalla suoriutumaan tehtävästään. Vaikka sovellus olisikin täysin ymmärrettävä, se ei välttämättä ole vaivaton käyttää. Esimerkkinä Wiio antaa tästä tekstiviestin kirjoittamisen kännykällä. Kirjainten syöttämisen toimintaperiaate on selkeä, mutta tavalliseen tietokoneen näppäimistöön verrattuna se on aikaa vievä toimenpide.
- **Kattavan** sovelluksen tulisi tarjota kaikki ne toiminnot, joita käyttäjä tarvitsee suoriutuakseen tehtävästään. Wiion mukaan vaivattomuus- ja kattavuusongelmat esiintyvät useimmiten yhdessä.
- **Esteettisesti miellyttävä** sovellus viestii laatua ja osaamista. Jokaisen sovelluksen ulkoasun tulisi olla esteettisesti miellyttävä. Epämiellyttävä ulkoasu yhdistetään useimmiten laadun puutteeseen. Miellyttävässä ja hillityssä ulkoasussa käyttäjän huomio kiinnittyy enemmän informaation sisältöön, kuin siihen miten se on esitetty. Sovelluksen ulkoasu viestii käyttäjälleen sen takana olevasta yrityksestä tai yhteisöstä. Esteettisesti oikein suunniteltu sovellus viestii yhtenäisesti yrityksen viestintätavoitteiden mukaista imagoa.

4.2 Käyttöliittymäsuunnittelu

4.2.1 Käsikirjoittaminen

Manninen ja Marttila (2006, 79–80, 397–381) puhuvat Flash-sovelluksen suunnittelun yhteydessä rakennemalleista. Tällä tarkoitetaan sovelluksen rakenteesta ja toiminnasta piirrettävää mallia, jota voidaan pitää erilaisena käsikirjoituksena sovellukselle. Vaikkei sovelluksen toteuttamiselle olekaan yhtä ainoata lähestymistapaa, jonkinlaista suunnitelmaa sen toteuttamisesta pidetään hyvänä lähtökohtana. Suunnitelmaan listataan vuokaavion omaisesti sovellukseen kuuluvat pääsivut, niiden sisällöt ja alasisivut.

Luukkonen (2000) puolestaan puhuu digitaalisen media käsikirjoittamisesta. Hän esittelee MaindMap- eli ajatuskarttamenetelmän käsikirjoittamisen työkaluksi asiasisältöjen jaottelemiseen sekä tiedon hierarkian määrittelemiseen. Hänen mukaansa käsikirjoitusprosessia on pidetty sovellusten ja sivustojen kehittämisessä vähemmän tärkeänä osa-alueena, eikä sen kunnolliseen tekemiseen ole kovin usein paneuduttu.

Luukkonen (Emt, 106–109) jakaa ajatuskartat kolmeen eri kategoriaan: ruoto- ja puumallisiin sekä niiden yhdistelmiin. Ruotomallissa asiakokonaisuudet etenevät lineaarisesti ja usein kronologisesti asiakokonaisuudesta toiseen. Puumallissa asiakokonaisuudet on aseteltu hierarkkisille tasoille. Näistä päällimmäisenä on fiktiivinen taso. Sen jälkeen tulevat fakta- ja detaljitason. Mitä alemmas hierarkiassa mennään, sitä yksityiskohtaisemmaksi informaatio muuttuu.

Luukkonen (2000, 156–168) jakaa käsikirjoituksen kolmeen eri kierrokseen. Jokaiselle vaiheelle on annettu eri tarkoituksensa. Luukkonen vertaa tätä prosessia sipulin kuorimiseen, jossa kerros kerrokselta päästään lähemmäs asian ydintä.

1. kierros

Ensimmäinen kierros on tarkoitettu läpikatsaukseksi koko kirjoitusprosessille. Siinä kartoitetaan sovellukseen liittyvät kokonaisuudet ja yksityiskohdat, asiat, jotka ovat jo tiedossa ja ne mistä pitää ottaa selvää. Jokainen sovellukseen tuleva ruutu nimitetään ja sen sisältö kuvaillaan. Sovelluksen ulkoasu ei vielä ole merkittävässä osassa. Tarvittavat linkit ja tiedostohierarkiat tulevat selvästi esille. Ensimmäisessä vaiheessa luodaan pohja perusnavigaatiolle.

2. kierros

Toisella kierroksella listataan jokaiseen ruutuun tuleva lähdemateriaali. Materiaali voi olla ääntä, kuvaa tai videokuvaa. Ulkoasusta voidaan luonnostella joitakin ideoita, mutta tässäkin vaiheessa ei tehdä lopullisia linjanvetoja. Navigoinnin lisäksi listataan muita linkkejä, joita samaan ruutuun on mahdollisesti tulossa. Tässä vaiheessa ei tulisi materiaalin suhteen olla liian kriittinen. Kaikki merkityksellinen aineisto tulisi listata, vaikkei se päätysikään lopulliseen sovellukseen.

3. kierros

Kolmatta kierrosta pidetään useimmiten koko käsikirjoituksen ainoana vaiheena. Tässä vaiheessa määritellään miten ruudussa oleva informaatio tuodaan käyttäjälle näkyviin. Liikkuvalla kuvalla, oli se animaatiota tai videokuvaa, määritellään sillä viestittävä informaatio. Mikäli sitä selvennetään tekstillä tai selostuksella, se kirjoitetaan muistiin. Käytettävistä kuvista määritellään sisältö ja rajaukset. Vasta käsikirjoituksen tässä vaiheessa muotoillaan sovelluksen tekstisisältö. Kolmannen vaiheen versio on vasta ensimmäinen versio, joka lähtee tilaajalle tarkastettavaksi. Tarkastuksessa tilaaja esittää korjauksia kieli- ja ulkoasuun sekä asiasisältöihin, sikäli kuin katsoo tarpeelliseksi.

4.1.2 Käyttäjryhmä suunnittelun lähtökohtana

Pettersonin (1996, 35–36) mukaan käyttöliittymä ei ole kääre, johon valmis sovellus kääritään, jotta se menisi kaupaksi. Yleisin virhe käyttöliittymäsuunnittelussa on se, että suunnittelu tehdään sovellusta silmälläpitäen, ei käyttäjää. Käyttöliittymä on kanava, jonka kautta käyttäjä käy dialogia sovelluksen kanssa. Käyttöliittymän suunnittelijan pitää siis tuntea kohderyhmä, jotta päästään toimivaan lopputulokseen.

Yksi Nielsenin (1993, 123–125) listaamista heuristisista säännöistä on puhua käyttäjän kieltä. Luonnollisesti sovellukset tulisi toteuttaa kielellä, joka on käyttäjän oma äidinkieli. Sen lisäksi tulisi pyrkiä käyttämään kohderyhmän omaa ammattisanastoa tai jargonia. Käyttäjän kielellä puhumista ei pitäisi nähdä rajoituksena suunnittelussa. Tietyillä termeillä puhumisesta on enemmän hyötyä kohderyhmälle kuin vähemmän tarkoilla yleisen puhe- tai kirjakielen termeillä.

4.2.3 Typografia

Sana typografia muodostuu kreikankielisistä piirtämistä ja kirjoittamista tarkoittavista sanoista. Kuoppalan ja muiden (2002, 144) mukaan typografialla tähdätään tyylik-

kääseen ja selkeään ulkoasuun. Laajemmin käsiteltynä Metsämäen (1995, 79–81) mukaan typografiaan kuuluu myös eri kappalemallien, kuten otsikoiden, ingressien ja leipätekstien sijoittelu ja sommittelu.

Wiion (2004, 201-202) mukaan pyritään hyvään luettavuuteen. Tavoitteena on nostaa esille tärkeimmät asiasisällöt tekstistä. Typografialla on lisäksi viestinnällisiä tarkoituksia, sillä visuaalisen ulkoasun kautta voidaan viestiä myös imagosta.

Kuoppala ja muut (2002, 144) jakavat kirjasimet muotoilun mukaan antiikvoiksi ja groteskeiksi, sekä koon mukaan versaaleiksi ja gemenoiksi. Antiikvat kirjasimet ovat päätteellisiä kirjasimia, joissa nousevat kirjasinpylväät ovat kapeampia kuin laskevat. Tätä kutsutaan kirjasimen kontrastiksi. Groteskit kirjasimet ovat päätteettömiä ja useimmiten niillä ei ole minkäänlaista kontrastia. Versaalit puolestaan tarkoittavat suuraakkosia, gemenoiden ollessa pienaakkosia.

Kirjasintyyppin valintaan vaikuttaa se, luetaanko tekstiä näytöltä vai näyttöpäätteeltä. Wiion (2004, 206-209) mukaan ihminen lukee painettua antiikvaa nopeammin kuin groteskia tekstiä. Näytöllä taas antiikvan tekstin yksityiskohdat eivät välttämättä toistu yhtä tarkasti kuin paperilla painettuna. Tämä johtuu näytön heikommasta piirto-tarkkuudesta, joka kuitenkin on parantumassa jatkuvasti. Tasapaksut ja päätteettömät groteskit kirjasimet piirtyvät tarkemmin näytölle ja ovat siksi parempia vaihtoehtoja. Painetussa tekstissä otsikot ja väliotsikot ovat usein yksirivisiä ja usein lukijan halutaan ensimmäisenä keskittävän huomionsa juuri niihin. Siksi ne ovat usein kirjoitettuja groteskilla.

Wiion (2004, 208) mukaan kirjasimilla ei tulisi keikaroida. Liian monen eri kirjasinleikkauksen käyttö aiheuttaa levottoman tunnelman, joka puolestaan vähentää esteettistä miellyttävyyttä. Hän suosittelee käytettäväksi korkeintaan kahta erilaista kirjasinta: yhtä otsikoille ja toista varsinaisille tekstile. Yhdestä kirjasintyylistä voidaan kuitenkin vaihtelun aikaansaamiseksi käyttää eri kokoja, kursivointia tai lihavointia. Hän ei kuitenkaan suosittele alleviivauksen käyttöä, koska se heikentää kirjasinten erottuvuutta toisistaan. Lisäksi alleviivaus selaimessa viittaa useimmiten linkkiin, ja sovelluksissa käytettynä se saattaa johtaa käyttäjää harhaan.

Luettavuus

Vaikka kirjasintyyppin valinta vaikuttaakin oleellisesti tekstin luettavuuteen, se ei ole ainoa asia, joka vaikuttaa siihen. Wiio (emt. 206–207) antaa rivivälille painoarvoa. Rivivälin pitäisi olla riittävän suuri. Kahden rivin väliin pitää jäädä tarpeeksi tyhjää tilaa, jotta silmän on helpompi löytää seuraavan rivin alkuun.

Kappalejaon pitää olla selvä (Emt. 209-210). Lukijalle pitää tehdä selväksi milloin siirrytään asiakokonaisuudesta seuraavaan. Yleensä tekstinkäsittelyohjelmissä tämä tehdään yleensä tyhjällä rivillä. Samalla tai pelkästään voidaan käyttää myös ensirivin sisennystä. Mitä vähemmän tyhjää tilaa jätetään kappaleiden väliin, sen enemmän tulisi ensimmäistä riviä sisentää.

Wiion mukaan (Technologos, 2007) katseen kuljetus on myös luettavuuden kannalta tärkeää. Tärkeimmät asiat tulisi pystyä poimimaan tekstin seasta yhdellä vilkaisulla. Sen vuoksi otsikoiden ja väliotsikoiden tulisi selvästi erottua tekstissä. Tämän lisäksi yksittäisiä tärkeitä asioita voidaan korostaa esimerkiksi lihavoinnilla.

4.2.3 Navigointi

Kuoppalan ja muiden (2002, 49) mukaan juuri navigointi saattaa olla sovelluksen suunnittelun pullonkaula. Navigointi ei juuri koskaan ole osa sovelluksen sisältöä vaan työväline sovelluksen sisällä liikkumiseen. Sen tulisi olla mahdollisimman helppoa ja luonnollista. Käyttäjryhmä ja sen tarpeet huomioiden navigointi ja sen elementit tulisi laatia juuri käytettävyyden näkökulmasta.

Eräs Nielsenin (1993, 138-141) listaamista heuristisista säännöistä kieltää johtamasta käyttäjää umpikujaan. Käyttäjällä tulisi olla jokaisesta tilanteesta mahdollisuus palata takaisin tai perua viimeisiä tekemiään toimintoja. Käyttäjällä pitää olla mahdollisuus siirtyä sovelluksen sisällä paikasta toiseen. Linkit tai pikanäppäimet pitää olla selkeästi ja loogisesti määritettynä sekä esitettynä, jotta niiden käyttö olisi tehokasta. Oikopolun ei koskaan pitäisi johtaa käyttäjää harhaan.

4.2.4 Ruutusuunnittelu

Nielsenin (1993, 117) mukaan hyvällä ruutusuunnittelulla saavutetaan yksinkertainen ja looginen dialogi käyttäjän ja sovelluksen välillä. Käyttöliittymän toteutuksessa tulisikin tukeutua Gestalt-sääntöihin, eli hahmolain periaatteisiin. Kun ymmärretään, miten ihminen tulkitsee ja prosessoi näkemiään asioita, voidaan rakentaa toimivampia käyttöliittymiä.

Sommittelun rakenteen tulisi auttaa käyttäjää ymmärtämään sovelluksen rakennetta. Valikoiden ja asiasisältöjen jäsentelyyn voidaan käyttää jakavia linjoja tai värikoodeja jakamaan asiakokonaisuuksia omiin ryhmiinsä. Käyttäjän tulisi kyetä erottamaan ruudun elementit toisistaan. Hahmolain mukaan kohteet, jotka ovat tarpeeksi lähellä toisiaan, mielletään yhdeksi kokonaisuudeksi. Kun värikoodeilla voidaan eriyttää tai sitoa asioita toisiinsa ruudun sisällä. Ruudun sisällä olevat elementit olla erillään toisistaan, se voidaan tehdä myös kehystämällä kohteita (Emt. 118–119).

Värienkäytön suhteen tulisi olla varovainen. Sovelluksen tulisi esteettisen miellyttävyyden vuoksi seurata jonkinlaista väriskaalaa. Väriskaala tulisi rajoittaa viidestä seitsemään väriin, koska sen verran keskivertokäyttäjä pystyy muistamaan ja erottamaan toisistaan. Kokeneinkin käyttäjä pystyy prosessoimaan korkeintaan yksitoista väriä. Yleensä harmaasävyt ja pastelliset värit ovat suositeltavampia suositeltavampia taustavärejä, koska räikeät värit saattavat häiritä tekstin lukua. Väriskaalaa valitessa tulisi myös huomioida kohderyhmä. Koska noin 8 %:lla ihmisistä on jonkinasteinen heikentynyt värinäkö, tulisi varmistaa, ettei ruudun värikoodeja voi tulkita väärin. (Emt. 119–120.)

Metsämäen (1995 17–21) mukaan oikeanlainen elementtien sommittelu antaa tasapainoisen vaikutelman ja helpottaa käyttäjää löytämään nopeasti tärkeimmät asiat. Kultaisen leikkauksen sääntöjen avulla ruudusta voidaan määrittellä niin sanottuja hotspotteja, jotka ovat ruudun lukemisen kannalta tärkeitä pisteitä. Näihin pisteisiin olisi hyvä sijoitella tärkeimmät asiasisällöt tai graafiset elementit. Tasapainoisessa sommittelussa on myös kevyet elementit yleensä ruudun yläreunassa ja raskaimmat elementit alareunassa.

Nielesen (1995, 120-121) antaa hyvän muistisäännön sanomalla "Less is more" eli vähemmän on enemmän. Tämä tarkoittaa ettei käyttäjän havaintokyky saisi kuormittaa ahtamalla ruutua täyteen informaatiota. Mitä vähemmän informaatiota annetaan kerralla, sen helpompaa sitä on prosessoida. Tämä ei koske pelkästään asiasisältöä vaan myös muita elementtejä, kuten painikkeita tai kuvia. Jos esimerkiksi navigoinnin suhteen annetaan kerralla kaikki oikopolut jokaiseen kohteeseen, halutun painikkeen löytäminen saattaa olla vaikeaa ja liian monet vaihtoehdot vievät tilaa muilta tärkeiltä asioilta. Käyttöliittymäsuunnittelussa joudutaan tämän vuoksi tekemään usein kompromisseja.

5 KÄYTTÖLIITTYMÄN TOTEUTUS

5.1 Käsikirjoitus

Sovelluksen käsikirjoitusta aloin rakentaa opetuksessa käytettävien oppimateriaalien pohjalta. Materiaali koostui monisteista, PowerPoint-esityksistä sekä CD-ROM -levyllä toimitetusta multimediaesityksestä. Käsikirjoitusta alettiin tehdä FreeMind-nimisellä ajatuskarttaohjelmalla, joka on ladattavissa ilmaiseksi Internetistä.

Ensimmäistä käsikirjoituksen versiota tarkasteltiin sekä painoviestinnän opettajien ja ammattimiehen kanssa. Ensimmäisen version kuuden pääkohdan sijaan opettajat katsoivat paremmaksi irrottaa puristuksen säädön kokonaan omaksi vaiheekseen ja lisätä kostutuslaitteen säädön uudeksi vaiheeksi. Lisäksi korjattiin ja täsmennettiin muutamia muita vaiheita. Sääntönä jokaisen vaiheen tekstisisällölle oli, ettei se saa olla yhtä tekstiviestiä, eli 160 merkkiä, pidempi, välilyönnit mukaan lukien.

Käsikirjoituksen toisen version katsottiin olevan valmis sovelluksen havainnollisen sisällön eli videoklippien kuvaamista varten. Kuvaukset toteutin opetustilanteessa yhdessä painotekniikan opiskelijoiden kanssa. silloisen ryhmän opettaja toimi tilanteessa mukana teknisenä avustajana, joka valvoi sekä opetus- että kuvaustilannetta.

Hänen roolinsa oli valvoa, että videolle tallennettava toiminta oli oikein suoritettua ja turvallisuusmääräysten mukaista.

Kuvauksia lähdettiin toteuttamaan käsikirjoituksen toisen version mukaan. Kohtaukset kuvattiin samassa järjestyksessä kuin ne oli merkittynä käsikirjoitukseen. Kuvausten yhteydessä päätettiin käsikirjoituksesta poistaa värilaatikon säätö kohdasta hyppytelan päälle kääntö sekä vedoksen ajosta väritelöjen esivärjäys. Molemmat toimenpiteet voidaan suorittaa GTO-52-painokoneella, mutta koska kyseisessä koneessa esivärjäys tapahtuu automaattisesti, ja hyppytelan päälle kääntö ei ole välttämätön toimenpide, niiden ei katsottu tuovan lisäarvoa lopulliselle sovellukselle. Niinpä käsikirjoituksen kolmas versio (liite 1) valmistui vasta kuvausten yhteydessä. Tämän version pohjalta olen rakentanut lopullisen sovelluksen käyttöliittymän.

5.2 Videon käyttö

5.2.1 Videokamerasta kännykän näytölle

Digitaalisella videokameralla kuvattu materiaali ei sellaisenaan ole käyttökelpoista tai yhteensopivaa. Ensiksi se täytyy editoida ja sen jälkeen tallentaa puhelimen kanssa yhteensopivaan formaattiin yhteensopivalla pakkausmenetelmällä.

Kuvamateriaali kaapattiin videokameralta tietokoneelle ja editoitiin Adobe Premiere CS3-ohjelmalla. Raakamateriaalia kertyi yhteensä 45 minuuttia josta editoitiin 28 yksittäistä klippiä yhteiskestoltaan 3,5 minuuttia. Hyvälaatuinen digitaalinen video vie muistitilaa 100 megatavua minuuttia kohti. 350 megatavua saattaa olla enemmän, kuin hieman vanhemman puhelimen muistiin tai muistikortille mahtuu. Lisäksi vaikka tiedostoformaatti olisikin yhteensopiva, puhelimen prosessointiteho ei riitä toistamaan videota.

Adobe Premierestä voidaan tallentaa videota useaan eri formaattiin ja sieltä löytyy valmiit esiasetukset erilaisille medioille, kuten juuri matkapuhelimille. Esiasetuksissa on valmiiksi määriteltynä kuvan pakkaussuhde, kuvan koko ja ruutujen toistonopeus, eli kuinka monta ruutua sekunnissa videota toistetaan.

PAL-järjestelmässä, joka on vallitseva Eurooppalainen formaatti, videosignaali muodostuu ylä- ja alakentästä. Yläkenttä sisältää signaalin parittomat rivit ja alakenttä parilliset rivit. Kentät piirretään näytölle vuorotellen yhteensä 50 kertaa sekunnissa, jolloin videosignaali näyttää yhteensä 25 täyttä kuvaa sekunnissa. Tätä kutsutaan kuvan lomittamiseksi ja sitä käytetään vähentämään putkitelevisioiden välkkymistä. Nykyisellä tekniikalla litteissä näytöissä, puhelimen näytöt mukaan lukien, kuva toistuu tarkempana, jos siitä poistetaan lomitus. Tällaista videokuvaa kutsutaan progressiiviseksi, jolloin kahden erillisen kentän vuorottelun sijaan näytetään kokonaisia kuvia.

Lopulliseksi tallennusmuodoksi valitsin mp4-formaatin, joka on yhteensopiva kaikkien Symbian 60-pohjaisten puhelimien kanssa. Videon kooksi valitsin 240x160 pikseliä, joka 4:3-mittasuhteessa on näytön levyinen vaakasuunnassa. Pakkaamisen jälkeen kaikki videoklipit veivät muistitilaa yhteensä seitsemän megatavua. Alkuperäisestä tiedostokoosta päästiin yhteen viideskymmenesosaan ja videon laatu säilyi vielä siedettävänä.

5.2.2 Videon tuonti sovellukseen

Videota voidaan Flash Litessä käyttää kolmella tavalla: videot voidaan sisällyttää sovellukseen, ne voidaan toistaa puhelimen omasta muistista ja ne voidaan hakea internetissä olevalta palvelimelta. Internetin kautta toistettavat videot edellyttävät puhelimelta yhteyttä verkkoon.

Ellei puhelimessa ole langatonta ominaisuutta ja käytössä langatonta verkkoa, tiedonsiirto on maksullista. Sovellukseen sisällytettävät videot taas kasvattavat lopullisen sovelluksen kokoa, joka saattaa rasittaa liikaa puhelimen muistikapasiteettia sekä prosessointikykyä.

Koska sovellusta käytetään opetuksen tukimateriaalina, ei voi vaatia maksamaan sen käytöstä eikä vaihtamaan puhelinmalliin, jossa on verkkoyhteysmahdollisuus. Mahdollisimman laajan puhelinmallien yhteensopivuuden saavuttamiseksi valitsin käytet-

täväksi Flash Lite 1-version, jossa videon toiston hoitaa puhelimen oma soitin ja videot toistetaan puhelimen omasta muistista. Tällöin itse sovelluksen koko pysyy pienenä eikä puhelimen suorituskyky rasitu liikaa.

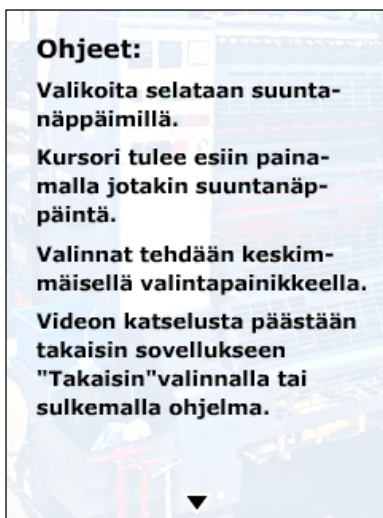
5.3 Ruutusommittelu

Alkuanimaatio pysähtyy alkuruutuun (kuvio 1), jossa ilmaistaan logoilla laitteen valmistaja Heidelberg sekä laite GTO 52-painokone. Alapuolella oleva teksti "Painokuntoonlaitto" kertoo sovelluksen sisällöstä. Logojen ja tekstin välille on jätetty tyhjää tilaa, jotta ne erottuvat toisistaan. Tämä kolmen elementin kokonaisuuden keskipiste sijoittuu ruudun yläkolmannekseen, jossa kultaisen leikkauksen linja kulkee. Kaikki elementit on keskitetty vaakasuunnassa.



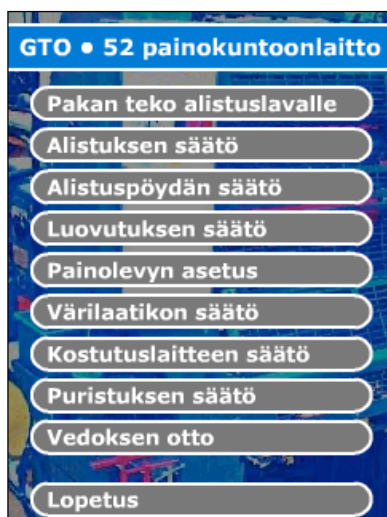
KUVIO 1: Näkymä ensimmäisessä pysäytysruudussa.

Toisessa pysäytysruudussa sijaitsee sovelluksen käytön kannalta oleellisia ohjeita (kuvio 2). Fontiksi on valittu Verdana Bold. Groteskina kirjasimena se piirtyy näyttölle tarkemmin ja on helpompi lukea. Kaikissa sovelluksessa olevissa teksteissä käytetään samaa fonttia. Otsikon fonttikoko on 16 pistettä ja muun teksti 13 pistettä. Riviväli on oletuksena olevan 120 %:in sijaan nostettu 150% luettavuuden parantamiseksi. Lauseet on erotettu toisistaan nostamalla riviväli 200 %:iin



KUVIO 2: Näkymä toisessa pysäytys ruudussa

Kolmas pysäytysruutu on sovelluksen päävalikko (kuvio 3) Sovelluksen otsikko on erotettu omaan palkkiinsa poikittaisilla viivoilla ja taustavärillä. Alapuolella on painikkeet, joista pääsee selaamaan alakohtaisia säätöjä. Alimpana olevalla lopetus-painikkeella poistutaan sovelluksesta. Se on erotettu muista painikkeista isommalla tyhjällä tilalla jottei, sitä luultaisi yhdeksi navigointipainikkeeksi.



KUVIO 3: Näkymä päävalikossa

Päävalikossa, alavalikossa ja yksityiskohta valikossa otsikko on aina samassa paikassa ruudun yläreunassa. Sen jälkeen pienen välin jälkeen tulee painikeryhmä yksityiskohtaisiin säätöihin. Alavalikoissa ensimmäisen painikkeen sijainti on aina sama, (kuvio 4) painikkeiden määrästä riippumatta. Navigointipainikkeet on sijoitettu ruudun alareunaa ja niiden välit on samat kuin kaikilla muilla painikkeilla. Koko sovelluksessa ruudun alimmaisen painikkeen paikka on aina sama.



KUVIO 4: Näkymä alavalikossa

Yksityiskohtavalikossa otsikon alla on tarkentava alaotsikko, joka kertoo kulloisenkin säätökohdan nimen. Keskellä ruutua on neutraaliharmaalla taustalla säätöä kuvaava ja tarkentava tekstikenttä, joka otsikko ja alaotsikkokentän tapaan on erotettu omaksi kokonaisuudekseen. Alavalikot on erotettu toisistaan eri koodiväreillä ja kaikissa alavalikon alla olevissa yksityiskohtavalikoissa toistetaan samaa värikoodia.



KUVIO 5: Näkymä yksityiskohtavalikossa

5.4 Navigointi

Sovelluksessa ei ole syviä hierarkkisia rakenteita. Tästä johtuen navigointi voidaan toteuttaa yksinkertaisella tavalla. Sovelluksessa on käytännössä kolme tasoa (liite 1). Päällimmäisenä tasona on päävalikko, josta lähtee linkki jokaiseen yhdeksään keskimmäisen tason ruutuun. Keskimmäinen taso on alavalikkotaso, jossa jokainen ruutu on otsikoitu sen sisältämän säätöalueen mukaan. Jokaiseen alavalikon ruutuun on listattu sen sisältämät yksityiskohtaiset säädöt, jolta on linkki sitä vastaavaan pohjatason ruutuun.

Sen lisäksi, että tasoilla voidaan navigoida pystysuuntaisesti, tasoilla voidaan liikkua myös vaakasuuntaisesti. Navigointia helpottaa, jos keskitasolla tai alatasolla voidaan siirtyä säätöalueesta tai yksityiskodasta seuraavaan käymättä sitä ennen edellisen tason valikossa.

Sivusuuntainen liikkuminen on rajoitettu pelkästään "edellinen" ja "seuraava" -komentoihin. Lisäksi alimmalta tasolta pääsee päävalikkoon vain edellisen alavalikon kautta. Sovellus on rakenteeltaan niin yksinkertainen, että muunlaiset mahdollisuudet anna minkäänlaista lisäarvoa sovellukselle. Lisäksi jokainen uusi linkki tarvitsisi uuden painikkeen, joka taas vie tilaa ruudulta. Käytettävyyden kannalta oli

parempi karsia kaikki ylimääräinen asia pois ruudulta. Pidin sitä hyvänä ratkaisuna, että navigoinnin toteuttaa loogisesti maksimissaan kolmella painikkeella ruutua koh-
ti.

5.5 Painikkeet

Kaikki navigointi sovelluksessa suoritetaan painikkeiden kautta. Painikkeiden avulla päästään myös katselemaan sovelluksen videoklippejä. Sovelluksessa on käytetty kahdessa eri muodossa toimiva painikkeita.

Sovelluksen kahdessa ensimmäisessä pysäytysruudussa (kuvat 1 ja 2), ruudun alareunassa vilkkuu alas päin osoittava kolmio. Tällä kerrotaan käyttäjälle, että kun ristiohjainta painetaan alaspäin, tapahtuu jotain. Painike on toteutettu laittamalla painikkeen Up-tilaan MovieClip, jossa on vilkkuva nuoli. Over-, ja Down-tilat on jätetty tyhjiksi. Kun ristiohjainta painetaan johonkin suuntaa, kursori tulee ruutuun painikkeen päälle. Kursorissa oleva koodi käskää sovellusta siirtymään seuraavaan ruutuun, jossa on sama painike, jossa on sama koodi. Itse asiassa sillä ei ole merkitystä mihin suuntaan käyttäjä painaa ristiohjainta. Koska ruudussa on vain yksi painike, kursori siirtyy aina sen päälle.

Muut sovelluksessa olevat painikkeet erottaa sovelluksessa niiden pyöreistä reunoista. Normaalityltilassa jokaisen taustaväri on neutraaliharmaa. Over-tilassa väri muuttuu painikkeesta riippuen. Down-tilassa niiden taustaväri muuttuu mustaksi ja ilmaisee samalla käyttäjälle, että se rekisteröi painalluksen.

Päävalikossa over-tilan väri on sama kuin alavalikon vastaava tunnusväri. Poistupainikkeessa over-väri on räikeä punainen, joka huomauttaa käyttäjälle, ettei kyseessä ole navigointipainike. Sekä alavalikossa että yksityiskohtavalikossa navigointipainikkeet ovat omassa ryhmässään alareunassa. Kaikkien navigointipainikkeiden over-väri on sininen kuin myös videoklipin avaavan painikkeen. Alavalikkoon ja päävalikkoon johtavien painikkeiden väri on oranssi.

5.6 ActionScript -koodaus

Koodia on sovelluksessa kirjoitettu sekä aikajanelle että painikkeisiin. Aikajanelle oleva koodi asettaa sovelluksen alussa toiston haluttuun tilaan ja pysäyttää sen ensimmäiseen pysäytysruutuun. Sovelluksen ensimmäisessä ruudussa on seuraava koodi:

```
fscommand2("SetQuality", "high");
fscommand2("FullScreen", true);
```

Koska sovellus sisältää tekstiä, näytön laatuasetukset pakotetaan parhaimmaksi koodin ensimmäisen rivin komennolla. Tällöin pienet kohteet piirtyvät tarkemmin ruudulle. Oletuksena kaikki flash-elokuvat ja sovellukset rajautuvat sen stagella olevien objektien mukaan. Eli jos käytössä olisi taustakuva, joka on suurempi kuin stage, niin se skaalautuu pienemmäksi siten, että taustakuva näkyy kokonaan. Koodin toinen rivi pakottaa sovelluksen toistumaan koko stagen kokoisena.

Ensimmäisessä pysäytysruudussa (Kuva 1) on komento *stop()*; joka pysäyttää sovelluksen toiston. Vilkkuvassa painikkeessa on seuraava koodi:

```
on(rollOver) {
        _focusRect = false;
}
on(rollOut) {
        _focusRect = true;
}
on(rollOver) {
        gotoAndStop(nextFrame());
}
```

Ensimmäiset komennot kätkevät painikkeen kohdistusneliön, joka muuten näkyisi keltaisena nelikulmiona painikkeen ympärillä over-tilassa. Viimeiset koodirivit käynnistävät elokuvan toiston ja pysäyttävät sen seuraavaan ruutuun. Samat koodirivit on myös seuraavan ruudun painikkeissa. Kohdistusneliön piilottava koodi on sijoitettu jokaiseen sovelluksessa olevaan painikkeeseen.

Koodi päävalikon painikkeissa on seuraava:

```
on(rollOver) {
        _focusRect = false;
}
on(rollOut) {
```

```

        _focusRect = true;
    }
    on(release) {
        gotoAndStop(35);
    }

```

Erona edelliseen painikkeeseen on se, että painikkeen toiminnan laukaisee vasta painikkeen nosto. ei kohdistimen tulo sen päälle. Tällöin käyttäjä ehtii nähdä, että painike reagoi sitä käytettäessä ennen kuin se laukaisee toiminnon. Viimeisen rivin koodi siirtää ja pysäyttää toiston sulkeiden sisällä olevaan ruudun numeroon. Vastaavat koodirivit on sijoitettu kaikkiin päävalikon ja alavalikoiden painikkeisiin, pois lukien videoklipin avaavat painikkeet, pää- ja alavalikkoihin palaavat painikkeet sekä edellinen- ja seuraava-painikkeet.

Videoklipin avaavan painikkeen koodi on seuraava:

```

    on(rollOver) {
        _focusRect = false;
    }
    on(rollOut) {
        _focusRect = true;
    }
    on(release){
        getURL("4_1.mp4");
    }

```

Koodin viimeisellä rivillä viitataan yleensä www-sivustoihin, mutta sillä voidaan myös avata tiedostoja puhelimen omasta muistista. koska komentoon ei ole lisätty tiedostopolkua, kyseessä on suhteellinen viittaus. Tämä tarkoittaa sitä, että ohjelma etsii lainausmerkkien sisällä olevaa tiedostoa ainoastaan samasta kansioista, josta itse flash-sovellus on avattu. Tällöin ei ole väliä, sijoitetaanko sovellus tiedostoineen puhelimen muistiin vai muistikortille.

Seuraava-painikkeissa koodi on seuraava.

```

    on(rollOver) {
        _focusRect = false;
    }
    on(rollOut) {
        _focusRect = true;
    }
    on(release) {
        gotoAndStop(nextFrame());
    }

```


Edellinen-painikkeessa koodi on aivan seuraava-painikkeessa, *nextFrame*-komennon tilalla on *prevFrame*. Tällöin täysin sama koodi voidaan sijoittaa kaikkien alavalikoiden ja yksityiskohtavalikoiden vastaaviin painikkeisiin.

Päävalikon lopeta-painikkeen koodi on seuraava:

```

on(rollOver) {
    _focusRect = false;
}
on(rollOut) {
    _focusRect = true;
}
on (release) {
    fscommand2("Quit");
}

```

Koodin viimeinen rivi katkaisee sovelluksen toiston ja sulkee sen.

6 POHDINTA

Opinnäytetyössä tutkittiin käyttöliittymän toteutusta Flash Lite-sovelluksessa. Kohteeksi valitsin juuri Flash Liten, koska sisällön tuottaminen kyseiselle ohjelmalle tapahtuu Adobe Flash-ohjelmalla. Mikäli kyseinen ohjelma on jossain määrin tuttu, perussovellusten tuottaminen sille ei vaadi paljoa kokemusta tai teknistä osaamista. Jouduin kuitenkin laajentamaan tietojani käytettävyyssasioista sekä Flash Lite Actionscriptistä, jotta sain aikaiseksi haluttua tarkoitusta palvelevan tuotteen.

Ammattiopiston toivomuksena oli saada kännykällä selattava muistilappu, jota voidaan käyttää opetuksen tukimateriaalina. Aineiston keräämisessä ja käsikirjoituksen kokoamisessa oli ongelmana se, että vaikka kaikki painotekniikan opettajat noudattavat samaa opetussuunnitelmaa, jokaisella on hieman erilainen tyyli opettaa asioita. Kaikkia säätöjä ei tehdä täysin samalla tavalla tai samassa järjestyksessä. Lisäksi jokaisella oli omat pikkukikkansa, joita varsinaisessa opetusmateriaalissa ei ollut esiteltynä. Tästä johtuen sovelluksen käsikirjoitus oli jatkuvasti elävä prosessi, joka valmistui vasta kun sovellus valmistui. Jokainen opettaja halusi tarkastaa käsikirjoituksen ja saada omat näkemyksensä esiin.

Yllättävintä oli se, miten rajoittavaksi tekijäksi käyttöliittymän suunnittelussa kohde-ryhmä muotoutui. Alkuperäisen suunnitelman mukaan halusin hyödyntää koko näytön videoklippien toistossa. Tämän vuoksi sovellus olisi pitänyt toteuttaa vaakasuuntaiseksi. Mutta selattuani kirjallisuutta, paljastui, että ominaisuus on tuettuna vasta Flash Lite 3-versiossa, joka tällä hetkellä on vielä harvinainen matkapuhelimissa. Sitä versiota ei siis voinut käyttää, koska se sulkisi lähes koko käyttäjäryhmän pois.

Mikäli klipit olisi haettu verkon kautta, käyttäjä olisi joutunut maksamaan siitä datansiirtomaksua. Lisäksi verkkoyhteys ei ole ominaisuus, joka on kaikissa puhelimissa tarjolla, joten paras ratkaisu mielestäni oli toistaa videot puhelimen omasta muistista ja unohtaa kokoruudun toistotila. Tällöin niitä ei rajoiteta puhelimen ominaisuuksilla.

Teknisenä toteutuksena klippien katselu on yksinkertaisin mahdollinen. Klippi avautuu ulkopuoliseen soittimeen, ja sovellukseen palataan sulkemalla ohjelma. Käytettävyyden kannalta tämä ei ehkä ollut paras mahdollinen ratkaisu. Uskon kuitenkin, että käyttäjä oppii tämän. Sovelluksen opittavuuskynnystä on pyritty madaltamaan antamalla muun muassa klippien käytöstä lyhyt ohjeistus.

Käytettävyydesti liittyä olennaisesti käytettävyyteen ja käyttöliittymäsuunnitteluun. Siksi on ikävä asia, ettei sitä kyetty toteuttamaan tämän työn puitteissa. Ensimmäiset kommentit sovelluksesta olivat positiivisia. Ulkonäköä pidettiin selkeänä ja hienona. Tämä ei kuitenkaan juuri kerro mitään tuotteen käytettävyydestä. Kuten aikaisemmin tässä työssä todettiin, käyttöliittymä ei ole hieno paperi, johon sovellus kääritään.

Pohjimmillaan käyttöliittymän suunnittelussa ja toteutuksessa pääsin pitkälle pelkällä maalaisjärjellä. ActionScript-koodauksessa olisi ollut paljon eri mahdollisuuksia toteuttaa asioita, joilla ei ulospäin olisi ollut merkitystä. Koodin kannalta oli tarjolla monimutkaisia ratkaisuja, joiden toteuttamisessa olisi kuitenkin saattanut tulla virheitä. Koodaus on ehdottoman tarkka kaikista merkeistä ja pitkässä koodissa on enemmän mahdollisuuksia virheille. Helpoin ratkaisu oli usein kaikkein toimivin.

Jatkotutkimuksia ja kehitysehdotuksia

Flash Lite on ylipäättään vähän tutkittu aihe. Oman työn aikana huomasin, että sen mahdollisuudet ovat äärettömän laajat erilaisille sovelluksille. Kirjallisuudessa huomasin, että Flash Liteä on jonkin verran hyödynnetty verkko-oppimisessa ja oppimateriaalin tuotannossa. Näihin asioihin kotimaassa on paneuduttu tuskin ollenkaan. Jyväskylän ammattikorkeakoulussa on mahdollisuus opiskella sekä oppimateriaalin tuotantoa sekä Adobe Flashiä, joten mahdollisuuksia tällaiselle tutkimukselle on varmasti.

Jyväskylän ammattiopiston painotekniikan henkilökunta antoi ymmärtää, että vastaavanlaiselle sovellukselle, kuten toteuttamalleni, on tarvetta todelliseen opetuskäyttöön. Ammattiopisto myy painotekniikan koulutusta muille oppilaitoksille, joilla ei ole omia resursseja toteuttaa sitä. Painokonetta tai paperileikkuria on kuitenkin mahdollonta siirtää koulutusta varten ja teorian opetuksessa olisi tärkeää saada havainnollistettua, miten laite todellisuudessa toimii. Toteutettiinpa materiaali puhelimelle tai videotykkille, prosessi on molemmille lähes sama.

LÄHTEET

Adobe. 2007. Getting Started With Flash Lite 1.x. Adoben ohjelmistojen tukisivusto. Viitattu 3.5. 2009 [Http://www.adobe.com/support/documentation](http://www.adobe.com/support/documentation).

Buyin, L., Pingchuan Z. & Qiaoling B. 2008. The Design of E-learning Platform Based on 3G Mobile Phone. Viitattu 9.5.2009. [Http://www.jamk.fi/kirjasto](http://www.jamk.fi/kirjasto), Nelli-portaali, IEEEExplore.

Bhargal, Sham. 2001. Inside Flash Actionscript. Helsinki: Edita Oyj.

Hakijan Opas 2009. 2009. Jyväskylän ammattiopisto.

Jeong-Dong L., Jongsu, L., Sang-Kyu, B. & Tai-Yoo K. 2003. Evaluation of technological innovation in the cellular phone display. Viitattu 10.5.2009. [Http://www.jamk.fi/kirjasto](http://www.jamk.fi/kirjasto), Nelli-portaali, IEEEExplore.

Kuoppala, H., Parkkinen, J., Sinkkonen, I. & Vastamäki, R. 2002. Käytettävyyden Psykologia. Helsinki: Edita Oyj.

Luukkonen, J. 2000. Digitaalisen median käsikirjoitusopas. Edita: Helsinki.

Manninen, P. & Marttila, J. 2006. Flash 8 & ActionScript. Porvoo: Docendo Finland Oy.

Metsämäki, M. 1995. Graafinen käyttöliittymä. Helsinki: Painatuskeskus Oy.

Nielesen, J. 1993. Usability Engineering. Academic Press, Inc: Lontoo.

Nokia. 2009. Nokia-matkapuhelimien valmistajan sivusto. Viitattu 29.4.2009. [Http://www.nokia.com](http://www.nokia.com)

Paananen, P. 2008. Flash-julkaisijan opas. Saarijärvi: WSOY.

Samsung. 2009. Samsung-matkapuhelimien valmistajan sivusto. Viitattu 29.4.2009 [Http://nordic.samsungmobile.com/eng](http://nordic.samsungmobile.com/eng).

Sony-Ericson. 2009. Sony-Ericson matkapuhelimien valmistajan sivusto. Viitattu 29.4.2009. [Http://www.sonyericsson.com](http://www.sonyericsson.com).

Symbian. 2009. Symbian käyttöjärjestelmän kehittäjän sivusto. Viitattu 25.4.2009. [Http://www.symbian.org](http://www.symbian.org).

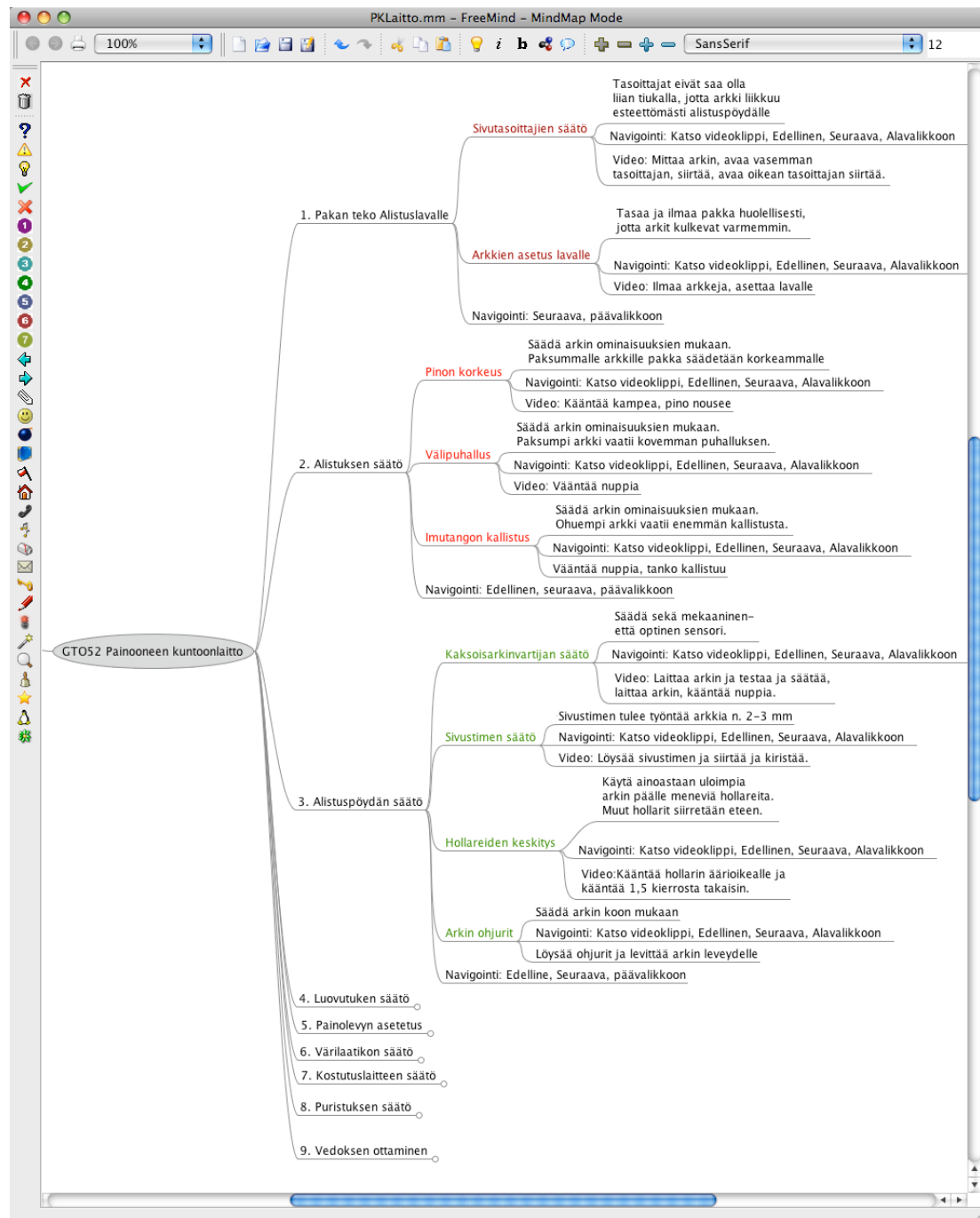
Petterson, P. 1996. Käyttöliittymän suunnitteluperiaatteet. Teoksessa Graafisen käyttöliittymän suunnittelu, Opas ohjelmistojen käytettävyyteen. Toimittanut A. Kalimo, Espoo: Suomen ATK-kustannus Oy.

Wiio, A. 2004. Käyttäjäystävällisen sovelluksen suunnittelu. Helsinki: Edita Publishing Oy.

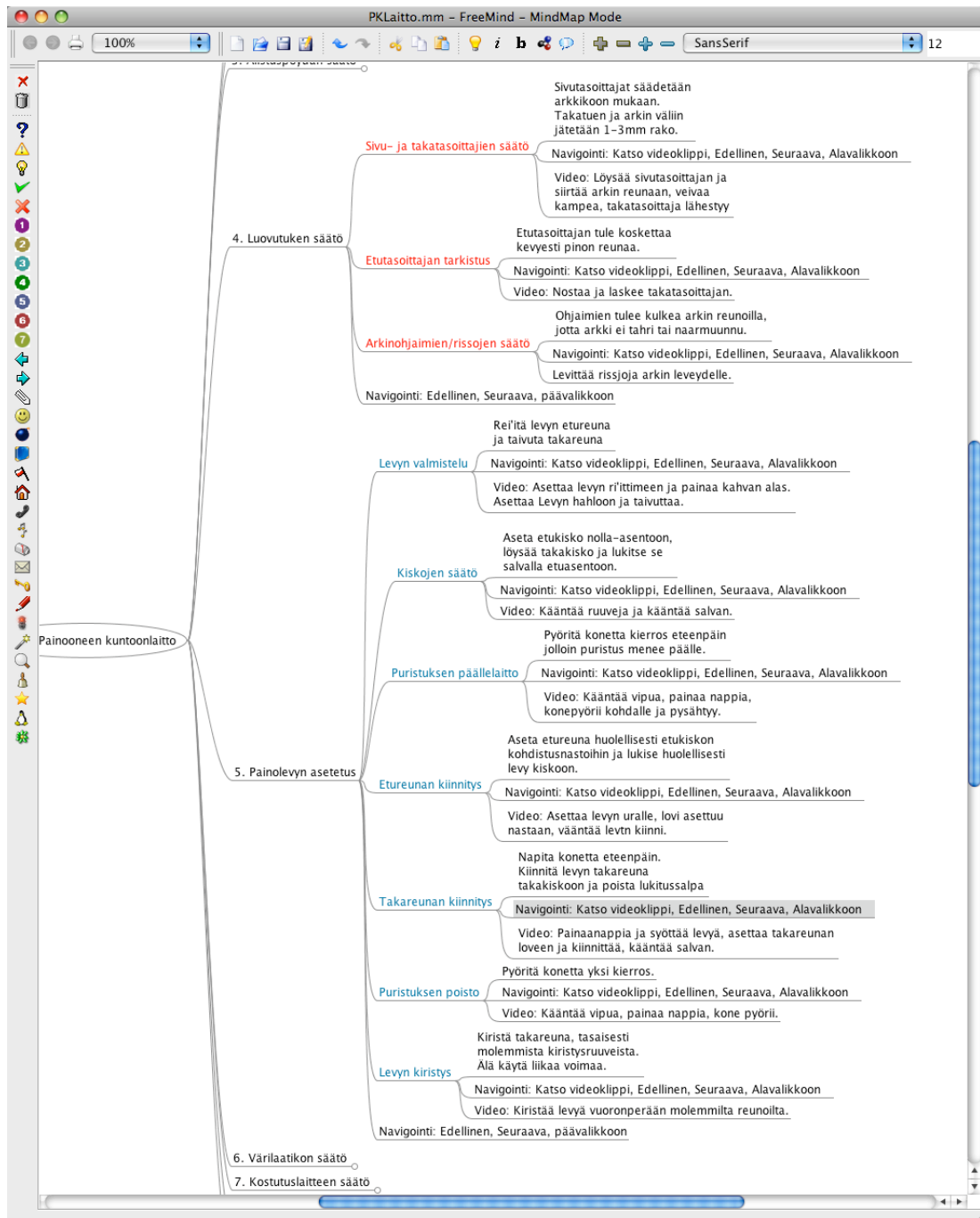
Wiio, A., Käyttäjäystävällisen sovelluksen suunnittelu. 2007. Helsinki: Tecnologos Oy. Viitattu 2.5.2009. [Http://www.technologos.fi/kirja.htm](http://www.technologos.fi/kirja.htm).

LITTEET

Liite 1: Sovelluksen käsikirjoitus



Liite 1 jatkuu



Liite 1 jatkuu

