

Tommi Holtinkoski

**MAAHAN KÄTKETYT: INTERAKTIIVINEN MULTIMEDIAESITYS RAAHEN
MUSEOLLE**

MAAHAN KÄTKETYT: INTERAKTIIVINEN MULTIMEDIAESITYS RAAHEN MUSEOLLE

Tommi Holtinkoski
Opinnäytetyö
Kevät 2015
Tietotekniikan koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan koulutusohjelma

Tekijä: Tommi Holtinkoski
Opinnäytetyön nimi: Maahan kätkeyt: Interaktiivinen multimediaesitys Raahen museolle
Työn ohjaaja: Eino Niemi
Työn valmistuslukukausi- ja vuosi: Kevät 2015
Sivumäärä: 73 + 0

Insinööriyön tarkoituksena oli käyttöliittymäprototyyppiä jatkokehittäen toteuttaa loppukäyttäjävalmis multimediaesitys Raahen museolle. Esitys käsitteli paikallisia arkeologisia löydöksiä, Tervakankaan rautakautista kalmistoa ja Pattijoelta löydettyä keskiaikaista rahakätköä. Työn tärkein tavoite oli toteuttaa täysin käyttö- ja esitysvalmis käyttöliittymä ja ohjelmisto.

Prototyyppissä käyttäjälle näkyvä ohjelma oli visuaalisen ilmeen ja -suunnittelun osalta joitakin muutoksia ja lisätarpeita lukuun ottamatta valmis. Ohjelman toiminnassa ja toimintavarmuudessa oli vielä hiottavaa. Käyttöliittymänä oli VTT:n suunnittelema ja toteuttama kapasitiivinen sensorikalvo, jonka kautta käyttäjä pystyi liikkumaan esityksessä käsieleiden avulla. Näkyvä ohjelma oli suunniteltu videoprojektorilla heijastettavaksi sensorikalvon päälle asetellulle valkokankaalle. Ohjelmisto oli kirjoitettu C++:lla, C#:lla ja alustana oli käytetty MFC:a, .NET Frameworkiin kuuluvaa WinForms -luokkakirjastoa.

Lopputuote muuttui käytännön testeissä todettujen ongelmien jälkeen radikaalisti. Sensorikalvo-videoprojektoriyhdistelmä vaihdettiin kosketusnäyttöön ja vahvasti sensorikalvon käyttöön pohjautunut ohjelmakoodi hylättiin. Uudessa alustavassa ohjelmointisuunnitelmassa voitiin tällöin ottaa paremmin huomioon työhön jo aiemmin halutut lisäykset, joihin kuului muun muassa animaatioita. Tämä huomioon ottaen käyttöliittymän toteutustavaksi valittiin WPF ja XAML näiden mukanaan tuoman helppokäyttöisen graafisen rajapinnan vuoksi, sekä kieleksi C#.

Työssä toteutettiin Raahen museolle käyttöön tullut kosketusnäyttöpohjainen multimediaesitys. Työn mahdollisimman helppoa muokattavuutta tavoiteltiin niin, että asiakas voi halutessaan muokata nykyisiä ääniä, taustakuvia, kuvatekstejä ja/tai opastustekstejä ohjelmakoodiin koskematta tai sitä tuntematta. Pohjakoodia voidaan hyödyntää museon mahdollisissa muissa vastaavanlaisissa projekteissa.

Asiasanat: C#, .NET, WPF, XAML, Visual Studio, Photoshop, Audacity, Creative Commons, käyttöliittymäsuunnittelu, käyttöliittymäohjelmointi, käyttäjäkokemus

ABSTRACT

Oulu University of Applied Sciences
Degree programme in Information Technology

Author: Tommi Holtinkoski

Title of thesis: Hidden in the ground: An interactive multimedia presentation for Raahe Museum

Supervisor: Eino Niemi

Term and year when the thesis was submitted: Spring 2015

Number of pages: 73 + 0

The objective of this thesis was to finish development on a multimedia presentation UI prototype designed for Raahe Museum. The presentation dealt with local archaeological findings, an Iron Age burial ground discovered in Tervakangas and a medieval coin hoard unearthed in Pattijoki. The main goal was to develop an end-user ready, fully functional and operational UI and software, which were to be put on display at Raahe Museum.

The visual image and design of the prototype were ready, excluding some required changes and enhancements. The program itself still had some bugs and reliability issues. A capacitive sensor board designed and manufactured by VTT was the UI, through which the user could control the flow of the presentation with hand gestures. The software was to be projected onto a projector screen covering the sensor board. The source codes were written with C++, C# and the selected platform was MFC, a .NET Framework-based WinForms.

The problems realized during practical testing ended up changing the end-product dramatically: The sensor board-video projector combination was replaced by a touch screen and the source codes heavily reliant on sensor board usage were abandoned. However, during the planning phase of the new program it was now possible to consider the possible needs of desired additions and improvements — such as animations — before deciding which programming language and platform to use. The coming changes taken into account WPF with XAML and C# were chosen, as WPF brings with it an easy-to-use graphical interface.

In this thesis the set goals of developing a multimedia presentation UI usable by the patrons of Raahe Museum were met. Customizability was to be as easy as possible, which has been pursued by making the personnel of the museum able to change sounds, background images, captions and/or other informational text without actual programming knowledge. The base code itself can be reused or otherwise exploited in other similar projects that the museum may have.

Keywords: C#, .NET, WPF, XAML, Visual Studio, Photoshop, Audacity, Creative Commons, UI design, UI programming, UX

SISÄLLYS

TIIVISTELMÄ.....	3
ABSTRACT.....	4
SISÄLLYS.....	5
LYHENTEET.....	7
1 JOHDANTO.....	8
2 PROTOTYYPPI JA LÄHTÖTILANNE.....	9
2.1 Kapasitiivinen sensorikalvo.....	9
2.2 Mikrokontrolleri.....	10
2.3 Graafinen käyttöliittymä.....	11
2.4 Testaus käytännössä.....	12
2.5 Prototyypin hylkäämisperusteet.....	13
2.5.1 Kuvien ja sensorikalvon yhteensovittamisen ongelmat.....	13
2.5.2 Sensorikalvon käytön ja herkkyyden ongelmat.....	14
2.5.3 Videoprojektorin käytön ongelmat.....	15
3 UUDEN PROJEKTIN ALOITUS.....	16
3.1 Käyttöjärjestelmä ja tietokone.....	16
3.2 Kosketusnäyttö.....	17
3.3 Windows Presentation Foundation.....	18
3.4 Extensible Application Markup Language.....	18
3.5 Visual Basic .NET.....	19
3.6 C# (C sharp).....	20
3.7 Microsoft Visual Studio 2012.....	20
3.8 Kuvaeditointiohjelmisto, äänieditointiohjelmisto.....	21
3.9 Creative Commons.....	22
4 KUVAT, ÄÄNET JA EDITOINNIT.....	24
4.1 Alkuperäiset kuvat ja ulkonäkö.....	24
4.1.1 Tervakankaan rautakautinen kalmisto.....	25
4.1.2 Tervakankaan rautakautinen kalmisto, kaivausosio.....	28
4.1.3 Pattijoen keskiaikainen rahakätkö.....	31
4.1.4 Apukuvien etsintä ja käytettäviksi päätyneet kuvat.....	34
4.2 Äänet ja äänieditointi.....	36

5	VERSIO 0.9.7.1 ALPHA	38
5.1	Version 0.9.7.1 ulkonäkö	38
5.1.1	Tervakankaan rautakautinen kalmisto.....	38
5.1.2	Pattijoen keskiaikainen rahakätkö	41
5.2	Version 0.9.7.1 ohjelmointi	44
5.2.1	Ohjelmakoodin esittelyä	45
5.2.2	MainWindow.xaml, MainWindow.xaml.cs	46
5.2.3	MainMenuPage.xaml, MainMenuPage.xaml.cs	46
5.2.4	BurialGroundMainPage.xaml, BurialGroundMainPage.xaml.cs	49
5.2.5	GraveDiggingIntroPage.xaml, GraveDiggingIntroPage.xaml.cs.....	50
5.2.6	GraveMainPage.xaml, GraveMainPage.xaml.cs ja rahakätköosio.....	52
6	KORJAUKSET, LISÄYKSET JA MUUTOKSET	53
6.1	Ohjelmistosta löydetyt viat ja niiden korjaaminen	53
6.2	Toivotut muutokset, lisäykset tai lisätoiminnot.....	56
6.3	Tehdyt muutokset toimintoihin	58
7	VERSIO 1.1.0.3 RELEASE CANDIDATE	67
7.1	Version 1.1.0.3 ulkonäkö	67
7.2	Tervakankaan rautakautinen kalmisto.....	68
7.3	Pattijoen keskiaikainen rahakätkö	70
8	YHTEENVETO	71
	LÄHTEET.....	72

LYHENTEET

API	Application Programming Interface, ohjelmointirajapinta
ASCII	American Standard Code for Information Interchange, ASCII-merkistö
BASIC	Beginner's All-purpose Symbolic Instruction Code, ohjelmointikieli
C	Yleiskäyttöinen, imperatiivinen ohjelmointikieli
C++	Yleiskäyttöinen oliopohjainen ohjelmointikieli
C#, C sharp	Moniparadigmmainen ohjelmointikieli
F#, F sharp	Vahvasti tyyteltä, moniparadigmmainen ohjelmointikieli
CC	Creative Commons
Direct3D	Graafinen ohjelmointirajapinta
DirectX	Ohjelmointirajapintakokoelma
GPU	Graphics Processing Unit, grafiikkaprosessori
GUI	Graphical User Interface, graafinen käyttöliittymä
IDE	Integrated Development Environment, ohjelmointiympäristö
IntelliSense	Intelligent code completion, älykäs koodinpäätely
JPEG, JPG	Joint Photographic Experts Group, häviöllistä pakkausta käyttävä bittigrafiikan tallennusformaatti
MFC	Microsoft Foundation Classes, C++ -luokkaohjelmointirajapintakirjasto
PC	Personal Computer, tietokone
PNG	Portable Network Graphics, häviötön bittikarttagrafiikan tallennusformaatti
SaaS	Software as a Service, ohjelmiston hankinta palveluna
UI	User Interface, käyttöliittymä
USB	Universal Serial Bus, oheislaitesarjaväyläarkkitehtuuri
VB, VB.NET	Visual Basic, Visual Basic .NET, BASIC-sukuinen yleiskäyttöinen ohjelmointikieli
WP	Windows Phone, mobiilikäyttöjärjestelmä
WPF	Windows Presentation Foundation, graafinen käyttöliittymärajapinta
XML	Extensible Markup Language, dokumenttien tallennus- ja tiedonvälitysformaatti
XAML	Extensible Application Markup Language, deklarativinen XML-pohjainen ohjelmointikieli

1 JOHDANTO

Tämän työn prototyyppi sai alkunsa keväällä 2013 kahden OAMK:n tietotekniikan koulutusohjelman opiskelijan opinnäytetöissä, KC Srijanan ”A Study of a Non-touchable and User Friendly Interface for Raahe Museum” ja Sun Shuon ”User Interface of Raahe Museum”. Töiden tuloksena suurilta osin valmiina olivat prototyypit sekä fyysisestä että graafisesta käyttöliittymästä.

Tämän työn tekijä kiinnostui opinnäytetyön asettamista haasteista. Käyttöön oli suunniteltu WPF, XAML ja C#, jotka olivat tekijälle kielellisesti ja tyyllisesti tuttuja Windows Phone -ohjelmistokehityksen kautta. Houkutusta lisäsi myös oma mielenkiinto tutkia, onko Creative Commons -lisenssien alla tehtyjen ääniefektien etsintä ja löytäminen helppoa, sekä onko lisenssien ehdot ja määrytykset huomioon ottaen mahdollista käyttää kyseisiä efektejä tämänkaltaisessa työssä. Niin ikään työn lievä pelimäisyys kiinnosti mahdollisen uravalinnan suhteen hyvinkin paljon.

Lisääntyvä interaktiivisuuden toive koskee yhä useampaa erilaista opetus- ja oppimisympäristöä. Erityisesti lasten kiinnostus käsiteltävää aihetta kohtaan kasvaa, kun opetustilanteessa saa itse olla aktiivisena ja toiminnallisena. Pelien ja leikkien kautta tapahtuva oppiminen on joillekin lapsille tavanomaisen pöytätyön sijaan tavoitteellisempaa – halutaan itse edetä pelissä, jota vastaan ollaan valmiita tekemään töitä. Palkinnoksi voi riittää tehtävässä onnistumisen tuoma edistys. Pelien kautta oppiminen ei ole vierasta nuorille tai aikuisillekaan; shakkia on käytetty sotastrategioiden opetustyökaluna jo keskiajalta lähtien (Wikipedia 2015e, viitattu 17.3.2015).

2 PROTOTYYPPI JA LÄHTÖTILANNE

Prototyypin fyysinen käyttöliittymä oli tehty mikrokontrolleriohjatusta sensorikalvosta, johon WinForms-pohjainen visuaalinen käyttöliittymä yhdistettiin projektoimalla se sensorikalvon päällä olevalle projektiokankaalle. Tässä luvussa prototyyppi käsitellään pääpiirteittäin, selostetaan prototyypin testausvaiheet ja -tapa sekä perustellaan prototyypin käyttöliittymän toteutustavan hylkäämiseen johtaneet seikat.

2.1 Kapasitiivinen sensorikalvo

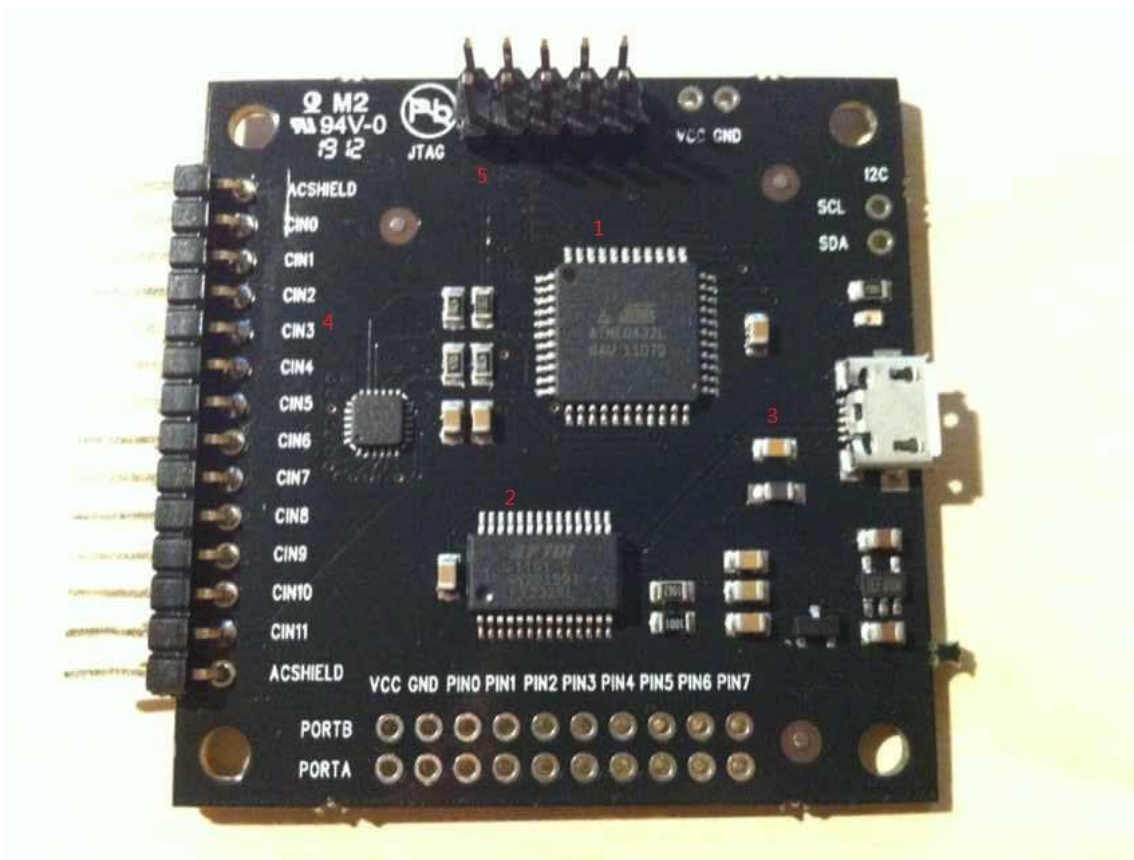
Sensorikalvo (kuvio 1) on taipuisa, kaksikerroksinen kalvo, jonka pinta-alalle (60 cm x 60 cm) on mahduttu 36 kosketuspistettä. Kosketuspisteet toimivat kapasitanssimuutoksia seuraamalla, joita käden tai muun sähköä johtavan kappaleen tuominen pisteiden alueelle aiheuttaa. Käden havaitsemisetäisyys riippuu ympäristöstä ja laitteen maadoituksesta, mutta on yleensä viidestä kymmeneen senttimetriä. Kosketusetäisyyttä voidaan muuttaa mikrokontrollerin kosketusrekisteröintiherkkyyttä säätämällä.



KUVIO 1. Kapasitiivinen sensorikalvo (Sun Shuo 2013, 11)

2.2 Mikrokontrolleri

Mikrokontrollerilla (kuvio 2) on 12 sisääntuloa (CIN0 – CIN11, kuvio 2 merkintä 4), joihin sensorikalvon kosketuspisteet on väylätty 6 x 6 matriisissa. Jokaisen väylän muuttuvaa kapasitanssia seuraa Analog Devicesin kapasitanssi-digitaalimuunnin, AD7147-piiri (kuvio 2 merkintä 2). 8-bittinen Atmelin ATmega1284P:a (kuvio 2 merkintä 1) käytetään dataprosessointiin. Se kuuntelee ASCII-muotoisia sarjaporttisiignealeja, lukee AD7147:n lähettämän datan ja hoitaa tarvittavat kalibroinnit ja dataprosessoinnin. Tietokoneen ja mikrokontrollerin välillä käytetään USB-kaapelia ja FT232R-ajuria, joka muuntaa sarjaporttidatan USB-formaattiin ja päin vastoin.



KUVIO 2. Sensorikalvon mikrokontrolleri (Sun Shuo 2013, 10)

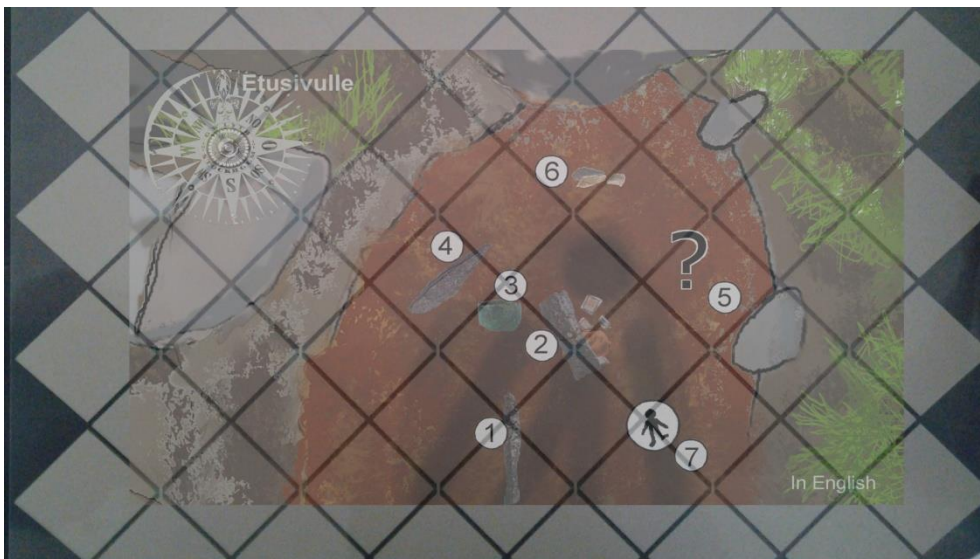
AD7147:ssa on sisäinen algoritmi, joka kompensoi ympäristön kapasitanssimuutoksia, mutta tässä sovelluksessa algoritmiä ei voida käyttää, koska AD7147:n kautta vain luetaan raakakapasitanssidataa mikrokontrollerin käsiteltäväksi. Sen sijaan käytetään käyttäjän määrittämää kynnyksiarvoa, jonka avulla päätellään onko kyseessä kosketus. Kontrolleri ei tue monikosketusta.

2.3 Graafinen käyttöliittymä

Graafisen käyttöliittymän pohjasuunnitelma oli muutamia pakollisia muutoksia ja haluttuja lisäominaisuuksia lukuun ottamatta valmis. Käyttöliittymä koostui kuvista, joihin oli piirretty halutut toiminta-alueet, kuten paluupisteet pääsivulle, kielenvaihtopisteet ja lisätietoa tarjoavat numerosymbolit erillisine pisteineen. Fyysiseen käyttöliittymään yhdistyminen tapahtui siis pistealueittain. Tietty sensorikalvon pisteet aktivoivat tietyn, halutun tapahtuman.



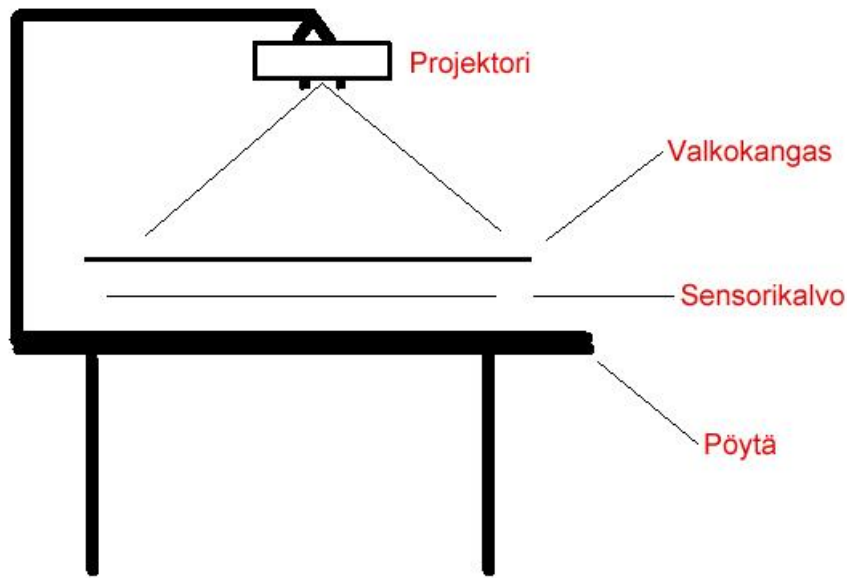
KUVIO 3. Tervakankaan rautakautinen kalmisto sovitettuna sensorikalvon päälle



KUVIO 4. Tervakankaan kalmiston haudankaivuuosio

2.4 Testaus käytännössä

Lopullinen esityksen asettelu- ja asennustapa oli suunniteltu kuvion 5 mukaisesti. Projektori asennettaisiin sensorikalvon ja valkokankaan yläpuolelle, jolle graafinen käyttöliittymä heijastettaisiin. Työlle rakennettaisiin joko kuvion mukainen teline tai projektori kiinnitettäisiin esimerkiksi kattoon.



KUVIO 5. Esimerkki 3: Esitystelinehahmotelma (Sun Shuo 2013, 8)

Testatessa suunnitelmaa projektori asetettiin vaakatasoon pöydälle ja sensorikalvo kiinnitettiin seinälle. Valkokankaana käytettiin tynnyliinaa. Graafinen käyttöliittymä sovitettiin oikeaan, tulevaan kokoonsa kankaalle. Ohjelman käyttö tapahtui suunnitelman mukaisesti käsielein.

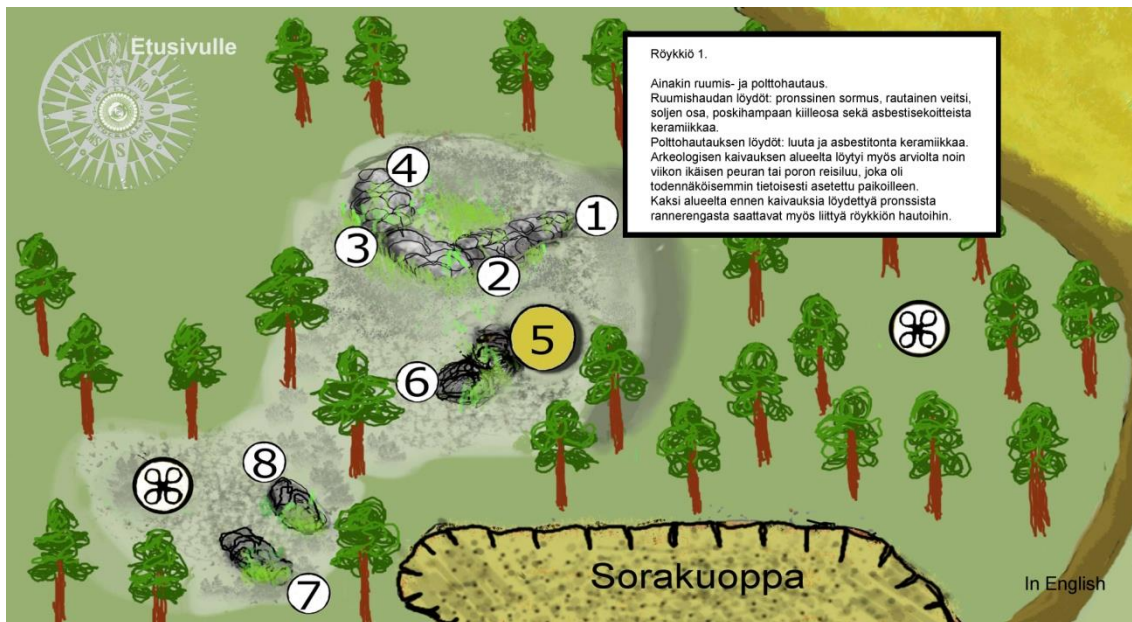


KUVIO 6. Projektorin, sensorikalvon ja valkokankaan testiasettelu

2.5 Prototyypin hylkäämisperusteet

2.5.1 Kuvien ja sensorikalvon yhteensovittamisen ongelmat

Kuvia 3. ja 4. vertailemalla ja tutkimalla pakolliset muutokset näkyvät ilmeisinä: Etusivulle-kompassikuviota ja numerosymbolien paikkoja sensorikalvon kosketuspisteisiin (vain risteämispisteet on mahdollista erotella tarkasti käytettäväksi) vertaillen huomataan, että joitakin satunnaistapauksia lukuun ottamatta lähes kaikki numerosymbolit ovat väärissä kohdissa. Lisäksi Etusivulle-kompassikuvion koko muuttuu eri osioiden välillä, jolloin myös toimintapisteiden tulisi muuttua tai kuvia täytyisi muokata. Kuviossa 4 kompassikuvio on lähempänä oikeaa kokoa, sillä sen tulisi kattaa neljän (2x2) toimintapisteiden alue. Aiemmin mainittu suoraan kuviin piirretty graafinen käyttöliittymä ja sen käyttämät elementit aiheuttaa sen, ettei yksittäisiä elementtejä (esim. numerosymbolit) voi siirtää muokkaamatta kaikkia kuvia, joissa kyseiset symbolit esiintyvät. Esimerkiksi kuviossa 3 esiintyvän Tervakankaan rautakautisen kalmiston tapauksessa tämä olisi tarkoittanut 11 kuvion täysimittaista uusintaa, koska jokaisen numerolinkin painallus avaa pelkän tekstilaatikon sijasta uuden kuvion, joissa linkit myös näkyvät (kuvio 7).



KUVIO 7. Esimerkki 3: Tervakankaan rautakautisen kalmiston linkki numerolla 1. avattuna

Kuviossa 6 huomataan, ettei koko sensorikalvo voi olla käytössä, koska graafinen käyttöliittymä on suunniteltu kuvasuhteella 16:9 käytettäväksi ja sensorikalvon toimintapisteet on aseteltu suhteella 1:1. Sensorikalvon 36 mahdollisesta toimintapisteestä jää siis asettelusta riippumatta

kaksi riviä (12 pistettä) käyttämättä. Jo tämä vähentää huomattavasti käytettävissä olevien pisteiden määrää. Sen lisäksi kuviossa 4 näkyvän Etusivulle-kompassikuvion tulisi ohjelmanlaajuisesti viedä näistä neljä toimintapistettä jokaisessa valikossa, sekä kielenvaihtoalueen vähintään yksi.

Kuvista nähdään myös, että suunnitelmassa numerosymbolien paikat ovat määritelty piirrosten mukaan mahdollisimman lähelle aitoa, tapahtunutta kaivausta kuvaavaksi. Symboleita ei siis voida siirtää sensorikalvon ristikon yhtymäkohtiin miten halutaan, vaan sen sijaan sensorikalvon täytyisi käytännössä toimiakseen olla mukautettavissa halutusti.

2.5.2 Sensorikalvon käytön ja herkkyiden ongelmat

Sensorikalvo on äärimmäisen herkkä staattiselle sähkölle. Kuvioista 6 ja 8 nähdään, kuinka kalvon taakse on täytynyt asentaa muun muassa sanomalehtiä eristeeksi, jotta staattiset sähkönpurkaukset eivät aiheuttaisi virhepainalluksia. Tästä huolimatta sensorikalvo vaati uudelleenkalibrointia todella usein, jotta purkaukset eivät rekisteröityisi liikkeiksi. Käyttäjä pystyi myös itsessään olevalla staattisella varauksella sekoittamaan matriisin sensoriherkkyiden, jolloin kalvo täytyi taas kalibroida. Kalvon kosketusherkkyttä säätämällä asiaa ei saatu korjattua ainakaan niin, että kosketukset olisivat vielä toimineet.



KUVIO 8. Suurenno sensorikalvon testiasettelusta

Herkkyden säädössä ongelmia toivat myös matriisin toimintapisteiden välinen etäisyys toisistaan, joka on noin 5 cm. Liikkeen rekisteröintiherkkyttä nostettaessa käyttäjän tuodessa koko kätensä kalvon alueelle on ohjelmallisesti hankala arvioida, mikä piste on tarkoitettu aktivoitavaksi. Tällöin useita pisteitä saatetaan aktivoida yhtäaikaisesti tai peräkkäin. Myöskään rekisteröintiherkkyden lasku ei tilannetta korjannut, sillä sitä ei voida laskea loputtomiin: joka tapauksessa käden tuominen alueelle aiheuttaa virhetulkintoja.

Sensorikalvon verrattain korkea hinta oli myös yksi hylkäyskriteereistä käytön hankaluuden lisäksi. Kyseinen kalvo olisi tullut maksamaan noin 3000 €, kun kosketusnäytön hinnaksi tuli lopulta noin 750 €.

2.5.3 Videoprojektorin käytön ongelmat

Projektorin käytön ongelmista ilmeisin on sen asettelu. Kuviossa 5 nähtävää esimerkkijärjestelyä apuna käyttäen voidaan kuvitella käyttäjän käsi projektorin valokeilan ja valkokankaan väliin. Tällöin valokeila heijastuisi valkokankaan sijasta käyttäjän kädestä, jolloin käyttäjä ei enää kätensä aiheuttaman varjon takia tietäisi, mitä kohtaa hänen tulisi painaa. Tähän ongelmaan ainoa ratkaisu olisi, että projektori sijoitettaisiin pöydän yläpuolen sijasta pöydän alle ja kuva heijastettaisiin normaalin sijasta peilikuvana. Tämä ei kuitenkaan ole mahdollista, koska itse sensorikalvo ei ole läpinäkyvä.

Myös muita huomioitavia ongelmia projektorin käytölle löydettiin, kuten itse projektorin äänekkyys muuten kohtalaisen hiljaisessa tilassa ja projektorin tuottama hukkalämpö sekä virrankulutus. Lisäksi projektorin käyttö sekä pöydän päälle että alle asennettaessa olisi täytynyt olla millintarkkaa, sillä pienikin virhe olisi kertautunut lyhyelläkin heijastusmatkalla moninkertaisesti. Samasta syystä projektori ja koko esitys olisi täytynyt suojata äärimmäisen hyvin kaikenlaisia tönäyksiä vastaan, jotta ne eivät olisi pystyneet liikauttamaan pöytää, projektoria, valkokangasta tai sensorikalvoa.

3 UUDEN PROJEKTIN ALOITUS

Ensikosketus projektin ohjelmoinnissa pohjasi prototyypin ohjelmoijan koodiin. Ennen kuin prototyypin hylkäyspäätös tehtiin, ehdittiin ohjelmaa joiltakin osin parannella ja joitakin haluttuja lisätoimintoja saatiin lisättyä. Tätä tehdessä huomioitiin samalla yksi tavanomaisimmista ohjelmointia tai erityisesti sen jatkamista vaikeuttavista seikoista: jos alkuperäinen ohjelmoija ei ole mukana projektissa, elää projektin jatkaja alkuperäisohjelmoijan dokumentoinnin varassa. Ja kun virallista dokumentointia harvoin tämänkaltaisissa projekteissa on, käytännössä ainoat avut nykyisen koodin ymmärtämiseen ovat edellisen ohjelmoijan koodiin sisällytetty kommentointi ja/tai - pelkkää vaihtoehtoisuutta voimakkaasti painottaen - todella hyvin suunniteltu koodin luettavuus. Vaikkakin luettavuuteen oli panostettu nimeämällä käytetyt muuttujat ja metodit selittävästi, kommentoinnit olivat jääneet vähemmälle. Metodien käyttötarkoitus (ja/tai tieto siitä, mitkä muut metodit käyttävät kyseistä metodia) on aina erityisen tärkeätä tietoa, jotta osataan ennakoida muiden metodien toiminnanvarmistus, jos tarkastelussa olevaa metodia halutaan muokata.

Lopullinen syy täysin puhtaalta pöydältä aloittamiseen oli sensorikalvo-projektoriyhdistelmän päätyminen hylättäväksi, jonka käyttöön nykyinen ohjelmisto ymmärrettävästi pohjasi toiminnallisuutensa täysin. Prototyypille suunnitellun ohjelman uudelleenohjelmointi kosketusnäytöllä toimivaksi ei tuntunut enää järkevältä.

Tässä luvussa esitellään lyhyesti käytettyjä ohjelmointikieliä ja ohjelmointityökaluja sekä muita projektin toteuttamisen vaatimia työkaluja tai tarvikkeita. Kappaleesta 4 eteenpäin käydään läpi esityksen ensimmäisen käytännön testaukseen asti edenneen version ohjelmakoodia, ohjelmointitapoja, kuvia ja kuvaeditointia sekä ääniä ja äänieditointia.

3.1 Käyttöjärjestelmä ja tietokone

Museon henkilökunnan tottuneisuus Windows-käyttöjärjestelmään oli tärkein yksittäinen syy valintoihin. Tarkoituksena oli tehdä ohjelmasta mahdollisimman helposti muokattava ilman ohjelmointitaitoja, joten myös käyttöjärjestelmän tiedostojärjestelmän oli oltava mahdollisimman tuttu. Esimerkiksi muuten halvempi, Linux-pohjainen käyttöjärjestelmä sulkeutui näin ollen pois vaihtoehtoista. Projektin budjettia ei kuitenkaan tarvinnut korottaa ostoksilla tai lisenssimaksuilla,

sillä museolla oli myös vielä käyttötarkoitusta vaille olevia ja käytöstä poistettuja Windows-pöytä tietokoneita, jotka soveltuivat tehoiltaan tämänkaltaiseen projektiin hyvin.

3.2 Kosketusnäyttö

Sensorikalvon hylkäykseen johtaneessa kokouksessa käsiteltiin myös tulevan kosketusnäytön hankintaa. Toivomuksena oli löytää sopuhintainen, mielellään mahdollisimman suuri, integroiduin kaiuttimin sekä kuulokeliitäntämahdollisuudella varustettu näyttö, joka kykenisi kymmenen kosketuspisteen yhtäaikaiseen tulkintaan. Museolla oli jo useita kosketusnäyttötoimisia, niin sanotuin all-in-one-tietokonein toteutettuja multimediaesityksiä, mutta toisin kuin pöytä tietokoneiden kohdalla nämä olivat kaikki jo käytössä. All-in-one-koneet olivat kyllä tarkoitukseen hyviä, mutta työn tekijän mielestä muihin mahdollisuuksiin nähden kalliita. Sovittiin, että tekijä etsii toiveet täyttävän näytön ja esittelee ehdotuksensa museon henkilökunnalle, joka tekee lopullisen päätöksen hankinnasta.

Etsinnän jälkeen ehdotuksena museon henkilökunnalle lähetettiin tiedot ViewSonicin 23-tuumaisesta (mallimerkintä TD2340) kosketusnäytöstä, joka oli tekijän näkökulmasta työhön hyvinkin sopiva. Museon henkilökunta hyväksyi ehdotuksen, haluten vain suuremman, 27-tuumaisen (mallimerkintä TD2740) version päätteestä.



Tekniset tiedot:

Koko: 23"
Kuvasuhde: 16:9
Paneelin tyyppi: IPS, anti-glare, 7H-kovapinnoite
Taustavalaisu: WLED
Resoluutio: 1920 x 1080
Kontrasti: 1000:1 (tyypillinen), 20 000 000:1 (dynaaminen)
Kirkkaus: 250 cd/m²
Vasteaika: 7ms (tyypillinen)
Katselukulma: 178 vertikaalinen, 178 horisontaalinen
Värisvyvyys: 16,7 miljoonaa väriä

Liitännät:

- 1 x HDMI v1.4
- 1 x DisplayPort
- 1 x VGA
- 1 x USB upstream
- 2 x USB downstream
- audiosisääntulo (3,5mm)
- kuulokeliitäntä (3,5mm)

Kaiuttimet: 2 x 2W

Ergonomia: pivot, korkeussäätö, kallistus jopa vaakatasoon

Mitat (jalustalla, KxLxS): 454 x 577 x 325 mm

Paino: 9,2 kg

Myyntipakkaus sisältää:

- Monitorin jalustoineen
- virtakaapelin
- äänikaapelin
- VGA-, HDMI- ja DP-kaapelit
- USB-kaapelin
- pikaoppaan

KUVIO 9. ViewSonic TD2340 ja laitteen tekniset tiedot (Jimm's PC-Store Oy 2014, viitattu 22.9.2014)

3.3 Windows Presentation Foundation

Windows Presentation Foundationia käytettäessä mahdollisia ohjelmointikieliä ovat XAML, C# ja Visual Basic .NET. WPF kirjastoineen soveltuu valmiiden, helppokäyttöisten audiovisuaalisten rajapintojensa vuoksi hyvin tämänkaltaisiin visuaalisia tehosteita vaativiin projekteihin. Renderöinti tapahtuu Direct3D/DirectX-rajapintaa käyttäen, jolloin tarvittavia grafiikan piirtotehtäviä voidaan jakaa erillisen, DirectX-tuetun grafiikkaprosessorin käsiteltäväksi. Toiminnallisena etuna vektorigrafikoilla on matemaattisiin funktioihin ja koordinaatistoon sidottuihin objekteihin perustuva skaalaus, jolloin vektorein piirretyn kuvan muotoa voidaan muuttaa ilman, että kuva pikselöityy. (Wikipedia 2015p, viitattu 15.2.2015; Wikipedia 2015i, viitattu 1.5.2015; Wikipedia 2015j, viitattu 4.5.2015.)



KUVIO 10. Vektorikuvan vertailu bittikarttakuvaan (Wikipedia 2015m, viitattu 22.2.2015)

Suuriin, 3D-grafiikan jatkuvaan käyttöön pohjautuviin sovelluksiin WPF:n valmiit animointimahdollisuudet ovat kuitenkin turhan raskaita, sillä vektorikuvien skaalaus tarvitsee hirvittävän määrän laskentatehoa. (Wikipedia 2015m, viitattu 22.2.2015.)

3.4 Extensible Application Markup Language

XAML on niin kutsuttu deklarativinen ohjelmointikieli, joka ei toiminnaltaan perustu resurssien mahdollisimman tehokkaaseen hyödyntämiseen, vaan enemmän ohjelmoijan työn

helpottamiseen. Sen sijaan, että ohjelmoija käskyttää käytettävät metodit, muuttujat ja muut tapahtuman suorittamiseen tarvittavat määreet, jätetään halutun tapahtuman toteutus kielen hallittavaksi. (Wikipedia 2015h, viitattu 19.4.2015.)

Esimerkkikuviossa 11 voidaan huomata kielen deklarativinen luonne. Koodissa tapahtuu seuraavaa: kun sivu avataan laukaistaan animaatio, joka muuttaa sivun läpikuultavuuden näkymättömästä näkyväksi sekunnin sisällä.

```
<!--When page is loaded image is eased into view by manipulating opacity-->
<Page.Triggers>
  <EventTrigger RoutedEvent="Page.Loaded" x:Name="OnPageLoadAnimation">
    <BeginStoryboard>
      <Storyboard>
        <DoubleAnimation Storyboard.TargetProperty="Opacity" Duration="0:0:1" From="0" To="1">
        </DoubleAnimation>
      </Storyboard>
    </BeginStoryboard>
  </EventTrigger>
</Page.Triggers>
```

KUVIO 11. XAML-koodiesimerkki

3.5 Visual Basic .NET

Visual Basic .NET ulottuu alkuperältään kauas, vuoteen 1964 asti ja pohjautuu helppokäyttöisyysperiaatteeltaan ja syntaksiltaan tällöin ohjelmoinnin alkeiden opetukseen kehitettyyn BASIC-kieleen. Huippusuosion alkuperäinen Visual Basic saavutti 1990-luvulla, ja pysyi suosittuna vielä 2000-luvullakin. Versio 1.0 Visual Basicista julkaistiin 1991 ja viimeinen versio 6.0 julkaistiin 1998. Virallinen tuki lopetettiin 2008. (Wikipedia 2015n, viitattu 9.5.2015; Wikipedia 2015d, viitattu 2.2.2015; Wikipedia 2013l, viitattu 20.3.2015.)

Nykyinen Visual Basic .NET on Microsoftin nimeämä Visual Basicin manttelinperijä ja on osa .Net Framework -ohjelmistokomponenttikirjastoja. Ensimmäinen .NET-versio julkaistiin vuonna 2002. Virallisesti .NET-päätteen jätettiin pois nimestä jo 2005, mutta kielen erottelun vuoksi sitä on tässä käytetty.

Ominaisuuksiltaan VB.NET on korkean tason oliopohjainen kieli ja hyvin läheistä sukua C#-kielelle. Suurimmat erot kielen välillä ovat syntaksissa: erilaisten sulkumerkkien sijaan lohko- tai lausekemerkkeinä käytetään avainsanoja. (Wikipedia 2015o, viitattu 1.5.2015.)

C#	Visual Basic .NET
<pre>for (int i = 0; i < number; ++i) { // loop from zero up to one less than number }</pre>	<pre>For i As Integer = 0 To number - 1 ' loop from zero up to one less than number Next</pre>
<pre>for (int i = number; i >= 0; --i) { // loops from number down to zero }</pre>	<pre>For i As Integer = number To 0 Step -1 ' loops from number down to zero Next</pre>
<pre>for (int i = 0; i <= 20; i += 2) { // loops from 0 to 20 by 2's }</pre>	<pre>For i As Integer = 0 To 20 Step 2 ' loops from 0 to 20 by 2's Next</pre>
	<pre>Exit For 'breaks out of a for loop</pre>
<pre>break; //breaks out of a loop</pre>	<pre>Exit While 'breaks out of a while loop</pre>
	<pre>Exit Do 'breaks out of a do loop</pre>

KUVIO 12. C#:n ja VB.NET:n silmukkaohjelmoinnin vertailu (Wikipedia 2015f, viitattu 23.04.2015)

3.6 C# (C sharp)

C# on yksinkertainen ja helppokäyttöinen, yleiskäyttöön tarkoitettu ISO-hyväksytty oliopohjainen ohjelmointikieli, jonka kehityksen pyrkimyksenä on yhdistää C/C++ -kielen syntaksi ja tehokkuus Javan helppokäyttöisyys.

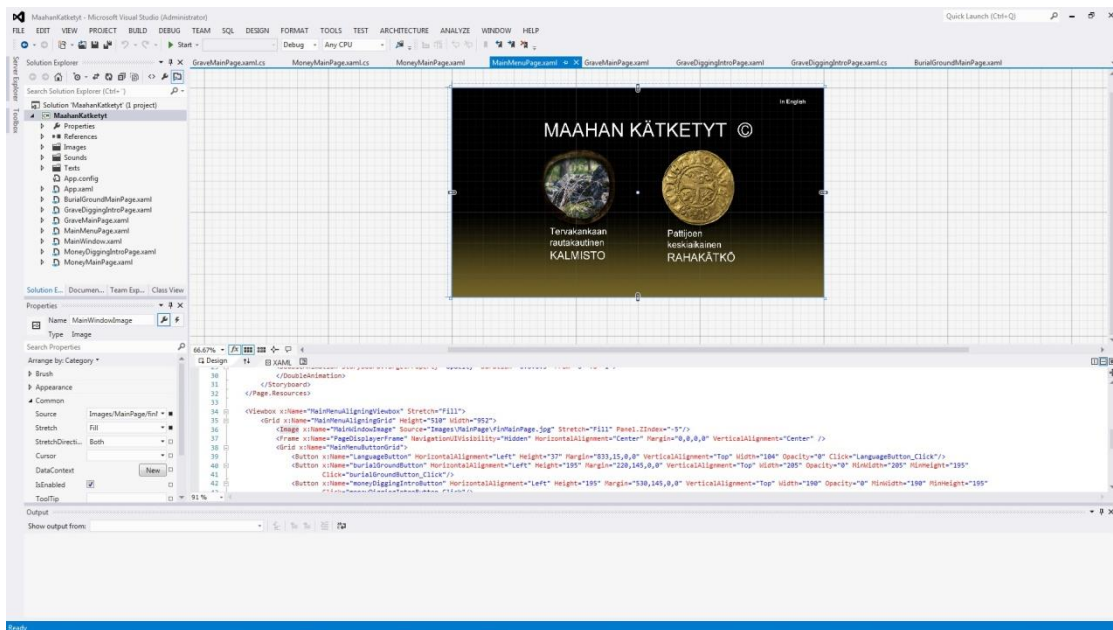
Ensimmäinen julkaisu tapahtui samalla kertaa Visual Basic .NET:n ja .NET Framework 1.0 -ohjelmistokomponenttikirjaston kanssa, vuonna 2002. Uusin versio (5.0) julkaistiin elokuussa 2012 ja seuraava versio (6.0, julkaisupäivä ei tiedossa) on kehitteillä. (Wikipedia 2015a, viitattu 3.3.2015.)

Tekijälle C# oli kielistä tutumpi Windows Phone -ohjelmointikurssien ja oman ohjelmointiharjoittelunsa vuoksi. Ainoa syy C#:n valintaan C#:n ja VB.NET:n välillä oli siis syntaksin parempi ymmärrys.

3.7 Microsoft Visual Studio 2012

Vuonna 2012 julkaistu versio Microsoftin Visual Studiosta oli käytössä ohjelmankehitysympäristönä (IDE) ja on ohjelman toiseksi uusin virallisesti julkaistu versio. Ensimmäinen versio ohjelmasta julkaistiin 1997 ja uusimman version julkaisupäivän odotetaan

olevan syyskuun 1. 2015. Visual Studio 2015 on tällä hetkellä saatavilla preview- eli esikatseluversiona.



KUVIO 13. Ohjelmistokehitysympäristö (IDE), Microsoft Visual Studio 2012

Verrattaessa vuoden 2012 versiota vuonna 2013 julkaistuun uudempaan versioon parannettuja ominaisuuksia versioiden välillä ovat esimerkiksi web-kehityksen ominaisuudet ja tuki Windows 8.1 -käyttöjärjestelmälle. Näiden ollessa projektille tarpeettomia ominaisuuksia päivitystä uudempaan versioon ei nähty välttämättömäksi.

Visual Studio sisältää lähdekoodieditorin, johon kuuluu IntelliSense -koodintäydennysosa, koodin refaktorointimahdollisuus debuggeri ja designeri. Sisäänrakennetusti tuettuja kieliä ovat: C, C++ ja C++/CLI (Visual C++), VB.NET, C# ja F#. Kuviossa 12 esillä on designer-näkymä, jossa näkyy visuaalinen representaatio ohjelmoitavana/editoitavana olevasta sivusta ja XAML-koodi-ikkuna, johon halutut muutokset ohjelmoidaan/kirjoitetaan. (Wikipedia 2015k, viitattu 4.5.2015.)

3.8 Kuvaeditointiohjelmisto, äänieditointiohjelmisto





Sekä kuva- että äänieditoinnit kuvataan tarkemmin omissa kappaleissaan (4.1 ja 4.2). Käytetyt ohjelmistot olivat Adobe Photoshop CS3 (Creative Suite 3) ja Audacity.

Adobe Photoshop on tunnetuimmaksi, käytetyimmäksi ja myydyimmäksi noussut kuvankäsittelyohjelma, jonka nimestä nykyinen kuvankäsittelyn synonyymi *photoshoppaus* juontuu. Ohjelmasta on olemassa useita versioita, joista Elements-nimellä kulkeva on suunnattu harrastelijakäyttöön. Elements on joiltakin osin karsittu versio ammattilaiskäyttöön tarkoitettu Creative Suitesta, vaikkakin siitä löytyy myös ylimääräisiä ominaisuuksia, joita ei ammattilaisversioissa ole. Versiomerkinä CS olevat Photoshop-ohjelmistot ovat nykyisin jo poistumassa. Tilalle ovat tulleet CC-versiot (Creative Cloud), jotka perinteisen lisenssipohjaisen hankinnan sijasta ovat SaaS-pohjaisia (Software as a Service) eli palveluina tilattuja, esimerkiksi kuukausimaksullisia tuotteita. (Wikipedia 2015b, viitattu 7.5.2015.)

Audacity on ilmainen, avoimen lähdekoodin digitaaliseen äänieditointiin ja -taltiointiin suunniteltu sovellus. Ilmaiseksi ohjelmaksi Audacity on ominaisuuksiltaan erittäin hyvä. Äänitiedostojen jälkikäsittelymahdollisuudet ovat monipuoliset, kuten esimerkiksi äänien kopiointi, leikkaus ja liittäminen, usean tiedoston yhtäaikaan editointi ja miksaus, normalisointi, kohinanpoisto näytteenoton avulla ja tempon sekä sävelkorkeuden muuttaminen ovat vain osa erityistarkoituksellisista editointimahdollisuuksista. (Wikipedia 2015c, viitattu 21.4.2015.)

3.9 Creative Commons

Creative Commons (tai lyhennettynä CC) on voittoa tuottamaton järjestö, joka pyrkii laajentamaan ja helpottamaan luovan työn tekijöiden luojuksien laillista jakamista ja niiden käyttöoikeuksien selkeyttämistä. Lisenssit mahdollistavat työn tehneelle helpon tavan ilmoittaa mistä oikeuksista hän luopuu ja mitkä oikeudet hän pidättää itsellään. Tämä alentaa suuresti verkkoon jaetun materiaalin käyttökynnystä, sillä yksittäiseltä tekijältä ei tarvitse erikseen tiedustella hänen kantaansa johonkin tiettyyn tekijänoikeudelliseen määreeseen. (Wikipedia 2015g, viitattu 21.4.2015.)

	Nimeä (BY)	Teosta saa kopioida, levittää, näyttää ja esittää julkisesti ja siitä saa luoda johdannaisteoksia, kunhan tekijän tai oikeudenomistajan nimi mainitaan asianmukaisesti.
	Epäkaupallinen (NC)	Teosta saa kopioida, levittää, näyttää ja esittää julkisesti ja siitä saa luoda johdannaisteoksia vain, kun niitä ei käytetä kaupallisiin tarkoituksiin.
	Ei muutoksia (ND)	Teosta saa kopioida, levittää, näyttää ja esittää julkisesti, mutta siitä ei saa luoda johdannaisteoksia.
	Tarttuva eli JaaSamoin (SA)	Johdannaisteoksia saa levittää vain samalla lisenssillä kuin alkuperäistä teosta.

KUVIO 14. Creative Commons -lisenssien neljä ehtoa. (Wikipedia 2015g, viitattu 27.4.2015)

Edellä kuvattuja ehtoja yhdistellään niin, että saadaan seuraavat kuusi päälisenssiä:

- Nimeä (CC BY)
- Nimeä-Tarttuva (CC BY-SA)
- Nimeä-Ei muutoksia (CC BY-ND)
- Nimeä-Epäkaupallinen (CC BY-NC)
- Nimeä-Epäkaupallinen-Tarttuva (CC BY-NC-SA)
- Nimeä-Ei muutoksia-Epäkaupallinen (CC BY-NC-ND)

Näiden lisäksi on olemassa myös CC0-lisenssi, jolla tekijä luopuu kaikista lain puitteissa mahdollisista oikeuksistaan teokseen. Lisenssi on kirjailtu näin, koska esimerkiksi Suomessa tekijän nimi täytyy aina mainita hänen moraalisten oikeuksiensa vuoksi. (Wikipedia 2015g, viitattu 27.4.2015)

4 KUVAT, ÄÄNET JA EDITOINNIT

Muokkausten aloittamista hankaloitti se, ettei alkuperäisiä, kerroksittain rakennettuja/piirrettyjä Photoshop Document (.psd) -tiedostoja ollut saatavilla. Kaikki muutokset jouduttiin tekemään hitaasti ja kärsivällisesti kuva kerrallaan. Ja koska tekijällä ei todellisia piirustustaitoja ole, koetettiin muutosten toteuttamisiin etsiä sellaisen Creative Commons -lisenssin alle tehtyjä kuvia, taustoja ja lisäyksiä, joita saisi tällaisessa projektissa käyttää ilmaiseksi.

Joitakin muutoksia pystyttiin myös tekemään Photoshopiin sisäänrakennetuin työkaluin, erityisesti filteröintityökalu tekstuurimahdollisuuksiensa (Filter/Texture...) ja taso- tai kerrostyökalu blandausefektiensä (Layer/Layer Style/Blending Options...) osoittautuivat erittäin käytännöllisiksi ja täydentäviksi luotaessa illustraatioihin näkyviä, elävöittäviä muutoksia.

4.1 Alkuperäiset kuvat ja ulkonäkö

Tässä osiossa esitellään alkuperäiset kuvat sellaisinaan ja jos tarpeelliseksi nähdään, kerrotaan kuvien tai kuvalajitelmien määreistä tai kuviin tehdyistä muutoksista enemmänkin. Ohjelman toimintaa avataan niin, että lukija saa käsityksen ohjelman toiminnasta.



KUVIO 15. Maahan kätkeyt, alkuperäinen etusivu (päävalikko)

Päävalikon kuvaan tuli vain vähän muutoksia, jotka museon henkilökunta teki itse. Kuviossa 15 on muista sivuista eroavia toimintanäppäimiä kolme: Tervakankaan rautakautinen kalmisto, Pattijoen keskiaikainen rahakätkö ja ©-merkki, josta päästään tekijänoikeuksien esittelysivulle. Alaoikealla on myös In English -merkintä, jota painamalla kuva vaihtuu englanninkieliseksi vastineekseen. Kyseinen merkintä löytyy jokaiselta sivulta ja sen toiminta on aina toimia nykyisen näkyvän kuvan vaihtamisena vastakkaiskieliseksi.

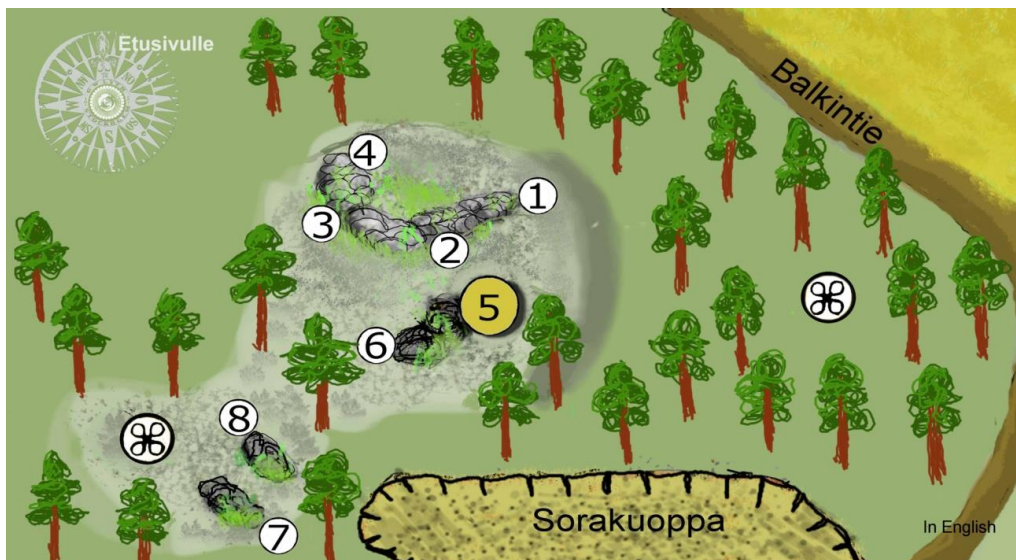


KUVIO 16. Maahan kätkeytyt, alkuperäinen tekijänoikeusselostesivu

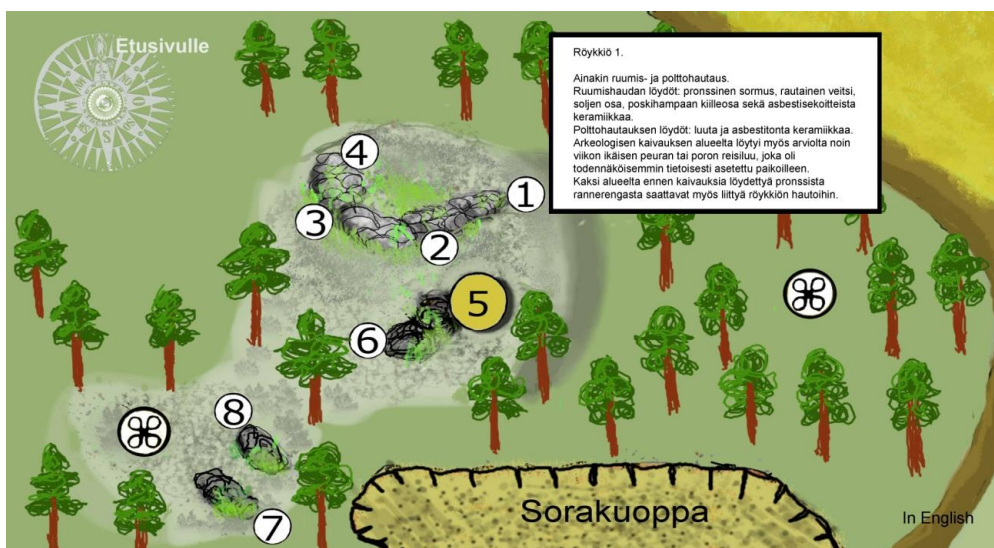
4.1.1 Tervakankaan rautakautinen kalmisto

Suurin määrä muutoksia tuli lopulta Tervakankaan rautakautinen kalmisto -osioon. Erityisesti kuvion 17 valikkoon muokkausaikaa kului paljon useiden erityylisten kokeiluiden ja vertailuiden muodossa. Kuviossa nähdään numerosymboleita, jotka esittävät kaivauksissa tehtyjä löydöksiä.

Ohjelman sisällä globaalin In English -merkinnän lisäksi kuviosta löytyy nyt myös kompassin muotoinen Etusivulle -merkintä, jonka toiminta on olemassa suurella osalla muista sivuista. Kohtaa painamalla ohjelma siirtyy takaisin päävalikkoon (kuvio 17).

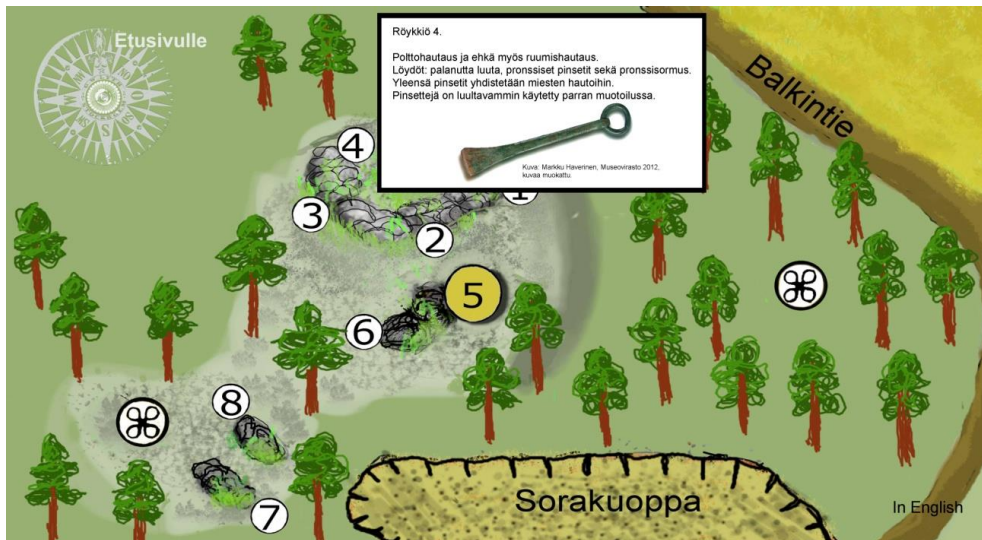


KUVIO 17. Tervakankaan rautakautinen kalmisto, valikko



KUVIO 18. Tervakankaan rautakautinen kalmisto, röykkiö 1:n infolaatikko

Käyttäjän painaessa kyseisiä symboleita ladataan uusi kuva (esimerkkeinä kuvat 18 ja 19), joissa kuvaan on tehty ja kirjoitettu lisätietoa tarjoava infolaatikko.



KUVIO 19. Tervakankaan rautakautinen kalmisto, röykkiö 4:n infolaatikko

Numerosta 5 käyttäjä pääsee jatkamaan uuteen osioon, jossa simuloidaan kaivauksen suorittamista.

Etusivulle

Arkeologinen kaivaus

Arkeologisessa kaivauksessa tutkittava alue **mitataan, paalutetaan** ja sidotaan peruskarttaan ja korkeuskiintopisteeseen. Alueelta poistetaan pintamaa ja kaivausalue **ruudutetaan**. Jokainen ruutu tutkitaan huolellisesti pienellä lapiolla tai lastalla. Maa-aines siivilöidään. Kun näkyviin saadaan isompi esine, se **tupsutellaan** ja harjataan varovasti esille. Kaivaus voidaan suorittaa joko maakerros kerrallaan tai tasokaivauksena.

Löydöt merkitään heti löytyessään karttaan ja suljetaan pusseihin, joihin merkitään löydön koordinaatit. Kaivaus **dokumentoidaan** tarkasti ja siitä **laaditaan tutkimusraportti**, jota hyödynnetään alueen tulevassa suojelussa ja hoidossa. Arkeologisiin kaivauksiin voit päästä tutustumaan esimerkiksi kesäisin eri puolella Suomea järjestettävillä yleisökaivauksilla.

Nyt Sinulla on mahdollisuus kurkistaa maan sisään.

Liikuta käsiäsi vuorotellen sivulle haudan päällä ikään kuin "siirtäisit maata" edestäsi pois.

Ympyränmallisia linkkejä OSOITTAMALLA saat esille tietosivun.

Kaivaukset Tervakankaalla 1990.

Aloita

In English

KUVIO 20. Tervakankaan rautakautinen kalmisto, numero 5, Arkeologisen kaivauksen infosivu

Osiossa oli yhteensä kymmenen erillistä, mutta taustakuvaltaan samanlaista kuvaa, joista yksi oli numerosta 5 avatuva infosivu (kuvio 20). Infosivua lukuun ottamatta muutoksia tehdessä täytyi siis ottaa huomioon, että yhteen kuvaan tehtävät muutokset olisi tehtävä myös muihin kahdeksaan kuvaan.

Kuviosta 20 voidaan huomata ohjelman tekstin käsittelysuunnittelun vaikeus. Yksi editointia lisäävistä ongelmista olivat kuviin sattuneet inhimilliset virheet. Infotekstin otsikko “Arkeologinen kaivaus” on muotoa “Argeologinen kaivaus”. Myös muista projektin tekstiä sisältävistä kuvista löytyi tällaisia, yllättävän paljon korjausaikaa vieviä lipsahduksia. Nämä kirjoitusvirheet edistivät kuitenkin korjausaikavaateineen ja mahdollisten uusien virheiden tekemisen pelkona viimeisen ohjelmistoversion kehitystä parempaan suuntaan, jossa mahdollisimman moni kuva olisi pelkkä tausta.

4.1.2 Tervakankaan rautakautinen kalmisto, kaivausosio

Kaivausosion (Tervakankaan rautakautisen kalmiston numero 5) toiminta oli suunniteltu niin, että käyttäjän liikuttaessa kättään kaivamisliikettä simuloiden kuva vaihtuu järjestyksessä seuraavana olevaan kuvaan.



KUVIO 21. Tervakankaan rautakautinen kalmisto, kaivausosio, aloituskuva

Kuvia oli yhteensä 29. Alkuperäisvaiheessa kuvien vaihtuminen seuraavaan oli välitön, eli minkäänlaista sulavuutta luovaa animointia ei vaihdosten välille ollut tehty.

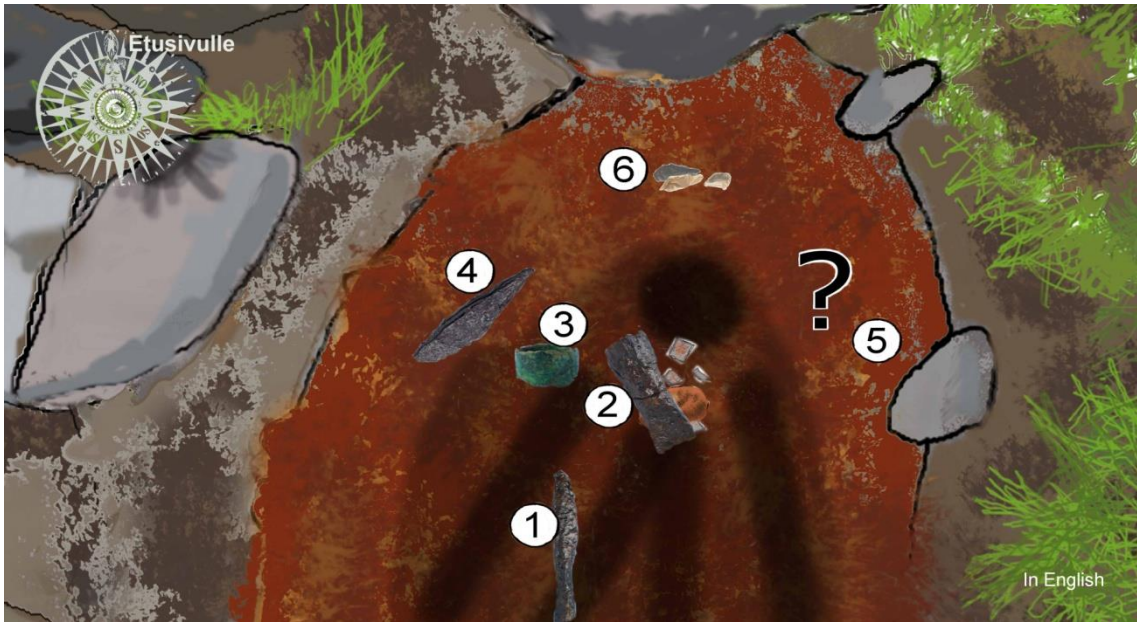


KUVIO 22. Tervakankaan rautakautinen kalmisto, kaivausosio, kuva 13

Kuvissa näkyvät värimuutokset merkitsevät maakerrosten vaihdoksia. Kerrosten vaihdoksista puuttuivat tällä hetkellä vielä kuvan merkitystä avaavat selostustekstit.

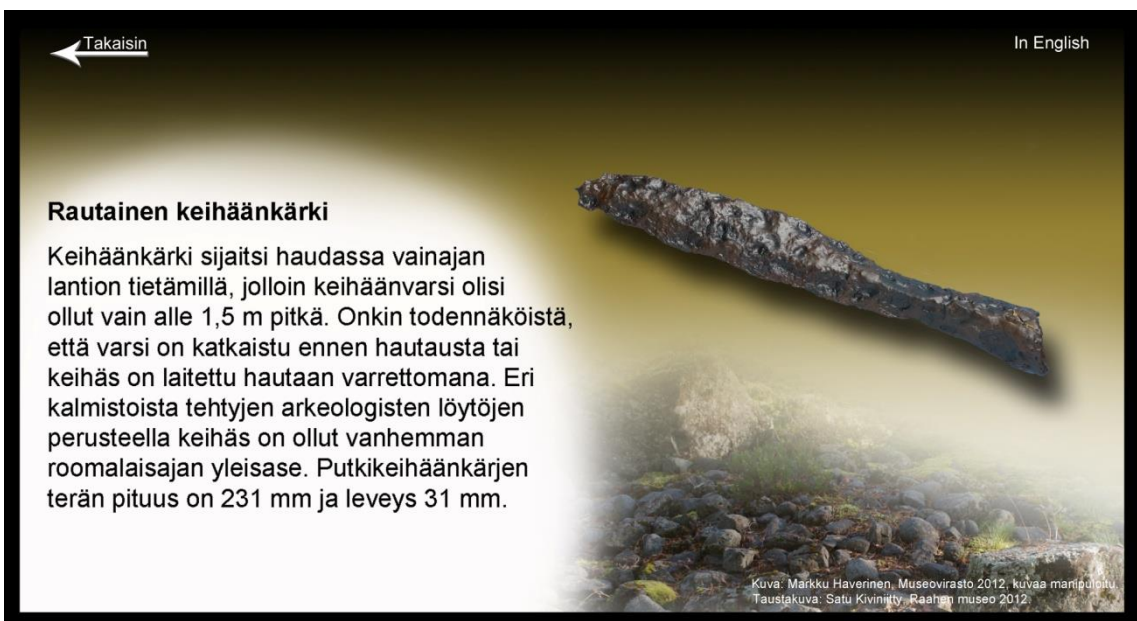


KUVIO 23. Tervakankaan rautakautinen kalmisto, kaivausosio, viimeinen kuva ennen esinevalikkoa



KUVIO 24. Tervakankaan rautakautinen kalmisto, kaivausosio, esinevalikko

Kaivaussimulaation päätyttyä käyttäjä ohjattiin valikkoon, josta numerosymboleita painamalla pystyttiin avaamaan esineistä lisää kertovia infokuvia. Kuvia oli yhteensä seitsemän. Kuvioista 24 voidaan huomata, ettei kaikille kuville ollut vielä vahvistettua symbolia. Numero 5:stä avautuu sama sivu kuin kysymysmerkistä ja seitsemättä selityskuvaa ei ole vielä ollenkaan. Esinesivuilla ei ole linkkiä takaisin pääsivulle, vaan sen sijasta käyttäjä pääsee Takaisin-nuolta painaessaan takaisin esinevalikkoon.



KUVIO 25. Tervakankaan rautakautinen kalmisto, kaivausosio, esinevalikko, esine 1

4.1.3 Pattijoen keskiaikainen rahakätkö

Rahakätkölle oli suunniteltu kalmiston kaivausosion kaltainen simulaatio, jonka toiminta oli likipitään samanlainen. Kuten koko Tervakankaan kalmiston osuudessa, myös rahakätköosuudesta puuttuivat sulavat animaatiot sekä äänitehosteet ja selitystekstit.



KUVIO 26. Pattijoen keskiaikainen rahakätkö, introkuva 1



KUVIO 27. Pattijoen keskiaikainen rahakätkö, introkuva 2



KUVIO 28. Pattijoen keskiaikainen rahakätkö, introkuva 3

Rahakätkön alkuun oli suunniteltu tehtäväksi tunnelmaa luova intro, jonka visuaalista puolta esittävät kuvat 26, 27 ja 28. Kuvat olivat niin sanottuja placeholdereita, eli vain esimerkkeiksi suunniteltuja, pikaisia malleja. Kuvien oli määrä vaihtua sulavasti ja taustalla olisi oltava ilmapiiriin sopivia äänitehosteita, kuten linnunlaulua, veden solinaa ja kaivuuääniä. Nämä jäivät tekijän ratkaistaviksi ongelmiksi.



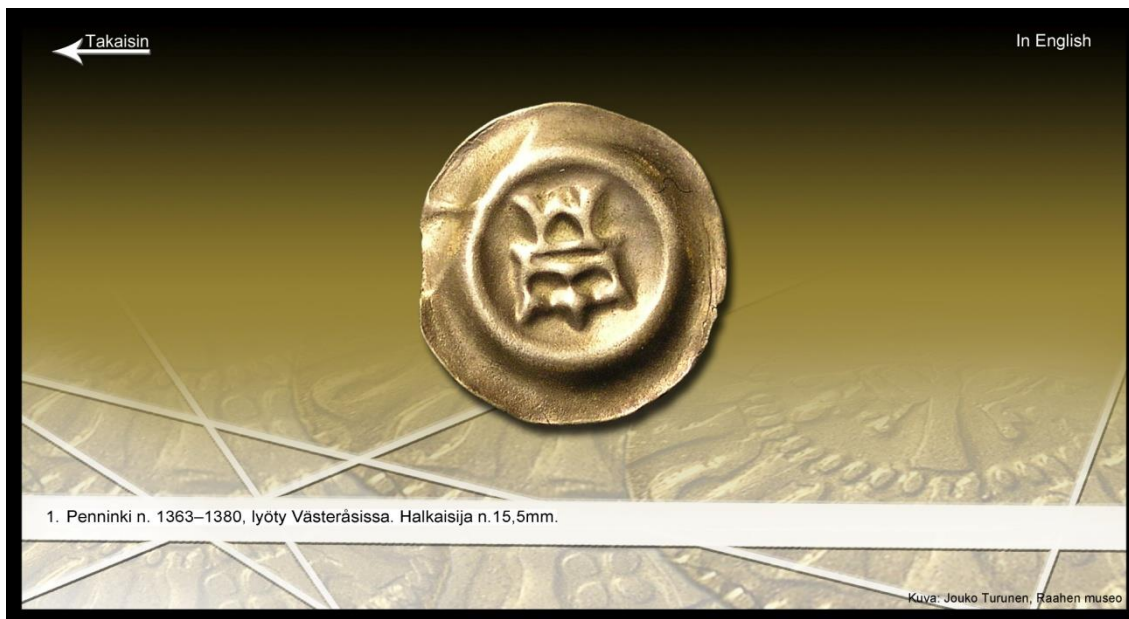
KUVIO 29. Pattijoen keskiaikainen rahakätkö, kaivuosio, aloituskuva



KUVIO 30. Pattijoen keskiaikainen rahakätkö, kaivuuosio, viimeinen kuva ennen rahavalikkoa



KUVIO 31. Pattijoen keskiaikainen rahakätkö, kaivuuosio, rahavalikko



KUVIO 32. Pattijoen keskiaikainen rahakätkö, kaivuuosio, raha 1

Rahavalikko ja rahat olivat kalmiston esinesivujen kaltainen esitys, jossa käyttäjä voi lähemmin tutkia rahakätköstä tehtyjä löydöksiä. Osio käsitti yhteensä 13 kuvaa.

4.1.4 Apukuvien etsintä ja käytettäviksi päätyneet kuvat

Alkuun pelättiin ettei kuvien löytäminen olisi hirvittävän helppoa. Epäilyksenä oli, että ensin täytyisi löytää sivustoja, jotka olisi tehty ja suunniteltu nimenomaan CC-lisenssioikeuksien käytettävissä olevien kuvien jakoon. Etsiminen kuitenkin aloitettiin kokeilemalla onnea Google-kuvahauulla käyttäen hakusanana lausetta "creative commons pine tree illustration". Huoli vaativuudesta häipyi pian, sillä kuvia löytyi valtava määrä. Samalla yksittäisten, ulkonäöltään tarkoitukseen sopivien kuvien osoitteet antoivat vinkkiä itse sivustoista, joilta löytyisi muitakin CC-käyttöoikeuksien alle piirrettyjä tai otettuja kuvia.

Kuvia selaillessa mietittiin työn toteutustapaa. Tarkoitus oli parannella kuvia jonkin verran, kuitenkin pitäen työmäärä hallittavana. Ideaksi muodostui muokata Tervakankaan kalmiston kuvituskuvista (kuviot 17, 18 ja 19) sellainen kokonaisuus, jossa olisi vain yksi pohjakuva. Infotekstikuvat hyljättäisiin kokonaan ja vaaditut tekstit esitettäisiin pohjakuvan päälle ilmestyvinä, erillisinä kuvina.

Tällöin editointityön määrä pysyisi järkevänä sekä ohjelmoinnin vaativuutta saataisiin lisättyä. Infotekstien taustaksi päätyi deviantart.comista löytynyt Anne Rackhamin tekemä kääroaiheinen kuva, joka tekijän mielestä sopi projektin luonteeseen ja aiheeseen.



KUVIO 33. Anne Rackham, "Old paper scroll background", annerackham.deviantart.com, CC BY-SA 3.0

Lisenssin CC BY-SA 3.0 mukaan kuvaa voidaan käyttää jopa kaupallisiin tarkoituksiin, kunhan tekijälle annetaan asianmukainen tunnustus ja kuvaan tehdyistä muutoksista ilmoitetaan. Uusi, luotu johdannaistyö täytyy myös jakaa samalla lisenssillä; lisenssiehtoja ei voi korottaa tai laskea halujensa mukaan. Lisenssi sopi siis tarkoituksiin todella hyvin.

Käärökuviin kirjoitettiin Photoshopilla infolaatikoiden tekstit sopivaksi katsotulla fontilla. Ohjelmoimalla animoitiin kuva liukumaan sulavasti näkyviin ruudun yläreunasta sekä palaamaan takaisin infotekstiä vaihdettaessa.

Toinen käyttöön päätyneistä kuvista oli nimimerkillä tyke44060 esiintyvän taiteilijan piirtämä puukollaasi. Kuva – tai kuvan puut – valittiin korvaamaan kuviossa 17 näkyvät puut. Ilmettä saatiin näin tehostettua luonnollisemmaksi. Yksi hyvä puoli käytetyssä teoksessa oli se, että se sisälsi kolme erilaista puuta. Muussa tapauksessa lisätyötä olisi aiheuttanut sellaisten kahden tai kolmen erillisen puukuvan etsintä, jotka jollakin tavalla sopisivat hyvin yhteen.

Kuvion 34 teoksen lisenssi CC BY-NC-ND 3.0 aiheutti hieman edellistä kuvaa enemmän päänvaivaa. Asianmukaisen tunnustuksen lisäksi kuvaa ei saa käyttää kaupallisiin tarkoituksiin, eikä kuvaa saa jakaa jos sitä on muokattu. Nimenomaan kaupallisen käytön kielto mietitytti. Museo kyllä veloittaa asiakkailtaan sisäänpääsymaksun, mutta rahallisesti se on hyvinkin muodollinen summa, 2-4 euroa/henkilö.



KUVIO 34. tyke 44060, "Pine Trees", tyke44060.deviantart.com, CC BY-NC-ND 3.0

Museon tiloissa on tietysti myös useita muita näyttelyjä ja näyttelyesineitä, joten rahallinen korvaus kuvan käytöstä yhden esityksen yhdessä taustakuvassa olisi jäänyt erittäin pieneksi. Varmistukseksi päätettiin kuitenkin tiedustella asiaa suoraan itse tekijältä. Kontakti tekijään saatiin deviantart.comin "Send a note" -toiminnon kautta, jonka käyttö vaati vain käyttäjätilin luomisen deviantartiin. Vastaus saatiin pian, eikä tekijällä ollut mitään hänen teoksensa käyttöä vastaan.

Kuvaa käytettiin leikkaamalla puut erilleen toisistaan ja lisäämällä niitä haluttuun taustakuvaan. Pieniä lisäeroavaisuuksia puiden leikkausten kopioiden välille aikaansaatiin muokkaamalla leikkeiden korkeutta, leveyttä ja käyttämällä Photoshopin pelikuva-efektiä.

4.2 Äänet ja äänieditointi

Ääniefektien teko alkoi samalla tavalla kuvien kanssa: etsinnällä. Tähänkin riitti Google-haku sopivilla hakusanoilla. Tärkeimmät löydetyt sivustot olivat tehosto.yle.fi, freesound.org ja freesfx.co.uk. Sivustoilla on yhteensä kymmeniä tuhansia useiden tekijöiden tuottamia efektejä ja pitkiäkin ääninäytteitä.

Ylen Tehosto (<http://yle.fi/vintti/yle.fi/tehosto.yle.fi/>, <http://www.freesound.org/people/YleArkisto/>) osoittautui erittäin hyväksi taustäänilähteeksi. Tehosteissa on erityisen hyvää niiden sopiminen suomalaiseen äänimaisemaan, jolloin esimerkiksi lintujen äänet ovat oikeasti kotimaiseen metsämaisemaan sopivia. Ongelmia tosin tuotti sivustolle 18.03.2014 tapahtunut arkistointi, jonka jälkeen sivuja ei ole enää päivitetty eivätkä kaikki sivun elementit toimi enää oikein. Yleisradio on kuitenkin siirtämässä tehosteitaan freesound.orgiin, jonne tulevatkin tehosteet on tarkoitus lisätä. Tavoitteena heillä on 80 000 ääninäytteestään digitoida noin 60 000. Yleisradiolta pyydettiin erillistä lupaa ääninäytteiden käyttöön varmuuden varalta, sillä vaikkakin heidän freesound.org -sivuillaan lisenssiksi on määritetty CC BY 3.0 Tehoston alkuperäisillä sivuilla mainitaan äänitteiden olevan tarkoitettu vain yksityiskäyttöön.

Muita taustääninä tai taustäänien lisukkeina käytettyjä artisteja olivat Alastair Cameron (www.cameronmusic.co.uk, www.freesound.org/people/cameronmusic/) ja Gerard Hadders (www.freesound.org/people/gerardhadders). Heistä Hadders on lisensoinut tuotteensa täysin tekijänoikeusvapaiksi (CC0 1.0). Cameron taas edellyttää asianmukaista tunnustusta ja mahdollisten muutosten selvää indikointia (CC BY 3.0).

Efekteinä käytettyjä ääniä tarvittiin Tervakankaan kalmiston kaivausosioon, Pattijoen rahakätkön kaivuusosioon ja rahakätkön rahavalikkoon. Nämä koostettiin leikkaamalla sopivia osia edellä mainittujen artistien tuotannoista sekä käyttämällä neljän muun taitajan tuotoksia. Nimimerkeillä Audiomicro, Blastwave FX ja Flow FX efektejä taiteilleet freesfx.co.uk:n käyttäjät (tai sivusto itse) vaativat lisenssisopimuksessaan (www.freesfx.co.uk/info/eula/) loppukäyttäjältä tunnustusta enemmänkin freesfx-sivustolle kuin käyttäjille. Sattumana sivuston lisenssiluvassa tälle projektille sopivasti mainitaan luvallisissa käyttöesimerkeissä erityisesti museokäyttö.

Projektin ainoa maksullinen aineeton resurssi oli yksi ääniefekteistä. Big Room Soundin (omistaja J.R. Fountain, www.bigroomsound.com, http://www.soundsnap.com/user-name/j_r_fountain) tuottamat tasoituslasta- tai laastilapioefektit olivat täydelliset kalmiston kaivausosiossa käytettäviksi. Muita vaihtoehtojakaan ei myöskään tahtonut löytyä. Efekti ei ollut kallis, sillä soundsnap.com -sivustolta voi halvimmillaan ostaa viisi ääniefektiä/ääniefektin latausta hintaan 9 US-dollaria (noin 8 euroa). Yhdelle äänelle hintaa jää siis noin 1,6 euroa. Soundsnapin lisenssi (www.soundsnap.com/licence) ei pidättänyt juuri mitään muita kuin jälleenmyyntioikeuden, eikä vaadi kaupalliseen tarkoitukseen käytettäessä etumaksun lisäksi muita maksuja tai rojalteja.

5 VERSIO 0.9.7.1 ALPHA

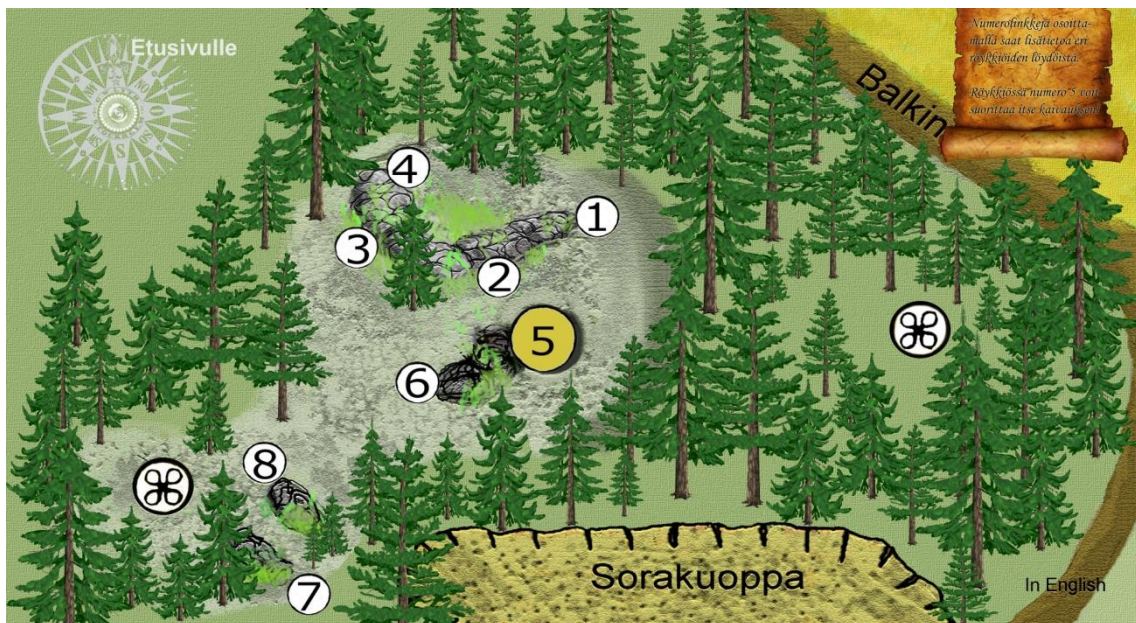
Ensimmäinen museolle testikäyttöön lähetty versio 0.9.7.1 oli alfaversio, jonka joitain käytön ja toiminnan ongelmia tiedettiin jo etukäteen. Testiversion päätarkoituksena oli visuaalisen ilmeen ja äänien käytön hyväksyntä sekä tarkistus ja mahdollisten uusien, tekijän testeissä huomaamattomien ohjelmointivirheiden löytäminen. Ohjelmasta puuttui tässä vaiheessa vielä haluttujakin toimintoja, joita lisättäisiin seuraaviin testeihin lähetettäviin versioihin.

5.1 Version 0.9.7.1 ulkonäkö

Alkuperäiset kuvat olivat jo sellaisinaan museon hyväksymät, eikä niitä olisi tarvinnut muokata tekijän toimesta. Toivomuksena kuitenkin oli että kuvia hieman elävöitetäisiin, jota tekijä lähti tavoittelemaan taitojensa mukaan.

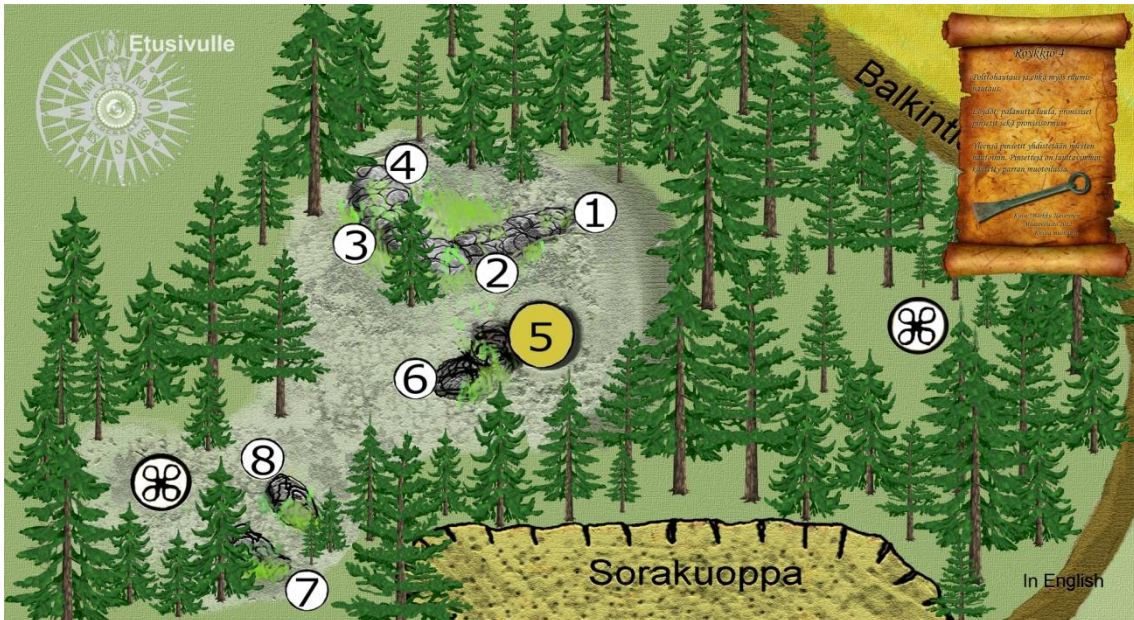
5.1.1 Tervakankaan rautakautinen kalmisto

Kalmiston valikkosivu oli muokattu kuvion 35 kaltaiseksi ja pysyi lopulliseen versioon saakka samanlaisena.



KUVIO 35. Tervakankaan rautakautisen kalmiston valikko, version 0.9.7.1 pohjakuva ja alkuinfo

Kuvan yläreunassa näkyvä infoteksi ilmestyi näkyviin alaspäin laskien käyttäjän siirtyessä sivulle. Numerosymboleita painamalla kääron teksti vaihtui niin, että kääro ensin palautui ylös piiloon, jossa teksti vaihtui ja sen jälkeen laskeutui takaisin näkyviin. Taustalla soivat äänitehosteet. Erilliset infotekstikuvat oli poistettu ja tilalle tehty jokaisesta kääromallin infoteksti.



KUVIO 36. Tervakankaan rautakautinen kalmisto, version 0.9.7.1 röykkiö 4:n infokääro avoinna



KUVIO 37. Tervakankaan rautakautinen kalmisto, kaivausosio, version 0.9.7.1 kuva 13

Kaivausosion muutoksia olivat kuviin toteutetut tekstuurimuutokset, lisätyt maakerroksia kuvaavat, halutuissa kohdissa vaihtuvat infotekstit, taustääänet ja sulavasti animoidut kuvavaihdokset.



KUVIO 38. Tervakankaan rautakautinen kalmisto, kaivausosio, version 0.9.7.1 esinevalikko

Esinevalikkoa muokattiin niin, että seitsemättä esinekuvaa merkitsi alaoikealla näkyvä ihmisoliosymboli ja numero 5:n virkaa ajoi kysymysmerkki. Esinekuvista oli poistettu paksut, mustat reunukset (kts./vrt. kuvio 25). Kuvaan lisättiin tekstuuria ja numerosymboleita esineineen järjesteltiin vähemmän virhepainalluksia aiheuttavaan järjestykseen.

5.1.2 Pattijoen keskiaikainen rahakätkö

Rahakätköösioon oli museon toimesta tehty uudet introkuvat ja lisätty introteksti. Kuvat oli animoitu vaihtumaan sulavasti toisiinsa kolmen sekunnin viiveellä ja taustalla soi alustava esimerkki taustaäänelle.



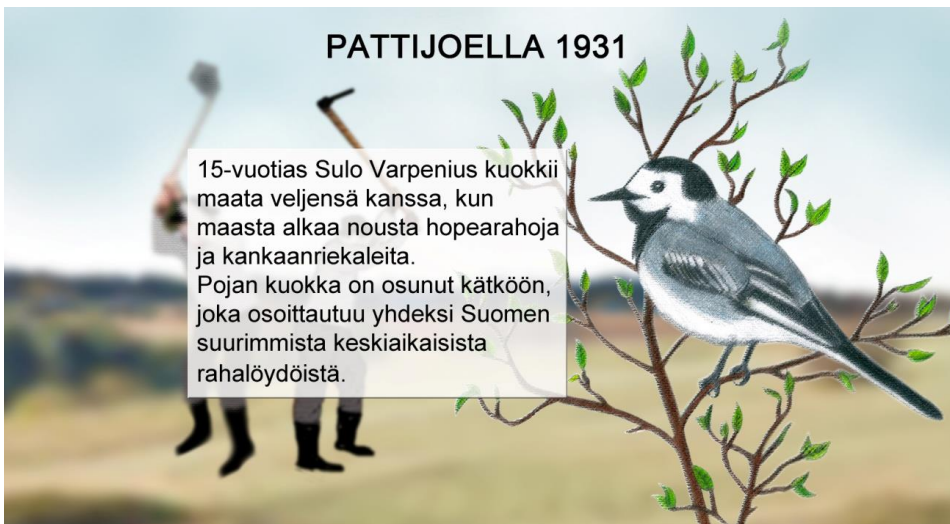
KUVIO 39. Pattijoen keskiaikainen rahakätkö, version 0.9.7.1 introkuva 1



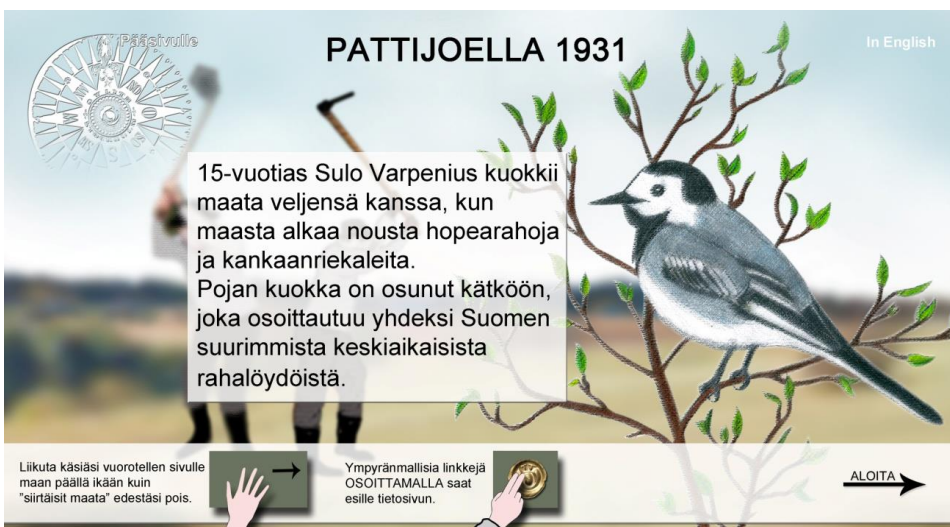
KUVIO 40. Pattijoen keskiaikainen rahakätkö, version 0.9.7.1 introkuva 2



KUVIO 41. Pattijoen keskiaikainen rahakätkö, version 0.9.7.1 introkuva 3



KUVIO 42. Pattijoen keskiaikainen rahakätkö, version 0.9.7.1 introkuva 4



KUVIO 43. Pattijoen keskiaikainen rahakätkö, version 0.9.7.1 introkuva 5



KUVIO 44. Pattijoen keskiaikainen rahakätkö, kaivuuosio, version 0.9.7.1 aloituskuva

Rahakätkön kaivuuosioon oli museon toimesta lisätty vihjetekstit käyttäjälle. Taustääniesitys soi taustalla.



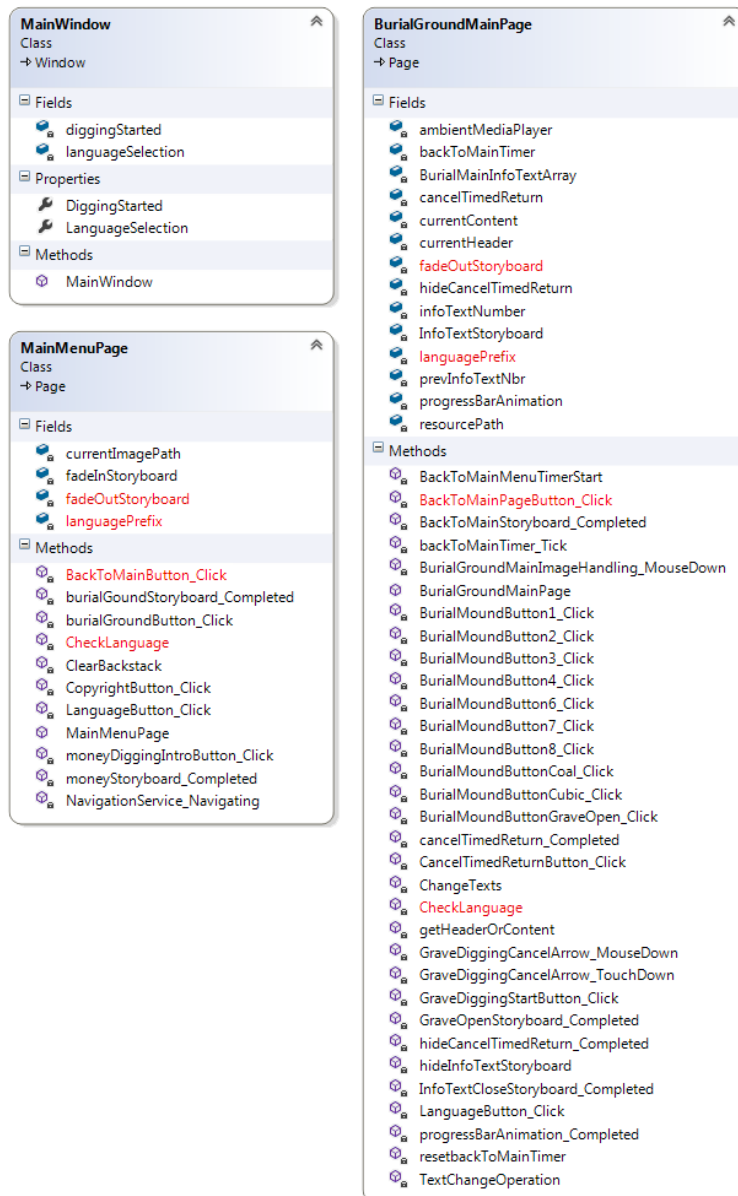
KUVIO 45. Pattijoen keskiaikainen rahakätkö, kaivuuosio, version 0.9.7.1 lopetuskuva

Rahakätkön kaivuuosion rahavalikko- ja rahaosuuksiin ei tullut muita muutoksia kuin kuvissa olleiden mustien reunojen poisto. Mustat reunat poistettiin myös kaikista muista kuvista, joihin oli aikoinaan jätetty yhtenäisemmän ilmeen saavuttamiseksi.

5.2 Version 0.9.7.1 ohjelmointi

Koska ohjelmointi aloitettiin puhtaalta pöydältä, saatiin projektin toteutus suunnitella alun alkaen täysin omatoimisesti. Suunnitelman pohjaksi muodostui ohjelman jakaminen osioihin käyttämällä XAML-sivuja. Yksi sivu käsittäisi aina toiminnallisesti saman tyylisen osion. kuten projektissa Tervakankaan kalmiston ensimmäinen valikkosivu ja kalmiston kaivaussimulaattorisivu ovat omilla sivuillaan, koska niiden toiminnot ja toimintatarpeet ovat toisistaan täysin eroavat. Sivuja käyttäen ohjelmiston kehitys saatiin myös pidettyä selkeästi osiin pilkottuna. Ensin tehtiin valmiiksi pääsivu, jonka jälkeen siirryttiin työstämään kalmiston valikkosivua, josta jatkettiin kalmiston kaivuusivun työstöön ja niin edelleen. Myös ohjelmoinnin suorittamista saatiin tällä tavoin nopeutettua ja helpommin jäsenneltyä.

Huono puoli kyseisessä tavassa oli, että lähes samanlaisia funktioita tehtiin useaan kertaan niiden toiminnan varmistamiseksi. Alkusuunnitelmassa olisi pitänyt ottaa huomioon, että kaikilla sivuilla on suuristakin eroavaisuuksista huolimatta useita yhteisiä tapahtumia tai -ketjuja, jotka olisi voinut ohjelmoida käyttäen jotakin tai useitakin apuluokkia yleistointeihin erikoistettuina. Kuvion 46 luokkadiagrammeihin on punaisella korostettu esimerkkejä metodeista tai muuttujista, jotka löytyvät kaikilta muiltakin sivuilta ja joiden toiminnot olisi voitu yhdistää apuluokan alle tai ohjelmaan globaaleiksi muuttujiksi ja metodeiksi. Ohjelmoinnissa kuitenkin keskityttiin pääasiaan, eli projektin saattamiseen käyttövalmiiksi.



KUVIO 46. Version 1.1.0.3 Luokkadigrammit luokista *MainWindow*, *MainMenuPage* ja *BurialGroundMainPage*

5.2.1 Ohjelmakoodin esittelyä

Ohjelman koodirivimäärä on suuri osittain edellisessä kappaleessa mainitusta syystäkin. Projekti käsittää XAML- ja C#-tiedostoineen yli 1700 riviä koodia, joten tästä määrästä päätettiin jokaiselta sivulta seuloa tärkeimmiksi katsotut funktiot ja metodit tai muut sellaiset aiheet, jotka mahdollisesti helpottaisivat työn ymmärtämistä. Seuraavaksi esitellään versiosta 0.9.7.1 valittua koodia.

5.2.2 MainWindow.xaml, MainWindow.xaml.cs

XAML:

```
<Window
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  x:Name="MainMenu" mc:Ignorable="d" x:Class="MaahanKatketyt.MainWindow"
  Title="Maahan Kätkeyt" ResizeMode="CanResizeWithGrip" Background="Black"
  WindowStyle="None" WindowState="Maximized">

  <Frame x:Name="MainFrame" Source="MainMenuPage.xaml"
    NavigationUIVisibility="Hidden" HorizontalAlignment="Center"
    Margin="0,0,0,0" VerticalAlignment="Center" />
</Window>
```

Ensimmäiseksi alustetaan ikkuna, johon kaikki ohjelman sisältö laitetaan näkyviin. Tärkeää sisältöä on Frame-elementti, johon päävalikkosivu avataan ohjelma käynnistettäessä.

C#:

```
//Digging started >> change when on digging pages to know where to go back to in
navigation
public static bool DiggingStarted
{
    get { return diggingStarted; }
    set { if (diggingStarted != value) { diggingStarted = value; } }
}

//Language boolean setter and getter
public static bool LanguageSelection
{
    get { return languageSelection; }
    set { if (languageSelection != value) { languageSelection = value; }}
}
```

Käynnistettäessä alustetaan kaksi julkista ja staattista funktiota, jotka käsittelevät yksityisiä totuusarvomuuuttujia. Muuttujia päästään näin käsittelemään myös muilta sivuilta. Vihreänä näkyvä on C#-tyylinen kommentointi, joka ei ole ajettavaa koodia, vaan ohjelmoijan jättämiä vihjeitä koodin toiminnasta.

5.2.3 MainMenuPage.xaml, MainMenuPage.xaml.cs

XAML:

```
<Viewbox x:Name="MainMenuAligningViewbox" Stretch="Fill">
  <Grid x:Name="MainMenuAligningGrid" Height="510" Width="952">
```

```

<Image x:Name="MainWindowImage"
  Source="Images\MainPage\finMainPage.jpg" Stretch="Fill"
  Panel.ZIndex="-5"/>
<Frame x:Name="PageDisplayerFrame" NavigationUIVisibility="Hidden"
  HorizontalAlignment="Center" Margin="0,0,0,0"
  VerticalAlignment="Center" />
<Grid x:Name="MainMenuButtonGrid">
  <Button x:Name="LanguageButton" HorizontalAlignment="Left"
    Height="37" Margin="833,15,0,0" VerticalAlignment="Top"
    Width="104" Opacity="0" Click="LanguageButton_Click"/>
  <Button x:Name="burialGroundButton" HorizontalAlignment="Left"
    Height="195" Margin="220,145,0,0" VerticalAlignment="Top"
    Width="205" Opacity="0" MinWidth="205" MinHeight="195"
    Click="burialGroundButton_Click"/>
  <Button x:Name="moneyDiggingIntroButton" HorizontalAlignment="Left"
    Height="195" Margin="530,145,0,0" VerticalAlignment="Top"
    Width="190" Opacity="0" MinWidth="190" MinHeight="195"
    Click="moneyDiggingIntroButton_Click"/>
  <Button x:Name="CopyrightButton" HorizontalAlignment="Left"
    Height="34" Margin="730,85,0,0" VerticalAlignment="Top" Width="92"
    Opacity="0" Click="CopyrightButton_Click"/>
  <Button x:Name="BackToMainButton" HorizontalAlignment="Left"
    Height="165" Margin="15,23,0,0" VerticalAlignment="Top" Width="181"
    Opacity="0" Click="BackToMainButton_Click"/>
  <!--<Button x:Name="CloseButton" Content="Sulkee ohjelman"
    IsCancel="True" HorizontalAlignment="Left" Height="34"
    Margin="16,457,0,0" Visibility="Hidden" VerticalAlignment="Top"
    Width="92" Opacity="50" Click="CloseButton_Click"/>-->
</Grid>
</Grid>
</Viewbox>

```

Pää- tai etusivun XAML-koodista nähdään, miten kaikkien sivujen perusta on rakennettu. Sivulle tehdään Viewbox, joka pitää sisällään Gridin, joka sisältää kaiken sivun sisällön. Näin sisältö saadaan helposti järjestettäväksi ja sisältö pitää määritellyn sijaintinsa näytön resoluutiosta riippumatta. Ensimmäisen Gridin sisään on lisätty myös toinen Grid-elementti (MainMenuButtonGrid). Syy tähän on nappien helpompi käsittely, kun napit täytyy esimerkiksi ottaa pois käytöstä. Tällöin voidaan estää nappien toiminta pelkästään estämällä ne sisältävän Gridin osumanrekisteröinti (IsHitTestVisible = false). Vihreällä XAML-tyylinen kommentointi, tässä käytetty jättämään väliaikaisesti yksi nappi pois koodin ajosta.

C#:

```

//Check which language has been selected
private void CheckLanguage()
{
  if (!MainWindow.LanguageSelection)
  {
    //Language defaults to Finnish
    languagePrefix = "fin";
    currentImagePath =
      "pack://application:,,,/MaahanKatketyt;component/Images/MainPage/finMainPa
ge.jpg";

```

```

this.MainWindowImage.Source = new BitmapImage(new Uri(currentImagePath,
    UriKind.Absolute));
currentImagePath = this.MainWindowImage.Source.ToString();
}
else
{
//If language has already been set and/or for some reason main page is
    reloaded
languagePrefix = "eng";
currentImagePath =
    "pack://application:,,,/MaahanKatketyt;component/Images/MainPage/finMainPa
    ge.jpg";
this.MainWindowImage.Source = new BitmapImage(new
    Uri(currentImagePath.Replace("fin", "eng"), UriKind.Absolute));
currentImagePath = this.MainWindowImage.Source.ToString();
MainWindow.LanguageSelection = true;
}
}
}

```

Pääsivun kielentarkistusfunktio. Funktio tarkistaa, onko MainWindown LanguageSelection-metodin palautusarvo false. Jos näin on, taustakuvaksi asetetaan suomenkielinen kuva. Jos palautusarvo on true, ladataan englanninkielinen kuva. Joka sivulta löytyy oma CheckLanguage-funktionsa, jotka sisältävät pieniä muutoksia tai lisäyksiä.

Tässä on samalla yksi esimerkki turhasta ja myöhemmissä versioissa korjatusta koodista. Kummassakin tapauksessa määritellään currentImagePath-merkkijonomuuttuja, vaikka muuttuja voitaisiin alustaa sivua ladattaessa ja muuttaa ajon aikana pelkästään MainWindowImage-lähteen muuttamista currentImagePath.Replace("fin", "eng") -metodia ja MainWindow.LanguageSelectionin asettamista käyttäen.

```

//Start DiggingIntro, eventually navigate to MoneyMainPage
private void moneyDiggingIntroButton_Click(object sender, RoutedEventArgs e)
{
    MainWindow.DiggingStarted = true;
    this.MainMenuButtonGrid.IsHitTestVisible = false;
    fadeOutStoryboard = this.FindResource("Opacity_OnNavigationFrom") as
        Storyboard;
    Storyboard.SetTarget(fadeOutStoryboard, this.MainWindowImage);
    fadeOutStoryboard.Completed += moneyStoryboard_Completed;
    fadeOutStoryboard.Begin(this);
}

private void moneyStoryboard_Completed(object sender, EventArgs e)
{
    fadeOutStoryboard.Completed -= moneyStoryboard_Completed;
    PageDisplayFrame.Navigate(new Uri("/MoneyDiggingIntroPage.xaml",
        UriKind.Relative));
}

```



```

PageDisplayerFrame.Navigating += new
    NavigatingCancelEventHandler(NavigationService_Navigating);
}

```

Rahakätkönäpin painalluksen käsittely. Suurin piirtein tällainen tapahtumaketju laukeaa myös kaikkien muiden sivulta-sivulle siirtyvien nappien painalluksista. Muiden sivuspesifisten toimintojen muassa käynnistetään sivulta pois siirtymiseen tehty animaatio, joka manipuloi sivun läpikuultamattomuutta muuttaen sen sekunnin kuluessa arvosta 1 (100%) arvoon 0 (0%). Kun animaatio on suoritettu, laukeaa animaation suoritusseuraajaksi määritelty tapahtumakäsittelijä, joka navigoi halutulle sivulle.

5.2.4 BurialGroundMainPage.xaml, BurialGroundMainPage.xaml.cs

XAML:

```

<Page.Resources>
  <Storyboard x:Name="ShowTextScroll" x:Key="Margin_OnTextScrollOpen">
    <ThicknessAnimationUsingKeyFrames Storyboard.TargetProperty="Margin"
      BeginTime="00:00:00">
      <SplineThicknessKeyFrame KeyTime="00:00:00" Value="0, -56,0,0" />
      <SplineThicknessKeyFrame KeyTime="00:00:01" Value="0, -8,0,0" />
    </ThicknessAnimationUsingKeyFrames>
  </Storyboard>
</Page.Resources>

<Rectangle x:Name="BurialMoundRect1" Fill="#FFF4F4F5" HorizontalAlignment="Left"
  Height="7" Margin="50,30,0,0" Stroke="Black" VerticalAlignment="Top"
  Width="4" Opacity="0" StrokeThickness="0"
  MouseDown="BurialMoundRect1_MouseDown"
  TouchDown="BurialMoundRect1_TouchDown"/>
<Image Source="Images\BurialGround\BurialMoundScrolls\
  finBurialMoundScrollInfo.png" x:Name="ScrollImageDisplayer"
  HorizontalAlignment="Right" Height="58" Margin="0,-56,0,0"
  VerticalAlignment="Top" Width="25" StretchDirection="DownOnly"
  Stretch="Fill" IsHitTestVisible="False" Panel.ZIndex="10"/>
<!--From -56 to -8 (Not visible to visible)-->

```

Sivulle voidaan määritellä resursseja, joita ei alustusvaiheessa vielä ajeta. Ne jäävät muistiin talteen, josta niitä voidaan halutessa kutsua. Tässä kääron avausanimaatio, jota käytetään C#-koodin puolelta.

Kalmiston napit eivät olleet vielä nappeja, vaan niiden sijasta käytettiin Rectangleja (neliöitä), joihin oli liitetty tilannekäsittelijät. Sekä hiiren painallukselle että kosketuksen tunnistukselle on tehty erilliset käsittelijät. Mukana myös kääro kuvan määrittely ja kommentti, jossa muistutuksena kuinka paljon margin-arvoa täytyy muuttaa saadakseen kääron näkymään sopivasti.

C#:

```

//Intillä valitaan kuva, tarkistetaan onko nykyistä, jos on, ensin ajetaan
piilotus, sitten vasta vaihdetaan kuva jne
private void hideInfoScrollStoryboard(int numberOfScroll)
{
    prevScrollNbr = infoScrollNumber;
    infoScrollNumber = numberOfScroll;
    if (!ScrollImageDisplayer.Margin.ToString().Equals("0,-56,0,0"))
    {
        InfoScrollStoryboard = this.FindResource("Margin_OnTextScrollClose")
            as Storyboard;
        Storyboard.SetTarget(InfoScrollStoryboard, this.ScrollImageDisplayer);
        InfoScrollStoryboard.Completed += InfoScrollCloseStoryboard_Completed;
        InfoScrollStoryboard.Begin(this.ScrollImageDisplayer);
    }
    else if (infoScrollNumber != 42)
    {
        if (infoScrollNumber == 5)
        {
            this.Tweezers.Source = new BitmapImage(new Uri("pack://application:
                ,,/MaahanKatketyt;component/Images/UIButtonImages/Tweezers.png"));
        }
        else
        {
            this.Tweezers.Source = null;
        }
        InfoScrollStoryboard = this.FindResource("Margin_OnTextScrollOpen")
            as Storyboard;
        Storyboard.SetTarget(InfoScrollStoryboard, this.ScrollImageDisplayer);
        InfoScrollStoryboard.Begin(this.ScrollImageDisplayer);
    }
}
}

```

Käärökuvan vaihtofunktio. Funktioon syötetään ensin "napin" (Rectangle) painalluksen mukaan numero. Sitä seuraa tarkastus, onko kääro jo piilotettuna. Jos näin ei ole, se piilotetaan. Ohjelma käy tämän jälkeen muissa funktioissa ja palaa sittemmin uudestaan tarkastamaan, pitääkö mitään kääroä näyttää ja jos pitää, laittaa se halutun kääron näkyviin. Tässä versiossa huonoa oli koodiin jääminen kiertämään silmukkaa, kunnes puoli sekuntia kestävä sulkemisanimaatio ehti suorittaa itsensä.

5.2.5 GraveDiggingIntroPage.xaml, GraveDiggingIntroPage.xaml.cs

XAML:

```

<Image x:Name="GraveDiggingIntroImage1"
Source="Images\BurialGround\BurialMoundGrave\GraveDigging\finGraveDig01.jpg"
Stretch="Fill" Panel.ZIndex="1"
MouseDown="GraveDiggingIntroImage1_MouseDown"
TouchDown="GraveDiggingIntroImage1_TouchDown"
MouseUp="GraveDiggingIntroImage1_MouseUp"
TouchUp="GraveDiggingIntroImage1_TouchUp"/>
<Image x:Name="GraveDiggingIntroImage2" Stretch="Fill" Panel.ZIndex="2"
MouseDown="GraveDiggingIntroImage2_MouseDown"
TouchDown="GraveDiggingIntroImage2_TouchDown"

```

```
MouseUp="GraveDiggingIntroImage2_MouseUp"
TouchUp="GraveDiggingIntroImage2_TouchUp"/>
```

Kaivaussivun toiminnoista esitellään keino, jolla toteutettiin kuvien jouheva sulautuminen toisiinsa. XAML-koodissa alustetaan kaksi kuvaa, joille määritellään z-akselin arvot kolmiulotteisessa avaruudessa (Panel.ZIndex). Tätä arvoa muuttamalla haluttua käyttöliittymän elementtiä saadaan siirrettyä z-akselilla lähemmäs katsojaa tai katsojasta pois päin. Koska tässä tapauksessa GraveDiggingIntroImage1:n z-akselin arvo on 1 ja GraveDiggingIntroImage2:n arvo on 2, päällimmäisenä – ja ainoana näkyvässä – on GraveDiggingIntroImage2. Käyttötavan esittely jatkuu C#-koodissa.

C#:

```
private void GraveDiggingIntroImage2_MouseDown(object sender,
        MouseButtonEventArgs e)
{
    int currentIndexInArray = getNextImage(new
        Uri(GraveDiggingIntroImage2.Source.ToString()).LocalPath);

    if (currentIndexInArray < ImageLocationList.Count - 1)
    {
        GraveDiggingIntroImage1.Source = new BitmapImage(new
            Uri(ImageLocationList[currentIndexInArray + 1],
                UriKind.Absolute));

        fadeOutStoryboard = this.FindResource("Opacity_OnChangeFrom") as
            Storyboard;
        Storyboard.SetTarget(fadeOutStoryboard,
            this.GraveDiggingIntroImage2);
        fadeOutStoryboard.Completed += ImageChangeFrom2To1_Completed;
        fadeOutStoryboard.Begin(this.GraveDiggingIntroImage2);
    }
}
```

Kun hiirellä painetaan GraveDiggingIntroImage2:a asetetaan GraveDiggingIntroImage1:n lähdekuvaksi seuraavana haluttu kuva. Lähde tarkistetaan erillisellä funktiolla, joka yksinkertaisesti hakee kuvien sijaintilistasta nykyisen kuvan listaindeksiin lisäten seuraavan kuvan sijainnin ja asettaa sen lähteeksi. Sen jälkeen aloitetaan animaatio, joka manipuloi läpikuultavuusarvoa yhdestä noltaan. Tällöin GraveDiggingIntroImage1:n ollessa GraveDiggingIntroImage2:n taustalla, näytävät siinä olevat eroavuudet ilmestyvän hiljalleen näkyviin.

```
private void ImageChangeFrom2To1_Completed(object sender, EventArgs e)
{
    fadeOutStoryboard.Completed -= ImageChangeFrom2To1_Completed;

    Panel.SetZIndex(GraveDiggingIntroImage1, 2);
    Panel.SetZIndex(GraveDiggingIntroImage2, 1);
    GraveDiggingIntroImage2.Opacity = 1;
}
```

}

Kun animaatio on valmis, asetetaan kuvien z-akselin arvot päinvastaisiksi (ja poistetaan nyt ylimääräiseksi jäänyt käsittelijä). Näin kuvan 2 läpikuultavuusarvo voidaan muuttaa jo takaisin täysin näkyväksi seuraavaan kuvanvaihtokierrosta varten. Kuvaa 1 painettaessa tapahtuma on sama, paitsi GraveDiggingIntroImage2 ja GraveDiggingIntroImage1 vaihtavat koodissa paikkojansa. Vastaavat tapahtumaseuraajat oli tehtynä myös kosketuskäytölle erikseen.

5.2.6 GraveMainPage.xaml, GraveMainPage.xaml.cs ja rahakätköösio

Kaivausosion esinevalikkosivu ei sisällä edellisiin sivuihin verrattuna erikoisia tai juurikaan erilaisia menetelmiä tässä ohjelman versiossa. Sivulla on käytännössä vain nappeja, jotka vaihtavat näkyvää esinettä eli kuvaa.

Sama koskee rahakätköösion sivuja MoneyDiggingIntroPage ja MoneyMainPage. Molemmat ovat toiminnaltaan samanlaisia kuin kalmiston kaivuusosion vastaavat. Korjausten, lisäysten ja muutosten jälkeen valmiiksi saatu, kappaleessa 7.1 tarkasteltava ohjelman viimeinen versio sisältää loput käsiteltävästä koodista.

6 KORJAUKSET, LISÄYKSET JA MUUTOKSET

Versio 0.9.7.1 lähetettiin museolle testattavaksi. Erittäin tärkeää tämä oli omaehtoisen ulkoasunmuokkausten tarkistuksen lisäksi siksi, ettei tekijällä ollut käytössään kosketusnäyttöä. Ennen koekäyttöä kosketusnäyttötoimintojen ohjelmointi tapahtui lähinnä päättelemällä ja C#- sekä XAML-kielien referenssisivustoja tutkimalla. Museolta saadun palautteen perustella ongelmia alettiin korjata.

Museon henkilökunta lähetti listan löydetyistä vioista ja uusia ehdotuksia ohjelman parantamiseksi. Moni ongelmista koski tapauksia, jotka voitiin korjata pikaisesti muuttamalla koodiin asetettuja arvoja tai määreitä. Esimerkiksi rahakätkön kaivuusion intron katsottiin etenevän liian hitaasti, joten animoinnin nopeutta nostettiin kolmesta puoleentoista sekuntiin. Osa taas vaati esimerkiksi tuplatarkistuksia ennen tapahtumaa tai tapahtuman ajon tai kosketusrekisteröinnin estämistä tietyssä vaiheessa. Joitakin merkittäviä tapauksia kuitenkin oli, jotka pysyivät aina ohjelman viimeiseen versioon asti. Riippuen ongelman laadusta, tasosta tai kiinnostavuudesta se käsitellään joko lyhyesti selostaen tapaus ja korjaustapa tai seikkaperäisemmin koodeineen.

6.1 Ohjelmistosta löydetyt viat ja niiden korjaaminen

Etusivun taustassa (kuvio 15) havaittiin virhe, joka oli jäänyt huomaatta sen tekijöiltä. Virhe voitiin havaita vain katsomalla näyttöä tarpeeksi sivusta tai kääntämällä kuvan väritys käänteiseksi. Ratkaisuna museon henkilökunta teki kuvasta uuden version.

Tervakankaan kalmiston valikkosivun toiminta oli jo sen verran varmaa, ettei korjattavaa ollut. Sen sijaan sivun lopulliseen versioon tuli olennaisia muutoksia, joista kerrotaan myöhemmissä luvuissa.

Kalmiston kaivausosuudessa ongelmana oli käärotekstien ilmestyminen väärissä kohdissa, käärön ilmestyminen pomppien suoraan näkyviin ilman animaatiota tai jättäytyminen täysin ilmestymättä. Tähän auttoi hieman käyttäjän kosketuksen rekisteröinnin esto käärön animaation ollessa kesken, tarkastusten pieni viilaaminen sekä käärön ilmestymis- ja poistumisnopeuden

nostaminen. Kun käyttäjä pystyi vaihtamaan kuvia niin nopeasti kuin halusi, animaation jälkeen tapahtuvaksi ajastettu infotekstin vaihdon uusintatarkistus ei toiminut. Tarkistukset oli tehty niin, että kääro vaihdettiin vain tietyn kuvan kohdalla ja jätettiin näkyviin, kunnes seuraava vaihtokuvaksi määritelty kuva tulisi vastaan. Jos käyttäjä ehti pyyhkäistä kuvia vaihtokuvaa pidemmälle ei infokääroä enää ladattu ollenkaan.

```
else if (infoTextChangeCountdown == 99 & currentDisplayedImage1.Contains
                                                ("GraveDig05.jpg"))
    {
        infoTextChangeCountdown = 2;
        currentInfoTextContent = languagePrefix + "GraveDiggingInfoText3";
        ChangeTexts();
        infoTextIsOpening = true;
        ShowInfoTextAnimation();
    }
```

Vaihtotarkistukseksi lisätty laskuri, jota vähennetään aina käyttäjän kosketus havaittaessa (jos kosketuksen tunnistusta ei sillä hetkellä ole estetty).

Samantapainen ongelma oli myös rahakätkön kaivuuosiossa, joka korjattiin vastaavahkoilla tarkistuksilla ja varmistuksilla.

Kalmiston esinevalikko ja rahakätkön rahavalikko toimivat muuten hyvin, mutta niissäkin huomattiin outoja kosketustapahtumiin liittyviä ongelmia. Kun käyttäjä painoi esinesivulla ollessaan Takaisin-nuolta päästäkseen takaisin valikkoon, siirtyi ohjelma suoraan takaisin pääsivulle. Syyksi epäiltiin, että käyttäjän painallus viipyi liian pitkään Takaisin-nuolen päällä ja ohjelma rekisteröisi kosketuksen vielä rahavalikkosivulle siirtyessäänkin, jossa samalla kohdalla on pääsivulle siirtävä nappi (kts. kuviot 47 ja 48). Ajateltiin, että tässäkin täytyy estää kosketuksen toiminta joksikin aikaa. Tämä ongelma osoittautui erittäin tärkeäksi ymmärtää koko ohjelman toiminnan kannalta. Ongelmaan ei keksitty ratkaisua eikä ongelman aiheuttajaa löydetty kuin vasta viimeisessä ohjelman versiossa, kun kosketusnäyttöä oli päästy paikan päällä kokeilemaan omin käsin.



KUVIO 47. Pattijoen keskiaikainen rahakätkö, kaivuuosio, version 1.1.0.3 raha 4



KUVIO 48. Pattijoen keskiaikainen rahakätkö, kaivuuosio, version 1.1.0.3 rahavalikko

Kosketusnäytön puuttuminen jatkuvasta testiympäristöstä oli huono idea. Näyttö olisi auttanut ohjelmoijaa huomaamaan erikoisuuden jo paljon aiemmin ja turhalta pohdinnalta ja työltä olisi varmasti vältytty. Nyt ongelma huomattiin vasta ohjelmistoa ja järjestelmää museolla asentaessa. Tässä aiemmin esiteltyä koodia kaivausosiosta:

...

```
<Image x:Name="GraveDiggingIntroImage1"
Source="Images\BurialGround\BurialMoundGrave\GraveDigging\finGraveDig01.jpg"
Stretch="Fill" Panel.ZIndex="1"
MouseDown="GraveDiggingIntroImage1_MouseDown"
TouchDown="GraveDiggingIntroImage1_TouchDown" ...
```

Huomaa erilliset tapahtumaseuraajat sekä hiiren napin painallukselle että kosketukselle. Tästä syystä jokaikisessä kohdassa, jossa samaa toiminnanvarmistusta käytettiin, aiheutui joka kosketuskerta kaksi kosketusta. Konkreettisen testaamisen puuttuessa tehtiin virhepäätelmä hiiri- ja kosketustoimintojen toiminnasta. Todellisuudessa aina tehtäessä hiiripainikkeen toimintaseuraaja se toimii myös kosketuksella. Asia selvisi, kun testattiin viimeiseen versioon tehtyä lisätoimintoa, jolla kaivauksen esinesivuja ja rahakätkön rahasisivuja voitiin selailla esineestä tai rahasta toiseen ilman palaamista valikkosivulle. Selaillessa rahasisivuja huomattiin ohjelman hyppäävän aina yhden kuvan yli ja avatun rahan järjestysluvusta riippuen osui aina vain joko parillisille tai parittomille sivuille. Johtopäätöksenä sittemmin todettiin, että ainoa syyllinen käyttäjän tai puuttuvien viiveiden sijasta voi olla automaattisesti tapahtuva kaksoiskosketus.

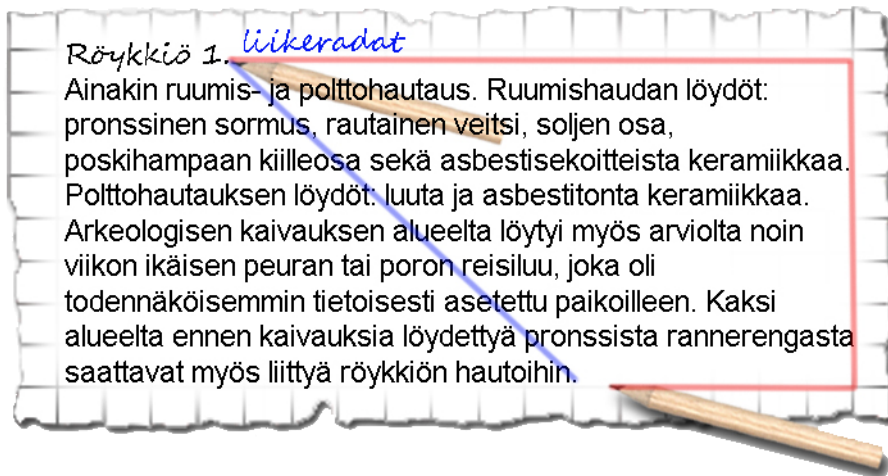
6.2 Toivotut muutokset, lisäykset tai lisätoiminnot

Museon henkilökunta toivoi muutoksia erityisesti kuviin ja ääniin. Kuvissa oli sensorikalvon käyttöä varten ajatellut käyttövihjeet, joissa käyttäjää ohjeistettiin osoittamaan haluamaansa kohtaa. Mietittiin neuvon sanamuodon vaihtamista kosketukseksi tai ohjeistuksen poistamista kokonaan, sillä kosketusnäytön käyttö on jo hyvin tuttua asiakkaalle kuin asiakkaalle. Kuvissa oli myös kirjoitusvirheitä jotka tulisi korjata. Erityisesti englanninkieliset tekstit vaativat korjaamisia niiden arkeologisen sanaston virheellisten käännösten vuoksi. Työn tekijä koetti parhaansa mukaan tehdä pieniä korjauksia, kuten kuvaelementtien siirtoja vastaamaan muiden sivujen vastaavia kohtia. Vaikkakin muutokset olivat pieniä ne olisivat lopulta vaatineet niin paljon aikaa, että museon henkilökunnan kanssa sovittiin heidän tekevän haluamansa jäljelle jääneet muutokset. Tätä työtä pyrittiin helpottamaan muuttamalla kääröpohjaiset infotekstit ohjelmoimalla tehdyiksi pohjiksi, joihin tekstit haettaisiin tavallisista editoitavista tekstitiedostoista.

Kääröpohjaisia infotekstejä kritisoitiin muutenkin. Ideasta sinänsä pidettiin, mutta kääröä itseään ei nähty aitoon tilanteeseen sopivaksi eikä liittyväksi. Todellisessa tilanteessa teksti olisi kirjattu ylös tavalliselle ruutupaperille. Lisäksi käärölle kirjoitetun tekstin katsottiin olevan turhan vaikealukuista heikkonäköiselle tai näkövammaiselle asiakkaalle.

Museolta saatiin esimerkki ehdotuksesta uudeksi infotekstipohjaksi (kuvio 49) ja ajatus tekstin animoinnista niin, että näyttäisi kuin joku olisi juuri kirjoittamassa sitä ylös. Tätä koetettiin

saavuttaa ohjelmoimalla, mutta todettiin lopulta liian vaikeaksi toteuttaa käytettävissä olevan ajan puitteissa.



KUVIO 49. Esimerkki uusien infotekstien esitystavalle

Kirjoittamista esittäviä ääniefektejä toivottiin infotekstien uuden version taustalle, mutta koska tekstilaatikoita ei saatu tehtyä kirjoitustyyliillä ei myöskään toiminnolle tehty erityisiä ääniefektejä. Muita haluttuja lisäefektejä olivat rahakätkön kaivuusiosion ja rahavalikossa rahojen käsittelyyn tulevat kolikkoäänät. Kaivuusiosioon haluttiin "artaan" löytymiseen jokin lisäefekti ja valikkosivulle kilahdus kun käyttäjä avaa jonkin rahasisivun.

Taustäänien lisäksi kaivaus- ja kaivuusosioissa oli jo palautetta odotellessa ehditty tehdä toiminnaltaan sellaiset äänitehosteet, joista alkuääni soitettiin käyttäjän koskettaessa näyttöä ja loppuääni käyttäjän nostaessa kätensä pois näytöltä. Efekti oli suunniteltu niin, että kaivamissimulaatioiden aikana asiakas tulisi kuulemaan lapioinnin, kuokkimisen, laastilapioinnin ja harjaamisen ääniä tilanteita esittävien kuvien kanssa.

Infotekstin kirjoitusanimaation lisäksi toivottiin myös muita lisäanimaatioita. Kaivausosion kuvissa näkyvien työkalujen käyttöönottamisen mahdollisuutta mielittiin, sillä erityisesti lapsiasiakkaat yrittivät ohjelmaa testatessaan tarttua niihin kuin käyttääkseen niitä. Tämä olisi kuitenkin vaatinut ensin kahdeksantoista kuvan muokkaamista työkaluttomiksi ja sen jälkeen erillisten työkalukuvien ohjelmoimista käyttöön otettaviksi. Toinen ekstra-animaatiopyyntö koski rahavalikkosivun elävöittämistä. Sivulle oli suunniteltu animaatio, jossa kolikot putoaisivat kuvan ylälaidasta ja käyttäjä voisi koettaa ottaa niitä kiinni. Toiveita ei kuitenkaan ehditty toteuttaa projektin pitämiseksi aikataulussa.

Muutokset, jotka ehdittiin toteuttaa olivat halutut äänitehostelisäykset ja infotekstien uudelleenohjelmointi. Projektin kuluessa oli toivottu myös raha- ja esinesivujen selaustoimintoa ja mahdollisena lisänä palautumista päävalikkoon, jos käyttö keskeytyy ja käyttäjä jättää ohjelman jollekin muulle sivulle. Toteutusta helpotti niiden halutuiksi tietäminen jo aikaisessa vaiheessa. Näiden toteuttamistavasta lisää seuraavassa osiossa.

6.3 Tehdyt muutokset toimintoihin

Kääröjen käytöstä jääminen ja niistä löydettyjen kirjoitusvirheiden vuoksi ajateltiin, ettei olisi järkevää tehdä enää infotekstien muokkaamisesta niin staattista. Jos kuvaan olisikin eksynyt kirjoitusvirhe, olisi kyseinen kuva täytynyt tehdä taas täysin uusiksi. Päätettiin tehdä ohjelmoimalla infotekstille sellainen menetelmä, jossa sivun alustuksessa ladattaisiin erillisestä tekstitiedostosta halutut infotekstit taulukkoon ja liitettäisiin ne vastaavalla numerolla avautuvaan infotekstipohjaan. Korjaamista varten ei tarvitsisi kuin avata muokkausta vaativa tekstitiedosto. Ohjelmointi aloitettiin tutkimalla miten WPF-C# -yhdistelmällä luku tiedostosta -operaatio suoritetaan. Testaamisen jälkeen tutkittiin, josko tiedostosta luettaessa voitaisiin jaotella teksti järkevästi niin, että tekstiin tehtävät muutokset eivät haittaisi ohjelman suoritusta millään tavalla. Tuloksena saatiin seuraavanlainen lukufunktio:

```
//Creates an array out of required language info texts
private String[] TextChangeOperation(string fileToUse)
{
    char endChar = '£';
    char startChar = '@';
    String[] arrayOfLines = null;
    try
    {
        using (StreamReader streader = new StreamReader(fileToUse,
                                                       Encoding.Default))
        {
            String line = streader.ReadToEnd();
            arrayOfLines = line.Split(startChar, endChar);

            var temp = new List<string>();
            foreach (var s in arrayOfLines)
            {
                if (!string.IsNullOrEmpty(s))
                    temp.Add(s);
            }
            arrayOfLines = temp.ToArray();
        }
    }
    catch (Exception e)
    {
```

```

        System.Diagnostics.Debug.WriteLine("The file could not be read");
        System.Diagnostics.Debug.WriteLine(e.Message);
    }
    return arrayOfLines;
}

```

Tekstitiedoston sisältö taas näyttää seuraavalta:

```
@finBurialMoundTextHeader1f@Röykkiö 1f
```

```
@finBurialMoundTextContent1f@Ainakin ruumis- ja polttohautaus.
```

Ruumishaudan löydöt: pronssinen sormus, rautainen veitsi, soljen osa, poskihampaan kiilleosa sekä asbestisekoitteista keramiikkaa.

Polttohautauksen löydöt: luuta ja asbestitonta keramiikkaa. Alueelta löytyi myös noin viikon ikäisen peuran tai poron reisiluu, joka oli luultavasti asetettu tietoisesti paikoilleen.

Kaksi alueelta ennen kaivauksia löydettyä pronssista rannerengasta saattavat myös liittyä röykkiön hautoihin.f

Tekstitiedostoon on siis määritelty symbolit, joiden väliltä funktio hakee merkit ja tallentaa ne yhteen taulukon elementeistä. Tämän avulla saadaan ensin luotua lajittelematon taulukko, joka sisältää kaikki tiedoston sisältämät rivit. Sen jälkeen taulukko järjestellään luomalla väliaikainen merkkijonolista, josta poistetaan kaikki tyhjät elementit. Näin jäljelle jää seuraavanlainen taulukko:

finBurialMoundTextHeader1
Röykkiö 1
finBurialMoundTextContent1
<p>Ainakin ruumis- ja polttohautaus.</p> <p>Ruumishaudan löydöt: pronssinen sormus, rautainen veitsi, soljen osa, poskihampaan kiilleosa sekä asbestisekoitteista keramiikkaa.</p> <p>Polttohautauksen löydöt: luuta ja asbestitonta keramiikkaa. Alueelta löytyi myös noin viikon ikäisen peuran tai poron reisiluu, joka oli luultavasti asetettu tietoisesti paikoilleen.</p> <p>Kaksi alueelta ennen kaivauksia löydettyä pronssista rannerengasta saattavat myös liittyä röykkiön hautoihin.</p>

Taulukosta voidaan näinollen etsiä aina haluttu teksti, esimerkiksi otsikko ensimmäistä rökkiölinkkiä painettaessa hakusanalla "finBurialMoundTextHeader1" ja lisäämällä kyseisen tekstisarakkeen sijaintiarvoon yksi, jolloin teksti haetaan seuraavasta sarakkeesta:

```
private int getHeaderOrContent(string requiredHeaderOrContent)
{
    return Array.IndexOf(BurialMainInfoTextArray, requiredHeaderOrContent)+1;
}

private void ChangeTexts()
{
    if (currentHeader != null)
    {
        this.InfoTextHeaderTextBlock.Text =
            BurialMainInfoTextArray[getHeaderOrContent(currentHeader)];
    }
    else
    {
        this.InfoTextHeaderTextBlock.Text = null;
    }

    if (currentContent != null)
    {
        this.InfoTextContentTextBlock.Text =
            BurialMainInfoTextArray[getHeaderOrContent(currentContent)];
    }
    else
    {
        this.InfoTextContentTextBlock.Text = null;
    }
}
```

Kun ohjelmaan vielä lisättiin automaattisesti tekstin pituuden mukaan säätävät tekstilaatikat, saatiin aikaiseksi kuvion 50 mukainen infotekstielementti, jonka muokkaamiseen ei tarvitse osata ohjelmoida tai käyttää photoshoppia. Samaa tekniikkaa käytettiin myös kalmiston kaivausosion infotekstikääröt korvattaessa.



KUVIO 50. Kalmiston infolaatikko, Röykkiö 4, versio 1.1.0.3

Rahakätkön rahoja ja kalmiston kaivauksen esineitä haluttiin pystyä selaamaan siirtyen aina seuraavaan esineeseen tai rahaan ilman, että käyttäjän täytyy palata valikkoon. Vanhan järjestelmän tilalle tehtiin systeemi, johon kuuluivat kolme nuolinappia. Nuolinappeja painamalla pääsi joko edelliseen tai seuraavaan rahaan/esineeseen tai takaisin valikkoon.



KUVIO 51. Suurennos Pattijoen keskiaikaisen rahakätkön nuolinäppäimistä, versio 1.1.0.3

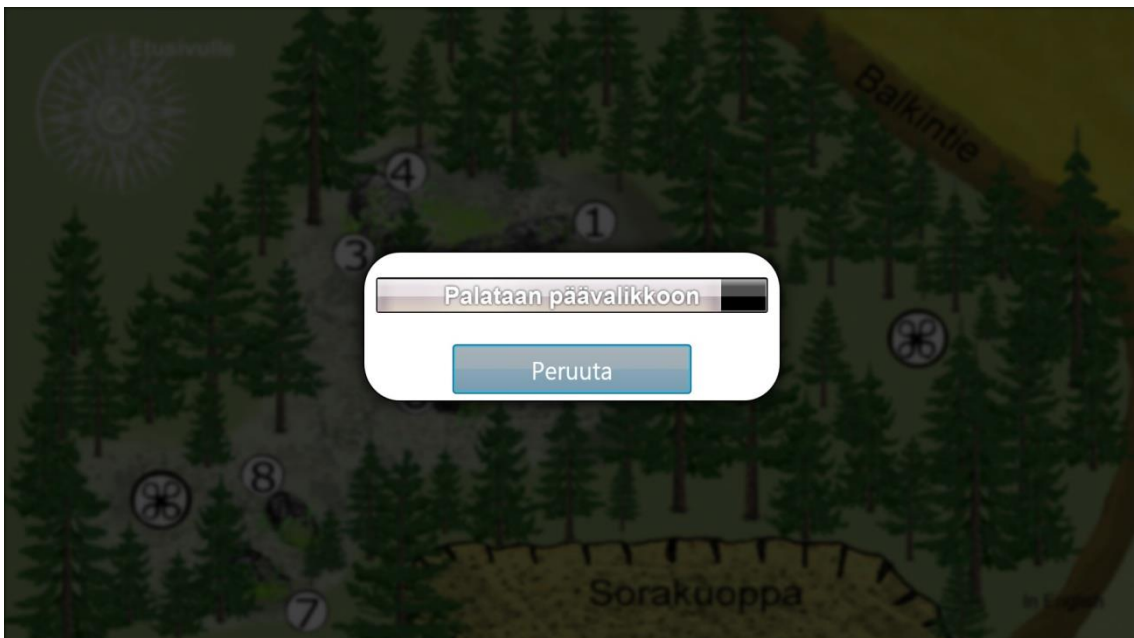
```
private void NextItemArrow_MouseDown(object sender, MouseButtonEventArgs e)
{
    resetbackToMainFromGraveMainTimer();
    if (currentItemIndex <= 6)
    {
        System.Diagnostics.Debug.WriteLine("currentItemIndex: " +
                                           currentItemIndex);

        this.GraveMainImageHandling.Source = new BitmapImage(new
        Uri(GraveImageLocationArray[currentItemIndex + 1], UriKind.Absolute));
        currentItemIndex++;
    }
    else
    {
        this.GraveMainImageHandling.Source = new BitmapImage(new
        Uri(GraveImageLocationArray[0], UriKind.Absolute));
        currentItemIndex = 0;
    }
}
```

}

Uutta koodia toiminto ei paljoa vaatinut, vain kolme lisänappia ja ylläolevan leikkeen verran kuvanvaihtojen manipulaatioita. Ylimääräistä koodia on lähinnä toiminnan tekeminen karusellimäiseksi, eli viimeisen kuvan jälkeen Seuraava-nuolta painettaessa siirrytään esine- tai rahakuvien ensimmäiseen kuvaan.

Ohjelman palauttaminen pääsivulle oli tuplakosketusongelman korjaamista lukuun ottamatta viimeinen ohjelmoitu funktio. Tarkoituksena oli siis tehdä toiminto, joka automaattisesti tietyn ajan kuluttua palautuu päävalikkoon, jotta ohjelma on alkutilassaan seuraavalle käyttäjälle. Tällöin seuraavan käyttäjän ei tarvitse ihmetellä miten ohjelma toimii, jos ohjelman suoritus on jäänyt kesken esimerkiksi kalmiston kaivausosiossa. Suunnitteluvaiheessa museon henkilökunnan kanssa mietittiin sopivaa pituutta odotusajalle. Ajateltiin, että liian lyhyt aika ärsyttäisi käyttäjää jos hän olisi vielä halunnut jatkaa ohjelmassa etenemistä, mutta olisi unohtunut keskustelemaan itse ohjelman aiheesta. Ajaksi suunniteltiin kymmenestä-viiteentoista minuuttia. Ohjelmoija sai kuitenkin kehitystä aloittaessaan idean, jossa siirtymisestä varoitettaisiin käyttäjää ja annettaisiin hänelle mahdollisuus estää se. Aika voitiin siten jättää lyhyemmäksi.



KUVIO 52. Tervakankaan rautakautinen kalmisto, versio 1.1.0.3, esimerkki paluulaatikosta

Paluulaatikon ulkonäön täytyi olla sellainen, että se herättää käyttäjän huomion. Suunnitelmaksi muotoutui sumentaa ja himmentää taustaa, visualisoida jäljellä oleva perumisaika käyttäjälle sekä

saada aikaan Peruuta-napin sykkivä ulkomuoto. Nämä suoritettiin laatimalla asianmukaiset XAML-animaatiot.

```
<Storyboard x:Name="CancelProgressBarAnimation" x:Key="Value_From100To0">
    <DoubleAnimation Storyboard.TargetName="CancelTimedReturnProgressBar"
        Storyboard.TargetProperty="Value" Duration="0:0:10" From="100"
        To="0" FillBehavior="HoldEnd">
    </DoubleAnimation>
</Storyboard>
<Storyboard x:Name="CancelTimedReturnFadeIn" x:Key="Opacity_ToView">
    <DoubleAnimation Storyboard.TargetName="CancelTimedReturnBorder"
        Storyboard.TargetProperty="Opacity" AccelerationRatio="0.2"
        DecelerationRatio="0.5" Duration="0:0:0.5" From="0" To="1">
    </DoubleAnimation>
</Storyboard>
<Storyboard x:Name="CancelTimedReturnFadeOut" x:Key="Opacity_FromView">
    <DoubleAnimation Storyboard.TargetName="CancelTimedReturnBorder"
        Storyboard.TargetProperty="Opacity" AccelerationRatio="0.2"
        DecelerationRatio="0.5" Duration="0:0:0.5" From="1" To="0">
    </DoubleAnimation>
</Storyboard>

<Storyboard x:Name="MainImageFadeOut" x:Key="Opacity_MainImageOut">
    <DoubleAnimation Storyboard.TargetName="GraveMainImageHandling"
        Storyboard.TargetProperty="Opacity" AccelerationRatio="0.2"
        DecelerationRatio="0.5" Duration="0:0:0.5" From="1" To="0.25">
    </DoubleAnimation>
    <DoubleAnimation Storyboard.TargetName="blurGraveMainImage"
        Storyboard.TargetProperty="Radius" Duration="0:0:0.5" From="0"
        To="1">
    </DoubleAnimation>
</Storyboard>
<Storyboard x:Name="MainImageFade" x:Key="Opacity_MainImageIn">
    <DoubleAnimation Storyboard.TargetName="GraveMainImageHandling"
        Storyboard.TargetProperty="Opacity" AccelerationRatio="0.2"
        DecelerationRatio="0.5" Duration="0:0:0.5" From="0.25" To="1">
    </DoubleAnimation>
    <DoubleAnimation Storyboard.TargetName="blurGraveMainImage"
        Storyboard.TargetProperty="Radius" Duration="0:0:0.5" From="1"
        To="0">
    </DoubleAnimation>
</Storyboard>
```

Animaatiot valmisteltiin sivun resursseihin ja laukaistaisiin, kun haluttu aika olisi kulunut. Ajastus ohjelmoitiin C#:llä. Ajastimen alustus tapahtuu seuraavasti:

```
private void resetbackToMainFromGraveMainTimer()
{
    backToMainFromGraveMainTimer.Stop();
    backToMainFromGraveMainTimer = null;
    BackToMainMenuFromGraveMainTimerStart();
}

private void BackToMainMenuFromGraveMainTimerStart()
```

```

{
    if (backToMainFromGraveMainTimer == null)
    {
        backToMainFromGraveMainTimer = new DispatcherTimer();
    }
    backToMainFromGraveMainTimer.Interval = TimeSpan.FromMinutes(5);
    backToMainFromGraveMainTimer.Tick += backToMainFromGraveMainTimer_Tick;
    backToMainFromGraveMainTimer.Start();
}

private void backToMainFromGraveMainTimer_Tick(object sender, EventArgs e)
{
    backToMainFromGraveMainTimer.Tick -= backToMainFromGraveMainTimer_Tick;
    backToMainFromGraveMainTimer.Stop();

    this.TextButtonGrid.IsHitTestVisible = false;
    this.ItemButtonGrid.IsHitTestVisible = false;
    this.GraveMainImageHandling.IsHitTestVisible = false;
    Panel.SetZIndex(CancelTimedReturnBorder, 10);

    Storyboard fadeBackground = this.FindResource("Opacity_MainImageOut")
    as Storyboard;
    fadeBackground.Begin();

    cancelTimedReturn = this.FindResource("Opacity_ToView") as Storyboard;
    Storyboard.SetTarget(cancelTimedReturn, this.CancelTimedReturnBorder);
    cancelTimedReturn.Completed += cancelTimedReturn_Completed;
    cancelTimedReturn.Begin(this.CancelTimedReturnBorder);
}

private void cancelTimedReturn_Completed(object sender, EventArgs e)
{
    cancelTimedReturn.Completed -= cancelTimedReturn_Completed;
    this.CancelTimedReturnButton.IsHitTestVisible = true;
    Keyboard.Focus(CancelTimedReturnButton);

    progressBarAnimation = this.FindResource("Value_From100To0")
    as Storyboard;
    progressBarAnimation.Completed += progressBarAnimation_Completed;
    progressBarAnimation.Begin();
}

void progressBarAnimation_Completed(object sender, EventArgs e)
{
    progressBarAnimation.Completed -= progressBarAnimation_Completed;

    fadeOutStoryboard = this.FindResource("Opacity_OnNavigationFrom")
    as Storyboard;
    Storyboard.SetTarget(fadeOutStoryboard, this);
    fadeOutStoryboard.Completed += BackToMainFromGraveMainPageFade_Completed;
    fadeOutStoryboard.Begin(this);
}

```

Yllä esitellyllä koodissa suoritetaan viiden minuutin viiveen jälkeen sivuston himmennys ja sumennus, näytetään paluulaatikko, animoidaan siirtymisviive ja asetetaan näppäimistöfokus peruutusnapille, jotta se vilkkuisi. Jos peruutusnappia ei paineta, aloitetaan samat siirtymistoiminnot, jotka suoritettaisiin Pääsivulle-nappia painettaessa.

Huomaa ensimmäisenä esitelty `resetbackToMainFromGraveMainTimer`-funktio. Tätä kutsutaan aina käyttäjän koskettaessa jotain sivun elementeistä ja löytyy esimerkiksi aiemmin esitellyn nuolinäppäintöiminnön yhteydessä. Ajastuksen alustuksen täytyy tietysti olla myös kaikkien muiden nappien, kuvien ja taustakuvankin yhteydessä. Tämä tekotapa on toimiva, mutta yksinkertainen ja hieman epävarma. On helposti mahdollista, että alustus unohdetaan laittaa tehtäväksi jollekin elementille. Parempi olisi ollut tehdä tapahtumakäsittelijä, joka kuuntelisi mitä tahansa kosketuksia. Silloin ei tarvitsisi lisätä alustusfunktion ajoa mihinkään muualle. Käsittelijää ei kuitenkaan ehditty lähteä rakentamaan ja testaamaan, joten se jää mahdolliselle jatkokehittäjälle.

```
private void CancelTimedReturnButton_Click(object sender, RoutedEventArgs e)
{
    progressBarAnimation.Completed -= progressBarAnimation_Completed;
    progressBarAnimation.Stop();
    this.CancelTimedReturnProgressBar.Value = 100;

    Storyboard showBackground = this.FindResource("Opacity_MainImageIn")
    as Storyboard;
    showBackground.Begin();

    hideCancelTimedReturn = this.FindResource("Opacity_FromView")
    as Storyboard;
    Storyboard.SetTarget(hideCancelTimedReturn, this.CancelTimedReturnBorder);
    hideCancelTimedReturn.Completed += hideCancelTimedReturn_Completed;
    hideCancelTimedReturn.Begin(this.CancelTimedReturnBorder);
}

private void hideCancelTimedReturn_Completed(object sender, EventArgs e)
{
    hideCancelTimedReturn.Completed -= hideCancelTimedReturn_Completed;

    this.TextButtonGrid.IsHitTestVisible = true;
    this.ItemButtonGrid.IsHitTestVisible = true;
    this.GraveMainImageHandling.IsHitTestVisible = true;

    if (this.GraveMainImageHandling.Source.ToString().Contains("GraveMain"))
    {
        this.TextButtonGrid.Visibility = Visibility.Visible;
        this.BackToMainPageButton.Visibility = Visibility.Visible;
        PrevItemArrow.Visibility = Visibility.Hidden;
        NextItemArrow.Visibility = Visibility.Hidden;
        MenuArrow.Visibility = Visibility.Hidden;
        this.CancelTimedReturnButton.IsHitTestVisible = false;
    } else
    {
        this.TextButtonGrid.Visibility = Visibility.Hidden;
        this.BackToMainPageButton.Visibility = Visibility.Hidden;
    }
}
```

```
        PrevItemArrow.Visibility = Visibility.Visible;
        NextItemArrow.Visibility = Visibility.Visible;
        MenuArrow.Visibility = Visibility.Visible;
        this.CancelTimedReturnButton.IsHitTestVisible = false;
    }
    Panel.SetZIndex(CancelTimedReturnBorder, -1);
    resetbackToMainFromGraveMainTimer();
}
```

Tässä peruutusnapin painalluksen jälkeiset tapahtumat. Ensin aloitetaan taustan näkyviin palautus, sitten paluulaatikon häivytytys. Kun häivytytys on suoritettu, sallitaan sivusta riippuen halutut näppäimet. Lopuksi asetetaan paluulaatikko piiloon siirtäen se z-akselilla taaimmaiseksi ja alustetaan ajastin.

Päävalikkoon paluun pitäisi ehdottomasti olla tehty globaaliksi funktioksi, joka himmentäisi koko näkyvää ikkunaa erillisen sivukohtaisen manipuloinnin sijaan. Mutta kuten aiemmin on mainittu, aikataulu asetti rajoitteita ja aiheutti valikointia ohjelmiston kehitykseen.

7 VERSIO 1.1.0.3 RELEASE CANDIDATE

Projekti saatiin tekijän osalta päätökseen version 1.1.0.3 jälkeen. Versioon oli korjattu ohjelmointivirheet ja se oli ohjelmoinnin puolesta täysin käyttövalmis asiakaskäyttöön. Museon henkilökunta halusi vielä muokata käytössä olevia kuvia. Tämä on käytettyjen ohjelmointitapojen vuoksi täysin mahdollista. Tämän version oletetaan olevan teknisesti viimeinen kehitettävä versio, mutta museon editointien vuoksi varmuutta ei ole.

Projektin jatkokehitysmahdollisuudet ovat lähinnä nimelliset. Ohjelman ohjelmointia voitaisiin siistiä huomattavasti jakamalla sisältöä apuluokkiin ja muuttamalla toimintoja globaaleiksi. Näillä ei kuitenkaan olisi itse ohjelman toiminnan kannalta mitään merkitystä.

Seuraavassa alaluvussa vielä viimeiset visuaaliset muutokset, joista tekijä oli tietoinen.

7.1 Version 1.1.0.3 ulkonäkö

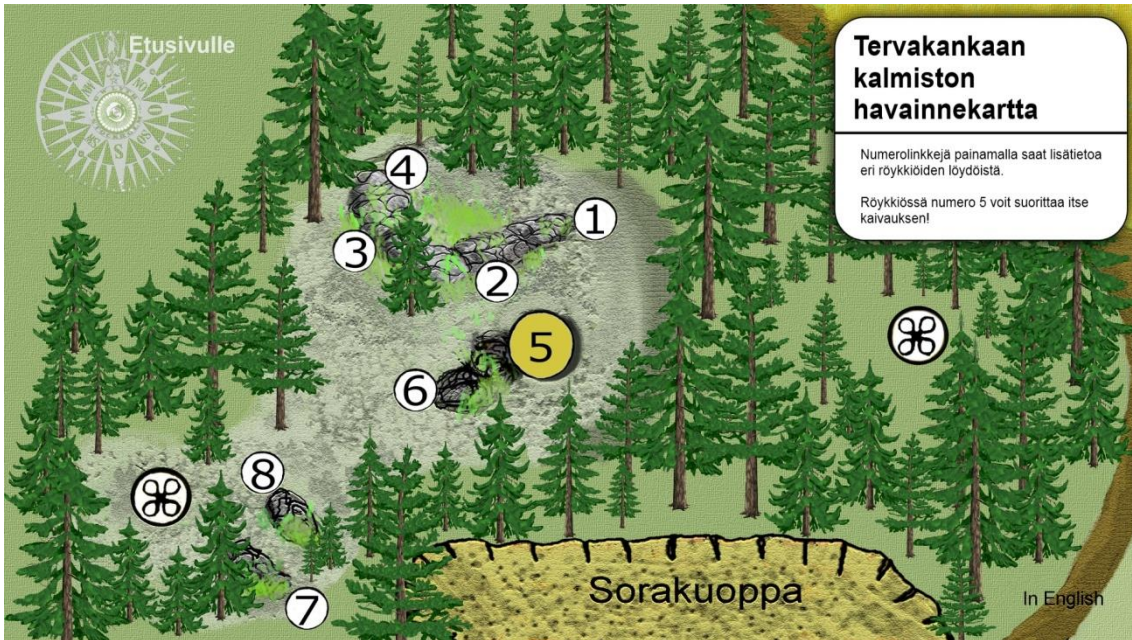
Päävalikosta oli poistettu ohjeistus ja yleisilmettä muokattu hieman.



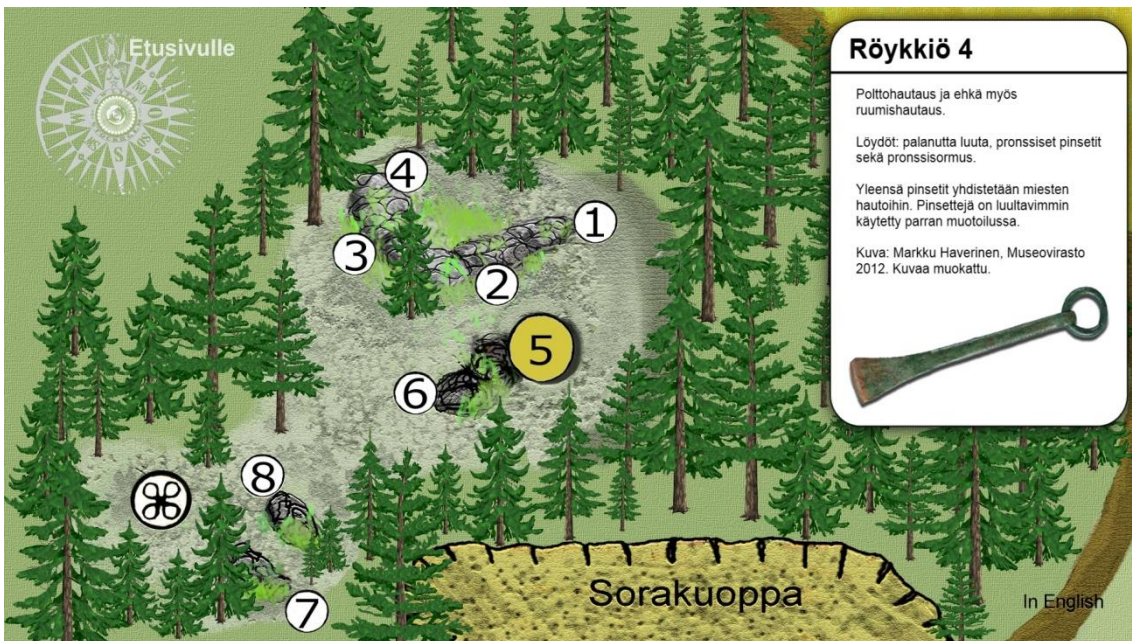
KUVIO 53. Maahan kätkeyt, version 1.1.0.3 etusivu (päävalikko)

7.2 Tervakankaan rautakautinen kalmisto

Kalmiston valikkosivuun ei tekstilaatikoita lukuun ottamatta tullut enää muutoksia versioon 0.9.7.1 jälkeen.



KUVIO 54. Tervakankaan rautakautinen kalmisto, versio 1.1.0.3, alkuinfoteksti avoinna



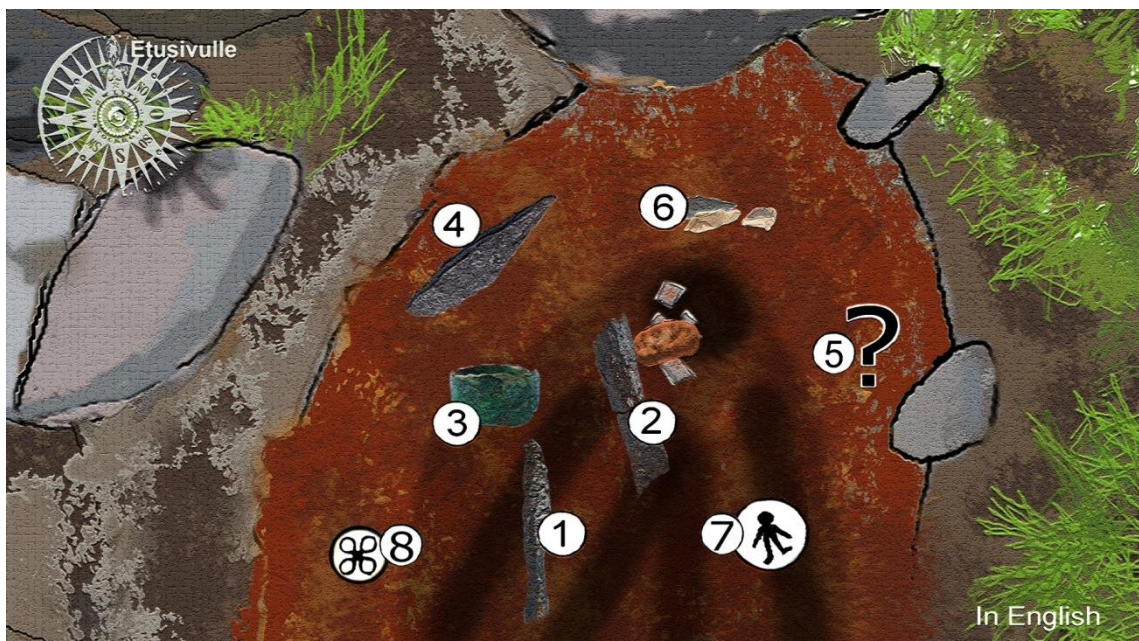
KUVIO 55. Tervakankaan rautakautinen kalmisto, versio 1.1.0.3, röykkiö 4 avoinna

Kaivausosioon oli vaihdettu käärejen tilalle kalmiston valikkosivua vastaavat tekstilaatikat.



KUVIO 56. Tervakankaan rautakautinen kalmisto, kaivausosio, versio 1.1.0.3, viimeinen kuva ennen valikkosivua

Kaivausosion valikkosivulle oli lisätty yksi linkki ja linkkien sijoittelua oli selkeytetty.



KUVIO 57. Tervakankaan rautakautinen kalmisto, kaivausosio, versio 1.1.0.3, valikkosivu

Esinesivuille oli tehty kolminuolinen navigointijärjestelmä. Kehitysajan rajallisuuden vuoksi kuviossa 58 vasemmassa ylälaudassa näkyvän vanhaan navigointisysteemiin kuuluvan Takaisin-nuolen kuvista poisto jäi museon henkilökunnalle.



KUVIO 58. Tervakankaan rautakautinen kalmisto, kaivausosio, versio 1.1.0.3, esinesivu 1

7.3 Pattijoen keskiaikainen rahakätkö

Kuvion 58 yhteydessä kuvattu Takaisin-nuoliongelma jäi korjausta vaille myös rahakätkön rahakuviin.



KUVIO 59. Pattijoen keskiaikainen rahakätkö, kaivuuosio, versio 1.1.0.3, raha 1

8 YHTEENVETO

Tässä lopputyössä ohjelmoitiin Raahen museon näyttelytilaan asiakaskäyttöön tarkoitettu interaktiivinen multimediaesitys käyttöliittymineen. Sovellus kertoo kahdesta Raahen lähialueilta tehdystä arkeologisesta löydöksestä, Tervakankaan rautakautisesta kalmistosta ja Pattijoen keskiaikaisesta rahakätköstä. Käyttöliittymä ohjelmoitiin Windows Presentation Foundation-pohjaiseksi käyttäen XAML- ja C# -kieliä. Toteutettua esitystä voidaan muokata kuvia, ääniä ja tekstityksiä vaihtaen ilman minkäänlaista ohjelmoinnin tuntemusta. Ohjelmisto on prototyypin pohjalta rakennettu, lopulliseksi versioksi valmistettu sovellus, jonka jatkokehitys ei ole tarpeellista.

Sovelluksen toimintoihin kuuluvat mm. kuvien esitys ja selaus, äänitehosteiden soitto, interaktiivinen arkeologisen kaivauksen simulointi sekä käyttäjän informointi käsillä olevasta aiheesta.

Tekijälle tämä lopputyö osoittautui erittäin mielenkiintoiseksi ja miellyttäväksi tehtäväksi. Alussa huolestuttanut ohjelmoinnin määrä osoittautui turhaksi, kun päin vastoin projektin kuluessa haluttiin lisätä ohjelmointia ja ohjelmoinnin vaativuustasoa tekemällä vapaaehtoisia, vähemmän tarpeellisia lisätoimintoja. Työ oli mielenkiintoinen myös siksi, että ohjelmointityön tilaajana oli museo. Museoiden tulevaisuudessa välttämätöntä digitalisoitumista tekijä ei ollut tullut ajatelleeksikaan, joten hän kiinnostui aiheesta jo senkin vuoksi.

Projekti oli luonteeltaan suunniteltu silmällä pitäen nykyihmisen – erityisesti nuorten ja lasten – maailmankokemuspohjaa. Digitalisoituneessa yhteiskunnassa museoidenkin täytyy ajankohtaisina ja opetukselliselta toiminnaltaan kiinnostavina säilyäkseen ylläpitää ja päivittää elämyksellisyyden tasoansa vastaamaan nykyajan ihmisen vaatimuksia. Mainion keinon ajan hermolla säilymiseen tarjoavat uudet ja erilaiset, jo toiminnan tasoltaan mielenkiintoa herättävät esitysvälineet.

LÄHTEET

Jimm's PC-Store Oy 2014, viitattu 22. syyskuuta 2014.

<http://www.jimms.fi/Product/Show/79446>

Sun, S., 2013, User Interface of Raahe Museum, Oulun seudun ammattikorkeakoulu, tietotekniikan koulutusohjelma, opinnäytetyö.

Wikipedia 2015a, viitattu 3.3.2015.

http://fi.wikipedia.org/wiki/.NET_Framework

Wikipedia 2015b, viitattu 7.5.2015.

http://en.wikipedia.org/wiki/Adobe_Photoshop

Wikipedia 2015c, viitattu 21.4.2015.

http://en.wikipedia.org/wiki/Audacity_%28audio_editor%29

Wikipedia 2015d, viitattu 2.2.2015.

<http://fi.wikipedia.org/wiki/BASIC>

Wikipedia 2015e, viitattu 17.3.2015

<http://en.wikipedia.org/wiki/Chess>

Wikipedia 2015f, viitattu 5.5.2015.

http://en.wikipedia.org/wiki/Comparison_of_C_Sharp_and_Visual_Basic_.NET

Wikipedia 2015g, viitattu 21.4.2015.

http://en.wikipedia.org/wiki/Creative_Commons

Wikipedia 2015h, viitattu 19.4.2015.

http://en.wikipedia.org/wiki/Declarative_programming

Wikipedia 2015i, viitattu 1.5.2015.

<http://en.wikipedia.org/wiki/Direct3D>

Wikipedia 2015j, viitattu 4.5.2015.

<http://en.wikipedia.org/wiki/DirectX>

Wikipedia 2015k, viitattu 4.5.2015.

http://en.wikipedia.org/wiki/Microsoft_Visual_Studio

Wikipedia 2013l, viitattu 20.3.2015.

<http://fi.wikipedia.org/wiki/QuickBASIC>

Wikipedia 2015m, viitattu 4.2.2015.

<http://fi.wikipedia.org/wiki/Vektorigrafiikka>

Wikipedia 2015n, viitattu 9.5.2015.

http://en.wikipedia.org/wiki/Visual_Basic

Wikipedia 2015o, viitattu 1.5.2015.

http://en.wikipedia.org/wiki/Visual_Basic_.NET

Wikipedia 2015p, viitattu 15.2.2015.

http://en.wikipedia.org/wiki/Windows_Presentation_Foundation