

TAMPEREEN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Tietokonetekniikan suuntautumisvaihtoehto

Arto Härkönen

**MOBIILIPELIN SOVITTAMINEN YHDYSVALTALAISILLE
MATKAPUHELINOPERAATTOREILLE**

Työn valvoja: Yliopettaja Kai Poutanen

Työn ohjaaja: DI Atte Kotiranta

Tampere 2007

| | |
|--------------------------------|--|
| Tekijä: | Arto Härkönen |
| Työn nimi: | Mobiilipelin sovittaminen yhdysvaltalaisille matkapuhelinoperaattoreille |
| Työn teettäjä: | Universomo Oy (www.universomo.com) |
| Päivämäärä: | 29.4.2007 |
| Sivumäärä: | 28 sivua ja 4 liitesivua |
| Hakusanat: | J2ME, MIDP, porting, mobiili |
| Koulutusohjelma: | Tietotekniikka |
| Suuntautumisvaihtoehto: | Ohjelmistotekniikka |

| |
|---|
| Työn valvoja: Yliopettaja Kai Poutanen |
| Työn ohjaaja: DI Atte Kotiranta |

Insinööriyö keskittyy projektiin, jonka päämääränä oli sovittaa J2ME-pohjainen LEGO Star Wars II -mobiilipeli eräille asiakkaan määräämille yhdysvaltalaisille matkapuhelinoperaattoreille. Näitä olivat Sprint, Nextel, Boost, T-Mobile, Cingular ja Virgin Mobile. Mobiilipelin on kehittänyt vuonna 2006 Universomo Oy, joka on myös insinööriyön teettäjä. Varsinainen kehitystyö tehtiin Los Angelesissa Kaliforniassa.

| | |
|--|--|
| Author | Arto Härkönen |
| Name of the thesis: | Porting of a mobile game to U.S carriers |
| Comissioning party: | Universomo, Ltd (www.universomo.com) |
| Date: | 29.4.2007 |
| Number of pages: | 28 pages and 4 appendices |
| Keywords: | J2ME, MIDP, porting, mobiili |
| Degree programme: | Computer Systems Engineering |
| Specialization: | Software Engineering |
| Thesis supervisor: | Kai Poutanen |
| Thesis instructor: | Atte Kotiranta |
| <p>This thesis will cover a project of which aim was to port a J2ME -game named LEGO Star Wars II developed by Universomo, Ltd to few customer determined U.S carriers. This includes carriers as Sprint, Nextel, Boost, T-mobile, Cingular and Virgin Mobile. The actual porting phase of this project took place in Los Angeles, California.</p> | |

SISÄLLYSLUETTELO

| | |
|---|----|
| KÄYTETYT TERMIT JA LYHENTEET..... | 5 |
| 1 JOHDANTO..... | 6 |
| 2 PROJEKTIN LÄHTÖKOHDAT..... | 7 |
| 2.1 Sovitettava peli..... | 7 |
| 2.2 Kohdelaitteet..... | 8 |
| 3 PROJEKTIN VALMISTELU..... | 9 |
| 3.1 Game Lobby..... | 9 |
| 3.1.1 Toteutus..... | 9 |
| 3.1.2 Viimeistely..... | 12 |
| 3.2 Lisenssi..... | 12 |
| 3.3 Sovitettavien laitteiden profiilien luonti..... | 13 |
| 3.3.1 Laitteiden jako..... | 14 |
| 3.3.2 Ruudun koko..... | 14 |
| 3.3.3 Käyttömuisti ja suorituskyky..... | 15 |
| 3.3.4 Jar-tiedoston maksimi koko..... | 17 |
| 3.3.5 Muut huomioon otettavat ominaisuudet..... | 18 |
| 3.4 Profiilien viimeistely..... | 19 |
| 4 TESTAUKSESSA LÖYDETYT VIRHEET..... | 21 |
| 4.1 Yleisimmät virheet..... | 21 |
| 4.2 Harvinaiset virheet..... | 23 |
| 5 YHTEENVETO..... | 27 |
| LÄHTEET..... | 28 |
| LIITTEET..... | |

KÄYTETYT TERMIT JA LYHENTEET

| | |
|------------|---|
| Java | Sun Microsystemsin kehittämä oliopohjainen ohjelmistokieli. Yleisesti Java -mobiilipelit tehdään käyttämällä J2ME:tä, Javan mobiililaitteille tarkoitettua kevyttä versiota. |
| J2ME | Java 2 Platform, Micro Edition. |
| API | Application Programming Interface. Ohjelmointirajapinta. |
| CLDC | Connected, Limited Device Configuration. J2ME-konfiguraatio, jota käytetään matkapuhelimissa. |
| MIDP | Mobile Information Device Profile. Mobiililaitteissa käytettävä API-kokoelma. |
| MMAPI | Mobile Media API. J2ME-alustaan sisältyvä multimediarajapintakokoelma, joka lisää tuen äänelle ja videolle. |
| Keskeytyks | Mobiilipelin sovitustyössä tällä viitataan pelin aikana puhelimeen tulleeseen tapahtumaan, jolloin mobiililaitte keskeyttää pelin. Näitä tapahtumia ovat muun muassa puhelinsoitot, teksti- tai multimediamviestit ja Bluetooth-tiedonsiirrot sekä laturin kiinni kytkeminen laitteeseen tai sen irrottaminen laitteesta. |
| Laiteryhmä | Laiteryhmässä voi olla yksi tai useampia mobiililaitteita. Laitteet kootaan näihin ryhmiin ruudun kokojen ja yleisten ominaisuuksiensa perusteella. |

1 JOHDANTO

Mobiilipelin ensimmäinen valmistuote on kehitetty yleensä ainoastaan yhdelle ”masteriksi” kutsutulle päälaitteelle. Universomolla se on usein ollut joko Nokia 6630 tai Nokia N70, joka on hieman Nokia 6630:tä parempi laite. Tämän jälkeen tehdään niin sanotut referenssilaitteet, jotka ovat kaikille lopuille tilatuille sovitettaville laitteille pohjana.

Tämä insinöörityö keskittyy Universomo Oy:n vuonna 2006 kehittämän LEGO Star Wars II -mobiilipelin sovitukseen yhdysvaltalaisille matkapuhelinoperaattoreille. Samalla tutustutaan sovitustyön taustoihin ja alaa koskeviin lainalaisuuksiin kuten, mitä ja minkä perusteella sovitettavan laitteen parametrit haetaan. Tämän lisäksi vertaillaan, miten yhdysvaltalaisille matkapuhelinoperaattoreille tehty sovitustyö eroaa niin sanotusta normaalista, eurooppalaisille matkapuhelinoperaattoreille tehtävästä sovitustyöstä.

Pelkän sovitustyön lisäksi projektiin liittyy tärkeänä osana Game Lobby- ja lisenssipalvelun toteuttaminen. Näitä lukuun ottamatta alkulähtökohtana on, että sovitustyöprojektin oletetaan kulkevan melko lailla samalla tavalla kuin eurooppalaisten sovitustyöprojektien keskimäärin. Yksi projektia helpottava seikka on myös se, että LEGO Star Wars II on valmistunut jo vuonna 2006, ja sitä on jo sovitettu monille eurooppalaisille puhelinlaitteille, joista osa on jopa samoja kuin tämän projektin kohdelaitteet.

2 PROJEKTIN LÄHTÖKOHDAT

2.1 Sovitettava peli

Sovitustyön kohteena oleva mobiilipeli LEGO Star Wars II sijoittuu vuonna 1977 George Lucasin ohjaamaan ja käsikirjoittamaan Tähtien sota: Episodi IV - Uusi Toivo -nimiseen tieteiselokuvaan, joka on Tähtien sota -elokuvista ensimmäinen, mutta episodijärjestysnumeroltaan neljäs, vaikka kolme ensimmäistä episodista tehtiin itse asiassa episodien IV-VI jälkeen. Mobiilipeliversiossa elokuvan pääsankari Luke Skywalker on joutunut ystävineen pahan Imperiumin Kuolemantähti-nimiselle suurelle avaruusalukselle. Pelin alussa pelaajalla on käytössään vain Luke, mutta pelin edetessä mukaan liittyvät niin R2-D2-droidi kuin prinsessa Leiaakin. Hahmojen mukaantulon jälkeen pelaaja voi vapaasti valita, ketä hahmoista pelaaja kulloinkin ohjaa. Pelialueet on suunniteltu siten, että kaikkia käytössä olevia hahmoja tarvitaan kentän läpäisemiseksi. Lopulta kahdenkymmenen alueen jälkeen pelaajan ryhmä pakenee Kuolemantähdeltä elokuvasta tutulla Millennium Falcon -aluksella. LEGO Star Wars II on Universomo Oy:n kehittämä ja TT Gamesin / THQ Wirelesin julkaisema. Kuvassa 1 esitellään pelin päävalikko ja kuvissa 2 - 4 on ruutukaappauksia itse pelitilasta. Kuvat on otettu Sanyo 5600 -laitteelle tarkoitettusta versiosta, eli käytetty ruudun koko kyseisessä laitteessa on 240x320. /1/ /2/



Kuva 1



Kuva 2



Kuva 3



Kuva 4

2.2 Kohdelaitteet

Kohdelaitelista oli asiakkaan toimittama raakalista, jossa olivat kaikki sovitustyön kohteena olevat laitteet jaoteltuina matkapuhelinoperaattorien mukaan. Tällainen lista ei ole sinänsä käyttökelpoinen, vaan tämän perusteella tehtiin vielä käytännöllisempi sovituslista, johon laitteet oli lajiteltu laitoryhmiin aikaisempien kokemusten perusteella. Esimerkiksi Sprintin laitoryhmät voitiin ottaa suoraan aikaisemmin Sprintille tehdystä LEGO Star Wars I -mobiilipelistä. Asiakkaan toimittama laitelista on liitteenä.

3 PROJEKTIN VALMISTELU

Projektin ensimmäisenä vaiheena oli toteuttaa LEGO Star Wars II:een tuki Game Lobby -palvelulle. Game Lobby on mobiilipeliin lisättävä lisäpalvelu, jonka avulla pelaajat voivat vertailla pelissä saamiaan pisteitä toisten pelaajien kanssa. LEGO Star Wars II:ssa pisteinä ovat kultaiset LEGO-palikat, joiden kerääminen avaa myös pelin sisäisiä lisäominaisuuksia.

3.1 Game Lobby

Käytännössä Game Lobby toimii niin, että pelin käynnistyksen yhteydessä Game Lobby pyytää pelaajaa rekisteröitymään palveluun pyytämällä pelaajan nimimerkin ja puhelinnumeron, jota ei kuitenkaan tarvita kaikkien operaattorien, kuten Sprintin kohdalla, vaan liittymän tunnistus käy muilla tavoin. Tässä vaiheessa pelaajalle annetaan mahdollisuus jatkaa myös ”vierailijana”, jolloin pelin aikana kerättyjä pisteitä ei talleteta eikä Game Lobby -valikko avaudu. Rekisteröitymisen jälkeen seuraavilla käynnistyskerroilla Game Lobby tunnistaa pelaajan automaattisesti ja pelaaja voi suoraan jatkaa itse peliin. Pelin alussa tapahtuvan tunnistuksen lisäksi pelin päävalikkoon täytyi lisätä valinta itse Game Lobby -valikkoon, jossa olivat Game Lobby -määrittelyn mukaiset toiminnot, jotka käyvät ilmi taulukosta 1.

3.1.1 Toteutus

Käytännössä Game Lobbyn pelaajan tunnistus ja rekisteröinti integroitiin peliin lisäämällä uusia tiloja pelin alkulatausten ja päävalikon väliin. Toinen vaihtoehto olisi ollut lisätä uudet tilat ennen varsinaisten peliresurssien latausta, mutta tällöin Game Lobby -tiloissa ei olisi ollut mahdollista käyttää pelin omia resursseja, kuten bitmap-fontteja, ja näin ollen myös pelin valmiiden käyttöliittymäkomponenttien käyttäminen olisi ollut kyseenalaista. Tämä vaihtoehto olisi ollut äärimmäisen karu sekä visuaalisessa että teknisessä mielessä, joten se päätettiin unohtaa. Koska lisätyt tilat sijaitsivat tilakoneessa peliresurssien latausten jälkeen, pystyttiin käyttämään pelissä

ennestään olevia käyttöliittymänkomponentteja luotettavasti. Nämä komponentit käsittävät tekstikenttä- ja valikkoruutujen piirrot, joita tarvittiin erilaisten ilmoitusruutujen ja Game Lobby -valikon esitykseen. Näin vältyttiin myös uuden käyttöliittymäkoodin lisäämiseltä, joka on aina toivottava asia, kun lopullinen Jar-tiedosto yritetään pitää mahdollisimman pienenä. Kuvissa 5 ja 6 esitellään Game Lobby -ruutujen lopullinen ulkonäkö. Näistä myös voidaan hyvin huomata käyttöliittymien yhdenmukaisuus kuvan 1 kanssa. Kuvassa 5 on Game Lobby -tervehdysruutu, joka siis tulee pelin käynnistyksessä asiakas- ja developer-logojen jälkeen silloin, kun käyttäjä ei ole ennestään järjestelmässä, eli kun käyttäjä käynnistää pelin ensimmäistä kertaa. Päävalikon alle toteutettu varsinainen Game Lobby -päävalikko on esitetty kuvassa 6. Valikkotilaan mahtuu vain neljä valintaa, joten alareunassa oleva nuoli ilmoittaa, että valintoja voi selata ruudun reunan yli, minne ”Visit Game Lobby” -valinta on jäänyt. Kuvat on otettu Sprintin Sanyo 5600 -laitteelle tarkoitettusta versiosta.

Taulukko 1 Game Lobby -päävalikon toteutetut toiminnot

| Game Lobby -valinta | Toiminto |
|----------------------------|---|
| Leaderboard | Hakee pistelistan keskuspalvelimelta ja esittää sen pelaajalle. |
| Stats | Hakee pelaajan viimeksi lähettämät pisteet keskuspalvelimelta ja esittää ne pelaajalle. |
| Rate Game | Lähettaa keskuspalvelimelle pelaajan pelille antaman arvosanan asteikolla 1-10. |
| Recommend | Lähettaa pelaajan ystävälistalle suosituksen kyseessä olevasta pelistä. |
| Visit Game Lobby | Sulkee pelin ja avaa laitteen wap-selaimen Game Lobby -kotisivuille. |

Game Lobbyn integraatiossa yksi ongelmallisimmista tavoitteista oli säilyttää alkuperäinen projekti sellaisessa kunnossa, että Game Lobbyä varten lisätyt koodit ja resurssit eivät vaikuttaisi alkuperäiseen peliin millään tavalla. Tämä on tärkeää siksi, että alkuperäistä peliä koostettaessa Jar-tiedosto pysyisi mahdollisimman lähellä alkuperäistä kokoa eikä se Game Lobby integroinnin jälkeenkään menisi esimerkiksi yli 64 kilotavun rajan. Tähän tosin jouduttiin tekemään korjauksia projektin myöhemmässä vaiheessa. Lisäksi huomioon ottamisen arvoista tässä oli se, että

operaattoreista vain Sprintille ja Cingularille lopulta tarvittiin oma Game Lobby ja näistäkin vain Sprintille se pantiin jokaiseen laiteryhmään ja Cingularilla vain muutamaan. Tarkennettuna Cingularin versio Game Lobbyä oli nimeltään ”gameroom” kaikki pienellä kirjoitettuna. Tämä aiheutti jonkin verran hankaluuksia toteutuksen osalta, sillä oletuksena pelissä käytettiin aina fonttia, jossa oli pelkästään isot kirjaimet. Koska ”gameroom” oli tuotemerkki, sitä ei voitu kirjoittaa samaan tapaan isoilla kirjaimilla, vaan fontteihin jouduttiin lisäämään pienet kirjaimet g:lle, a:lle, m:lle, r:lle, ja o:lle. Vain ”gameroom” -tekstin vaatimat kirjaimet lisättiin, koska loppuja pieniä kirjaimia ei tarvittaisi mihinkään ja näin ollen ne veisivät vain turhaa tilaa lopullisessa Jar-tiedostossa sekä ajonaikaisesti kohdelaitteen käyttömuistissa.



Kuva 5



Kuva 6

Toinen huomattavasti ongelmallisempi kohta toteuttamisessa oli se, että aikaisemmat toteutukset toisille Universomon peleille oli tehty vanhalle Universomon alustalle. Tästä seurasi se, että suurin osa vanhasta toteutuksesta oli sellaisenaan käyttökeltotonta ja täytyi kirjoittaa uudestaan. Erityisesti tilakonejärjestelmään oli tullut ajan mittaan paljon muutoksia.

3.1.2 Viimeistely

Game Lobby -toteutuksen viimeistelyyn kuului omien emulaattoritestien ajamisen lisäksi kokeiltavan version lähettäminen asiakkaalle. Testauslaitteeksi valittiin Sanyon 5600, joka oli aiemmista projekteista tiedetty vakaaksi, hyvin toimivaksi ja nopeaksi laitteeksi, joten pääpaino testauksessa voitiin pitää Game Lobbyn puolella. Sanyo 5600 oli siis Sprintin laite ja Game Lobbyä käytettiin näin ollen Sprintin versiota. Lisäksi viimeistelyyn tietysti kuuluivat testauksessa löydettyjen virheiden korjaus ja asiakkaan ehdotusten toteutus.

Teknisesti testauksessa ollut versio olikin liki täysin virheetön, mutta koska peliä tehtiin Yhdysvaltojen markkinoille ja pelin alkuperäinen kohdealue oli ollut Eurooppa ja pelissä oli siksi käytetty tietynlaista englantia, piti tietyt termit muuttaa amerikanenglantia vastaaviksi. Esimerkiksi ”armoured” täytyi muuttaa ”armored”:ksi. Muihinkin samankaltaisiin, mutta lähinnä kieliopillisiin eroihin törmättiin myöhemmin, mutta niitä ei enää katsottu aiheelliseksi muuttaa. Lisäksi THQ Wirelesin lakitekstit täytyi lisätä pelin ”About” ja ”Credits” -teksteihin.

3.2 Lisenssi

Game Lobby -palvelun lisäksi mukaan vaadittiin operaattorikohtainen MRC-lisenssipalvelu (Media Resource Control), jota käytetään pitämään kirjaa ostetusta peliajasta. Tämä kuului MRC:n määrittelyn mukaan toteuttaa niin, että pelin startApp() -metodissa kutsutaan MRC:n validateLicense() -metodia, jonka paluuarvona saadaan lisenssin tila, ja sen perusteella suoritusta joko jatketaan, jos lisenssi on kunnossa tai lopetetaan, jos se ei ole kunnossa. Aivan pelin alussa kutsutun startApp()-metodin muutoksen lisäksi pelin päävalikkoon kuului määrittelyn mukaan myös lisätä ”Cancel Substriction” -valinta, joka antaa käyttäjälle mahdollisuuden peruuttaa lisenssinsä tilaamaansa peliin.

Kuten Game Lobbyä, myöskään MRC-lisenssiä ei tarvinnut panna joka operaattorikohtaiseen laitoryhmään, vaan tässäkin oli operaattorikohtaisia eroja.

Cingularilla lisenssi pantiin kaikkiin laiteryhmiin ja Sprintillä vain MIDP 1.0 laitteisiin. T-Mobile ja Virgin Mobile eivät vaatineet erillistä MRC-lisenssiä lopulta ollenkaan.

Lisenssituen toteutus oli hyvinkin suoraviivaista Game Lobby -toteutukseen verrattuna, koska lisätiloja ei tarvittu, eikä myöskään uusia tekstijonoja ”Cancel Subscription”:ia lukuun ottamatta. Näiden lisäksi joitain operaattorikohtaisia määritelmän mukaisia parametreja täytyi lisätä Jad-tiedostoon. Nämä parametrit antoivat muun muassa tietoja siitä, onko lisenssijärjestelmä ylipäätään käytössä, ja koska nykyinen lisenssi umpeutuu.

3.3 Sovitettavien laitteiden profiilien luonti

Kun Game Lobby ja lisenssijärjestelmä oli toteutettu ja testattu asiakkaalla toimivaksi sekä tarvittavat muutokset tehty, oli aika siirtyä projektissa seuraavaan vaiheeseen eli laiteprofiilien luontiin. Kuten aiemmin oli jo esillä, käytännössä kaikki laitteet ovat jollain tapaa erilaisia, joskin joitain laitteita voidaan ryhmittää isommiksi kuin yhden laitteen sisältäviksi laiteryhmiä. Tämä on yleistä etenkin Samsungin laitteissa, jotka saattavat päätyä hyvin pieneen määrään ryhmiä, joissa kussakin on paljon laitteita. Liiallinen ryhmittäminen on tosin potentiaalinen virhevuon aiheuttaja, sillä vaikka päällisin puolin laitteet näyttävätkin samalta eli niissä on sama ruudun koko, saman verran käyttömuistia ja ulkonäöllisestikin muistuttavat toisiaan, niissä saattaa olla Java-virtuaalikoneen toteutuksessa eroja. Nämä erot voivat tulla esiin vasta tarkemmassa testauksessa. Tämän jälkeen ryhmän jakaminen omiksi laitteikseen voi olla hyvinkin työlästä, sillä muista eroavien laitteiden etsintä lopusta ryhmästä on hyvin aikaa vievää, varsinkin jos kyseessä oleva virhe on hyvin vaikeasti toistettavissa. Esimerkkinä mainittakoon tilanteita, että peli kaatuu puhelimen valikkoihin kahden tunnin pelaamisen jälkeen.

3.3.1 Laitteiden jako

Tavallisesti laitteet lajitellaan muutamalla eri tavalla. Tärkeimpänä on laitteen tukema MIDP-kokoelma, joka on joko MIDP 1.0 tai kehittyneempi MIDP 2.0.

Ilahduttavasti lähes poikkeuksetta kaikki uudet laitteet tukevat MIDP 2.0:aa, mutta vanhemmat halvemmat laitteet tukevat yleisesti ottaen vain MIDP 1.0:aa.

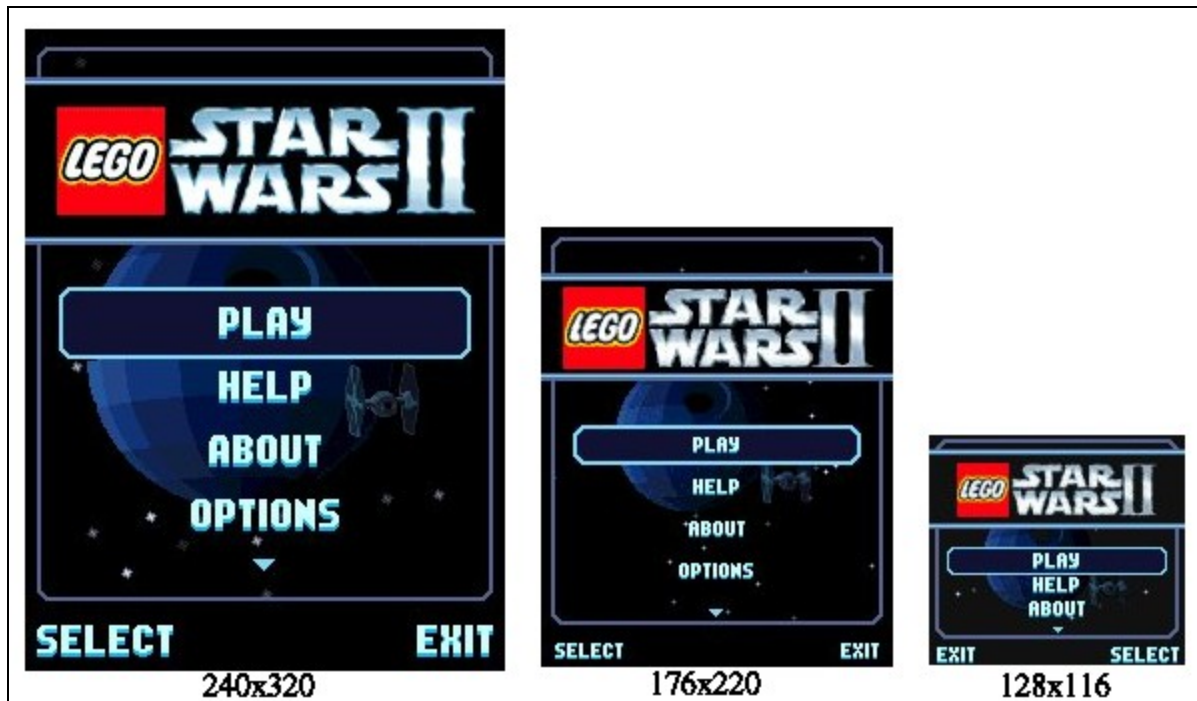
3.3.2 Ruudun koko

Seuraavaksi ilmeisin jakoperuste ovat laitteen päänäytön dimensiot. Näyttökoot voidaan jakaa pääryhmiin, jotka on hahmotettu taulukkoon 2.

Taulukko 2 /3/

| Kategoria | Ruudun koko (leveys x korkeus) | Esimerkkilaite |
|------------------|---------------------------------------|-----------------------|
| Small | alle 128 x 128 (96 x 65) | Nokia 3150i |
| Medium | 128 x 116 - 128 x 160 | Nokia 6101 |
| Large | 176 x 160 - 176 x 220 | Nokia 6630 |
| QVGA | 240x320, 320x240 | Nokia 6270, Nokia E61 |
| Super Large | 352 x 416 | Nokia N90 |

Kaikki LEGO Star Wars II -sovitusprojektiin kuuluvat laitteet olivat ruudun kooltaan välillä 128 x 128 - 320 x 240, joten Small- ja Super Large -kategoriat voitiin tämän projektin osalta unohtaa. Varsinkin Small-kategoria on usein hyvin ongelmallinen, koska niin pieneen ruutuun ei vain yksinkertaisesti mahdu juuri mitään. Super Large -kategoriakaan ei ole aivan niin ongelmaton kuin voisi luulla. Tämä johtuu siitä, että pelit on nykyisin suunniteltu alun perin Large-kategorialle, joka on siis puolet Super Largea -kategoriaa pienempi. Näin ollen kaksinkertaiseen kokoon pelitila näyttää usein äärimmäisen karulta, joten uuden grafiikan teolta ei yleensä vältytä. Lisäksi yleisesti voidaan sanoa, että kaikki MIDP 1.0 -laitteet kuuluvat myös Medium-näyttöluokkakategoriaan ja MIDP 2.0 -laitteet taas Large-kategoriaan tai suurempiin. Tästäkin on tietysti poikkeuksia, sillä Medium-kategoriaan kuuluu myös joitakin MIDP 2.0:aa tukevia laitteita, esimerkiksi juuri Nokia 6101. Näyttöluokkakategoriat havainnollistetaan kuvassa 7. /3/



Kuva 7

3.3.3 Käyttömuisti ja suorituskyky

Laitteet voitaisiin jakaa myös erillisesti käyttömuistin suuruuden ja suorituskyvyn mukaan, mutta nämä kulkevat lähes kaikissa laitteissa käsi kädessä, joten erillinen jako olisi melko hyödytöntä. On olemassa kuitenkin laitteita, joissa on suuret määrät käyttömuistia, mutta suorituskyky jättää toivomisen varaa. Käyttömuistilla tarkoitetaan puhelinlaitteen ”heap”-muistia, johon puhelin siis käyttää ajonaikaisen datansa tallentamiseen ja manipuloimiseen. Suorituskyvyllä viitataan puhelinlaitteen keskussuorittimen nopeuteen. Tälle ei voida antaa mitään suoria arvoja, koska tämä vaatisi ylimääräistä testaamista ja sovitustyössä käytännössä suorituskyky jaetaan suoraan ruudun päivitysnopeuden mukaan karkeasti kolmeen eri luokkaan, jotka ovat huono, tyydyttävä ja hyvä. Näistä luokista huono ja hyvä on selvästi suurimmat, koska tyydyttävän rajat ovat hyvin epäselvät ja yleensä projektissa päädytään näille käyttämään huonolle suorituskyvylle suunniteltuja asetuksia. Samoin kuin MIPD-jaossa, myös käyttömuistin suuruus ja suorituskyky korreloittuu lähes aina suoraan ruudun kokoon, mutta koska laitteita on hyvin suuri määrä ja paljon eri

valmistajia mukaan mahtuu tietysti myös poikkeuksia. Esimerkiksi Nokian valmistamat Medium-näyttökategoriaa edustavat laitteet ovat usein myös tehokkaita suorituskyvyltään.

Yleisesti voidaan siis sanoa, että huonon suorituskyvyn takia niin kutsutusta 'high-end' -versiosta täytyy poistaa muun muassa visuaalisia efektejä, jotta ruudun päivitysnopeus pysyisi tyydyttävänä. Tyydyttävä on tosin tässä kohtaa hyvin liukuva käsite, sillä esimerkiksi monissa halvoissa Samsungin laitteissa suorituskyky on niin heikko, että minimiasetuksillakin pelin ruudun päivitysnopeus jää välillä vain muutamaan kertaan sekunnissa pelin intensiivisimmissä kohdissa. Maksimi ruudun päivitysnopeus on Universomon peleissä, siis myös LEGO Star Wars II:ssä, rajoitettu 18 päivitykseen sekunnissa. Vaikka 18 kuvaa tuntuu melko vähäiseltä, tämän on katsottu olevan riittävä puhelinlaitteiden pienille näytöille, jotka eivät paljon nopeampaan päivitysnopeuteen usein pystyisikään.

Käyttömuistin suuruus tai oikeammin sen puute sen sijaan määrää sen, kuinka paljon täyden pelin ominaisuuksista täytyy karsia ennen kuin peli toimii luotettavasti pitkiäkin aikoja. Yleensä käyttömuistin loppuminen näkyy vasta pitkien pelisessioiden jälkeen ja vain hyvin harvoin suoraan pelin käynnistyksessä tai juuri sen käynnistyttyä. Kaatuminen tällaisissa tapauksissa johtuu yleensä siitä, että kun muistiin ladataan resursseja ja käytön jälkeen niitä vapautetaan sieltä, muisti pirstaloituu pitkän ajan kuluessa. Sitten kun pirstaloitumisen jälkeen täytyykin muistiin ladata esimerkiksi jokin suuri, paljon muistia vievä kuva, se ei enää mahdukaan, koska muistista ei enää löydy tarpeeksi suurta yhtämittaista muistialuetta vaikka yhteenlaskettuna muistia olisikin riittävästi vapaana. Toinen syy tällaiseen kaatuiluun voi olla myös ohjelman muistinvuotaminen, mutta tällaiset ongelmat on aina korjattu jo ennen pelin sovituvaihetta varsinaisen pelinkehitystyön puolella. Kuitenkin huomioon ottamisen arvoista on myös se, että sovitettavana oleva kohdelaite voi myös itse vuotaa muistia, kun pelissä käytetään heikosti suunniteltua tai virheellisesti toteutettua laitteen rajapintaa.

Käyttömuistia laitteissa on yleensä noin 300 kilotavusta noin kahteen megatavuun, mutta suurempiakin määriä voi joissain laitteissa olla. Sovitustyössä riittävän

käyttömuistin määrä menee projektikohtaisesti yleensä noin 600 kilotavussa. Jos laitteessa on enemmän käyttömuistia kuin 600 kilotavua, ei sen käytöstä yleensä tarvitse huolehtia ja kaikki muistiriippuvaiset ominaisuudet voidaan kytkeä käyttöön. Näistä hyvänä esimerkkinä on kaikkien pelin tarvitsemien resurssien pitäminen muistissa aina pelin alkulatauksesta lähtien. Tämä ominaisuus lyhentää latausaikoja huomattavasti, mutta ei tietenkään voida käyttää laitteissa, joissa on reilusti alle 600 kilotavua käyttömuistia.

3.3.4 Jar-tiedoston maksimi koko

Kuten aikaisemmin tuli esille, koko suoritettava ohjelma pakataan Jar-tiedostoon, joka siirretään puhelimen käyttämää verkkoa pitkin kohdelaitteeseen. Tälle tiedostolle on joissakin laitteissa asetettu maksimi koko, joka johtuu puhelimen rajoituksesta siirtää yksittäistä tiedostoa. Useimmiten tämä raja ei ole ongelma, koska raja voi olla hyvinkin suuri, eikä näin ollen ole olennainen korkeintaan muutaman sadan kilotavun pelille. Kuitenkin joissakin laitteissa raja voi olla niinkin pieni kuin 64 kilotavua, joka asettaa suuren haasteen pelin pakkaamiselle ja ominaisuuksien karsimiselle. Pelin kehitystyöhön kuuluu myös tuon alle 64 kilotavun version toteuttaminen, mutta LEGO Star Wars II:stä Game Lobby ja lisenssipalvelun lisäämisen jälkeen ei enää ollut mahdollista koostaa tarpeeksi pientä Jar-tiedosta ilman toteutukseen tehtyjä korjauksia. Muita vastaan tulleita rajoja ovat 80 kilotavua ja 100 kilotavua. Nämä eivät yleensä ole ongelma, mutta esimerkiksi äänien mukaan mahduttaminen voi joissakin tapauksissa olla ongelmallista. LEGO Star Wars II:n sovitushankkeessa ei ollut muita laitteita kuin LG:n C1300, jossa oli kireä 64 kilotavun rajoitus.

Jar-tiedostorajoitteiden puuttuminen toikin suuren helpotuksen projektiin kokonaisuudessaan, koska Game Lobby ja lisenssi-palvelut toivat noin 15 kilotavua ylimääräistä kokoa Jar-tiedostoon. Tämä olisi tarkoittanut hyvin suuria ongelmia, koska 15 ylimääräisen kilotavun kuristaminen vaatii hyvin paljon raskaita toimenpiteitä.

Lisäksi T-Mobilen versioita tehdessä täytyi muistaa jättää T-Mobilen vaatimat ylimääräiset 12 kilotavua ”tyhjää” tilaa Jar-tiedostoon. Siis siten, että peli ja tuo T-Mobilen vaatimat 12 kilotavua eivät yhteensä ylitä kohdelaitteen maksimi Jar-tiedoston rajoitusta.

3.3.5 Muut huomioon otettavat ominaisuudet

Yllä tarkasteltujen seikkojen lisäksi profiilia luodessa täytyy ottaa huomioon muutamia muita laiteominaisuuksia. Näistä yksi tärkeimmistä on laitteessa käytettävä äänituki, joita on muutamia mahdollisia. Yleisin on MMAPI-standardi, joka kuuluu optionaalisena osana J2ME-alustaan. Tätä ei kuitenkaan ole toteutettu aivan kaikkiin laitteisiin, joten myös muita rajapintoja täytyy tukea, jotta äänet saataisiin toimimaan mahdollisimman laajasti. Muun muassa Samsungilla on oma äänirajapintansa (Samsung API), joka voi toistaa midi-äänitiedostojen lisäksi myös mmf-tiedostoja. Mmf-formaatti on Yamahan kehittämä SMAF (Synthetic music Mobile Application Format) musiikkidataformaatti. Jotkut Samsungin laitteet tukevat kuitenkin molempia rajapintoja, mikä antaakin mukavasti liikkumavaraa sovitustyössä. Jos toinen rajapinta ei toimi halutulla tavalla, voidaan yrittää käyttää toista rajapintaa. Tämäkin tilanne oli yllättävän yleinen myös LEGO Star Wars II:n sovitusprojektissa. /4/

Äänirajapinnan lisäksi toinen huomioon otettava asia, on laitteen ohjausmenetelmä. Suurimmassa osassa laitteista on normaali totuttu ohjaustikku tai suuntanuolet, joiden lisäksi on tietysti numerot nollasta yhdeksään ja tähti- ja risuaitamerkit. Etenkin niin kutsutuissa design-laitteissa on yritetty käyttää jotain muuta eksoottista vaihtoehtoa, joita tässä projektissa tuli onneksi vastaan vain yksi kappale. Virgin Mobilen Switch-Back -mallinimellä kulkevassa laitteessa oli normaalien numeronappien sijaan QWERTY-näppäimistö. Tämä ei kuitenkaan ollut projektin kannalta vaikeakaan ongelma, sillä sovitettavassa pelissä ei välttämättä tarvinnut oletuksena käyttää kuin suuntanuolia, jotka Virgin Mobilen laitteesta löytyi. Koska LEGO Star Wars II:ssä keskinapin lisäksi oletusohjauksiin kuului myös tulitus viitos- ja tähtinäppäimellä, täytyi nämä poistaa käytöstä tässä laitteessa sillä ne eivät olleet millään tavalla

loogisilla paikoilla pelaamisen kannalta. Tämän takia täytyi tietysti muuttaa myös pelin sisältämiä avustustekstejä ohjauksen osalta.

QWERTY-näppäimistön lisäksi paljon suurempi ongelma tuli vastaan eräästä Samsungin laiteryhmästä, jossa ei kokoruututilaa käytettäessä saanut kummastakaan softkey-napista mitään näppäinkoodia, joten näiden painallusta ei saanut mitenkään kiinni Java-ohjelmassa. Normaalissa MIDP 1.0:n käyttämässä tilassa, joka siis ei käytä koko ruutua, näppäimet sen sijaan toimivat hyvin, mutta tässä tilassa ilmeni hyvin arvoituksellista kaatuilua, jolle ei keksitty kunnollista syytä. Ongelma kierrettiin siten, että softkey-napit korvattiin laitteen tähti- ja risuaitamerkeillä, joiden näppäinkoodit toimivat oikein. Softkey-rivi kuitenkin piirrettiin normaalisti ruudun alareunaan, mutta näihin lisättiin tähti- ja risuaitamerkit vastaamaan muuttuneita softkey-nappeja. Idea otettiin laitteeseen tehtaalla asennetuista alkuperäispeleistä, joissa tätä samaa ratkaisua oli käytetty.

3.4 Profiilien viimeistely

Yllä olevien seikkojen kohdilleen asettamisen ja profiilin luomisen jälkeen kunkin laitteen kohdalla tehtiin ohjelmoijan suorittama pikainen tarkistus, johon kuului muun muassa pelin pelaaminen muutaman ensimmäisen pelialueen läpi. Tässä tarkoituksena oli nopeasti tarkistaa, että äänet toimivat oikein eikä niissä esiintynyt mitään kappaleeseen kuulumattomia räähdyksiä tai muita ongelmia esimerkiksi saman kappaleen uudelleen aloittamisen kanssa. Äänissä olikin yksi toistuvimmista ongelmista tämän projektin alkuvaiheessa, sillä LEGO Star Wars II:ssä oli ensimmäistä kertaa käytetty kahta eri kappaletta saman pelialueen sisällä. Tätä oli aikaisemmin vältetty, sillä useille laitteelle kappaleen nopea vaihtaminen on edelleen ongelmallista. Joko se on äärimmäisen hidasta eli kestää monta sekuntia, joka aiheuttaa epämukavuutta pelaamiseen, tai se ei vaihdu ollenkaan seuraavaan kappaleeseen ja puhelimen äänijärjestelmä menee osittain tai kokonaan jumiin. Tällöin äänien palauttamiseksi täytyy usein uudelleenkäynnistää koko peli. Näiden ongelmien kiertämiseksi täytyi tämä ominaisuus yleensä poistaa profiileista. Toinen yleinen ongelma äänien kanssa oli projektissa käytössä oleva äänentason säätäminen

sovellustasolla. Tämäkin ominaisuus täytyi tietysti kytkeä pois puhelimista, jotka eivät tukeneet tätä toimintoa.

Äänien lisäksi nopeassa tarkastuksessa tarkistettiin nopeasti, että peli reagoi oikein erilaisiin keskeytyksiin, kuten puhelinsoittoon, tekstiviestiin ja laturin kiinni kytkemiseen. Koska keskeytyksiä ei useinkaan ehditty tekemään kovin montaa kertaa, tässä kohdassa ei tyypillisesti löytynyt ongelmia. Keskeytyksiin liittyvät ongelmat esiintyvätkin usein, kun peliä on pelattu pitemmän aikaa, tai kun keskeytyksiä tehdään useita yrityksiä peräkkäin.

Normaalissa sovitushankkeissa tässä vaiheessa koostettu peliversio annettaisiin Universomon sisäiseen laadunvarmistustestaukseen, mutta tässä projektissa se toimitettiin suoraan asiakkaan testausryhmälle.

4 TESTAUKSESSA LÖYDETYT VIRHEET

Tyypillisessä sovituspöytätyössä noin puolet koostetuista testiversioista menee laadunvarmistustestauksesta ensimmäisellä testauskerralla läpi eli niistä ei löydetä mitään virheitä, joita olisi pakko korjata. Näiden jälkeen puolista eli 25 % kaikista koostetuista versioista löytyy keskimääräisesti sellaisia virheitä, jotka ovat nopeita ja helppoja korjata ja nämä versiot menevät toisella yrityksellä läpi. Loput 25 % ovat niin sanottuja ongelmalaitteita, joiden virheiden korjaaminen on hankalaa sekä aikaavievää ja lopulliseen läpikäymiseen voidaan tarvita useitakin kertoja. Tällaisia laitteita tai laiteryhmiä ovat useimmiten laitteet, joiden versio sisältää sellaisia virheitä, mitä ei voida luotettavasti toistaa. Esimerkiksi yksi tällainen virhe oli muun muassa myöhemmin paremmin käsiteltävä virhe Sprintin LG pm-325:ssa, jossa taustalle piirrettävä taustakuva katosi, kun sama pelialue käynnistettiin uudelleen riittävän monta kertaa.

4.1 Yleisimmät virheet

Koska suurin osa kohdelaitteesta oli laitteita, joille ei ollut aikaisemmin sovitettu mitään, suurin osa testauksessa löydettyistä virheistä liittyi profiilien puutteellisuuksiin. Osa näistä virheistä olikin sellaisia, jotka olisi pitänyt löytää ohjelmoijan suorittamassa pikaisessa tarkistuksessa, kuten pelin ikonin puuttuminen. Yleisesti ikonien puuttuminen selittyi väärällä ikonin koolla, jonka oikeaksi vaihtamisen jälkeen ikoni alkoi toimia ongelmitta. Oikean ikonikoon määrittäminen tosin on välillä hankalaa, mutta hyviä johtolankoja tähän saa laitteeseen tehtaalla asennetuista peleistä.

4.1.1 Laiteryhmittelyä koskevat virheet

Kuitenkin määrällisesti eniten virheitä löytyi niistä laitteista, joita ei ollut mahdollista profiilien tekohetkellä tarkistaa ja joiden tarkkoja ominaisuuksia ei tiedetty, vaan laitteet jouduttiin sokeasti ryhmittelemään muiden tunnettujen laitteiden joukkoon. Haastavuudeltaan tällaiset virheet eivät onneksi olleet kovinkaan hankalia, vaan esimerkiksi niissä saattoi vain olla käytössä väärä äänirajapinta. Tällöin laite täytyi erottaa alkuperäisestä laitoryhmästä, koska suurella todennäköisyydellä laitoryhmän kaikki muut laitteet eivät tuekaan tätä toista äänirajapintaa.

4.2.1 Keskeytysvirheet

Toinen hyvin yleinen virheluokka on niin sanotut keskeytysvirheet, joita on hyvin monia erilaisia. Pahin näistä on tietysti pelin totaalinen kaatuminen esimerkiksi puhelinoiton tullessa. Tätä esiintyi yleensä vain kaikista huonoimmissa laitteissa, kuten Samsungin A620:ssä, jolle ei pitkänkään tutkiskelun jälkeen voitu tehdä mitään. Tämä sama ongelma esiintyi myös laitteen esiasennetuissa peleissä. Hypoteesi oli, että soiton tullessa laite antaa Java-ohjelmalle tietyn määrään aikaa suorittaa `hideNotify()`-metodi, ja jos suoritus jää kesken, laite pakkosulkee ohjelman. Tähän näytti vaikuttavan myös se, että missä kohdassa pelin pääkiertoa suoritus keskeytyksen tullessa on, mutta LEGO Star Wars II:n kohdalla taustapelialueen piirtäminen yksinkertaisesti kestää liian kauan.

Kaatumisongelmien lisäksi lievempiä ongelmia ilmeni laitteissa, joissa äänet katosivat keskeytyksen jälkeen. Puhelin siis muuten osasi hoitaa keskeytyksen oikein, mutta keskeytyksestä palatessa äänirajapinta oli sotkeutunut. Tällaiset tilanteet voidaan hoitaa normaalisti helposti, jos puhelin vain kutsuu `hideNotify()`- ja `showNotify()`-metodeja oikein keskeytyksen tullessa ja sen loppuessa. Ilman näitä kutsuja Java-ohjelmalla on hyvin vähän tapoja tietää, että keskeytys on tullut. Kaikkiin näihin virheisiin ei näin ollen löydettykään ratkaisua, mutta kyseiset virheet päätettiin sivuuttaa sillä perusteella, että samat ongelmat ilmenivät poikkeuksetta myös laitteiden esiasennetuissa peleissä.

4.1.2 Pelin suunniteltuja ominaisuuksia koskevat virheet

Edellä mainittujen virheiden lisäksi testausryhmä raportoi myös virheitä, jotka jouduttiin ohittamaan suoraan. Tällaiset virheet koskivat muun muassa pelin alkuperäistä tasapainotusta tai muuten vaan pelin suunniteltua toiminnallisuutta. Näiden raporttien vaatimia muutoksia ei enää sovitussvaiheessa tehdä, koska tämä aiheuttaisi liikaa epäyhtenäisyyttä jo maailmalle toimitettujen versioiden kanssa, elleivät ne siis ole täysin fataaleja.

4.1.3 ”Boarc”-virhe

Yksi projektin alkuvaiheessa ilmenneistä virheistä, oli niin kutsuttu ”boarc”-virhe, joka esiintyi kaikissa laitteissa, joiden ruudun leveys oli 128 pikseliä. Tarkemmin virhe ilmeni Tähtien sodasta tutussa alkuintrotekstin, lauseessa ”Little does she know that destiny took a farm boy named Luke onboard the same armored space station, along with the very droid that can restore freedom to the galaxy...” sanasta ”onboard” leikkaantui oikean puoleisin pikselirivi pois tekstikentän reunan tullessa vastaan, siten että jäljelle jäänyt sana näytti ruudulla ”onboarc”:lta. Vaikealta tuntunut virhe korjaantui kuitenkin helposti päivittämällä pelin käyttämä Universomon alusta uuteen versioon, jossa tämä ongelma oli jo korjattu.

4.2 Harvinaiset virheet

Harvinaisiksi voidaan katsoa virheitä, jotka esiintyivät joko yhdessä laitteessa tai laiteryhmistä. Koko projektin kannalta nämä veivät varovaisesti arvioiden jopa puolet projektiin käytetystä ajasta ja voimavaroista, joskaan tästä ei ole kerätty tarkkaa tietoa. LEGO Star Wars II -projektissa nämä virheet johtuivat poikkeuksesta itse kohdelaitteiden sisältämisestä virheistä tai puutteista.

4.2.1 Sprint Samsung A707 tekstikentän sekoaminen keskeytyksessä

Tämä virhe ilmeni tietävästi vain Sprintin Samsung A707:ssä, joka on kohtuullisen uusi ja muutenkin suorituskykyinen laite. Virhe ilmeni siten, että pelin introtekstissä tullut keskeytys aiheutti koko introtekstilaatikon hyppäämistä aivan ruudun yläreunaan tai kokonaan pois kuvasta. Tämäkään virhe ei ollut toistettavuudeltaan helpoimmasta päästä vaan vaati useita toistokertoja. Lisäksi mikä tahansa keskeytys ei riittänyt, vaan keskeytyksen täytyi olla puhelinsoitto, joka kesti 7-10 sekuntia. Ongelman tarkemmissa tutkimuksissa selvisi, että virhe johtui tekstikentän piirrosta käytettävästä `System.currentTimeMillis()`-metodista, joka palautti välillä (ei siis aina) vääriä arvoja keskeytyksen loppumisen jälkeen. Näin ollen täytyi siis toteuttaa ratkaisu, joka ei luottanut `System.currentTimeMillis()`-metodiin.

4.2.2 Keskeytysverhon katoaminen

Universomon peleissä on pidetty standardina, että keskeytyksien jälkeen piirretään aina niin kutsuttu keskeytysverho pelitilan päälle. Verho on siis vain ruudun alareunan peittävä musta suorakaide, jossa on teksti ”paused”, ja softkey-valintana on ainoastaan ”resume”-valinta. LEGO Star Wars II -projektin aikana ilmeni kuitenkin uudenlaista ongelmaa tämän verhon piirtoon liittyen, jota ei koskaan aiemmin ollut tullut vastaan. Virhettä ilmeni muun muassa Cingularin LG C2000:ssa, jossa se oli helpoiten toistettavissa. Nopealla asetusten vaihtelulla ja koodin pienillä muutoksilla toistumisprosentti saatiin hyvin pieneksi eli noin 5 % luokkaan. Tämä ei tietenkään ole riittävä tulos vaan ongelman todellinen syy täytyi ratkaista, että tästä ongelmasta päästäisiin kokonaan. Kävikin ilmi, että näissä laitteissa säie, jossa keskeytysverhon piirto tehtiin, lopetettiin liian aikaisin. Tämä kuitenkin korjaantui ylimääräisen `synchronized()`-käskyn avulla. Ratkaisun jäljille päästiin, kun eräällä toistokerralla verhon ”paused” tekstistä jäi puuttumaan ”sed”-loppu. Tätä ei voitu selittää mitenkään muuten kuin piirtosuorituksen liian aikaisella loppumisella.

4.2.3 Samsung A620 ja LG pm325 ongelmat taustakuvan piirrosta

Ehkäpä koko projektin oudoin virhe tuli vastaan Sprintin Samsung A620:ssä, mutta esiintyi myös hieman pahempana versiona Sprintin LG pm-325:ssä. Virhe ilmeni Samsung A620:ssä siten, että kaikissa pelialueissa joissain kohdissa piirtyi vaakasuora viiva. Viivan sijainti vaikutti aluksi sattumanvaraiselta, mutta ilmeni kuitenkin aina samassa kohdassa kullakin pelialueella. Virhettä tarkemmin tutkiessa todettiin, että taustakuvan alin rivi korruptoitui piirrosta niin, että sille ei piirtynytäkään haluttua kuvaa, vaan alin pikselirivi jäi valkoiseksi, joka siis oli kuvan alkuperäisenä taustaväriinä. LG pm-325:ssä tämä virhe esiintyi muutamalla pelialueella oikeassa reunassa suorakaiteen muotoisena valkoisena alueena. Tätä valkoista aluetta ei kuitenkaan näkynyt jokaisessa alueessa, siis toisin kuin Samsung A620:n virhe. Koska taustakuvan piirrosta ei ollut ongelmia niin emulaattorissa kuin missään muussakaan laitteessa, täytyi vika olla itse laitteen tavassa hoitaa double buffer -kuvaa. Double buffer siis tarkoittaa tekniikkaa, miten grafiikka piirretään ruudulle. Tässä tekniikassa piirto-operaatiot tehdään ensin muistissa olevaan kuvaan, joiden jälkeen kuva piirretään kokonaisuudessaan ruudulle. Tämä poistaa ruudun välkkymistä, jota ilmenisi, jos kaikki graafiset operaatiot suoritettaisiin suoraan ruudulle. Koska double buffer -kuvan käyttö oli pakollista, sillä muuta tapaa ei ollut projektiin toteutettu, ainut jäljelle jäävä ratkaisu oli pienentää käytettävää double buffer -kuvaa, joksi siis tätä muistissa olevaa manipuloitavaa kuvaa kutsutaan. Peliprojektiin oli jo valmiiksi suunniteltu ja toteutettu niin kutsuttu ”reduced viewport” -asetus, jolla näkyvää pelialuetta pienennettiin koko ruudun maksimista. Tällä saatiin suorituskyvyltään huonoimpiin laitteisiin hieman lisää vauhtia ruudunpäivitysnopeuteen. Tämäkään pienennys ei suoraan riittänyt näillä kahdella laitteella vaan tästä valmiiksi pienennetystä taustakuvasta täytyi toteuttaa uusi vieläkin pienempi versio, jossa taustakuvan korkeutta ja leveyttä oli kutakin vähennetty 12 pikselillä

4.2.4 LG pm325 taustakuvan luonnin epäonnistuminen

Sprintin LG pm325:ssa löytyi asiakkaan testauksessa toinenkin taustakuvaan liittyvä ongelma yllä kuvatun valkoisen alueen lisäksi. Ongelma oli, että pelialueen taustakuva katosi, kun samaa kenttää uudelleen käynnistettiin riittävän monta kertaa peräkkäin. Yleensä tämä virhe ilmeni toisessa pelialueessa kuudennen toistokerran jälkeen. Kun taustakuvan luonti epäonnistui oletettavasti muistin puutteeseen, pelihahmot piirtyivät mustalle taustalle, mutta koska taustaa ei ikinä piirretty uudelleen, hahmoista jäi liikkeessa varjokuvat jälkeensä. Aluksi tätä yritettiin korjata luomalla taustakuva uudestaan heti perään, mikäli ensimmäinen yritys oli epäonnistunut, mutta tässä kävi kuitenkin niin, että toinenkin taustakuvan luonti epäonnistui. Tällöin tosin toistokertoja vaadittiin huomattavasti enemmän. Alustavassa tarkistuksessa virhettä ei saatu toistumaan edes 20 yrityksellä. Kuitenkin testausryhmä tässä jotenkin onnistui. Tämän jälkeen taustakuvan luonti laitettiin while-silmukan sisään, jonka pois pääsyn ehtona oli, että taustakuva oli onnistuneesti luotu. Lisäksi while-silmukkaan lisättiin pieni odotus, jotta Java virtuaalikone ehtisi kuvanluonnin väleissä hoitamaan muistin uudelleen järjestämistä ja roskien keruuta. Korjaus on sikäli riskialtis, koska jos kuvanluonti ei onnistu esimerkiksi muistin vähyyden takia ikinä, pelin suoritus jää kokonaan jumiin ikuiseseen while-silmukkaan. Toisaalta tämäkään tapaus ei olisi huomattavasti koko taustakuvan puuttumista huonompi vaihtoehto, sillä molemmat tapaukset ovat liki yhtä fataaleja pelaajalle.

5 YHTEENVETO

LEGO Star Wars II -projektin yhtenä lähtökohtana oli, että projektin pitäisi sujua kohtuullisen mutkattomasti, eikä suuria ongelmia pitäisi syntyä. Tämä ajatus johtui siitä, että osalle kohdelaitteista oli jo tehty Euroopan markkinoille versiot ja Amerikan markkinoilla olevat puhelinlaitteet eivät eroa kovinkaan paljon eurooppalaisista serkuistaan. Tämä oletus osui kohtalaisen hyvin oikeaan, joskin yllättäviä ongelmia ilmeni odottamattoman paljon. Laitteiden eroavaisuuksista kertoo ehkä eniten se, että vain muutamat vastaan tulleet ongelmat olivat ennestään tuttuja aikaisemmista projekteista. Kuitenkin ne ongelmat, jotka olivat tulleet Euroopan sovituksissa vastaan aiemmin, olivat myös näissä laitteissa helposti korjattavissa. Ilmeisin syy, miksi projektin kuluessa nousi paljon uusia ongelmia, oli Game Lobbyn sekä lisenssipalvelun lisääminen projektiin. Näitä ei ole totuttu tekemään Euroopan markkinoille.

Jatkossa sovitustyön oletetaan käyvän yhä helpommaksi, kun vanhat laitteet ja niiden eri ongelmat käyvät tutuiksi, ja niihin on toteutettu jo valmiit ratkaisut. Tämän lisäksi niiden oletetaan putoavan pian pois koko sovitustilastoilta, kun niiden markkinaosuudet käyvät tarpeeksi pieniksi. Lisäksi sovitustyö tulee helpottumaan senkin takia, että markkinoille tulevat uudet laitteet ovat jo nyt hyvin kehittyneitä ja niissä on paljon muistia sekä suorituskykyä muita ominaisuuksia unohtamatta, että niille omien versioiden tekeminen tulee olemaan varsin suoraviivaista. Ei kuitenkaan saa unohtaa, että myös sovitustyön kohteena olevat mobiilipelit tulevat laitteiden mukana kehittymään ja monimutkaistumaan, niin kuin PC- ja konsolipelipuoleltakin tiedetään. Lisäksi kaikki uudetkaan laitteet eivät aina ole täysin ongelmattomia ja tarjoavat varmasti uusia haasteita sovitustyölle.

LÄHTEET

- 1 Universomo: LEGO Star Wars II. [www-sivu]. [viitattu 17.4.2007]
Saataavissa:
<http://www.universomo.com/internet/game-legosw2.html>

- 2 Star Wars Episode IV: A New Hope. [www-sivu]. [viitattu 17.4.20]
Saataavissa:
http://en.wikipedia.org/wiki/Star_Wars_Episode_IV:_A_New_Hope

- 3 Forum Nokia - resources for mobile applications developers.
[www-sivu]. [viitattu 19.4.2007] Saataavissa:
<http://forum.nokia.com/devices/3510i>
<http://forum.nokia.com/devices/6101>
<http://forum.nokia.com/devices/6630>
<http://forum.nokia.com/devices/6270>
<http://forum.nokia.com/devices/E61>
<http://forum.nokia.com/devices/N90>

- 4 Yamaha - Outline of SMAF specification. [www-sivu]. [viitattu 20.4]
Saataavissa:
http://smaf-yamaha.com/what/smaf_spec.html

LIITTEET

Kohdelaiteluettelo

| NameO2 | Carrier |
|-----------------------------------|----------|
| Motorola i275 (JAVA) | Boost |
| Motorola i415 (JAVA) | Boost |
| Motorola i450 (JAVA) | Boost |
| Motorola i560 (JAVA) | Boost |
| Motorola i580 (JAVA) | Boost |
| Motorola i670 (JAVA) | Boost |
| Motorola i730 (JAVA) | Boost |
| Motorola i830 (Roxy) (JAVA) | Boost |
| Motorola i833 (JAVA) | Boost |
| Motorola i835 (JAVA) | Boost |
| Motorola i855 (JAVA) | Boost |
| Motorola i860 (JAVA) | Boost |
| Motorola i875 (JAVA) | Boost |
| Motorola i885 (JAVA) | Boost |
| LG C1300 (JAVA) | Cingular |
| LG C1500 (JAVA) | Cingular |
| LG C2000 (JAVA) | Cingular |
| LG CG225 (JAVA) | Cingular |
| LG CU400 (JAVA) | Cingular |
| LG CU500 (JAVA) | Cingular |
| Motorola IZAR Z3 (JAVA) | Cingular |
| Motorola K1 KRZR (JAVA) | Cingular |
| Motorola L7 SLVR (JAVA) | Cingular |
| Motorola V180 (JAVA) | Cingular |
| Motorola V220 (JAVA) | Cingular |
| Motorola V3 RAZR (JAVA) | Cingular |
| Motorola V365 (JAVA) | Cingular |
| Motorola V3i RAZR (JAVA) | Cingular |
| Motorola V3r RAZR (JAVA) | Cingular |
| Motorola V3XX RAZR (JAVA) | Cingular |
| Motorola V551 (JAVA) | Cingular |
| Motorola V557 (JAVA) | Cingular |
| Nokia 2610 (JAVA) | Cingular |
| Nokia 6030 (JAVA) | Cingular |
| Nokia 6061 (JAVA) | Cingular |
| Nokia 6102 (JAVA) | Cingular |
| Nokia 6102i / 6103 (JAVA) | Cingular |
| Nokia 6126 (formerly 6131) (JAVA) | Cingular |
| Nokia Cougar (JAVA) | Cingular |
| Nokia N75 (JAVA) | Cingular |
| Pantech PG-C120 (JAVA) | Cingular |

| | |
|---|----------|
| Pantech PG-C300 (JAVA) | Cingular |
| Samsung Panther (JAVA) | Cingular |
| Samsung SGH-A707 (JAVA) | Cingular |
| Samsung SGH-C417 (JAVA) | Cingular |
| Samsung SGH-D407 / D347 (JAVA) | Cingular |
| Samsung SGH-D807 (JAVA) | Cingular |
| Samsung SGH-X497 (JAVA) | Cingular |
| Sony Ericsson W300i (JAVA) | Cingular |
| Sony Ericsson W600i (JAVA) | Cingular |
| Sony Ericsson W810i (JAVA) | Cingular |
| Sony Ericsson Z500a (JAVA) | Cingular |
| Sony Ericsson Z520a (JAVA) | Cingular |
| Audiovox CDM-8920 (JAVA) | Sprint |
| LG 150 (JAVA) | Sprint |
| LG LX-350 (JAVA) | Sprint |
| LG LX550 (Fusic) (JAVA) | Sprint |
| LG PM-225 (JAVA) | Sprint |
| LG PM-325 (JAVA) | Sprint |
| LG VI-125 (JAVA) | Sprint |
| Motorola C290 (JAVA) | Sprint |
| Motorola K1m KRZR (JAVA) | Sprint |
| Motorola L7c SLVR - Honolulu (JAVA) | Sprint |
| Motorola Racer - ic502 (JAVA) | Sprint |
| Motorola RacerLite - ic402 (JAVA) | Sprint |
| Motorola V3m RAZR (JAVA) | Sprint |
| Nokia 3155i / VI-3155 (JAVA) | Sprint |
| Nokia 6165i (JAVA) | Sprint |
| Samsung A580 (JAVA) | Sprint |
| Samsung M500 (JAVA) | Sprint |
| Samsung M610 Glint (JAVA) | Sprint |
| Samsung MM-A700 / SPH-A700 (JAVA) | Sprint |
| Samsung MM-A880 (JAVA) | Sprint |
| Samsung MM-A940 (JAVA) | Sprint |
| Samsung SPH-A620 / VGA 1000 (JAVA) | Sprint |
| Samsung SPH-A640 (JAVA) | Sprint |
| Samsung SPH-A660 / VI660 (JAVA) | Sprint |
| Samsung SPH-A680 / VM-A680 (JAVA) | Sprint |
| Samsung SPH-A740 / PM-A740 (JAVA) | Sprint |
| Samsung SPH-A760 / RL-A760 (JAVA) | Sprint |
| Samsung SPH-A840 / PM-A840 (JAVA) | Sprint |
| Samsung SPH-A900 / MM-A900 / Ace (JAVA) | Sprint |
| Samsung SPH-A900M / SPH-A900P / MM-A900P (JAVA) | Sprint |
| Samsung SPH-A920 / MM-A920 (JAVA) | Sprint |
| Sanyo 2400 (JAVA) | Sprint |
| Sanyo 6600 Katana (JAVA) | Sprint |
| Sanyo 7000 (JAVA) | Sprint |
| Sanyo 8400 (JAVA) | Sprint |
| Sanyo M1 (JAVA) | Sprint |

| | |
|---|---------------|
| Sanyo MM-5600 (JAVA) | Sprint |
| Sanyo MM-7400 (JAVA) | Sprint |
| Sanyo MM-7500 (JAVA) | Sprint |
| Sanyo SCP-3100 (JAVA) | Sprint |
| Sanyo SCP-4920 / RL-4920 (JAVA) | Sprint |
| Sanyo SCP-4930 (JAVA) | Sprint |
| Sanyo SCP-5500 / VM 4500 (JAVA) | Sprint |
| Sanyo SCP-7300 / RL-7300 (JAVA) | Sprint |
| Sanyo SCP-8100 (JAVA) | Sprint |
| Sanyo SCP-8200 / PM-8200 (JAVA) | Sprint |
| Sanyo SCP-8300 / MM-8300 (JAVA) | Sprint |
| Sanyo SCP-9000 / MM-9000 (JAVA) | Sprint |
| Sanyo VI-2300 / SCP-2300 (JAVA) | Sprint |
| Motorola i265 (JAVA) | Sprint/Nextel |
| Motorola i275 (JAVA) | Sprint/Nextel |
| Motorola i355 (JAVA) | Sprint/Nextel |
| Motorola i355 TTS (JAVA) | Sprint/Nextel |
| Motorola i560 (JAVA) | Sprint/Nextel |
| Motorola i580 (JAVA) | Sprint/Nextel |
| Motorola i605 (JAVA) | Sprint/Nextel |
| Motorola i605 (With Game Lobby) (JAVA) | Sprint/Nextel |
| Motorola i615 (JAVA) | Sprint/Nextel |
| Motorola i670 (JAVA) | Sprint/Nextel |
| Motorola i710 (JAVA) | Sprint/Nextel |
| Motorola i730 (JAVA) | Sprint/Nextel |
| Motorola i733 (JAVA) | Sprint/Nextel |
| Motorola i736 (JAVA) | Sprint/Nextel |
| Motorola i760 (JAVA) | Sprint/Nextel |
| Motorola i760 (With Game Lobby) (JAVA) | Sprint/Nextel |
| Motorola i830 (Roxy) (JAVA) | Sprint/Nextel |
| Motorola i833 (JAVA) | Sprint/Nextel |
| Motorola i836 (JAVA) | Sprint/Nextel |
| Motorola i850 (JAVA) | Sprint/Nextel |
| Motorola i850 (With Game Lobby) (JAVA) | Sprint/Nextel |
| Motorola i860 (JAVA) | Sprint/Nextel |
| Motorola i860 (With Game Lobby) (JAVA) | Sprint/Nextel |
| Motorola i870 (JAVA) | Sprint/Nextel |
| Motorola i870 (With Game Lobby) (JAVA) | Sprint/Nextel |
| Motorola i875 (JAVA) | Sprint/Nextel |
| Motorola i880 (i885) (JAVA) | Sprint/Nextel |
| Motorola i880 (i885) (With Game Lobby) (JAVA) | Sprint/Nextel |
| Motorola PEBL U6 (JAVA) | T-Mobile |
| Motorola V180 (JAVA) | T-Mobile |
| Motorola V188 (JAVA) | T-Mobile |
| Motorola V195 / V197 (JAVA) | T-Mobile |
| Motorola V3 RAZR (JAVA) | T-Mobile |
| Motorola V300 (JAVA) | T-Mobile |
| Motorola V330 (JAVA) | T-Mobile |

| | |
|--|----------|
| Motorola V360 (JAVA) | T-Mobile |
| Nokia 3220/3220i (JAVA) | T-Mobile |
| Nokia 3595 (JAVA) | T-Mobile |
| Nokia 6010 (JAVA) | T-Mobile |
| Nokia 6030 (JAVA) | T-Mobile |
| Nokia 6101 / 6102 (JAVA) | T-Mobile |
| Nokia 6103 (JAVA) | T-Mobile |
| Samsung SGH-E335 (JAVA) | T-Mobile |
| Samsung SGH-E635 (JAVA) | T-Mobile |
| Samsung SGH-T209 / X495 / X497 / X496 (JAVA) | T-Mobile |
| Samsung SGH-T309 (JAVA) | T-Mobile |
| Samsung SGH-T509 (JAVA) | T-Mobile |
| Samsung SGH-T519 Trace (JAVA) | T-Mobile |
| Samsung SGH-T609 (JAVA) | T-Mobile |
| Samsung SGH-T619 (JAVA) | T-Mobile |
| Samsung SGH-T809 / SGH-D820 (JAVA) | T-Mobile |
| Samsung SGH-X495 (JAVA) | T-Mobile |
| Samsung T319 (JAVA) | T-Mobile |
| Samsung T329 (JAVA) | T-Mobile |
| Samsung T629 (JAVA) | T-Mobile |
| Samsung T709 (JAVA) | T-Mobile |
| Audiovox CDM-8910 Flasher / PM-8912 (JAVA) | Virgin |
| Audiovox CDM-8915 Snapper (UT Starcom) (JAVA) | Virgin |
| Kyocera K325 Cyclops (aka Thunder / K24) (JAVA) | Virgin |
| Kyocera KX21 Switch-Back / KX21-2XO Storm / KX612b Strobe (JAVA) | Virgin |
| Kyocera KX5C Slider Sonic (JAVA) | Virgin |
| UT Starcom 1400 Slice (JAVA) | Virgin |