

Matti Lintilä

MOBIILIPELIN KÄYTTÖLIITTYMÄN SUUNNITTELU

MOBIILIPELIN KÄYTTÖLIITTYMÄN SUUNNITTELU

Matti Lintilä
Mobiilipelin käyttöliittymän suunnittelu
Syksy 2016
Tietojenkäsittelyn koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma, internetpalvelut ja digitaalinen media

Tekijä: Matti Lintilä
Opinnäytetyön nimi: Mobiilipelin käyttöliittymän suunnittelu
Työn ohjaaja: Pekka Ojala
Syksy 2016

Sivumäärä: 33

Tammikuussa 2016 Oulu Game LAB-tiloissa ryhmä opiskelijoita aloitti kehittämään mobiilipeliä nimeltä Riverian Pirates. Liityin mukaan ryhmään helmikuussa 2016 ja olen siitä lähtien toiminut ryhmässä käyttöliittymän suunnittelijana. Aiheen opinnäytetyöhön sain tehdessäni tutkimustyötä käyttöliittymän suunnittelusta. Raportin kirjoittamisen aloitin syyskuussa 2016. Työn toimeksiantajana toimin itse.

Opinnäytetyön aiheena oli suunnitella ja toteuttaa käyttöliittymä työn alla olevalle mobiilipelille Riverian Pirates. Käyttöliittymän tuli olla toimiva ja visuaalisesti näyttävä. Käyttöliittymä toteutettiin Unity-pelimootorilla. Työvälineinä käytettiin myös MonoDevelop-kehitysympäristöä ja GIMP-kuvankäsittelyohjelmaa.

Raportti koostuu kolmesta pääosasta. Se sisältää perehdytyksen käyttöliittymän suunnitteluun, itse suunnittelun ja toteutuksen. Opinnäytetyön tuloksena oli keskeneräinen, mutta näyttävä ja täysin toimiva käyttöliittymä. Riverian Pirates -mobiilipeliä ei ole vielä julkaistu ja sen kehittäminen jatkuu edelleen.

Asiasanat: mobiilipeli, käyttöliittymä, suunnittelu

ABSTRACT

Oulu University of Applied Sciences
Degree programme, option

Author: Matti Lintilä

Title of thesis: User interface design for a mobile game

Supervisor: Pekka Ojala

Term and year when the thesis was submitted: November 2016 Number of pages: 33

In January 2016 at the Oulu Game LAB premises, a group of students began to develop a mobile game called Riverian Pirates. In February 2016 I joined this group and ever since I have worked as their user interface designer. The topic of my thesis I got while I was doing a research about user interface design. I started writing a report in September 2016. I'm my own client.

The topic of my thesis was to design user interface for a work-in-progress mobile game called Riverian Pirates. The user interface had to be functional and visually stunning at the same time. The user interface was implemented with Unity game engine. Also MonoDevelop environment and the GIMP image manipulation program were used as a tools.

The report consists of three main parts. It contains orientation to the user interface design, the design part itself and it's implementation. For the thesis, the current user interface is incomplete, yet stunning and fully functional for use. Riverian Pirates mobile game has not yet been published and the development still continues.

Keywords: Mobile game, user interface, design

SISÄLLYS

1	JOHDANTO	6
2	KÄYTTÖLIITTYMÄÄN JA SEN SUUNNITTELUUN PEREHTYMINEN	7
3	SUUNNITTELU.....	9
3.1	Suunnittelun alkuvaiheet	9
3.2	Työnkulku ja työvälineet	11
3.3	Käyttöliittymäsuunnittelun eri vaiheet	13
4	TOTEUTUS	17
4.1	Taistelu.....	19
4.2	Yleinen	25
4.3	Kartta.....	26
5	TULOKSET JA NIIDEN TARKASTELU	28
5.1	Taistelunäkymän tulokset.....	28
5.2	Yleisen-näkymän tulokset.....	29
5.3	Karttanäkymän tulokset.....	30
6	POHDINTA	31
	LÄHTEET.....	32

1 JOHDANTO

Tämän opinnäytetyön aiheena on suunnitella ja toteuttaa käyttöliittymä mobiilipelille Riverian Pirates. Aiheen opinnäytetyöhön sain tehdessäni tutkimustyötä käyttöliittymän suunnittelusta Game Lab -opintojeni aikana. Pelin kehittäminen alkoi tammikuussa 2016 Oulu Game Lab -opiskelijoiden toimesta. Liityin itse mukaan ryhmään vasta helmikuussa 2016 ja olen siitä asti toiminut ryhmämme käyttöliittymän suunnittelijana. Riverian Pirates on edelleen keskeneräinen peliprojekti ja sen kehitys jatkuu edelleen. Tämän opinnäytetyön toimeksiantajana toimin minä itse.

Riverian Pirates on työn alla oleva ilmainen mobiilipeli Android- ja iOS-alustoille, missä strategiset taistelut käydään reaaliajassa. Pelin tarkoituksena on seilata satunnaisesti tuotettujen karttojen halki ja kehittää merirosvojoukkoaan sekä laivastoaan aina vain entistä paremmaksi.

Raportissa perehdytään pääasiassa mobiilipelin käyttöliittymään ja sen suunnitteluun. Koska Riverian Pirates on edelleen keskeneräinen projekti, sen koko toteutusvaihetta ei käydä tässä raportissa läpi. Toteutuksen ja suunnittelun lisäksi raportissa käydään lyhyesti läpi hyvän käyttöliittymän ominaisuuksia, sekä työvälineitä sen toteuttamiseen. Peliä kehitetään Unityn pelimoottorilla ja sitä käytetään myös käyttöliittymän toteuttamiseen.

2 KÄYTTÖLIITTYMÄÄN JA SEN SUUNNITTELUUN PEREHTYMINEN

Käyttöliittymä (User Interface) on osa päätelaitteen ohjelmistoa, joka mahdollistaa vuorovaikutuksen käyttäjän ja käytettävän laitteen tai ohjelman välillä. Käyttöliittymä voi pohjautua graafiseen, tekstipohjaiseen tai auditiiviseen informaatioon, jota ohjelma välittää käyttäjälleen. (Tutorialspoint 2016, hakupäivä 30.9.2016.)

Käyttöliittymä käsitteenä voi olla erittäin laaja. Peleistä puhuttaessa se tarkoittaa usein informaatiota jota käyttäjä näkee ruudulla. Tämän lisäksi se voi tarkoittaa myös tapaa jolla pelaaja kykenee vaikuttamaan pelin kulkuun. Tässä työssä se on hiukan molempia.

Käyttöliittymällä voidaan siis yhdistää pelaaja peliin niin, että tämä kykenee näkemään pelimaailman tapahtumat ja vaikuttamaan sen kulkuun. Vaikuttaminen pelin aikana voi tapahtua esimerkiksi painamalla puhelimen kosketusnäytöstä heijastuvaa nappia. Tämä nappi ei välttämättä edes esiinny itse pelimaailmassa, vaan se on luotu näkymään ainoastaan pelaajalle. Kuitenkin, jos pelaaja uskaltautuu painamaan tätä nappia, saattaa pelimaailmassa olevasta kanuunasta lentää tykinkuula tai vaikkapa kymmenen vihamielistä lintua.



KUVIO 1. Esimerkki käyttöliittymästä (Kuvankaappaus pelistä Angry Birds Rio 2016)

Tässä työssä käsiteltävä käyttöliittymän suunnittelu ja toteutus on juuri tätä: Visuaalisten UI-elementtien tuomista ruudulle niin, että se olisi pelaajan kannalta helposti lähestyttävä ja hyvä käyttää.

Hyvä käyttöliittymä

Mikä sitten tekee käyttöliittymästä hyvän? Objektivisesti sitä ei tässä raportissa aleta mittaamaan, sillä jokaisella pelaajalla on oma käsityksensä ”hyvästä”. Tätä voidaan kuitenkin lähteä pohtimaan esittämällä kysymys: Mitä hyvä käyttöliittymä oikein tekee?

Yksinkertaisimmillaan, hyvä käyttöliittymä kertoo käyttäjälleen juuri sen mitä tämän tulee sillä hetkellä tietää, eikä mitään ylimääräistä. Esitettävän informaation tulee olla selvää ja helposti saatavilla, niin ettei käyttäjän tarvitse miettiä pitkään miten edetään. Etenemisen tulee olla vaivatonta, jotta käyttäjän ei tarvitse tehdä turhia toistoja. (Quintans 2013, hakupäivä 30.9.2016.)

Hyvä suunnittelija

Kenestä sitten on käyttöliittymän suunnittelijaksi? Monet ohjelmoijat karttavat käyttöliittymän suunnittelutyötä, koska se koetaan jokseenkin taiteellisena. Mitä se tavallaan onkin. Mikäli suunnittelijalla itsellään on graafista osaamista, se on aina hyödyksi. Mutta tämä ei suinkaan ole välttämätöntä. Suunnittelijan parhaita ominaisuuksia ovat kuitenkin kokemukset käyttöliittymistä ja kyky hahmottaa niiden kokonaisuuksia.

Käyttöliittymän suunnittelun kannalta on siis olennaista, että suunnittelijalla itsellään on jo entuudestaan kokemusta eri pelien käyttöliittymistä. Mikäli tätä ei ole, nopein tapa hankkia sitä on yksinkertaisesti pelata paljon pelejä ja pelaamisen ohella tutkia samalla, mikä niiden käyttöliittymissä on toimivaa ja mikä ei. Mitä enemmän oppii näkemään visuaalisen ulkokuoren lävitse ja tunnistamaan sen takana olevaa suunnittelua, sitä helpompaa on itse alkaa suunnitella toimivaa kokonaisuutta.

3 SUUNNITTELU

3.1 Suunnittelun alkuvaiheet

Alusta ja resoluutio

Ensimmäinen askel suunnittelussa on selvittää mitä alustaa suunniteltava käyttöliittymä tukee. Riverian Piratesin tapauksessa peli julkaistaan iOS- ja Android- alustoille. Näin ollen suunnittelussa on myös ensisijaisen tärkeää huomioida tukevien laitteiden resoluutio. Esim. iPhone 3GS:n resoluutio on 480x320, kun taas iPhone4 ja iPhone 4S:n resoluutio on 960x640 (Derk 2012, Hakupäivä 1.10.2016).

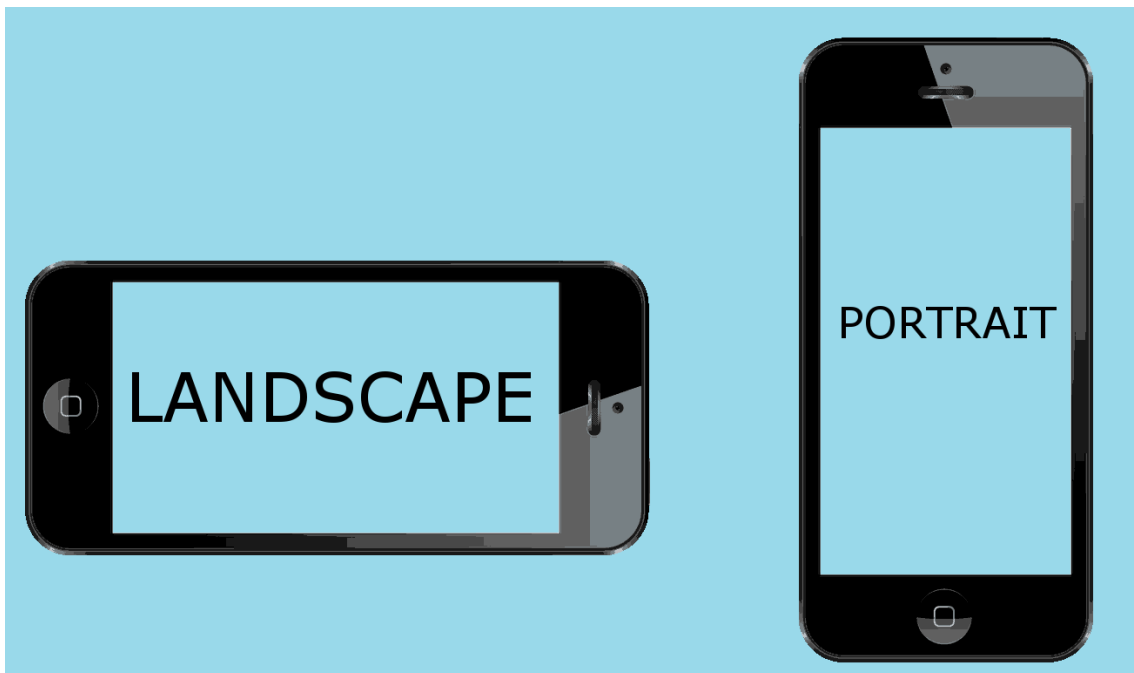
Riverian Pirates-peli suunnitellaan niin, että se tukee kaikkia mahdollisia resoluutioita, joten niitä ei erikseen tähän ole listattu. Alla on kuitenkin taulukko laitteista ja resoluutioista, joilla Riverian Pirates pelin käyttöliittymää testataan.

TAULUKKO 1. Testaus laitteet ja resoluutiot

Valmistaja	Malli	Kuvasuhde	Resoluutio	Tyyppi
Samsung	Galaxy Tab A	4:3	1024x768	Tabletti
HP	Pro Slate 8	4:3	2048x1536	Tabletti
Huawei	Honor 7	16:9	1080x1920	Älypuhelin

Orientaatio

Kun resoluutiot on selvitetty, seuraava kysymys on pelin orientaatio. Tämä pieni, mutta kuitenkin oleellinen osa käyttöliittymää unohtuu helposti ja se on hyvä päättää, ennen kuin varsinaista suunnittelua edes aloitetaan. Orientaatioita on kaksi, Landscape (vaakasuora) ja Portrait (pystysuora). Yleensä peli lukitaan jompaankumpaan, mutta on myös pelejä jotka nimenomaan on suunniteltu tukemaan molempia vaihtoehtoja. Päätös pelin orientaatiosta kannattaa tehdä yksinkertaisesti sen perusteella, kumpi on pelattavuuden kannalta parempi.



KUVIO 2. Puhelimen orientaatiomalli

Riverian Pirates peli lukitaan Landscape-orientaatioon, niin tabletilla kuin puhelimella. Kun orientaatio on päätetty, päätöksestä kannattaa pitää kiinni loppuun saakka, sillä sitä ei ole jälkeenpäin niin yksinkertaista muuttaa.

Toteutuksen suunnittelu

Käyttöliittymän suunnitteluun riittää periaatteessa pelkkä kynä ja paperi, mutta sen toteuttamiseen tarvitaan muutakin. Ensiksi on hyvä lähteä liikkeelle siitä, toteutetaanko kaikki UI-grafiikka itse alusta alkaen vai ei. Mikäli taitoa ja graafista silmää löytyy, on visuaalisen ulkoasun toteuttaminen itse aina parempi vaihtoehto. Jos kuitenkin kokee, että grafiikoiden tekeminen omin käsin on liian taiteellista tai aikaa vievää työtä, niitä voi ostaa esimerkiksi Unity Asset Storesta. On myös olemassa valmiita UI-pohjia, mutta niiden käyttäminen vie pohjan koko suunnitteluprosessilta.

Riverian Pirates-pelin tapauksessa käyttöliittymän grafiikat tekee tiimin graafikko. Usein käyttöliittymä hahmotellaan ensin paperille ennen grafiikkojen tuottamista. Joissakin tapauksissa suunnitellaan käyttöliittymä ensiksi placeholdereita, eli niin sanottuja ”paikanpitäjiä” käyttäen. Niitä voivat olla esimerkiksi Unityn omat perus assetit. Kun hahmottelu on valmis, graafikon on helpompi alkaa toteuttamaan lopullisia grafiikoita.

3.2 Työnkulku ja työvälineet

Työnkulku on seuraava: Valmiit grafiikat käsitellään kuvankäsittelyohjelmalla, jonka jälkeen ne tuodaan Unityn editoriin. Toteuttaminen tapahtuu itse editorissa. Editorin lisäksi tullaan käyttämään myös MonoDevelopia muutamien visuaalisten efektien koodaamiseen.

Unity

Unity on Unity Technologiesin kehittämä pelimoottori. Sillä voidaan tehdä kaksi- tai kolmiulotteisia pelejä eri alustoille. Unitystä on saatavilla neljä eri versiota, Personal, Plus, Pro ja Enterprise (Unity 3D 2016.) Eroavaisuudet näiden versioiden välillä ovat suurimmaksi osaksi niiden lisäominaisuudet ja lisenssioikeudet.

Tässä työssä käytän Unityn Personal-versiota, joka on ilmainen. Tällä toteutetaan sekä peli, että sen käyttöliittymä.

MonoDevelop

MonoDevelop on avoimeen lähdekoodiin pohjautuva ilmainen sovellus. Se on kehitysympäristö, jolla voidaan tehdä sovelluksia eri alustoille. MonoDevelop tukee muuan muassa GTK#, C#- ja muita NET-kieliä (Monodevelop 2016.)

MonoDevelop sovellusta voidaan käyttää Unityn kanssa skriptien luomiseen. Vaikka työ painottuu pääsääntöisesti visuaalisen käyttöliittymän suunnitteluun ja järjestelyyn, tulen käyttämään toteuttamisen aikana myös C#-kieltä. MonoDevelop sovellus valittiin arvostetun Visual Studion sijaan yksinkertaisesti siksi, että se käynnistyy nopeammin eikä sen asentamisessa kestä kauan.

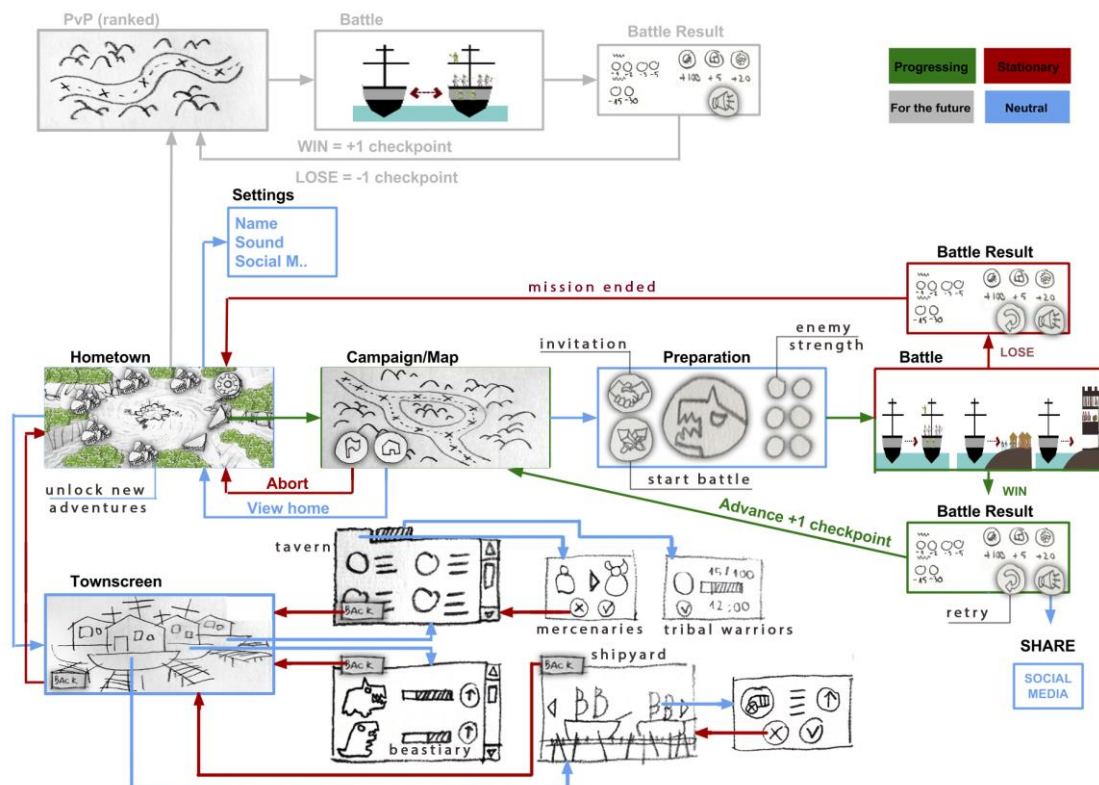
GIMP

GIMP on avoimen lähdekoodin ilmainen kuvankäsittelyohjelma. Se on saatavilla muun muassa GNU/Linux-, OS X- sekä Windows-käyttöjärjestelmiin (Gimp 2016.) GIMP-kuvankäsittelyohjelma valittiin siksi, että se on täysin ilmainen ja se kuitenkin vetää vertoja kaupallisille ohjelmille, kuten esimerkiksi Adoben Photoshopille.

GIMP-kuvankäsittelyohjelmaa käytetään grafiikoiden käsittelemiseen ennen Unity Editoriin tuomista.

3.3 Käyttöliittymäsuunnittelun eri vaiheet

Ennen kuin yksittäisiä käyttöliittymän näkymiä lähdetään suunnittelemaan, on hyvä olla ensin jokinlainen käsitys siitä, mitä tarkalleen ottaen ollaan tekemässä. Suunnitteluvaiheessa ryhmä listasi kaikki peliin tulevat näkymät/kohtaukset (scene) ja niistä lähdettiin luomaan pelinkulkua kuvaavaa kaaviota (flow chart), jonka pohjalta ryhmän graafikko ja projektin vetäjä lopulta toteuttivat seuraavan lopullisen kaavion.



KUVIO 3. Flow Chart (Tran, Valkola 2016)

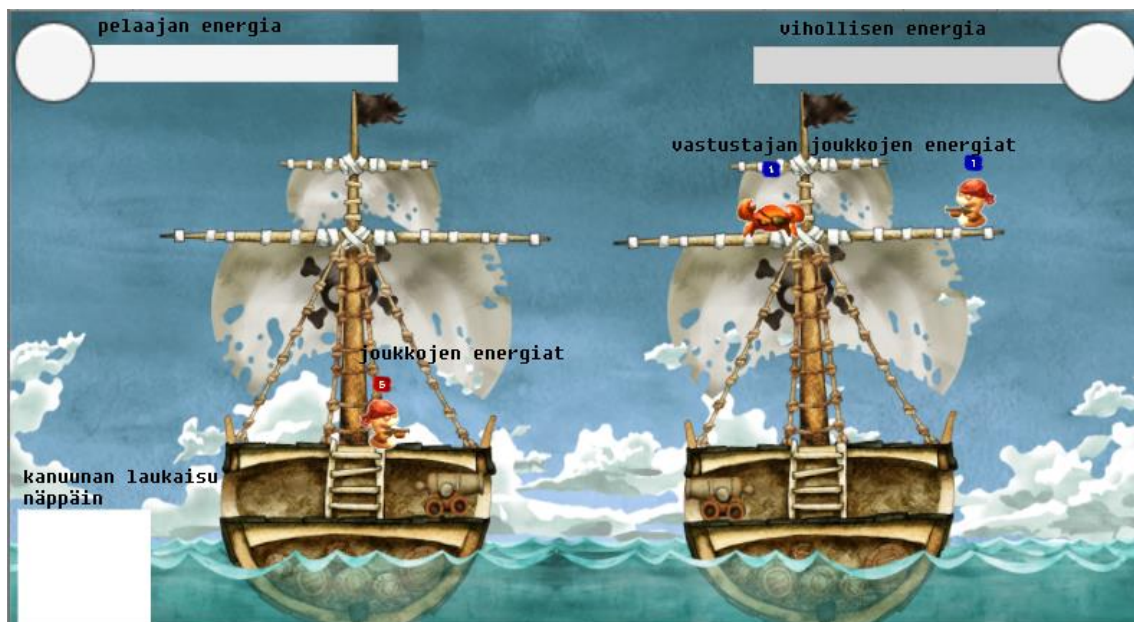
Yllä olevan kaavion pohjalta aloitettiin käyttöliittymän suunnittelu ja toteutus pelin eri näkymiin. Kuitenkin toteutuksen aikana peli on kokenut suuria muutoksia ja tästä syystä kaavion kulkukin on hiukan muuttunut alkuperäisestä. Tässä raportissa käsitellään ainoastaan Taistelu (Battle)-, Yleinen (General)- ja Kartta (Map)- näkymät.

Taistelu

Koska taistelukohtaus on Riverian Pirates pelin niin sanottu kohokohta, sen näkymän käyttöliittymän suunnittelu aloitettiin ensimmäisenä. Taistelukohtauksessa pelaajan tavoite on voittaa vastustaja joko upottamalla tämän laiva tai päihittämällä vastapuolen joukot. Tästä lähdettiin miettimään, mitä perustekijöitä pelaajaan tulisi taistelun aikana nähdä, taistelukohtauksen itsensä lisäksi. Lähdettiin liikkeelle seuraavista:

- Laivan energia
- Vastustajan laivan energia
- Joukkojen energia (Valmiina jo pelissä)
- Vastustajan joukkojen energia (Valmiina jo pelissä)
- Kanuunan laukaisunappi

Monien väliaikaisten käyttöliittymän hahmottelumallien jälkeen päästiin lopulta kaikkia miellyttävään ratkaisuun.



KUVIO 4. Käyttöliittymän hahmottelumalli

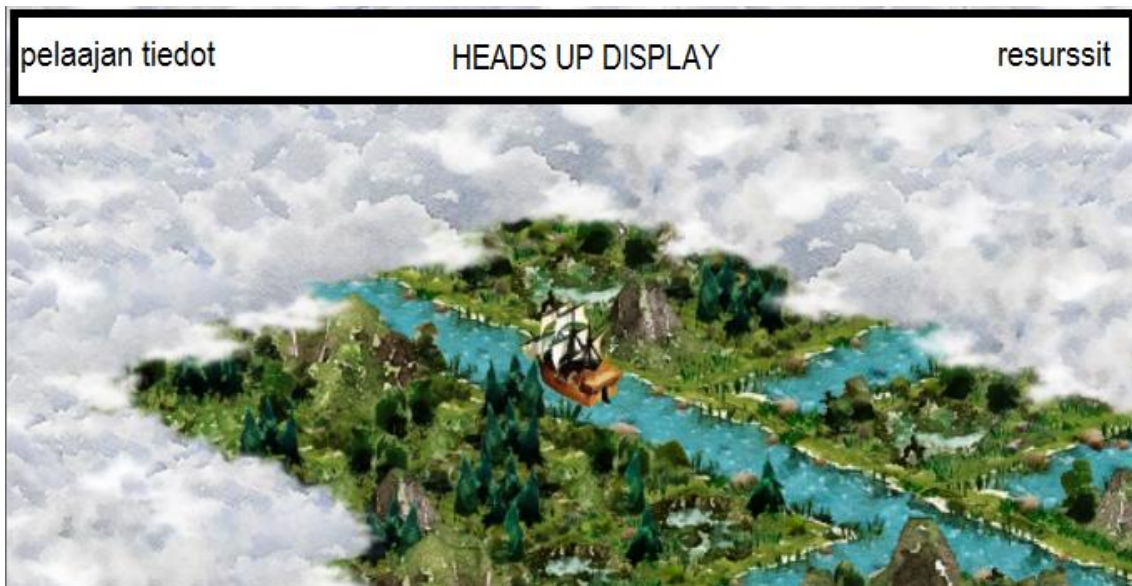
Laivojen energiapalkkien on oltava ruudun molemmissa yläkulmissa, kun taas joukkojen energioiden tulee näkyä itse hahmojen yläpuolella. Pelaajan tulee pystyä liikuttamaan joukkojaan valitsemalla hahmon ja alueen minne haluaa hahmon liikkuvan. Kanuunan laukaisunapissa tulee näkyä

latausaika, jonka täytyttyä pelaaja voi ampua. Kanuunalla voi ampua ainoastaan silloin kun jokin hahmoista on asetettuna sen viereen. Joukkojen ei tule pystyä liikkumaan samaan kohtaan. Kun taistelu on käyty, tulee ruudun keskelle ilmestyä voitto- tai häviö-teksti sen mukaan, miten taistelu päättyi.

Yleinen

Yleinen näkymä on se, mikä tulee näkymään jokaisessa näkymässä paitsi taistelussa. Tämän vuoksi tätä voitaisiin periaatteessa kutsua nimellä HUD, eli Heads-Up Display. Karkeasti suomennettuna tämä tarkoittaa "heijastusnäyttöä".

- Pelaajan tiedot
- Resurssit: Peliraha



KUVIO 5. HUD-ikkunan hahmottelu

Kartta

Kartalla liikkuminen on hyvin keskeisessä osassa Riverian Pirates peliä. Se toimii polkuna kotikaupungin ja jokitaistelujen välillä. Kartat toteutuvat satunnaisesti ja niissä liikutaan jokia pitkin. Pelaajan tavoite on löytää kartalta aarrearkku. Saadakseen aarteen, pelaajan on kuitenkin varauduttava kohtaamaan matkalla vihollislaivoja, joihin törmätessä pelaaja voi siirtyä taistelunäkymään.

Kartalla pelaajan tulee nähdä HUD-ikkunan lisäksi seuraavat tiedot:

- Joukkojen luku
- Laivan kunto
- Kotikaupunkiin paluunappi
- Dialogit

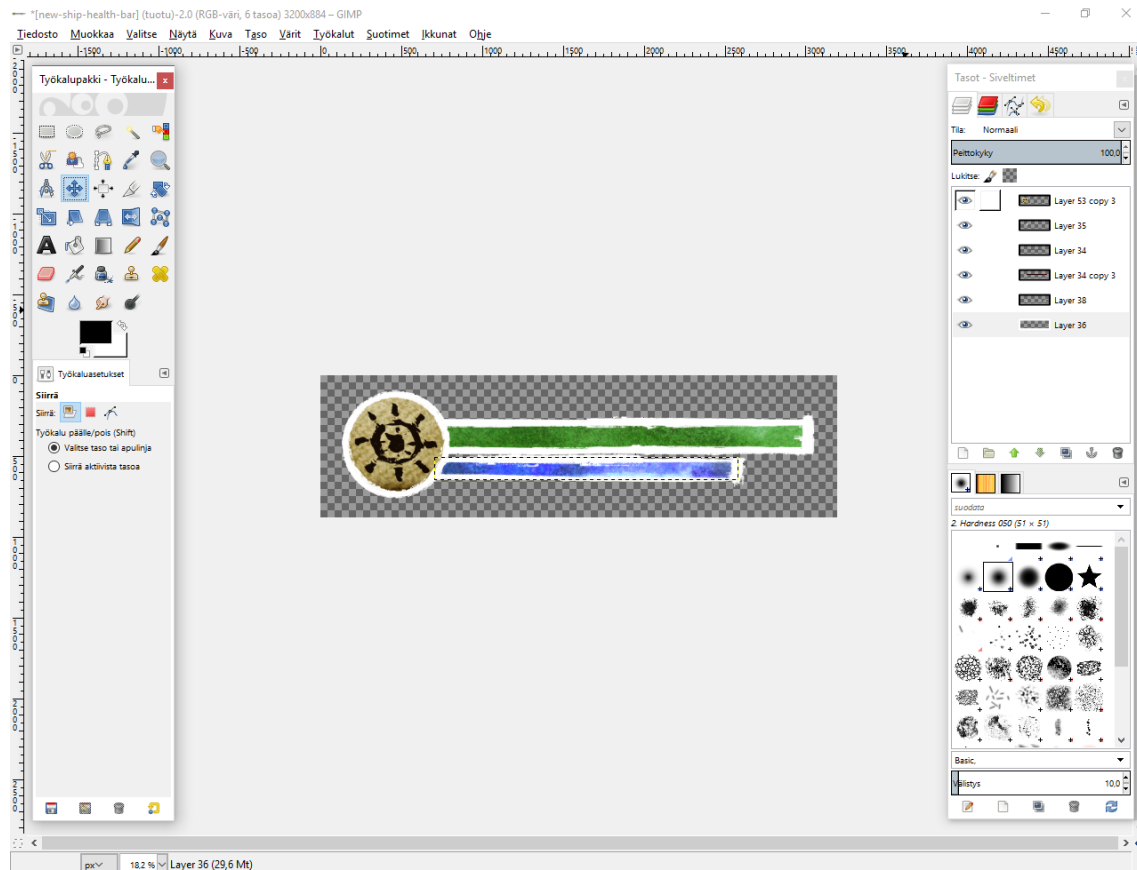
Kartta näkymään tehdään alustavasti viisi eri dialogia. Dialogit aukeavat, kun pelaaja pääsee kartalla tiettyyn tapahtumaan. Näitä ovat esimerkiksi Taisteluun siirtyminen tai majakalla vierailu. Dialogeissa pelaajan tulisi voida vaikuttaa pelin kulkuun eri tavoilla.



KUVIO 6. Dialogin konsepti

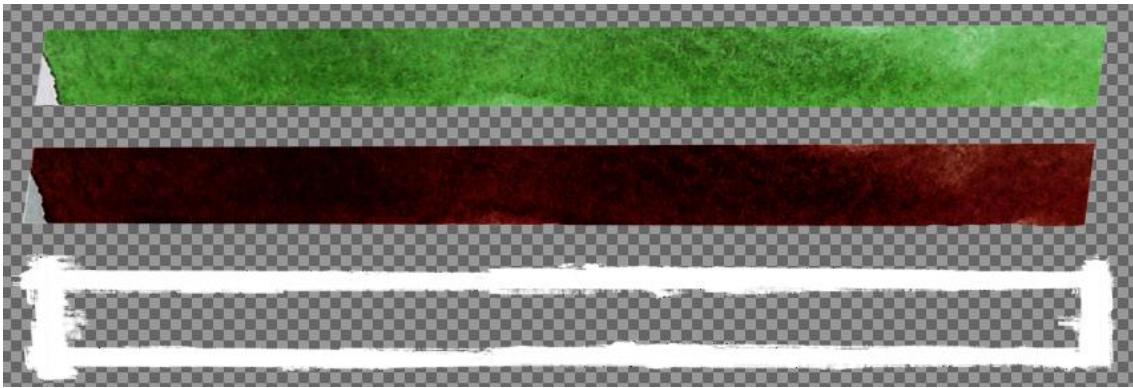
4 TOTEUTUS

Ennen varsinaista työn toteutuksen aloittamista grafiikat käsiteltiin GIMP kuvankäsittelyohjelmalla. Kuvat tulevat graafikolta PSD -tiedostoina. Graafikko on asetellut grafiikat kerroksittain siten, miten niiden tulee näkyä itse pelissä.



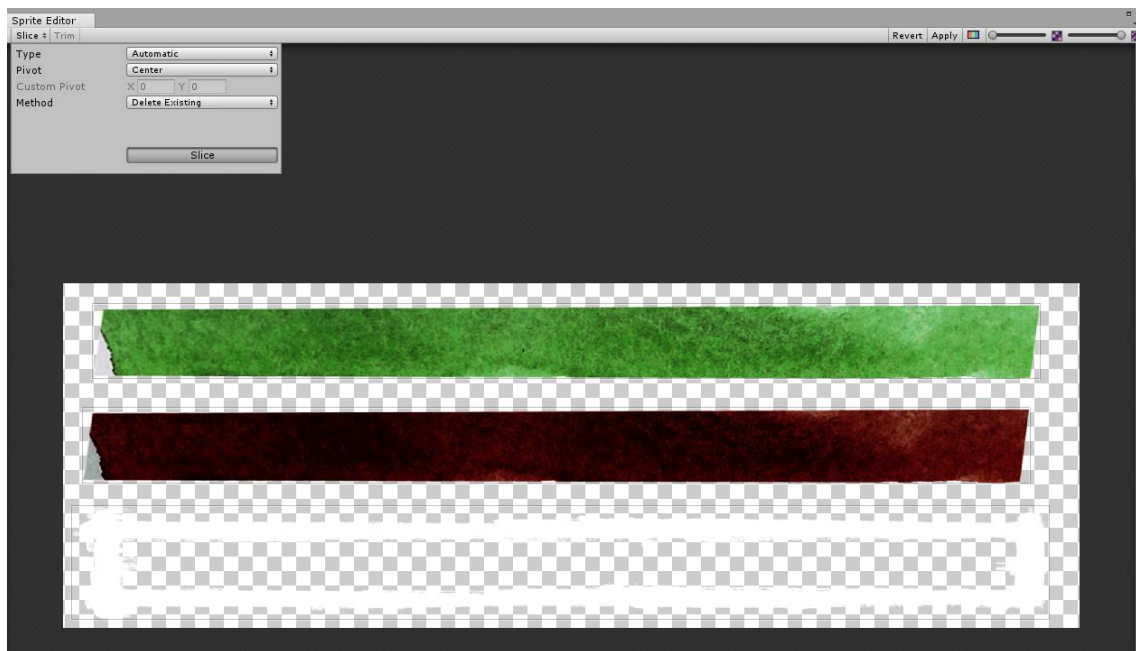
KUVIO 7. GIMP-käyttöliittymä ja energiapalkin grafiikat

Tämä asetelma jouduttiin kuitenkin purkamaan, sillä grafiikoiden tulee nimenomaan olla erillään toisistaan kun niitä viedään Unity-editoriin. Yllä oleva energiapalkki puretaan siis kerros kerrokselta ja ne asetellaan toisistaan eriteltynä seuraavasti:



KUVIO 8. Energiapalkin kerrokset eriteltyinä

Erotuksen jälkeen kuva tallennettiin PNG-tiedostomuotoon. Lopuksi kuva tuotiin Unity-editoriin missä se paloiteltiin kolmeen osaan Unityn omalla Sprite Editorilla.



KUVIO 9. Unity Sprite Editor ja energiapalkin

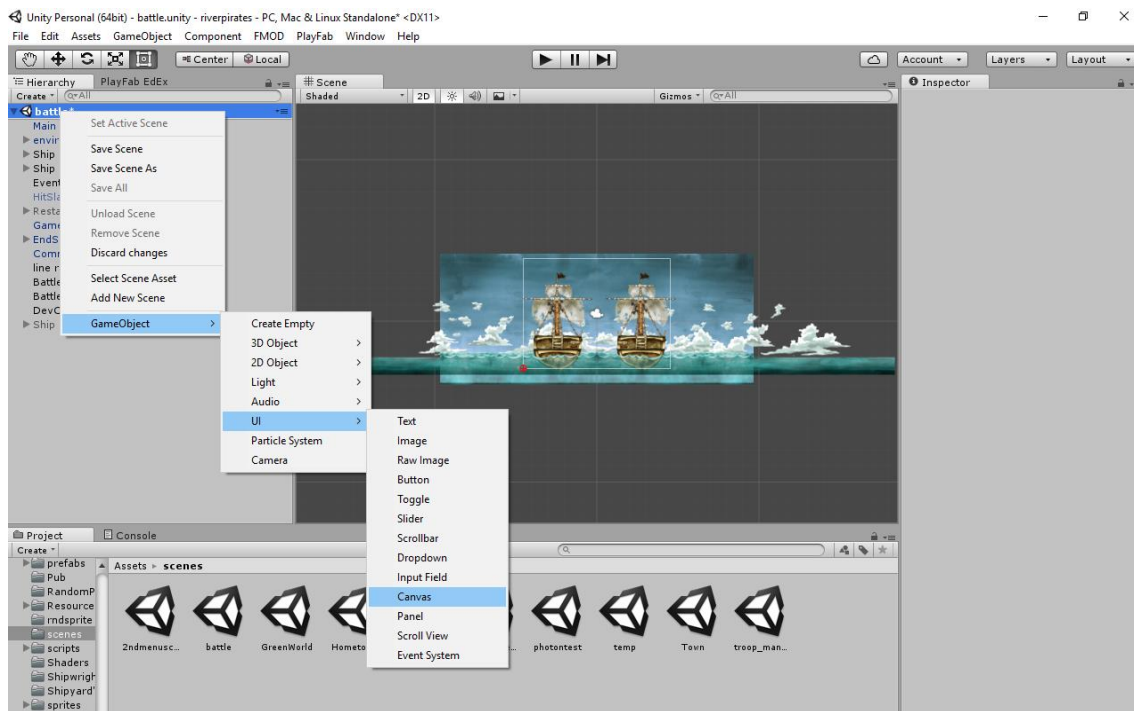
Tämä sama prosessi suoritetaan jokaisen pelin grafiikan kohdalla. Niitä ei kuitenkaan käydä enää erikseen läpi tässä raportissa.

Toteutus aloitettiin toukokuussa 2016 ja sen toteutus jatkuu edelleen. Käyttöliittymä toteutetaan suunnitelmien mukaisesti.

4.1 Taistelu

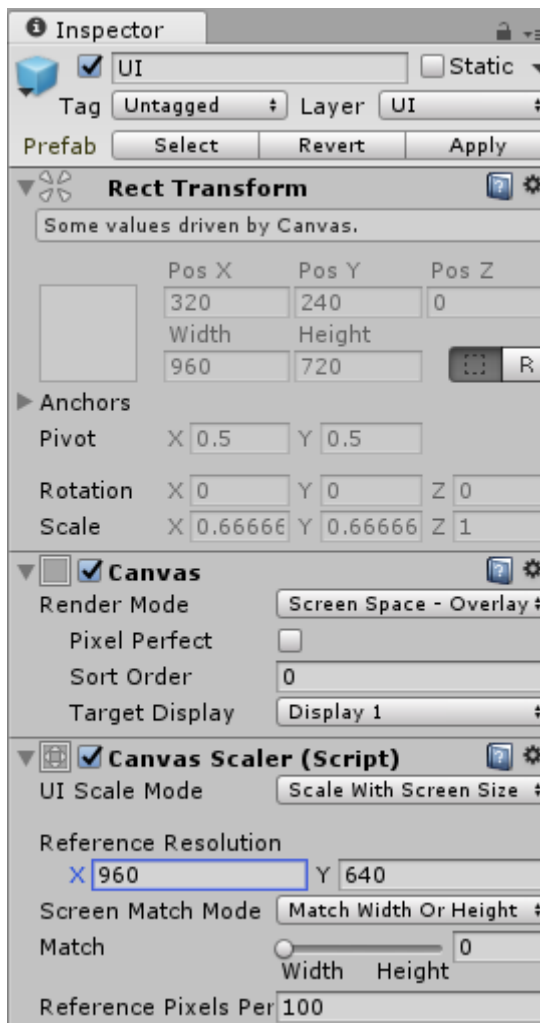
Canvas

Käyttöliittymän tekeminen Taistelukohtaukseen aloitettiin luomalla uusi Canvas.



KUVIO 10. Canvaksen luonti

Tälle Canvakselle annettiin nimeksi UI. Tämän jälkeen lähdettiin määrittelemään kyseisen Canvaksen asetuksia.



KUVIO 11. UI Canvaksen asetukset

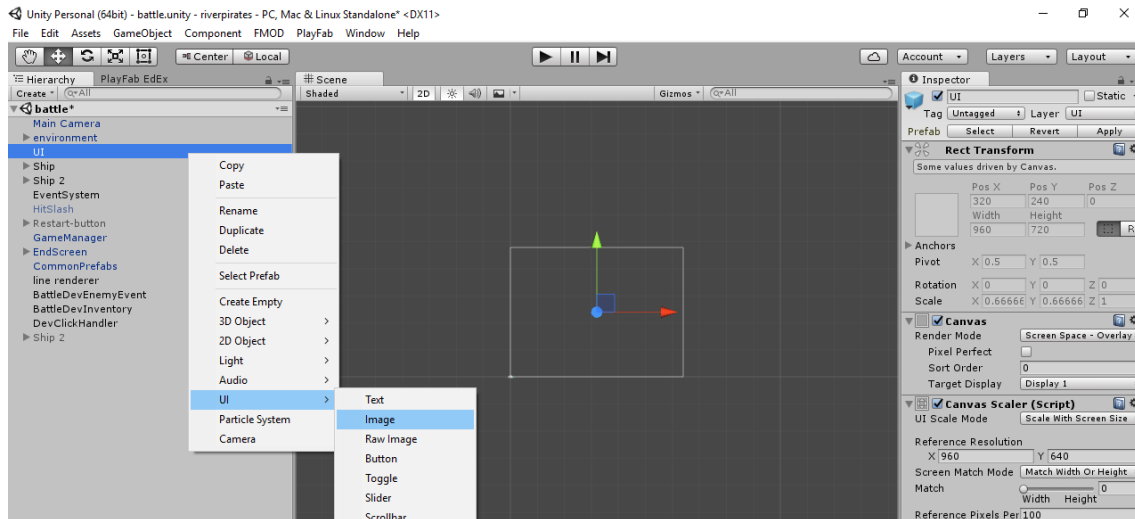
Canvaksen Render Modeksi valittiin Screen Space – Overlay. Tämä siksi, että UI-elementit pysyvät Scenen päällä. Tämä vaihtoehto mahdollistaa myös sen, että UI:n elementit skaalautuvat näytön resoluution mukaan.

UI Scale Mode -kohtaan valittiin Scale With Screen Size. Tällä skaalaustilalla määritellään referenssiresoluutio, eli se, missä resoluutiossa Unityn editorissa ollessamme näemme Canvaksen sisällä olevat elementit.

Referenssiresoluutioksi valittiin 960 x 640, koska se on hyvin yleinen resoluutio HD-puhelimen landscape-orientaatiossa.

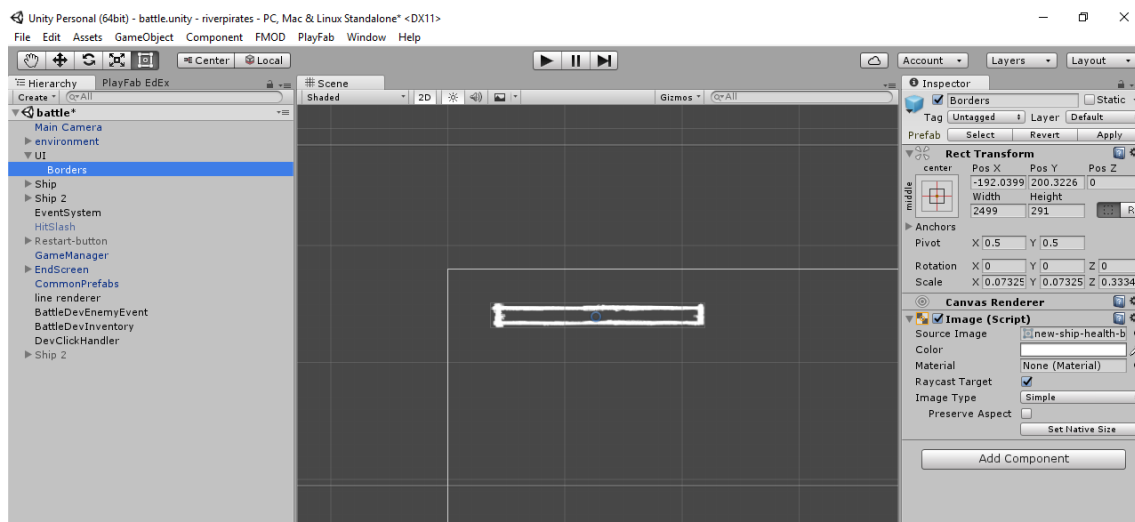
Laivan energiapalkki

Seuraavaksi lähdettiin tekemään ensimmäistä käyttöliittymässä esiintyvää elementtiä, eli laivan energiaa. Laivan energia koostuu kolmesta eri kuva elementistä: Reunat, Naamio ja Sisältö.



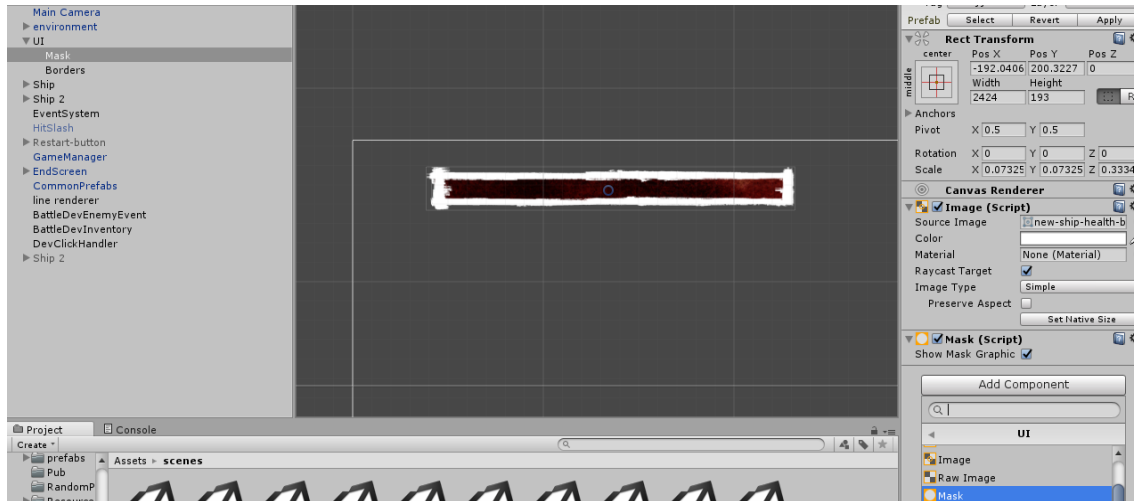
KUVIO 12. Kuvan luonti

Kun ensimmäinen kuva oli luotu, haettiin sille graafikon tekemät reunat. Kyseinen kuva sai nimekseen Borders ja se asetettiin Canvaksen vasempaan yläkulmaan suunnitelman mukaisesti.



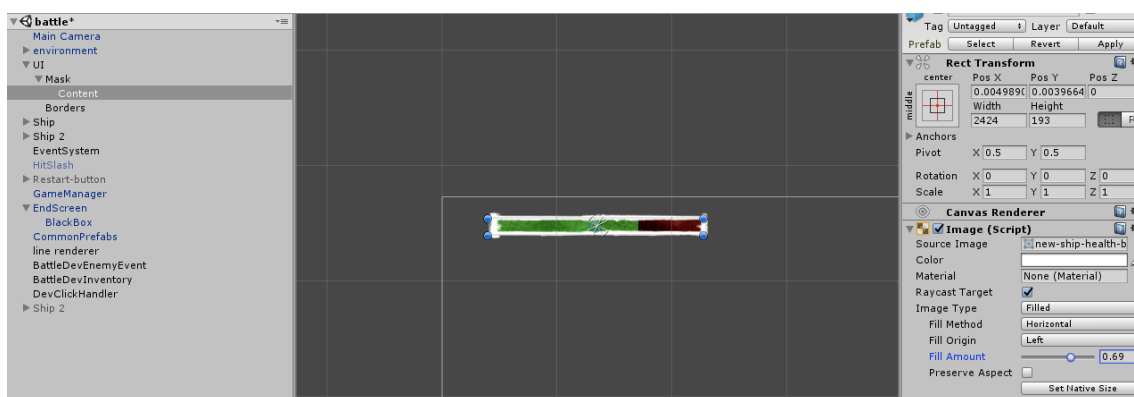
KUVIO 13. Reunat

Reunojen jälkeen luotiin toinen kuva mille annettiin nimeksi Mask. Tämä aseteltiin näennäisesti reunojen sisälle. Nimensä mukaisesti Mask-kuvaan tuli liitettyä myös UI-komponentin Mask, jonka ansiosta sen sisältö ei tulvi yli.



KUVIO 14. Naamio

Kun Mask oltiin saatu paikoilleen ja sen grafiikat oli tuotu, luotiin sille lapsiobjektiksi kuva nimeltä Content. Contentin, eli sisällön, asettelin kirjaimellisesti naamion sisään. Koska Content on itse energia, sen kuva-asetuksista määriteltiin tyypiksi Filled (täytetty). Täyttötavaksi valittiin kuvanmukaisesti Horizontal (Vaakasuora)- vaihtoehto. Energiapalkin täyttö asetettiin alkamaan luonnollisesti vasemmalta.

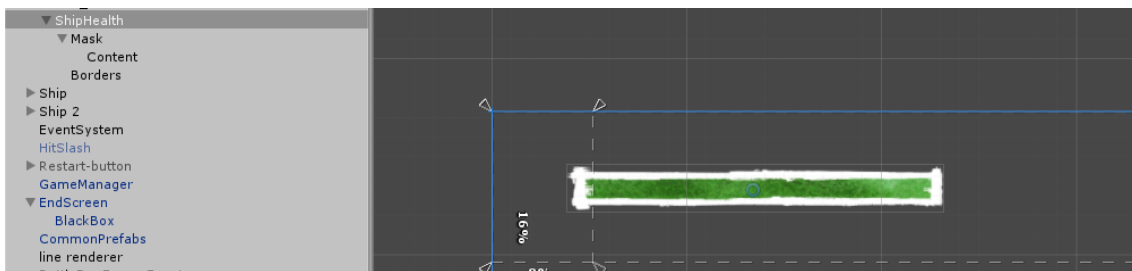


KUVIO 15. Sisältö

Fill Amount -arvoa vaihtamalla pystytään säätämään energian näennäistä määrää. Sillä voi myös heti editorissa kokeilla miltä energian vähentäminen käytännössä näyttää.

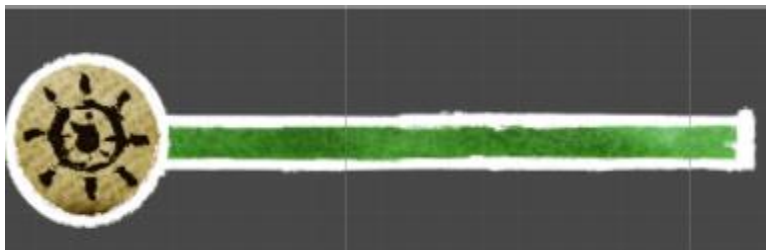
Kun tämä kolminaisuus oli viimein saatu paikoilleen, niputettiin se yhteen tyhjällä GameObjectilla. Tälle annettiin nimeksi ShipHealth. Kaikki edellä mainitut kuvat asetettiin sitten ShipHealthin lapsiojekteiksi.

Tämän jälkeen ankkuroitiin ShipHealth canvaksen vasempaan yläkulmaan, minkä ansiosta edellisten kuvien ankkurointipisteitä ei tarvinnut alkaa enää erikseen säätämään. Ankkurointi varmistaa muuan muassa sen, että resoluution muuttuessa elementit pysyvät aina oikealla kohdalla.



KUVIO 16. ShipHealthin ankkurointi

Lopuksi Energiapalkki koristeltiin näennäisesti pelaajan avatarilla, joka perusasetuksiltaan on ruori.



KUVIO 17. Valmis energia palkki

Energiapalkin valmistuttua sille kirjoitettiin vielä MonoDevelopilla skripti.

```
ShipHealth.cs
ShipHealth > Update ()
1 using UnityEngine;
2 using UnityEngine.UI;
3 using System.Collections;
4
5 public class ShipHealth : MonoBehaviour {
6     Image content;
7     private float lerpSpeed = 6;
8     private float fillAmount;
9     public float MaxValue { get; set; }
10
11     void Start () {
12         content = transform.GetChild(0).GetChild(0).GetComponent<Image>();
13     }
14
15     void Update () {
16         if (fillAmount != content.fillAmount) {
17             content.fillAmount = Mathf.Lerp(content.fillAmount, fillAmount, Time.deltaTime * lerpSpeed);
18         }
19
20         if (currentFlash > 0) {
21             currentFlash -= flashStep;
22         }
23
24         SetFlashAmount(currentFlash);
25     }
26
27     public float Value {
28         set {
29             currentFlash = 1f;
30             fillAmount = Map(value, 0, MaxValue, 0, 1);
31         }
32     }
33 }
```

KUVIO 18. MonoDevelop käyttöliittymä ja koodi

Tällä koodilla saatiin energia kulumaan visuaalisesti itse pelissä. Koodin tarkoitus oli myös tasoittaa energian katkonainen kuluminen tasaisemmaksi. Joka kerta kun energiaa menetetään, vihreä energiapalkki välähtää. Tällä tavoin onnistutaan kiinnittämään pelaajan huomio energiaan paremmin.

Kanuunan laukaisunappi

Energiapalkin jälkeen lähdettiin toteuttamaan Kanuunan laukaisunappia. Tämä nappi näkyy ainoastaan pelaajan puolella vasemmassa alakulmassa. Nappi toteutettiin pitkälti samalla periaatteella kuin edellinen energiapalkki. Eroavaisuutena oli, että napin ”energia” joka tässä tapauksessa kuvastaa latausaikaa, tehtiin seuraavasti: Täyttötavaksi valittiin Radial 360, joka siis käytännössä saa aikaan sen, että energia kuluu ympyrän muotoisena. Täyttäminen taas asetettiin alkamaan ylhäältä oikealta.



KUVIO 19. Valmis kanuunan laukaisunappi

Kun latausaika tulee täyteen, muuttuu kanuunan kuva. Sen tarkoitus on viestiä pelaajalle, että kanuuna on valmis laukaistavaksi.

4.2 Yleinen

Yleinen käyttöliittymäosuus aloitettiin luomalla uusi Canvas latausnäkömään. Tälle Canvasille annettiin nimeksi General UI. Canvasen lapsiobjektiksi luotiin tyhjä Gameobjectin jonka nimeksi tuli Resources.

Resurssit

Resources-objektin alle tehtiin jälleen tyhjän Gameobjectin, joka nimettiin Goldiksi. Gold-objektin lapseksi kopioitiin aiemmin Taistelukäyttöliittymään tehty energiapalkki ja sen kaikki elementit aina Ship Health -skriptiä myöten. Tämän jälkeen skaalattiin energiapalkkia pienemmäksi. Lopuksi tekstiobjekti asetettiin valkoisella värillä keskelle energiapalkin sisältöä. Tämä objekti kopioitiin, väri vaihdettiin mustaksi ja se asetettiin ensimmäisen tekstiobjektin taakse. Lopuksi sitä vedettiin hiukan vaaka- ja pystysuorassa pois ensimmäisestä tekstiobjektista. Näin saatiin tälle tekstiobjektille näennäinen varjostus. Tekstiobjekteihin käytettiin HVD Poster Clean -fontteja.

Kun tämä kaikki oli tehty, otettiin energiapalkista kaksi kopiota ja molemmille niistä vaihdettiin niimet sen mukaan mitä resursseja ne kuvastavat.



KUVIO 20. Resurssipalkkien placeholder

Resursseihin käytettiin samaa skriptiä kuin aiemmin tehtyyn energiapalkkiin.

Pelaajan tiedot

Resurssien jälkeen siirryttiin tekemään pelaajan tietoja vasempaan ylänurkkaan. Tähän kopioitiin aiemmin Taistelukäyttöliittymään tehty kanuunanlaukaisunappi. Aluksi siitä poistettiin sen napipiominaisuus. Tämän jälkeen vaihdettiin grafiikat ja sen ympyrän keskelle asetettiin Tier-numeron tekstiobjekti. Lopuksi lisättiin pelaajan Avatarin viereen tekstirivi, jonka tarkoitus on näyttää pelaajan nimi. Tekstiobjekteihin käytettiin HVD Poster Clean -fontteja.



KUVIO 21. Pelaajan tiedot placeholder

4.3 Kartta

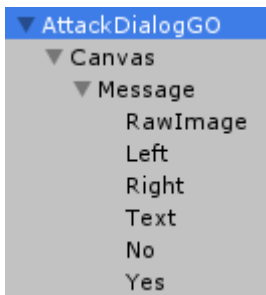
Karttanäkymässä käytetään myös HUD-ikkunaa. Kun pelaaja siirtyy karttanäkymään, siihen päivitetään uusia toimintoja. Siihen lisättävät toiminnot näkyvät ainoastaan tässä scenessä. Karttanäkymä aloitettiin lisäämällä kotikaupunkiinpaluu-nappi HUD-ikkunaan. Tämä tehtiin luomalla uusi Button-objekti General UI-canvakseen. Tämän jälkeen sille vaihdettiin grafiikat.



KUVIO 22. Kotikaupunkiin paluunappi

Dialogit

Dialogipohjan tekeminen aloitettiin luomalla tyhjä GameObject, jonka sisään tehtiin Canvas. Canvasin sisään tehtiin uusi GameObject, jolle annettiin nimeksi Message. Messageen sisään tehtiin kuva, jolle vaihdettiin grafiikaksi kirjakäärön tausta. Seuraavaksi luotiin kuva jonka grafiikaksi vaihdettiin kirjakäärön vasen reuna. Vasen reuna siirrettiin luonnollisesti vasemmalle puolelle. Tämän jälkeen vasen reuna kopioitiin ja vietiin käärön oikealle puolelle. Oikealle puolelle vaihdettiin oikean puoleisen käärön reunan grafiikat.



KUVIO 23. Dialogin rakenne

Kun dialogipohja oli valmis, siihen lisättiin tekstiobjekti. Tekstiobjektin kirjoitusta muokataan aina sen mukaan, mistä tapahtumasta on kysymys. Tekstiobjekteihin käytettiin Gooddog New -fontteja.

Tämän jälkeen dialogipohja kopioitiin viisi kertaa ja niihin laitettiin teksti kunkin tapahtuman mukaisesti. Esimerkiksi, kun pelaaja kohtaa vihollislaivan, aukeaa ruudulle seuraava Dialogi.



KUVIO 24. Taisteluun siirtymä -dialogi

5 TULOKSET JA NIIDEN TARKASTELU

5.1 Taistelunäkymän tulokset

Taistelunäkymän käyttöliittymän toistaiseksi lopullinen versio valmistui lokakuussa 2016. Sen toteutus onnistui suunnitelmien mukaan ilman suurempia ongelmia. Käyttöliittymä tukee kaikkia mahdollisia resoluutioita ja skaalautuu aina laitteen resoluution sekä kuvasuhteen mukaisesti.



KUVIO 25. Taistelu

Käyttöliittymä näyttää ja tuntuu loogiselta. Se sopii hyvin itse kohtaukseen. Sitä on helppo käyttää ja ymmärtää. Energiapalkit välähtelevät ottaessa osumaa, samoin myös hahmot. Kanuunan latauksessa valkoinen ympyrä sulkeutuu laukaisu napin ympärille. Pelaajan laivan energiapalkki tullaan tulevaisuudessa korvaamaan HUD-ikkunan energiapalkilla.

Tällä hetkellä vihollisen laivan energiapalkki ei erotu vielä mitenkään pelaajan laivan energiapalkista, joten sen ulkoasua tullaan todennäköisesti muuttamaan tulevaisuudessa. Tämä siksi, että pelaajan olisi helpompaa erottaa oma puolensa vihollislaivan joukoista.

5.2 Yleisen-näkymän tulokset

Yleisen näkymän väliaikainen versio valmistui marraskuussa 2016. Tämä on väliaikainen siksi, että vielä ei ole päätetty mitä kaikkea lopullisessa versiossa tulee näkymään. Energiapalkki ja resurssi Placeholder korvattiin lopuksi niille tarkoitetuilla grafiikoilla.



KUVIO 26. HUD

HUD on yksinkertainen, selkeä ja helposti lähestyttävä. Se antaa pelaajalle sillä hetkellä tarpeellisen tiedon, eikä mitään ylimääräistä. Tämän hetkinen HUD sisältää pelaajan nimen, tason ja resurssit. Siihen on myös suunnitelmista poiketen liitetty laivan energiapalkki.

Tulevaisuudessa ikkuna tulee heijastamaan pelaajalle tietoja eri kohtauksien mukaan. Tässä vaiheessa tarpeelliset tiedot ovat ainoastaan pelaajan tiedot ja valuutta.

Pelaajan tiedoista löytyy pelaajan nimi -tekstiobjekti, joka muuttuu sen mukaan minkä nimen pelaaja itselleen kirjoittaa. Numero tiedoissa kuvastaa pelaajan tasoa. Se muuttuu suuremmaksi sitä mukaan kun pelaaja etenee pelissä. Energiapalkki toimii käytännössä täysin samalla tavalla kuin taistelunäkymässä.

Kultakala oikeassa yläkulmassa kuvastaa pelissä saatavaa rahaa jolla voidaan ostaa lisää joukkoja ja parannuksia. Kultakalan vieressä oleva helmi taas kuvastaa valuuttaa jolla voidaan ostaa muun muassa kaikkia edellä mainittuja ja niiden lisäksi myös kosmeettisia tavaroita. Käytännössä kaikkea sellaista, mitä ei tavallisella pelirahalla voi saada. Resurssien sisältöpalkit täyttyvät pelissä resurssin määrän mukaan.

5.3 Karttanäkymän tulokset

Karttanäkymän väliaikainen versio valmistui marraskuussa 2016. Kuten Yleiseen-näkymään, myös Karttanäkymään tullaan vielä lisäämään elementtejä ja toimintoja.



KUVIO 27. Kartta

Suunnitelmista poiketen, laivan kunto päätettiin näyttää HUD-ikkunassa. Karttanäkymään ei myöskään asetettu joukkojen lukumäärää, sillä sen ei katsottu olevan vielä ajankohtaista. Jos ja kun tämä ominaisuus tullaan lisäämään, se lisätään todennäköisesti myös HUD-ikkunaan.

Karttanäkymä jäi toistaiseksi todella tyhjäksi. HUD-ikkunan lisäksi, siellä on ainoastaan kotikaupunkiin paluunappi vasemmassa alakulmassa. Kun pelaaja painaa kotikaupunkiin paluunappia, hän keskeyttää seikkailunsa kartalla. Tämän jälkeen hän ei voi enää palata samalla kartalle takaisin. Tästä syystä nappia voisi kutsua tästä eteenpäin enemminkin luovuta tästä -napiksi.

6 POHDINTA

Opinnäytetyön aiheena oli suunnitella ja toteuttaa käyttöliittymä pelille Riverian Pirates. Työtä teutettiin Game Lab -opintojen aikana. Olin itse työn toimeksiantaja. Työn tavoite oli toteuttaa mahdollisimman toimiva käyttöliittymä. Käyttöliittymän tuli olla myös visuaalisesti näyttävä. Tämän ohella oli myös tarkoitus perehtyä hyvän käyttöliittymän ominaisuuksiin ja sen suunnitteluun.

Käyttöliittymän suunnitteluun perehtyminen oli itselleni erityisen opettava ja mielenkiintoinen osuus. Sain huomattavasti uutta tietoa itse käyttöliittymästä niin käsitteenä kuin konkreettisena osana peliä. Koen, että tämän työn ansiosta seuraavan peliprojektin suunnittelu ja toteutus tulee onnistumaan entistä paremmin.

Suunnitteluvaihe käytiin mielestäni läpi onnistuneesti. Se oli itselleni yksi työn mielenkiintoisimmista vaiheista, sillä siinä pääsi käyttämään omaa luovuutta. Suunnitelmiin kuitenkin tuli muutamia muutoksia toteutuksen aikana, mutta pääosin onnistuttiin pysymään alkuperäisessä suunnitelmassa. Suunnittelutavoista sekä suunnittelumenetelmistä olisi aina voinut kirjoittaa vieläkin laajemmin.

Myös toteutus onnistui mielestäni hyvin. Oli erittäin helppoa ja miellyttävää lähteä rakentamaan käyttöliittymä, kun oli valmiit suuntaviivat mitä seurata. Raportin kirjoittaminen toteutusosuudesta sen sijaan oli jokseenkin haastavaa. Koin toteutuksesta kirjoittamisen paikoitellen työtä hidastavana tekijänä. Tämä on merkki siitä, että työstä raportoimisen tyyliä olisi voinut miettiä hieman paremmin. Kaikkea ei olisi ehkä tarvinnut selittää niin yksityiskohtaisesti.

Kaiken kaikkiaan olen tyytyväinen työhön. Aiheen laajuus oli minulle juuri sopiva ja tavoitteisiin päästiin. Riverian Pirates mobiilipeli, eikä sen käyttöliittymä ole kuitenkaan vielä valmis. Tulen todennäköisesti jatkamaan vielä pitkään kyseisen pelin parissa Game Lab -opintojen päätyttyäkin.

LÄHTEET

Angry Birds Rio, 2016, kuvankaappaus pelistä

Derk 2012. THE 8-STEP GUIDE TO INTERFACE DESIGN FOR IPHONE GAMES. Viitattu 1.10.2016 <http://www.paladinstudios.com/2012/04/23/the-8-step-guide-to-interface-design-for-iphone-games/>

GIMP 2016 2016. Viitattu 30.9.2016, <https://www.gimp.org>

MonoDevelop 2016. Viitattu 30.9.2016, <http://www.monodevelop.com>

Quintans, D. 2013. Game UI By Example: A Crash Course in the Good and the Bad. Viitattu 30.9.2016, <https://gamedevelopment.tutsplus.com/tutorials/game-ui-by-example-a-crash-course-in-the-good-and-the-bad--gamedev-3943>

Tran, Valkola. Flowchart Chart, 2016, Viitattu 23.9.2016

Tutorialspoints 2016. Software User Interface Design. Viitattu 30.9.2016 https://www.tutorialspoint.com/software_engineering/software_user_interface_design.htm

Unity 3D 2016. Viitattu 30.9.2016, <https://www.unity3d.com>