Laivanupotus loppuraportti

Tuomas Luukinen

Jaakko Mäkinen

Harjoitustyö

Huhtikuu 2017

Tieto- ja viestintätekniikka

Tekniikan ja liikenteen ala

Sisältö:

[1 Esittely 2](#_Toc480956303)

[2 Lopullinen rakennekaavio 2](#_Toc480956304)

[3 Toteutetut luokat 3](#_Toc480956305)

[4 Työaikaraportti 4](#_Toc480956306)

[5 Lopullisen version näyttökaappaukset 5](#_Toc480956307)

[6 Keskeisimmät ongelmat ja niiden ratkaisut 5](#_Toc480956308)

[7 Testaus 6](#_Toc480956309)

[8 Itsearviointi 6](#_Toc480956310)

[9 Kommentit kurssista 6](#_Toc480956311)

# Esittely

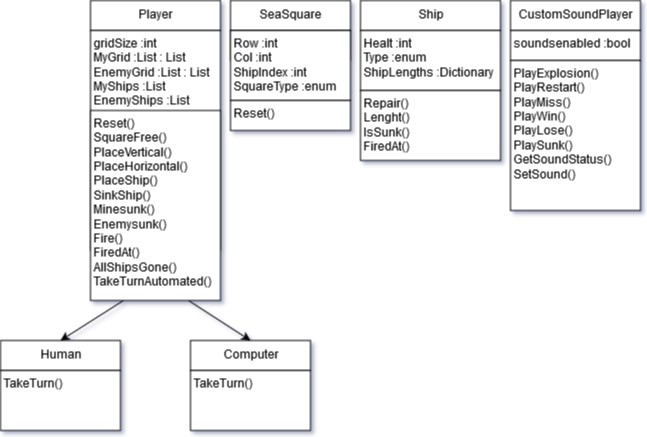
Tämän sovelluksen tekijät ovat Tuomas Luukinen ja Jaakko Mäkinen. Sovellus tulee olemaan JAMK:in olio-ohjelmointi ja käyttöliittymien ohjelmointi -kurssien harjoitustyö. Harjoitustyö löytyy GitHubista osoitteesta: <https://github.com/TuomasL96/TTOS0300Laivanupotus>

Sovellus on klassinen tietokonetta vastaan pelattava laivanupotuspeli. Ohjelmointi tullaan tekemään C# -ohjelmointikielellä, ja käyttöliittymä toteutetaan XAML:illa. Ohjelmointiympäristönä toimii Microsoft Visual Studio 2015.

Peli alkaa sillä, että pelaajien laivat sijoitetaan satunnaisesti ruudukkoon. Käyttäjä ampuu tietokonepelaajan ruudukkoon painamalla sen neliöitä hiirellä. Tämän jälkeen tietokonepelaaja ampuu satunnaisesti pelaajan ruudukkoon. Pelaaja ja tietokone ampuvat vuorotellen, kunnes toinen on saanut kaikki vastustajan laivat upotettua.

# Lopullinen rakennekaavio

Ainoat muutokset rakennekaavioon ovat nimien vaihdoksien ja methodien sekä muuttujien määrän kasvamisen lisäksi uusi ”Mysound” -luokka, jonka tehtävä pelin aikana kuuluvien äänien soittaminen. Muutokset suunnitelmaan johtuvat siitä, että kaikkia tarvittavia muuttujia ja metodeja ei tiedetty suunnitteluvaiheessa. CustomSoundPlayer- luokka lisättiin, koska tulimme johtopäätökseen että peli tarvitsee äänet selkeyttämään näytöllä tapahtuvaa toimintaa.



Kuva 1 Luokkakaavio

# Toteutetut luokat

Työn tärkein luokka Player sisältää suurimman osan pelin logiikasta, kuten listan pelaajan sekä vihollisen ”merialueesta” ja laivoista, laivojen sijoittelun ruudukkoon sekä niiden ampumisen logiikat. Playeristä periytyvät luokat Human ja Computer sisältävät vain pelivuoron toteutumiseen liittyvän metodin.

”SeaSquare” -luokka pitää sisällään tiedon neliön tilasta enumina (Squaretype), neliön sijainnin arvot Row ja Col, sekä Ruudun resetoimisen toimittavan methodin. Tämän luokan olioden avulla luodaan peliruudukot pelaajille. Ship -luokka taas on tehty pitämään sisällään tietoa laivojen tyypeistä, pituuksista sekä kunnosta, tämän luokan toimintoja ovat mm. laivan uppoamisen tarkistus ja korjaus.

CustomSoundPlayer luokka on tehty helpottamaan äänien soittamista pelin aikana. Se on tehty NAudio -audiokirjastoa käyttämällä, jotta peli kykenee soittamaan montaa ääntä yhtä aikaa. Luokka antaa myös mahdollisuuden poistaa kaikki pelinsisäiset äänet käytöstä.

Näiden ”Model” luokkien lisäksi on olemassa Viewin luokka TileConverter, joka vaihtaa käyttöliittymässä näytettävän ruudukon neliöt oikean värisiksi niihin liitettyjen SeaSquareiden mukaan, sekä ViewModelin GridVMBase, joka on perusta ComputerGrid ja HumanGrid luokille. Näiden luokkien tehtävänä on hoitaa käyttöliittymän mallinnus sekä pelivuorojen antaminen ja pelin lopputilanteen tunnistus.

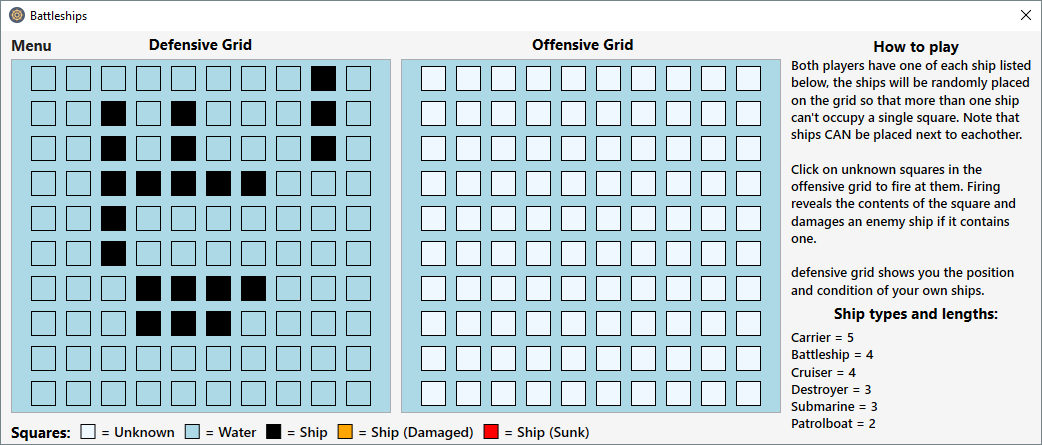
# Työaikaraportti



Kuva 2 Työaikaraportti

Yliviivatut tehtävät eivät päätyneet lopulliseen työhön. Toteutetut tunnit ovat melko epätarkkoja arvioita.

# Lopullisen version näyttökaappaukset



Kuva 3 Pelin käyttöliittymä

Pelissä ei ole päänäytön lisäksi muita näyttöjä, mutta kaikki oleellinen tieto on silti saatu hyvin mahtumaan mukaan. Päänäytön kokoa ei voi muuttaa, koska se ei tahtonut skaalautua hyvän näköisesti. Menu -näppäin avaa valikon jota painamalla pelaaja pystyy poistamaan äänet päältä, nollaamaan pelin, pelaamaan automoidun pelin tai sammuttamaan ohjelman. Näytön Offensive ja Defensive Gridien neliöt vaihtavat väriä niihin ammuttaessa.

# Keskeisimmät ongelmat ja niiden ratkaisut

Laivojen sijoittelu ruudukkoon osoittautui haastavaksi, monen yrityksen jälkeen kuitenkin päätimme muokata pelin sääntöjä niin, että laivat saavat olla toisissaan kiinni. Tämä ongelma olisi ollut ensimmäisenä korjattavien listalla, mutta ikävä kyllä aikaa ei sille jäänyt.

Toinen päänvaiva projektissa oli äänien toistaminen. Windowsin vakio äänentoisto PlaySoundin kautta ei anna toistaa monta ääntä samanaikaisesti, joten jouduimme opetelemaan käyttämään kolmannen osapuolen audiokirjastoa: Naudiota.

# **Testaus**

Pelin testaus on suoritettu pääosin pelaamalla ja kaikkia näppäimiä nopeasti painelemalla. Peliin on jäänyt testausta varten tehty toiminto, jossa voit antaa tietokoneen pelata yhden pelikierroksen nopeasti läpi. Äänien päällekkäisyyttä testattu aloittamalla uusi peli heti voiton jälkeen sekä testattu toimivatko pelin äänet, jos voitto- tai häviömusiikki on yhä soimassa taustalla.

Testattu myöskin uudelleen käynnistämällä pelin useaan kertaan ja tarkistamalla että kaikki ruudut missä niiden kuuluukin ja ei päällekkäin.

# Itsearviointi

Tuomas Luukinen: 4 - Harjoitustyössä käytettiin kurssilla opittuja, sekä itseopittuja asioita monipuolisesti ja sen yleinen toimivuus on hyvä. Koodi pääosin selkeää, mutta varsinkin xamlin puolella paljon ”toistoa” ja nopeita ratkaisuja helpostiluettavan koodin sijaan. MVVM – mallin käyttöä pyrittiin noudattamaan, mutta koska se on meille uusi asia, jäi sen toteutuksen onnistuminen vähän epäselväksi.

Mäkinen Jaakko: 3 - Harjoitustyö toimii hyvin, mutta osalleni jäi pienempi työmäärä.

# Kommentit kurssista

Kurssi oli mielenkiintoinen. Oli mukavaa saada työskennellä omaan tahtiin työn parissa.