Oliogotchi

Raportti kurssin TTOS0300 harjoitustyöstä

Joona Hautamäki

Kioto Hiirola

Minttu Mäkäläinen

Loppuraportti, v1.0

Huhtikuu 2017

Tieto- ja viestintätekniikan koulutusohjelma

Tekniikan ja liikenteen ala

Sisältö

[1 Johdanto 2](#_Toc480916344)

[2 Yleiskuvaus 3](#_Toc480916345)

[3 Pelin näkymät 4](#_Toc480916346)

[3.1 Päävalikko 4](#_Toc480916347)

[3.2 Pelinäkymä 5](#_Toc480916348)

[3.3 Minipelinäkymä 6](#_Toc480916349)

[3.3.1 Kivi-sakset-paperi 7](#_Toc480916350)

[3.3.2 Pallon nappaus 7](#_Toc480916351)

[3.4 Ohjeet 8](#_Toc480916352)

[3.5 Game Over 9](#_Toc480916353)

[4 Pelin toiminnallisuus 11](#_Toc480916354)

[4.1 Oliogotchin eläminen 11](#_Toc480916355)

[4.2 Minipelit 11](#_Toc480916356)

[4.3 Pelin animaatio 12](#_Toc480916357)

[4.4 Tallennus ja lataus 13](#_Toc480916358)

[5 Luokkien käyttö ja UML-kaavio 14](#_Toc480916359)

[6 Testaaminen 16](#_Toc480916360)

[7 Kehitettävää 17](#_Toc480916361)

[8 Yhteenveto 18](#_Toc480916362)

[8.1 Käytetty työaika 18](#_Toc480916363)

[8.2 Haasteista onnistumisiin 18](#_Toc480916364)

[8.3 Kommentit kurssista 20](#_Toc480916365)

# Johdanto

Kyseessä on Käyttöliittymien ohjelmoinnin (TTOS0300) harjoitustyö. Ryhmään kuului Joona Hautamäki (K1647), Kioto Hiirola (K8252) sekä Minttu Mäkäläinen (K8517). Joona toimi pääkoodarina, Kioto kirjoitti koodia ja teki kaikken pelin kuvituksen sekä Minttu toimi koodarina sekä projektipäällikkönä. Ryhmä valitsi aiheeksi Tamagotchimaisen lemmikkipelin, koska ryhmäläiset halusivat tehdä jonkinlaisen pelin aikaiseksi.

Työnimeksi peliprojektille annettiin Oliogotchi, joka on risteytys olio-ohjelmoinnista sekä Tamagotchista.

Projektin GitHub-kansio löytyy osoitteesta:

https://github.com/Gilyan/Olio-harjoitustyo

# Yleiskuvaus

Oliogotchi toteutettiin Visual Studio 2015:lla. Ohjelma rakennettiin Windows Presentation Foundation –sovelluspohjalle (WPF), joten käyttöliittymä tehtiin XAML-kielellä ja toiminnot C#-kielellä. EER-kaaviot piirrettiin Violet UML 2.1.0 –editorilla.

Oliogotchin pääkäyttäjiä olisivat kaikki nuoret ja lapsenmieliset, joita kiinnostaa hoivata virtuaalista lemmikkiä.

Oliogotchia voi pestä, silittää, ruokkia porkkanoilla tai lihalla sekä sen kanssa voi pelata kahta minipeliä. Pelaajan valintojen mukaan Oliogotchin eri attribuutit muuttuvat (onnellisuus, nälkäisyys, likaisuus). Jos Oliogotchista ei pidä tarpeeksi hyvin huolta, se kuolee. Mikäli Oliogotchi ei ole kuollut, sen tila voidaan tallentaa ja jatkaa peliä myöhemmin samasta kohdasta. Jos Oliogotchista pidetään tarpeeksi hyvää huolta, sen voi lopulta kehittää aikuiseksi. Aikuisen Oliogotchin lopullinen ulkonäkö riippuu siitä, mitä sille on syötetty.

# Pelin näkymät

Tässä kappaleessa esitellään jokainen pelinäkymä ja niiden toiminnot.

## Päävalikko

Kun Oliogotchi-sovelluksen avaa, pelaaja pääsee pelin päävalikkoon (ks. kuvio 1). Päävalikosta voi aloittaa uuden pelin (Start) tai jatkaa vanhaa peliä (Load) sekä poistua pelistä (Quit). Päävalikosta pääsee myös tarkastelemaan pelin ohjeista (Instructions).



Kuvio 1: Oliogotchin päävalikko

Jos peliä koitetaan jatkaa (Load) ja aiemmin tallennettua peliä ei löydy, aloitetaan uusi peli automaattisesti aloitusarvoilla. Jos tallennettu peli löytyy, jatketaan vanhaa peliä siitä mihin jäätiin.

Uuden pelin aloitus (Start) puolestaan poistaa mahdollisesti aiemmin tallennetun pelin ja aloittaa uuden pelin alkuarvoilla.

## Pelinäkymä

Pelinäkymä (ks. kuvio 2) on Oliogotchin päänäkymä, jossa suurin osa tapahtumista tehdään.



Kuvio 2: Pelin päänäkymä

Pelinäkymän ylälaidassa olevat mittarit kertovat Oliogotchin eri tiloista, keltainen on onnellisuusmittari, punainen on nälkäisyysmittari ja vihreä on siisteysmittari. Maksimiarvo jokaiselle mittarille on 100, arvot laskevat Living-metodin mukaisesti kahden sekunnin välein. Kun peli aloitetaan alkuarvoilla, jokainen mittari on 50:ssa.

Pese-painikkeesta Oliogotchia voi putsata, eli siisteysmittarin arvo nousee aina 1 per klikkaus.

Silitä-painikkeesta Oliogotchia voi silittää, eli onnellisuusmittarin arvo nousee 1 per klikkaus.

Ruoki kasviksia ja ruoki lihaa –painikkeista Oliogotchia voi ruokkia, tällöin nälkäisyysmittarin arvo nousee 1 per klikkaus.

Pelaa minipelejä –painikkeesta pelaaja pääsee pelaamaan kahta eri minipeliä, kivi-sakset-paperi tai pallon kiinniotto. Oliogotchin Living-metodin aiheuttama arvojen lasku pysäytetään minipelien pelaamisen ajaksi, joten minipelejä voi pelata rauhassa.

Main Menu –painikkeesta päästään takaisin päävalikkoon ja Instructions vie ohjeisiin.

## Minipelinäkymä

Minipelinäkymästä (ks. kuvio 3) pelaaja voi valita kahdesta eri pelistä, joita voi pelata Oliogotchin kanssa. Kun pelin kuviota klikataan, alle tulevat pelin ohjeet.

Saksikuva = kivi-sakset-paperi

Pallokuva = pallonheitto



Kuvio 3: Minipelinäkymä

Let’s Play –painikkeesta valittu peli lähtee käyntiin.

Back –painikkeesta palataan takaisin pelinäkymään.

### Kivi-sakset-paperi

Perinteinen kivi-sakset-paperi –peli (ks. kuvio 4). Kivi voittaa sakset, paperi voittaa kiven ja sakset voittaa paperin.

Jos Oliogotchi voittaa, sen onnellisuus kasvaa 20 pistettä. Jos Oliogotchi häviää, onnellisuus laskee 5 pistettä.

Pistetilanne on koko ajan näkyvissä ruudulla.



Kuvio 4: Kivi-sakset-paperi -peli

### Pallon nappaus

Pallon nappauspelissä (ks. kuvio 5) pelaaja pääsee liikuttamaan Oliogotchia, joka koittaa saada ylhäältä putoavaa palloa kiinni.

Kun Oliogotchi saa pallon kiinni, sen onnellisuusarvo kasvaa 20 pistettä. Jos Oliogotchi ei saa palloa kiinni, pallo putoaa uudelleen satunnaisesta paikasta ja Oliogotchin iloisuus laskee 5 pistettä ja nälkäisyys 2 pistettä.

Pistetilanne on koko ajan näkyvillä ruudulla.



Kuvio 5: Pallon nappaus

## Ohjeet

Ohjesivulta (ks. kuvio 6) pääsee lukemaan pelin tavoitteesta. Täältä löytyy myös tarkemmat kuvaukset pelinäkymässä oleville napeille.



Kuvio 6: Ohjenäkymä

## Game Over

Jos Oliogotchin jokin arvo (onnellisuus, nälkäisyys tai siisteys) laskee nollaan, Oliogotchi kuolee (ks. kuvio 7). Tällöin on jatkettava joko viimeisestä tallennuspisteestä tai aloitettava kokonaan uusi peli.



Kuvio 7: Game Over

# Pelin toiminnallisuus

## Oliogotchin eläminen

Oliogotchin elämismetodeja ovat GetSad(), GetDirty(), GetHungry() ja näiden lisäksi Age-ominaisuutta kasvatetaan Oliogotchin vanhentuessa. Elämismetodeja käytetään DispatchTimerilla, jonka tickien välinen aika on 2 sekunti. Jokaisella tickillä metodit pienentävät Creature-luokassa olevia Oliogotchin ominaisuuksia yhdellä. Näin Oliogotchin nälkäisyys, iloisuus ja puhtaus laskevat jatkuvasti. Age puolestaan suurenee aina yhdellä. Alkuarvoina Oliogotchin puhtaus, nälkäisyys ja iloisuus ovat kaikki 50, ikä alkaa aina nollasta.

Tausta elää samalla tavalla kuin Oliogotchi, sen ominaisuuksina ovat Trash ja Cleanliness. Taustan elämismetodeja ovat GetMessy() ja AddTrash(). GetMessy()-ominaisuus toimii DispatchTimerin mukaan, eli puhtaus laskee joka sekunti yhdellä. Roskia aletaan lisäämään (AddTrash()) vasta kun taustan puhtaus tippuu liian alhaiseksi. Roskien lisäämistä ei toteutettu visuaalisesti, koodissa se on olemassa ja toimii hyvin. Jos taustan puhtausarvo putoaa liian alhaiseksi (alle 10) ja roskien määrä on liian suuri, Oliogotchin iloisuus- ja puhtausarvot laskevat nopeammin. Alkuarvoina siisteys on 100 ja roskien määrä on 0.

Oliogotchi elää niin kauan, kunnes jokin sen ominaisuuksista putoaa nollaan. Sen jälkeen Oliogotchi kuolee ja peli täytyy aloittaa alusta.

## Minipelit

Kivi-sakset-paperi –pelissä pelataan Oliogotchia vastaan. Peliä pelaamalla on mahdollisuus nostaa tai laskea Oliogotchin iloisuutta. Jos pelaaja voittaa Oliogotchin, sen iloisuus laskee 5 pisteellä ja jos Oliogotchi voittaa, iloisuus kasvaa 20 pisteellä.

Pallopelissä Oliogotchilla liikutaan A- ja D-näppäimillä vasemmalla ja oikealle, tarkoituksena on saada pallo kiinni. Pallo putoaa aina satunnaisesta paikasta. Jos pallo osuu maahan osumatta Oliogotchiin, pallo putoaa uudelleen satunnaisesta paikasta. Tällöin Oliogotchin iloisuus laskee 5 pistettä ja nälkäisyys 2 pistettä. Tämä toistuu niin kauan, kunnes Oliogotchi saa pallon kiinni. Kun Oliogotchi saa pallon kiinni, Oliogotchin iloisuus nousee 20 pistettä.

Kumpaakin minipeliä voi pelata niin kauan kuin jaksaa, mutta jos minipelistä poistuu ja pisteitä on niin vähän, että Oliogotchin iloisuus tai nälkäisyys putoaa nollaan, Oliogotchi kuolee. Pisteiden saantia on myös rajoitettu siten, ettei Oliogotchin iloisuusarvo voi ylittää sataa. Pistetilanne näkyy molemmissa peleissä ruudulla.

## Pelin animaatio

Oliogotchin animaatio koostuu useasta erillisestä kuvasta, jotka on piirretty käsin.

Pelinäkymän XAML-puolella on Image-tyyppinen kenttä, johon päivitetään Oliogotchin tyyppiä vastaava animaatio. Tässä kutsutaan Creature-luokan muuttujaa Ani, joka on samalla x:key arvo animaation kutsuntaan.

Kaikki animaatiot on määritelty Window.Resources -puolelle, jotta niitä voidaan kutsua haluttuina hetkinä koodin puolelta GameView.xaml.cs tiedostossa.

Oliogotchin animaatiot toteutettiin XAML:ssa käyttäen kahta Storyboardia. Käynnistettäessä ladataan sb-niminen Storyboard sekä evo-niminen muuttuja. Evolve-nappia painettaessa alkuperäinen Storyboard pysäytetään ja sen jälkeen Image-kenttään alustetaan ja otetaan käyttöön uusi evo-niminen Storyboard. Storyboardin lähdettä ei pysty muuttamaan dynaamisesti, jonka vuoksi toteutus on tehty kahdella eri muuttujalla.

Evolve-nappia painamalla Oliogotchi kehittyy seuraavaan muotoon sen syömän ruuan perusteella seuraavasti:

* jos se on syönyt enimmäkseen lihaa, siitä tulee lihansyöjä
* kasviksia syöttämällä siitä tulee kasvinsyöjä
* jos sekä lihaa että kasviksia on syötetty suunnilleen yhtä paljon, siitä tulee sekasyöjä.

Evolve -nappi on näkyvissä ainoastaan silloin, kun olio on alkulimamuodossa ja sillä on kaikissa kolmessa pääarvossaan (Happiness, Hunger, Cleanliness) yli 80 pistettä. Nappi piilotetaan, jos arvot laskevat tämän alapuolelle tai olio kehittyy aikuiseksi.

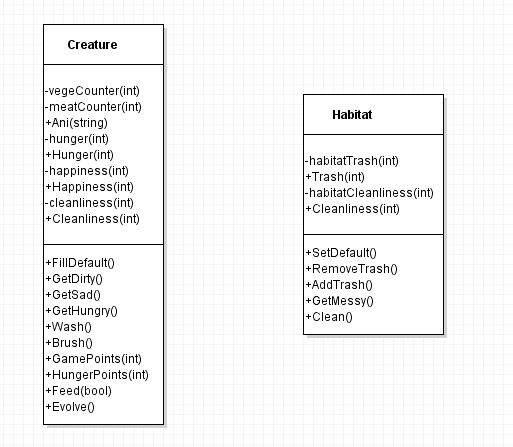
## Tallennus ja lataus

Oliogotchin ja elinympäristön tiedot tallennetaan kahteen erilliseen binaaritiedostoon, olio.bin ja tausta.bin. Tiedostot tallentuvat käyttäjän tietokoneelle, My Documents -kansioon. Tietojen tallennus tapahtuu aina kun aloitetaan uusi peli sekä pelinäkymän Main Menu –napista, eli kun siirrytään päävalikkoon. Mikäli pelin sulkee esim. rastista, pelitilannetta ei tallenneta.

Lataamista yritetään, kun pelaaja klikkaa päävalikossa painikkeesta Load. Tällöin peli jatkuu viimeisimmästä tallennuspisteestä. Mikäli tallennustiedosta ei löydy, peli aloitetaan Oliogotchin alkuarvoilla ja tästä ilmoitetaan pelaajalle pop-up –ikkunalla.

# Luokkien käyttö ja UML-kaavio

Lopullinen UML-kaavio (ks. kuvio 8) erosi jonkin verran vaatimusmäärittelyyn tehdystä kaaviosta. Vaatimusmäärittelyssä Oliogotchin kehittynyt muoto oli jaettu kolmeen eri luokkaan perustuen siihen, millainen Oliogotchista olisi tulossa (lihansyöjä, kasvissyöjä tai sekasyöjä). Peliä tehtäessä kävi kuitenkin ilmi, ettei kehittyneelle muodolle tarvitse luoda enää uutta oliota, sillä saimme tarvittavan toiminnon toteutettua kuvatiedostoa vaihtamalla.



Kuvio 8: Lopullinen rakennekaavio

Creature-luokassa on kapseloituna kaikki Oliogotchin attribuutit, joita olio tarvitsee. Kaikista on tehty public-muuttujat, joita käyttöliittymä käyttää, sekä private-muuttujat, joita voi käyttää vain itse olion sisällä.

Vaatimusmäärittelyssä määriteltyä Ruoka-luokkaa ei toteutettu ollenkaan, sillä siitä tehtiin napinpainalluksella käytettävä metodi. Tämä sisällytettiin suoraan itse Creature-luokkaan.

Taustalle tehtiin oma Habitat-luokka suunnitellun mukaisesti. Taustaan lisättäviä erillisiä koristeita ei toteutettu. Roskien lisäämiselle ja poistamiselle tehtiin toiminnot, vaan niitä ei toteutettu visuaalisesti.

# Testaaminen

Pyysimme Jyväskylän Ammattikorkeakoululla työskentelevää käyttäjäkokemuskehittäjää Heli Sutista testaamaan Oliogotchia. Heliltä saimmekin paljon palautetta, pääasiassa pelin visuaalisesta puolesta. Helin palautteen perusteella muutimme fonttia erilaiseksi, siirsimme nappien paikkoja loogisemmaksi, selkeytimme värejä ja saimme näin pelistä käyttäjäystävällisemmän. Heli toi myös esille, että Oliogotchin elämismekaniikka oli liian nopea, arvot laskivat liian nopeasti (1 sekunnin välein). Korjasimme ajaksi 2 sekuntia, joka on sopivampi.

Muutama eri testaaja koitti ensi alkuun käyttää pelin päänäkymässä olevia nappeja (pese, silitä, syötä) raahaamalla ne Oliogotchin päälle. Visuaalisesta ilmeestä ei käy heti ilmi, että niitä on tarkoitus klikata. Tämä olisi voitu korjata useallakin eri tavalla, esimerkiksi ohjeteksteillä tai hiiren kuvaketta muuttamalla. Tähän ei kuitenkaan enää ollut aikaa.

Pelistä tehtiin release-versio, joka annettiin parille testaajalle käyttöön omalle koneelleen. Tästä opittiin, etteivät pelin fontit ja äänet toimineet ollenkaan. Sama ongelma oli kaikissa release-versioissa. Ääniongelma korjattiin oikealla hakemistorakenteella, jotta sound-kansion tiedostot löytyivät suoritettavaan tiedostoon nähden oikeasta paikasta. Fonttiongelma olisi voitu varmasti ratkaista lisäämällä käytetty fonttitiedosto pelipakettiin mukaan ja kirjoittamalla sille polku koodiin. Fonttiongelmaa ei lähdetty sen enempää ratkomaan, sillä sitä ei koettu niin tärkeäksi. Todennäköisesti se olisi ratkennut samalla tavalla, kuin ääniongelmakin.

Yksikkötestausta ei tehty, koska ryhmä ei nähnyt tätä tarpeelliseksi. Koodissa käytetyt laskulliset toiminnot on helppo nähdä suoraan toiminnasta.

# Kehitettävää

Oliogotchiin voisi tehdä vaikeustason valinnan, jolla pääsisi esimerkiksi muuttamaan Oliogotchin arvojen laskemisen nopeutta. Vaikeimmalla tasolla Oliogotchin arvot laskisivat nopeammin ja minipeleissä ns. miinuspisteitä tulisi enemmän ja pluspisteitä vähemmän. Helpoimmalla tasolla taas luonnollisesti toisin päin, Oliogotchin arvot laskisivat hitaammin ja minipeleistä saisi paremmin pisteitä.

Peliin oli alun perin tarkoitus tehdä monta erilaista taustaa, jotka vastaisivat Oliogotchin kehittynyttä muotoa. Peliin tehtiin lopulta vain yksi tausta, joka näkyy kaikkien muotojen kanssa. Jokaiselle olisi kuitenkin mukava saada oma, yksilöllinen taustansa.

Pelinäkymässä eri napeilla (pese, silitä, syötä, kehity, pelaa minipelejä) voisi olla selkeämmät selitteet. Vaihtoehtoisesti ensimmäistä kertaa peliä käynnistettäessä voisi tulla pieni tutoriaali, joka opastaisi käytössä.

Pelinäkymän napeille voisi hyödyntää pallopeliin koodattua drag-and-drop –koodia, jotta pelaaminen olisi miellyttävämpää.

# Yhteenveto

Ryhmän mielestä tehty työ on arvosanan 5 arvoinen. Ryhmä oppi paljon ohjelman jakamisesta osiin, työnjaosta, aikataulutuksesta, työtuntien kirjaamisesta sekä tietenkin itse ohjelmoinnista. Ohjelmoinnin aikana tuli paljon erilaisia ongelmia, joiden ratkaisemisessa hyödynnettiin Googlea, opettajaa sekä luokkakavereita. Vaikka ihan jokaista suunniteltua ominaisuutta ei saatu toteutettua halutulla tavalla, ryhmälle kävi kuitenkin useimmissa tapauksissa selväksi, mikä meni vikaan ja miten asia olisi voitu korjata.

## Käytetty työaika

Työaikaa oli suunniteltu käytettäväksi noin 60 tuntia per henkilö (yhteensä siis 180 tuntia). Todellisuudessa tunteja ei kuitenkaan mennyt ihan niin paljoa, yhteensä ryhmällä meni noin 145 tuntia.

Tarkempi työaikakooste, josta näkyy jokaisen ryhmäläisen tekemät tunnit löytyy Githubista, osoitteesta

https://github.com/Gilyan/Olio-harjoitustyo/wiki/TODO

## Haasteista onnistumisiin

Ensimmäinen haaste koko työstä oli sisäistää ajatus peliohjelmoinnista. Tämän ratkaiseminen onnistui vain tekemällä harjoitustöitä ja etsimällä netistä apua. Haasteena oli myös havaita, mitä piti tehdä ensin. Muutaman tunnin päästä koodaaminen alkoi jo sujua hyvin.

Aluksi peli-ikkunoihin tehtiin rakenteet käyttäen Gridejä, jotta kaikki napit olisivat tasaisesti. Pian kuitenkin kävi ilmi, ettei itse Oliogotchin sijoittaminen Gridiin toiminut kovin hyvin. Gridit vaihdettiin tämän vuoksi Stackpaneleihin, joiden kanssa ei ollut samoja ongelmia.

Nappien aktiivisuutta ilmaisevan taustavärin poistaminen oli haastavaa. Kun hiiren vie minkä tahansa napin päälle, oletuksena napin taustaväri aktivoituu. Tämä ei sopinut pelin ulkonäköön, joten taustaväri haluttiin pois. Tätä varten luotiin oma tyyli XAML-tiedostoihin, jolla taustaväri saadaan läpinäkyväksi.

Napin tilan vieminen toiseen ikkunaan oli alkuun ongelmallista. Start-napista pelin tulisi alkaa alkuarvoilla alusta, kun taas Load-napista edellisen tallennetun pelin pitäisi jatkua. Pelinäkymään piti saada siis tieto, kumpaa nappia on painettu päävalikossa. Päävalikkoon tehtiin oma boolean-muuttuja, jonka arvo viedään pelinäkymässä sijaitsevaan paikalliseen muuttujaan.

Minipeleistä kertyneiden pisteiden siirtäminen Oliogotchille oli myös hankalaa. Tämä saatiin ratkaistua tekemällä kokonaan uusi luokka, jolla oli pari ominaisuutta ja metodia.

Jokaiselle pelin näkymälle tehtiin oma ikkuna. Tämä johti ongelmaan, että uusi ikkuna avautui pääsääntöisesti eri kohtaan kuin vanha ikkuna, joka näytti typerälle. Tämä ratkaistiin ylikuormittamalla uusi ikkuna, jolle vietiin tietoina vanhan ikkunan x- ja y-koordinaatit. Uusi ikkuna avautuu samoihin koordinaatteihin. Vaihtoehtoisesti tämä olisi saatu ratkaistua käyttämällä eri Stackpaneleita, joita olisi tarpeen mukaan piilotettu näkyvistä.

Pallopelin drag-and-drop-ominaisuuden toteutus oli haasteellista, sillä kunnollisia ohjeita ei meinannut löytyä mistään. Lopulta drag-and-drop saatiin kuitenkin toimimaan hyvin, mutta myöhemmän katselmoinnin jälkeen ominaisuus olikin täysin turha pelille ja ryhmä päätyi toiseen ratkaisuun. Drag-and-drop –toiminto on jätetty koodiin kommentoituna mahdollista myöhempää käyttöä varten.

Aluksi tarkoituksena oli toteuttaa animointi C#-puolella päivittämällä Image-objektiin uusi kuva määritellyin väliajoin. Tämä kuitenkin osoittautui todella työlääksi, sillä aika ei ollut sama kaikkien framejen välillä ja koko homma räjähti käsiin (40 riviä koodia yhteen 10 kuvan animaatioon). Lopulta Oliogotchin animaatiot toteutettiin XAML:ssa käyttäen Storyboardia.

Peliin tehtiin aluksi Settings-näkymä, josta pääsisi esimerkiksi poistamaan Oliogotchin ja muuttamaan sen nimeä. Peliä tehtäessä kuitenkin todettiin, että Oliogotchi ei tarvitse erillistä nimeä ja Oliogotchin nollaaminen käytännössä tapahtuu Start-nappulan kautta. Settings-näkymä päätettiin muokata ohjesivustoksi, josta löytyisi tarkemmat selitykset napeille ja toiminnoille.

Loppusilauksena peliin lisättiin vielä ääniä, jotta pelaamisesta tulee elävämpää. Tästä koitui uusia ongelmia, sillä äänet eivät toimineetkaan enää pelin release-versiossa. Ongelma ratkesi pakkaamalla sound-kansio mukaan ja laittamalla .exe-tiedosto oikeanlaiseen hakemistorakenteeseen ("../../Resources/sound/). Oikean hakemistorakenteen luominen oli helpompaa, kuin lähteä muuttamaan jokaiseen koodin kohtaan erilainen tiedosto-osoite.

Oliogotchin elämismekaniikka, kivi-sakset-paperi –peli sekä animaatio olivat parhaiten onnistuneet osiot työstä. Äänet toivat myös mukavan lisän pelikokemukseen. Ryhmä olisi ollut valmis laittamaan pelin Windows Storeen, mutta sinne rekisteröityminen maksaa. Peli päätettiin laittaa ryhmäläisen tekemälle sivulle vapaaseen jakoon.

## Kommentit kurssista

Kurssilla oli mukavan rauhallista, kun osa teki harjoitustaan kotona (tai jätti tekemättä). Tunneille sai tulla halutessaan tekemään ja apua oli tarjolla.

Esimerkkikoodinpätkiä ja demokertoja olisi voinut olla lisää. Demokerroilta jäi parhaimmat opit päähän ja opettajalta sai samalla kysyä, jos joku oli epäselvää.

Ehkä jonkinlainen puolivälitarkistus olisi hyvä, jossa jokainen tiimi esittelisi oman vaiheessa olevan työnsä muillekin. Verkkosivukurssilla pidimme pienen vilkaisun muiden raakiletöihin ja saimme siinä samalla itsekin ideoita, joita omaan työhön voisi toteuttaa.