

LUT-yliopisto  
School of Engineering Science  
Software Engineering

Kandidaatintyö

**Niku Grönberg**

## **KOULULAISILLE JÄRJESTETTÄVIEN OHJELMOINTIKURSSIEN VAIKUTTAVUUDEN MITTAAMINEN**

Työn tarkastaja(t): Tutkijaopettaja Jouni Ikonen

Työn ohjaaja(t): Tutkijaopettaja Jouni Ikonen  
Apulaisprofessori Antti Knutas

# TIIVISTELMÄ

LUT-yliopisto

School of Engineering Science

Tietotekniikan koulutusohjelma

Niku Grönberg

## **Koululaisille järjestettävien ohjelmointikurssien vaikuttavuuden mittaaminen**

Kandidaatintyö

2018

35 sivua, 6 taulukkoa

Työn tarkastaja: Tutkijaopettaja Jouni Ikonen

Hakusanat: kandidaatintyö, koululainen, mittaaminen, ohjelmointitaidot, kiinnostus

Keywords: bachelor's thesis, k-12 students, measuring, programming skills, interest

Tällä hetkellä yliopistoihin hakevat ylioppilaat eivät todennäköisesti ole ohjelmoineet aikaisemmin, jonka takia he eivät hae opiskelemaan tietotekniikkaa. Tästä syystä yliopistot ovat järjestäneet erilaisia ohjelmointikursseja koululaisille. Tässä työssä tutkitaan kahta erilaista koululaisille suunnattua ohjelmointikurssia, joissa on kummassakin teemana ohjelmointi ja pelien tekeminen. Ensimmäinen kurssi järjestettiin ala-astelaisille, ja toinen kurssi oli lukiolaisille suunnattu, mutta sen testiajo järjestettiin käytännössä yläasteen ja lukion opettajille.

Tavoitteena oli selvittää, miten kurssit vaikuttavat osallistujien ohjelmointitaitoihin sekä, miten kurssit vaikuttavat osallistujan kiinnostukseen ohjelmointia kohtaan. Lisäksi tavoitteena oli kehittää kurseja. Ohjelmointitaitoja ja kiinnostusta ohjelmointiin mitattiin kyselyillä. Kummaltakaan kurssilta ei saatu tilastollisesti merkittäviä tuloksia ohjelmointitaitojen tai kiinnostuksen ohjelmointia kohtaan muuttumisesta. Ala-astelaisten kanssa tämä johtui todennäköisesti siitä, että he olivat jo valmiiksi hyvin kiinnostuneita ohjelmoinnista, ja ohjelmointia mittaavat avoimet kysymykset eivät olleet hyvä tapa mitata heidän osaamistaan ohjelmoinnista. Opettajien suorittamalla kurssilla loppukyselyyn tuli niin vähän vastauksia, että tilastollisesti ei voida tehdä johtopäätöksiä. Kummallakin kurssilla suorittajat olivat kuitenkin hyvin tyytyväisiä kursseihin, ja kokivat itse, että heidän ohjelmointitaitonsa ja kiinnostuksensa ohjelmointia kohtaan kasvoi.

## **ABSTRACT**

LUT-university

School of Engineering Science

Degree program in Software Engineering

Niku Grönberg

### **Measuring the impact of programming courses held to k-12 students**

Bachelor's Thesis

2018

35 pages, 6 tables

Examiners: Associate professor Jouni Ikonen

Keywords: bachelor's thesis, k-12 students, measuring, programming skills, interest

Students currently applying for universities have most likely not programmed before and therefore do not apply for computer science studies. Because of this, universities have held extracurricular programming courses for k-12 students. This thesis studies two different programming courses both of which center around programming and making games. The first course was held to primary school pupils and the second was targeted to upper secondary school students but was actually held to teachers from middle and upper secondary schools.

The objective was to study how the courses affect the programming skills and interest towards programming of the participants. Additional objective was to improve these courses. Programming skills and interest towards programming were measured by questionnaires. Neither of the courses gave statistically significant results about the change in programming skills or interest towards programming. With the primary school pupils this was most likely a result of the fact that they were already very interested in programming and open-ended questions about programming were not a good way to measure their programming skills. In the other course where the participants were teachers, the questionnaire held at the end of the course was answered too few times, making it difficult to draw statistical conclusions. In both courses, participants who finished the course were happy with it and felt that their programming skills were improved and their interest towards programming had increased.

# SISÄLLYSLUETTELO

<b>1</b>	<b>JOHDANTO</b> .....	<b>3</b>
1.1	TUTKIMUSKYSYMYKSET JA TAVOITTEET .....	3
1.2	TUTKIMUSAINEISTO JA RAJAUKSET.....	4
1.2.1	<i>Peliohjelmoinnin alkeet Unitylla</i> .....	4
1.2.2	<i>Scratch-kurssi</i> .....	5
1.3	TYÖN RAKENNE .....	5
<b>2</b>	<b>OHJELMOINTITAITOT JA KIINNOSTUS OHJELMOINTIIN</b> .....	<b>7</b>
2.1	TIEDON HAKEMINEN .....	7
2.2	YLIOPISTOJEN KOULULAISILLE JÄRJESTÄMIÄ KURSSEJA .....	7
2.3	OHJELMOINTITAITOT .....	8
2.4	KIINNOSTUS OHJELMOINTIIN.....	10
<b>3</b>	<b>KURSSIKYSLEYIDEN LAATIMINEN</b> .....	<b>11</b>
3.1	TAUSTATIEDOT .....	11
3.2	MIELIPIDEOSIO .....	11
3.3	OHJELMOINTITAITOTESTI.....	12
3.4	IKÄEROT .....	13
<b>4</b>	<b>KYSELYIDEN TULOKSET</b> .....	<b>15</b>
4.1	SCRATCH-KURSSIN TULOKSET .....	15
4.1.1	<i>Scratch-kurssin mielipideosio</i> .....	16
4.1.2	<i>Scratch-kurssin toteutuminen</i> .....	17
4.1.3	<i>Scratch-kurssin ohjelmointitaitotesti</i> .....	17
4.2	PELIOHJELMOINNIN ALKEET UNITYLLA -KURSSIN TULOKSET .....	18
4.2.1	<i>Peliohjelmoinnin alkeet Unitylla -kurssin mielipideosio</i> .....	19
4.2.2	<i>Peliohjelmoinnin alkeet Unitylla -kurssin toteutuminen</i> .....	20
4.2.3	<i>Peliohjelmoinnin alkeet Unitylla -kurssin ohjelmointitaitotesti</i> .....	21
<b>5</b>	<b>TULOSTEN TULKINTA</b> .....	<b>23</b>
5.1	SCRATCH-KURSSIN TULOSTEN TULKINTA.....	23
5.2	PELIOHJELMOINNIN ALKEET UNITYLLA -KURSSIN TULOSTEN TULKINTA .....	24

<b>6</b>	<b>KURSSIEN KEHITTÄMINEN .....</b>	<b>26</b>
6.1	SCRATCH-KURSSIN PALAUTE JA KEHITTÄMINEN .....	26
6.2	PELIOHJELMOINNIN ALKEET UNITYLLA -KURSSIN PALAUTE JA KEHITTÄMINEN .....	27
<b>7</b>	<b>YHTEENVETO.....</b>	<b>29</b>
7.1	SCRATCH-KURSSI.....	29
7.2	PELIOHJELMOINNIN ALKEET UNITYLLA -KURSSI.....	29
	<b>LÄHTEET.....</b>	<b>31</b>

# 1 JOHDANTO

Vuodesta 2015 eteenpäin opetussuunnitelman mukaan myös ala- ja yläasteilla on opetettu ohjelmointia (Opetushallitus 2014). Tätä ennen peruskoulussa ei kuitenkaan ollut pakko sisällyttää ohjelmointia osaksi opetusta, ja tämä ohjelmoinnin opetuksen puute tarkoittaa, että tämän hetken yliopistohakijoista suurin osa ei ole ohjelmoinut aikaisemmin. Vuoden 2000 aikoihin tietotekniikkaa opiskelemaan hakevien opiskelijoiden määrä on laskenut paljon edeltävistä vuosista (Vegso 2005). Syitä hakijoiden määrän laskuun on monia, mutta yksi tärkeimmistä syistä on se, ettei hakijalla ole käsitystä, mitä tietotekniikassa opiskellaan ja millaisia mahdollisuuksia se tarjoaa työelämässä (Lomerson & Pollacia 2006, Carter 2006). Tähän tiedon puutteeseen ja vaillinaisiin käsityksiin tietotekniikasta voidaan vaikuttaa ja tätä tehdään yliopistoissa erilaisten ohjelmointikerhojen ja -työpajojen avulla, joita järjestetään peruskoululaisille sekä lukiolaisille.

Viime vuosina tietotekniikkaa opiskelemaan hakevien opiskelijoiden määrä on kuitenkin ollut nousussa. Hakijoiden määrä LUT-yliopiston tietotekniikan koulutuslinjalle on melkein tuplaantunut vuoden 2015 238 hakijasta 2018 vuoden 417 hakijaan (Vipunen 2018).

## 1.1 Tutkimuskysymykset ja tavoitteet

Työssä tutkitaan kahden hyvin erilaisen kurssin vaikuttavuutta oppilaiden taitoihin ja kiinnostukseen. Lisäksi selvitetään mikä kurseissa onnistui ja mikä ei onnistunut ja miten kurseja voitaisiin kehittää.

Tavoitteena on arvioida, miten LUT-yliopiston kurssit vaikuttavat osallistujien mielenkiintoon ohjelmointia kohtaan ja millaisia muutoksia tapahtuu osallistujien ohjelmointitaidoissa. Lisäksi tavoitteena on kehittää näitä kurseja. Tutkimuskysymykset ovat seuraavat:

- 1) Millä tavalla yliopiston koululaisille tarjoamat kurssit vaikuttavat osallistujan kiinnostukseen ohjelmointia kohtaan?
- 2) Miten kurssit vaikuttavat osallistujan ohjelmointitaitoihin?

3) Miten kurssveja voidaan kehittää annetun palautteen sekä kokemusten perusteella?

Tutkimuskysymysten lisäksi tutkitaan, miten ohjelmointitaitoja ja kiinnostusta ohjelmointia kohtaan voidaan mitata ja miten niitä on jo mitattu aikaisemmissa tutkimuksissa.

## **1.2 Tutkimusaineisto ja rajaukset**

Työssä tutkitaan kahta eri kurssia. Kurssien vaativuus, toteutustavat, ja osallistujat ovat hyvin erilaisia, mutta kummassakin kurssissa on sama teema, eli ohjelmoinnin perusteiden opettaminen ja pelien tekeminen. Tutkimalla kahta eri kurssia, saadaan laajempi näkemys siitä, miten kurssit vaikuttavat osallistujan ohjelmointitaitoihin ja kiinnostukseen ohjelmointia kohtaan.

Ensimmäinen tutkittava kurssi on Peliohjelmoinnin alkeet Unitylla, jossa opetellaan ohjelmoinnin perusteita, sekä pelien tekemistä pelimoottorilla Unity. Toinen tutkittava kurssi on ala-astelaisille järjestettävä Scratch-kerho, jossa tehdään pelejä käyttäen visuaalista ohjelmointikieltä Scratchia.

### **1.2.1 Peliohjelmoinnin alkeet Unitylla**

Peliohjelmoinnin alkeet Unitylla on itsenäisesti suoritettava verkkokurssi, jossa osallistuja oppii perusteet Unityn käytöstä, sekä ohjelmoinnista kielellä C#. Kurssi on mitoitettu noin 26 tuntiseksi. Kurssi on suunnattu 8.—9.-luokkalaisille sekä lukiolaisille eli 14—18 vuotiaille. Unity on suosittu pelimoottori, jota käytetään yleisesti myös ammattimaiseen pelinkehitykseen. Unityn käyttö on ilmaista, jos käyttäjän yrityksen tulot ovat alle 100 000 dollaria (Unity, 2018).

Kurssi on rakennettu siten, että suorittamalla tehtäviä pääsee etenemään aina seuraavalle tasolle, jossa käsitellään uusia aiheita. Unitya opetetaan käyttämään esimerkkivideoiden avulla, joita seuraamalla saadaan tehtyä kurssin aikana melko yksinkertainen pallopeti. Ohjelmoinnin oppiminen tapahtuu Microsoftin oman C#-tutoriaalini, itseopiskelumateriaalin ja harjoitustehtävien avulla. Kielenä on C#, sillä se on sama kieli, jota käytetään Unityssa pelien skriptaamiseen. Näin vältetään siltä, että suorittaja joutuu

heti aluksi opettelemaan useamman ohjelmointikielen. Vaikka kyseessä on oliokieli, niin sitä ei käytetä kurssilla olio-ohjelmointiin, vaan kurssissa keskitytään ohjelmoinnin tärkeisiin peruseräisiin, kuten muuttujiin, tietotyyppeihin, valintarakenteisiin, toistorakenteisiin, listoihin ja funktioihin. Kurssilla suoritettavat ohjelmat ovat siis täysin yhden luokan sisäisiä, jolloin suorittajan ei tarvitse opetella olio-ohjelmoinnin perusteita. Kurssin jälkeen osallistujan pitäisi osata tehdä yksinkertaisia pelejä Unitylla sekä ohjelmoida yksinkertaisia konsolissa ajettavia ohjelmia.

### **1.2.2 Scratch-kurssi**

Scratch-kurssi on 4.—6.-luokkalaisille (10—12 vuotiaille) järjestettävä ohjelmointikurssi. Scratch on visuaalinen ohjelmointikieli, jossa ohjelmointi tapahtuu yhdistelemällä värikkäitä palikoita, jotka sisältävät toimintoja, kuten vertailua, ehtolauseita tai toistorakenteita. Scratchia pystyy käyttämään suoraan selaimella, joten oppilaat voivat käyttää sitä niin kurssilla kuin myös kotona. Scratch perustuu siihen, että koodilla liikuttamalla hahmoja ruudulla (tehdään pelejä) ja tämä välitön visuaalinen palaute auttaa oppilaita näkemään, mitä koodi tekee käytännössä.

Kurssia järjestettiin viikoittain neljän viikon ajan, ja yksittäinen kerta kesti 90 minuuttia. Opetus tapahtui esimerkin avulla siten, että ohjaaja teki peliä luokan edessä ja osallistujat seurasivat ja tekivät itse. Kurssilla oli yksi ohjaaja neljää oppilasta kohden, joten kaikki oppilaat saivat tarvitsemansa tuen pelin tekemiseen. Loppua kohden oppilaille annettiin myös vapaat kädet kokeilla itse lisätä omia ominaisuuksiaan tekemäänsä peliin. Kerhossa pyrittiin myös opettamaan ohjelmoinnin periaatteita, kuten muuttujia, valinta- ja toistorakenteita, ja funktioita teoreettisemmalla tasolla kuitenkin niin, että ohjelmoinnin periaatteet sidottiin kiinni Scratchin palikoihin.

## **1.3 Työn rakenne**

Teoriaosiossa käydään läpi vastaavia tutkimuksia ohjelmointikursseista ja niiden mittaamisesta. Lisäksi käydään tarkemmin läpi erilaisia tapoja mitata ohjelmointiosaamista sekä kiinnostusta. Tätä seuraa selostus siitä, miten tässä työssä on mitattu ohjelmointikursseja.



Teoriaosion jälkeen käydään läpi mittaustulokset ja arvioidaan niitä. Lopussa ehdotetaan tapoja kehittää kurseja tunnistettujen ongelmakohtien perusteella.

## **2 OHJELMOINTITAITODOT JA KIINNOSTUS OHJELMOINTIIN**

Yliopiston koululaisille järjestämät ohjelmointikurssit ovat siitä mielenkiintoisessa tilanteessa, että osallistujat eivät yleensä ole ohjelmoineet aikaisemmin, tai se on tapahtunut vastaavalla kurssilla aiemmin. Tämä tarkoittaa, että kursseilla täytyy aloittaa ohjelmoinnin opettaminen alusta asti, mutta kuitenkin niin, että kurssi pysyy kiinnostavana, etenkin nuorempien parissa työskennellessä. Tästä syystä etenkin hyvin lyhyissä kursseissa ohjelmoinnin teorian opettaminen jää usein vähäiseksi, sillä kurssilla pitää saada jotain aikaseksi, että oppilaat kokevat tehneensä jotain.

### **2.1 Tiedon hakeminen**

Tavoitteena oli selvittää aikaisempien vastaavien kurssien olemassaolo, miten aikaisempia kursseja on mitattu, ja millaisia tuloksia on saatu aikaisempien vastaavien kurssien mittaamisesta. Lisäksi tavoitteena oli löytää tapoja mitata ohjelmointitaitoja.

Tieteellisiä tekstejä aiheesta haettiin palvelulla Google Scholar. Hakusanoina käytettiin aiheeseen liittyviä termejä kuten: ”programming”, ”computer science”, ”teaching”, ”beginner”, ”outreach event”, ”k 12”, ”programming skills”, ja erilaisia yhdistelmiä näistä termeistä. Löytyneistä hakutuloksista arvioitiin otsikon perusteella niiden oleellisuutta tämän tutkimuksen kannalta sekä sitä, olivatko ne ilmaiseksi saatavilla.

### **2.2 Yliopistojen koululaisille järjestämiä kursseja**

Yliopistot ovat järjestäneet kursseja, kerhoja ja muita työpajoja koululaisille jo pitkään. Tarkoituksena on ensisijaisesti lisätä osallistujien kiinnostusta tietotekniikkaa kohtaan. Tällaisia kursseja on järjestetty useita, ja niitä on mitattu vaihtelevilla tavoilla. Osassa kursseja keskityttiin tutkimaan osallistujan kiinnostusta tietotekniikkaa kohtaan, kun taas toisissa keskityttiin osallistujan taitoihin.

Isomöttönen et al. (2011) järjestämässä kurseissa n. 150 yläastelaista käytti Jypeli-kirjastoa C#:ssa pelien tekemiseen. Kurssit järjestettiin kesinä 2009 ja 2010. Kurssit olivat noin 25 tuntisia ja ne pidettiin viikon aikana. Tietoa kurssista kerättiin kyselyillä kurssin alussa ja lopussa. Osallistujissa mitattiin kiinnostuksen lisääntymistä hieman ohjelmoinnin opiskelusta kohtaan, ja osallistujat luulivat ohjelmointia vaikeaksi ennen kuin he pääsivät kokeilemaan sitä käytännössä, jonka jälkeen osallistujat pitivät ohjelmointia helpompana. Osallistujat olivat tyytyväisiä kursseihin. (Isomöttönen et al. 2011.). Kaksi vuotta myöhemmin osallistujat olivat suurimmaksi osaksi tehneet jotain ohjelmointiin liittyvää omatoimisesti, ja kiinnostus tieteiden opiskeluun oli vieläkin suurempi kuin ennen kurssia (Lakanen et al. 2012).

Meerbaum-Salant et al. (2013) tutkivat kurssia, jonka he järjestivät yläastelaisille osana tavallista opetusta. Kurssilla käytettiin Scratchia ohjelmoinnin opettamisessa, ja se kesti lukukauden verran kaksi tuntia viikossa. Kurssia mitattiin testillä alussa, keskivaiheessa, ja lopussa. Testit perustuivat SOLO- ja uusittuun Bloom -taksoniaan. Osallistujien ymmärtäminen ohjelmointikonsepteista kehittyi kurssin aikana. (Meerbaum-Salant et al. 2013.)

Franklin et al. (2013) järjestivät Scratch-ohjelmointikurssin kesällä 2012 yläastelaisille. Osallistujia oli 35 kappaletta ja kurssi kesti kaksi viikkoa. Tässä kurssissa ei järjestetty testiä tai kyselyä aluksi, sen sijaan osallistujien taitoja arvioitiin ohjaajien kurssin aikana käymien keskusteluiden pohjalta, sekä osallistujien tekemien projektien kautta. Kurssin jälkeen osallistujilla oli taitoja monissa ohjelmoinnin peruskonsepteissa. (Franklin et al. 2013.)

### **2.3 Ohjelmointitaidot**

Tässä työssä tutkitaan ohjelmointia lähinnä oppimisen ja opettamisen kannalta, sillä työssä tutkittavat kurssit ovat kummatkin suunnattuja aloittelijoille. Ohjelmointi sisältää monia eri taitoja, jotka ovat sidoksissa toisiinsa ja joita kaikkia tarvitaan ohjelmoinnin osaamiseen. Näitä taitoja ovat esimerkiksi yleiset ongelmanratkaisutaidot, ymmärtäminen miten tietokone suorittaa ohjelmia, syntaksi, ja erilaisten ohjelmointimallien ja -konseptien ymmärtäminen kuten ehtolauseet ja toistorakenteet.

Ohjelmointitaitojen mittaaminen ei ole siis helppoa, sillä ohjelmointi on hyvin kattava kokonaisuus. McCracken et al. (2001) laativat kehyksen ohjelmointitaitojen mittaamiselle, joka koostuu viidestä osa-alueesta:

1. Ongelman abstrahointi sen kuvauksesta
2. Ongelman jakaminen pienempiin osaongelmiin
3. Yksittäisten osaongelmien ratkaiseminen
4. Ratkaisujen yhdistäminen kokonaiseksi ohjelmaksi
5. Arvioiminen ja iteroiminen

Tämän kehyksen tarkoituksena oli mitata ensimmäisen vuoden tietotekniikan opiskelijoiden ohjelmointitaitoja. Osallistujien taitojen mittaaminen tehtiin ohjelmointikokeella. Käyttämällä McCracken et al. (2001) kehystä saadaan kattava kuva osallistujan taidoista ja käytännön ohjelmointitestillä voidaan varmistua, että osallistuja kykenee varmasti ohjelmoimaan. Tällainen testi on kuitenkin turhan vaativa, jos tarkoituksena on mitata ala-astelaisten ohjelmointitaitoja Scratch-kerhon jälkeen, tai vähintäänkin heille pitäisi laatia Scratchilla suoritettava testi.

Scratch-kurssille sopivampi lähestymistapa voisi seurata Franklin et al. (2013) toteuttamaa tutkimusta, jossa dataa kerättiin oppilaiden tekemistä lopullisista töistä, sekä keskusteluista, joita kurssin vetäjät kävivät oppilaiden kanssa. Tällöin välttyään testeiltä, jotka tuntuvat koululta oppilaille (Franklin et al. 2013). Tällä lähestymistavalla ei kuitenkaan pysty mittaamaan osallistujien lähtötasoa, sekä tätä ei pysty toteuttamaan verkkokurssilla, sillä keskusteluja osallistujien kanssa käydään parhaimmillaankin sähköpostin välityksellä, ja kurssin palautettavien töiden tekijästä ei voida olla täysin varmoja.

Meerbaum-Salant et al. (2013) tutkimuksessa käytettiin testejä, jotka perustuivat SOLO- ja uusittuun Bloom -taksonomiaan. Näiden taksonomioiden pohjalta Meerbaum-Salant et al. (2013) laativat kehyksen ohjelmointitaitojen mittaamiseen, jonka avulla saadaan melko kattavasti käsitystä oppilaan taidoista. Tällainen testaaminen kuitenkin tuntuu kokeelta osallistujalle, eikä myöskään ajallisesti sovi tässä työssä tutkittavalle Scratch-kurssille.

## 2.4 Kiinnostus ohjelmointiin

Lakanen et al. (2012) ja Isomöttönen et al. (2011) mittasivat ohjelmointikurssin vaikuttavuutta koululaisiin kysymyksellä ”Aion hakeutua opiskelemaan tieteitä korkeakoulussa”. Isomöttönen et al. (2011) kysymys oli muotoiltu ”Aion hakeutua opiskelemaan ohjelmointia/tieteitä korkeakoulussa”. Kummassakin tapauksessa kurseille osallistuneet olivat kiinnostuneempia tieteiden opiskelusta kurssin jälkeen kuin ennen kurssia. (Lakanen et al. 2012, Isomöttönen et al. 2011).

Zorn et al. (2013) tutkivat alku- ja loppukyselyiden avulla, miten Minecraftiin tehty ohjelmointilaajennus vaikuttaa osallistujien kiinnostukseen ohjelmointia kohtaan. Minecraft on peli, jossa pelaajat päättävät itse mitä tekevät loputtomassa avoimessa palikoista koostuvassa maailmassa (Mojang 2018). Zorn et al. (2013) jakoivat kyselyn väittämät neljään eri kategoriaan: kiinnostus, vaikeus, nautinto/huvi, ja hyödyllisyys. Väite kiinnostuksesta oli esimerkiksi: ”Olen kiinnostunut ohjelmoinnista”, ja väite hyödyllisyydestä oli esimerkiksi: ”On hyödyllistä osata ohjelmoida”. Väitteisiin vastattiin 7-portaisella Likert-asteikolla. (Zorn et al. 2013).

### **3 KURSSIKYSLEYIDEN LAATIMINEN**

Tiedon kerääminen kummaltakin tutkittavalta kurssilta tapahtui kyselyiden avulla. Aluksi tehtiin taustakysely, jossa selvitettiin osallistujista perustietoja, aiempaa ohjelmointikokemusta, mielipiteitä ohjelmoinnista ja syitä miksi osallistui kurssille. Lisäksi suoritettiin ohjelmointitaitoja mittaava testi. Kurssin lopuksi järjestettiin loppukysely, joka sisälsi saman mielipideoision ohjelmoinnista ja saman ohjelmointitaitotestin, sekä lisäksi osallistujalta itseltään kysyttiin mielipiteitä siitä, miten kurssi vaikutti hänen ohjelmointitaitoihinsa ja kiinnostukseen ohjelmointia kohtaan. Alku- ja loppukyselyn vastauksia verrataan toisiinsa ja tavoitteena on löytää eroja, jotka voidaan perustella sillä, että kurssi on vaikuttanut osallistujaan ja hänen mielipiteisiinsä.

#### **3.1 Taustatiedot**

Kyselyn alussa selvitettiin taustatietoja osallistujasta. Osallistujan nimeä kysyttiin, että alku- ja loppukysely voidaan yhdistää toisiinsa. Lisäksi kysyttiin muita yleisiä taustatietoja, kuten ikä, sukupuoli, koulu ja luokka.

Kummatkin mitattavat kurssit käsittelevät ohjelmointia ja pelien tekemistä, joten osallistujalta kysyttiin aiempaa ohjelmointikokemusta, sekä sitä kuinka paljon osallistuja pelaa videopelejä viikossa. Viimeisenä taustatietona kysyttiin, miksi osallistuja osallistui kurssille. Kurssille osallistumisen syyt otettiin osittain Lakanen et al. (2012) kyselystä.

#### **3.2 Mieliideo**

Kyselyn mieliideo rakennettiin useasta tutkimuksesta ja niissä käytetyistä kyselyistä. Isomöttönen et al. (2011) kyselystä otettiin väittämä: ”Olen kiinnostunut hakemaan opiskelemaan tieteitä/ohjelmointia” pohjalle. Tämä väite yksinkertaistettiin muotoon ”Ohjelmointi on minusta kiinnostavaa”. Mieliideoon lisättiin väittämä Lakanen et al. (2012) kyselystä: ”Voisin nähdä itseni työskenteleväni ohjelmoinnin parissa”. Lakanen et al. (2014) kyselystä otettiin vielä väittämä: ”Koen ohjelmoinnin vaikeaksi”. Näiden

väittämien pohjalta mielipideosioon lisättiin vastaavat väittämät, joissa kysyttiin ohjelmoinnin sijaan pelien tekemisestä samoja asioita.

Vastaajan mielipiteitä ohjelmoinnista tutkittiin esittämällä väittämiä ja vastaaja ilmaisi mielipiteensä väittämistä neliportaisen Likert-asteikon avulla. Neliportaista asteikkoa käytettiin, jotta välttyttiin tyhjiltä ”ei mielipidettä asiasta” vastauksilta. Loppukyselyssä kysyttiin myös vastaajan omaa mielipidettä siitä, onko kurssi kehittänyt hänen ohjelmointitaitojaan. Tätä voidaan käyttää mittarina sille, toimivatko ohjelmointitaitojen avoimet kysymykset ohjelmointitaitojen mittaamisessa, jos osallistujan omat kokemukset eroavat merkittävästi siitä, mitä ohjelmointitaitotesti antaa tulokseksi.

Ohjelmointi mielipiteiden lisäksi kysyttiin mielipiteitä pelien tekemisestä, sillä kummassakin kurssissa teemana oli pelien tekeminen. Tämä tieto voi auttaa kurssien kehittämisessä, sillä se kertoo, kumpi aihe koetaan kurssilla kiinnostavammaksi pelien tekeminen vai ohjelmointi. Tämän tiedon perusteella voidaan ehkä muuttaa kurssien painotusta suuntaan tai toiseen.

Mielipideväittämät ohjelmoinnista ja pelien tekemisestä käsittelevät suunnilleen aihepiirejä ohjelmoinnin ja pelien tekemisen kiinnostavuus, hyödyllisyys ja haastavuus. Kysymällä useampi kysymys useasta eri näkökulmasta saadaan kattavampi kuva osallistujan mielipiteistä.

### **3.3 Ohjelmointitaitotesti**

Ohjelmointitaitoja mitattiin avoimilla kysymyksillä tärkeistä ohjelmointiperiaatteista. Tärkeiksi ohjelmointiperiaatteiksi valittiin ohjelmointi, muuttujat, ehtolauseet, toistorakenteet ja aliohjelmat. Nämä ohjelmointiperiaatteet valittiin sen takia, että kummatkin kurssit käyvät ainakin nämä asiat läpi ja esimerkiksi listat jätettiin pois, sillä niitä ei käsitellä Scratch-kurssilla. Kysymykset olivat sellaisia, että vastaajaa pyydettiin selittämään mitä nämä peruseriaatteet tarkoittavat. Tällaiset avoimet kysymykset eivät mittaa henkilön ohjelmointitaitoja kokonaisuudessaan, eivätkä ota kantaa vastaajan ongelmaratkaisutaitoihin. Todellisen ohjelmointitestin järjestäminen näille kahdelle eri kurssille ei kuitenkaan ollut mahdollista, johtuen aikarajoitteista, suuresta erosta iässä, ja

kurssien eri järjestämistavoista (lähiopetus ja verkkokurssi). Eli vaikka kyselyn avoimet kysymykset ohjelmoinnin peruseriaatteista eivät olekaan suora mittari vastaajan ohjelmointitaidoista, niin nämä peruseriaatteet ovat kuitenkin tärkeä osa ohjelmointiosaamista ja niiden ymmärtäminen on edellytys ongelmien ratkaisemiselle ohjelmoimalla. Samat kysymykset kysytään niin kurssin alussa, kuin kurssin lopussa, jolloin voidaan nähdä, onko kurssi vaikuttanut osallistujiin mitenkään.

Kysymyksiin on laadittu kriteerit, mitä oikean vastauksen tulisi ainakin sisältää, ja arvioimalla vastauksia näillä kriteereillä, voidaan todeta vastaukset oikeiksi tai vääriksi. Vastaukset arvioidaan siis ainoastaan skaalalla oikein/väärin. Ala-astelaisilta ei edellytetä niin tarkkoja vastauksia kuin, Peliohjelmoinnin alkeet Unitylla -kurssin suorittajilta johtuen ikäerosta ja suuresta erosta kurssien kestossa.

Yksi avoimien kysymyksien ongelmista on se, että ohjelmointikonsepteista käytetään tiettyjä termejä, ja on täysin mahdollista, että vastaaja on ymmärtänyt jonkin ohjelmointikonseptin, mutta ei muista oikeaa termiä kyselyn aikana, jolloin vastaus saattaa jäädä puuttumaan vain sen takia, ettei vastaaja muistanut oikeaa sanaa. Tätä on yritetty ehkäistä käyttämällä useita samaa tarkoittavia termejä samassa kysymyksessä, esimerkiksi kysymys toistorakenteista sisältää myös tavallisemman termin looppi.

Kyselyssä painotettiin, että kurssin aluksi näitä ohjelmointikonsepteja ei tarvitse vielä osata, ja vastaajaa kannustettiin joko arvaamaan, tai jättämään vastaus tyhjäksi, jos hänellä ei ole mitään aavistusta. Lisäksi painotettiin, ettei kyselyllä ole merkitystä kurssin suorittamiseen, vaan että se on puhtaasti tutkimuksen tekemiseen. Tällä yritetään välttää oppilaille kyselystä aiheutuvaa stressiä, joka voi heti kurssin aluksi antaa negatiivisemmän kuvan ohjelmoinnista. Tämä voi kuitenkin aiheuttaa ongelmaksi sen, että vastaaja ei laiskuutakaan jaksa vastata kysymyksiin, vaikka osaisikin vastata kysymykseen oikein.

### **3.4 Ikäerot**

Yksi ongelmista kyselyn laatimisessa oli se, että miten voidaan samalla kyselyllä mitata niin ala-astelaisia, kuin 8.—9.-luokkalaisia ja lukiolaisia. Kyselystä täytyi tehdä tarpeeksi yksinkertainen ja selkeä, että ala-astelaiset pystyvät suorittamaan sen ilman ulkopuolista



apua. Samalla kyselyn täytyi kuitenkin olla tarpeeksi kattava, että sen pohjalta voidaan tehdä johtopäätöksiä. Suurimmaksi ongelmaksi muodostui ohjelmointitaitojen mittaaminen kummallakin tasolla, sillä ikäeroa oli monta vuotta. Tästä syystä päädyttiin avoimiin kysymyksiin ohjelmoinnin tärkeimmistä konsepteista. Ne pysyvät samana ohjelmointikielestä riippumatta, ja niiden ymmärtäminen on tärkeää kummallakin kurssilla. Kysymykset pyrittiin vielä muotoilemaan mahdollisimman vähän uhkaaviksi, jolloin alastelaisten vastausten määrä saattaa nousta.

## 4 KYSELYIDEN TULOKSET

Kyselyt järjestettiin kummallakin kurssilla heti kurssin aluksi, ennen kuin mitään opetusta oli pidetty, ja kurssin lopuksi, kun kaikki aiheet oli käyty läpi. Scratch-kerhossa kyselyt järjestettiin paperisena, jotta ne olisivat mahdollisimman helppoja ala-astelaisille täyttää. Peliohjelmointikurssilla kyselyt järjestettiin nettikyselynä. Ohjelmointikiinnostusta arvioitiin usealla kysymyksellä, joihin vastattiin neliportaisella Likert-asteikolla. Ohjelmointitaitoja mitattiin avoimilla kysymyksillä, jotka arvioitiin joko oikeiksi tai vääriksi.

Alkukyselyn mielipideosiota ja ohjelmointitaitotestiä verrattiin loppukyselyn mielipideosioon ja ohjelmointitaitotestiin Wilcoxonin testillä. Wilcoxonin testissä verrataan diskreettejä arvopareja toisiinsa, ja tuloksena on p-arvo, joka kertoo todennäköisyyden, että arvojoukoissa tapahtuvat muutokset ovat sattumanvaraisia (Wohlin et al. 2012). Nollahypoteesina toimii se, että kurssi ei ole vaikuttanut oppilaiden ohjelmointitaitoihin tai kiinnostukseen ohjelmointia kohtaan. Nollahypoteesi voidaan hylätä, jos p-arvo on merkittävä, eli perinteisesti alle 0,05. Tämä ei kuitenkaan yksinään ole hyvä mittari tilastolliselle merkittävyydelle (Bangdiwala 2016). Tämän takia kurssin vaikutusta oppilaisiin kysyttiin myös suoraan loppukyselyssä.

### 4.1 Scratch-kurssin tulokset

Kerhoa järjestettiin yliopiston tietokoneluokassa, ja luokassa olevien tietokoneiden avaaminen vaati käyttäjätunnuksen ja salasanan. Salasana oli pitkä ja satunnainen, jonka takia oppilaat eivät voineet kirjautua sisään ennen kuin heille näytettiin taululta kyseinen salasana. Alkukysely pidettiin ensimmäisellä kerralla ennen kuin oppilaille näytettiin tunnukset, joilla kirjautua koneelle. Viimeisellä kerralla oppilaiden ei annettu kirjautua koneelle heti aluksi, vaan pidettiin lyhyt kertaus, jossa kyselyssä kysytyjä ohjelmointikonsepteja sidottiin kiinni Scratch:in ohjelmointipalikoihin. Heti kertauksen jälkeen oppilaat vastasivat loppukyselyyn, jonka jälkeen kerhoa jatkettiin normaalisti. Kummatkin kyselyt pidettiin heti kerhon aluksi, jotta oppilaiden Scratch työskentelyä ei jouduttu keskeyttämään ennenaikaisesti kyselyyn vastaamisen takia.

#### 4.1.1 Scratch-kurssin mielipideosio

Alkukyselyyn vastasi 12 oppilasta 13:sta. Vastaamatta jättänyt oppilas saapui myöhässä paikalle, joten häntä ei laitettu vastaamaan kyselyyn. Loppukyselyyn vastasi 10 oppilasta 13:sta, puuttuvat vastaukset johtuivat poissaoloista. Kaikki osallistujat olivat poikia ja he olivat iältään 10-12 vuotiaita. Viisi kahdestatoista oppilaasta vastasi alkukyselyssä ohjelmoineensa aikaisemmin. Taulukossa 1 on mielipideosiosta saadut tulokset. Taulukko 1 sisältää väitteen, alkukyselyn tulosten keskiarvon ja mediaanin, loppukyselyn tulosten keskiarvon ja mediaanin ja Wilcoxonin testistä saadun p-arvon. Vastaukset olivat Likert-asteikolla, jossa 1=erimieltä ja 4=samaa mieltä.

**Taulukko 1 Scratch-kerhon mielipideosio**

Väite	Alkukysely keskiarvo (mediaani)	Loppukysely keskiarvo (mediaani)	p-arvo
Ohjelmointi on minusta kiinnostavaa	3,75 (4)	3,9 (4)	0,157
Koen ohjelmoinnin hyödylliseksi taidoksi	3,08 (3)	3,2 (3)	0,414
Pelien tekeminen on minusta kiinnostavaa	3,75 (4)	3,8 (4)	0,564
Koen pelien tekemisen hyödylliseksi taidoksi	3,1 (3)	3,2 (3)	1,0
Voisin nähdä itseni työskentelevän ohjelmoinnin parissa	3,36 (4)	3,2 (3,5)	0,655
Voisin nähdä itseni työskentelevän pelien tekemisen parissa	3,63 (4)	3,3 (3,5)	0,18
Koen ohjelmoinnin vaikeaksi	2,09 (2)	1,8 (2)	0,564

Nollahypoteesia ei voida hylätä p-arvon perusteella missäkään väittämässä. Otoskoko väittämässä on kuitenkin pieni, joten riski virheelle on olemassa. Väitteiden keskiarvot näyttäisivät pääasiassa kuitenkin kasvavan loppukyselyssä, paitsi väitteissä ohjelmoinnin ja pelien tekemisen parissa työskentelyn suhteen, jotka ovat vähentyneet. Ohjelmointi myös näyttäisi tuntuvan helpommalta oppilaille kurssin jälkeen. Kyselyn perusteella ei myöskään

ollut suurta eroa siinä pitikö ohjelmointia vaikeana, sen perusteella oliko ohjelmointi aikaisemmin vai ei. Keskimääräisesti jos oppilas oli ohjelmoinut aikaisemmin, niin hänen mielestään ohjelmointi oli hieman helpompaa, kuin oppilaan mielestä, joka ei ollut ohjelmoinut aikaisemmin. Tämä ero näkyi niin alkukyselyssä kuin myös loppukyselyssä, mutta erot tasoittuvat loppukyselyä kohden.

#### 4.1.2 Scratch-kurssin toteutuminen

Kyselyiden mielipideosion lisäksi oppilaat vastasivat itse loppukyselyssä, miten heidän kiinnostuksensa on muuttunut ohjelmointia kohtaan. Taulukko 2 sisältää väittämiä kurssista ja sen toteutumisesta. Vastaukset olivat Likert-asteikolla, jossa 1=erimieltä ja 4=samaa mieltä.

**Taulukko 2 Scratch-kerhon toteutuminen**

Väite	Keskiarvo
Koen, että ohjelmointitaitoni kehittyivät kurssin aikana	3,4
Koen kurssin hyödylliseksi	3,5
Voisin suositella kurssia tutulle	3,5
Kurssi lisäsi kiinnostustani ohjelmointia kohtaan	3,3

Oppilaat kokivat, että kurssi lisäsi heidän kiinnostustaan ohjelmointia kohtaan. Tämä tukee mielipideosion väitettä ”Ohjelmointi on minusta kiinnostavaa”, vaikka väitteen vastauksissa tapahtunut positiivinen muutos ei ollutkaan tilastollisesti merkittävä. Sama pätee väitteille ohjelmoinnin ja pelien tekemisen hyödyllisyydestä, kummassakaan väitteessä tapahtunut muutos ei ollut tilastollisesti merkittävä, mutta oppilaat silti kokivat kurssin erittäin hyödylliseksi. Kurssilla ei kuitenkaan opetettu mitään muuta kuin ohjelmointia ja pelien tekemistä Scratchilla. Oppilaat olivat hyvin tyytyväisiä kurssiin, joka näkyi väitteessä ”Voisin suositella kurssia tutulle”.

#### 4.1.3 Scratch-kurssin ohjelmointitaitotesti

Ohjelmointitaitotestiin tuli Scratch-kerhon osallistujilta todella vähän vastauksia niin alkukyselyssä kuin myös loppukyselyssä. Alkukyselyssä tämä selittyy sillä, että termit ohjelmointiin liittyvät termit olivat todennäköisesti tuntemattomia oppilaille.

Loppukyselyyn mennessä jokaisesta termistä oli ollut puhetta kurssin aikana, sekä ne kerrattiin ennen kyselyä, mutta vastauksia tuli silti melko vähän. Suurin osa jätti vastaamasta kysymyksiin, mutta he ketkä vastasivat, olivat useimmiten oikeassa tai ainakin oikeilla jäljillä. Taulukko 3 sisältää ohjelmointitaitotestin tulokset. Tehtävien vastaukset arvioitiin joko oikeaksi (1 piste) tai vääräksi (0 pistettä). Suluissa oleva n kertoo ei-tyhjiä vastauksien määrän.

**Taulukko 3 Scratch-kerhon ohjelmointitaitotestin tulokset**

Aihe	Alkukysely keskiarvo	Loppukysely keskiarvo	p-arvo
Ohjelmointi	0,25 (n=5)	0,5 (n=5)	0,157
Muuttujat	0 (n=0)	0,1 (n=1)	0,317
Ehtolauseet	0 (n=0)	0,2 (n=2)	0,157
Toistorakenteet	0,17 (n=2)	0,5 (n=5)	0,083
Funktiot	0 (n=0)	0,1 (n=3)	0,317

Nollahypoteesia ei voida hylätä tässäkin tapauksessa minkään kysymyksen kohdalta. Ei-tyhjiä vastauksia saatiin loppukyselyssä kuitenkin huomattavasti enemmän kuin alkukyselyssä. Alkukyselyssä tyhjiä vastauksia oli yhteensä 53/60, kun 60 on yhteensä kysytyjen kysymysten määrä (5 kysymystä kertaa 12 oppilasta). Loppukyselyssä määrät olivat 34/50. Tyhjiksi vastauksiksi laskettiin myös sellaiset vastaukset kuten ”en tiedä” ja ”en osaa selittää”. Tyhjiä vastauksien määrän laskeminen kertoo siitä, että oppilaille oli jonkinlainen käsitys tai ajatus kysytystä ohjelmointikonseptista, vaikka vastaus saattoikin olla väärä. Kiinnostavaa on se, että toistorakenteisiin osattiin vastata huomattavasti paremmin, kuin esimerkiksi ehtolauseisiin, jotka ovat ohjelmointikonseptina yksinkertaisempia. Tämä voi johtua siitä, että kurssin peleissä käytettiin paljon Scratchin Ikuisesti-looppia, joka voi olla konseptina helppo ymmärtää, sillä siinä ei ole ehtoa lainkaan, se vain toistaa sen sisällä olevaa koodia loputtomiin.

## 4.2 Peliohjelmoinnin alkeet Unitylla -kurssin tulokset

Peliohjelmoinnin alkeet Unitylla -kurssia suoritettiin ensimmäistä kertaa. Kurssi on mitoitettu noin 26-tuntiseksi, ja suorittajille annettiin marraskuu aikaa käydä kurssi läpi. Suorittajat olivat pääasiassa kurssista kiinnostuneita yläasteen ja lukion opettajia eivätkä lukiolaisia ja yläastelaisia, joten he eivät välttämättä edusta todellista kurssin kohderyhmää. Yhteensä osallistujia oli 25 kappaletta, joista 16 aloitti kurssin

suorittamisen. Kuukauden aikana kurssin sai suoritettua vain kaksi osallistujaa ja kolme suorittajaa heidän lisäksi pääsi yli puolen välin.

Kyselyt järjestettiin kurssilla sähköisinä, sillä kyseessä oli verkkokurssi ja osallistajat olivat eri puolella Suomea. Alun perin loppukyselyyn pystyi vastaamaan vasta, kun sai kurssin suoritettua. Tämä johti kuitenkin vain kahteen vastaukseen, joten ylimääräinen loppukysely lähetettiin sähköpostilla kaikille osallistujille, jotka olivat lähteneet suorittamaan kurssia. Vastaus aikaa ylimääräiseen loppukyselyyn oli reilu päivä. Tähän saatiin kolme vastausta lisää, mutta yhden ylimääräiseen loppukyselyyn vastanneen osallistujan alkukyselyä ei löytynyt, joten Wilcoxonin testiä varten otoskooksi saatiin neljä. Ylimääräisessä loppukyselyssä kartoitettiin myös syitä, miksi kurssi jäi kesken, ja oltaisiinko jotain voitu tehdä paremmin, ettei kurssi olisi jäänyt kesken.

#### **4.2.1 Peliohjelmoinnin alkeet Unitylla -kurssin mielipideosio**

Alkukyselyyn vastasi 15 osallistujaa ja loppukyselyyn viisi. Wilcoxonin testi saatiin suoritettua neljälle osallistujalle. Taulukossa 4 on väittämiä ohjelmoinnista ja pelien tekemisestä, vastauksien keskiarvot ja mediaanit ja Wilcoxonin testistä saatu p-arvo. Vastaukset olivat Likert-asteikolla, jossa 1=eri mieltä ja 4=samaa mieltä. Niissä tapauksissa, joissa p-arvoa ei ole annettu, vastaukset olivat täysin samat alku- ja loppukyselyssä.

Tuloksista ei löydy yhtenäistä teemaa, ja nollahypoteesia ei voida hylätä tilastollisesti merkittävällä tasolla. Suurin muutos tapahtui siinä, että ohjelmointi koettiin helpommaksi kurssin jälkeen kuin ennen kurssia. Verrattuna Scratch-kurssiin osallistujat eivät ole yhtä innokkaita työskentelemään pelien tekemisen tai ohjelmoinnin parissa, mutta Unity-kurssilaiset pitävät ohjelmointia huomattavasti hyödyllisempänä taitona.

**Taulukko 4 Peliohjelmoinnin alkeet Unitylla -kurssin mielipideoisio**

Väite	Alkukysely keskiarvo (mediaani)	Loppukysely keskiarvo (mediaani)	p-arvo
Ohjelmointi on minusta kiinnostavaa	3,87 (4)	3,8 (4)	-
Koen ohjelmoinnin hyödylliseksi taidoksi	3,93 (4)	3,8 (4)	-
Pelien tekeminen on minusta kiinnostavaa	3,67 (4)	3,8 (4)	1,0
Koen pelien tekemisen hyödylliseksi taidoksi	3,4 (4)	3,2 (3)	0,317
Voisin nähdä itseni työskentelevän ohjelmoinnin parissa	3,07 (3)	3,2 (3)	1,0
Voisin nähdä itseni työskentelevän pelien tekemisen parissa	2,8 (3)	2,6 (3)	0,157
Koen ohjelmoinnin vaikeaksi	2,33 (2)	1,8 (2)	0,705

#### 4.2.2 Peliohjelmoinnin alkeet Unitylla -kurssin toteutuminen

Loppukyselyyn vastaajat arvioivat myös itse kurssin vaikutusta heidän kiinnostukseensa ohjelmointia kohtaan ja heidän ohjelmointitaitoihinsa. Taulukossa 5 on väitteitä kurssista, kurssin suorittaneiden henkilöiden vastausten keskiarvo ja kurssin kesken jättäneiden henkilöiden keskiarvo. Vastaukset olivat Likert-asteikolla, jossa 1=eri mieltä ja 4=samaa mieltä. Kurssin suorittajia oli kaksi kappaletta ja kesken jättäneitä kolme kappaletta.

Kurssin loppuun asti suorittaneet osallistujat kokivat, että kurssi oli hyödyllisempi ja mielenkiintoisempi, kuin he joilla kurssi jäi kesken. Kurssin kesken jättäneillä kiinnostus ohjelmointia kohtaan jopa hieman laski, ja he ovat eivät ole innokkaita suosittelemaan kurssia muille. Kurssin suorittajat sen sijaan voisivat suositella kurssia muille.

**Taulukko 5 Peliohjelmoinnin alkeet Unitylla -kurssin toteutuminen**

Väite	Kurssin suorittajien keskiarvo	Kurssin kesken jättäneiden keskiarvo
Koen, että ohjelmointitaitoni kehittyivät kurssin aikana	3,5	2,67
Koen kurssin hyödylliseksi	3,5	3,0
Voisin suositella kurssia tutulle	3,5	2,67
Kurssi lisäsi kiinnostustani ohjelmointia kohtaan	3,0	2,33

### 4.2.3 Peliohjelmoinnin alkeet Unitylla -kurssin ohjelmointitaitotesti

Toisin kuin Scratch-kurssin kanssa, Peliohjelmoinnin alkeet Unitylla -kurssille osallistujat vastasivat aktiivisesti ohjelmointitaitotestiin, ja tyhjiä vastauksia tuli vain muutama. Kyselyt järjestettiin kuitenkin sähköisenä, joten on mahdollista, että osallistujat selvittivät oikeat vastaukset kysymyksiin ja saivat siksi vastattua enemmän. Taulukko 6 sisältää ohjelmointitaitotestin kysymyksen aiheen, alkukyselyn tulosten keskiarvon, loppukyselyn tulosten keskiarvon ja Wilcoxonin testistä saadun p-arvon. Yksi vastaajista, joka oli vastannut alkukyselyssä ohjelmointitaitotestiin, jätti vastaamasta loppukyselyn ohjelmointitaitotestiin, joten hänen tyhjät vastauksensa ovat karsittu pois loppukyselyn keskiarvosta. Vastauksia alkukyselyn ohjelmointitaitotestiin saatiin siis 15 ja loppukyselyn ohjelmointitaitotestiin neljä. Koska yhden loppukyselyyn vastanneen osallistujan alkukyselyä ei löytynyt, niin Wilcoxonin testin otoskoko on kolme. Vastaukset arvioitiin joko hyväksytyiksi (1 piste) tai hylätyiksi (0 pistettä). Jos p-arvoa ei ole määritelty, niin alku- ja loppukyselyn tulokset olivat samat.

**Taulukko 6 Peliohjelmoinnin alkeet Unitylla -kurssin ohjelmointitaitotesti**

Aihe	Alkukysely keskiarvo	Loppukysely keskiarvo	p-arvo
Ohjelmointi	0,73	0,8	-
Muuttujat	0,53	1,0	-
Ehtolauseet	0,64	1,0	0,317
Toistorakenteet	0,73	1,0	0,317
Funktiot	0,47	1,0	-



Alkukyselyn tuloksia ei voi verrata loppukyselyn tuloksiin, sillä loppukyselyyn vastanneet saivat pääsääntöisesti kaikki tehtävät oikein jo alkukyselyssä. Pelkästään ohjelmointitaitotestin perusteella ei voida sanoa mitään osallistujien ohjelmointitaitojen kehittymisestä. Alkukyselyssä osallistujien ohjelmointitaidot olivat kuitenkin jo hyvällä tasolla, sillä pääsääntöisesti yli puolet jokaisen kysymyksen vastauksista olivat oikein.

## 5 TULOSTEN TULKINTA

Kummallakaan kurssilla ei saatu tilastollisesti merkittäviä tuloksia ohjelmointitaitojen tai kiinnostuksen ohjelmointia kohtaan muuttumisesta. Tässä luvussa käydään läpi mahdollisia syitä tuloksille.

### 5.1 Scratch-kurssin tulosten tulkinta

Scratch-kurssin ohjelmointitaitotestin avoimiin kysymyksiin tuli paljon vähemmän vastauksia, mitä olisi toivottu ohjelmointitaitojen mittaamisen kannalta. Tämä todennäköisesti johtuu siitä, että ala-asteikäiset eivät ole tottuneet vastaamaan melko haastaviin avoimiin kysymyksiin. Vaikka kysymykset oltiin pyritty muotoilemaan selkeästi, niin ne olivat silti usealle varmasti vaikeita. Parempi tapa kerätä samat tiedot voisi onnistua haastattelun kautta, jossa oppilaan annetaan sanallisesti selittää vastauksensa. Tällöin oppilaalla on mahdollisuus ilmaista ja selittää ajatuksensa, vaikka hän ei saisikaan muotoiltua vastaustaan järkeväksi kirjoitettavaksi lauseeksi. Tämä olisi kuitenkin vaatinut huomattavasti enemmän aikaa, joka ei sopinut kurssin tiukkaan aikatauluun.

Oppilaat kokivat itse, että heidän taitonsa kehittyivät kurssin aikana, ja heidän kiinnostuksensa lisääntyi ohjelmointia kohtaan. Tästä huolimatta kyselyn data ei tue näitä väittämiä ainakaan tilastollisesti merkittävällä tasolla. Jonkin verran nousua näkyy niin ohjelmointitaidoissa kuin kiinnostuksessa, mutta määrät eivät ole samaa luokkaa, mitä oppilaat arvioivat itse. Yksi selitys voi olla, että väittämät olivat huonoja, eivätkä sen takia kuvasta oppilaiden todellista mielenkiintoa ohjelmointia kohtaan. Ohjelmointitaitotestissä on vielä todennäköisemmin kyse tästä ilmiöstä. Toinen selitys ohjelmointikiinnostuksen pieneen nousuun voi olla myös se, että se oli jo valmiiksi niin korkea. Kurssi oli oppilaille vapaaehtoinen ja ylimääräinen normaaliin koulutyöskentelyyn. Tästä syystä kurssille osallistuvat oppilaat olivat jo valmiiksi kiinnostuneita ohjelmoinnista, eikä kurssi juurikaan muuttanut tätä. Tätä tukee kurssille osallistumisen syyt, joista yleisimmät liittyivät peleihin ja pelien tekemiseen, mutta niiden jälkeen yleisin syy oli se, että ohjelmointi on kiinnostavaa.

Mielipideosion lopputuloksiin vaikutti mahdollisesti myös se, että osa oppilaista saattoi tarkoituksellisesti pyrkiä vastaamaan samalla tavalla kuin ensimmäisellä kerralla, sillä väittämät olivat samat. Loppukyselyn aikana yksi oppilaista huudahti, että tähän on sama kysely, kuin aiemmin. Tämä olisi pitänyt varmistaa vielä oppilaille, että vastaavat sen mukaan mitä mieltä ovat asioista nyt.

Avoimien kysymyksien pieneen vastausmäärään todennäköisesti vaikutti myös se, että kyselyt järjestettiin ennen opetuksen aloittamista. Tällöin oppilaille tuli kiire vastata kyselyihin, että he pääsisivät tietokoneelle ja tekemään mahdollisimman nopeasti. Oppilaita ei tietenkään ohjaajien puolesta patistettu vastaamaan nopeasti, mutta oppilaat todennäköisesti tajusivat itse, että mitä nopeammin kysely on valmis, niin sitä nopeammin he pääsevät koneelle. Oppilaille ei kuitenkaan annettu salasanaa tietokoneelle ennen kuin kaikki olivat valmiita, mutta viimeisten kanssa ryhmäpaine saattoi saada heidät lopettamaan vastaamisen, vaikka heillä olisikin ollut jonkinlainen vastaus jo mielessä. Alastelaisista monet eivät ole vielä nopeita kirjottamaan, joten juuri avoimet kysymykset kärsivät tästä kiireestä. Loppukysely oltaisiin voitu laittaa kerho kerran loppuun, mutta silloin riskinä on se, että oppilaat eivät ole tyytyväisiä, että Scratch:in käyttö joudutaan lopettamaan jonkun kyselyn takia. Myöskin lopussa joudutaan silloin varaamaan tietyn mittainen aikamäärä kyselylle. Liian lyhyt aikamäärä estää kyselyyn vastaamisen kokonaan ja liian pitkä aikamäärä on tylsää oppilaille, jotka saavat vastattua kyselyyn nopeasti.

## **5.2 Peliohjelmoinnin alkeet Unitylla -kurssin tulosten tulkinta**

Peliohjelmoinnin alkeet Unitylla -kurssin suorittajien määrä jäi alhaiseksi. Kurssin suoritti vain kaksi osallistujaa 25:stä ilmoittautuneesta. Kyseessä oli kurssin ensimmäinen testiajo, ja osallistujat olivat vapaaehtoisia opettajia, jotka vapaa-ajallaan suorittivat kurssia. Oletuksena oli, ettei suurin osa tule suorittamaan kurssia loppuun asti, mutta suorituksia toivottiin kuitenkin enemmän kuin mitä saatiin.

Kurssille osallistumisen syytä kysyttiin alkukyselyssä monivalinnalla, jossa vastaaja valitsi kaikki ne syyt, joiden takia hän osallistui kurssille. Kurssille osallistumisen yleisin syy oli,

että oltiin kiinnostuneita oppimaan pelien tekemistä (87% vastaajista), ja toiseksi yleisin syy oli, että oltiin kiinnostuneita oppimaan ohjelmointia (80% vastaajista). Osallistujilla olivat siis oikeat syyt osallistua kurssille, mutta tämä kiinnostus pelien tekemistä ja ohjelmointia kohtaan ei riittänyt motivaatioksi kurssin suorittamista varten.

Mielipideosiosta on vaikea tehdä johtopäätöksiä johtuen hyvin pienestä vastausmäärästä. Kaksi henkilöä, jotka suorittivat kurssin loppuun asti, olivat tyytyväisiä, mutta he eivät edusta koko joukkoa. Kesken jättäneet taas eivät olleet tyytyväisiä vaan lähinnä neutraaleja kurssin suhteen. Tämä kuitenkin johtuu todennäköisesti siitä, että kurssi jäi kesken.

Ohjelmointitaitotestissä havaitut erot olivat, että vastaaja sai vastattua yhteen tehtävään enemmän oikein kuin alkukyselyssä, eli niidenkään pohjalta on vaikea sanoa mitään. Osallistajat saivat kuitenkin vastattua alkukyselyssä hyvin ohjelmointitaitotestiin ja pääsääntöisesti yli puolet tehtävien vastauksista oli oikein. Kurssilla piti kuitenkin palauttaa kaksi ohjelmointitehtävää, joissa hyödynnettiin ohjelmointitaitotestissä kysyttäviä periaatteita, eli näiden tehtävien suorittaminen kertoo siitä, että osallistuja osaa ohjelmoida. Kummatkin kurssin suorittajat vastasivat kuitenkin alkukyselyssä ohjelmoineensa aikaisemmin, joten senkään pohjalta ei voida sanoa, että kurssi lisäsi heidän ohjelmointiosaamistaan. Loppukyselyssä suorittajat kuitenkin olivat sitä mieltä, että heidän ohjelmointitaitonsa kehittyivät kurssin aikana. Kurssi siis kehittää osallistujien ohjelmointitaitoja, mutta tämä näkyy vain suorittajien omissa mielipiteissä.

## **6 KURSSIEN KEHITTÄMINEN**

Scratch-kurssista kerättiin palautetta niin kirjallisesti loppukyselyn yhteydessä, kuin myös sanallisesti viimeisellä kerralla. Peliohjelmoinnin alkeet Unitylla -kurssilta kerättiin palautetta kurssin loppukyselyssä, sekä kurssin aikana tapahtuneiden Help-foorumi keskusteluiden kautta.

### **6.1 Scratch-kurssin palaute ja kehittäminen**

Hyödyllistä kirjallista palautetta saatiin vain kahdelta oppilaalta kymmenestä loppukyselyyn vastanneesta. Toinen vastauksista kehui kurssia sanoin ”Todella hyvä”, kun taas toinen toivoi, että kurssi olisi kestänyt pidempään. Sanallisessa keskustelussa tuli myös ilmi, että kurssin oltaisiin toivottu jatkuvan vielä pidempään. Negatiivista sanallista palautetta ei tullut, eikä oppilailta ei myöskään saatu muita kehitysehdotuksia.

Toimin kurssin ohjaajana ja keräsin myös itse havaintoja kurssista ja miten kurssia voisi kehittää. Oppilaat olivat hyvin kiinnostuneita Scratch:in taiteellisesta puolesta. Monet halusivat itse piirtää hahmonsia tai ottaa hahmoksi jonkin kuvan Internetistä. Kurssi oli kuitenkin suunniteltu hyvin tiiviiksi, joten tällaiseen ei ollut aikaa, vaan oppilaita jouduttiin hoputtamaan valitsemaan joku valmiiksi tehty hahmo, että päästään ohjelmoinnissa eteenpäin. Jos kurssille saadaan varattua enemmän aikaa, niin jokaisen kerran lopuksi voitaisiin ylimääräinen aika käyttää siten, että oppilaat saavat vapaasti muokata pelinsä ulkonäköä.

Oppilaat eivät päässeet kurssilla itse ratkaisemaan ongelmia tehtävien muodossa johtuen rajallisesta ajasta. Sen sijaan kurssilla tehtiin kaikki esimerkkiä seuraamalla. Jos kurssille saataisiin enemmän aikaa, voitaisiin kokeilla vastaavaa mallia kuin Franklin et al. (2013) leirillä, jossa oppilaat työskentelivät pareina, ja ratkoivat tehtäviä omatoimisesti, kuitenkin niin, että ohjaajat auttoivat oppilaita tarpeen mukaan. Tämä antaisi todenmukaisemman kuvan siitä, mitä ohjelmointi on oikeasti, ja oppilaat saisivat positiivisia kokemuksia siitä, että saivat itse ratkaistua ongelmia.

## 6.2 Peliohjelmoinnin alkeet Unitylla -kurssin palaute ja kehittäminen

Palautetta kurssista kysyttiin loppukyselyssä sekä ylimääräisessä loppukyselyssä. Yhteensä palautetta saatiin viideltä osallistujalta. Vapaa palaute oli positiivista ja keskittyi yksittäisten ongelmakohtien ympärille ja pieniin kehitysehdotuksiin. Ajan puute oli useammalla syynä kurssin kesken jäämiseksi. Yhteisenä teemana palautteessa oli, että toivottiin enemmän perustehtäviä niin Unitylla kuin myös ohjelmoinnissa. Osa vastaajista koki, että tehtävät etenivät hyvin ja vaikeusaste oli sopiva, kun taas osa koki, että harppaukset olivat liian suuria ja tehtävät vaikeutuivat liian nopeasti. Useampi maininta tuli myös siitä, että kurssin Unity-videot olivat vanhemmalla versiolla kuin nykyinen Unity ja tämä johti ongelmiin. Kurssin videoita oltiin testattu ennen kurssin testiajoa Unityn uusimmalla versiolla, eikä tällaisiin ongelmiin törmätty, joten tämä tuli yllätyksenä.

Ajan puute oli seurausta siitä, että hallinnollisesti kurssi saatiin toimimaan vasta marraskuun alussa, ja se piti saada suoritettua vielä tämän vuoden puolella. Koska osallistujat olivat vapaaehtoisia niin kurssin työmäärä, joka oli monta tuntia viikossa, oli todennäköisesti vain liikaa pyydetty suurimmalta osalta.

Kurssilla ei ole erityisemmin niin sanottuja perustehtäviä. Kurssilla on ylimääräisiä ohjelmointitehtäviä, mutta niiden taso saattaa olla hieman liian korkea. Näitä tehtäviä voisi käydä uudestaan läpi, ja lisätä enemmän aivan perustehtäviä. Esimerkiksi toistorakenteet osiossa on jo tehtävä, jossa pitää for-loopilla tulostaa numerot yhdestä kymmeneen. Seuraava tehtävä voisi siis olla sellainen, että pitää tulostaa luvut viidestä kymmeneen. Kurssin nykyiset haastavammat tehtävät voitaisiin kuitenkin jättää halukkaille vielä suoritettavaksi.

Unitylle toivottiin myös perustehtäviä. Näiden laatiminen on kuitenkin vaikeaa, sillä kurssi lähtee liikkeelle siitä, että osallistuja ei ole koskaan käyttänyt Unitya. Tästä johtuen voi olla hyvin vaikea tehdä tehtäviä, jotka olisivat mahdollista suorittaa osallistujan sen hetkisillä taidoilla. Tällaisten tehtävien pitäisi myös olla hyvin selkeitä, sillä on monta eri tapaa, miten tehtävän voi tehdä väärin. Tästä huolimatta tehtävässä pitäisi kuitenkin saada jotain hienoa tehtyä, että osallistuja haluaa suorittaa sen. Tehtävät voisivat olla muodoltaan

minitutoriaaleja, jossa käydään läpi jokin Unityn ominaisuus tai toiminto, ja tehdään sillä jotain hyödyllistä tai yksinkertainen peli.

Kurssin Unity-videot voitaisiin tehdä uudestaan. Ne tulevat aina olemaan muutaman version jäljessä uusinta Unityn versiota, mutta tällä hetkellä ero saattaa vain olla liian suuri. Yhdessä palautteessa oli myös toivomus, että videotuterialista olisi myös diaesitys muodossa ohjeet. Tämä voitaisiin toteuttaa esimerkiksi tekemällä lista videolla suoritettavista tehtävistä, ja tekemällä videot uusiksi siten, että ne aina kertovat missä kohtaa listaa edetään. Tämän pitäisi selkeyttää mitä videolla tehdään.

## **7 YHTEENVETO**

Tässä työssä tutkittiin kahta eri yliopiston koululaisille järjestämää ohjelmointikurssia, ja miten ne vaikuttivat osallistujan ohjelmointitaitoihin ja kiinnostukseen ohjelmointia kohtaan. Ensimmäinen tutkittava kurssi oli ala-astelaisille järjestetty Scratch-kerho ja toinen tutkittava kurssi oli yläastelaisille ja lukiolaisille järjestettävä itsenäinen verkkokurssi pelien tekemisestä Unitylla ja ohjelmoinnin perusteista.

### **7.1 Scratch-kurssi**

Scrath-kurssi ei onnistunut lisäämään osallistujien kiinnostusta ohjelmointiin tai ohjelmointitaitoja tilastollisesti merkittävällä tasolla. Tästä huolimatta oppilaat itse arvioivat kurssin, että kurssi lisäsi heidän kiinnostustaan paljon, ja heidän ohjelmointitaitonsa kehittyivät kurssin aikana. Kurssi oli kestoltaan niin lyhyt, että oppilaat eivät päässeet itse ratkomaan tehtäviä, vaan seurasivat esimerkkiä, jonka takia tärkeä osa ohjelmointitaitoja kuten ongelmanratkaisu ei välttämättä kehittynyt lainkaan. Tästä huolimatta oppilaat pääsivät ainakin näkemään erilaisia ohjelmointikonsepteja ja niiden toimintaa käytännössä, vaikka suurin osa ei osannutkaan kirjallisesti selittää miten ne toimivat. Tutkimuskysymyksiin 1 ja 2 ei siis saatu tilastollisesti merkittäviä vastauksia, mutta oppilaat itse arvioivat ohjelmointitaitojensa ja kiinnostuksensa ohjelmointia kohtaan lisääntyneen.

Kolmanteen tutkimuskysymyksen, joka käsitteli kurssien kehittämistä, saatiin vastaus. Ottaen huomioon kuinka lyhyt aika kurssilla oli käydä läpi asioita, kurssi toimi varsin hyvin. Kurssia kannattaisi ensisijaisesti kehittää lisäämällä sen kestoja, etenkin jos tavoitteena on vaikuttaa osallistujan ohjelmointitaitoihin, sillä tällä hetkellä kurssi toimii vain lyhyenä esittelynä ohjelmoinnin maailmaan.

### **7.2 Peliohjelmoinnin alkeet Unitylla -kurssi**

Peliohjelmoinnin alkeet Unitylla -kurssin mielipideosiosta ja ohjelmointitaitotestistä ei saatu hyödyllisiä tuloksia johtuen siitä, että loppukyselyyn saatiin liian vähän vastauksia.



Tutkimuskysymyksiin 1 ja 2 ei siis saatu hyviä vastauksia tämän kurssin osalta. Palautteen perusteella voidaan kuitenkin kehittää kurssia ja vastata kolmanteen tutkimuskysymykseen.

Ajan puute oli vain kurssin testiajon ongelma ja jatkossa kurssia järjestetään siten, että se on useamman kuukauden auki, jolloin kurssi pitäisi saada suoritettua vain muutaman tunnin työmäärällä per viikko. Perustehtäviä pitäisi laatia lisää, jotta kurssin vaikeusastetta saadaan laskettua, ja harppaukset eri aiheiden välillä eivät tuntuisi niin suurilta. Kurssin Unity-videot kannattaisi tehdä uudestaan. Ne ovat vanhentuneella versiolla, joka ainakin osalle suorittajista aiheutti ongelmia, ja niiden pitäisi selkeämmin ilmaista mitä tehdään missäkin vaiheessa videota.

Peliohjelmoinnin alkeet Unitylla -kurssia pitäisi mitata vielä oikealla kohdeyleisöllä. Opettajat eivät edusta yläastelaisia ja lukiolaisia, ja kurssin testiajon mittaaminen ei onnistunut kyselyillä. Lukiolaisten tai yläastelaisten mittaaminen saattaa paljastaa uusia kehityskohteita kurssista, tai vaihtoehtoisesti kurssi voi olla heille hyvä sellaisenaan.

## LÄHTEET

Bangdiwala SI. Understanding significance and p-values. 2016. Nepal J Epidemiol. 2016;6(1); 522-524.

Butler, M. & Morgan, M. 2007. Learning challenges faced by novice programming students studying high level and low feedback concepts. In ICT: Providing choices for learners and learning. Proceedings ascilite Singapore 2007.

Carter, L. 2006. Why students with an apparent aptitude for computer science don't choose to major in computer science. In Proceedings of the 37th SIGCSE technical symposium on Computer science education (SIGCSE '06). ACM, New York, NY, USA, pp. 27-31.

Franklin, D., Conrad, P., Boe, B., Nilsen, K., Hill, C., Len, M., Dreschler, G., Aldana, G., Almeida-Tanaka, P., Kiefer, B., Laird, C., Lopez, F., Pham, C., Suarez, J., Waite, R. 2013. Assessment of computer science learning in a scratch-based outreach program. In Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE '13). ACM, New York, NY, USA, pp. 371-376.

Isomöttönen, V., Lakanen, A., Lappalainen, V. 2011. K-12 Game programming course concept using textual programming.

Lakanen A. Isomöttönen V. Lappalainen V. 2012. Life two years after a game programming course: longitudinal viewpoints on k-12 outreach.

Lomerson & Pollacia (2006). Declining CIS Enrollment: An Examination of Pre-College Factors. Information Systems Education Journal, 4 (35).

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y., Laxer, C., Thomas, L., Utting, I., Wilusz, T. 2001. A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. In Working group reports from

ITiCSE on Innovation and technology in computer science education (ITiCSE-WGR '01). ACM, New York, NY, USA, pp. 125-180.

Meerbaum-Salant, O., Armoni M., Ben-Ari, M. 2013. Learning computer science concepts with Scratch, *Computer Science Education*, 23:3, pp. 239-264.

Mojang. What is Minecraft?. [WWW-dokumentti]. [Viitattu 20.11.2018]. Saatavissa: <https://minecraft.net/en-us/what-is-minecraft/>

Opetushallitus. 2014. Perusopetuksen opetussuunnitelman perusteet 2014, pp. 235.

Unity. Unity Personal. [WWW-dokumentti]. [Viitattu: 07.11.2018]. Saatavissa: <https://store.unity.com/products/unity-personal>

Vegso, J. 2005. Interest in CS as a Major Drops Among Incoming Freshmen. *Computing Research News*, Vol. 17/No.3.

Vipunen. Opetushallinnon tilastopalvelu. [WWW-dokumentti]. [Viitattu: 09.12.2018]. Saatavissa: [https://vipunen.fi/fi-fi/\\_layouts/15/xlviewer.aspx?id=/fi-fi/Raportit/Haku-%20ja%20valintatiedot%20-%20korkeakoulu%20-%20yo%20-%20analyysi.xlsb](https://vipunen.fi/fi-fi/_layouts/15/xlviewer.aspx?id=/fi-fi/Raportit/Haku-%20ja%20valintatiedot%20-%20korkeakoulu%20-%20yo%20-%20analyysi.xlsb)

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. 2012. *Experimentation in software engineering*. Springer Science & Business Media.

Zorn, C., Wingrave, C. A., Charbonneau, E., & LaViola Jr, J. J. 2013. Exploring Minecraft as a conduit for increasing interest in programming. In *FDG* (pp. 352-359).