

**Opettajaopiskelijoiden kokemuksia  
pariohjelmointiharjoitukseen vaikuttaneista tekijöistä**  
Veeti Hartikainen

Kasvatustieteen pro gradu -tutkielma  
Monografiamuotoinen  
Kevätlukukausi 2022  
Opettajankoulutuslaitos  
Jyväskylän yliopisto

## TIIVISTELMÄ

**Hartikainen, Veeti. 2022. Opettajaopiskelijoiden kokemuksia pariohjelmointiharjoitukseen vaikuttaneista tekijöistä. Kasvatustieteen pro gradu -tutkielma. Jyväskylän yliopisto. Opettajankoulutuslaitos. 52 sivua.**

Tämän tutkimuksen tarkoituksena oli perehtyä opettajaopiskelijoiden pariohjelmointiharjoitukseen vaikuttaneisiin tekijöihin sekä selvittää sitä, miten ja mitkä tekijät vaikuttivat koettuun oppimiseen. Tutkittava aihe on ajankohtainen, koska opettajien ohjelmoinnillisen osaamisen kehittämistarve on selkeä. Ohjelmointi on sisällytetty Suomessa Perusopetuksen opetussuunnitelmaan vuonna 2016 ja ohjelmoinnin opetusyleistyy kouluissa maailmanlaajuisesti.

Tutkimus toteutettiin osana Oppimisen teknologiat ja pedagogiikka opintojaksoa, jonka yhteydessä opettajaopiskelijoille (N = 8) järjestettiin pariohjelmointiharjoitus. Ennen harjoitusta opiskelijat vastasivat ohjelmointikokemusta kartoittavaan ennakkokyselyyn. Pariohjelmointitilanne videoitiin. Harjoituksen jälkeen opiskelijat kutsuttiin parihaastatteluun ja lisäksi heitä pyydettiin merkitsemään pariohjelmointiharjoitusvideoon kokemansa avainhetket työskentelyn ajalta. Tutkimukseen osallistui 8 opettajaopiskelijaa Jyväskylän yliopistosta ja aineisto kerättiin 2021 joulukuussa. Parihaastattelut analysoitiin sisällönanalyysin keinoin.

Pariohjelmointikokemukseen vaikuttaneet tekijät jaettiin neljään pääteemaan: parin kanssa tehty yhteistyö, parilta ja ohjaajalta saatu tuki, avoimet ja haastavat tehtävät sekä oppimisympäristö. Nämä tekijät jaettiin edelleen alateemoihin.

Pariohjelmointi on tämän ja aiemman tutkimuksen perusteella hyvä tapa ohjelmoinnin harjoittamiseen opettajaopiskelijoille, mutta se vaatii ohjaajalta toteutuksen huolellista valmistelua. Aihetta on kuitenkin syytä tutkia enemmän, jotta opettajien koulutusta voidaan kehittää edelleen vastaamaan selvää tarvetta opettajien ohjelmoinnillisen osaamisen lisäämiselle.

Asiasanat: pariohjelmointi, opettajaopiskelijat, peruskoulu, yhteisöllinen oppiminen, opettajankoulutus

# SISÄLTÖ

## TIIVISTELMÄ

<b>1</b>	<b>JOHDANTO</b> .....	<b>5</b>
<b>2</b>	<b>OHJELMOINTI JA OHJELMOINNILLINEN AJATTELU</b> .....	<b>8</b>
	2.1 Ohjelmointi ja ohjelmoinnillisen ajattelun taidot .....	8
	2.2 Ohjelmoinnillisen ajattelun ja ohjelmoinnin opetus peruskouluissa .....	9
	2.3 Pariiohjelmointi yhteisöllisen oppimisen viitekehyksessä .....	10
	2.4 Scratch ohjelmointiympäristönä .....	13
<b>3</b>	<b>TUTKIMUSTEHTÄVÄ JA TUTKIMUSKYSYMYKSET</b> .....	<b>15</b>
<b>4</b>	<b>TUTKIMUKSEN TOTEUTTAMINEN</b> .....	<b>16</b>
	4.1 Tutkimuskonteksti .....	16
	4.2 Tutkimukseen osallistujat ja tutkimusaineisto .....	16
	4.3 Tutkimusaineiston keruu.....	18
	4.4 Aineiston analyysi .....	23
	4.5 Eettiset ratkaisut.....	27
<b>5</b>	<b>TULOKSET</b> .....	<b>29</b>
	5.1 Pariiohjelmointikokemukseen vaikuttaneita teemoja .....	29
	5.2 Yhteistyö .....	30
	5.3 Tuki .....	34
	5.4 Tehtävät .....	36
	5.5 Oppimisympäristö.....	37
<b>6</b>	<b>POHDINTA</b> .....	<b>39</b>
	6.1 Tulosten tarkastelu ja johtopäätökset.....	39
	6.2 Tutkimuksen arviointi.....	42
	6.3 Jatkotutkimushaasteet .....	44
	<b>LÄHTEET</b> .....	<b>46</b>

<b>LIITTEET</b> .....	<b>50</b>
Liite 1 Ennakkotehtävän kysymykset .....	50
Liite 2 Pariohjelmoinnin reflektointi .....	51
Liite 3 Parihaastattelun runko .....	52

# 1 JOHDANTO

Digitaalinen teknologia on läsnä monipuolisesti yhteiskunnassamme esimerkiksi tietokoneiden, älypuhelinien, Internet-palveluiden ja robotiikan muodossa (Korhonen, 2017) ja kansalaisten tulisi oppia ymmärtämään tieto- ja viestintäteknologian erilaisia sovelluksia ja käyttötarkoituksia (Opetushallitus, 2014). Ohjelmoinnillisen ajattelun voidaan katsoa olevan oleellinen taito kaikille muun muassa näiden digitaalisten teknologioiden ymmärtämistä varten, jotta voimme esimerkiksi nähdä niiden mahdollisuuksia ja rajoitteita (Wing, 2006; Wing, 2011). Ohjelmoinnillista ajattelua voidaan hyödyntää muun muassa ongelmanratkaisussa ja suunnittelussa ohjelmoinnin ulkopuolellakin (Wing, 2006). Vaikka perinteisesti ohjelmointia ja ohjelmoinnillista ajattelua on opiskeltu pääasiassa yliopistossa tietojenkäsittelytieteen alalla, on niihin viime vuosina alettu kiinnittää huomiota entistä enemmän maailmanlaajuisesti jo peruskoulussa niiden alati kasvavan merkityksen myötä (Balanskat & Engelhardt, 2015; Heintz ym., 2016; Wei ym., 2021; Wing, 2006). Suomessa on edetty ohjelmoinnin opettamisessa peruskouluissa ilmiön kärjessä, kun ohjelmointi sisällytettiin perusopetuksen opetussuunnitelmaan 2016 (Opetushallitus, 2014). Kuten muuallakin maailmassa, on Suomessa huomattavissa selvä tarve opettajien koulutukselle ohjelmoinnin ja ohjelmoinnillisen ajattelun saralla (Heintz ym., 2016; Korhonen, 2017). Esimerkiksi Kaarakaisen ym. (2017) tekemän selvityksen mukaan 69 % vastanneista peruskoulun opettajista ei kokenut omaavansa riittäviä taitoja graafisten ohjelmointiympäristöjen käyttöön. Selvityksessä esitetään useita toimenpiteitä digiosaamisen kehittämiseksi, kuten ohjelmoinnin opetukseen tarjottava täydennyskoulutus (Kaarakainen ym., 2017).

Ohjelmointia voisi lähestyä yhteisöllisen oppimisen viitekehyksessä, koska yhteisölliseen oppimiseen liittyy useita etuja esimerkiksi luento-opetukseen verrattuna (Barkley ym., 2014). Yhteisöllisen oppimisen hyötyjä on tutkittu jo

pitkään ja Dillenbourgin (1999) mukaan karkeimmillaan se voidaan määritellä tilanteeksi, jossa kaksi tai useampia henkilöitä oppii tai pyrkii oppimaan jotain yhdessä. Hänen mukaansa yhteisöllistä oppimista edesauttaa tilanne, jossa henkilöt ovat kutakuinkin samalla tasolla tiedoiltaan ja taidoiltaan sekä pystyvät toteuttamaan samankaltaisia tehtäviä, heillä on yhteinen tavoite sekä he työskentelevät yhteisen tehtävän parissa. Ploetznerin ym. (1999) mukaan yhteisöllisessä oppimistilanteessa oppimista voi edistää myös se, että oppijat joutuvat luonnostaan ajattelemaan ääneen ja perustelevaan muille, mitä he ovat tekemässä ja miksi. Ymmärryksen rakentamisen lisäksi yhteisöllisen oppimisen etuja ovat myös mahdollisuus kehittää ryhmätyötaitoja sekä oppia toisten näkökulmista (Barkley ym., 2014)

Yksi luonnollinen tapa ohjelmoinnin harjoitteluun yhteisöllisesti on pariohjelmointi (Bryant ym., 2006). Pariohjelmoinnissa kaksi ohjelmoijaa työskentelee yhdessä yhteisen projektin tai ohjelmointitehtävän ratkaisemiseksi ja työskentely on luonnostaan yhteisöllistä (Bryant ym., 2006; Tsan, 2020). Yleensä parin roolit voidaan jakaa kuskiin ja kartturiin, kuskin käyttäessä tietokonetta ja kartturin suunnitellessa tulevaa ja ratkaisten esiintyviä pulmia. Roolit myös yleensä vaihtuvat työskentelyn aikana useasti (Fagerlund, 2021b; Tsan, 2021). Pariohjelmoinnin hyödyt pohjautuvat pitkälti yhteistyöhön sekä tiedonjakoon koko ohjelmointitehtävän kaikissa vaiheissa (Bryant ym., 2006). Näyttö pariohjelmoinnin eduista itsenäiseen ohjelmointiin verrattuna on kirjavaa ja etenkin kokeneiden ohjelmoijien näkökulmasta formaali, ennalta määrätty pariohjelmointi herättää kritiikkiä (Coman, 2014). Koulussa tehdyissä tutkimuksissa sen sijaan on havaittu viitteitä siitä, että pariohjelmoinnin avulla voitaisiin saavuttaa muun muassa ohjelmoijan korkeampaa luottoa omiin kykyihin (Wei, 2021; Zhong, 2016) sekä vahvempaa ohjelmointiosaamisen kehittymistä verrattuna itsenäiseen ohjelmointiin (Zhong, 2016). Koska opettajien ohjelmointitaidot ovat vähäiset (Karakainen ym., 2017), voisi pariohjelmoinnista olla etua työskentelymuotona ohjelmoinnin harjoittelussa (Hawlitschek ym., 2022). Opettajien koulutuksen osalta pariohjelmointia käsitteleviä tutkimuksia on vielä vähän, ja siksi pariohjelmoinnin soveltuvuutta

tulisi tutkia opettajien koulutuksen näkökulmasta. Opettajien ohjelmointiosaamisen kehittäminen on aiheena ajankohtainen ohjelmoinnin ja ohjelmoinnillisen ajattelun opetuksen lisääntyessä kouluissa maailmanlaajuisesti (Balanskat & Engelhardt, 2015; Heintz ym., 2016).

Tämän tutkimuksen tavoitteena on opettajien koulutuksen kehittäminen ohjelmoinnin osa-alueella. Tässä tapaustutkimuksessa ohjelmoinnin opettamista opettajaopiskelijoille lähestytään pariohjelmoinnin kautta. Aiempi tutkimus on osoittanut, että pariohjelmointi edellyttää suunnitelmallisuutta. Huomiota tulee kiinnittää muun muassa yhteistyötä edellyttävien tehtävien rakentamiseen (Hawlitschek ym., 2022), parien muodostamiseen (Zhong ym., 2016) sekä parien aiempaan ohjelmointiosaamiseen (Campe ym., 2020).

## 2 OHJELMOINTI JA OHJELMOINNILLINEN AJATTELU

### 2.1 Ohjelmointi ja ohjelmoinnillisen ajattelun taidot

Ohjelmoinnilla voidaan yleisesti tarkoittaa ohjeiden syöttämistä tietokoneeseen muodossa, jonka tietokone pystyy käsittelemään (Blackwell, 2002). Tarkemmin ohjelmoinnin voidaan käsittää tarkoittavan ohjelmistojen suunnittelua sekä luomista, jotka toteuttavat automaattisesti prosesseja ohjelmissa (Denning & Tedre, 2019). Ohjelmointiin liittyy keskeisesti ohjelmoinnillisen ajattelun käsite. Ohjelmoinnillinen ajattelu on hyvin moniulotteinen käsite, joka Wingin (2011) mukaan tarkoittaa sitä ajatusprosessia, jossa ongelmat ja ratkaisut muunnetaan sellaiseen muotoon, jonka tietokone, ihminen tai nämä yhdessä voivat toteuttaa. Aho (2012) taas painottaa ohjelmoinnillisessa ajattelussa sen algoritmisuutta ja laskennallisuutta. Ohjelmoinnillisen ajattelun peruseriaatteiden ymmärtäminen on 2000-luvulla tärkeää, ja sitä voidaan soveltaa myös arkipäiväisiin ongelmiin (Wing, 2006; Wing, 2011). Näitä peruseriaatteita opetuksen näkökulmasta ovat Fagerlundin (2021b) mukaan ongelman abstrahointi ja käsillä olevan ongelman tai tehtävän purkaminen pienempiin, olennaisiin osiin, jolloin ongelmat voidaan ratkaista tehokkaasti käytössä olevan datan perusteella ja ratkaisumalleja voidaan yleistää vastaaviin tilanteisiin tulevaisuudessa. Peruseriaatteisiin kuuluu Fagerlundin (2021b) mukaan myös luovuus, logiikka ja yhteistyö ongelmia ratkoessa sekä toteutusta suunnitellessa. Kaikki tämä vaatii koordinaatiota, ongelman kannalta oleellisten asioiden ymmärtämistä sekä ratkaisuiden arvioimista tavoitteiden mukaisesti (Fagerlund, 2021b; Wing, 2006). Ohjelmointi on keino opetella näitä ohjelmoinnillisen ajattelun taitoja, joita taas voidaan soveltaa ohjelmoinnin ulkopuolella, esimerkiksi arkisissa ongelmanratkaisutilanteissa (Fagerlund 2021a; García-Penalvo & Mendes, 2017; Wing, 2006).

Monet tämän päivän oppilaista tulevat tulevaisuudessa olemaan mukana uusien teknologioiden kehittämisessä (Balanskat & Engelhardt, 2015), mutta sen



lisäksi ohjelmointi voidaan nähdä oppimisen kohteena ja lisäksi oppimisen välineenä monipuolisten ja nyky-yhteiskunnassa erittäin oleellisten taitojen oppimisessa. Digitalisoituneessa yhteiskunnassa tarvitaan yhä enemmän niin sanottuja 2000-luvun taitoja: vuorovaikutus- ja yhteistyötaitoja sekä ongelmanratkaisukykyä (OECD 2017; Trilling & Fadel, 2009). Ongelmanratkaisukykyyn Trillingin ja Fadelin (2009) mukaan kuuluu olennaisesti myös kyky esittää ratkaisun kannalta olennaisia kysymyksiä, jotka taas johtavat parempiin ratkaisuihin.

## **2.2 Ohjelmoinnillisen ajattelun ja ohjelmoinnin opetus peruskouluissa**

Ohjelmoinnin ja ohjelmoinnillisten taitojen yhä kasvavan merkityksen myötä niiden opetus on saanut laajalti jalansijaa koulujärjestelmissä ympäri maailman (Heintz ym., 2016). Esimerkiksi 2014 European Schoolnetin julkaisemasta raportista käy ilmi, että osallistuneista 21 opetusministeriöstä Euroopassa ja Israelissa valtaosa integroi tai suunnittelee lähitulevaisuudessa sisällyttävänsä ohjelmoinnin ja ohjelmoinnillisen ajattelun opetusta opetussuunnitelmiinsa. Lähestymistavat ohjelmoinnin sisällyttämiseen tosin erosivat jonkin verran maiden välillä, osan integroidessa ohjelmointia opetukseen ja osan nostaessa sitä esiin omaksi tavoitteekseen. (Balanskat & Engelhardt, 2015.) Myös Heintz ym. (2016) tekemä selvitys tukee ohjelmoinnin opetuksen yleistymistä maailmanlaajuisesti peruskouluissa. Ohjelmoinnillisen ajattelun taitoja mainitaan harvoin opetussuunnitelmissa erillisenä tavoitteena, mutta usein sen peruseriaatteet tulevat ilmi jossain muodossa (Heintz ym., 2016).

Suomessa ohjelmointi ja ohjelmoinnillinen ajattelu tuli laajasti osaksi Perusopetuksen opetussuunnitelmaa 2016 (OPH, 2014). Opetussuunnitelmassa ohjelmointi mainitaan vuosiluokilla 1–2 matematiikan tavoitteisiin ja vuosiluokilla 3–9 matematiikkaan ja käsitöihin, jonka lisäksi ohjelmointi sisältyy tieto- ja viestintäteknologisen osaamisen laaja-alaiseen tavoitteeseen (Heintz ym., 2016; OPH, 2014). Suomessa opettajalla on melko vapaat kädet sen suhteen, miten hän toteuttaa ohjelmoinnin opetuksen peruskoulussa (Lindberg ym.,

2019). Kuitenkin selvityksen mukaan vuonna 2017 peruskoulun opettajien ja oppilaiden ohjelmointitaidot olivat heikot, eikä esimerkiksi 69 % vastaajista kokenut osaavansa käyttää graafisia ohjelmointiympäristöjä riittävän hyvin (Kaarakainen ym., 2017).

Kuten muuallakin maailmassa, Suomessa on siis selvä tarve kehittää opettajien ohjelmointiosaamista ja valmiuksia opettaa ohjelmointia (Heintz ym. 2016; Kaarakainen ym., 2017; Korhonen, 2017). Opettajat tarvitsevat kokonaisvaltaista koulutusta ja mahdollisuuden luoda riittävä perustaso ohjelmoinnin ja ohjelmoinnillisten taitojen opettamista varten (Hsu ym., 2018). Kaarakainen ym. (2017) esittävät selvityksessään useita toimenpiteitä opettajien laajemman digiosaamisen kehittämiseksi muun muassa koulun digistrategioiden ja toimintakulttuurien kautta, mutta painottavat myös riittävää täydennyskoulutusta opettajille opetussuunnitelman edellyttämään ohjelmoinnin opetukseen.

Toinen varteenotettava keino voisi olla ohjelmoinnin sisällyttäminen jo opettajankoulutukseen ja tarjota näin kokemuksia ohjelmoinnista opettajankoulutuksen perusopintojen aikana (Hawlitschek ym., 2022). Ohjelmoinnin opettamisesta opettajaopiskelijoille erityisesti pariohjelmoinnin keinoin käsittelevää tutkimusta on toistaiseksi vielä vähän, joten tämä tutkimus pyrkii syventymään pariohjelmointiin opettajankoulutuksen kontekstissa, kartoittamaan siihen vaikuttavia tekijöitä, jotka tulisi ottaa huomioon tulevien opettajien opetusta suunnitellessa ja toteutettaessa.

### **2.3 Pariohjelmointi yhteisöllisen oppimisen viitekehyksessä**

Pariohjelmointi on yksi työskentelytapa ohjelmoinnin opettelemiseen ja sitä on käytetty perinteisesti esimerkiksi ohjelmoinnin opettamisessa yliopistossa (Hawlitschek ym., 2022). Pariohjelmoinnissa kaksi ohjelmoijaa työskentelee yhdellä tietokoneella työstäen yhdessä samaa ohjelmointitehtävää, kummankin toimiessa välillä kuskina ja kartturina (Tsan ym., 2020). Kuskin tehtävänä on käyttää näppäimistöä ja hiirtä kartturin suunnitellessa toteutusta ja etsiessä

mahdollisia virheitä (Tsan ym., 2021; Tsan ym., 2020). Näin selkeää roolijakoa on kuitenkin kritisoitu, ja usein roolit vaihtuvatkin lennosta, painottuen kuitenkin jokseenkin epätasaisesti parin osapuolien välille (Fagerlund, 2021b; Höfer, 2008). Pariohjelmoinnin hyödyt perustuvat luonnolliseen yhteistoimintaan ja tiedonjakoon ohjelmointitehtävien kaikissa vaiheissa roolijaosta riippumatta ja on luonnostaan hyvin yhteisöllistä (Bryant ym., 2006). Roolijakoon tulee kuitenkin tästä huolimatta kiinnittää huomiota ja osallistujia tulee ohjeistaa ja kannustaa huolehtimaan riittävästä roolien vaihtamisesta työskentelyn lomassa, jotta parin toinen osapuoli ei jää tehtävän ulkopuolelle (Hawlitschek ym., 2022).

Ohjelmointia opetellessa pariohjelmointiin liittyy useita hyötyjä työskentelymuotona itsenäiseen ohjelmoinnin harjoitteluun verrattuna (Hawlitschek ym., 2022). Yleisesti yhteisölliseen oppimiseen liittyy mahdollisuus yhteiseen sitoutumiseen tehtävään ja työskentelyyn. Tämän on aiempi tutkimus tunnistanut vahvistavan myös opiskelijoiden oppimismotivaatiota (Lavy, 2017). Onnistuessaan pariohjelmoinnissa pari käy tuottoisaa keskustelua ratkaistessaan ongelmia (Tsan ym., 2021). Kyky kommunikoida, asettaa tavoitteita ja suunnitella toimintaa yhdessä muiden kanssa ovat keskeisessä asemassa digitaalisia teknologioita hyödyntävässä ongelmanratkaisussa sekä laajemmin nyky-yhteiskunnassa (OECD, 2017).

Pariohjelmoinnin on osoitettu olevan toimiva tapa ohjelmoinnillisen ajattelun ja ohjelmoinnin opettelussa (Hawlitschek ym., 2022). Esimerkiksi Wein ym. (2021) tutkimuksessa osittainen pariohjelmointi oli yhteydessä itsenäisesti ohjelmoinutta vertailuryhmää korkeampaan ohjelmoinnillisen ajattelun taitojen kehitykseen, joskin tutkittavien ja parien sukupuolijakaumalla oli jonkin verran yhteyttä tapahtuneeseen kehittymiseen. Kyseisessä Wein ym. (2021) tutkimuksessa pariohjelmoinnin myötä poikien kokema luottamus omiin ohjelmointitaitoihin nousi myös tilastollisesti merkittävästi, tosin tytöillä samanlaista tilastollisesti merkitsevää kehitystä ei havaittu. Toisaalta esimerkiksi Zhong ym. (2016) toteuttamassa tutkimuksessa tyttöjen itseluottamus ja ohjelmointiosaaminen kehittyi poikia enemmän pariohjelmoitaessa verrattuna itsenäiseen ohjelmointiin. Myös niiden pariohjelmoijien, joilla on vähän aiempaa

ohjelmointiosaamista, on osoitettu hyötyvän eniten pariohjelmoinnista lähestymistapana ohjelmoinnin opetukseen (Hawlitschek, ym., 2022).

Pariohjelmointi ei kuitenkaan ole aina paras ratkaisu ohjelmoinnin harjoitteluun ja sen toimivuus on tunnistettu melko tilannesidonnaiseksi (Hawlitschek ym., 2022). Comanin ym. (2014) mukaan pariohjelmointia tulisi käyttää työskentelymuotona itsenäisen ohjelmoinnin lisäksi esimerkiksi tilanteissa, joissa ohjelmoija jää jumiin jonkin yllättävän ongelman kanssa. On kuitenkin huomattava, että kyseinen Comanin ym. (2014) tutkimus käsittelee pariohjelmointia kokeneiden, ohjelmoinnin parissa työskentelevien tiimien ja henkilöiden näkökulmasta. Tämä ei kuitenkaan ole ristiriidassa Hawlitschekin ym. (2022) näkemyksen kanssa siitä, että pariohjelmointi on toimivimmillaan kokemattomien ohjelmoijien kanssa. Kuitenkin Coman ym. (2014) nostavat esiin oleellisen huomion siitä, että pariohjelmointi ei voi olla ainoa lähestymistapa ohjelmointiin ja jossain tapauksissa on tehokkaampaa työskennellä itsenäisesti.

Vaikka Bryant ym. (2006) toteuttaman tutkimuksen mukaan pariohjelmointi on luonnostaan hyvin yhteisöllistä, vaihtelee yhteistyön määrä ja laatu tehtävän mukaan. Yhteistyön määrä oli heidän mukaansa korkeimmillaan haastavissa ohjelmointitehtävissä, mikä taas näyttäisi viittaavan pariohjelmoinnin olleen tällöin hyödyllisintä. Myös Hawlitschek ym. (2022) mukaan pariohjelmointi vaatii riittävän haastavia ja laajoja tehtäviä. Toisaalta kokemattomat ohjelmoijat tarvitsevat tukea ja ohjausta ymmärtääkseen ohjelmoinnin peruserätykset (Mayerin & Johnsonsonin, 2010). Tehtävät kannattaakin siksi jakaa esimerkiksi kolmivaiheisen Use-Modify-Create -mallin mukaan (Franklin ym., 2020). Mallissa ohjelmoinnin taso kehittyy Use -vaiheen ohjelmointiympäristön ohjeistetusta käyttämisestä Modify -tasolle, jossa ohjelmoija muokkaa valmista koodia esimerkiksi korjaamalla virheitä. Kolmannessa Create -vaiheessa ohjelmoija luo itse jotain, esimerkiksi pelin. Fagerlundin ym. (2020) mukaan mallin Use ja Modify vaiheet voivat lisätä ohjelmointiosaamista ja vastavuoroisesti pelkkä uuden luominen saattaa aiheuttaa liian monimutkaisen tavoitteen ilman riittävää aiempaa osaamista. Myös Franklinin ym. (2020) mukaan kyseinen malli tarjoaa sopivan tasapainon

strukturoidumman ja avoimen oppimisen välillä, vahvistaa aiempaa osaamista ja kannustaa kokeilemaan uusia lohkoja sekä ohjelmoimaan luovasti.

Hawlitschekin ym. (2022) mukaan pariohjelmointiharjoitusta järjestettäessä tulisi opettajan kiinnittää huomiota osallistujien aiempaan osaamiseen, pyrkiä asettamaan pareihin ohjelmointitaidoiltaan samankaltaisia opiskelijoita, huolehtia siitä, että parin kumpikin osapuoli osallistuu työskentelyyn tasapuolisesti sekä ohjeistaa ja kannustaa merkitykselliseen yhteistyöhön ja vuorovaikutukseen. Myös Campen ym. (2020) havainnot parien muodostamisesta tukevat osaltaan taitotasoltaan samankaltaisten ohjelmoijien yhdistämistä pareiksi. Aiemman osaamisen lisäksi parien keskinäinen suhde ja yhteensopivuus saattavat vaikuttaa myös pariohjelmoinnin onnistumiseen (Zhong ym. 2016). Pariohjelmointi vaatii siis ohjaajalta huomiota parien muodostamiseen (Campe ym., 2020; Zhong ym., 2016) sekä huolellisesti suunniteltujen tehtävien valmistelua, joissa yhteistyö on hyödyllistä opiskelijoille (Hawlitschek ym., 2022).

## 2.4 Scratch ohjelmointiympäristönä

Peruskoulussa ohjelmoinnissa käytettävät ohjelmointikielien ja -ympäristöt vaihtelevat luonnollisesti oppilaiden ikätason mukaisesti. Useissa aloittelijoille suunnatuissa ohjelmointiympäristöissä ohjelmoidaan komentojen kirjoittamisen sijaan lohkojen avulla, jolloin jokainen lohko on ohjelmointikielen osa (Ouahbi ym., 2014). Kaarakaisen ym. (2017) toteuttaman Oppika-kyselyn mukaan toisen luokka-asteen oppilaista 46 % oli kokeillut koodaamista tai robottien ohjaamista koulussa ja 31 % kotona. Viidesluokkalaisista 45 % kertoi käyttäneensä koulussa graafista ohjelmointiympäristöä, kuten Scratchia. Kahdeksaluokkalaista 65 % ei Oppika-tutkimusta tehdessä ollut käyttänyt mitään ohjelmointiympäristöä koulussa. Toisaalta kahdeksaluokkalaisissa oli myös oppilaita, jotka olivat graafisten ohjelmointiympäristöjen lisäksi käyttäneet muun muassa verkkosivujen merkkakieliä (10 %) sekä muita ohjelmointiympäristöjä ja kieliä.

Koska tähän tutkimukseen osallistui erityisesti luokanopettajaopiskelijoita, oli järkevää valita ohjelmointiympäristöksi myös alakouluissa suosituin graafinen ohjelmointiympäristö Scratch (Hsu ym., 2018; Kaarakainen, 2017). Scratch on Scratch-säätiön kehittämä ja ylläpitämä ilmainen selainpohjainen ohjelmointiympäristö, jossa käyttäjä pystyy ohjelmoimaan esimerkiksi pelejä, animaatioita tai interaktiivisia tarinoita (Scratch, 2022). Scratchissa ohjelmointi tapahtuu yhdistämällä värikkäitä komentolohkoja, jotka taas ohjaavat hahmojen toimintaa (Scratch, 2022; Maloney ym., 2010). Scratchin keskeisenä tavoitteena on esitellä ja mahdollistaa ohjelmointia 8–16-vuotiaille lapsille ja nuorille, joilla ei ole aiempaa ohjelmointikokemusta (Maloney ym., 2010; Scratch, 2022). Maloney ym. (2010) mukaan Scratchissa ohjelmoinnista on tehty helposti lähestyttävää käyttäjälle esimerkiksi visuaalisilla avusteilla lohkoja siirrettäessä sekä eliminoimalla syntaksivirheet lohkoilla. Maloney ym. (2010) myös kuvaavat Scratchia ympäristönä ja ohjelmointikielenä, joka on nopea oppia, mutta toisaalta samaan aikaan riittävän syvälinen ja monipuolinen jotta samaa ympäristöä voi käyttää pidempäänkin. Fagerlundin (2021a) mukaan Scratchissa toteutettavissa projekteissa pystytään toteuttamaan kaikkia ohjelmoinnillisen ajattelun osa-alueita.

### 3 TUTKIMUSTEHTÄVÄ JA TUTKIMUSKYSYMYKSET

Ohjelmoinnin ja ohjelmoinnillisen ajattelun taitojen opetus perusopetuksessa on yhä merkittävämmässä asemassa maailmanlaajuisesti ja ohjelmoinnin opetuksen aseman voidaan odottaa vain kasvavan tulevaisuudessa (Balanskat & Engelhardt, 2015). Samalla maailmanlaajuisesti on haasteena kouluttaa opettajia omaamaan riittävät taidot ohjelmoinnin ja ohjelmoinnillisen ajattelun opettamiseen (Hsu ym., 2018). Tämän vuoksi opettajille on annettava mahdollisuuksia kehittää omaa osaamistaan ohjelmoinnin opettamiseen. Nykyisten opettajien täydennyskoulutuksen lisäksi yksi hyvä mahdollisuus on antaa opettajaopiskelijoille ohjelmointikokemuksia ja ohjausta ohjelmointiin jo opettajan perusopinnojen aikana. Yksi hyvä lähestymistapa kokemattomien ohjelmoijien kanssa on pariohjelmointi (Bryant ym., 2006; Hawlitschek ym., 2022). Aiheesta on kuitenkin julkaistu verrattain vähän tutkimusta ja siksi tässä tapaustutkimuksessa pyritään perehtymään opiskelijoiden pariohjelmointiharjoitukseen vaikuttaneisiin tekijöihin. Lisäksi tällä tutkimuksella pyritään etsimään tietoa siitä, millaiset tekijät ovat hyödyllisiä tai haastavia oppimiselle pariohjelmoinnin aikana. Tutkimuksen tutkimuskysymykset muokkaantuivat tutkimuksen edetessä ja lopulta tarkentuivat seuraaviksi:

1. Millaisten tekijät opiskelijat kokivat vaikuttavan pariohjelmointiharjoitukseen?
2. Mitkä tekijät opiskelijat kokivat hyödyllisiksi tai haastaviksi oppimiselle pariohjelmoinnin aikana?

## 4 TUTKIMUKSEN TOTEUTTAMINEN

### 4.1 Tutkimuskonteksti

Tutkimus toteutettiin Jyväskylän yliopiston Opettajankoulutuslaitoksen valinnaisen Oppimisen teknologiat ja pedagogiikka -opintokokonaisuuden kursseilla. Kyseisen kurssin nimi oli: Ohjelmoinnillista ajattelua perusopetukseen. Kyseessä on tapaustutkimus (Hirsjärvi ym., 2009), jossa opiskelijoille tarjottiin pariohjelmointikokemus ja havainnoitiin opiskelijoiden toimintaa pariohjelmointiharjoitusten aikana ja selvitettiin, millainen pariohjelmointikokemus heidän mielestään oli. Tutkimusstrategiana tapaustutkimus soveltui tässä tapauksessa lähestymistavaksi hyvin, koska pariohjelmointiharjoituksesta haluttiin saada yksityiskohtaista, kuvailevaa tietoa (Hirsjärvi ym., 2009; Patton, 2015;).

Opettajien ohjelmointiosaamisen kehittämistarve on selkeä. Tämän vuoksi tässä tutkimuksessa oli tavoitteena perehtyä opiskelijoiden pariohjelmointiharjoitukseen vaikuttaneisiin tekijöihin. Lisäksi tavoitteena oli selvittää sitä, miten ja mitkä tekijät vaikuttivat koettuun oppimiseen. Tutkittava aihe on ajankohtainen, koska ohjelmoinnin aseman kouluissa voidaan odottaa kasvavan tulevaisuudessa (Balanskat & Engelhardt, 2015).

### 4.2 Tutkimukseen osallistujat ja tutkimusaineisto

Tutkimukseen osallistui yhteensä kahdeksan opiskelijaa (N=8). Suurin osa osallistujista oli luokanopettajaopiskelijoita (n = 6), mutta mukana oli myös kaksi aineenopettajaopiskelijaa. Tapaustutkimukselle tyypillisesti tutkimuksessa kerättiin aineistoa useita metodeja käyttäen (Hirsjärvi ym., 2009). Tutkimusaineisto koostui ennakkokyselystä, video-observoinnista, haastatteluaineistosta sekä opiskelijoiden tuottamista avainhetki-muistiinpanoista. Tässä tutkimuksessa hyödynnettiin ennakkokyselyä, haastatteluaineistoa ja opiskelijoiden avainhetki-muistiinpanoja.



Ennakkokyselyn perusteella voidaan todeta, että osallistujien aiempi ohjelmointikokemus oli yleisesti ottaen vähäistä. Ennakkokyselyssä kartoitettiin omien ohjelmointitaitojen ja kokemuksen lisäksi osallistujien käsityksiä ohjelmoinnista yleisesti, ohjelmoinnin merkitystä perusopetuksessa, käsityksiä pariohjelmoinnista, ohjelmoinnin ja pariohjelmoinnin herättämiä tunteita sekä käsityksiä itsestä ohjelmoinnin opettajana. Ennakkokyselyn kysymykset ovat tämän raportin liitteissä (LIITE 1). Koska kysely toteutettiin kurssin ennakkotehtävänä, vastasi siihen myös kaksi tämän tutkimuksen ulkopuolista opiskelijaa. Näitä vastauksia ei tässä tutkimuksessa otettu huomioon.

Osallistujat jaettiin ohjelmointikerralla pareihin ja nämä parit numeroitiin 1–4 aineiston käsittelyn helpottamiseksi. Näitä parinnumeroita käytettiin näyttötallenteiden sekä parien työskentelyä tallentaneiden kameroiden tiedostoja nimettäessä. Osallistujia myös haastateltiin viikko pariohjelmointikerran jälkeen näissä samoissa pareissa. Haastattelut kestivät 25–30 minuuttia. Haastattelut toteutettiin kurssin opetuskerralla etänä ja niistä tallennettiin äänitiedostot.

Ennakkokyselyt kerättiin aluksi vastaajien nimikirjaimilla, kunnes ne yhdistettiin litteroituihin haastatteluihin avainlistan avulla. Yhdistettäessä ennakkokyselyitä haastatteluihin niistä muunnettiin nimikirjaimet tunnistekoodeiksi. Täten kirjallisessa aineistossa ei enää ollut mitään henkilötietoja.

Yhden parin yhdistetyn ennakkokyselyn ja litteroidun parihaastattelun laajuus vaihteli kahdentoista ja neljäntoista sivun välillä (fonttikoko 11, riviväli 1,5). Avainhetkiä (taulukko 1) koskevat muistiinpanot olivat yleisesti melko suppeat, johtuen todennäköisesti suurelta osin niiden täyttämiseen käytettävissä olleen ajan rajallisuudesta. Muistiinpanoihin osallistujat merkitsivät pariohjelmointiharjoituksesta lomakkeen (LIITE 2) avulla hetkiä, jotka tukivat toimintaa tai haastoi tai sai aikaan ristiriitaa (taulukko 1).

## Taulukko 1

*Tutkittavien kirjaamien avainhetkien määrät*

ID	A: Hetki, joka tuki tai oli hyödyksi, <i>f</i>	B: Hetki, joka haastoi tai sai aikaan ristiriitaa, <i>f</i>
H3	3	1
H4	2	3
H5	4	1
H6	3	1
H7	1	1
H8	4	2
	KA 2,83	KA 1,5
	KH 1,07	KH 0,76

*Huom.* H1 ja H2 eivät palauttaneet avainhetkilomakkeita. KA = keskiarvo, KH = keskihajonta, ID = tunniste.

Avainhetkien merkitsemisen tukena opiskelijat käyttivät pariohjelmointikerralla kuvattuja näytöntallenteita sekä videokuvaa. Avainhetkien keräämistä on kuvattu tarkemmin luvussa 4.3. Videomuodossa aineistoa kerättiin tutkimuksessa yhteensä noin 18 tuntia. Tutkimuksessa hyödynnetty aineisto rajattiin tutkimustavoitteen kannalta oleellisiin osiin, jonka vuoksi tässä pro gradu -tutkimuksessa ei hyödynnetä kerättyä videoaineistoa.

### 4.3 Tutkimusaineiston keruu

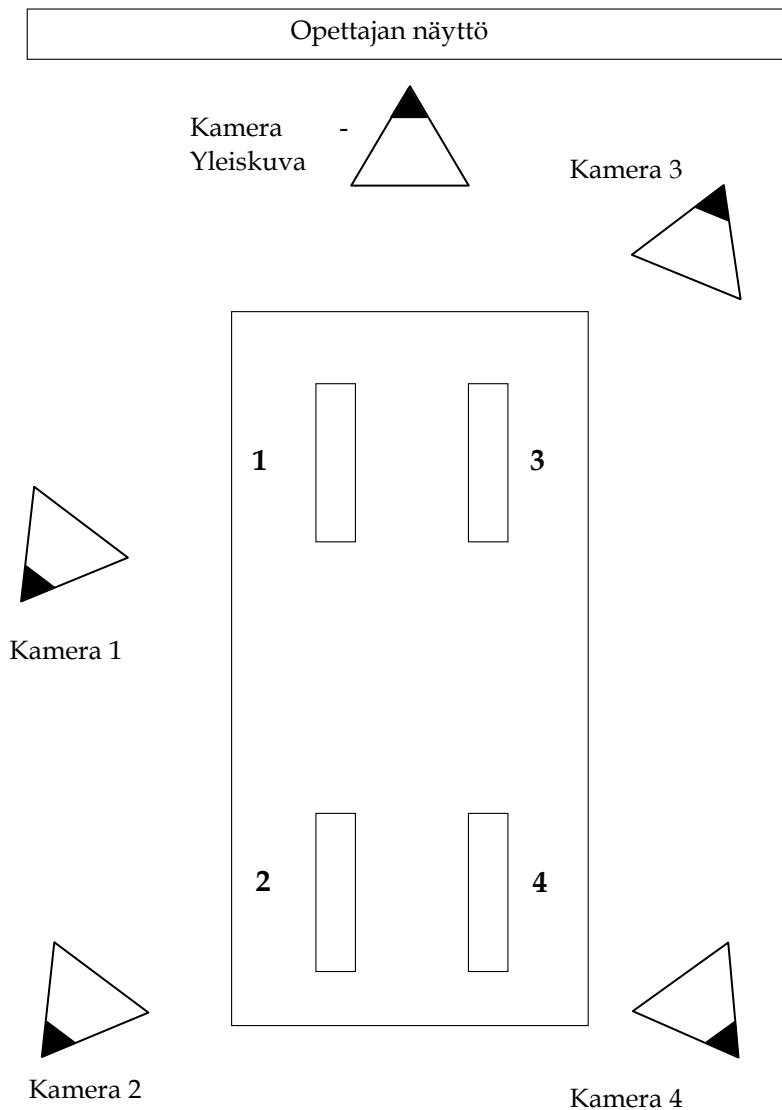
Tutkimusaineisto kerättiin marras-joulukuussa 2021. Aineiston kerääminen aloitettiin osallistujien ennakkokyselyllä. Ennakkokyselyssä kartoitettiin osallistujien aiempaa ohjelmointi- ja pariohjelmointikokemusta, sekä käsityksiä ohjelmoinnin ohjaamisesta. Kyselyyn vastattiin Webropol -palvelussa. Ennakkokyselyllä saatiin kerättyä tietoa opiskelijoiden aiemmasta ohjelmointikokemuksesta. Tämän tiedon ansiosta kaksi kokeneempaa ohjelmoijaa voitiin laittaa työskentelemään parina ja muiden osallistujien tiedettiin olevan oman kuvailunsa mukaan samalla viivalla

ohjelmointikokemuksen suhteen. Samalla kerättiin osallistujilta tarvittavat luvat aineiston keräämiseen, sekä informoitiin heitä aineiston säilyttämisestä ja käsittelystä.

Kaksi viikkoa ennakkokyselyn jälkeen osallistujille järjestettiin pariohjelmointiharjoitus. Tämän aikana vastaajat suorittivat Scratch-ympäristössä kolme erilaista ohjelmointitehtävää. Tehtävät etenivät niin sanotun Use-Modify-Create mallin (Franklin ym., 2020) mukaisesti. Ennen ohjelmoinnin aloittamista osallistujia muistutettiin osallistumisen vapaaehtoisuudesta ja heiltä kerättiin allekirjoitetut tutkimusluvut. Ennakkokyselyillä kartoitetun aiemman osaamisen perusteella kahta enemmän kokemusta omaavaa osallistujaa ehdotettiin muodostamaan pari, mutta muuten osallistujat muodostivat parit vapaasti. Kaikki parit työskentelivät samassa tilassa ja kaikilla pareilla oli käytössään yksi tietokone, näyttö, hiiri ja näppäimistö. Jokaisen parin työskentelyä tallennettiin GoPro -kameralla sekä näytönkaappauksella IRIS Connectin avulla. Lisäksi koko tilasta tallennettiin yleiskuvaa yhdellä kameralla. Parit työskentelivät koko harjoituksen samoilla tietokoneilla yhtä paria lukuun ottamatta, joka vaihtoi toisen tehtävän jälkeen tietokonetta näytönkaappaukseen liittyneen ongelman takia. Lisäksi samassa tilassa oli kolme ohjaajaa, jotka vastasivat kuvauksesta sekä tarpeen mukaan ohjasivat opiskelijoita. Tehtävien ohjeistukset antanut ohjaaja sen sijaan toteutti opetuksen etäyhteyksien välityksellä. Ohjaajan näyttö sijaitsi luokan etuosassa, josta jokainen pystyi seuraamaan ohjeistusta. Tehtävissä ohjaajat auttoivat osallistujia vain, mikäli he erikseen kysyivät apua. Ohjelmointikerran järjestelyjä on havainnollistettu kuviossa 1.

**Kuvio 1.**

*Havainnollistaminen pariohjelmoinnin järjestelyistä. Numerot vastaavat parien numeroita*



Ensimmäisessä tehtävässä osallistujat ohjelmoivat aluksi opettajan antamien ohjeiden mukaisesti pareissa. Opettajan ohjeiden mukaan harjoiteltiin lohkojen lisääminen sekä taustan vaihtaminen. Tämän jälkeen parit seurasivat itsenäisesti Innokas-verkoston luomaa pelinkehitysohjetta (Ilola & Pukkila, 2019). Kaikki parit toteuttivat samalla periaatteella toimivan yksinkertaisen sokkelopelin, jossa pelihahmo liikkui nuolinäppäimillä ja palasi alkuun, mikäli

hahmo osui labyrintin seiniin. Ensimmäinen tehtävä kesti kokonaisuudessaan noin 30 minuuttia.

Parien toinen tehtävä oli valita ja suorittaa debugging-harjoituksia. Tehtävissä parien tuli etsiä ja korjata virheitä valmiista koodeista tehtävänannon mukaisesti. Tehtävät oli jaettu kolmeen eri taitotasoon (simple task, medium task ja hard task) ja kuhunkin taitotasoon oli tarjolla kolme tehtävää (simple task 1, simple task 2, simple task 3 jne.) Ohjeistuksen jälkeen pareilla oli 15 minuuttia aikaa valita ja suorittaa haluamiaan tehtäviä. Lopuksi ohjaajat pyysivät osallistujia sanallistamaan yksittäisten tehtävien ratkaisuja muille opiskelijoille. Ainakin yksi pari ratkoi samalla kesken jäänyttä tehtävää. Debugging-osio kesti kokonaisuudessaan noin 30 minuuttia ja sitä seurasi kymmenen minuutin tauko. Toisen harjoituksen jälkeen parin 2 näytönkaappaus ei kuitenkaan tallentunut ja täten kyseisestä harjoituksesta saatiin parin osalta talteen vain GoProlla kuvattu aineisto.

Tauon jälkeen ohjelmointiharjoituksen viimeinen tehtävä oli suunnitella ja toteuttaa peli. Tehtävän ohjeistuksena oli tuottaa pelattava peli, jossa vähintään yhtä hahmoa pystyi ohjaamaan. Ohjeistuksessa myös kehoitettiin suunnittelemaan pelistä sopivan haastava ja rajattiin ajaksi 55 minuuttia. Ohjeistus tehtävään kesti noin viisi minuuttia. Pelit esiteltiin lopuksi muille osallistujille.

Viikko pariohjelmoinnin jälkeen järjestettiin osallistujille reflektiokerta Zoom -alustalla. Reflektiokerralla tavoitteena oli palata pariohjelmointiin ensin itsenäisesti ja tarkastella pariohjelmointikerran avainhetkiä, ja tämän jälkeen parien kokemuksia pariohjelmoinnista kartoitettiin parihaastatteluna. Avainhetkiin palaamisen tueksi vastaajat pääsivät selaamaan videoita omista ohjelmoinneista Iris Connectin kautta, jonne vastaajille luotiin omat tunnukset ja näkyviin joko näytönkaappaus tai GoProlla kuvattu tallenne ohjelmointikerralta. Datat katoamisen myötä osa pareista palasi joihinkin tehtäviin GoProlla kuvatun videon kautta. Tallenteista vastaajat etsivät ja kuvailivat omia avainhetkiään lomakkeen (LIITE 2) avulla. Avainhetket tarkoittivat tässä tapauksessa hetkiä, jotka he olivat kokeneet hyödyllisiksi oppimiselle tai tukivat pariohjelmointia,

taikka hetkiä, jotka osallistujat kokivat haastavina tai saivat aikaan ristiriitaa. Avainhetkistä kirjoitettiin muistiin tallenteen nimi sekä aikaleima, jotta niihin voidaan tarvittaessa tarkastella myöhemmin. Avainhetkiin palaamiseen ja muistiinpanojen kirjaamiseen annettiin osallistujille aikaa noin 20 minuuttia. Täytetyt avainhetkimuistiinpanot palautettiin kurssin Peda.net -sivustolle jaettuun palautuskansioon.

Avainhetkien käsittelyn jälkeen osallistujat jaettiin ohjelmointikerran pariin mukaan Zoomin breakout roomeihin. Jokaista paria meni lisäksi haastattelemaan yksi ohjaaja. Ohjaaja informoi osallistujia haastattelun tallentamisesta ja jakoi kysymykset osallistujille. Haastattelun oli luonteeltaan puolistrukturoitu teemahaastattelu (Hirsjärvi ym., 2009) ja tämän myötä osallistuvat parit keskittyivät keskusteluissaan hieman eri osa-alueille. Haastatteluun käytettävissä oleva aika oli kuitenkin suhteellisen lyhyt, joten valmiilla haastattelurungolla pyrittiin pitämään haastattelu tutkimuksen kannalta oleellisissa teemoissa (Patton, 2015). Haastattelurungossa (LIITE 3) käsiteltiin keskustelua virittelevän alun jälkeen pariohjelmointia, opettajien toimintaa, tehtäviä ja ympäristöä, omaa opettajuutta sekä kokemuksen jälkeistä pohdintaa.

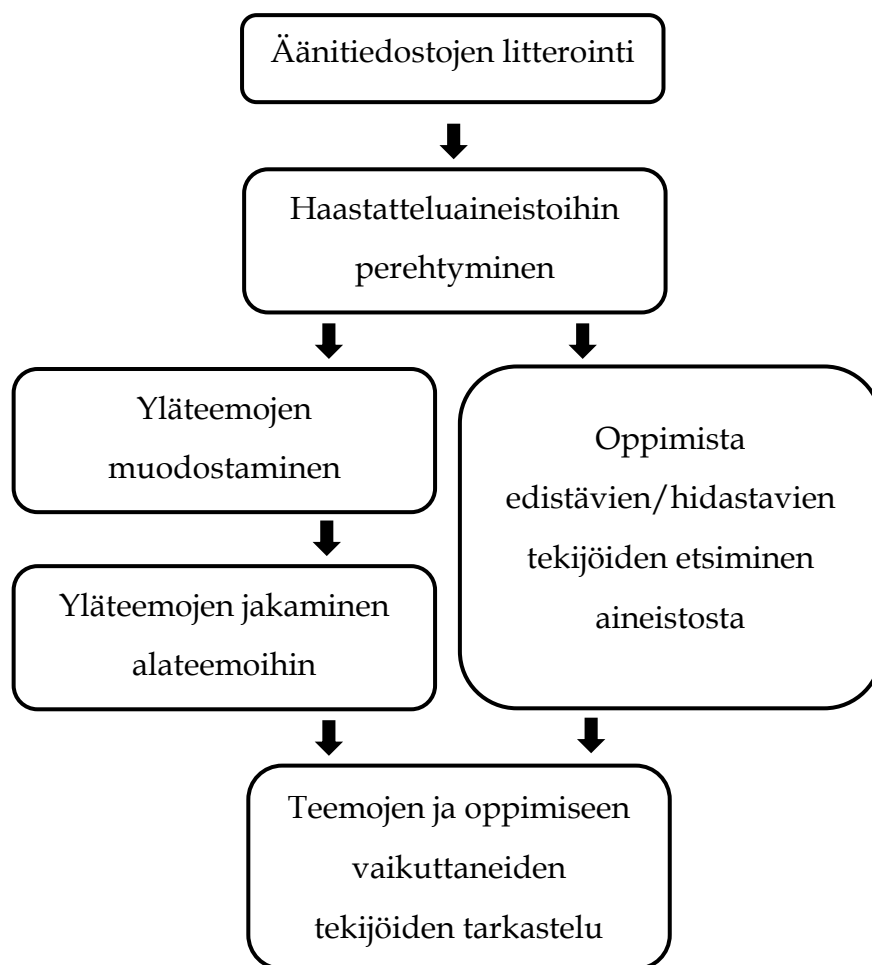
Haastattelujen äänitiedostoja litteroitaessa osallistujien nimet muunnettiin tunnistekoodiksi H1-H8 (H1 = haastateltava 1 jne.) anonymiteetin säilyttämiseksi. Tunnistekoodit jaettiin haastatteluista litteroitaessa parinumeroiden mukaan. Ensimmäisessä parissa toinen haastateltava nimettiin H1 ja toinen H2. Sama käytäntö pätee myös muihin pareihin, jolloin pari 2 on H3 ja H4, pari 3 H5 ja H6 ja pari 4 on H7 ja H8. Myös ohjaajien nimet muunnettiin haastatteluista tunnistekoodiksi ohjaajat 1-4. Tunnisteet annettiin numerojärjestyksessä sitä mukaan, kun ohjaajat mainittiin haastatteluissa. Näistä pseudonyymeistä koottiin avainlista, jonka avulla osallistuja pystyttiin yhdistämään tunnistekoodiin. Avainlistan avulla pystyttiin myöhemmässä vaiheessa yhdistämään esimerkiksi ennakkokyselyn aineisto oikeaan haastatteluaineistoon. Haastattelut litteroitiin kokonaisuudessaan sana sanalta niin kuin ne haastattelussa puhuttiin (Hirsjärvi ym., 2009).

## 4.4 Aineiston analyysi

Analyysiä varten ennakkokyselyt ja litteroidut haastattelut siirrettiin Atlas.ti ohjelmistoon, jota käytettiin analyysin tukena (Patton, 2015). Aineistoa päädyttiin analysoimaan laadullisen sisällönanalyysin mukaisin menetelmin, koska tapa soveltuu hyvin haastatteluaineistojen analysointiin tapaustutkimuksissa (Patton, 2015). Aineiston analyysi eteni Tuomen ja Sarajärven (2018) kuvaaman teemoittelun kautta aineistolähtöisesti, koska tässä tapauksessa haluttiin selvittää sitä, mitä kustakin teemasta on haastatteluissa sanottu. Analyysin etenemistä on havainnollistettu kuviossa 2.

### Kuvio 2

*Kuvio aineiston analyysistä*



Analyysin tavoitteena oli etsiä ja merkitä aineistoon ne kohdat, joissa vastaajat keskustelivat pariohjelmointikokemuksestaan ja kokemukseen vaikuttaneista asioista, kuten yhteistyöstä parin kanssa tai ohjelmointitehtävistä. Tutkimuksen analyysiyksikköinä toimivat ajatuskokonaisuudet, joiden pohjalta teemat määritettiin aineistosta (Tuomi & Sarajarvi, 2018). Nämä analyysiyksiköt saattoivat sisältää useitakin lauseita, joista teemat pelkistettiin. Esimerkki sitaattien teemoittelusta on kuvattuna taulukossa 2.

## Taulukko 2

*Esimerkki teemojen muodostamisesta aineistosta*

Pseudonyymi	Sitaatti	Pääteema/alateema
H5	"Niinpä. Ja mä jotenkin koen, että meillä oli ainakin <u>Scratchin suhteen sama taso</u> . Että sekin, että jotenkin että <u>saatiin yhdessä pohtia niin kuin samalla tasolla, samalla viivalla</u> , niin sekin auttoi. Että ehkä jos sä oisit ollut jotenkin tosi ammatti Scratchin käyttäjä, niin en tiedä oisko se sitten ollut sama juttu enää."	Yhteistyö - Taitoero
H3	"... siihen että sai itse rajata sen tehtävän ja sitten tehdä joku oma homma, niin siinä niin en muista että ei varmaan ikinä kysytty apua, mutta itse muistan kyllä miettineeni sitä, että <u>on ihan hyvä, että on joku semmoinen itseäkin vielä parempi olemassa, jos tulee joku ongelma</u> , että voi kysyä sitten neuvoja."	Tuki - Opettaja
H6	"... koin, että se motivoituminen tapahtui ja se innostuminen ja se semmoinen työskentelyn imu, niin se lähti lentoon sitten, kun me <u>päästiin itse kehittelemään sitä meidän [peliä]</u> ."	Tehtävät - Tehtävätyyppi

Haastattelujen litteroinnin ja niihin perehtymisen jälkeen aineistosta ryhmiteltiin haastateltavien kokemuksista pääteemoja, joiden voitiin katsoa vaikuttaneen oppimiskokemukseen. Analyysissa muodostettiin seuraavat pääteemat: yhteistyö parin kanssa, tuki, tehtävät ja oppimisympäristö. Pääteemoja jaettiin aineiston pohjalta edelleen alateemoihin (taulukko 3). Yhteistyö parin kanssa jaettiin päätöksentekoon, rooleihin ja niiden jakautumiseen, sekä parin keskinäiseen taitoeroon. Pääteema, tuki, jaettiin



seuraaviin alateemoihin: opettajan tarjoama tuki, parin tarjoama tuki ja muiden opiskelijoiden tarjoama tuki. Tehtäviä koskevat kokemukset luokiteltiin haastavuuden sekä tehtävätyypin mukaan. Tehtävätyyppi jaettiin edelleen avoimiin tehtäviin sekä suljettuihin tehtäviin. Suljetussa tehtävässä tarkoitetaan tässä tapauksessa rajattua opettajajohtoista tehtävää, jossa oppija seuraa esimerkiksi joko opettajan mallia tai muita yksityiskohtaisia ohjeita saavuttaakseen tietyn lopputuloksen. Avoimessa tehtävässä oppijalla taas on suuri päätäntävalta siihen, mitä ja miten lähtee toteuttamaan. Pääteema oppimisympäristö jaettiin aineiston perusteella fyysiseen oppimisympäristöön, välineisiin sekä Scratchiin oppimisympäristönä.

### Taulukko 3

*Pariohjelmointikokemukseen vaikuttaneet tekijät*

Pääteema	Alateema
1. Yhteistyö	a. Päätöksenteko b. Roolit ja niiden jakaantuminen c. Parin välinen taitoero
2. Tuki	a. Parilta saatu tuki b. Opettajalta saatu tuki c. Muilta opiskelijoilta saatu tuki
3. Tehtävät	a. Haastavuus b. Tehtävätyyppi
4. Oppimisympäristö	a. Välineet b. Scratch c. Fyysinen oppimisympäristö

Analyysin toisessa vaiheessa kokemuksista etsittiin vastausta siihen, tukiko opiskelijoiden mielestä mainittu asia oppimista vai haastoiko se sitä. Tässä vaiheessa hyödynnettiin ensimmäisen vaiheen kategorisointia ja tarkasteltiin, mainitsivatko opiskelijat niiden yhteydessä kyseisen tekijän tukevan vai hidastavan oppimista. Luonnollisestikaan osasta koodauksista tätä seikkaa ei

pystytty tarkastelemaan, jolloin tukemista tai vaikeuttamista ei merkitty. Esimerkkejä oppimista tukevista tekijöistä sekä oppimista haastavista tekijöistä on esitettyä taulukossa 4.

#### Taulukko 4

*Esimerkkejä oppimista tukevien ja haastavien tekijöiden maininnoista*

Pseudonyymi	Sitaatti	Vaikutus oppimiseen
H8	”Yhden minkä voim sanoa vielä, niin ehkä just se kun teki sitä omaa peliä, niin jotenkin mun mielestä <u>sitoutti hirveesti siihen työskentelyyn</u> tavallaan, että siihen jotenkin <u>uppoutui ihan tosi tiiviisti sen tekemiseen</u> . Mulla ainakin meni siinä vähän niin kuin ajantaju, kun oli niin tavallaan <u>keskittynyt</u> siihen ja halusi sen saada tehtyä. Ja sitten se oli jotenkin niin <u>hauskaakin tehdä sitä yhdessä, niin ja suunnitella</u> , niin se oli ainakin semmoinen.”	Tuki oppimista
H1	”No mun mielestä ainakin, mä en ollu siis Scratchia ihan hirveesti oikein ennen käyttäny, ja kyl mä tavallaan ymmärsin sen mikä siellä on niinku taustalla. Että sit se kehitty siinä koko ajan, mut mun mielestä se oli kiva kun (H2) oli siinä kaverina, niin sitten tavallaan <u>pystyi keskustelemaan vähän siitä että miten nää toimii</u> , ja sitten ku (H2) oli vähän etevämpi siinä Scratchin käytössä ainakin alkuun. Mulla olis kestäny ikuisuus varmaan etsiä ne sieltä. Niin sain sellasta tukea.”	Tuki oppimista
H4	”No siis kyllä ainakin omasta näkökulmasta se alku oli todella tylsä. Että kun se oli semmoinen tosi rautalankamalli ja semmoinen opettajajohtoinen, että tehdään vaihe yksi, sitten tehdään vaihe kaksi, tehdään vaihe kolme, oletko päässyt tähän vaiheeseen. Ja sitten kun on tuttua asiaa, niin se oli vähän semmoinen, että <u>odotteli vaan, että päästäänkö kohta tekemään niin kuin semmoisia haasteellisia tai semmoisia vähän niin kuin oman tason tehtäviä</u> .”	Haastoi oppimista
H6	”...mä en edes muistanut tuota labyrinttiä, että mulla, mä muistan vaan siitä alusta sen, että me tehtiin tosi paljon ohjatusti ja sitten mä varmaan siinä sanoinkin ääneen, että <u>mulle tämän tapainen tekeminen, tää ei tuota sitä oppimista</u> . Että tää ehkä osoittaa hyvin sen, että mulla ei ole jäänyt siitä hirveästi mieleen.”	Haastoi oppimista

## 4.5 Eettiset ratkaisut

Koko tämän tutkimuksen ajan noudatettiin hyvän tieteellisen käytännön periaatteita (Hirsjärvi ym., 2009; Tuomi & Sarajärvi, 2018). Osallistuminen tutkimukseen oli koko tutkimuksen ajan vapaaehtoista ja osallistujia informoitiin tutkimuksen tarkoituksesta (Mäkelä, 2005). Tutkimusaineisto kerättiin opiskelijoiden kurssin aikana, joten samalla tuli varmistua siitä, että heidän päätöksensä olla osallistumatta tutkimukseen ei esimerkiksi vaikuta heidän kurssisuoritukseensa tai arviointiin. Tämä oli erityisen tärkeää huomioida tässä tutkimuksessa tutkimusryhmän ja tutkittavien suhteesta johtuen. Osa tutkimusryhmästä toimi samalla kurssilla opettajina. Osallistumisen vapaaehtoisuudesta muistutettiin ennakkokyselyn, pariohjelmoinnin ja parihaastattelun yhteydessä. Ennen osallistumista opiskelijat myös allekirjoittivat suostumuslomakkeen tutkimukseen osallistumisesta, sekä tietosuojailmoituksen tutkimusaineiston käsittelystä ja säilytyksestä.

Kerätty aineisto säilytettiin Jyväskylän yliopiston verkkoasemalla, jolloin aineistoon pääsi käsiksi vain erikseen määritetyt kirjautuneet henkilöt VPN-yhteyden kautta tai yliopiston verkossa. Parihaastatteluissa käytetty Zoom-ohjelma tuottaa nauhoituksesta sekä ääni- että videotiedoston. Ääniraidat siirrettiin verkkoasemaan ja videotiedosto tuhottiin. Myös pseudonymisoitujen haastattelujen avainlista säilytettiin verkkoasemalla. Pseudonymisoitua ennakkokysely- ja haastatteluaineistoa säilytettiin tutkijaryhmän yhteisessä Teams-kansiossa.

Esitin aineiston analyysisuunnitelman litteroinnin jälkeen muulle tutkijaryhmälle, jolloin sain myös palautetta analyysisuunnitelmasta. Muut tutkijaryhmän jäsenet tutustuivat myös aineistoon ja kirjasivat omia huomioitaan, jolloin pystyin vertaamaan niitä omiin huomioihini ja tekemiini teemoitteluihin.

Tutkimuksessa on verrattain pieni otos ( $N = 8$ ) ja se vaikuttaa väistämättä tutkimuksen tulosten laajuuteen ja tarkkuuteen. Kuitenkin laadullisen tutkimuksen tarkoituksenmukaisuus on toisaalta vahvasti yhteydessä aineiston laatuun ja laajuuteen otoskoon sijaa (Patton, 2015). Tämän tutkimuksen

tavoitteen sekä etsivän otteen huomioon ottaen aineisto oli riittävän laaja ja tarpeeksi laadukas tarkoituksenmukaisen analyysin toteuttamiseen.

Tutkimuksen löydöt pyrin perustelemaan laajasti kaunistelematta niitä ja perustelemaan löytöjä lainauksilla aineistosta (Hirsjärvi ym., 2009). Vaikka laajat kuvaukset tutkimusaineistosta luovat pohjan tulkinnoille, ei kuitenkaan ole tarkoituksenmukaista, että tuloksissa raportoidaan kaikki, mitä tutkittavat ovat sanoneet (Patton, 2015). Täten pyrin tasapainottelemaan riittävän laajan kuvailun ja tarkan tulkinnan välillä.

## 5 TULOKSET

### 5.1 Pariohjelmointikokemukseen vaikuttaneita tekijöitä

Aineistosta muodostettiin neljä pääteemaa: yhteistyö, tuki, tehtävät ja oppimisympäristö. Pääteemat jaettiin edelleen alateemoihin. Pääteemojen sekä alateemojen jakautumista kuvataan taulukossa 5 ja tekstinä seuraavissa kappaleissa. Yhteistyö -pääteemaa käsittelevät kokemukset ( $f = 64$ ) jaettiin seuraaviin alateemoihin: päätöksenteko ( $f = 27$ ), parin taitoero ( $f = 15$ ) sekä roolit ja niiden vaihtuminen ( $f = 22$ ). Yhteistyöteemasta keskustelivat selvästi eniten pari kaksi (21 mainintaa) ja neljä (22 mainintaa), kun taas pareilla yksi (10 mainintaa) ja kolme (11 mainintaa) yhteistyöstä keskusteleminen oli haastattelussa vähäisempää. Yhteistyöhön liittyviä löytöjä sekä yhteistyötä oppimisen kannalta käsitellään kappaleessa 5.2.

Tuki -pääteemaan ( $f = 26$ ) muodostui kolme alateemaa: opettaja, pari ja muut oppijat. Merkittävimmäksi teemaksi nousi selvästi opettajan tarjoama tuki ( $f = 16$ ) ja vähäisimmäksi muiden oppijoiden tuki ( $f = 1$ ). Parin tuki ( $f = 9$ ) mainittiin, mutta tämän alateeman erottelu parin kanssa tehtävästä yhteistyöstä oli jokseenkin pulmallista. Pari numero kolme toi esiin muita useammin esiin huomioita tuesta ( $f = 10$ ) kuin muut parit. Pari numero yksi mainitsi teeman hieman harvemmin ( $f = 6$ ), pari numero kaksi ja pari numero neljä ( $f = 5$ ). Pari numero kolme myös keskittyi selvästi enemmän opettajan tarjoamaan tukeen (Opettajan tuki  $f = 8$ , parin tarjoama tuki  $f = 2$ , muiden oppijoiden tarjoama tuki  $f = 0$ ). Tuloksia kuitenkin hieman vääristää parin tarjoaman tuen ja yhteistyön häilyvä linjanveto. Tuki -pääteemaa sekä koettua yhteyttä oppimiseen käsitellään kappaleessa 5.3.

Tehtävät - pääteemaan havaittiin liittyvän yhteensä 44 mainintaa. Näistä 14 käsitteli tehtävien haastavuutta ja 30 tehtävätyyppiä (suljettu-avoin). Pari kolme ja neljä painottivat haastatteluisaan tehtävätyyppiä (Pari 3: haastavuus  $f = 6$ , tehtävätyyppi  $f = 12$ . Pari 4: haastavuus  $f = 1$ , tehtävätyyppi  $f = 9$ ). Tehtäviin liittyviä tuloksia käsitellään kappaleessa 5.4.

Haastatteluista pystyttiin muodostamaan myös oppimisympäristö - pääteema ( $f = 7$ ). Teemasta puhuttiin kuitenkin vain kahdessa haastattelussa. Teema kuitenkin pystyttiin jakamaan edelleen fyysiseen ympäristöön ( $f = 1$ ), Scratch-ohjelmointiympäristöön ( $f = 3$ ) sekä välineisiin ( $f = 3$ ). Scratch oppimisympäristönä sekä välineet mainittiin kummatkin kolme kertaa ja fyysinen oppimisympäristö kerran. Yhteensä osallistujat mainitsivat oppimisympäristön seitsemän kertaa.

## Taulukko 5

*Pariohjelmointiin liittyneiden teemojen mainintojen määrät*

Pääteema	$f$	Alateema	$f$
Yhteistyö	64	Päätöksenteko	27
		Roolit ja niiden jakaantuminen	22
		Parin välinen taitoero	15
Tuki	27	Parilta saatu tuki	9
		Opettajalta saatu tuki	16
		Muilta opiskelijoilta saatu tuki	1
Tehtävät	44	Haastavuus	14
		Tehtävätyyppi	30
Oppimisympäristö	7	Välineet	3
		Scratch	3
		Fyysinen oppimisympäristö	1

## 5.2 Yhteistyö

Kaikki parit kertoivat haastatteluissa yhteistyöstään ja erityisesti, miten päätöksentekoprosessi eteni pariohjelmoinnissa ( $f = 27$ ). Keskustelulle, yhteisellä ideoinnilla ja kokeilemisella oli keskeinen rooli. Kaikki haastateltavat toivat haastatteluissa ilmi, että kokivat tärkeänä ”hyvän vuorovaikutuksen” (H7) harjoituksen kannalta. Päätöksenteko eteni toisen ehdottamasta ideasta toteutuksen suunnitteluun, ”toteutuksen kokeiluun”, mahdollisten ”ongelmien

korjaamiseen” ja tämän jälkeen jatkoideaan (H6). Ongelmatapauksissa vastaajat kertoivat ”palanneensa takaisin ja kokeilleensa uutta” (H7).

Oli myös mielenkiintoista, että yhdessä haastattelussa tuli esiin tilanne, jossa pari oli peliä kehitettäessä eri mieltä ”yksityiskohtien hiomisesta” ennen kuin ”perusjutut” olivat valmiina (H4). Parin toinen osapuoli (H3) kuitenkin kuvaili kokeneensa erimielisyyden ennemminkin ”valaisevaksi” hyökkäävän sijaan. Muut parit eivät tuoneet haastatteluissa esiin vastaavista erimielisyyksistä.

Yhteistyö päätöksenteossa koettiin valtaosin hyödyllisenä oppimisen kannalta ( $f = 14$ ). Vain yhdessä ( $f = 1$ ) tapauksessa vastaaja kertoi tilanteesta, jossa koki yhteistyön puutteen haastaneen oppimista, kun H8 ”teki enemmän” ja ”H7 ehkä vähän katsoi”. H8 kertoi tämän tapahtuneen ensimmäisessä tehtävässä ja yhteistyön lisääntyneen myöhemmissä tehtävissä.

Roolit ( $f = 24$ ) jakautuivat aluksi kaikilla pareilla pitkälti sattumalta muun muassa ”sen mukaan miten istuttiin” (H2). Rooleihin liittyneistä maininnoista viiteen ( $f = 5$ ) liittyi myös oppimista edistävä kokemus ja neljään ( $f = 4$ ) oppimista hidastava tai haastava kokemus. Roolien vaihtelu työskentelyn aikana kuitenkin erosi huomattavasti parien välillä ja roolijako koettiin eri tavoin oppimisen kannalta. Yksi pari toteutti koko ohjelmoinnin samalla kuski-kartturi jaolla kummankin pitäessä sitä toimivana työskentelytapana pariohjelmointiharjoitukseen. H4 kertoi, että häntä kiinnosti enemmän ”miettiä niitä ongelmia” ja ”keskittyä siihen, miten tää toteutetaan tai miten ratkaista toi ongelma”. Myös parin toinen osapuoli H3 kertoi tavan toimineen ja koki H4 sanoittaneen ”mitä hän pohtii”. Toisaalta toiset kokivat merkittäväksi itse tekemisen oppimisen kannalta ja kiinnittivät enemmän huomiota roolien tasaiseen jakaantumiseen. Esimerkiksi pari 3 kertoi, että he olivat päättäneet ”aika alkuun, että me vaihdellaan” ja kumpikin kertoi kokeneensa, että roolien vaihto oli ”luontevaa” (H5). H6 myös mainitsi, että työmäärä jakautui tasaisesti eikä hän ”tehnyt jotain liian vähän tai liikaa”. Kumpikin oli H5 mukaan ”aktiivisia, vaikka toisella oli hiiri tai näppäimistö”, ”ehdotti koko ajan jotain, ”auttoi toisiamme” sekä oli ”läsnä siinä tilanteessa”.

Yksi pareista kiinnitti haastattelussa huomiota seuraamisen ja työstämisen väliseen eroon oppimisen kannalta. Haastateltava kertoi tilanteesta, jossa huomasi seuraamisen jälkeen, ettei osannutkaan ohjelmointia samalla tasolla, kuin ajatteli seuraamisen aikana osaavansa.

Joo, mutta itse asiassa mä huomasin siinä, että kun mä en ollut tehnyt niitä, niin sitten tuntui, tai siis vaikka mä olin seurannut vieressä, että mitä (H8) tekee ja tuntui, että mä olin koko ajan mukana, mutta sitten kun jouduin itse ottamaan sen hiiren ja rupee laittamaan niitä, sitten mä en yhtäkkiä niin kuin osannutkaan samalla lailla, vaikka musta tuntui se ihan selkeältä, mitä (H8) teki. Niin sitten se ei ollutkaan sama asia, kun mä rupesin itse tekemään. Se tuntui, että piti eri lailla lähteä ajattelemaan. (H7)

Taitoeroa koskeva keskustelu oli luonteeltaan hyvin pohtivaa, ja se oli haastateltavien mukaan herättänyt keskustelua myös koko ryhmän kesken kahvitauolla. Enemmistö vastaajista (n = 5) mainitsi pitäneensä parin osapuolien samankaltaista osaamisen tasoa hyvänä asiana oppimisen kannalta tämän kaltaisessa työskentelyssä. Samalla viivalla ohjelmointiosaamisen suhteen oleminen mahdollisti vastaajien mukaan muun muassa sen, että he "saivat yhdessä pohtia samalla tasolla" (H5), "tasavertaisemman vuorovaikutuksen" sekä "vaikuttamisen projektiin" (H4).

Moni silti pohti haastatteluissa eri tasoisten ohjelmoijien hyötyjä ja haittoja pariohjelmoinnin kannalta. Esimerkiksi H7 pohti sitä, että eritasoisista ohjelmoijista muodostetut parit olisivat "saattaneet tukea niitä heikompi-tasoisia" ja edistyneemmät taas olisivat "joutuneet selittämään toiselle". Toisaalta esimerkiksi H4 mietti sitä, "jos edistyneempi vääntää perusasioita rautalangasta moneen kertaan, niin turhautuuko siihen tai saako hän siitä mitään".

Aineistosta pystyi tekemään kiinnostavan havainnon osallistujasta (H2), joka ennakkokyselyssä kuvasi itseään "hyvin aloittelevaksi ohjelmoijaksi" ja koki pariohjelmoinnin "herättävän paniikkia" sekä kyseenalaisti oman panoksensa merkittävyyttä pariohjelmoinnin kannalta. Tämä opiskelija kertoi myös haastattelussa kokeneensa parin sanomisen vaikuttaneen ohjelmointikokemukseen negatiivisesti.

(H2) No mulla oli ehkä sellanen, että ku mä oon ehkä aiemminkin kokenu, että mä oon vähän niinku hidas, niin sitte ku me neuvoteltiin siitä, että kumpi käyttää hiirtä siinä ku me tehtiin



sitä projektia, niin sitten toi niinku just toi niinku (H1) sano että hän on niin nopea, että on parempi että mä käytän hiirtä ja se ei ollu mitenkään silleen, et se ois niinku positiivisessa mielessä sanottu, mutta mulle tuli sellanen olo että oonks mie jotenkin hidas, hah. Vaikka mä tiedän, että et sä sitä tarkottanu silleen. Mut mul tuli vaan sellanen olo.

(H1) Joo.

(H2) Että oonko mä sit vaikka se idea oli siinä se kun (H1) on pelannu niin paljon tavallaan et sitten (H1) ois klikkaillu niitä juttuja niin että mä en ois kerenny seuraamaan sitä.

Parin toinen osapuoli H1 kuitenkin kuvasi haastattelussa H2 "etevämmäksi Scratchin käytössä, ainakin alussa" ja kertoi löytäneensä esimerkiksi tarvittavia toimintoja nopeammin ohjelmointiympäristöstä H2 avulla.

Osa opiskelijoista toi haastatteluissa esiin "avoimen ja turvallisen ilmapiirin" (H6) merkityksellisenä. Erityisen merkitykselliseksi koettiin samalla tasolla oleminen ja uskallus tuoda esiin omaa osaamattomuutta.

(H5) Ja mä jotenkin koen, että meillä oli ainakin Scratchin suhteen sama taso. Että sekin, että jotenkin että saatiin yhdessä pohtia niin kuin samalla tasolla, samalla viivalla, niin sekin auttoi. Että ehkä jos sä oisit ollut jotenkin tosi ammatti Scratchin käyttäjä, niin en tiedä oisko se sitten ollut sama juttu enää.

(H6) Niin totta, ja samat sanat. Koska sitten varmasti se, että se just vielä varmasti osaltaan sitten kuitenkin vaikutti siihen, että sitä uskalsi myös kysyä kaikkea mahdollista ja olla jotenkin silleen.

(H5) Pihalla.

(H6) Avoin. Niin avoimesti pihalla.

Turvallinen ilmapiiri rakentui parin numero kolme (H5 & H5) mukaan nopeasti, ja H6 koki "uskaltavansa kysyä ja ehdottaa". Haastattelussa H6 myös mainitsi huomanneensa avainhetkitekävän aikana videolta välittyneen sen, kuinka ilmapiirin myötä tehtävä eteni ja "oma ymmärrys rakentui nimenomaan yhteistyön avulla". H6 myös korosti samassa yhteydessä "ajatusten,

kysymysten, ehdotusten sekä ratkaisuiden” ääneen sanoittamisen tärkeyttä.  
Myös parit

### 5.3 Tuki

Tuki -pääteemaan koodatuista kohdista ( $f = 26$ ) valtaosa käsitteli opettajan tarjoamaa tukea ( $f = 16$ ). Paikalla olleet ohjaajat sopivat keskenään, että opiskelijoita autetaan silloin kun he pyytävät apua. Muuten heidän annettiin työskennellä keskenään. Luonnollisesti parit tarvitsivat eri määrän tukea opettajalta pariohjelmointiharjoituksen aikana, mutta kaikki kokivat mahdollisuuden opettajan tarjoamaan tukeen hyödyllisenä oppimisen kannalta. Esimerkiksi H3 kertoi pitäneensä tärkeänä sitä, että ”on joku semmoinen itseäkin vielä parempi olemassa, jos tulee joku ongelma, että voi kysyä sitten neuvoja”. H3 ei kuinkaan muistanut, että olisivat missään vaiheessa kysyneet apua harjoituksen aikana. Ohjaajien toiminnassa koettiin myös hyväksi rauhallinen olemus, oikea-aikaisuus sekä tapa, jolla opettajat tukivat.

- (H6) Joo kyllä mä koin, kirjoitinkin tuonne että, koin jotenkin että (Ohjaaja 1) sai ihan älyttömän hyvin tukea ja apua, että hän on jotenkin ihan, olemus oli niin semmoinen rauhallinen ja sitten, että hän osasi silleen jotenkin oikea-aikaisesti antaa sitä tukea ja ohjausta ja neuvoja. Ja sitten just silleen että ”hei nyt te ootte oikeilla jäljillä” ja silleen ohjata sinne oikeeseen suuntaan. Että sieltä ei tullut niitä valmiita että näin, nyt teette näin ja näin ja näin, vaan että jotenkin tuki sitä meidän työskentelyä ja oppimista.
- (H5) Mm, kyllä. Niin (Ohjaaja 1):lta uskalsi kysyä.
- (H6) Kyllä, nimenomaan. Ja sitten just ehkä oikeesti syntyi sen kautta niitä semmoisia omia oivalluksia ja jotenkin tuntui, että ton aamupäivän aikana niin kuin omat ohjelmointitaidot harppasi huimasti eteenpäin. Ja ehkä semmoinen niin kuin varmuus siihen, että uskaltaisi lähteä kokeilemaan oppilaiden kanssa myöskin

Yksi pari myös kertoi kokeneensa ohjaustyylin lisäksi hyödylliseksi oppimisen kannalta sen, että heitä neuvoi joka kerralla sama ohjaaja. Tällöin

heidän ohjaajansa "tiesi mitä he tekevät " eikä heidän "tarvinnut selittää projektinsa suunnitelmaa aina uudestaan" (H5 & H6), vaan kokivat voivansa keskittyä opettajan kanssa heti käsillä olevaan pulmaan. Tällöin ohjaaja pystyi tarttumaan heti H5 kokemuksen mukaan heti kiinni käsillä olevaan pulmaan.

Opettajien kannustava puhe koettiin tärkeäksi tilanteen ilmapiirin kannalta. Esimerkiksi H1 kuvaili ohjaajia "helposti lähestyttäviksi" sekä ilmapiiriä "kysymiseen kannustavaksi". Myös pari numero neljä kuvaili ohjaajia kannustavaksi ja yleistä ilmapiiriä hyväksi. Ainoa asia opettajien toiminnassa, joka mainittiin aineistossa oppimista hidastavaksi, oli tehtävät ohjeistaneen opettajan osallistuminen etäyhteyksin. Tämäkin seikka mainittiin kerran yhden osallistujan toimesta ( $f = 1$ ). Opettajien tuki mainittiin oppimista edistäväksi aineistossa 11 kertaa ( $f = 11$ ).

Parin tuki koettiin tärkeäksi. Tämä teema sitoutui vahvasti yhteistyön alateemoihin ja niiden erottelu oli hieman pulmallista. Kuitenkin aineiston perusteella pystyttiin osoittamaan, että parin koettiin muun muassa "voimavarana" (H3), "täydentävän omia heikkouksia" (H1) ja "antavan vahvistuksen" (H8) työskentelyyn. Yksi vastaaja (H5) toi haastattelussa esiin matalamman kynnyksen kysyä apua parilta kuin opettajalta, ja näin ollen pareittain työskentelyn vähentäneen opettajan tuen tarvetta. Parin tarjoaman tuen koettiin myös lisäävän "sinnikkyyttä", "rohkeutta" ja "motivaatiota" (H6). Aineistossa oli merkillepantavaa, että kaikki osallistujat kertoivat tavalla tai toisella kokeneensa parin voimavarana ja hyödyllisenä oppimisen kannalta ( $f = 9$ ).

Ryhmän muiden jäsenten vaikutus oppimisen kannalta mainittiin vain kerran ( $f = 1$ ). Maininta muiden liittyi muiden opiskelijoiden toteuttamiin projekteihin tutustumiseen vapaa-ajalla ja mahdollisuuksiin harjoitella "samanlaisten tai lähes samanlaisten pelien luomista itse"(H2).

## 5.4 Tehtävät

Parit suorittivat pariohjelmointiharjoituksessa kolme tehtävää, joista ensimmäinen oli hyvin vahvasti ohjattu, toisessa pareilla oli jonkin verran valinnanvaraa ja viimeisessä tehtävä oli vapaa ja avoin. Viimeisen tehtävän tavoitteena oli rakentaa pieni pelattavissa oleva peli aikarajan puitteissa. Pariohjelmointi oli selvästi vastaajien mielestä mielekkäämpää avoimemmissa tehtävätyypeissä, joissa he pääsivät itse suunnittelemaan tavoitteensa ja toteutuksensa verrattuna suljetumpaan tehtävätyyppiin, jossa opettaja tai annetut ohjeet määrittivät toiminnan tavoitteet ja toteutuksen. Kokemuksiin tehtävistä linkittyi vahvasti aineistossa myös tehtävien haastavuustaso. Pelin ohjelmoiminen kahdestaan lisäsi vastaajien mukaan keskustelua ja suunnittelua yhdessä, kun parin kanssa oli ”vähän niin kuin pakko keskustella keskenään mitä me halutaan tehdä, millä lailla me halutaan tehdä ja minkä näköinen me halutaan tehdä” (H1). Viimeinen tehtävä myös koettiin innostavana, mukaansa tempaavana ja motivoivana. Kaiken kaikkiaan aineiston perusteella voidaan todeta sopivasti rajatun avoimen tehtävän ja pariohjelmoinnin sopineen hyvin yhteen.

Sen sijaan etenkin ensimmäinen, hyvin selkeästi ohjattu tehtävä jakoi mielipiteitä osallistujissa. Toisaalta osa vastaajista piti alkua hyödyllisenä myöhempien tehtävien kannalta ja koki sen ”nopeuttavan alkuun pääsemistä” (H7) myöhemmissä harjoituksissa sekä ”madaltavan kynnystä työskentelyn aloittamiseen” (H6) sekä toimivan ”hyvänä mieleen palautuksena” (H7).

No mun mielestä oli tosi hyvin toimiva se, että ensin katsottiin perusjutut, koska ilman niitä se olisi ollut tosi vaikea lähteä tekemään mitään omaa, varsinkaan tuottavaa. Mä koin, että se oli tosi hyvä että katsottiin ensin yhdessä se tosi opettajajohtoisesti ja sitten sen jälkeen vähän muokattiin niitä ja sitten lopuksi oli se oma, koska sitten huomasi että siellä mitä oli aikaisemmin käyttänyt tai oppinut, niin kuin joitakin juttuja. (H8)

Kuitenkin moni osallistujista (n = 5) koki ensimmäisen tehtävän työskentelytavan jollain tasolla turhauttavana tai ei kokenut sitä tarkoituksenmukaisena. Haastateltavat kertoivat kokeneensa, että parin läsnäololla ei ollut sinällään merkitystä ja keskustelun parin kanssa jääneen muun muassa lähinnä toteamuksiksi, kuten ”laitetaan tää tähän ja tää

seuraavaksi, joo” (H4), kun opettaja kertoi toimintaohjeet vaihe vaiheelta. Jotkut myös kokivat ”toteuttavansa opettajan ohjeita sen kummemmin ymmärtämättä, mitä oikeastaan tekivät” (H6). Vastaajat myös kertoivat, että ”työskentelyn imu” syntyi vasta kolmannessa tehtävässä, jossa osallistujat pääsivät kehittämään itsen peliään.

- (H6) ...koin, että se motivoituminen tapahtui ja se innostuminen ja se semmoinen työskentelyn imu, niin se lähti lentoon sitten, kun me päästiin itse kehittelemään sitä meidän.
- (H5) Niin niin. Niin ehkä se vähän tappoi sitä, siinä se niinku alkuun se semmoinen, että piti oikeasti niinku tehdä vaan, mitä toinen käskee ja niin, kyllä.

Viimeisen tehtävän avoimuus antoi vastaajien mielestä hyvän mahdollisuuden asettaa omalla taitotasolle sopivia haasteita, ja osallistujat lähestyivät kolmatta tehtävää erilaisilla tavoilla. Esimerkiksi pari 3 kertoi lähteneensä liikkeelle ”hyvin simppeleistä pelistä” ja ”kehittelleensä sitä monimutkaisemmaksi” ja laajemmaksi onnistumisten myötä (H6). Pari 2 taas kertoi lähteneensä heti toteuttamaan ”uudenmallista peliä, jota ei ollut vielä kokeillut” (H3). Kuitenkin aineiston perusteella pariohjelmointi koettiin ”motivoivammaksi” (H6) ja tarkoituksenmukaisemmaksi silloin, kun työskenneltiin riittävän haastavien ja avointen tehtävien parissa:

onko se kauheen tehokasta tehdä pareittain tai aloittaa ohjelmointi pareittain, jos opettaja kummiskin näyttää siellä aina edessä ne vaiheet. Että onko se miten tarkoituksenmukaista siinä vaiheessa se pari, ehkä enemmän sitten koen sen silloin kun on joku tällainen oma taitotaso, jota lähdetään testaamaan tai haastamaan jollakin tavalla. (H3)

Toisessa, valmiita koodeja muokkaavassa tehtävässä koettiin oppimisen kannalta hyödylliseksi vapaus ”valita itse sopivia debuggaus-tehtäviä” (H1) ja ”mennä suoraan vaikeimpiin tehtäviin” (H3).

## 5.5 Oppimisympäristö

Oppimisympäristöön liittyneet maininnat käsittelivät käytössä olleita välineitä, Scratchia oppimisympäristönä sekä yleistä fyysistä

oppimisympäristöä. Yksi osallistuja koki "riittävän suurikokoiset näytöt" (H1) olennaiseksi pariohjelmoinnin kannalta, ja toivoi ettei vastaavaa tarvitsisi toteuttaa myöhemmin esimerkiksi tableteilla.

Scratchin mainittiin olevan ohjelmointiympäristönä "yksinkertainen käyttää ja selittää" (H2). Lisäksi Scratch koettiin oppimista edistävänä ( $f = 3$ ) "hyvien ohjevideoiden"(H2), kielivaihtoehtojen (H1) sekä "monipuolisten vaihtoehtojen " (H7) vuoksi. Ohjelmointiympäristön ei mainittu hidastaneen oppimista.

Fyysisen oppimisympäristö mainittiin vain yhden parin toimesta (Pari 1). He kokivat pienen luokkatilan vuoksi muiden parien olleen niin lähellä, että he olivat vahingossa "keskittyneet muiden parien keskusteluihin ja ohjaajan neuvomisiin" ja kertoivat tämän "tuoneen häiriötä" työskentelyyn (H2 & H1).

## 6 POHDINTA

### 6.1 Tulosten tarkastelu ja johtopäätökset

Tämän tutkimuksen tarkoituksena oli perehtyä opiskelijoiden pariohjelmointiharjoitukseen vaikuttaneisiin teemoihin sekä selvittää sitä, miten ja mitkä tekijät vaikuttivat koettuun oppimiseen. Aineiston perusteella pääteemoja olivat yhteistyö, tuki, tehtävät sekä oppimisympäristö. Tutkimuksen löydöt osoittavat osaltaan sen, että pariohjelmointi voi toimia hyvin ohjelmoinnin harjoittelussa opettajaopiskelijoilla. Tutkimuksen tulokset vahvistavat muun muassa Hawlitschekin ym. (2022) käsitystä siitä, että pariohjelmoinnin ohjaaminen vaatii huolellista suunnittelua etukäteen, jotta pariohjelmointi on tarkoituksenmukaista. Huomiota tulee kiinnittää tämän tutkimuksen perusteella ainakin roolien jakamiseen, osallistujien ohjelmointiosaamiseen, pariin muodostamisen sekä sopiviin tehtäviin haastavuuden ja tehtävän luonteen kannalta.

Pariohjelmoitaessa yhteistyö eteni osallistujilla suunnittelusta kokeiluun, mahdollisten virheiden etsimiseen ja korjaamiseen ja taas uuden ominaisuuden suunnitteluun. Toiminta eteni etenkin kahdessa jälkimmäisessä tehtävässä yhteistyön ja vuorovaikutuksen kautta, mikä on pariohjelmoinnille tyypillistä (Bryant ym., 2006) ja näyttäisi olevan hyödyllistä oppimisen kannalta (Hawlitschek ym., 2022). Aineiston perusteella parit sanallistivat toimintaansa ja toivat näkemyksiään esiin pariohjelmointiharjoituksen aikana, jonka taas on todettu aiemmassa tutkimuksessa lisäävän oppimista ja syventävän ymmärtämistä (Bryant ym., 2006). Täten pariohjelmoinnilla työskentelytapana voitaisiin kehittää opettajaopiskelijoiden ohjelmointiosaamista.

Roolit jakautuivat aineiston perusteella alkuun lähinnä satunnaisesti istumapaikan mukaan. Roolien vaihtelussa oli sen sijaan eroa. Esimerkiksi pari numero kaksi työskenteli koko harjoituksen ajan samoissa rooleissa ja kumpikin kertoi kokeneensa työn jakaantuneen tasavertaisesti ja kummankin osallistuneen tasaisesti. Sen sijaan yksi toinen osallistuja oli huomannut, että kuskina hän ei

osannutkaan työskennellä samalla tasolla, kuin luuli kartturina ollessaan osaavansa. Muutoin parit kertoivat jakaneensa rooleja tasaisesti. Hawlitschekin ym. (2022) näkemys on, että pariohjelmoinnin ohjaajan tulisi ohjeistaa pareja kiinnittämään huomiota roolien jakamiseen. Herääkin kysymys, olisiko oppimista tapahtunut enemmän, mikäli osallistujia olisi ohjeistettu vaihtelevaan rooleja tasaisesti työskentelyn ajan. Pariohjelmointiharjoituksessa tulisi siis ohjeistaa opiskelijoita roolien tasaiseen jakamiseen.

Aineiston perusteella parit kokivat samankaltaisesta ohjelmointiosaamisesta olleen hyötyä oppimiselle. Tämä havainto on samassa linjassa muun muassa Dillenbourgin (1999) yhteisöllisen oppimisen käsityksen kanssa sekä Hawlitschekin ym. (2022) havaintojen kanssa, joissa korostetaan samankaltaisen osaamistason merkitystä ohjelmointiosaamisen kehityksen kanssa. Oli kuitenkin mielenkiintoista seurata osallistujien käymää pohdintaa mahdollisesti eritasoisten parien yhteistyöstä. Kuitenkin valtaosa koki samankaltaisen ohjelmointiosaamisen hyödylliseksi työskentelyn kannalta. Näyttäisi siis siltä, että pariohjelmointiharjoituksessa parit kannattaisi muodostaa siten, että he ovat kutakuinkin samalla viivalla osaamisensa suhteen. Tämä taas edellyttää ohjaajalta tietoa osallistujien aiemmasta ohjelmointiosaamisesta.

Tutkittavat kertoivat saaneensa tukea parilta ohjelmoinnin aikana ja tehneensä yhteistyötä ohjelmointitehtävän sekä sen aikana nousseiden ongelmien ratkaisemiseksi. Haastatteluaineistosta kävi ilmi turvallisen ilmapiirin merkitys työskentelyyn ja yhteistyöhön parin kanssa. Tasavertaisuus parin ohjelmointiosaamisen suhteen koettiin turvallisuutta lisääväksi, jolloin osallistujat uskalsivat esittää rohkeasti kysymyksiä ilman pelkoa siitä, että asettaisivat itsensä naurunalaiseksi. Tämä saattaisi siis tarkoittaa sitä, että pariohjelmoinnissa kannattaisi kiinnittää huomiota parien ohjelmointiosaamisen samankaltaisuuteen myös turvallisen ilmapiirin kannalta. Toisaalta voidaan pohtia myös sitä, kuinka osallistujien kaverisuhteet vaikuttivat tässä tutkimuksessa. Parien aiemmista keskinäisistä suhteista oli aineistossa tietoa vain kahden parin osalta, yhden parin ollen hyviä kavereita keskenään ja toisen



parin maininta siitä, että he eivät ole ennen työskennelleet yhdessä. Tämä puoli jäi tässä tutkimuksessa pimentoon, ja olisi ollut mielenkiintoista tietää ryhmän keskinäisistä suhteista lisää sekä niiden vaikutuksesta esimerkiksi turvalliseksi ja kannustavaksi mainittuun ilmapiiriin.

Ohjaajien läsnäolo ja tuki koettiin oppimista edistäväksi, vaikka parit kysyivätkin apua ohjaajilta vaihtelevasti. Pariohjelmoinnin ohjaaminen näyttäisi siis vaativan ohjaajalta tiettyä sensitiivisyyttä sen suhteen, miten ja milloin hän auttaa opiskelijoita projektin tai tehtävän toteuttamisessa. Toisaalta pareille tulee antaa tilaa toimia ja ratkoa ongelmia keskenään, mutta toisaalta tämän tutkimuksen aineistossa ohjaajia kiitettiin oikea-aikaisesta tuesta. Tässä tutkimuksessa käytetty tapa tarjota parille apua vain heidän kysyessä sitä vaikutti aineiston perusteella toimivalta tavalla ja se voisi olla hyvä lähtökohta pariohjelmointiharjoituksia toteutettaessa opettajaopiskelijoille. Tällöin kuitenkin pariohjelmoinnin ohjaajan tulee varmistua siitä, että esimerkiksi ryhmän ilmapiiri on sellainen, jossa kaikki osallistujat uskaltavat tuoda esiin näkemyksensä ja kysyä tukea.

Haastatteluissa osallistujat toivat esiin tarpeen riittävän avoimille ja sopivan haastaville tehtäville, jotta yhteistyö koetaan merkityksellisenä. Vastaavasti osa opiskelijoista koki liian helpot ja vaihe vaiheelta käydyt tehtävät turhauttaviksi ja oppimista haastaneiksi. Siltä osin tämä tulos tukee ja on linjassa Bryantin ym. (2006) tekemien havaintojen kanssa siitä, että pariohjelmoitaessa yhteistyön määrä on suurempaa haastavissa tehtävissä. Myös Hawlitschekin ym. (2022) suosittelee pariohjelmointiin riittävän laajoja ja tarpeeksi vaativia tehtäviä, jotta parin yhteistyö on tarkoituksenmukaista. Osa haastateltavista kertoi yhteistyön jääneen ensimmäisessä tehtävässä vähäiseksi ja vuorovaikutuksen lähinnä toteavaksi. Toisaalta osa haastateltavista kertoi kokeneensa alun vahvasti opettajajohtoisena osion hyödylliseksi työskentelyn kannalta, vaikka siinä parien välinen yhteistyö jäikin haastateltavien mukaan pintapuoliseksi. Mayerin ja Johnsonsonin (2010) mukaan kokemattomat ohjelmoijat tarvitsevatkin ohjeistusta ja tukea oppiakseen ohjelmoinnin peruseräatteen. Tällöin tämän ja aiemman tutkimuksen perusteella voisi olla mielekästä toteuttaa

opettajajohtoinen osuus, esimerkiksi Use-vaihe yksin, jonka jälkeen joko Modify tai Create-vaiheessa vaihdettaisiin työskentelymuoto pariohjelmointiin. Tehtävien suunnittelu vaatii harjoitteen ohjaajalta myös huomiota ja tietoa opiskelijoiden aiempaan osaamiseen, jotta tehtävät eivät ole liian helppoja taikka vaativia. Kuten tässäkin tutkimuksessa, opiskelijoiden osaaminen kuitenkin vaihtelee. Siksi useat vaihtoehdot ja vapaus valita tehtäviä etenkin Modify ja Create-vaiheissa näytti toimivan hyvin tässä tutkimuksessa. Opettajaopiskelijoiden ohjelmointiharjoitteissa tulisikin täten antaa opiskelijoille riittävästi vapautta valita itselleen sopivan tasoisia tehtäviä.

Haastatteluaineistoon perehtyessä nousee myös esiin kysymys siitä, mitä jää sanomatta. Oppimisympäristöön liittyviä mainintoja oli vain vähän ja esimerkiksi vain yksi pari kertoi muiden osallistujien tuoneen häiriötä omaan toimintaan. Voiko muiden mainintojen puutteesta päätellä fyysisen toimintaympäristön ja välineiden toimineen siis hyvin tämänkaltaisessa työskentelyssä, vai oliko niillä sen suurempaa merkitystä. Tietenkään tästä ei mainintojen vähyydestä ei voi vetää tarkkoja johtopäätöksiä, mutta siltikin asia jää kiinnostamaan.

Kokonaisuudessaan pariohjelmointi voisi tämän tutkimuksen perusteella toimia osaltaan työskentelytapana ohjelmoinnin harjoittelussa opettajaopiskelijoilla. Jo toteutusta suunniteltaessa tulee kuitenkin kiinnittää huomiota edellä käsiteltyihin teemoihin, jotta pariohjelmoinnilla on parhaat mahdollisuudet toimia työskentelymuotona.

## **6.2 Tutkimuksen arviointi**

Tutkimuksen luotettavuuteen liittyy olennaisesti aineiston keruutekniikat sekä aineiston analyysin menetelmät (Tuomi & Sarajärvi, 2018). Tutkimuksen luotettavuutta lisää laajat aineistoesimerkit sekä avoimuus luokittelussa (Hirsjärvi ym., 2009). Laadullisessa tutkimuksessa olisi luotettavuuden kannalta hyvä, että kaksi tutkijaa luokittelisi samaa aineistoa (Tuomi & Sarajärvi, 2018). Tämä ei kuitenkaan ollut mahdollista täysin määräisesti, mutta luokiteltuani

yhden haastattelun toinen tutkija perehtyi luokitteluun ja antoi omat kommenttinsa. Lisäksi tutkimuksen päätelmiä on perusteltu laajoilla aineistoesimerkeillä.

Aineistonkeruumenetelmänä puolistrukturoitu haastattelu mahdollistaa muun muassa myös sellaisten näkökulmien nostamisen esiin, joita haastattelua suunniteltaessa ei osattu odottaa (Hirsjärvi ym., 2009). Kysymykset olivat varsin avoimia, mikä mahdollistaa syvemmän tiedon saavuttamisen haastattelukohteesta (Patton, 2015). Haastattelurungon testaaminen etukäteen olisi hyvä tapa varmistua kysymysten toimivuudesta (Hirsjärvi ym., 2009). Tässä tutkimuksessa haastattelurunkoa ei kuitenkaan pystytty aikataulullisista syistä johtuen testaamaan etukäteen. Haastattelussa käytetty haastattelurunko toimi hyvin tässä tutkimuksessa, etenkin kun haastattelut toteutettiin kolmen tutkijan toimesta.

Parihaastattelun etuna oli tässä tutkimuksessa se, että haastateltavat pystyivät tarttumaan toisen esittämiin näkemyksiin ja jakamaan muistinvaraisesta aiheesta omia kokemuksiaan (Hirsjärvi ym., 2009). Kuitenkin voidaan pohtia sitä, kuinka tarkasti osallistujat pystyvät refleктоimaan sitä, mikä todellisuudessa on haastanut heidän. Haastateltavilla on myös yleisesti tapana antaa sosiaalisesti hyväksyttäviä vastauksia (Hirsjärvi ym., 2009). On siis syytä pohtia sitä, haluavatko osallistujat kertoa parihaastattelussa oppimiseen vaikuttaneita tekijöitä, mikäli ne jollain tavalla liittyvät parin toiseen osapuoleen. Haastatteluissa osallistujat kuitenkin toivat esiin myös ristiriitatilanteita.

Aineistosta katosi parin numero kolme näytönkaappaustallenne toisen harjoituksen osalta. Tämän tutkimuksen osalta datan katoamisen merkitys ei ole merkittävä, mutta näyttötallenne olisi ollut todennäköisesti toimivampi tuki ohjelmointikertaan palaamiseen avainhetkien osalta. Tämä tulee huomioida, mikäli samaa aineistoa aiotaan käyttää myöhemmissä tutkimuksissa.

Tutkimuksen osallistujat koostuivat Opettajankoulutuslaitoksen valinnaisen opintokokonaisuuden opiskelijoista. Opiskelijat olivat siis itse valinneet opintokokonaisuuden, jolloin tämä saattaa näkyä muun muassa opiskelijoiden suhtautumisessa ohjelmointiin. Osallistujissa kuitenkin oli

ennakkokyselyn ja haastatteluiden perusteella opiskelijoita erilaisilla ohjelmointikokemuksilla ja käsityksillä omista kyvyistään ja tällä tutkimuksella saatiin kartoitettua pariohjelmointiin vaikuttaneita teemoja, sekä opiskelijoiden kokemuksia niiden vaikutuksesta koettuun oppimiseen. Osallistujien määrä on myös verrattain pieni, jolloin aineistosta ei voida tehdä kovinkaan yleistettäviä johtopäätöksiä. Tutkimuksen tavoitteen huomioiden tutkittavien ja kerätyn aineiston määrä kuitenkin oli riittävä tämänkaltaiseen tutkimukseen. Tutkimuksen löydöt ovat varsin hyvin siirrettävissä opettajien ohjelmointiosaamisen kehittämiseen opettajien perusopinnoissa.

### 6.3 Jatkotutkimushaasteet

Aihetta olisi mielekästä tutkia yksilöhaastattelujen kautta sekä mitata ohjelmointiosaamisen kehittymistä muilla keinoilla ja mahdollisesti pidemmällä aikavälillä. Tässä tutkimusprojektissa myös kerättiin paljon aineistoa, mutta kaikkea tässä tutkimuksessa hyödynnetty. Esimerkiksi kuvatuista videoista sekä näytönkaappauksista voisi tutkia parien käymää vuorovaikutusta sekä työskentelyn jakaantumista pariohjelmoinnin aikana. Myös avainhetkilomakkeissa kuvattuina hetkiä voisi tarkastella videoilta ja etsiä niistä havaintoja esimerkiksi siitä, millaista vuorovaikutusta niissä on havaittavissa tai oliko opettaja läsnä näissä tilanteissa. Haastatteluaineistossa oli myös havaittavissa ajattelun muutosta ohjelmoinnin opettamista koskien, mutta aihetta tulisi tutkia tarkemmin esimerkiksi ohjelmointia käsittelevän kurssin yhteydessä ja pidemmällä aikavälillä.

Tutkimukseen osallistuneet pohtivat haastatteluissa sekä pariohjelmointiharjoituksen aikana parien jakamista osaamisen. Olisikin mielenkiintoista tutkia sitä, mitä vaikutusta parien taitoerolla on ohjelmoinnin oppimiseen erilaisissa tilanteissa, ja saattaisiko taitoerosta olla mahdollisesti hyötyä tietyissä tilanteissa. Myös parien erilaiset tavat toteuttaa roolijakoa herättävät kysymyksen siitä, onko kuskin ja kartturin oppimisella eroa

pariohjelmoitaessa? Entä onko ohjelmointikokemuksella yhteyttä ohjelmoinnin ja ohjelmoinnillisen ajattelun oppimiseen eri rooleissa.

Comania ym. (2014) mukailten ohjelmoinnin harjoittelussa tulisi jättää tilaa myös informaalille ja suunnittelemattomalle pariohjelmoinnille yllättävien pulmien esiintyessä. Esimerkiksi opiskelijat voisivat toteuttaa omia projektejaan, mutta kuitenkin niin, että pystyvät tarvittaessa ratkaisemaan yhdessä esiin nousevia ongelmia. Tällöin jokainen toisaalta toteuttaisi itse projektiaan, mutta samalla saisi tarvittaessa tukea muilta opiskelijoilta. Olisikin mielenkiintoista tutkia tällaisen työskentelytavan toimivuutta ohjelmoinnin harjoittelussa opettajaopiskelijoiden kohdalla, verrattuna esimerkiksi kokonaan itsenäiseen ohjelmointiin ja niin sanotusti ennalta määrättyyn pariohjelmointiin.

Tämä tutkimus osoittaa, että opettajaksi opiskelevien ohjelmointiosaamista voidaan harjoittaa pariohjelmoinnin avulla. Pariohjelmoitaessa osallistujat sanallistivat toimintaansa, tukivat toisiaan sekä samalla kuitenkin harjoitteen ohjaajan tulee kiinnittää huomiota niin tässä, kuin myös aiemmassa tutkimuksessa osoitettuihin teemoihin, kuten pariohjelmointiin soveltuviin tehtäviin, pariin aiempaan osaamiseen sekä roolijakoon. Nämä tekijät saattavat vaikuttaa koko pariohjelmoitiharjoitteen onnistumiseen. Ohjelmoinnin tehokasta ja tarkoituksenmukaista kouluttamista opettajaopiskelijoille tulee kuitenkin vielä tutkia lisää, jotta opettajien koulutusta voidaan kehittää edelleen vastaamaan selvää tarvetta opettajien ohjelmoinnillisen osaamisen lisäämiselle. On kaikkien edun mukaista, että opettajien koulutuksessa käytettävissä oleva aika ja resurssit käytetään tehokkaasti, toimivaksi tutkittuja tapoja käyttäen.

## LÄHTEET

- Aho, A. V. (2012). Computation and Computational Thinking. *Computer journal*, 55(7), 832-835. <https://doi.org/10.1093/comjnl/bxs074>
- Balanskat, A. & Engelhardt. (2015). *Computing our future: Computer programming and coding – Priorities, school curricula and initiatives across Europe*. European Schoolnet.
- Barkley, E. F., Major, C. F. & Cross, K. P. (2014). *Collaborative Learning Techniques*. Jossey-Bass.
- Blackwell, A. F. (2002). What is programming? Teoksessa J. Kuljis, L. Baldwin & R. Scoble (toim.). *Proc. PPIG 14* (s. 204–2018).
- Bryant, S., Romero, P. & Du Bolay, B. (2006). The collaborative nature of pair programming. Teoksessa P. Abrahamsson, M. Marchesi & G. Succi (toim.), *Extreme programming and agile processes in software engineering* (s. 53–64). Springer. [https://doi.org/10.1007/11774129\\_6](https://doi.org/10.1007/11774129_6)
- Campe, S., Denner, J., Green, E., & Torres, D. (2020). Pair programming in middle school: Variations in interactions and behaviors. *Computer Science Education*, 30(1), 22–46. <https://doi.org/10.1080/08993408.2019.1648119>
- Denning, P. & Tedre, M. (2019). *Computational thinking*. MIT Press Ltd.
- Dillenbourg, P. (1999). *Collaborative Learning : Cognitive and Computational Approaches*. Amsterdam: Pergamon Press.
- Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2020). Assessing 4th grade students' computational thinking through Scratch programming projects. *Informatics in Education*, 19(4), 611–640. <https://doi.org/10.15388/infedu.2020.27>
- Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021a). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 29(1), 12–28. <https://doi.org/10.1002/cae.22255>

- Fagerlund, J. (2021b). *Teaching, learning and assessing computational thinking through programming with Scratch in primary schools*. Jyväskylän yliopisto. Opettajankoulutuslaitos. Väitöskirja.
- Franklin, D., Coenraad, M., Palmer, J., EATINGER, D., ZIPP, A., ANAYA, M., WHITE, M., PHAM, H., GÖKDEMİR, O. & WEINTROP, D. (2020). An Analysis of Use-Modify-Create Pedagogical Approach's Success in Balancing Structure and Student Agency. Teoksessa A. Robins, A. Moskal, A. J. Ko & R. McCauey (toim.), *ICER '20: Proceedings of the 2020 ACM Conference on international Computing Education Research* (s. 14–24). Association for Computing Machinery.
- García-Penalvo, F. J. & Mendes, A. J. (2017). Exploring the computational thinking effects in pre-university education. *Computers in human behavior*, 80, 407–411. <https://doi.org/10.1016/j.chb.2017.12.005>
- Hawlitshchek, A., Berndt, S., & Schulz, S. (2022). Empirical research on pair programming in higher education: A literature review. *Computer Science Education*, 1–29. <https://doi.org/10.1080/08993408.2022.2039504>
- Hirsjärvi, S., Remes, P. & Sajavaara, P. (2009). *Tutki ja kirjoita*. Kustannusosakeyhtiö Tammi.
- Hsu, T.-C., Chang, S.-C. & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers and education*, 126, 296-310. <https://doi.org/10.1016/j.compedu.2018.07.004>
- Höfer, A. (2008). Video analysis of pair programming. *APOS '08: Proceedings of the 2008 international workshop on Scrutinizing agile practices or shoot-out at the agile corral*, 37–41. <https://doi.org/10.1145/1370143.1370151>
- Ilola, M. & Pukkila, N. (10.11.2019). *Scratch kilpajuoksusokkelo-peli*. Haettu 14.2.2022 osoitteesta <https://www.innokas.fi/wp-content/uploads/2019/11/Scratch-kilpajuoksusokkelopeli.pdf>
- Karakainen, M.-T., Karakainen, S.-S., Tanhua-Piironen, E., Viteli, J., Syvänen, A. & Kivinen, A. (2017). *Digiajan peruskoulu 2017 – Tilanearvio ja*

toimenpidesuosituksset. *Valtioneuvoston selvitys- ja tutkimustoiminnan julkaisusarja 72/2017*.

- Korhonen, T. (2017). *Kodin ja koulun digitaalinen kumppanuus*. Väitöskirja, Helsingin yliopisto.
- Lavy, S. (2017). Who benefits from group work in higher education?: An attachment theory perspective. *Higher education* 73(2), 175–187.
- Lindberg, R. S. N., Laine, T. H. & Haaranen, L. (2019). Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games. *British Journal of Educational Technology*, 50(4), 1979–1995. <https://doi.org/10.1111/bjet.12685>
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education*, 10(4), 1–15. <https://doi.org/10.1145/1868358.1868363>
- Mayer, R. E., & Johnson, C. I. (2010). Adding Instructional Features That Promote Learning in a Game-Like Environment. *Journal of Educational Computing Research*, 42(3), 241–265. <https://doi.org/10.2190/EC.42.3.a>
- Mäkelä, K. (2005). Laadullisen sosiaalitutkimuksen eettinen sääätely. Teoksessa *Sosiaali- ja terveysalan tutkimus- ja kehittämiskeskus. Laadullisen sosiaalitutkimuksen eettiset kysymykset : kutsuseminaari 2.5.2005*. (s. 9–20). Helsinki: Sosiaali- ja terveysalan tutkimus- ja kehittämiskeskus.
- OECD. (2017). *PISA 2015 results (Volume V – Collaborative Problem Solving)*. PISA, OECD Publishing, Paris. <https://doi.org/10.1787/9789264285521-en>
- Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A. & Lahmine, S. (2014). Learning Basic Programming Concepts By Creating Games With Scratch Programming Environment. *Procedia - Social and Behavioral Sciences* 191, 1479-1482. <https://doi.org/10.1016/j.sbspro.2015.04.224>
- Opetushallitus. (2014). *Perusopetuksen opetussuunnitelman perusteet*. Määräykset ja ohjeet 96. Opetushallitus
- Patton, M. Q. (2015). *Qualitative Research & Evaluation Methods*. (4. painos). Thousand Oaks, Kalifornia: Sage Publications.



- Ploetzer, R., Dillenbourg, P., Preier, M. & Traum, D. (1999). Learning by Explaining to Oneself and Others. Teoksessa P. Dillenbourg (toim.), *Collaborative Learning : Cognitive and Computational Approaches* (s. 103–121). Amsterdam: Pergamon Press.
- Tuomi, J. & Sarajärvi, A. (2018). *Laadullinen tutkimus ja sisällönanalyysi*. Helsinki: Kustannusosakeyhtiö Tammi.
- Tsan, J., Vandenberg, J., Zakaria, Z., Boulden, D. C., Lynch, C., Wiebe, E. & Boyer, K. E. (2021). Collaborative Dialogue and Types of Conflict: An Analysis of Pair Programming Interactions between Upper Elementary Students. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21)*. 1184-1190.  
<https://doi.org/10.1145/3408877.3432406>
- Tsan, J., Vandenberg, J., Zakaria, Z., Wiggins, J. B., Webber, A. R., Bradbury, A., Lynch, C., Wiebe, E. & Boyer, K. E. (2020). A Comparison of Two Pair Programming Configurations for Upper Elementary Students. *SIGCSE '20: Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*, 346-352. <https://doi.org/10.1145/3328778.3366941>
- Scratch. (4.4.2022). *Tietoa Scratchista*. <https://scratch.mit.edu/about>
- Zhong, B., Wang, Q., & Chen, J. (2016). The impact of social factors on pair programming in a primary school. *Computers in Human Behavior*, 64, 423–431. <https://doi.org/10.1016/j.chb.2016.07.017>
- Wei, X., Lin, L., Meng, N., Tan, W., Kong, S-C. & Kinshuk. (2021). The effectiveness of partial pair programming on elementary school students' Computational Thinking skills and self-efficacy. *Computers & Education*, 160, 104023. <https://doi.org/10.1016/j.compedu.2020.104023>
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- Wing, J. (22.3.2011). A definition of Computational Thinking from Jeanette Wing. *Computing Education Research Blog*.  
<https://computinged.wordpress.com/2011/03/22/a-definition-of-computational-thinking-from-jeanette-wing/>

## LIITTEET

### Liite 1 Ennakkotehtävän kysymykset

1. Mitä mielestäsi ohjelmointi tarkoittaa? (esim. Mitä se on? Mitä sillä tehdään ja miksi? Missä kaikkialla sitä on?)
2. Millaisena näet ohjelmoinnin merkityksen perusopetuksessa? (esim. Pitäisikö kaikkien oppilaiden oppia ohjelmointia koulussa, miksi/miksi ei? Mitä ohjelmoinnin kautta oppii?)
3. Mitä tulee mieleesi, kun kuulet sanan "pariohjelmointi"?
4. Millaisia aiempia kokemuksia sinulla on seuraavista asioista?
  - a) ohjelmoiminen
  - b) ohjelmoinnin oppiminen/opiskelu
  - c) pariohjelmointi
5. Millaisena näet/näkisit itsesi tällä hetkellä seuraavissa rooleissa?
  - a) ohjelmoijana
  - b) ohjelmoinnin oppijana
  - c) parityöskentelijänä ohjelmoinnissa
6. Millaisia tunteita ja tuntemuksia herättävät seuraavat asiat?
  - a) ohjelmointi
  - b) ohjelmoinnin oppiminen/opiskelu
  - c) pariohjelmointi
7. Millaisia aiempia kokemuksia sinulla on ohjelmoinnin opettamisesta?
8. Millaisena näet/näkisit itsesi ohjelmoinnin opettajana?

## Liite 2 Pariohjelmoinnin reflektointi

**Nimi:** [NIMI]  
**Pari:** [NUMERO]

Käy läpi omia pariohjelmointivideoita itsenäisesti. **Luettelo ja kuvaa** alla olevaan taulukkoon, millaisia (isoja tai pieniä) **avainhetkiä** muistat tai havaitset, jotka vaikuttivat:

- oppimiseesi tai ymmärrykseesi
- Scratch-projektienne kehittämiseen
- pariohjelmointiin
- pariohjelmointinne antamaasi henkilökohtaiseen panokseen tai siihen sitoutumiseen
- johonkin muuhun tärkeään asiaan

**Kuvaile mahdollisimman monipuolisesti mm.:**

- mitä tapahtui?
- mikä mielestäsi aiheutti tämän hetken?
- miten koit tämän hetken?
- miten tämä hetki vaikutti yllä mainittuihin seikkoihin?

Videon nimi + aikaleima	A. Kuvaus hetkestä, joka TUKI / OLI HYÖDYKSI / tms.	B. Kuvaus hetkestä, joka HAASTOI / SAI AIKAAN RISTIRIITAA / tms.

(lisää uusia rivejä tarvittaessa)

### **Liite 3 Parihaastattelun runko**

#### **Yleistä/lämmittely**

10. Mitä tapahtui pariohjelmointitunnin aikana? Kertokaa omin sanoin. Esim. mitä harjoituksia teitte, miten työskentely mielestänne sujui.

#### **Pariohjelmointi**

11. Kuvailkaa, miten työskentelitte yhdessä. Esim. miten aloititte? Muuttuiko yhteistyönne työskentelyn aikana?
12. Kuvailkaa, miten jaotte tehtävät/roolit.
13. Kuvailkaa, miten teitte yhdessä päätöksiä ohjelmoinnin aikana esim. eri tilanteissa, tehtävissä, tehtäväkohdissa.

#### **Opettajat/tehtävät (tutorial, debuggaustehtävät, oma peli)/ympäristö**

14. Mikä opettajien toiminnassa lisäsi yhteistyötänne, oppimistanne tai sitoutumista pariohjelmointiin? Mikä vähensi?
15. Mikä tehdyissä tehtävissä lisäsi yhteistyötänne, oppimistanne tai sitoutumistanne pariohjelmointiin? Mikä vähensi?
16. Miten oppimisympäristö (esim. muut ihmiset, käytettävissä olevat resurssit) lisäsi yhteistyötänne, oppimistanne tai sitoutumista pariohjelmointiin? Mikä vähensi?

#### **Oma opettajuus**

17. Mitä kaikkiaan opitte pariohjelmoinnista tai pariohjelmoinnin ohjaamisesta?
18. Muuttiko pariohjelmointikokemuksenne käsityksiänne ohjelmoinnin opetuksesta tai ohjaamisesta?
19. Mitä opettajan tulisi mielestänne keskeisesti huomioida ohjelmoinnin opetuksessa?
20. Mitä opettajan olisi mielestänne hyvä osata ohjelmoinnissa kyetäkseen ohjaamaan oppilaiden (pari)ohjelmoinnin oppimista?

#### **Jälkeen**

21. Mitä olisitte tarvinneet lisää yhteistyönne tehostamiseksi?
22. Mistä kokisitte, että teidän olisi tulevana opettajina tärkeä oppia lisää?
23. Haluatteko nostaa esiin jotain muuta, esim. muistiinpanoistanne?