

*Viia
Ranta
2020*

KUVVIS



ohjelmointi

*Opettajien kokemuksia kuvataiteen ja
ohjelmoinnin yhdistämisestä peruskoulussa*

Kuvis ♥ ohjelmointi

*Opettajien kokemuksia kuvataiteen ja ohjelmoinnin
yhdistämisestä peruskoulussa*

Viia Ranta 2020

Taiteen maisterin opinnäyte, 72 s.

Kuvataidekasvatuksen koulutusohjelma. Taiteen laitos

Aalto-yliopiston taiteiden ja suunnittelun korkeakoulu

Ohjaaja Tomi Slotte Dufva

Tekijä Viia Ranta

Työn nimi Kuvien :heart: ohjelmointi. Opettajien kokemuksia kuvataiteen ja ohjelmoinnin yhdistämisestä peruskoulussa

Laitos Taiteen laitos

Koulutusohjelma Kuvataidekasvatuksen koulutusohjelma

Vuosi 2020

Sivumäärä 72

Kieli suomi

Tiivistelmä

Tutkin taiteen maisterin opinnäytetyössäni, minkälaisia kuvataidetta ja ohjelmointia yhdisteleviä opetusprojekteja on toteutettu peruskouluissa uuden peruskoulun opetussuunnitelman (POPS 2014) ilmestymisen jälkeen. Toteutuneiden opetusprojektien esittelemisen lisäksi tarkastelen opettajien kokemuksia ohjelmoinnin ja kuvataiteen yhdistämisen äärellä.

POPS 2014 toi ohjelmoinnin osaksi opetussuunnitelman laaja-alaisia osaamisen tavoitteita ja mahdollisti ohjelmoinnin yhdistämisen mihin tahansa oppiaineeseen kouluissa. Ohjelmoinnin ja taidekasvatuksen yhdistäminen ei ole mikään uusi avaus: aihetta on aikaisemmin tutkinut mm. Aalto-yliopistossa Tomi Dufva, ja aiheesta on kirjoitettu lukuisia opinnäytteitä. Tietääkseni ohjelmoinnin ja taidekasvatuksen yhdistävää opetusta peruskouluissa ei ollut aikaisemmin tutkittu Suomessa, eli opinnäytteelleni oli olemassa aukko kuvataidekasvatuksen kentällä. Menetelmänäni käytin teemahaastattelua ja aineiston analyysimuotona teemoittelua. Opinnäytteeni on aineistolähtöinen ja vastaan tutkimuskysymyksiini haastatteluja ja kirjallisuutta keskusteluttamalla.

Aineistoni muodostui neljästä haastattelusta, jotka toteutin keväällä 2020. Teemahaastattelin kolmea kuvataideopettajaa sekä yhtä luokanopettajaa heidän tekemistään opetusprojekteista, jotka yhdistelivät ohjelmointisisältöjä kuvataiteeseen. Haastateltavien opetusprojektit toivat yllättävän monipuolisia näkökulmia ohjelmoinnin ja taidekasvatuksen yhdistämiseen: mukana oli useita eri ohjelmointiympäristöjä ja -kieliä hyödyntäviä projekteja, sekä taidekasvatuksellisia sisältöjä mm. pelimaailmasta, taidehistoriasta ja muotoilusta.

Aineiston teemoittelussa nostin esiin kolme teemaa: ymmärtämisen, tuottamisen ja kokemuksellisuuden. Ymmärtämisen teeman kautta käsitelin haastateltavien ajatuksia siitä, minkälaista tiedollista ymmärtämistä ja ajattelun taitoja ohjelmoinnin ja taidekasvatuksen yhdistäminen voi tuottaa. Ymmärtämisen teemassa keskeisiksi aihealueiksi nousivat ohjelmoinnin lukutaito, ohjelmoinnillinen ajattelu ja taiteellinen ajattelu. Tuottamisen teemassa erittelin käytännön tekemiseen liittyviä kokemuksia ja pohdin, minkälaista tuottamista ja tekemistä ohjelmoinnin ja taidekasvatuksen yhdistäminen voi olla. Kokemuksellisuuden teemassa nostin esiin haastateltavien ajatuksia taideoppimisesta kokonaisvaltaisena sekä tunteille alttiina toimintana, sekä tuon esiin ohjelmoinnin ja taidekasvatuksen yhdistämiseen liittyvän yhteistyön ja jakamisen ulottuvuuden.

Opinnäytteeni valossa näyttäisi siltä, että ohjelmointi taidekasvatuksen kontekstissa voi palvella hyvin monenlaisia oppimistavoitteita. Vaikka opinnäytteeni onkin tapaustutkimuksen omainen, eikä haastateltavien kokemukset ole yleistettävissä koskemaan laajemmin koko kenttää, opinnäytteeni sisältää paljon esimerkkejä siitä, minkälaista ohjelmoinnin ja taidekasvatuksen yhdistäminen voi olla käytännössä. Toivoisin, että opinnäytteeni voisi parhaimmillaan inspiroida kuvataideopettajia rohkeasti tarttumaan ohjelmoinnin ja kuvataiteen yhdistämisen mahdollisuuksiin.

Avainsanat taidekasvatus, ohjelmointi, ohjelmoinnin lukutaito, luova ohjelmointi, monialaiset oppimiskokonaisuudet, teemahaastattelu, teemoittelu

Author Viia Ranta

Title of thesis Art Edu :heart: Programming. Teachers' Experiences in Combining Art Education with Programming in Finnish Primary and Secondary Schools

Department Department of Art

Degree programme MA Degree Programme in Art Education

Year 2020

Number of pages 72

Language Finnish

Abstract

In this Master of Art's thesis, I study interdisciplinary teaching projects that combine visual arts with programming in Finnish primary and secondary schools. In addition to presenting different case examples, I study the teachers' experiences in combining art education with programming.

The new Finnish Curriculum (2014) brings programming into the Curriculum as part of transversal competencies that can be combined with any subject. Combining art education with programming is not a new phenomenon: for example, in Aalto University Tomi Dufva has researched the topic, and many theses have been made on the topic. However, there seems to be no research on the topic in the context of Finnish primary and secondary schools, which prompted me to fill the gap in the research field of Finnish Art Education. I used a thematic interview as my research method, and I analyzed the material using thematic analysis. I attempt to answer my research questions by creating dialog between the interviews and literature.

My material consisted of four interviews I conducted in the Spring of 2020. I interviewed three visual arts teachers as well as one classroom teacher about their teaching projects that combined programming with visual arts. The interviewees' teaching projects brought surprisingly diverse perspectives into the combination of programming and art education: several projects utilizing different programming environments and languages were involved, as well as diverse art education content, e.g. game world, art history and design.

Through the thematic analysis, I found three main themes: *comprehension*, *production* and *experience*. Through the theme of *comprehension*, I addressed the interviewees' thoughts on what kind of cognitive comprehension and thinking skills a combination of programming and art education can produce. Code literacy, computational thinking and artistic thinking emerged as key topics in the theme. In the theme of *production*, I analyzed the experiences related to the practical making and considered what kind of making combination of programming and art education can be in the school context. In the theme of *experience*, I highlighted the interviewees' ideas about art learning as a holistic and emotionally sensitive activity and ponder the aspect of cooperation and sharing.

In the light of my thesis, it would seem that programming in the context of art education can serve a wide variety of learning objectives. Although my thesis is kind of a case study, and the interviewees' experiences cannot be generalized to address the whole field, my thesis contains many examples of what a combination of programming and art education can be in practice. I wish that my thesis could, at its best, inspire art educators to boldly seize the opportunities of combining programming with visual arts.

Keywords Art Education, Programming, Code Literacy, Creative Coding, Interdisciplinary learning modules, semi-structured interview, thematic analysis

Sisällysluettelo

| | |
|--|-----------|
| 1. Johdanto | 8 |
| Kipinä pohdinnoille | 8 |
| Perusteluja aiheelle | 9 |
| Tutkimuskysymys, metodit & tutkimuskenttä | 10 |
| 2. Teoriapohja & kirjallisuus | 12 |
| Mitä on ohjelmointi? | 12 |
| Ohjelmoinnillinen ajattelu | 13 |
| Ohjelmointi yhteiskunnassa | 15 |
| Ohjelmoinnin lukutaito osana monilukutaitoa | 16 |
| Opetussuunnitelma ja ohjelmointi | 17 |
| Metaforat: ohjelmointi laajana ilmiönä | 17 |
| Ohjelmointi kuvataidekasvatuksessa | 20 |
| Luova ohjelmointi | 20 |
| Taidekasvatus ja kriittinen ajattelu | 21 |
| 3. Tutkimusmenetelmä | 23 |
| Tieteenfilosofinen perusta | 23 |
| Teemahaastattelu aineistonkeruun menetelmänä | 23 |
| Haastateltavien hankkiminen | 24 |
| Haastattelujen eettisyydestä | 26 |
| Haastateltavat | 28 |
| Haastattelutilanteet | 30 |
| Teemoittelu | 31 |
| 4. Haastateltavien opetusprojektit | 33 |
| 5. Teemat | 36 |
| Ymmärtäminen | 37 |
| Tuottaminen | 44 |
| Kokemuksellisuus | 54 |
| 6. Loppupäätelmät | 61 |
| Yhteenveto: minkälaista tietoa tutkielma tuotti? | 61 |
| Tutkielman luotettavuudesta | 64 |
| 7. Lähdeluettelo | 66 |
| 8. Liitteet | 69 |
| Haastattelulupa | 69 |
| Haastattelurunko | 70 |

Luvun 5. esimerkit

Luvussa 5. Teemat esittelen tarkemmin joitakin haastatteluissa ilmi tulleita opetusprojekteja analyysini lomassa. Opetusprojektien esittelyt olen nivonut analyysitekstin sekaan, joten tämän sisällysluettelon olisi tarkoitus auttaa lukijaa löytämään opetusprojektien kuvailut helposti.

| | |
|---|-----------|
| ♥ Pelimaailma-konseptin luominen | 41 |
| <i>Esimerkki ohjelmoinnillisesta ajattelusta</i> | |
| ♥ Suomen kultakausi Scratchillä | 45 |
| <i>Esimerkki visuaalisesta ohjelmoinnista</i> | |
| ♥ Karkkiautomaatti | 49 |
| <i>Esimerkki käsillä tekemisestä ja muotoilusta</i> | |
| ♥ Abstraktia taidetta koodilla | 50 |
| <i>Esimerkki luovasta ohjelmoinnista</i> | |
| ♥ Piirtorobotti | 57 |
| <i>Esimerkki yhteistyöstä</i> | |

1. Johdanto

Kipinä pohdinnoille

Kiinnostuin ohjelmoinnista ensimmäisen kerran lapsena, kun isäni opetti minulle vuosituhaten vaihteessa nettisivujen rakentamiseen tarkoitettuja ohjelmointikieliä, HTML:ää, CSS:ää ja JavaScriptiä. Oli jännittävää, miten tietokoneella piirtämistäni kuvista saattoi tehdä interaktiivisia seikkailupelejä liittämällä kuvat koodin avulla osaksi nettisivua. Tiettyä kuvan kohtaa, kuten esimerkiksi ovea, klikkaamalla saattoi siirtyä toiselle sivulle, jolla oli jälleen uusi klikattava kuva, ja näin loin yksinkertaisia pelimaailmoja. En silloin ajatellut ohjelmoivani, saati tekeväni taidetta. Pelillisten nettisivujen tekeminen oli minulle leikkiä, ja ohjelmointi vain tapa toteuttaa mielikuvituksellisia fantasioita.

Löysin kiinnostuksen ohjelmointia kohtaan uudelleen yliopistossa, kun osallistuin ensin Aalto-yliopistossa Tietotekniikan laitoksen järjestämälle Ohjelmointi 1-verkkokurssille, ja sittemmin Helsingin yliopiston avoimille MOOC-verkkokursseille. Scala- ja Java-ohjelmointi poikkesivat huomattavasti aiemmin oppimistani HTML- ja CSS-kielistä, ja huomasin innostuvani ohjelmoinnin tuomista älyllisistä ja taiteen tekemiseen verrattuna selkeistä haasteista. Oli ihanaa, kun pitkästä ajasta oli tarjolla “oikeita” ja “väärä” vastauksia – vaikka saman funktionaalisuuden voikin ohjelmoinnilla monesti toteuttaa monilla eri tavoilla, lopputuloksena on aina ohjelma, joka joko toimii tai ei toimi. Taiteessa ei samalla tavalla ole selkeästi oikeita tai väärä tapoja toimia, vaan asiat ovat tulkinnanvaraisempia.

Jossain vaiheessa havahduin huomaamaan, että olin siirtynyt suorittamaan kurseja miettimättä ollenkaan, mitä keräämälläni ohjelmointitaidolla tekisin. Kurssien läpäiseminen hyvin arvosanoin oli antanut minulle kuvan, että olin hyvä ohjelmoimaan. Silti koin, ettei minulla ollut mitään valmiuksia lähteä toteuttamaan soveltavia taiteellisia projekteja ohjelmoinnin avulla, sillä olin oppinut käyttämään taitoa hyvin rajatussa ympäristössä. Se, että olin lapsena leikkinyt tekemällä pelillisiä nettisivuja, oli opettanut minulle paljon enemmän ohjelmoinnin soveltamisesta taiteelliseen toimintaan kuin yliopistolla käydyt kurssit. Tämä oivallus sytytti kipinän pohdinnoille: pitäisikö ohjelmointia opettaa myös muillakin tavoilla kuin kumulatiivisesti? Jos leikki ja taide yhdistettynä ohjelmointiin voi tuottaa soveltavaa oppimista, voisiko taidekasvatuksen ja ohjelmoinnin yhdistäminen olla laajemminkin toimiva tapa opettaa lapsille ja nuorille ohjelmointia? Lisäksi taidekasvattajana minua alkoi kiinnostaa, mitä ohjelmoinnin ottaminen taidekasvatukseen mukaan voisi opettaa taidekasvatuksesta tai maailmasta?

Perusteluja aiheelle

Ohjelmoinnin merkitystä tulevaisuuden taitona on korostettu mediassa käydyssä keskustelussa 2010-luvulla: se nähdään paitsi Suomen kilpailukykyä parantavana valttina (Jungner, 2015) myös osana tulevaisuustaitoja ja yleissivistystä (Liukas & Mykkänen, 2015). Olivat perustelut ohjelmointitaidon tärkeydelle mitkä tahansa, monet tutkijat ovat sitä mieltä, että koodikielet, algoritmit ja ohjelmointi ovat erottamaton osa nykypäivän ihmisen elämää (mm. Dufva & Dufva, 2016; Knochel & Patton, 2015). Mediassa viime aikoina usein viljelty lause, ”koodi on kaikkialla” tuntuu kuluneelta, mutta kuvaa hyvin tämänhetkistä todellisuutta. Ohjelmointikoodi määrittää kaikkien digitaalisten välineiden ja palveluiden toimintalogiikkaa, ja myös pyrkii vaikuttamaan käyttäytymiseemme esimerkiksi räätälöityjen mainosten ja koukuttavien puhelinapplikaatioiden kautta.

Se, että osaa käyttää puhelintaan, ei vielä takaa, että ymmärtäisi logiikkaa käyttöliittymän takana (Dufva & Dufva, 2016, s. 98). Ohjelmoinnin lukutaito (*code literacy*) on käsite, jonka kautta ohjelmointi rinnastuu lukutaitoon; se on yksi edellytys digitalisoituneen maailman ymmärtämiselle ja siinä täysivaltaisena jäsenenä toimimiselle. Jos yhteiskunta on jakautunut teknologiaa ymmärtäviin ja ymmärtämättömiin ihmisiin, on riskinä, että syntyy digitaalinen kuilu näiden eri ihmisryhmien välille (Dufva & Dufva, 2016, s. 98). Kaikista suomalaisista ei tarvitse tulla ohjelmistoasiantuntijoita, mutta kaikkien olisi hyvä ymmärtää, millä logiikalla heidän päivittäin käyttämänsä laitteet toimivat, jotta he voisivat osallistua digitalisoituneeseen yhteiskuntaan aktiivisina toimijoina. Näkemykseni mukaan digitaalisten teknologioiden ymmärtäminen on siis ennen kaikkea tasa-arvokysymys.

Myös Opetushallitus noteerasi ohjelmoinnin merkityksen nyky-yhteiskunnassamme, kun Perusopetuksen opetussuunnitelman perusteet 2014 toi ohjelmoinnin osaksi peruskouluopetusta 2016 alkaen. Ohjelmointi mainitaan opetussuunnitelmassa opetettavana sisältönä sekä matematiikan että käsityön oppiaineissa, mutta ohjelmointi on huomioituna myös laaja-alaisen osaamisen tavoitteissa osana mm. tieto- ja viestintäteknologista osaamista (Opetushallitus, 2016). Laaja-alaiset osaamisen tavoitteet koskevat kaikkia oppiaineita, joten sikäli ohjelmoinnin opetuksen voi katsoa sivuavan ainakin opetussuunnitelman tasolla kaikkia muitakin oppiaineita, myös kuvataidetta. Laaja-alaiset oppimisen tavoitteet eivät varsinaisesti velvoita ohjelmointisisältöjen ottamista kuvataiteeseen, mutta mahdollistavat sen aiheesta kiinnostuneelle opettajalle. Taidekasvattajan näkökulmasta minua kiinnostaakin ohjelmoinnin ja kuvataiteen yhdistämisen ilmiönä ja mahdollisuutena: minkälaista se voisi ylipäätään olla?

Tutkimuskysymys, metodit & tutkimuskenttä

Lähtökohtana opinnäytteelleni on kysymys: minkälaisia ohjelmoinnin ja taidekasvatuksen yhdistäviä oppimiskokonaisuuksia on tehty peruskouluissa uuden perusopetuksen opetussuunnitelman (myöhemmin POPS 2014) ilmestymisen jälkeen? Tarkoitukseni on kartoittaa erilaisia hankkeita ja kokeiluja, ja niiden perusteella hahmotella kuvaa ohjelmoinnin ja taidekasvatuksen yhdistämisen nykytilasta. Minua kiinnostaa toteutuneiden opetusprojektien esittelemisen lisäksi, miten opettajat ovat kokeneet ohjelmoinnin ja taidekasvatuksen yhdistämisen, miten he perustelevat sen koulukontekstissa, ja minkälaisia mahdollisuuksia oppimiselle he siinä ylipäättään näkevät. Tutkin siis opettajien kokemuksia kuvataiteen ja ohjelmoinnin yhdistämisestä peruskoulukontekstissa.

Tutkimuskysymyksiini vastatakseni olen valinnut keskeiseksi tutkimusmenetelmäkseni teemahaastattelun. Haastattelen opettajia, jotka ovat työssään peruskoulussa kokeilleet ohjelmoinnin yhdistämistä kuvataidekasvatuksen sisältöihin. Haastateltaviksi valikoitui kaksi yläkoulussa opettavaa kuvataideopettajaa, yksi sekä ylä- että alakoulussa opettava kuvataiteen ja käsityön opettaja, sekä yksi luokanopettaja. Vaikka haastateltavia on määrällisesti vähän (yhteensä neljä), olen pyrkinyt valikoimaan haastateltavat niin, että joukossa olisi mahdollisimman erilaisia näkemyksiä taidekasvatuksen ja ohjelmoinnin yhdistämisestä.

Aineiston analyysimuotona käytän teemoittelua. Teemoittelussa nostan esiin haastateltavien haastatteluissa nousseita ajatuksia ja kokemuksia ohjelmoinnin ja taidekasvatuksen yhdistämisestä. Teemoiksi nostamani ymmärtämisen, tuottamisen ja kokemuksellisuuden teemat voivat äkkiseltään tuntua ohjelmoinnista kaukaisilta, mutta teemat ovat kuvataidekasvatukselliselta näkökulmalta katsottuna mielestäni perusteltuja. Teemat myös toistuivat kaikkien haastateltavien haastatteluissa.

Ohjelmoinnin ja taidekasvatuksen yhdistävää opetusta on Aalto-yliopistossa kehitellyt Tomi Dufva (Dufva, 2018), lisäksi aiheen tiimoilta on kirjoitettu useita opinnäytteitä (Rybatzki-Tiensivu, 2017; Saarinen, 2017; Tujula, 2016). Kuvataidekoulujen ja vapaan taidekasvatuksen parissa on myös tiedettävästi tehty ohjelmoinnin ja taidekasvatuksen yhdistäviä kokeiluja, mm. pääkaupunkiseudulla Käsityökoulu Robotin toimesta. Rajaan tutkielmani koskemaan kuitenkin nimenomaan peruskouluja, sillä ei tietääkseni ole olemassa tutkittua tietoa siitä, ovatko peruskoulussa toimivat kuvataideopettajat ottaneet ohjelmoinnillisia sisältöjä osaksi taideopetustaan uuden perusopetuksen opetussuunnitelman ilmestymisen jälkeen. Ohjelmoinnin ja taidekasvatuksen yhdistämisen nykytila kouluissa on siis tutkimatonta maastoa, ja näen, että tutkielmalleni on näin ollen selkeä aukko olemassa kuvataidekasvatuksen kentällä.

Tutkielmani voisi katsoa olevan kartoittava sarja tapaustutkimuksia. Sen tarkoituksena on esitellä kentällä jo olemassa olevia ideoita, nostaa esiin hyviä käytäntöjä kuvataideopettajille, ja myös huomioida mahdollisia haasteita. Haluan kuitenkin huomauttaa, että kyseessä on tapaustutkimuksen (vrt. Saarela-Kinnunen & Eskola, 2015) omainen tutkielma. Vaikka aineistosta nousseet oppimiskokonaisuudet ja -kokeilut ovatkin mahdollisesti sellaisia, joita tutkielmaa lukevat opettajat voisivat käyttää melko pienin muokkauksin omassa opetuksessaan, haastateltavien kokemukset ovat yksittäisten henkilöiden kokemuksia, ja ne eivät ole yleistettävissä koskemaan koko kenttää. Pyrin siis välttämään liian yleistävää kieltä tutkielmaa tehdessäni. En halua vetää liian suoria johtopäätöksiä aineistosta, vaan suhtautua erityisesti haastatteluihin ainutlaatuisina yksittäistapauksina.

Näen oman positioni kirjoittajana aktiivisena tiedon tuottajana, joten oma ääneni luultavasti väkisinkin heijastuu tapaan, jolla analysoin aineistoa. Näkemykseni mukaan tutkimus ei ole koskaan täysin arvovapaata, ja siksi pyrin tuomaan mahdollisimman avoimesti esiin seikat, jotka ovat saattaneet vaikuttaa ajatteluuni tai tulkintaani opinnäytettä tehdessä, sekä kuvailemaan tutkimusprosessin mahdollisimman läpinäkyvästi. Tutkielman eettisyyden varmistamiseksi olen varmistanut, että haastatellut opettajat ovat antaneet luvan siihen, että heidän nimiään ja lainauksiaan saa käyttää opinnäytteessä.

2. Teoriapohja & kirjallisuus

Ohjelmointi on tullut erottamaksi osaksi nykyihmisen elämää, ja uuden perusopetuksen opetussuunnitelman perusteiden (POPS 2014) myötä myös osaksi kouluopetusta. Opetusalalla käydyissä keskusteluissa itse ohjelmointitaitoa tärkeämmäksi taidoksi nostetaan monesti ohjelmoinnillinen ajattelu ja ohjelmoinnin lukutaito (code literacy) osana media- tai monilukutaitoa. Lisäksi taidekasvatus tuo luova ohjelmointi-käsitteen mukaan ohjelmoinnin ympärillä käytävään kasvatuskeskusteluun. Tässä luvussa avaan nämä koulumaailman ja kuvataidekasvatuksen kannalta keskeiset ohjelmointiin liittyvät käsitteet ja diskurssit käyttämäni kirjallisuuden pohjalta. Avaan lopuksi myös lyhyesti ennakkokäsityksiäni taidekasvatuksen luonteesta ja tavoitteista suhteessa ohjelmointiin.

Mitä on ohjelmointi?

Kaiken digitaalisen teknologian ytimessä on jonkinlainen tietokone. En tarkoita nyt pelkästään kaikille tuttuja pöytä-tietokoneita, joissa on yleensä näyttö, näppäimistö, elektroniikkaa sisältävä pöytä ja hiiri vaan tietokonetta laajemmin: autoissa, jääkaapeissa, hisseissä, älypuhelimissa, televisioissa on kaikissa analogisen laitteiston lisäksi ohjelmitavaa digitaalista teknologiaa, ja laajassa merkityksessä niiden voisi ajatella olevan tietokoneita. Jotta tietokoneen saisi toimimaan halutulla tavalla, tarvitaan ohjelmointia; ilman ohjelmointia monet laitteet olisivat vain kasa hyödyöntä elektroniikkaromua.

Tietokone ei ymmärrä ihmisten kieltä suoraan, joten ohjelmoidakseen ohjelmoija käyttää tietokoneen ymmärtämää ohjelmointikieltä; ohjelmointikieliet toimivat ikään kuin ihmisen ja tietokoneen välisenä rajapintana (Heikkinen, 2016). Erilaisia ohjelmointikieliä on satoja, ja ne on jaoteltavissa abstraktiotasonsa mukaan alhaisen tason konekieliin, keskitason symbolisiin ohjelmointikieliin ja korkean tason ohjelmointikieliin (Heikkinen, 2016). Alkeellisimmat konekielet ovat ns. binäärikoodia, ykkösten ja nollien jonoja. Ykköset ja nollat symboloivat tietokoneessa olevia muistipaikkoja, ja konekieliset käskyt ovatkin suoraan siirrettävissä tietokoneen suorittimelle. Korkeamman tason kielet muistuttavat enemmän ihmisten kieliä, usein englantia, mutta pohjimmiltaan nekin on muunnettava takaisin ykkösiksi ja nolliksi, jotta tietokone voisi niitä käsitellä (Dufva, 2018, ss. 27–28). Nykyään yleisesti käytössä olevat ohjelmointikieliet ovat usein korkeamman tason ohjelmointikieliä niiden ihmisille helpomman ymmärrettävyyden takia.

Kieli mielikuvana kuvaa ihan hyvin ohjelmointikielten olemusta: kone käsittelee

ja lukee koodia rivi riviltä, ja rivit muodostavat yhtenäisiä kokonaisuuksia, ohjelmistoja. Kuten ihmisten kielissä, ohjelmointikielissäkin on oma ”kielioppinsa”, syntaksi, joka määrittelee, minkälaisia ilmaisuja voi käyttää ohjelmakoodia kirjoittaessa, jotta viesti menee perille ja ohjelma toimii. Ihmiskielistä poiketen ohjelmointikieliet ovat sikäli armottomia, että ne eivät salli virheitä: pienikin kirjoitusvirhe aiheuttaa syntaksivirheen ja kaataa ohjelman. Mitä alhaisemman tason ohjelmointikielillä operoidaan, sitä tarkempi ohjelmakoodin kirjoittajan pitää olla. Ylemmän tason kielissä on enemmän valmiita rakennuspalikoita, jotka antavat ohjelmoijalle liikkumatilaa ja usein valmiita tapoja kiertää ohjelmointivirheet. Ohjelmoida voi myös visuaalisten ohjelmointiympäristöjen avulla, jolloin ohjelmoijan ei tarvitse itse kirjoittaa varsinaista koodia ollenkaan – esimerkiksi koulukäytössä melko yleinen Scratch (<https://scratch.mit.edu>) antaa ohjelmoijan käyttää valmiita visuaalisia palikoita, joita yhdistelemällä ohjelmoija voi luoda melko monimutkaisiakin sovelluksia.

Ohjelmoinnin synonyyminä käytetään usein myös puhekieliseltä kuulostavaa sanaa koodaus. Koodaus ei eroa ohjelmoinnista mitenkään, anglosaksisista kielistä lainattu sana on suomen kieleen jo hyvin vakiintunut ja se löytyy Kotimaisten kielten keskuksen sanakirjasta ohjelmoinnin synonyymina (Kotus 2020). Käytän opinnäytteessäni molempia sanoja ohjelmointi ja koodaus – sillä erotuksella, että koodauksella viitataan yleensä ”perinteiseen” ohjelmointikoodin kirjoittamiseen, kun taas ohjelmointia käytän laajemmin koskien myös visuaalisia ohjelmointiympäristöjä.

Ohjelmoinnillinen ajattelu

Ohjelmoinnillisen ajattelun (eng. ”computational thinking”) termille ei ole yhtä suomenkielistä vastinetta: lukemissani lähteissä käytetään mm. ohjelmointi-, ohjelmallista, laskennallista, automatisointi- ja algoritmista ajattelua puhuttaessa samasta asiasta. Kaikille suomennoksille tuntuu olevan omat perusteensa, ja on kirjoittajan näkökulmasta sääli, ettei käsitteelle ole vakiintunutta suomennosta. Opetusalan puheenvuoroissa on viime vuosina alettu yhä enemmän käyttää sekä ohjelmoinnillisen ajattelun termiä (mm. Fagerlund, 2020) että ohjelmallista ajattelua (mm. Horila & Tammi, 2017; Jormanainen, 2018). Molemmat ovat mielestäni hyviä käännöksiä, sillä toisin kuin esimerkiksi automatisointiajattelu tai algoritmien ajattelu, ohjelmallisen/ohjelmoinnillisen ajattelun käsitteet voisi laveasti katsottuna pitää sisältään myös laajempia teknologiakasvatukseen sisältyviä teemoja, kuten esimerkiksi ohjelmoinnin lukutaidon. Näistä kahdesta valitsin käyttää nimenomaan ohjelmoinnillista ajattelua Fagerlundin (2020) ja taidekasvatuksen ohjelmointia sivuavien opinnäytteiden (mm. Rybatzki-Tiensivu, 2017) jatkumoksi.

Ohjelmoinnillisen ajattelun käsitteen juuret ovat Kekäleisen (2015) mukaan

jäljitettävissä Seymont Papertin teokseen *Mindstorms* vuodelta 1980. Papert ei kirjassaan nimeä ohjelmoinnillista ajattelua käsitteenä, mutta hän ennustaa, että tulevaisuudessa tietokoneet yleistyvät ja lapsilla on mahdollisuus tietokoneiden avulla kehittää omaa ajatteluaan. Papert kuvaa myös, miten tietokone kehittää ongelmanratkaisukyvyyn kehittymistä ja syy-seuraussuhteiden tunnistamista (Papert, 1985). Myöhemmässä julkaisussaan Papert käyttää ensimmäisen kerran termiä *computational thinking* (Kekäleinen, 2015). Termi ohjelmoinnillinen ajattelu tuli kuitenkin laajaan tietoisuuteen vasta, kun tietojenkäsittelytieteen professori Jeanette M. Wing kirjoitti 2006 artikkelin otsikolla *Computational Thinking*.

Wingin (2006, 2008) mukaan ohjelmoinnillinen ajattelu lähestyy ongelmanratkaisua, systeemien suunnittelua ja ihmisen toiminnan ymmärtämistä tietojenkäsittelytieteen konseptien kautta. Ohjelmoinnillisella ajattelulla ei siis tarkoiteta pelkästään mekaanista ohjelmointitaitoa, vaan se on laajempi, analyttinen ajattelemisen taito, kyky hahmottaa maailma ikään kuin tietojenkäsittelytieteilijä. Ohjelmoinnillinen ajattelu sisältää laajojen ja monimutkaisten ongelmien ratkaisemisen abstraktion, mallinnuksen, arvioinnin, ja osiin purkamisen avulla. Wingin teeseihin kuuluu, että ohjelmoinnillinen ajattelu on nimenomaan ihmisen tapa ajatella, ei tietokoneen. Ohjelmoinnillinen ajattelu on ideatason ajattelua, eikä sen tarvitse olla kytköksissä mihinkään laitteeseen, eikä edes tietokoneeseen. (Wing, 2006, 2008). Ohjelmoinnillinen ajattelu ammentaa matemaattisesta ajattelusta ja luonnontieteistä, mutta sitä eivät koske samat lainalaisuudet: ohjelmoinnillisessa ajattelussa yleistämisen ja abstrahoinnin ei tarvitse olla ”puhdasta” kuten matematiikassa, eikä siinä toisaalta päde fyysikaalisen maailman rajoitukset, kuten luonnonlait (Wing, 2008).

Suomalaisen kasvatustieteiden tutkimuksen kontekstissa mm. Horila & Tammi (2017) ovat jäsenelleet ohjelmoinnillisen ajattelun koostuvan ongelman purkamisesta osiin, kaavojen ja säännönmukaisuuksien tunnistamisesta, algoritmien (ohjeiden ja käskyjen) luomisesta, sekä ratkaisun yleistämisestä ja automatisoinnista (Horila & Tammi, 2017). Ohjelmoinnillisessa ajattelussa tarvittavia ominaisuuksia ovat kyky käsitellä vaikeita asioita, sitkeys, epävarmuuden (ja epämääräisyyden) sietäminen ja kyky toimia avoimien haasteiden parissa. Nämä ominaisuudet ovat taidekasvatuksen tutkijoiden Knochel & Pattonin (2015, s.23) mukaan samoja, mitä taiteen tekemisessä monesti tarvitaan - ohjelmoinnillinen ajattelu ei siten ole välttämättä täysin vastakkaista taiteelliselle ajattelulle, vaikka äkkiseltään saattaisi siltä tuntua.

Ohjelmointi yhteiskunnassa

Ohjelmoinnin merkitystä digitalisoituneessa yhteiskunnassa on korostettu 2010-luvulla käydyssä julkisessa keskustelussa hyvin usein: milloin elinkeinoelämän vaikuttajat ovat huolestuneita Suomen kilpailukyvyistä ja siitä, riittääkö Suomessa tulevaisuudessa tarpeeksi koodareita yritysmaailman tarpeisiin (esim. Jungner, 2015; Liukas & Mykkänen, 2015); milloin yritykset ja tasa-arvoa ajavat järjestöt ovat havahtuneet ohjelmoijien kapeaan diversiteettiin ja haluavat lisää naisia hyvin miehistyneelle alalle (Ukkonen, 2019); milloin ohjelmointitaito nähdään lapsen oikeutena oppia koulussa taitoja, jotka parhaimmillaan edesauttavat hänen työllistymistään tulevaisuudessa (mm. Liukas & Mykkänen, 2015).

Suomen kilpailukyky, ohjelmistoalan työvoimapula ja kapea työntekijäkuva ja lapsen oikeus oppia työelämässä tarvittavia taitoja kuulostavat pohjimmiltaan yritysmaailmasta kumpuavilta perusteluilta tuoda ohjelmoinnin opetus osaksi kouluopetusta. Tutkijat Mertala, Palsa & Dufva (2020) kritisoivat diskursseja, jotka korostavat ohjelmoinnin opetuksen tärkeyttä elinkeinoelämästä kumpuavien perustein. Heidän mukaansa nämä diskurssit keskittyvät usein ohjelmoinnin funktionaaliseen ulottuvuuteen jättäen ohjelmoinnin ja ohjelmakoodin yhteiskunnalliset vaikutukset huomioimatta (Mertala, Palsa, & Dufva, 2020).

Ohjelmointia on kaikkialla, mutta sen yhteiskunnalliset vaikutukset jäävät helposti arjessa huomaamatta. Dufva & Dufvan (2016) mukaan digitaalisuuden vaikutukset ulottuvat päivittäiseen elämäämme ja muokkaavat käyttäytymistämme. Navigaattorina toimiva puhelin taskussa, ostoskäyttäytymistä ennustavat ja siten mainontaa kohdentavat algoritmit ja sosiaalinen media ovat jo muokanneet tapojamme kommunikoida, liikkua ja kuluttaa. On jopa todisteita siitä, että digitaaliset teknologiat muokkaavat ihmisten ajattelua tukemalla, jakamalla ja laajentamalla kognitiivisia kykyjämme (Dufva & Dufva, 2016, s. 98). Pelkkä yksittäisten laitteiden käytön hallitseminen ei kuitenkaan vielä takaa ymmärrystä siitä, miten laite toimii tai millaista koodia sen takana on. Itse asiassa, teknologian käytön hallitseminen ei takaa edes sitä, että laitteita osattaisi käyttää hyödyllisesti – monilla nuorilla on vaikeuksia hakea netistä tietoa ja arvioida sitä (Kiili, 2012). Jos yhteiskunta on jakautunut teknologiaa ymmärtäviin ja ymmärtämättömiin ihmisiin, yhteiskuntaan syntyy pahimmillaan digitaalinen kuilu näiden eri ihmisryhmien välille. Digitaalisten teknologioiden ymmärtäminen on siis myös tasa-arvokysymys yhteiskunnassa. (Dufva & Dufva, 2016)

Ohjelmoinnin lukutaito osana monilukutaitoa

Ohjelmoinnin merkitystä yhteiskunnassa laajemmin, tasa-arvon ja demokratian toimivuuden kannalta jäsentää ajatus ohjelmoinnin lukutaidosta (*code literacy*) osana monilukutaitoa. Dufva & Dufva (2016, s. 98) määrittelevät ohjelmoinnin lukutaidon taidoksi ymmärtää ohjelmointia ja digitaalisuuden yhteiskunnallisia vaikutuksia. Ohjelmoinnin lukutaito ei suoraan tarkoita ohjelmoinnin oppimista perinteisessä merkityksessä, vaan se tarkoittaa pikemminkin ohjelmoinnin, sen tarkoituksien ja kontekstien ymmärtämistä laajemmin. Ajatusta voi avata lukutaidon käsitteen kautta: samaan tapaan kuten kaikista lukutaitoisista ihmisistä ei tule kirjailijoita, ei kaikista ohjelmoinnin lukutaitoisista tule ohjelmistokehittäjiä. Silti lukutaitoisilla ihmisillä on tarpeelliset taidot ja kyvyt voidakseen lukea ja kirjoittaa. (Dufva & Dufva, 2016, s. 98)

Jos koodia ajattelee tekstinä (vrt. Mertala ym., 2020) laajan tekstikäsitteen mukaisesti, tekstin lainalaisuudet alkavat päteä myös ohjelmoidussa maailmassa. Mertalan ja kumppaneiden (2020) mukaan tekstiä määrittää sisällön lisäksi myös se, mitä teksti tuottaa; koodi voi siis toimia tasa-arvoa edistävänä tekijänä tarjoamalla avoimia palveluita ja ajattelun välineitä, mutta sitä voidaan myös käyttää vaikuttamisen, hallinnan ja manipuloinnin välineenä (Mertala ym., 2020). Koodiin, kuten mihin tahansa tekstiin, kietoutuu siis väkisin kysymyksiä vallasta, ja nämä aspektit jäävät näkymättömiin, mikäli ohjelmointiin suhtaudutaan pelkästään arvoneutraalina ongelmanratkaisun välineenä.

Mertala, Palsa & Dufva (2020) esittävät, että ohjelmoinnin ja siihen liittyvien ilmiöiden käsittelyä kouluissa voisi monipuolistaa liittämällä ohjelmoinnin opetukseen monilukutaidon pedagogiikkaa. Monilukutaito määritellään POPSissa (Opetushallitus, 2016, ss. 22–23) erilaisten tekstien tuottamisen, tulkitsemisen ja arvottamisen taidoksi. Monilukutaitoon sisältyy laajan tekstikäsitteen mukaisesti mm. visuaalinen, audiitiivinen, numeerinen ja kineettinen tieto (ja näiden yhdistelmät). Monilukutaito siis laajentaa lukutaidon käsitteen koskemaan myös muutakin kuin painettua sanaa. Monilukutaidon pedagogiikan yhtenä tavoitteena on auttaa oppilasta tulkitsemaan ja hahmottamaan monikanavaista ja -mediaista maailmaa ympärillään. Hahmottamisen lisäksi monilukutaito pyrkii vahvistamaan oppilaan kriittisen ajattelun taitoja, eettistä ja esteettistä arvottamista. (Opetushallitus, 2016).

Mertala ja kumppanit esittävät, että monilukutaidon näkökulmasta yksi ohjelmoinnin opetuksen ydintehtävistä on tarjota oppilaille työkaluja kriittiseen toimijuuteen koodia ja algoritmeja vilisevässä maailmassa. Keinona siihen he näkevät, että ohjelmointiin, algoritmiin käytänteisiin ja teknologioihin liittyviä epäkohtia pitäisi tehdä näkyviksi oppilaille ja käsitellä heidän kanssaan opetuksessa. (Mertala ym., 2020, s. 30)

Opetussuunnitelma ja ohjelmointi

Vuonna 2016 voimaan astunut opetussuunnitelma, POPS 2014 otti ohjelmoinnin osaksi peruskouluopetusta (Opetushallitus, 2016). Ohjelmointi ei ole koulun tuntijaossa oma oppiaineensa, vaan sitä opetetaan ensisijaisesti osana matematiikan ja käsityön oppiaineita. Lisäksi ohjelmointi mainitaan laaja-alaisen oppimisen tavoitteissa tieto- ja viestintäteknologisen osaamisen yhtenä osana (Opetushallitus, 2016). Laaja-alaisen oppimisen tavoitteet koskevat kaikkia oppiaineita, mutta OPS:issa ei ole määritelty, miten tai kuinka paljon mitään sisältöjä opettajan pitäisi omaan opetukseensa sisällyttää. Laaja-alaisen tavoitteiden myötä ohjelmoinnin voi katsoa opetussuunnitelman tasolla koskettavan löyhästi kaikkia oppiaineita, myös kuvataidetta.

Ohjelmointi mainitaan opetussuunnitelmana osana laaja-alaisen osaamisen tavoitteita, ja niissä se linkittyy seuraaviin osaamisalueisiin: tieto- ja viestintäteknologian osaamiseen, oppimaan oppimiseen ja ajattelun taitoihin, yrittäjyyteen ja työelämätaitoihin sekä monilukutaitoon. Mertala, Palsa & Dufva korostavat erityisesti viimeisen, monilukutaidon, merkitystä, sillä se ottaa muista osa-alueista poiketen huomioon myös ohjelmoinnin yhteiskunnalliset, poliittiset ja sosiaaliset ulottuvuudet. Monilukutaito on Mertalan ja kumppaneiden mukaan jäänyt OPSissa ja opetuksessa vähemmälle huomiolle kuin muut osaamisalueet, luultavasti siksi, ettei ohjelmoinnin ja monilukutaidon välistä yhteyttä selitetä OPSissa sen kummemmin (Mertala ym., 2020, ss. 27–28).

Metaforat: ohjelmointi laajana ilmiönä

Suhtaudun ohjelmointiin laajana ilmiönä, johon liittyy paljon muutakin kuin pelkkä näennäisen neutraali koodin kirjoittamisen tapahtuma. Tomi ja Mikko Dufva (2016) ovat kehittäneet yhdeksän metaforaa ohjelmoinnin moniulotteisuuden hahmottamiseksi: kone, organismi, aivot, muutos, kulttuuri, poliittinen järjestelmä, henkinen vankila, vallan väline ja karnevaali¹. Metaforista neljä ensimmäistä edustavat funktionaalista lähestymistapaa ohjelmointiin; tulkitsevaa näkökulmaa edustavat kulttuuri ja poliittinen järjestelmä; emansipatorista näkökulmaa henkinen vankila ja vallan väline; postmodernia lähestymistapaa kuvaa karnevaali (Dufva & Dufva, 2016). Viittaa ohjelmoinnin metaforiin toisinaan myöhemmin analyysiosiossa, joten lienee aiheellista avata ne lukijalle.

Funktionaaliset metaforat (kone, organismi, aivot ja muutos) ovat olleet pitkään perinteisin tapa hahmottaa ohjelmointia. Niiden kautta ohjelmointi näyttäytyy

¹ Vapaat suomennokset. Alkuperäiset englanninkieliset nimet ovat: *Machine, Organism, Brain, Flux & Transformation, Culture, Political System, Psychic Prison, Instrument of domination, Carnival*. (Dufva & Dufva, 2016)

arvovapaana ongelmanratkaisun välineenä. Mertala ja kumppanit (2020) ottavat esiin, että myös koulumaailmassa funktionaalinen paradigma on ollut voimakkaasti esillä esimerkiksi opettajille tarjottavassa täydennyskoulutuksessa, sekä siinä, että ohjelmointi nähdään ensisijaisesti työmarkkinoilla tarvittavana taitona ja koulussa opetettavien ohjelmointikielten tulisi olla linjassa työelämässä tarvittavien kielten kanssa (Mertala ym., 2020, s. 26).

Esimerkiksi kone -metaforan kautta ohjelmointi näyttyy pelkästään lineaarisena joukkona komentoja, jotka syötetään tietokoneelle, jotta tietokone voi suorittaa komennot konelogiikan mukaisessa järjestyksessä. Koodin tarkoituksena on vain saada aikaiseksi haluttu ratkaisu, eikä koodiin liity muita arvokeskusteluja kuin “mikä olisi paras koodikieli ratkaista ongelma x”. Organismi-metafora muistuttaa kone-metaforaa sillä erotuksella, että koodi nähdään ei-lineaarisenä, yksittäisistä soluista koostuvana kokonaisuutena: ohjelmointikieliin liittyvistä paradigmoista organismi-metaforan voisi katsoa edustavan ns. olio-ohjelmoinnin paradigmaa, kun taas kone edustaa funktionaalista paradigmaa. Aivot-metafora näkee koodin ihmisen luomina aivoina: koodi ei vain järjestele ja käsittele tietoa, vaan myös aktiivisesti tuottaa sitä. Esimerkiksi tekoäly ja koneoppiminen ovat esimerkkejä aivot-metaforaan liittyvistä käytännön sovellutuksista. Aivot-metafora ei tuo esiin kriittisiä ääniä, joita tekoälyyn liittyy, vaan esittää tekoälyn olevan evoluution tulosta. Muutos-metafora näkee ohjelmoinnin mullistavana ratkaisuna maailman ongelmiin, mutta ei kyseenalaista, onko muutokselle todella tarvetta. (Dufva & Dufva, 2016, ss. 100–102). Yhteistä kaikille funktionaalisten näkökulman metaforille on, että ohjelmointi typistetään matematiikan kaltaiseksi, arvoneutraaliksi ongelmanrakaisuksi, johon ei liity esimerkiksi kysymyksiä vallasta, saavutettavuudesta, tai ohjelmointiin liittyvistä laajemmista ilmiöistä.

Tulkitsevampaa näkemystä ohjelmointiin edustaa Dufva & Dufvan (2016) mukaan kulttuuri-metarofa, joka näkee ohjelmoinnin kulttuuria muokkaavan vaikutuksen. Koodi ei ole vain ohjeiden antamista tietokoneelle, vaan ohjelmointi ilmiönä synnyttää omia alakulttuurejaan ja siten vaikuttaa myös valtakulttuuriin. Esimerkiksi avoimen lähdekoodin ja hakkerikulttuurin ajatukset liittyvät kulttuuri-metaforaan. Koodi poliittisena järjestelmänä-metafora liittyy kulttuuri-metaforaan nähden koodin kulttuuria muokkaavan vaikutuksen, mutta se ottaa mukaan poliittisen ulottuvuuden. Internetin myötä koodi vaikuttaa yhä useampien ihmisten elämään, ja kysymykset siitä, kuka saa määritellä ja hallita koodia, ja miksi sitä ylipäätään pitäisi hallita, nousevat poliittinen järjestelmä-metaforan ytimeen. (Dufva & Dufva, 2016, s. 103).

Dufva & Dufvan (2016) emansipatorisiksi lähestymistavoiksi nimeämät metaforat (psykykinen vankila ja vallan väline) risteävät osittain tulkitsevien näkökulmien

kanssa, mutta emansipatoriset näkökulmat keskittyvät enemmän kysymyksiin vallasta, sosiaalisesta oikeudenmukaisuudesta niin yksilön kuin yhteiskunnankin tasoilta. Metaforat tuovat näkyväksi ihmisen ja koodien väliset valtasuhteet, ja nostavat koodiin liittyviä eettisiä kysymyksiä esiin. Psykkinen vankila-metaforan kautta voi tarkastella esimerkiksi kysymyksiä siitä, miten ohjelmoidut ympäristöt rajoittavat tai muokkaavat ihmisten käyttäytymistä ja ajattelua: ovatko esimerkiksi pelit tai sosiaalinen media rakennettu tarkoituksella niin koukuttaviksi, että niistä on yksilölle enemmän haittaa kuin hyötyä? Hyvin samankaltainen vallan väline-metafora korostaa ohjelmoinnin ja koodin kautta kerättyyn personoituun dataan liittyviä valta-asetelmia. Kysymykset, kuten kuka omistaa digitaalisten palvelujen käyttäjistä kerätyn datan, ja mihin kerättyä tietoa käytetään, ovat vallan väline-metaforan keskiössä. Sosiaalinen media on mahdollistanut ruohonjuuritasolta nousevia muutoksia yhteiskunnassa esimerkiksi arabikevään muodossa, eli valta-asetelmat voivat kompleksisessa ohjelmoidussa todellisuudessa monensuuntaisia. (Dufva & Dufva, 2016, s. 105).

Mertala & kumppanit (2020) nostavat esimerkkinä emansipatorisesta lähestymistavasta, että kouluissa oppilaiden kanssa voisi havainnollistaa algoritmien mahdollisuuksia sosiaalisen oikeudenmukaisuuden edistämiseksi (Mertala ym., 2020, s. 30). Emansipatorisen ja tulkinnallisen näkökulman voisi nähdä toteutuvan ohjelmoinnin lukutaidon tavoitteissa mm. kriittisen lukutaidon ja sosiaalisesti oikeudenmukaisen toimijuuden edistämiseksi. Sekä psykkinen vankila- että vallan väline-metaforaa voisi kuvataidekasvatuksessa liittää melko luontevasti osaksi mediakasvatusta tuomalla esille koodiin liittyviä valtarakenteita esimerkkitapausten kautta. Avoimen datan käyttäminen esimerkiksi taideteoksen tekemiseen voisi mahdollisesti olla kiinnostava käytännön projekti, joka tekisi näkyväksi dataan liittyviä emansipatorisia kysymyksiä.

Yhdeksäs ja viimeinen paradigma on kuvataidekasvatuksen kannalta yksi kiinnostavimmista. Karnevaali poikkeaa muista metaforista postmodernilla luonteellaan: siinä voivat näkyä kaikki aikaisemmat metaforat yhtäaikaan, tai sitten se voi rikkoa niitä. Leikki ja kokeellinen suhtautuminen ohjelmointiin on metaforan keskiössä. Kuten taide, karnevaali suhtautuu koodiin enemmän kokemuksen kuin teorian kautta. (Dufva & Dufva, 2016, ss. 105–106). Kuvataidekasvatuksessa luova ohjelmointi taiteen tekemisen välineenä on yksi mahdollinen sovellutus, joka lähestyy ohjelmointia nimenomaan karnevaalin metaforan kautta.

Ohjelmointi kuvataidekasvatuksessa

Kuvataidekasvatuksen kontekstissa ohjelmoinnin ottaminen mukaan opetussisältöihin on ollut pitkään enemmänkin harvinainen poikkeus kuin sääntö (Knochel & Patton, 2015). Mistään uudesta ilmiöstä ohjelmoinnin ja taidekasvatuksen yhdistämisessä ei kuitenkaan ole kyse: Knochel & Patton (2015) esittelevät artikkelissaan *If Art Education Then Critical Digital Making: Computational Thinking and Creative Code* erilaisia taiteen, kasvatuksen ja ohjelmoinnin monitieteisiä sovelluksia viimeisen 25 vuoden ajalta. Artikkelin taidemaailman esimerkit sisältävät mm. interaktiivista varjoteatteria (s.23), piirustuksen muuntamisen digitaaliseksi sienirihmastoksi (s.24), kasvien tarpeita ilmaisevan kommunikaattorin (s.27) ja nettisivun, jonka lähdekoodista on löydettävissä yllätyspommi (s.28-30).

Knochel & Patton (2015) esittelevät myös erilaisia taiteelliseen tekemiseen ja taidekasvatukseen soveltuvia ohjelmointikieliä ja tapoja ohjelmoida. Esimerkkeiksi he nostavat aloittelijaystävälliset visuaaliset ohjelmointieditorit, kuten aiemmin mainitsemani Scratchin, joissa ohjelmointi tapahtuu kuin Lego-palikkoja siirtelemällä. Toisena esimerkkinä he antavat Processing-kielen, avoimen lähdekoodin ohjelmointikielen, joka on nimenomaan suunniteltu taiteilijoille taiteen ja visuaalisten objektien tekemiseen. Kolmantena, näistä hieman erilaisena esimerkkinä he esittelevät Arduinon, joka on ohjelmointiin tarkoitettu fyysinen virtapiiri, jonka avulla voi ilman tietokoneen ruutua ohjelmoida esimerkiksi elektronisia laitteita (tai vaikka robotteja) tai tehdä interaktiivista taidetta. Knochel & Patton korostavat, että taideopetuksessa pitäisikin voida nähdä ohjelmointi taiteen tekemisen välineenä siinä missä mikä tahansa perinteinen tekniikkakin, eikä pelästyä ohjelmointiin liittyviä matematiikan tai logiikan konnotaatioita. (Knochel & Patton, 2015)

Jaan Knochel & Pattonin (2015) ajatuksen siitä, että taidekasvattajat voisivat hyödyntää digitaalisia innovaatioita ja ohjelmointikieliä paitsi taiteen tekemisen välineinä, myös kriittisen ajattelun taitojen kehittämisessä (Knochel & Patton, 2015, s. 22).

Luova ohjelmointi

Usein taiteen ja ohjelmoinnin yhdistämisen yhteydessä nousee käsiin esite luova ohjelmointi. Luova ohjelmointi sisältää ohjelmoinnillisen ongelmanratkaisun taitoja (vrt. ohjelmointitaitoja), mutta sille leimallista on leikkimielinen ja kokeellinen asenne ohjelmointiin (Knochel & Patton, 2015, s. 22). Luovaa ohjelmointia voi käyttää taiteen tekemisen välineenä samalla logiikalla kuin mitä tahansa tekniikkaa: ensin on opeteltava säännöt, jotta niitä voi alkaa luovasti rikkoa (Knochel & Patton, 2015,

s. 26). Dufva (2018) määrittelee luovan ohjelmoinnin ohjelmoinniksi, jossa ilmaisu on tärkeämpää kuin funktionallisuus. Luova ohjelmointi laajentaa ohjelmoinnin koodin kirjoittamisesta laajempaan taiteelliseen toimintaan: pelkän ohjelmakoodin kirjoittamisen lisäksi luovaan ohjelmointiin voi kuulua fyysisiä elementtejä kuten laitteita, elektroniikkaa, jotka vuorovaikuttavat fyysisen ja digitaalisen maailman välillä. (Dufva, 2018, ss. 42–43). Luova ohjelmointi on siis tutkimista, kokeilua, ja ilmaisua ohjelmoinnillisia välineitä ja teknologiaa käyttäen, mutta sen ”ydin” on pikemminkin taiteessa kuin ohjelmoinnissa.

Taidekasvatus ja kriittinen ajattelu

Koen tärkeäksi avata lyhyesti omaa taidekasvatusnäkemystäni, sillä käsitykseni taidekasvatuksen perimmäisistä syistä on mitä luultavimmin vaikuttanut ennako-oletuksiini läpi koko tutkimusprosessin: aina tutkimuskysymyksen muodostamisesta haastattelutilanteisiin sekä tapaan, jolla analysoin aineistoa.

Oman taidekasvatusnäkemykseni taustafilosofia ammentaa kriittisestä ja feministisestä pedagogiikasta. Marjo Räsästä (2000) mukailen kriittinen pedagogiikka korostaa todellisuuden monitulkinnallisuutta: ei ole yhtä objektiivista todellisuutta, vaan yksilöiden kokemusten erilaisiksi värittämät elämismaailmat avaavat monenlaisia tulkintoja todellisuudesta. Kokemus ei ole kriittisen pedagogiikan näkökulmasta irrallaan tiedosta, vaan kaikki tieto on sidoksissa yksilön sosiaaliseen taustaan, ja sitä kautta tieto on sidoksissa myös valtaan. Esimerkiksi kysymykset luokasta, sukupuolesta ja kulttuurisesta taustasta vaikuttavat siihen, minkälaisia kokemuksia maailmasta yksilöllä on. Kaikki tieto on siis paikallista, osittaista ja värittyä, tieto ilmenee toiminnassa ja sosiaalisissa suhteissa, eikä siis voi olla universaalista tai kulttuurisesti neutraalia. (Räsänen, 2000, s. 11).

Kriittisen pedagogiikan ja oman kasvatusnäkemykseni mukaan kasvatus ei ole koskaan ideologioista vapaata, sillä kasvatuksella on aina jokin tavoite, oli se sitten yhteiskuntakelpoiseksi kansalaiseksi kasvattamista, vapautumista valtarakenteista (ks. esim. bell hooks 2007) tai esimerkiksi vastuullisen aikuisuuden tavoittelua (ks. esim. Biesta 2017). Jos kasvatuksen tehtävänä on muuttaa tai muovata kasvatettavista jotakin, kasvatukseen liittyy aina vallankäyttöä, tavalla tai toisella. Vastuullinen kasvattaja pyrkii tulemaan tietoiseksi näistä kasvatukseen liittyvistä valtarakenteista ja omista kasvatustyönsä taustalla vaikuttavista arvoistaan, jotta hän pystyisi aktiivisesti purkamaan niitä. Feministisen pedagogiikan kritiikki osoittaa paradoksin: vaikka opettaja pyrkii purkamaan valtarakenteita, hän tulee samalla luoneeksi uusia. Näkemykseni mukaan valtarakenteista ei voi mitenkään täysin luopua, mutta vastuullinen kasvattaja pyrkii aktiivisesti tiedostamaan oman positionsa, ja tekemään

valtarakenteita näkyväksi myös oppilaille.

Valtarakenteiden näkyväksi tekemiseen ja purkamiseen tarvittavia taitoja ovat kriittisen ajattelun taidot. Tässä yhteydessä kriittisyydellä tarkoitan Tomperia (2017) mukaillen taitavaa ja vastuullista harkinta- ja ajattelukykyä, joka kohdistuu sisällöllisesti tai kokemuksellisesti merkityksellisiin ajattelun kohteisiin. Kriittinen ajattelu kytkeytyy siis yksilön persoonan ja maailmankuvan muodostumiseen: se sisältää mm. eettisen arvottamisen, itsereflektion ja -arvioinnin ja toiminnan oikeuttamisen. Kriittinen ajattelu parhaimmillaan kehittyy siihen, että yksilö pystyy tarkastelemaan omia oletuksiaan kriittisesti, sekä tarvittaessa muuttamaan ajatuksiaan ja toimintatapojaan (Tomperi, 2017).

Kriittisen ajattelun taitojen tärkeys korostuu mielestäni erityisesti ajassa, jota on myös ”totuudenjälkeiseksi” ajaksi kutsuttu: 2000-luvulla alkanutta murrosta leimaa tiedon pirstaleisuus, digitaalisuus ja esimerkiksi sosiaalisen median vahvistamat kollektiiviset ajatusvääristymät. Kriittiseen ajatteluun liittyvät arvottamisen taidot ja vastuullisen aikuisuuden taidot ovat mielestäni tärkeitä taitoja kenelle tahansa oppia. Oma taidekasvatusnäkemystäni leimaa siis näkemys siitä, että kriittisen ajattelun taidot ovat tärkeitä opittavia taitoja kuvataidekasvatuksessa. Myös POPSissa mainitaan kriittisen ajattelun taitojen kehittyminen yhtenä kuvataidekasvatuksen oppiaineen tehtävistä (Opetushallitus, 2016, s. 426) – tietenkään kriittinen ajattelu ei ole pelkästään kuvataiteen yksinoikeus, sillä toki myös muissa oppiaineissa harjoitellaan samoja taitoja. Taidekasvatukselliset menetelmät kriittisen ajattelun kehittämiseksi kumpuavat kokemuksellisuudesta, kehollisuudesta, tekemisestä ja kokemisesta.

Varto (2006) on todennut, että erityisesti taidekasvatuksessa tai taiteen avulla opettamisessa kriittisyys voidaan ymmärtää niin, että taidekasvatuksen parissa voidaan käsitellä mitä tahansa asioita, mistä tahansa näkökulmasta, kunhan se edistää ihmisen ajattelua, keskinäistä ymmärtämistä ja arvostamista (Varto, 2006, s. 156). Näen, että taidekasvatus on oiva paikka kriittisten ajattelun taitojen kehittymiselle, oli aihe mikä tahansa. Tässä opinnäytteessä yhtenä taustaoletuksenani tutkielmaa tehdessäni on ollut, että ohjelmoinnillisten sisältöjen ottaminen mukaan taidekasvatukseen voisi mahdollisesti olla hedelmällistä maaperää ohjelmoinnin ja digitaalisten ympäristöjen kriittiselle tarkastelulle (vrt. Dufva, 2018; Knochel & Patton, 2015).

3. Tutkimusmenetelmä

Tieteenfilosofinen perusta

Tutkielmani tieteenfilosofisena lähtökohtana käytän sosiaalista konstruktionismia. Sosiaalisen konstruktionismin ontologia, eli filosofinen oppi siitä, mitä on olemassa, erottaa todellisuuden ja ihmisten käsityksen todellisuudesta toisistaan. Suuntauksen mukaan todellisuus rakentuu sosiaalisten merkitysten kautta, ja tieto on tuotettu aina jostakin näkökulmasta, ihmisten kielellisten käsitysten pohjalta (Löytönen, ei pvm.). Suuntauksen mukaan kieli rakentaa todellisuutta, ja ihmisten käsitykset todellisuudesta rakentuvat yhteisten käsitysten pohjalta (Saaranen-Kauppinen & Puusniekka, 2006).

Näen, että tutkielmani kohteena olevat keskeiset isot käsitteet, kuten taide, kasvatustieteet ja ohjelmointi ovat ihmisen tekemiä konstruktioita, joiden merkitys on sidoksissa mm. aikaan, paikkaan ja kulttuuriin: siten niiden ”todellisen” tai ”muuttumattoman” olemuksen määrittely on kenties mahdotonta. Opinnäytteessäni tarkastelen, mikä niiden merkitys on tässä ajassa ja paikassa, niiden tapausten valossa, joita tutkin tässä opinnäytteessä.

Omassa tutkielmassani tutkin yksittäistapauksia, jotka piirtävät kuvaa siitä, minkälaista taidekasvatuksen ja ohjelmoinnin yhdistäminen voi olla peruskouluissa Suomessa 2020. Tieto ilmiöstä rakentuu yksittäisten haastateltavien käsitysten ja kokemusten pohjalta. Opinnäytteeni tuottama tieto rakentuu sosiaalisissa konstruktioissa, ja siihen vaikuttaa haastattelemieni opettajien käsitykset, miten he sanallistavat omia kokemuksiaan, sekä minun oma tulkintani keskusteluista, kirjalliseen aineistoon nojaten. Tutkijana en ole objektiivinen tarkkailija, vaan aktiivinen osallistuja ja tiedon tuottaja siinä missä haastateltavatkin. Tulkinallani on tiedon muodostumisessa olennainen rooli; joku toinen kirjoittaja saattaisi tehdä toisenlaisia päätelmiä samasta aineistosta. Pysin kuitenkin tekemään ennakkokäsitykseni ja -oletukseni lukijalle näkyviksi parhaani mukaan.

Teemahaastattelu aineistonkeruun menetelmänä

Opinnäytteeni keskeisenä aineistonkeruun menetelmänä käytän teemahaastattelua. Kasvatustieteilijä Sirkka Hirsjärven ja Helena Hurmeen (1979) käyttämä termi *teemahaastattelu* on noussut suomen kielessä eräänlaiseksi yleiskäsitteeksi. Teemahaastattelussa tutkija ei lyö lukkoon tarkkoja haastattelukysymyksiä, vaan määrittelee keskeiset teemat, joita haastattelussa käsitellään (Hyvärinen, 2017, s. 21).

Kuten Eskola & Vastamäki (2015) suosittelevat, valitsin teemahaastattelussa

käyttämäni teemat kirjallisuudesta ja aiheeseen liittyvästä teoriasta, pitäen mielessäni oleellisimman, eli tutkimuskysymyksen. Juuri tutkimuskysymys oikeuttaa erilaisten kysymysten esittämisen (Eskola & Vastamäki, 2015, s. 35). Haastattelun teemoiksi nostin opettajien tekemien kokeilujen sisällöt ja tavoitteet, haastateltavien suhteen ohjelmointiin ja taidekasvatukseen, ohjelmoinnin yhteiskunnassa & tulevaisuudessa, sekä oppilaiden suhtautumisen ohjelmointiin ja taidekasvatukseen. Haastatteluja varten laatimani haastattelupohja on luettavissa opinnäytteen liitteenä. Haastatteluissa käyttämäni teemat poikkeavat huomattavasti lopullisesta aineiston analyysissä käyttämäni teemoista; palaan tämän muutoksen perusteluihin myöhemmin teemoittelua ja aineiston analyysiä koskevassa luvussa.

Teemahaastattelu on usein puolistrukturoitu haastattelu. Puolistrukturoidussa teemahaastattelussa haastatteli on etukäteen määritellyt samat kysymykset kaikille haastateltaville, mutta haastateltavat saavat vastata omin sanoin kysymyksiin (Eskola & Vastamäki, 2015, s. 29). Olin laatinut tekemässäni haastattelussa etukäteen valmiit kysymykset kaikille haastateltaville, mutta olin varautunut muuttamaan keskustelun suuntaa sen mukaan, mitä kukakin haastateltava haastattelutilanteessa halusi tuoda sillä hetkellä esiin ja mihin teemoihin haastatteluissa tuntui mielekkäimmältä keskittyä. Valmiita kysymyksiä tärkeämpiä olivatkin juuri teemat: varmistin, että kaikki teemat tulivat keskustelluiksi haastattelun aikana, vaikka täsmälleen samoja kysymyksiä ei kaikkien haastateltavien kanssa käytykään läpi.

Yhdistelen opinnäytteeni teoriapohjassa kirjallisuutta niin kasvatuksen, taidekasvatuksen kuin ohjelmoinnin saralta. Opinnäytettäni voisi siis kutsua monialaiseksi laadulliseksi tutkimukseksi. Opinnäytteessäni on myös tapaustutkimuksen piirteitä (vrt. Saarela-Kinnunen & Eskola, 2015). Teemahaastattelin neljää opettajaa, jotka ovat kokeilleet työssään ohjelmoinnin ja taidekasvatuksen yhdistämistä, ja jokaista haastattelua voisi pitää omana tapaustutkimuksenaan. Tapaustutkimuksessa on erityisen tärkeää tehdä näkyväksi itse tutkimusprosessi (Saarela-Kinnunen & Eskola, 2015, s. 180), ja pyrin siihen kuvailemalla sekä aineiston keruun että analyysinkin vaiheita niin läpinäkyvästi, kuin vain suinkin pystyn. Aineiston analyysivaiheessa vertailen ja rinnastan haastatteluja toisiinsa, mutta pyrin pitämään lukijalle selkeänä, milloin on kyse mistäkin tapauksesta, jotta ne eivät tekstinä sekoittuisi toisiinsa.

Haastateltavien hankkiminen

Tutkimusprosessin alussa pohdin erilaisia tapoja hankkia haastateltavia. Arvelin etukäteen, että aiheeni ei herätä luultavasti suuria tunnekuohuja opettajakunnassa, joten haastateltavien saaminen voisi olla hieman haastavaa. Minulla ei myöskään ollut aavistustakaan, kuinka monet opettajista ylipäätään olivat ottaneet taideopetuk-

seensa ohjelmoinnillisia sisältöjä.

Pohdin vaihtoehtoa ottaa suoraan yhteyttä jonkun tietyn alueen, esimerkiksi Helsingin kouluihin ja niissä kuvataidetta opettaviin opettajiin. Tällöin olisin voinut saada myös hieman numeraalista dataa siitä, kuinka moni kuvataideopettaja ylipäätään on ottanut ohjelmoinnillisia sisältöjä mukaan kuvataiteen opetukseen. Jätin tämän ajatuksen kuitenkin pois laskuista byrokraattisten kiemuroiden ja koronakevään luomien paineiden vuoksi.

Tutkielmani laadullinen luonne vaati, että saisin haastateltaviksi opettajia, joilla on aiheesta omakohtaista kokemusta. Päädyin siis hakemaan haastateltavia sosiaalisen median kanavien, Facebookin Kuvista-ryhmän ja Kuvataideopettajat ry:n sähköpostilistan kautta. Esimerkiksi Facebookissa hain haastateltavia seuraavalla ilmoituksella:

Viia Ranta
29. tammikuuta

Kuvis ❤️ ohjelmointi - etsin haastateltavia 🌟

Hei opettaja! Oletko joskus kokeillut ohjelmoinnin ja taidekasvatuksen yhdistämistä opetuksessasi? Tai tunnetko jonkun, joka on näin tehnyt? Ilmianna itsesi tai kollegasi!

Teen Aalto-yliopistossa maisterin opinnäytettä taidekasvatuksen ja ohjelmoinnin yhdistävistä opetuskokeiluista peruskouluissa. Kokeilu on ymmärrettävissä laajasti: kokeilut voivat olla yhden opettajan yksittäisiä kokeiluja, muiden opettajien kanssa yhteistyössä tehtyjä MOK:eja tai esimerkiksi laajempia, tavoitteellisempia projekteja, jotka on tuotu peruskoulukontekstiin.

Haastattelen kevään 2020 aikana 1-5 opettajaa heidän toteuttamistaan ohjelmoinnin ja taidekasvatuksen yhdistävistä projekteista. Tarkoitukseni on kartoittaa erilaisia tapoja yhdistää ohjelmointia ja taidekasvatusta opetuksessa ja kerätä tietoa opettajien kokemuksista ohjelmoinnin ja taidekasvatuksen yhdistämisen äärellä.

Laita minulle viestiä yv:nä tai sähköpostia viia.ranta@aalto.fi 10.2.2020 mennessä, jos voit osallistua tai tunnet jonkun, joka saattaisi olla kiinnostunut! Tämä on alustava ja ei-sitova kysely, ja ilmoittamalla minulle kiinnostuksestasi et vielä osallistu tutkimusaineistoni keruuseen. Kerään viralliset tutkimusluvut ennen haastatteluja, haastateltavat saavat halutessaan esiintyä aineistossa anonyymeinä. Haastattelut tapahtuvat helmi- ja maaliskuussa 2020.

Kiitos ja aurinkoista alkuvuotta!

Ystävällisin terveisin,
Viia Ranta

Lisätietoja voi kysyä minulta sähköpostitse:
viia.ranta@aalto.fi

👍❤️ [redacted] ja 16 muuta 2 kommenttia

Tykkää Kommentti

Sain ilmoittautumisia yhteensä viisi. Viiden haastateltavan joukossa oli kaksi vapaan taidekasvatuksen parissa toimiva opettaja, jotka työskentelivät ensisijaisesti kuvataidekouluilla. Kuitenkin olin päättänyt rajata opinnäytteeni koskemaan nimenomaan peruskoulua, sillä peruskoulun piirissä aihetta ei ole tutkittu ollenkaan,

toisin kuin vapaan taidekasvatuksen parissa, joten he rajautuivat ulos tutkielmastani. Haastateltaviksi opettajiksi valitsin nimenomaan peruskoulussa opettavia opettajia, joilla on oman työnsä kautta edes yhden kokeilun verran kokemusta ohjelmoinnin ja taidekasvatuksen yhdistämisestä; tutkielman ulkopuolelle jäävät siis aiheesta potentiaalisesti kiinnostuneet, mutta vielä opetuskokeiluja toteuttamattomat opettajat sekä sellaiset opettajat, joita aihe ei lähtökohtaisesti kiinnosta.

Lopullisissa haastatteluissa haastateltavia oli yhteensä neljä. Yksi heistä on luokanopettaja, kolme yläkoulussa opettavaa kuvataideopettajaa. Yhdellä haastatellulla kuvataideopettajalla on lisäksi käsityön opettajan pätevyys, ja häneen olin suoraan yhteydessä kuultuani vinkin, että hän saattaisi olla potentiaalinen haastateltava. Näin pienellä otoksella en voi lähteä tekemään vertailevaa tutkimusta, enkä voi lähteä tekemään oletuksia esim. siitä, vaikuttaako haastateltavan ikä, sukupuoli, koulun sijaintikunta tai opetusaste siihen, miten hän suhtautuu aiheeseen, joten jätän analyysissäni nämä seikat analysoitavien asioiden ulkopuolelle.

Ilmoituksestani poiketen haastattelut ajoittuivat maaliskuulle 2020. Maaliskuussa Suomeen rantautunut koronaviruspandemia toi hieman lisähaasteita haastattelutilanteiden järjestämiseen, mutta päädyin yhteisymmärryksessä haastateltavien kanssa toteuttamaan kaksi haastattelua kasvotusten ja kaksi videopuhelun välityksellä.

Haastattelujen eettisyydestä

Kysyin ennen haastattelua jokaiselta opettajalta, haluaisivatko he esiintyä aineistossa anonymiminä vai omalla nimellään. Kukaan haastatelluista opettajista ei ilmaissut haluavansa olla ehdottomasti anonymiminä tai omalla nimellään, vaan he luottivat valinnan tekemisen minulle. Haastateltavien anonymiteetti on eettinen kysymys, johon aloin etsimän vastausta Jaana Hallamaan ja kumppaneiden oppaasta *Etiikkaa ihmistieteille* (Hallamaa, Launis, Lötjönen, & Sorvali, 2006).

Huomasin olettaneeni etukäteen, että haastateltavat tietenkin haluaisivat esiintyä anonymimeinä, sillä lukemissani taidekasvatuksen haastattelututkimuksina tehtyinä opinnäytteissä anonymiteetti tuntui olevan enemmän sääntö kuin poikkeus. Vain muutamissa lukemissani opinnäytteissä haastateltavat esiintyivät omilla nimillään (esim. Rybatzki-Tiensivu, 2017). Yleensä silloin, kun haastateltavat esiintyvät opinnäytteiden aineistoissa omilla nimillään, on ollut kyse asiantuntijahaastattelusta, esimerkiksi projektista, jonka parissa haastateltavat ovat yhdessä tutkielman tekijän kanssa työskennelleet. Itsemääräämisoikeuden kunnioittaminen on ensisijaisen tärkeää (Kuula, 2006), mutta nyt kun valinta anonymiteetistä oli annettu minun käsiini, halusin pohtia, mikä olisi juuri tämän tutkielman eettisyyden ja uskottavuuden

kannalta perustelluin vaihtoehto.

Yksityisyyden suojan kannalta anonymiteetti on varmasti aina perustellumpi vaihtoehto. Jos haastateltavia ei voi mitenkään tunnistaa aineistosta, on selvää, että silloin haastateltavalle ei voi tulla mitään seuraamuksia haastattelusta. Omassa aineistossani opettajat hyvin avoimesti kertoivat esimerkiksi työpaikoistaan, ja haastatteluisissa toistuu usein paikkakuntien ja koulujen nimiä. Jos anonymiteetin haluaisi taata kunnolla, kaikki nämä tunnistetiedot pitäisi vaihtaa aineistossa toisenlaisiksi, eikä sekään välttämättä takaisi täydellistä anonymiteettiä, jos haastateltavat ovat esimerkiksi lähipiirilleen kertoneet osallistuvansa tutkimukseen tai heidät voi esimerkiksi työhistorian tarkan kuvauksen takia tunnistaa (Kuula, 2006).

Jos tutkielman luonne vaatii anonymiteetin, se pitää ehdottomasti haastateltaville taata – tutkielman tarkoituksella onkin suuri merkitys siinä, onko anonymiteetille tarvetta vai ei. Minun tutkielmani tarkoituksena on kartoittaa, minkälaisia opetuskokeiluja ja -kokonaisuuksia ohjelmoinnin ja yhdistämisen saralta on tehty, ja pohtia ohjelmoinnin ja taidekasvatuksen yhdistämisen syitä, merkityksiä ja mielekkyyttä. Aihe on suhteellisen uusi diskurssi taidekasvatuksen alalla, sillä oikeastaan vasta uuden opetussuunnitelman myötä se on tullut ajankohtaiseksi Suomessa. Näen, että kokeiluja tehneet opettajat ovat tietyllä tapaa aiheen pioneereja, ja kun tarkoituksenani on nimenomaan esitellä opettajien tekemää työtä, olisi sääli jättää heidät mainitsematta. Opettajien esitleminen heidän omilla nimillään kertoo, että he ovat asiantuntijaroolissa ja seisovat tekemänsä opetustyön takana. Kaikilla opettajilla oli myös haastattelun perusteella kiinnostusta jatkaa ohjelmoinnin ja taidekasvatuksen yhdistämisen saralla, ainakin jollain tavalla: voisi siis katsoa, että opettajat ovat kiinnostuneita seuraamaan aiheesta käytyä keskustelua ja mahdollisesti halukkaita osallistumaan siihen. Haastatellut opettajat esiintyvät haastattelussa myös ensisijaisesti opettajina, työroolinsa kautta, eivätkä haastattelukysymykset koskeneet heidän henkilökohtaista elämäänsä tai persoonaansa kuin niiltä osin, mitä he toivat haastattelussa esiin. Minulle oli tärkeää, että jos päätyisin esittämään heidät omilla nimillään, tutkijan roolissa minun on pidettävä huoli, etten analyysivaiheessa saattaisi haastateltavia epäedulliseen valoon, tai että opinnäytteestä ei voisi tunnistaa esimerkiksi heidän oppilaitaan.

Jotta en olisi tehnyt päätöstä pelkästään itse, ja varmasti kunnioittaisin haastateltavien yksityisyydensuojaa, halusin vielä varmistaa jälkikäteen haastatelluilta, että he vielä haastattelujen jälkeenkin olisivat valmiita esiintymään omilla nimillään. Päätin lähettää heille sähköpostia ja kysyä vielä kertaalleen heidän suostumuksensa, jotta en vahingossakaan tekisi päätöstä ilman heidän lupiaan. Kaikki haastateltavat ilmaisivat suostumuksensa vielä toisellakin kyselykerralla siihen, että he voisivat esiintyä opinnäytteessäni omalla nimillään. Päädyin siis esittämään haastateltavat

opinnäytteessäni ilman anonymisointia, sillä en nähnyt sille tutkimuseettistä estettä. Päinvastoin, haastattelemani opettajat ovat silmissäni asiantuntijoita, ja haluan, että he saavat ansaitsemansa tunnustuksen tekemästään opetustyöstä.

Lukijan olisi hyvä tiedostaa, että pyrkimykseni kohdella haastateltavia analyysissäni ensisijaisesti reilusti ja haastateltavia kunnioittaen voi rajata joitakin opetuskokeiluja tai haastatteluja kritisoivia näkökulmia pois analyysistä. Toki pyrin tarkastelemaan haastateltujen tekemiä opetuskokeiluja monelta kantilta, myös mahdollisesti kriittisempiä tai pohtivampia näkökulmia esittäen, mutta eettisyyden varmistamiseksi lähtökohtani analyysilleni on olla kunnioittava haastateltuja kohtaan. Tämän tyyppinen valinta tekee hyvin näkyväksi tieteenfilosofiset lähtökohtani ja sen, että näen, ettei tutkimus ole koskaan arvovapaata – omat arvolatautuneet ajatukseni eettisyydestä, ja valintani esittää haastateltavat omilla nimillään saattavat ohjailla tapaa, jolla myöhemmin analysoin aineistoa.

Haastateltavat

Ensimmäinen haastateltavani oli luokanopettaja Pinja Herttovuo. Haastattelun aikana hän opetti ensimmäistä virallista opetusvuottaan, hänellä oli oma ekaluokka ja lisäksi hän opetti 4.- ja 5.-luokkalaisille valinnaista ohjelmointia. Pinja on minulle entuudestaan tuttu, mutta en tiennyt hänen tekemistään opetustöistä juuri mitään etukäteen – ainoastaan sen, että hänellä oli erityinen kiinnostus kuvataiteen opettamiseen, sillä hänellä on lyhyt sivuaine kuvataiteesta ja useamman vuoden opetuskokemus vapaa-ajan kuvataidetoiminnan opettamisesta. Opinnäytteeni koskee peruskoulua, mutta olin selvästi itse ajatellut peruskoulun tarkoittavan nimenomaan yläkoulua: ajattelin, että on kiinnostavaa ottaa mukaan luokanopettajan näkökulma, jotta saisin hieman näkökulmaa nuorempien oppilaiden opettamisesta. Pinja kertoi haastatteluun ilmoittautumisen yhteydessä, että hän oli tehnyt pelimaailmaa käsittelevän kuvataidetta ja ohjelmoinnillisia sisältöjä yhdistelevän opetuskokeilun viidesluokkalaisten kanssa Microsoft Powerpointilla. Pinja kuvaili haastattelun alussa, että hänen suhtautumisensa digitaalisuuteen on aina ollut varsin mutkatonta – hänelle on luontevaa ottaa haltuun uusia laitteita, sisältöjä ja teknologioita ja hyödyntää niitä opetuksessa. Hän ei ollut käynyt omissa opinnoissaan ohjelmoinnin kursseja, mutta hän oli työelämässä itse opetellut ohjelmoinnin perusteita mm. Linda Liukkaan Ruby-kirjoista, jotta voisi ottaa ohjelmoinnillisia sisältöjä mukaan opetukseensa.

Toinen haastateltava, kuvataideopettaja Joanna Puro-Aunola ilmoittautui minulle sähköpostitse. Ilmoittautumisen yhteydessä hän kertoi kokeilleensa Scratchia kuvataideopetuksen osana. Haastattelussa tuli ilmi, että Joanna on kokenut kuvataideopettaja, ja hän on tehnyt pitkän työuran kuvataideopettajana peruskoulussa. Hän

kertoi haastattelussa, että hän haluaa olla digitaalisuuden suhteen etukenossa, ja hän kertoi käyttävänsä opetuksessaan hyväksi monipuolisesti erilaisia digitaalisia välineitä ja -alustoja. Ohjelmointi alkoi kiinnostaa häntä, kun aihe nousi ajankohtaiseksi mediassa, mutta hän koki sen pitkään hyvin vieraaksi itselleen: hän kertoi käyneensä opettajille suunnatuissa Scratch-koulutuksissa useamman vuoden ilman, että otti ohjelmoinnillisia sisältöjä mukaan opetukseensa. Vasta omien kokeilujensa kautta hän löysi varmuutta ottaa Scratch mukaan kuvataiteen opetukseen. Haastattelun aikaan hän oli teettänyt useampana vuotena peräkkäin sekä 7.- että 8.-luokkalaisten kanssa taidehistoriaa ja animaatiota yhdistäviä tehtäviä Scratchilla.

Kolmanneksi haastattelin kuvataiteen ja käsityön opettaja Iiro Tujulaa videopuhelun välityksellä. Otin itse suoraan yhteyttä Iiroon kuultuani vinkin, että hänellä saattaisi olla näkemyksiä ohjelmoinnin ja kuvataiteen yhdistämisestä. Minua kiinnosti ajatus, että yksi haastateltavista olisi suorittanut kuvataideopettajan koulutuksen lisäksi käsityön opettajan koulutuksen – toisin kuin kuvataiteen, käsityön oppiainesisältöihin kuuluu ohjelmointi, ja ennakko-oletukseni oli, että hänen näkökulmansa saattaisi olla sen vuoksi erilainen muihin haastateltaviin verrattuna. Iirolla oli haastattelun aikana kokemusta koulumaailmasta viitisen vuotta, ja hän oli opettanut sekä alakoulu- että yläkouluikäisille oppilaille ohjelmointia osana kuvataidetta. Iiro olikin haastateltavista ainoa, joka oli tehnyt oppilaiden kanssa useita erilaisia ohjelmointia ja kuvataidetta yhdisteleviä opetuskokeiluja. Iiro kertoi haastattelussa, että hänen suhtautumisensa digitaalisiin teknologioihin on sujuva: hänelle on ollut aina luontevaa lähteä kokeilemaan ja käyttämään uusia digitaalisia alustoja ja -ympäristöjä. Hän kuvaili lähestymistapaansa ohjelmointiin kokeilevaksi, leikkisäksi, tarvelähtöiseksi ja jatkuvasti uutta oppivaksi. Varsinaisia ohjelmoinnin opintoja hänellä ei ollut taustalla, vaan hän oli opetellut ohjelmointia perehtymällä pääsääntöisesti itsenäisesti aina kulloinkin tarvitsemiinsa ohjelmointiteknologioihin, -kieliin tai -ympäristöihin.

Viimeisenä haastattelin videoyhteyden välityksellä kuvataideopettaja Pietari Kellokumpua, joka opetti yläkoulussa kuvataidetta toista vuotta. Hänet tunsin entuudestaan opinnoista Aalto-yliopistosta, mutta minulle tuli uutena tietona, että hän oli ottanut ohjelmoinnillisia sisältöjä osaksi kuvataideopetusta. Kuvataidetta ja ohjelmointia yhdistelevänä opetuskokeiluna hän oli tehnyt useamman seitsemäsluokan kanssa kokeilun OpenProcessing-ympäristössä. Kokeilu yhdisteli luovalla tavalla abstraktin taiteen ohjelmointiin, ja ohjelmointia kirjoitettiin koodikielellä. Pietari kuvaili suhdettaan digitaalisiin teknologioihin aika neutraaliksi: hän ei omien sanojensa mukaan ole ollut mitenkään erityisen kiinnostunut aiheesta, mutta häntä ei toisaalta myöskään ollut koskaan pelottanut tarttua laitteisiin. Opetuskontekstissa hänelle laitteet ja digitaaliset ympäristöt olivat tuntuneet luontevilta käyttää, sillä

hän koki, että peruskoulussa lähdetään yleensä niin perusteista, ettei hänen oman ammattitaitonsa raja ollut tullut vastaan. Ohjelmointi oli Pietarille tuore aluevaltaus haastattelun aikana: hän oli ennen opetuskokeiluaan käynyt verkkokurssin, jossa tutustuttiin OpenProcessing-ympäristöön, mutta muuta kokemusta ohjelmoinnista hänellä ei ollut.

Haastattelutilanteet

Haastattelijana olin ennen tätä opinnäytettä täysin kokematon, ja näin jälkeenpäin olen pohtinut, että se on saattanut hyvinkin vaikuttaa aineistoni laatuun. Tiedostin epävarmuuteni ja kokemattomuuteni haastattelijana jo ennen haastatteluja, ja pyrin välttämään pahimmat sudenkuopat perehtymällä kirjallisuuteen etukäteen ja nostamaan haastattelun teemat kirjallisuuden pohjalta. En kuitenkaan osannut varautua siihen, että itse haastattelutilanteissa epävarmuuteni ilmeni paikoin sekavuutena: takeltelin paljon sanoissa, ja saatoin kysyä suunnittelemani kysymykset epämääräisesti, niin että haastateltavat saattoivat ymmärtää kysymykset keskenään eri tavalla, tai johdatella haastateltavia niin, että saamani vastaukset heijastelevat paitsi haastateltavien ajatuksia, myös omia ennakkokäsityksiäni. Lisäksi minun oli paikoin vaikeaa pysyä neutraalin haastattelijan roolissa, sillä olin todella innoissani haastateltavien toteuttamista opetusprojekteista. Vasta viimeisen haastattelun kohdalla minusta tuntui, että minulla oli edes jonkinlainen ymmärrys siitä, miten haastattelutilanteessa tulisi haastattelijan roolissa toimia, ja ehkä juuri siksi koen, että viimeisenä tekemäni haastattelu vastasi selkeimmin niihin kysymyksiin, joita oikeasti halusin tutkia. Jälkeenpäin ajateltuna olisi ollut kannattavaa toteuttaa muutama harjoitushaastattelu ennen oikeita haastatteluja. Aineistoni on kuitenkin sellaisenaan riittävä analyysin tekemiseen, enkä nähnyt tarvetta lähteä tekemään uusintahaastatteluja.

Haastattelujen aikana pohdin, olinkohan ottanut haastatteluun ehkä muutaman teeman liikaa. Haastattelut venyivät yhtä lukuun ottamatta pidemmiksi kuin mitä olin suunnitellut: olin ajatellut, että haastattelut kestäisivät n. 45 minuuttia, mutta kaikki paitsi yksi haastatteluista kestivät 55min-1,5h. Kuitenkin halusin pitää haastattelujen sisällöt keskenään samoina, joten pitäydyin etukäteen suunnittelemissa teemoissa kaikissa haastatteluissa, vaikka se tuntuikin joidenkin haastattelujen kohdalla hieman epäoleelliselta. Haastattelujen pohjalta litteroitua tekstiä kertyi yhteensä 75 sivua. Litteroinnissa noudatin peruslitterointia, johon kirjoitin ylös ainoastaan merkityssisällön kannalta oleellimmat äänenpainot, kuten naurun, sarkasmin ja tauot, mutta jätin pois ylimääräiset täytesanat (kuten ”niinku”) ja vähensin haastattelijan haastateltavia myötäileviä kommentteja (”hmm, niin, joo”). En nähnyt syytä tarkempaan litterointiin tutkimuskysymykseni ja aineiston analyysin

kannalta – tarkoitukseni on ensisijaisesti analysoida opettajien tekemiä opetuskokeiluja ja heidän kokemuksiaan ohjelmoinnin ja taidekasvatuksen yhdistämisestä, eikä esimerkiksi tutkia vuorovaikutusta haastattelijan ja haastateltavan välillä.

Annoin haastatelluille opettajille mahdollisuuden näyttää haastattelussa minulle materiaalia, kuten kuvia tai opetusmateriaaleja liittyen heidän opetuskokeiluihinsa, mikäli he kokivat, että opetuskokeilujen sisältö aukeaa paremmin niiden kautta. Kerroin haastateltaville, että kuvamateriaalia käytetään vain opetuskokeilujen selittämiseen paremmin auki, enkä aio tarttua niihin analyysivaiheessa kuin haastatteluja selventävänä aineistona, tai liittää niitä osaksi opinnäytettäni. Kaikilla haastattelemillani opettajilla oli jonkinlaista kuvallista aineistoa oppilaiden töistä haastattelun aikana näytettäväksi, mikä ilahdutti minua suuresti: kuten olin arvaillutkin, kuvallinen aineisto konkretisoi minulle valtavasti opetuskokonaisuuksien ja -kokeilujen sisältöjä, ja sai minut ymmärtämään niitä paremmin kuin pelkkä suullinen kuvailu. Kuvalla on valtava kerronnallinen voima kirjoitettuun tai puhuttuun sanaan verrattuna, ja toivon, että pystyn tekstissäni edes osittain avaamaan lukijalle, mistä opetusprojekteissa oli kyse.

Teemoittelu

Teemahaastattelun analyysinmuotona käytän teemoittelu. Hirsjärvi & Hurmeen (2008) mukaan teemoittelulla tarkoitetaan teemahaastattelun purkamista tarkastellen niitä piirteitä, jotka ovat toistuvat useissa haastatteluissa. Teemat saattavat pohjautua teemahaastattelun teemoihin, ja on oikeastaan odotettavissa, että ainakin lähtökohdatteemat nousevat esiin. Sen lisäksi tulee tavallisesti esille muita teemoja, jotka voivat olla lähtöteemoja mielenkiintoisempia (Hirsjärvi & Hurme, 2008). Teemoittelussa tutkin haastateltujen käsityksiä aiheesta, minkälaisia yhteneväisyyksiä ja toisaalta eroavaisuuksia esitellyissä haastateltavien opetuskokonaisuuksilla on toisiinsa nähden. Jäsentelen myös haastateltavien käsityksiä ohjelmoinnin ja taidekasvatuksen opettamisesta ja niiden merkityksestä koulussa ja yhteiskunnassa. Lisäksi pyrin tuomaan ilmi käytännössä toteutuneiden kokeilujen yhtäläisyyksiä ohjelmoinnin ja taidekasvatuksen teoriaan. Taustaoletukseni on, että taidekasvatuksen yhteydessä tapahtuva ohjelmoinnin opetus voisi olla luonteeltaan erilaista kuin funktionaaliseen ohjelmointikäsitteeseen pohjautuva ohjelmoinnin opetus (vrt. metaforat Dufva & Dufva, 2016). Pyrin aineistoa analysoidessani tiedostamaan tämän oletukseni, ja suhtautumaan aineistoon avoimella otteella, haastateltujen äänen mahdollisimman hyvin säilyttäen ja huomioon ottaen.

Lähdin toteuttamaan teemoittelu käymällä haastattelumateriaalit yksitellen läpi. Aloitin tutkimalla, mitä haastateltavat ovat vastanneet haastattelussa esittelemiini

teemoihin, ja ryhmittelin haastateltavien vastaukset niin, että pystyin vertailemaan, mitä he ovat vastanneet kunkin teema-alueen kohdalla. Haastatteluja analysoidessani huomasin, että suurin osa valmiiksi asettamistani teemoista ei selvästikään ollut kovin hedelmällisiä tutkimuskysymykseni kannalta: olin ajatellut, että esimerkiksi uusi OPS olisi ollut merkittävä syy ottaa ohjelmointia mukaan taidekasvatukseen, ja olin nostanut siksi OPSin omaksi teemakseen, mutta kukaan haastattelemistani opettajista ei ollut sitä mieltä, että OPS:illa olisi ollut erityisen suurta merkitystä heidän opetusprojekteilleen. Kaikki haastateltavat kertoivat kuin yhteen ääneen, että heidän mielestään on hyvä, että ohjelmointi on OPSissa läpäisyaineena, jolloin sitä voi halutessaan ottaa kuvataideopetukseen mukaan, mutta siihen ei ole varsinaisesti pakkoa. Tutkimuskysymykseni kannalta OPSiin liittyvät keskustelut eivät kuitenkaan nostaneet mitään erityisen kiinnostavaa esiin. Jouduin siis luopumaan OPS:ista ja samoin useimmista muistakin teemoista, joita olin etukäteen ajatellut, ja katsomaan aineistoa ennakkoluulottomampien linssien läpi.

Aineistoa läpikäydessäni huomasin, että haastatteluissa haastateltavat saattoivat puhua ennakkoon valitsemistani teemoista täysin eri näkökulmista, joten ne eivät lopulta oikein soveltuneet yhdistäväksi teemoitteluksi. Lopulta valitsemassani teemoittelussa korostuu kolme varsin laajaa eri ulottuvuutta ohjelmoinnin ja kuvataiteen yhdistämisestä: ymmärtämisen, tuottamisen ja kokemuksellisuuden ulottuvuudet. Nämä kolme teemaa toistuivat kaikkissa haastatteluissa, ja pystyn niiden kautta avaamaan kaikkien haastateltavien ajatuksia aiheesta, joten katson, että teemoitteluni on varsin perusteltu.

Haastatteluja analysoidessani pyrin pitämään kaikkien haastateltavien äänet tasapuolisesti esillä, mutta on myönnettävä, että analyysiosiossa erityisesti Iiron ääni saa monesti paljon tilaa. Muut kolme haastateltavaa olivat tehneet yhden opetusprojektin (joskin esimerkiksi Joanna oli vetänyt saman tehtävän useampana vuotena), kun taas Iiro oli tehnyt useita erilaisia kokeiluja useiden vuosien aikana. Iiron voi siis sanoa olleen haastateltavista kaikkein kokenein ohjelmoinnin ja taidekasvatuksen yhdistämisen saralla, ja pitkän kokemuksensa vuoksi hänellä oli hyvin ymmärrettävästi paljon sanottavaa aiheeseen. Koin monesti analyysin kannalta relevantiksi nostaa hänen ajatuksiaan esiin, sillä hän toisinaan toi keskusteluun sellaisia näkökulmia, joita muiden haastatteluissa ei noussut esiin. Haluan kuitenkin korostaa, että pidän kaikkia haastattelemiani opettajia ohjelmoinnin ja taidekasvatuksen yhdistämisen erityisasiantuntijoina, ja toivon, että lämpö ja kunnioitus, jolla heihin kaikkiin suhtaudun, välittyy myös tekstistäni.

4. Haastateltavien opetusprojektit

Tässä luvussa vastaan ensimmäiseen tutkimuskysymykseeni: minkälaisia ohjelmoinnin ja taidekasvatuksen yhdistäviä oppimiskokonaisuuksia on kokeiltu peruskouluissa kuvataiteen oppiaineessa uuden perusopetuksen opetussuunnitelman (myöhemmin POPS 2014) ilmestymisen jälkeen?

Haastattelemieni opettajien tekemillä opetusprojekteilla oli monta nimeä. Haastatteleman opettajat käyttivät keskenään erilaisia sanoja kuvailemaan opetustaan: Joanna puhui tehtävistä, Pinja tehtävästä ja opetuskokonaisuudesta, Iiro opetuskokeiluista. Pietari empi opetuskokeilun ja opetuskokonaisuuden välillä; päädyimme haastattelussa käyttämään opetuskokeilua. Pyrin käyttämään yksittäisiä opetusprojekteja kuvaillessani niitä sanoja, joilla haastateltavat opetuksiaan nimittivät säilyttääkseni niiden alkuperäisen luonteen mahdollisimman hyvin. Silloin kun puhun useammasta kuin yhdestä opetuskokonaisuudesta tai -kokeilusta, puhun opetusprojekteista. En pyytänyt haastateltavia nimeämään opetusprojektejaan, joten olen ottanut tutkijan vapauksia nimeämisen suhteen.

Esittelen seuraavassa taulukossa opettajien tekemät opetusprojektit lyhyesti ja pintapuoleisesti, käyttäen omaa tulkintaani erityisesti opetussisältöjen ja -tavoitteiden käsitteellistämisen osalta. Osa esimerkiksi opetusprojektien yhteydessä luettelemistani tavoitteista ja -sisältöalueista on sellaisia, joita haastatteleman opettajat itse sanoittivat, osa on haastattelujen, näkemieni kuvien ja kuvausten perusteella tekemääni tulkintaa. Esittelen opetusprojektit lyhyesti, jotta lukijan olisi helpompi hieman hahmottaa, minkälaisista opetusprojekteista seuraavassa luvussa tarkemmin teemoittelmani opettajien kokemukset kumpuavat; en usko, että kaikkien yksittäisten tapausten yksityiskohtaiselle esittelylle on opinnäytteeni tutkimuskysymyksen kannalta tarvetta. Aineiston analyysin kannalta oleelliset opetusprojektit esittelen tarkemmin luvussa 5 teemoittelun yhteydessä.

Sekä Pinja, Joanna että Pietari olivat tehneet yhden opetusprojektin (Joanna ja Pietari olivat vetäneet saman projektin useammille ryhmille, Joanna useampana vuotena tehtävää hieman varioiden) ja haastatteluissa pystyimme käymään läpi varsin yksityiskohtaisesti, miten opetuskokeilut olivat menneet. Sen sijaan Iiro oli tehnyt lukuisia kuvataidetta ja ohjelmointia yhdisteleviä kokeiluja useamman vuoden aikana, eikä hänen haastattelunsa kohdalla yksittäisen kokeilun merkitys noussut yhtä suureksi, kuin haastatteluissa, jotka pyörivät yksittäisten opetusprojektien ympärillä. Tämän vuoksi joidenkin Iiron tekemien kokeilujen kohdalla en tullut kysyneeksi tarkkoja tietoja esimerkiksi tekniikasta, joilla ohjelmointia oli toteutettu, ja osa taulukossa olevista kohdista saattaa jäädä tyhjiksi.

| Lyhyt kuvaus / tehtävänanto | Kuviksen sisältöjä | Ohjelmointisisältöjä | Teemoja |
|--|---|--|---|
| Pelimaailma-konsepti (Pinja) | | | |
| 5.-luokkalaisten oppilaat tutustuivat pelimaailmisiin. He suunnittelivat oman pelimaailman, johon he piirsivät hahmot ja ympäristön. Hahmot animoitiin ja liitettiin osaksi ympäristöä Powerpointilla. Ohjelmointi oli opetuskokonaisuudessa osana keskustelujen kautta. | <ul style="list-style-type: none"> - pelimaailma - elokuvakasvatus - tarinallisuus - maalaaminen-piirtäminen - kuvankäsittely | <ul style="list-style-type: none"> - visuaalinen ohjelmointi - ohjelmoinnin "näkeminen" keskustelun kautta (ohjelmoinnin lukutaito) | <ul style="list-style-type: none"> - ohjelmointi taiteen tekemisen välineenä - ohjelmoinnin lukutaito |
| Suomen kultakausi Scratchillä (Joanna) | | | |
| 7.-8.-luokkalaisten. Tehtävä yhdisti Suomen taidehistoriaa, tarinallisuutta ja ohjelmointia Scratchillä. Seiskaluokkalaisten versiossa valittiin kaksi kultakauden hahmoa, jotka kuvankäsiteltiin toisenlaisiksi, ja hahmojen välille lisättiin Scratchissä dialogia. Lopputuloksena oli animoitu videonpätkä, jossa tehtävän tavoitteiden mukaan tuli olla liikettä, ääntä ja dialogia. | <ul style="list-style-type: none"> - taidehistoria (Suomen kultakausi) - video - tarinallisuus - kuvankäsittely - piirtäminen | <ul style="list-style-type: none"> - visuaalinen ohjelmointi - Scratch | <ul style="list-style-type: none"> - ohjelmointi taiteen tekemisen välineenä - yhteistyö |
| Abstraktia taidetta OpenProcessingilla (Pietari) | | | |
| 7.-luokkalaisten tuli luoda liikkuva abstrakti teos OpenProcessingilla. Ensin oppilaat tutustutettiin 1900-luvun alun abstraktiin maalaustaiteeseen, ja sitten he tekivät koodausharjoituksia teosten pohjalta: esimerkiksi Malevitšin Musta neliö toteutettiin saman tien koodaten. Yksinkertaisten koodausharjoitusten jälkeen oppilaat toteuttivat omat abstraktit teoksensa OpenProcessingilla. | <ul style="list-style-type: none"> - taidehistoria (modernismi ja abstrakti taide) - digitaalinen kuva: RGB-värijärjestelmä - geometrisuus / orgaanisuus | <ul style="list-style-type: none"> - luova ohjelmointi - Processing - ohjelmoinnin perusteita: syntaksi, muuttuja-käsite, koordinaatisto (miten kuva piirtyy tietokoneelle) - ohjelmoinnillinen ajattelu | <ul style="list-style-type: none"> - ohjelmointi taiteen välineenä - luova ohjelmointi - digitaalisuuden luonne - tekemällä oppiminen |
| Piirtorobotti (Iiro) | | | |
| 5.-6.-luokkalaisten rakensivat robotin, jonka pystyi ohjelmoida liikkumaan. Robottiin kiinnitettiin kynä, ja se annettiin nuoremmille oppilaille ohjelmoitavaksi: 1.-4.-luokkalaisten kirjoittivat "ab"-ohjelmakoodia ja koneenkäyttäjänä toiminut vanhempi oppilas paineli nappuloita liikuttaen robottia ohjeiden mukaan. Paperille muodostui lopulta piirros. Lopuksi oppilaiden kanssa pohdittiin, kuka on teoksen taiteilija. | <ul style="list-style-type: none"> - nykytaide - yhteisötaide - kuka on taiteilija? | <ul style="list-style-type: none"> - rakentaminen - ohjelmointi - ohjelmoinnillinen ajattelu | <ul style="list-style-type: none"> - yhteistyö - yhteisöllisyys - tekemällä oppiminen |

| Lyhyt kuvaus / tehtävänanto | Kuviksen sisältöjä | Ohjelmointisisältöjä | Teemoja |
|--|---|---|---|
| Arcade-pelikone (liro) | | | |
| 6.-luokkalaiset rakensivat yhdessä Arcade-pelikoneen. Aluksi oppilaat tekivät Scratchillä pelejä, tutustuttiin pikseligrafiikkaan. Valmiin pelin ympärille rakennettiin pahvista yms. pelikone. Lopuksi koneeseen liitettiin ohjaimet, niin että peliä pystyi oikeasti pelaamaan. | <ul style="list-style-type: none"> - pelimaailma - muotoilu - pikseligrafiikka | <ul style="list-style-type: none"> - rakentaminen - visuaalinen ohjelmointi - pelimodaaminen | <ul style="list-style-type: none"> - kokemuksellinen oppiminen - tekemällä oppiminen - yhteistyö |
| Karkkiautomaatti (liro) | | | |
| Alakoulun oppilaiden kanssa rakennettiin laite, joka sylki karkkeja ulos kolikkoja vastaan. Oppilaat muotoilivat ympärille hammaspeikon näköisen kotelon, joten lopputuloksena oli varsin hauska karkkiautomaatti. | <ul style="list-style-type: none"> - muotoilu | <ul style="list-style-type: none"> - rakentaminen - ohjelmointi Arduinolla (?) | <ul style="list-style-type: none"> - kokemuksellinen oppiminen - yhteistyö - tekemällä oppiminen |
| Oma peli Scratchillä (liro) | | | |
| Alakoulun oppilaat toteuttivat omat pelit Scratchillä ohjelmoiden. Aluksi tutustuttiin pelien historiaan, ja sen jälkeen lähdettiin kokeilemaan, miten pelejä voisi tehdä. Ohjelmoiminen aloitettiin etsimällä netistä valmis Scratch-peli, jota pystyi sitten itse modaamaan. | <ul style="list-style-type: none"> - pelimaailma - pelien historia | <ul style="list-style-type: none"> - pelimodaaminen - visuaalinen ohjelmointi - Scratch | <ul style="list-style-type: none"> - ohjelmointisällöt edellä - yhteistyö - avoin lähdekoodi - tekemällä oppiminen |
| The Hand of Elvis (liro) | | | |
| 8.-luokkalaisille. Ensin oppilaita perehdytetään Anssi Kasitonin taiteeseen, ja rakennetaan The Hand of Elvis-teosta muistuttava käsi, josta nappia painamalla saa keskisormen pystyyn. Tehtävä oli siis samalla ohjelmointihaaste oppilaille. Haastattelun hetkellä projektia ei oltu vielä toteutettu, liro oli rakentanut itse prototyypin. | <ul style="list-style-type: none"> - nykyaide | <ul style="list-style-type: none"> - rakentaminen - "ohjelmointihaaste" - ohjelmoinnillinen ajattelu - ohjelmointi Arduinolla (?) | <ul style="list-style-type: none"> - ohjelmoinnillinen ajattelu - tekemällä oppiminen - yhteistyö - ohjelmointi taiteen välineenä |
| Valotaidetyöpaja kirjastossa (liro) | | | |
| Kirjastossa työpaja ala-asteikäisille oppilaille. Oppilaat ohjelmoivat salaviestejä kirjaston valokylttiin. | <ul style="list-style-type: none"> - yhteisötaide - nykyaide | <ul style="list-style-type: none"> - ohjelmoinnillinen ajattelu - ohjelmointi | <ul style="list-style-type: none"> - yhteistyö - kokemuksellisuus |

5. Teemat

Toteutuneiden opetusprojektien esittelemisen lisäksi toisena tutkimuskiinnostukseni on tutkia opettajien kokemuksia ohjelmoinnin ja taidekasvatuksen yhdistämisen äärellä. Minua kiinnostaa, miten opettajat ovat kokeneet ohjelmoinnin ja taidekasvatuksen yhdistämisen koulukontekstissa, miten he perustelevat ohjelmoinnilliset sisällöt opetuksessaan ja minkälaista oppimista heidän käsitystensä mukaan ohjelmoinnin ja taidekasvatuksen yhdistäminen voi tuottaa. Tässä luvussa vastaa näihin kysymyksiini erittelemällä haastattelemieni opettajien kokemuksia ymmärtämisen, tuottamisen ja kokemuksellisuuden teemojen ympärillä. Teemat ovat osittain päällekkäisiä, limittäisiä ja rihmastomaisesti kytköksissä toisiinsa: tuntuu keinotekoiselta ja ehkä jopa hieman väkivaltaiselta yrittää eritellä kokonaisuuksia erillisiksi osakseen, mutta yritän parhaani mukaan tuoda tekstissä esiin siltoja teema-alueiden välille.

Valitsin nämä teemat, sillä ne tuntuivat korostuvan aineistossa kaikkien opettajien puheissa, tavalla tai toisella. Pyrin teemoja valitessani olemaan aineistolle avoin, ja aidosti löytämään sellaiset teemat, jotka koskisivat kaikkia haastatteluja jollakin tavalla – tämä johti siihen, että valitsevani teemat poikkesivat huomattavasti alkuperäisistä teemoista, joita olin ajatellut etukäteen. Ainoastaan ymmärtäminen oli teema, jonka olin osannut kuvitella etukäteen, ja sekin kulki vielä haastatteluvaiheessa nimellä ohjelmoinnin lukutaito. Muut haastattelussa käyttämäni teemat, joita olin etukäteen ajatellut merkityksellisiksi kuten oppiminen ja taiteen ja ohjelmoinnin suhde tuntuivat kuitenkin soljahtavan paremmin osaksi joko ymmärtämisen, tuottamisen tai kokemuksellisuuden teemoja.

Vasta jälkeenpäin tätä teemoittelua jo hahmoteltuani huomasin, että pitkän linjan taidekasvatuksen tutkija Marjo Räsänen (2000) on jäsennellyt kokemuksellisen taideoppimisen perustuvan kokemuksellisuuden, käsitteellistämisen (ymmärtämisen) ja tuottamisen ympärille (Räsänen, 2000). En tiedä, vaikuttiko oma taustani taidekasvatuksen opiskelijana juuri tämän teemoittelun muodostumiseen, mutta on kieltämättä kiinnostavaa, miten ohjelmoinnin ja taidekasvatuksen yhdistäminen korreloi kokemuksellisen taideoppimisen teorian kanssa. Ainoa teema-alue, joka omasta analyysistäni puuttuu Räsänen (2000) jaottelun mukaan, on reflektio. Omassa analyysissäni reflektio-osuutta on oikeastaan koko aineiston keruu eli haastattelu: se mitä haastatteleman opettajat puhuvat, miten he jäsentelevät kokemuksiaan, on heidän tekemäänsä reflektiota ymmärtämisestä, tuottamisesta ja kokemuksellisuudesta. Tässäkin teemojen limittäisyys korostuu.

Ymmärtäminen

Ymmärtämisen teemassa nostan esiin haastateltavien ajatuksia taidekasvatuksen ja ohjelmoinnin yhdistämisen tiedollisesta puolesta, maailman hahmottamisesta ja ajattelun taidoista. Nostan esiin myös muutamia haastateltavien kuvailemia syitä ottaa ohjelmointia mukaan taidekasvatukseen. Aloitan esittelemällä, miten haastateltavat ymmärtävät ohjelmoinnin, ja siirryn sitten kuvailemaan, minkälaisia ajattelun taitoja haastateltavat näkevät ohjelmoinnillisten sisältöjen tuovan kuvataidekasvatukseen (ja toisinpäin).

En alustanut omaa suhtautumistani ohjelmointiin haastatelluille, mutta haastateltavat suhtautuivat ohjelmointiin pääsääntöisesti hyvinkin laajana ilmiönä, kuten minäkin: haastatteluissa puhuimme koodaamisesta sekä koodikielen kirjoittamisena että visuaalisena ohjelmointina, ja myös ohjelmoituista ympäristöistä laajasti ja ohjelmoinnin lukutaidosta ajattelun taitona. Usein keskustelu ohjelmoinnista laajeni vielä laajemmaksi keskusteluksi digitaalisuudesta. Lisäksi keskustelu laajeni toisinaan koskemaan teknologiaa ja digitaalisia laitteita: erityisesti Iiro painotti, että hänen suhtautumisensa ohjelmointiin sisältää myös itse fyysisen teknologian, johon ohjelmointi on sidoksissa. Pohjimmiltaan myös ohjelmoinnilla on fyysinen olomuoto: se on ykkösten ja nollien jonoja, jotka symboloivat tietokoneessa olevia muistipaikkoja (Dufva & Dufva, 2016, s. 97). Ohjelmointi ei siis ole irrallaan fyysisestä todellisuudesta, vaikka siltä helposti voikin tuntua.

Kaikki haastateltavat olivat yhtä mieltä siitä, että ohjelmoinnilla on iso rooli yhteiskunnassamme: se määrittelee päivittäin käyttämiemme laitteiden toimintalogiikkaa, ja useimmat ovat tavalla tai toisella tekemisissä ohjelmoitujen ympäristöjen kanssa. Ohjelmointi itsessään näyttäytyi lähes kaikkien haastateltujen puheissa usein vähän kätkeytyneenä; jonakin, mitä on kaikkialla, mutta mihin ei tule kiinnittäneeksi arjessa huomiota.

”Onhan se ohjelmointi koko ajan läsnä meidän yhteiskunnassa. Kaikki liikennevalot ja... kaikki katuvalaistukset, automaatiot, kassajärjestelmät... Kyl se on joka paikassa, et ei sitä ilman oikeen voi elää.” (Pinja)

”Mut onhan se jokapäiväisessä elämässä, kaikki on jollain tavalla kosketuksissa ohjelmoitujen asioiden kanssa, niin.. kyl se on must tärkeätä, tietää siitä [ohjelmoinnista] jotain.” (Pietari)

Esimerkiksi Pietari pohti, että ohjelmointi jää usein piiloon, sillä monet nykyaikaiset digitaaliset laitteet, kuten kosketusnäytöt ovat helppokäyttöisiä ja toimivat todella intuitiivisesti, eikä käyttäjä pääse itse ohjelmakoodia sisältävään osaan käsiksi. Tämä voi korostua erityisesti nuorilla, jotka ovat varttuneet suljettujen ja varsin intuitiivisten teknologioiden, kuten älypuhelinien parissa, eivätkä he välttämättä hahmota, mitä

käyttöliittymän takana on.

Ohjelmoinnin lukutaito on käsite, jonka olin odottanutkin nousevan haastateluista esiin. Haastatelluista kaikki korostivat ohjelmoinnin lukutaidon tärkeyttä mm. saavutettavuuden ja yhdenvertaisuuden näkökulmasta: ohjelmointi nähtiin niin merkittävänä yhteiskuntaa muokkaavana tekijänä, että sen ymmärtämisen koettiin olevan oleellinen taito yhteiskunnassa täysivaltaisena jäsenenä toimimisen kannalta. Osa haastateltavista kanssa otin ohjelmoinnin lukutaidon puheeksi, osa haastatelluista puhui ohjelmoinnin lukutaitoon viittaavista asioista oma-aloitteisesti. Osa haastatelluista käytti ohjelmoinnin lukutaito -käsitettä, osa ilmaisi saman asian kertomalla, että he toivoivat oppilaiden oppivan hahmottamaan, mitä ohjelmointi on ja missä kaikkialla sitä on, ja miten se vaikuttaa elämään.

Ohjelmoinnin lukutaito tekee näkymättömästä ohjelmoinnista näkyvää. Pinja kertoi oman opetuskokonaisuutensa yhteydessä, että hänelle on tärkeää tehdä ohjelmointi näkyväksi oppilaille keskustelun kautta: hän kertoi laittavansa oppilaitaan usein miettimään, missä kaikkialla ohjelmointia on, käyttäen esimerkkeinä niinkin arkisia asioita kuin liikennevaloja. Hän toki luokanopettajana opettaa kaikkia muitakin oppiaineita kuin kuvista oppilailleen, joten jäi epäselväksi, olivatko keskustelut ohjelmoinnin näkymisestä käyty pelkästään kuvataidetuntien yhteydessä vai myöskin muiden oppiaineiden tunneilla. Hänen opetuskokonaisuutensa yhteydessä Pinja kuvaili, että vaikka oppilaat eivät varsinaisesti koodanneet peliä, he toivottavasti oppivat hahmottamaan, minkälainen työmäärä on esimerkiksi pelin rakentamisen takana: oppilaat siis oppivat vähintäänkin ymmärtämään, että ohjelmoitua ympäristöt on joku tehnyt.

Joanna vertasi ohjelmoinnin hahmottamista muotoillun tai rakennetun ympäristön hahmottamiseen:

”Ehkä se on sellasta maailman hahmottamista, ensisijaisesti -- Että onhan se... niin se ymmärrys, nimenomaan koodaamisessa, se on hirveen tärkeä. Et mistä siin on oikein kysymys. Vähän samalla lailla, kun mä olin hirveen pieni, tai nuori, niin en mä aatellut, että on olemassa jotakin muotoilijoita, jotka muotoilee kaikki lamput ja tuolit... koska kukaan ei ollut mulle sellaista opettanut! En mä osannut ajatella, et miten ne on tähän maailmaan tupsahtaneet. Et mulle itelleni se oli hirveen herättävä ajatus, kun mä ymmärsin, että joku on ihan oikeesti ne muotoillut, tuolit, ja funtsineet, että ei ne oo vaan sieniiä sateella, vaan et joku on ne tehnyt.” (Joanna)

Ohjelmoitujen ympäristöjen ymmärtäminen rakennettuina ympäristöinä onkin ensimmäinen askel ohjelmoinnin lukutaitoon: jos ei ole koskaan tullut ajatelleeksi, että joku on suunnitellut kaiken ympärillämme, maailman hahmottaminen on erilaista kuin jos ottaisi kaiken annettuna. Pietari otti esiin kriittisen ajattelun näkökulman:

jos ohjelmoituja ympäristöjä ei hahmota rakennettuina ympäristöinä, on vaikea nähdä, minkälaisia vaikuttimia niiden takana on ja miten ne muokkaavat ihmisten käyttäytymistä.

”Mut kyllä olis hyvä tietää, koska se, että on jonkun asian pelkästään käyttäjä, mut ei tiedä miten se on tehty, niin silloin on vaikee havaita kaikkia niitä tapoja, joilla meihin vaikutetaan...” (Pietari)

Pietari totesi, että hän ei omassa opetuskokeilussaan juurikaan ottanut ohjelmoinnin lukutaitoon liittyviä yhteiskunnallisia tai kriittisiä ulottuvuuksia esiin, mutta hän pohti mahdollisuutta paneutua siihen enemmänkin opetuksessaan tulevaisuudessa. Haastatelluista Iiro nosti esiin, että hänelle ohjelmoinnin lukutaito vaikuttaa hänen opetuksensa taustalla leimaavana eli ohjelmoinnin opetusta jollain tapaa jopa määrittelevänä asiana - sen voisi katsoa siis olevan jopa yksi syy ottaa ohjelmointia mukaan taidekasvatukseen. Iiro puhui haastattelussa yleensä nimenomaan teknologiasta, sillä hänen lähestymistapansa ohjelmointiin sisälsi myös fyysiset laitteet, joihin ohjelmointi on kytköksissä.

Ohjelmoinnin lukutaidosta keskusteltaessa monet haastateltavista pohtivat sen suhdetta varsinaiseen ohjelmointitaitoon. Haastateltavista sekä Iiro että Pietari nostivat ohjelmoinnin lukutaidon tärkeämmäksi taidoksi oppia koulussa kuin varsinaisen ohjelmointitaidon.

”Kyl mä nään, et jos ei nyt ehkä ohjelmointitaito, niin ainakin ohjelmoinnin lukutaito on ihan kansalaistaito, mikä jossakin määrin pitäis opettaa koulussa” (Iiro)

Pietarin haastattelussa tuli lisäksi esiin ajatus, että ohjelmoinnin lukutaito voisi olla pysyvämpi taito kuin ohjelmointitaito: yksittäisessä ohjelmointikokeilussa käytetty ohjelmointikieli unohtuu nopeasti, mutta ymmärrys siitä, että on ohjelmoinut, voi auttaa ymmärtämään ohjelmoitujen laitteiden ja ympäristöjen toimintalogiikkaa.

”Vaikee sanoo et kuinka pysyvää se ohjelmoinnin osaamistaito on, et jos nyt laittais ne uudestaan, niin en usko, et kovinkaan moni pystyis itsenäisesti sen tehdä suoraan riviltä nolla aloittaen, mutta ehkä se tärkeempi on, että on joku kokeilu, et ymmärtää pikkasen sitä logiikkaa, et tajuaa näitä.. mikä siellä takana vaikka pyörii jossakin jutussa” (Pietari)

Mertala ja kumppanit (2020, s.39) nostavat esiin kysymyksen, voivatko ohjelmoinnin lukutaidossa keskeiset kriittisyys ja emansipatorisuus olla mahdollisia ilman jonkinlaista funktionaalista ymmärrystä siitä, miten koodia tuotetaan? Se, että ymmärtää ohjelmoitujen ympäristöjen toimintalogiikkaa, ohjelmoinnin mahdollisuuksia ja rajoituksia, antaa konkreettisia työkaluja ohjelmoinnin hahmottamiseen ja sen vaikutusten ymmärtämiseen. (Mertala ym., 2020). Ohjelmoinnin lukutaito ja konkreettinen ohjelmoinnin kirjoittamistaito eivät siis ole toisensa poissulkevia, vaan

päin vastoin parhaimmillaan toisiaan täydentäviä taitoja.

Ohjelmoinnin lukutaito näyttäytyi paitsi digitaalisen sivistyksen ja yhteiskunnallisen toimijuuden kannalta tärkeänä taitona oppia, myös mahdollisuutena iloa tuottavaan oppimisen kokemukseen. Joannalla oli omakohtainen esimerkki tilanteesta, jossa hän koki oivaltamisen iloa hahmottaessaan ohjelmoidun laitteen toimintalogiikkaa:

”Se [ohjelmoinnin lukutaito] liittyy siihen ymmärrykseen, et mitä se koodaus ylipäättään, mistä siinä on kysymys. Et jotenkin, kun mulle itelleni valkeni nää asiat, niin mä olin ihan tohkeissani jostain apteekin robotista, joka hakee sieltä hyllystä jotakin. Et [innostuu] ’miten voi olla tehty tollanen, toi on tosi hieno! Vau, osaisinpä koodata tuollaisen koodin!’ -- Vaik ei se varmaan ollenkaan vaikeeta oo. Et ymmärtää sen kuitenkin, et ahaa, kun annan käskyn noin, niin se kääntyy ja noin... Niin se oli sellanen hauska tilanne, että hahmottaa, että missä sitä [ohjelmointia] on. Ehkä se on just tätä lukutaitoo” (Joanna)

Olin olettanut, että **ohjelmoinnillisen ajattelun** käsite olisi tullut haastatte- luissa esiin voimakkaammin, onhan ohjelmoinnillinen ajattelu ollut paljon esillä opetusmaailman keskusteluissa. Ohjelmoinnilliseen ajatteluun liittyy ymmärrys siitä, miten ohjelmointi toimii käytännön tasolla. Dufvan & Dufvan (2016) metaforista ohjelmoinnillinen ajattelu lähestyy funktionaalisia metaforia, sillä siihen ei liity arvola- tatuneita ajatuksia ohjelmoinnista, ainoastaan toiminnallisuuteen liittyvää ajattelua. Kuitenkaan haastattelemani opettajat eivät arvottaneet ohjelmoinnin funktionaalisen osaamisen merkitystä kuvataiteen kontekstissa kovin korkealle, eikä siten ohjelmoin- nillisen ajattelunkaan teemat nousseet juurikaan esiin. Haastatelluista Pinja ilmaisi näkevänsä ohjelmointitaidon ja ohjelmoinnillisen ajattelun tulevaisuustaitona, josta voi olla oppilaille hyötyä heidän tulevaisuuden työnäkymiään ajatellen.

”No kylhän se on nyt tullut aika selväksi, et monissa tulevaisuuden aloissa [ohjelmointia] tarvitaan. Ihminen tarvitsee sitä ohjelmointitaitoa, tehdäkseen sitä työtä.” (Pinja)

Haastatelluista Pietari pohti ohjelmoinnillisen ajattelun luonnetta loogisena ajatteluna, joka ei etene samaan tapaan kuin arkijärjellä ajattelu.

”Ehkä siinä ohjelmoinnissa voi oppia... mä en tiää, onks se abstraktia ajattelua, se sellanen, et näkee mielessään jonkun asian, ja ei tehä sitä suoraan, sillä tavalla kun on totuttu tekemään, vaan se tehään, opitaan toinen ajattelulogiikka...” (Pietari)

Pietari oli asettanut opetuskokeilunsa tavoitteisiin myös ohjelmoinnilliseen ajatteluun liittyviä tavoitteita: hän kannusti oppilaita pohtimaan, mitä heidän kirjoittamansa koodinpätkät tekisivät, ja vasta sitten testaamaan, pitivätkö ennako- oletukset paikkaansa.

”Et aina kun kirjoittaa sitä koodia, niin pyrkis jo vähän etukäteen jo näkemään, tai ainakin ymmärtämään, et jos se nyt on täällä värjytussa, niin mitä siitä vois tapahtua, ennen kuin sen näkee sitten. Et voi testaila aina, pitikö ne omat näkökulmat tai arvaukset, paikkaansa.” (Pietari)

Pinja kertoi yhtenä opetuskokonaisuutensa tavoitteena olleen nimenomaan ohjelmoinnillisen ajattelun oppimisen. Nostan siksi hänen tekemänsä opetuskokonaisuuden esimerkkinä ohjelmoinnillisesta ajattelusta. Pinjan opetuskokonaisuudessa korostui ohjelmoinnillinen ajattelu ja ohjelmoinnin lukutaito keskustelujen kautta.

♥ Pelimaailma-konseptin luominen

Esimerkki ohjelmoinnillisesta ajattelusta

Pinjan opetuskokonaisuuden lähtökohtana oli pelimaailma. 5.-luokkalaiset suunnittelevat ja toteuttavat videon, jonka liikkuvat osat ohjelmoitiin liikkumaan Microsoft PowerPointilla. Ensin oppilaiden tehtävänä oli suunnitella käsin piirtämällä oma hahmo, sen jälkeen hahmolle suunniteltiin sille sopiva koti, ja kodin lisäksi maalattiin maisema paikasta, jossa koti voisi sijaita. Lisäksi oppilaat pohtivat, mitä muuta pelimaailmassa on, ja millainen peli olisi, ja pelin maailmasta tehtiin matkaesite. Pinjan mukaan ohjelmoinnillinen sisältö tuli mukaan, kun kaikki palaset olivat valmiina: tausta, koti ja hahmo kuvattiin pädeillä, hahmo ja koti muokattiin taustattomiksi ja siirrettiin PowerPointiin. Hahmot olivat ikään kuin omilla tasoillaan (layereillä), ja niille pystyi PowerPointissa piirtämään omat reitit, joita pitkin hahmot liikkuvat. PowerPoint ei ehkä perinteisesti ohjelmoinnin työkaluna pidetty ohjelma, mutta Pinja laajensikin ohjelmoinnin koskemaan myös visuaalisen ohjelmoinnin ja ohjelmoinnillisen ajattelun. Hän kuvaili, että ohjelmoinnillinen ajattelu tuli mukaan, kun oppilaat piirsivät reitit, joita pitkin hahmot liikkuisivat kuvassa. Pinja näki, että ohjelmointi oli työssä enemmänkin väline ja tavoitteet oppimisen suhteen tulivat kuvataiteen sisällöistä. Ohjelmointisisältöä oli ymmärrys siitä, että oppilaat oppivat ajattelemaan, missä kaikkialla koodia on, ja minkälainen työmäärä vaikka pelin tai elokuvan tekemiseen menee. Pinja kävi oppilaiden kanssa myös keskusteluja siitä, missä kaikkialla ohjelmointia on, ja sitä kautta teki ohjelmointia näkyväksi oppilaille.

”Et ne oppi sitä, et miten se reitti rakennetaan Powerpointilla, ja ymmärs osa ainakin, että mikä esimerkiksi työmäärä tarvitaan elokuvan tekemiseen. Et tuli vähän sellaista elokuvakasvatustakin, mikä on tietty kuviksen tavoitteita... Ja sit näki vähän sitä pelimaailman luomista, et mitä se vaatii kaikkiaan.” (Pinja)

Taiteellisen ajattelun ja ohjelmoinnillisen ajattelun suhde oli minulla haastattelussa yhtenä teemana. Haastattelussa minua kiinnosti selvittää, miten haastateltavat kokivat, että taide ja ohjelmointi eroavat toisistaan ja toisaalta minkälaisia yhtäläisyyksiä niillä on, sekä minkälaisia ajattelun taitoja näiden yhdistäminen voi tuottaa. Oletuksenani oli, että taiteellisesta näkökulmasta katsottuna maailman hahmottaminen voisi olla erilaista kuin jos maailmaa katsotaan puhtaasti ohjelmoinnillisten lasien läpi. Kysymyksenasetteluni haastattelutilanteessa oli kieltämättä hieman erikoinen ja kysymys oli muutenkin niin laaja, että siitä olisi voinut pitää kokonaisen, erillisen haastattelun, mutta onneksi asiantuntevilla haastateltavilla oli tarjota aiheeseen varsin kiinnostavia näkökulmia.

Kun kysyin Joannalta, mitä yhtäläisyyksiä hän näkee kuvataiteessa ja ohjelmoinnissa, vastaus tuli kuin apteekin hyllyltä:

”Luova ajattelu, ongelmanratkaisukyky... En mä oikeen muuta näin nopeesti keksi. Mutta ne on tosi tärkeitä toki, kummatkin. Ne tukee toisiaan.” (Joanna)

Myös muut haastateltavista kokivat sekä ohjelmoinnin että kuvataiteen molempien kehittävän **luovaa ajattelua** ja **ongelmanratkaisukykyä**.

” Hmm. No niissä on ainakin paljon samaa, koska ne molemmat edellyttää luovaa ajattelua, ja just sellaista ”outside the box”, että sä pystyt vapauttamaan mielen siihen luovaan kuvitteluun, mitä voi tehdä..” (Pinja)

”Ja ehkä just ajattelun taidot on mun mielestä sellainen tärkeä... Et sä pystyt luoviin ratkaisuihin, ja jos sä näät jonkun ongelman, niin sä lähdet tutkimaan sitä joka kantilta, et miten sä saat sen murrettua ja avattua... Et ehkä just niitä ajattelun ja logiikan kautta mä lähtisin sitä lähestymään.” (Pinja)

Haastatteluissa tuli myös puheeksi, että taiteessa tapahtuva luova ajattelu voi olla luonteeltaan erilaista kuin ohjelmoinnissa tapahtuva luova ajattelu. Iiro pohti, että vaikka ohjelmointi on hyvin luovaa, luovuus ohjelmoinnissa keskittyy ongelmanratkaisuun, asioiden yksinkertaistamiseen ja abstrahoimiseen ja siihen, miten fyysisessä maailmassa olevat ilmiöt voidaan yksinkertaistaa kaavioiksi ja algoritmeiksi. Erotuksena tästä taiteessa luovan ajattelun tarkoituksena on nimenomaan pyrkiä luomaan jotakin täysin uutta, eikä ratkaista jo olemassa olevia ongelmia. Kärjistäen voisi ilmaista, että Iiron haastatteluun perusteella taiteellinen luova ajattelu näyttäytyi monitahoistavana ja uusia näkökulmia avaavana, kun taas ohjelmoinnillinen luova ajattelu näyttäytyi yksinkertaistavana ongelmanratkaisuna.

Marjo Räsänen (2000) kirjoittaa, että taideoppimiseen liittyvä tietoisuuden kasvu voi muuttaa oppijan suhdetta itseensä ja ympäröivään maailmaan. Taidekasvatuksen yksi tehtävä Räsänen mukaan on luoda merkityssuhteita maailmasta, kasvattaa oppijan kokemusta ympäröivästä maailmasta. Ymmärtäminen ei ole mahdollista

ilman, että opittavaa asiaa suhteuttaa omaan henkilöhistoriaansa - pelkkä tieto ei siis riitä ymmärtämiseen, vaan oppiminen edellyttää, että oppijan koko persoonallisuus on mukana oppimisessa. Ymmärtäminen on aina subjektiivista, ja kytköksissä yksilön kokemusmaailmaan. (Räsänen, 2000, s. 15) Ohjelmoinnin ottaminen mukaan taideoppimisen osaksi voisi siis mahdollisesti kasvattaa tietoisuuden kasvua digitaalisista ilmiöistä. Kuvataiteen kriittinen luonne mahdollistaa myös ohjelmoinnin käytäntöjen kriittisen tarkastelun.

”ehkä... taide voi auttaa ohjelmoinnissa jäsentämään erilaisia ilmiöitä, ehkä yllättävistä ja uusista näkökulmista analysoimaan niitä.. ja toisaalta, ehkä myös kyseenalaistamaan ohjelmoinnin merkityksiä ja rakenteita ja se, et minkä takii joku ihminen on jonkun ohjelmointikielen tehnyt, jotakin asiaa varten, ja miksi se toimii tietyllä tavalla... minkä takia se tekee tietynlaisia oletuksia... vaikka siinä rakenteessaan...” (Iiro)

Pietari koki, että kuvataiteen vahvuutena on sen monipuolisuus. Kuvataiteeseen on mahdollista yhdistää mitä vain, ja siten käsitellä hyvinkin laajoja ilmiöitä. Ohjelmoinnin yhdistäminen kuvataiteeseen ja taiteelliseen ajatteluun voisi mahdollisesti hälventää taiteen ja tieteen rajoja, ja mahdollistaa digitaalisten ilmiöiden käsittelyn taiteellisesta näkökulmasta.

”Oon nähnyt kyllä tosi kiinnostavia taideteoksia, joissa on käytetty vaikka datamassojen käsittelyä, ja tällaisia tekoälytaideteoksia, ne on tosi kiinnostavia... Mikä ehkä veis ajattelua siihen, että taiteen ja tieteen rajat... Ehkä se voi yhdistää niitä... Tai ohjelmointi voi tuoda taidetta osaksi muita asioita, ja tuoda taiteeseen osa-alueita muilta kentiltä...” (Pietari)

Iiro pohti myös, että ohjelmoinnista perinteinen näkemys on, että ohjelmoinnin tulisi olla funktionaalista ja hyödyllistä, kun taas taiteeseen kuuluu oleellisemmin ajatus siitä, että kaiken ei tarvitse olla toimivaa ollakseen merkityksellistä. Ohjelmointi taidekasvatuksen kontekstissa voisi näin lähestyä Dufva & Dufvan (2016) koodi karnevaalina-metaforaa.

”Jos puhutaan ohjelmoinnin [ja kuvataiteen] opetuksen yhtäläisyyksistä ja eroista, niin toivoisin, et siinä olisi enemmän yhtäläisyyksiä ja samoja teemoja, mitä voitais pyöritellä, ja et se puhetapa yhtenäistyis... Se vaatis sitä, et rohkeesti kuviksen puolella omaksuttais niitä oikeita käsitteitä, mutta myös ehkä sellaista... Tavallaan hyväksyttäis se, että ei tarvii aina olla asiantuntija, jotta voi puhua ohjelmoinnista... Ja sitten jotenkin kuviksessa nään, että jos kuviksessa tehdään ohjelmointia, niin just se funktionaalisuuden ajatus... et jutun ei tarvikaan olla funktionaalista ollakseen merkityksellistä. Et joku pieleen mennyt ohjelmointi voikin synnyttää jonkun hauskan sattuman, minkä ympärille voi sit lähteä rakentamaan jotain muuta. Tai se voikin olla sitten se jutun juju.” (Iiro)

Haastattelujen valossa vaikuttaisi siltä, että ainakin teoriatasolla taiteellisen ajattelun yhdistäminen ohjelmoinnilliseen ajatteluun voisikin tuoda esiin Dufva & Dufvan (2016) emansipatorisiksi nimeämiä näkökulmia ohjelmointiin, ja kasvattaa ymmärrystä digitaalisuudesta ja ohjelmoinnista tuomalla myös kriittistä ajattelua mukaan ohjelmoinnin opetukseen.

Tuottaminen

Tuottamisen teemassa nostan esiin ohjelmoinnin ja taidekasvatuksen käytännön ulottuvuuden: minkälaista tuottamista ohjelmointi voi olla kuvataidekasvatuksen kontekstissa, ja minkälaista oppimista itse tekeminen voi tuottaa. Pohdin myös taidon merkitystä tuottamisessa. Taidekasvatuksessa moni tiedollinen asia, kuten ajattelun taidot, kumpuavat itse käytännön tekemisestä. Siten tuottaminen on erottamattomasti kytköksissä myös edellisen luvun ymmärtämisen teemaan.

Haastateltavien opetusprojektit toivat esiin yllättävänkin moninaisia tekemisen ja tuottamisen tapoja. Osassa opetusprojekteista tuottaminen oli hyvinkin konkreettista käsillä tekemistä: sekä Pinjan ja Joannan opetuskokonaisuuksissa visuaaliset elementit toteutettiin piirtämällä käsin tai digitaalisesti, ja Iiron kokeilut laajensivat käsillä tekemisen myös kolmiulotteisen rakentelun maailmaan. Sekä Pinjan, Joannan että Iiron opetusprojekteissa oli läsnä myös abstraktimpaa, tietokoneella tai pädillä tapahtuvaa tuottamista, ohjelmointia. Pietarin kokeilussa tuottaminen tapahtui käytännössä pelkästään tietokoneella, ohjelmakoodia kirjoittamalla.

Ohjelmointiin itsessään käytettiin monenlaisia tuottamisen välineitä, mutta tällä pienellä otoksella **visuaalisen ohjelmoinnin** välineet tuntuivat olevan kaikkein suosituimpia. Joanna, Iiro, ja Pinja kertoivat painottavansa nimenomaan visuaalisia ohjelmointiympäristöjä opetuksessaan. Joanna ja Iiro käyttivät Scratchiä; Pinja oli tehnyt oman opetuskokeilunsa myöskin hyvin visuaalisella PowerPointilla. Joanna kuvaili, että Scratch sopii hyvin kuvataiteen tarpeisiin, sillä Scratch visuaalisena ohjelmanä on helposti ymmärrettävissä kuvallisen työskentelyn yhteydessä:

”Se on musta sen takii kiva, se koodaaminen Scratchilla, et se perustuu siihen kuvaan. Ja me ollaan kuitenkin kuvallisia täällä kuviksessa. Että mä en tavallaan kaipaakaan sitä koodia sitten siihen. Et se on sitten matikan ja teknisten ja muiden juttu.” (Joanna)

Iiro kertoi, että hän suosii Scratchiä, sillä se on visuaalisena ohjelmanä helppo käyttää, se on yleensä oppilaille entuudestaan tuttu, ja se taipuu hyvin monenlaiseen ohjelmointiin: Scratchillä pystyy tarvittaessa tekemään pelejä, tai ohjelmoimaan mikrobittä tai Arduinoa. Iiro piti myös siitä, että Scratchillä ohjelmointi ”tuntuu kyhäilyltä” – Scratchin visuaalisuus ja sen tuoma mutkattomuus siis toi ohjelmointiin

ei-niin-vakavan ulottuvuuden:

”Mut se on vähän sellasessa... tai kun välillä rakentaa jotain laitteita, mitkä on sellasia pahvista ja jeesusteipistä tehtyjä, niin Scratchissä on vähän sama filis ohjelmointimailmassa — et se on vähän sellaista... purkkavirtelyä... ei ehkä ihan oikeeta ohjelmointia.” (Tiro)

Pinja toivoi, että visuaalisten ohjelmointialustojen käyttäminen laajentaisi oppilaiden käsitystä siitä, minkälaista ohjelmointi voi olla. Pinjalle oli tärkeää, että oppilaiden käsitys sisältäisi myös muunlaisen ohjelmoinnin kuin pelkän koodin kirjoittamisen, jotta he voisivat kokea ohjelmoinnin helposti lähestyttävämmäksi. Myös Pietarin opetuskokeilussa korostui visuaalisuus, vaikka hän ei haastattelussa asiaa tuonutkaan esille: hänen käyttämässään OpenProcessingissa kirjoitetaan koodia perinteisesti koodikielellä, mutta alusta on tarkoitettu nimenomaan visuaaliseen materiaalin työstämiseen. OpenProcessingin etuna on, että kirjoitetun koodin pääsee näkemään visuaalisena tuotoksena yhdellä klikkauksella: abstrakti koodi muuttuu siis konkreettisempaan visuaaliseen muotoon varsin kätevästi.

♥ Suomen kultakausi Scratchillä

Esimerkki visuaalisesta ohjelmoinnista

Joannan tehtävä yhdisti Suomen taidehistoriaa, tarinallisuutta ja visuaalista ohjelmointia Scratchillä. Joanna oli teettänyt saman tehtävän useampana vuotena, pienillä variaatioilla ryhmästä riippuen. Esimerkiksi seitsemäsluokkalaaisilla tehtävä yksinkertaisimmillaan meni seuraavasti:

”Eli ihan siis simppeleisti, sillä tavalla, et otetaan kultakauden hahmo. Leikataan se irti. Ja sit pitää olla joku toinen kultakauden hahmo, ja niiden välillä pitää olla dialogia, pitää olla liikettä, ja pitää olla musiikkia. Ja sit jos vielä kerkes, niin joku tausta sille.” (Joanna)

Joanna oli teettänyt samaa tehtävää myös varioiden valinnaisen digikuviksen ryhmälle, jolloin tehtävään laajennettiin esimerkiksi animoinnilla. Joanna korosti haastattelussa, että tehtävässä usein kuvataiteelliset sisällöt olivat kaikkein tärkeimpiä: esimerkiksi videossa näkyvän materiaalin piti olla itse tehtyä, eikä suoraan netistä kopioitua. Tehtävän tavoitteisiin kuului tavoitteita myös ohjelmoinnille, sillä tehtävässä tuli toteuttaa ohjelmoimalla dialogi hahmojen välille, sekä ääntä ja liikettä. Lopputuloksena tehtävästä syntyi hauskoja videopätkiä, jotka herättivät kultakauden ajan hahmoja henkiin, nykyaikaistettuina versioina.

Kaikki haastateltavat korostivat **tekemällä oppimisen** merkitystä ohjelmoinnissa. Tekemällä oppiminen korostui esimerkiksi niin, että osa haastateltavista kertoi kannustavansa oppilaitaan kokeilemaan ja tekemään itse, eivätkä he antaneet välttämättä heti oppilaille valmiita vastauksia. Pinja kertoi esimerkiksi kannustavansa oppilaita kokeiluun ja tutkimiseen, ja sitä kautta oppimiseen:

”Et ei se haittaa, vaikka vähän tökit sitä, et mikä noista napeista on se oikee, mistä se näyttö tulee päälle. Et antaa niille sellasta rohkeutta myöskin tutkia, niin se on musta tärkeit” (Pinja)

Tekemällä oppimisen, ”learning by doing”-kasvatusajattelun isä John Dewey (1859-1952) korosti pragmaattisessa kasvatusfilosofiassaan tiedon ja toiminnan kiinteää yhteyttä: ajattelu ja toiminta eivät ole koskaan toisistaan erillisiä, vaan kaiken oppimisen ytimessä on pohjimmiltaan kyse oppijan aktiivisesta vuorovai-
kutussuhteesta maailman kanssa (Anttila, 2017). Itse tekemällä opittava asia tulee konkreettisella tavalla osaksi oppija-kokijan elämismaailmaa, ja asian syvällisem-
mälle ymmärtämiselle avautuu mahdollisuus, jota ilman omakohtaista kokemusta tuottamisesta ei olisi. Haastatteluissa esimerkiksi Pietari pohti, että omakohtainen kokemus ohjelmoinnista voi auttaa oppilasta hahmottamaan, onko ohjelmointi subjektiivisesti tärkeää tai kiinnostavaa:

”Kun mun mielestä jos sanotaan vaan, et on ohjelmointi, mut sit ei oo koskaan kokeillut sitä, ei oo mitään kokemusta eikä tiedä siitä mitään, niin sit on vaikee myös sanoo, et onks se tärkeitä tai onks se kiinnostavaa...” (Pietari)

Aktiivinen tekemällä oppiminen oli läsnä haastateltavien kaikissa opetuskokeiluissa mutta myös siinä, miten he itse olivat opetelleet ohjelmointia. Joanna oli käynyt pitkään Scratch-koulutuksissa, mutta hän kertoi alkaneensa ymmärtää ohjelmointia paremmin vasta siinä vaiheessa, kun hän alkoi tehdä omia kokeiluja Scratchillä. Oma kokemukseni ohjelmoinnin opetteluun äärellä oli kulkenut samanlaista kaavaa: pelkkä valmiina annettujen tehtävien suorittaminen tai valmiiden ohjelmakoodien kopiointi ei tuonut ymmärrystä siitä, miten ohjelmointia voisi soveltaa tehtävän ulkopuolelle, vaan vasta kokeilut ja soveltavammat projektit ovat synnyttäneet aitoa oppimista.

Ohjelmoinnin ja taidekasvatuksen yhdistävä tekeminen voi poiketa siitä, minkälaisia töitä kuvataidetunnilla on normaalisti totuttu tuottamaan. Kysyin haastateltavilta, miten oppilaat olivat suhtautuneet siihen, että kuvataidetunnille oli tuotu ”poikkeuksellista” tuottamisen tapaa, ohjelmointia. Haastateltavien kokemukset vaihtelivat: Pinja kertoi oppilaiden suhtautuneen alusta asti hyvin, kun taas Iiro kertoi, että oppilaat saattoivat joskus osoittaa mieltään ohjelmoinnillisille aiheille kuvataiteessa esimerkiksi nostamalla äläkän: ”yöö, ei tää oo fyken tunti!”. Iiro tosin huomautti, että yleensä kyseessä oli ensireaktio ja oppilaat useimmiten innostuivat,

kun pääsivät kiinni itse tekemiseen. Pietarin kokemuksen mukaan se, että tunneilla lähdettiin nopeasti teoriasta käytäntöön, tekemään ja kokeilemaan hälvensi oppilaiden ennakkoluuloja ohjelmointia kohtaan kuvistunnilla.

”Must tuntuu, et se on monesti enemmän opettajilla mielikuva, et mitä oppilaat ajattelee, et kyllä aika avoimin mielin mä oon kokenut, että oppilaat lähtee asioihin mukaan. Ja sitten... mä koitin pitää silleen, aika nopeesti, et tässä olis nyt tää abstrakti juttu, ja sit tosi nopeesti jo tehään se ensimmäinen juttu... Eli täs on nyt tää musta neliö, okei nyt tehään se musta neliö... Et mahollisimman nopeesti yhdistäis sen johonkin konkreettiseen, ja pääsee myös tekee, niin ehkä se helpotti myös sitä.” (Pietari)

Se, että ohjelmointi kuvataiteessa on erilaista tekemistä kuin mitä kuvataiteessa on totuttu tekemään, monipuolista useammankin haastateltavien mielestä oppilaiden käsitystä siitä, mitä kaikkea kuvataide voi olla. Lisäksi Joanna ja Pietari kertoivat, että ohjelmointi antoi sellaisillekin oppilaille mahdollisuuden loistaa, jotka eivät perinteisesti olleet olleet hyviä koulukuviksessa esimerkiksi hienomotoristen taitojen puutteen vuoksi.

”Tietysti ne tyypit, jotka on silleen digin kans tekemisissä, niin ne on niinku.. Päässeet näyttään osaamistaan. Ne, joilla ne hienomotoriset taidot, on vähän niin ja näin...” (Joanna)

Erityisesti Iiron näkökulma tuottamiseen korosti nimenomaan **käsillä tekemistä**, mikä on aika luontevaa, onhan hän myös käsityön opettaja. Käsillä tekeminen on kiinnostava lähtökohta ohjelmoinnin ja taidekasvatuksen yhdistämiselle, sillä ohjelmoinnin ja digitaalisen maailman ylipäätään on kritisoitu vieraannuttavan ihmisen käsillä tekemisestä ja sitä kautta luonnosta (Dufva, 2017, s. 134). Käsillä tekeminen on ihmiselle, kädelliselle eläimelle, hyvin ominaista toimintaa: käsityön tutkija Seija Kojonkoski-Rännäli on kirjoittanut, että ihmisen koko olemassaolo niin perustavanlaatuisella tavalla yhteydessä käsillä tekemiseen, että käsillä tekeminen sitoo ihmisen maailmaan. Käsillä tekeminen on paitsi olemassa olemisen myös ymmärtämisen tapa: käsillä tekemällä voi käsillä ymmärtää eli käsittää ympäröivää maailmaa. (Dufva, 2017). Käsillä tekeminen ja hienomotoriset taidot liittyvät myös oleellisella tavalla kuvataiteeseen, onhan kuvataide perinteisesti nähty nimenomaan taitoaineena ja kuvataide-oppiaineen olemassaolon yhtenä perusteluna on pitkään käytetty juuri kädentaitojen harjaannuttamista (ks. esim. Pohjakallio, 2005).

Käytännön tasolla Iiron kokeiluissa käsillä tekeminen näyttäytyi niin, että hänen tekemissään opetuskokeiluissa ei ainoastaan ohjelmoitu tietokoneella tai pädillä, vaan myös rakennettiin itse käsin laite, johon sitten ohjelmoitiin toiminnallisuutta. Rakentaminen ja käsillä tekeminen kuuluivat siten erottamattomasti osaksi ohjelmointia. Iiro kertoi ajattelevansa toisinaan ohjelmoinnin opetusta modernin kädentaidon opettamisena: ohjelmoinnin ja käsillä tekemisen ei tarvitse siis olla

toisensa pois sulkevia. Iiro otti esiin, että hänen mielestään käsillä tekemisen mukana pitäminen ohjelmoinnissa voisi parhaimmillaan saada oppilaat suhtautumaan digitaaliseen teknologiaan aktiivisesti tekijänä, eikä pelkästään passiivisena vastaanottajana.

”Niin tavallaan se, kun etti käsin tehtyä digitaalisuutta tai... sitä käsin tekemisen mukana pitämistä siinä digitaalisuuden ja ohjelmoinnin käsittelyssä, niin... jotenkin et voisko sillä lisätä, sitä vastuun kokemista maailmasta? Jaa tavallaan sit toisaalta myös se, että se on mun mielestä keino siihen, että hälventää sellasta laitepelkoa ja laiteihmetystä... mustan laatikon ajatusta, et ne laitteet ei oo vaan laatikoita, jotka joku on jossakin tehnyt, ja meille annettu, vaan et niitä voi itekin, ehkä vähän yksinkertaisemmassa mittakaavassa, mut kuitenkin, niitä voi ite suunnitella ja avata ja... nähdä, mitä siellä on sisällä... Ja jotenkin alkaa rakentaa semmosta aktiivista suhdetta siihen digitaaliseen maailmaan ja teknologiaan.” (Iiro)

Ohjelmoinnin näkeminen kädentaitona liittyi siis Iiron näkökulmasta vahvasti siihen, että laitteita voitiin avata, väärinkäyttää, tutkia ja muokata. Myös Dufva (2017) esittää, että laitteiden ottaminen omakseen suhtautumalla niihin hakkerimaisella tutkivalla uteliaisuudella, voisi synnyttää aktiivista toimijuutta ja siten myös vastuunottoa digitaalisissa ympäristöissä (Dufva, 2017, s. 138). Iiron seuraava lainaus kiteyttää hyvin, miten ymmärtäminen on sidoksissa tuottamiseen. Itse käsin tekemällä myös syvällisempi oivaltaminen, ja siten esimerkiksi vastuullisten ajattelun taitojen kehittyminen, voi olla mahdollista:

”Tai et kun huomio kiinnittyy tekemisen kautta joihinkin vaikka laitteisiin, tai kun puhutaan niistä ja niit pääsee itse availemaan ja tekemään ja tutkimaan, muokkaamaan, niin... miten se oma ajattelu sitten ehkä muuttuu... kun on vaikka jotain laitetta heittämissä pois, ja hankkimassa uutta, et.. tuleeks ehk se oivallus, et täältäähän vois jotain vaik tehokkaammin kierrättää, ja osata vaalia tollasia... et osaa ajatella ja tiedostaa tollasia kysymyksiä. Tai et voisko rikkinäisen laitteen korjata sittenkin, et mikähän siellä on rikki, et siellä saattaa olla joku yks pieni liitin vaan rikki... et herää ehkä toivottavasti uteliaisuus sellaseen korjaamisen etiikkaan, tai.. Et nehän on sit suoraan aika vahvoja yhteiskunnallisia kysymyksiä.” (Iiro)

Esimerkkinä käsillä tekemistä korostavasta opetuskokeilusta nostan Iiron oppilaiden tekemän karkkiautomaatin. Kyseisessä kokeilussa korostui käsillä tekemisen lisäksi myös muotoilun näkökulma, mikä kytkee käsillä tekemisen luontevasti nimenomaan kuvataiteen kontekstiin.

”Niin jotenkin sen [karkkiautomaatin] mä löisin ehdottomasti kuviksen sisältöihin muotoilun näkökulmasta.. Et jotenkin tuntuu, et monesti tää ohjelmointi liittyy siihen, et kun tehdään joku juttu toimivaks, niin sit kuviksen sisältö tulee vaikka muotoilun kautta. Et se on enempi semmoista, muotoilun teemojen ympärillä pyörivää.” (Iiro)

♥ Karkkiautomaatti

Esimerkki käsillä tekemisestä ja muotoilusta

Iiro rakensi alakoululaisten kanssa kolikkoautomaatin, joka sylki karkkia ulos, kun laitteen sisälle laittoi kolikon. Automaatti oli alusta asti käsin rakennettu, ja automaatin ympärille oppilaat saivat rakentaa haluamansalaisen kotelon. Lopputuloksena oli hammaspeikon näköinen otus, joka sylki suustaan karkkeja ulos kolikkoja vastaan. Kokeilu yhdisti leikkisällä ja hausalla tavalla ohjelmoinnin, käsillä tekemisen ja muotoilun.

Tuottamisen teemaan liittyy myös ohjelmoinnin ja taiteen tekemisen tapojen välinen suhde, ja minkälaisena tuottamisena haastateltavat kokivat ohjelmoinnin taidekasvatuksen kontekstissa. Useimpien opetusprojektien kohdalla **ohjelmointi** näyttäytyi **välineenä tehdä taidetta**. Oli tavallaan odotettavissa, että ohjelmoinnin välineellinen arvo korostui kuvataidekasvatuksen kontekstissa: käsittelihän haastatteluni lähtökohtaisesti nimenomaan kuvataiteen oppiaineessa tapahtunutta ohjelmoinnin opetusta, jolloin kuvataiteen sisällöt olivat ikään kuin pääruokalajina ja ohjelmointi mausteena. Painotus olisi saattanut mennä toisin päin, jos olisin haastatellut ohjelmoinnin opettajia, jotka käyttävät kuvataiteellisia menetelmiä ohjelmoinnin opetuksessa.

Esimerkiksi Pinja kertoi, että hänen opetuskokonaisuudessaan ohjelmointi oli ensisijaisesti väline tehdä taidetta ja tavoitteet tulivat pääasiassa kuvataiteen sisällöistä. Ohjelmoinnilliset sisällöt, kuten ohjelmoinnin lukutaito ja ohjelmoinnillinen ajattelu, olivat toki kokonaisuudessa mukana, mutta kuvataiteelliset oppimistavoitteet olivat painotetumpia. Samoin Joannan tehtävässä korostuivat taiteelliset sisällöt, vaikkakin hän oli nostanut tavoitteeksi myös konkreettisia ohjelmoinnillisia sisältöjä, kuten ohjelmointivälineen (Scratchin) käytön oppimista. Pietari pohti, että hänelle oli helpompi mieltää ohjelmointi kuvan tekemisen välineeksi, sillä hän ei kokenut olevansa ohjelmoinnissa ammattilainen, toisin kuin kuvataiteessa. Välinearvo ohjelmoinnille oli siis osittain omista lähtökohdista johtuvaa.

Iirokin korosti käytännönläheistä ja välineellistä suhtautumista ohjelmointiin. Hän kuvaili, että monia hänen kokeilujaan yhdisti se, että ohjelmointi on väline jonkin toiminnallisuuden saavuttamiseksi, ja että hän harvemmin otti ohjelmoinnillisia sisältöjä pelkästään ohjelmoinnin itsensä vuoksi, vaikka pitikin ohjelmointia tärkeänä myös itsessään.

”Mut sitten ehkä mistä se tarve omaan ohjelmoinnin opetukseen on tullut, et on maksanut sitä ohjelmointia, et mistä se on lähtenyt, niin... aika puhtaasti tarpeesta, et on tarvinnut rakentaa jotain vaik toiminnallisuutta – et pitää vaik jotain liikettä saada, et no otetaan tää moottori... no moottori pitää saada päälle, sit sitä pitää hallita. – se ehkä näkyy omassa opetuksessakin, et tavallaan... koittaa tota kautta rakentaa

oppilaiden kiinnostusta aiheeseen... et ne kokis tarpeelliseksi jotain, ekaks, ja sit lähetään opettelemaan, miten se tehdään.” (Iiro)

♥ Abstraktia taidetta koodilla

Esimerkki luovasta ohjelmoinnista

Pietari tutustutti oppilaat 1900-luvun abstraktiin maalaustaiteeseen ja OpenProcessing-ohjelmointikieleen. OpenProcessing (<https://www.openprocessing.org>) on selaimessa toimiva ohjelmoinnin alusta, joka mahdollista luovan ohjelmoinnin varsin helposti: kirjoittamaansa koodia pystyy kokeilemaan saman tien, ja näkemään toimiiko se odotetusti. Pietari lähti hyvin nopeasti oppilaiden kanssa tekemään koodausharjoituksia esimerkkiteosten pohjalta: esimerkiksi Malevitšin Musta neliö toteutettiin saman tien koodaten. Yksinkertaisten harjoitusten jälkeen oppilaat toteuttivat omat abstraktit teoksensa kirjoittamalla koodia OpenProcessingilla.

”Meil oli lähtökohtana abstrakti taide, ja sitten oli tarkoitus luoda tällainen oma liikkuva, abstrakti teos, ja ottaa siihen koodauksesta sisältönä muuttuja-käsite... mikä sit ajattelin, että on myös matematiikassa tullut siihen seiskaluokalle se, että on semmosia asioita, kuin muuttujia... Ja sit lähettiin samaan aikaan kattomaan esimerkkejä siitä maalaustaiteesta, missä oli se, että nää nyt ei mitenkään liiku, ja kun on vaikka tehty 1900-luvun alussa maalaustaidetta, niin silloin ei ollut tällaista teknologiaa, missä asiat vois saada liikkumaan... Ja sitten ihan Mustasta neliöstä lähdettiin liikkeelle, ja se oli myös ensimmäinen koodausharjoitus, et kuinka voidaan luoda tällainen hyvin simppele asia, kuin musta neliö, siihen taustalle...” (Pietari)

Useimmat oppilaista saivat tehtyä työn, jossa oli geometrisia muotoja ja jonkinlaista liikettä, sekä vähintään yksi muuttuja. Töissä saattoi olla interaktiivisuutta hiiren kanssa: esimerkiksi klikkaamalla hiiren oikealla painikkeella kuviota se vaihtoi väriä valkoisesta siniseksi, vasemmalla painamalla väri vaihtui oranssiksi. Edistyneimmät oppilaat olivat itsenäisesti tutkineet OpenProcessing-sivustoa ja muokanneet teostaan sieltä löytyneillä koodipätkillä. Yhdessä työssä esimerkiksi pystyi hiirtä raahaamalla piirtämään kuvioita teokseen.

Pietari kuvaili, että kokeilun aikana he pohtivat oppilaiden kanssa fyysisen ja digitaalisen maailman eroja: miten esimerkiksi orgaaniset muodot ovat luonnossa hyvin tyypillisiä, mutta ohjelmoinnissa on helpompi lähestyä asioita geometrinen muotojen kautta. Opittavia asioita kokeilussa oli lisäksi tutustua abstraktiin taiteeseen, RGB-värijärjestelmään sekä ohjelmoinnin perusteita kuten muuttuja-käsite. Pietari kuvaili, että oppilaille esimerkiksi abstrakti taide tuli kokeilun kautta tutuksi, eikä hänen ole myöhemmin tarvinnut enää muistutella, mitä sana ”abstrakti” tarkoittaa, eli ohjelmointi toimi välineenä paitsi taiteen tekemisessä myös abstraktin taiteen käsitteen ymmärtämisessä.

Taiteen tekeminen ohjelmoinnin avulla lähestyy teoriatasolla luovan ohjelmoinnin ajatusta: luova ohjelmointi laajentaa ohjelmoinnin koskemaan laajempaa taiteellista toimintaa, johon liittyy koodi jollain tapaa (Dufva, 2018, s. 43). Oikeastaan mikä tahansa haastattelemistani opetusprojekteista voisi olla esimerkkinä ohjelmoinnista taiteen tekemisen välineenä ja luovasta ohjelmoinnista. Nostan Pietarin opetuskokeilun esimerkkinä (ks. esimerkki edellisellä sivulla), sillä se oli ainoa, jossa ohjelmointi tapahtui pelkästään koodia kirjoittamalla.

”Ja ehkä tässä se ohjelmointi on enemmän tekniikka, kuvan tekemisen tapa, kuin sisältö, mikä ois pakko oppia kuviksessa.. Mut tässä se oli tapa luoda abstrakti taideteos.”
(Pietari)

Haastatteluista viimeinen tuottamiseen liittyvä näkökulma oli **luovuus** ja siihen oleellisesti liittyvä harjaantuminen eli **taito**. Luovuus liitetään sekä taiteeseen että ohjelmointiin kumpaankin, ja tämä tuli ilmi myös monissa haastatteluissa. Sekä ohjelmoinnista että taiteesta puhuttiin luovana tuottamisena. Varto (2011) kuvaa luovuuden suhdetta taitamiseen seuraavasti (s. 29):

”Luovuus on keskeinen kuvaaja taiteen ja taitojen kohdalla: kyse on uudenlaisen synnyttämisestä ja uudenlaisessa merkitysyhteydessä näkemisestä. Luovuus-sanaa käytetään tavoilla, jotka eivät ole aina perusteltuja, joten taidepedagogiikassa on kuvattava selvästi, mitä tarkoitetaan luovuudella silloin, kun on kyse taidoista. Harjaantunut taitaminen takaa tekijälle uuden luomisessa toimintatilan, jossa tekijä on selvillä luomisesta, sen askelista, uhkista etenemisessä, tuloksen tunnistamisesta ja arvioimisesta. Tällä tavoin luovuus on taitamisen ominaisuus ja tavoitteellista toimintaa, mikä on taiteenkin idea.”
(Varto, 2011)

Haastateltavista opettajista Iirolla oli laajin kokemus ohjelmoinnista. Hän oli tehnyt monia erilaisia kokeiluja viiden vuoden aikana peruskoulussa, ja lisäksi oli kiinnostunut ohjelmoinnista myös taiteellisessa toiminnassa. Pinja ja Pietari olivat ottaneen ohjelmoinnillisia sisältöjä opetukseensa vasta melko hiljattain, ja heidän opetusprojektinsa jäivät yhteen kokonaisuuteen, sillä erotuksella, että Pietari oli vetänyt saman kokeilun monelle eri ryhmälle. Joanna oli tehnyt kokeiluja Scratchillä useamman vuoden ajan, mutta ei ollut ottanut ainakaan toistaiseksi muita ohjelmia tai ohjelmointisisältöjä mukaan opetukseensa, ja hänellä oli usein saman tyyppinen tehtävä eri ryhmille.

Kokemuksen määrä näkyi vastauksissa, mikä tietysti on aivan luonnollista. Iirolla erilaisten ohjelmointialustojen, -kielten ja ohjelmien käyttö oli tuttua, eikä hän tuonut esiin vaikeuksia itse ohjelmointitaidon kannalta, kun taas Pietari ja Joanna kumpikin toivat haastattelussa useasti esiin, että eivät ole ohjelmoinnin ammattilaisia, ja että heille jo ohjelmoinnin opettelemisessa on ollut haasteita. Sekä Pietari että Joanna korostivat harjaantumisen merkitystä paitsi ohjelmoinnissa, myös missä tahansa

luovassa toiminnassa.

Pietari pohti, että sekä ohjelmointiin ja kuvataiteeseen liitetään luovuus molempiin, mutta hän koki, että hänelle itselleen on vaikeampaa saada luovat ideat toimimaan ohjelmoinnilla kuin perinteisillä taiteen tekemisen välineillä.

”--niin ainakin itelle erona on se, että on vaikeeta saada luovat ideat toimimaan, koska se ei oo niin tuttu väline, sellainen, mistä tietäis tarkalleen, mitä tehdä, vaan siinä tosi pienistä asioista voi olla kiinni, et joku asia ei toimi..”(Pietari)

Joanna nosti harjaantumisen tärkeäksi myös opettajalle. Hän koki, että opettajana on varmemmalla pohjalla silloin, kun on jonkinlaista käytännön kokemusta siitä, mitä kulloinkin opettaa.

”Tai sanotaan nyt näin. Opettajalla pitää olla kyllä jonkinlaiset perusteet, kun lähtee sinne digipuolelle, et tokihan se on sama juttu, et jos sä opetat piirtämistä, etkä oo ikinä piirtänyt, niin vähän huonohan se on sit sitä mennä opettamaan. Et sä voit laittaa tietysti jonkun tuubivideon tuolta pyörimään, ja niinhän mä jonkun verran teenkin oppilaiden kanssa – Kyl siin pitää vähän jotain itekin tietää. Tai sanotaan näin, että on varmemmalla pohjalla.” (Joanna)

Lisäksi Pietari otti esiin, että oma harjaantumattomuus ohjelmoinnissa saattoi hidastaa itse opetusta: kun hän meni auttamaan oppilaita ohjelmointiongelmassa, hänellä saattoi kestää, ennen kuin virhe ohjelmointikoodissa löytyi.

”se välillä korostu, jos virheen löytäminen kesti itsellekin jonkun aikaa.. Et mikä siellä mättää...” (Pietari)

Pietari kuvaili, että oman ohjelmointitaidon loppuminen kesken saattoi aiheuttaa välillä turhautumisen tunteita. Hän koki, että ohjelmoinnin opetteleminen itsenäisesti oli antanut hänelle ymmärryksen siitä, mitä kaikkea ohjelmoinnilla voisi tehdä, mutta kun itse ohjelmointitaitoa ei ole tarpeeksi, hänellä ei aina ollut välineitä toteuttaa ideoitaan.

”Ja se on ehkä näissä, mitä mä oon näiden ihan perusjuttujen jälkeen ite yrittänyt tehdä, niin jotenkin sitä tietää, että tällaista ja tällaista vois tehdä, mut taito loppuu kesken.. Ja se ymmärrys, et miten tää kuuluis tehdä, ja sit se ei vaan toimi, ja turhauttaa tosi helposti...” (Pietari)

Pinja näki, että sekä perinteisiä kuvataiteen menetelmiä kuin ohjelmointiakin voi molempia käyttää luovaan kuvitteluun. Pinja vertasi, että ohjelmoinnissa on omat välineensä, jotka toimivat kuin kynä ja paperi piirtämisessä. Harjaantumisen merkitys on luettavissa rivien välistä: kun käytössä olevaa välinettä osaa käyttää, voi vapauttaa mielensä luovaan kuvitteluun ja toteuttaa lähes minkälaisia ideoita tahansa. Samanlaisia ajatuksia on pohtinut mm. Casey Reas, yksi Processing-kielen tekijöistä: hän on todennut, että ohjelmointi voi olla paitsi tapa ajatella, myös piirtämiseen

verrattava, kuin itsestään soljuva tapa toteuttaa ideoitaan (Dufva, 2018, s. 43), jos vain on tarpeeksi harjaantunut.

”Hmm. No niissä [ohjelmoinnissa ja kuvataiteessa] on ainakin paljon samaa, koska ne molemmat edellyttää luovaa ajattelua, ja just sellaista ”outside the box”, että sä pystyt vapauttamaan mielen siihen luovaan kuvitteluun, mitä voi tehdä... Kynällä ja paperilla on helppo tehdä, kun se on luontainen lapselle, lähtee luomaan joku oma kiva piirustus... Mut kyl ku koodatessa on ne omat työkalut, mitkä toimii sit kynänä, ja se tietokoneen ruutu sit paperina... Et kun vaan oppis sen, että ne on vähän kuin sama asia, niin pystyis luomaan melkein mitä tahansa.” (Pinja)

Iiro ei tuonut esiin omakohtaisia haasteita ohjelmointiin liittyen, mutta hän pohti oppilaiden kannalta tuttuuden ja harjaantumisen merkitystä luovuuteen. Hän nosti esimerkiksi esiin, että hän usein käyttää Scratchiä ohjelmoinnin opetuksessa siksi, että alusta on oppilaille entuudestaan tuttu. Oppilaiden on huomattavasti helpompi lähtee toteuttamaan ideoitaan ohjelmalla, jonka käyttöä ei tarvitse opetella alusta asti uudelleen.

”ja sitten kun se on ehkä myös tullut sitä kautta aina se Scratch, että... kun jotenkin se oma opetus perustuu niin vahvasti siihen, että saadaan joku juttu toimimaan ohjelmoimalla.. Ja sit antaa niille vapautta valita ite sen työkalun, millä... niin se aika luontevasti niillä tulee.” (Iiro)

Haastateltavien kokemusten valossa näyttäisi siis siltä, että niin taiteen tekemisessä kuin ohjelmoinnissakin luovuus on kytköksissä harjaantumiseen. Sillä ei ole merkitystä, onko luovuuden välineenä kynä vai koodi, kunhan väline on tuttu ja siihen on kehollinen tuntuma, harjaantunut ote helpottaa luovien ideoiden toteuttamista. Liian vaikean tekniikan edessä turhautuminen helposti ottaa vallan, ja oppiminen häiriintyy.

Kokemuksellisuus

Kokemuksellisuuden teeman kautta käsittelen haastatteluissa nousseita ajatuksia kokemuksellisesta (taide)oppimisesta, tunteista, jakamisesta ja yhteistyöstä. Lopuksi kuvailen lyhyesti, miten haastateltavat kuvailivat kokemuksiaan ohjelmoinnin yhdistämisestä kuvataiteeseen.

Monet kokemuksellisuuden teemaan liittyvät asiat kumpusivat haastattelujen oppimista käsittelevästä osiosta, ja siksi onkin minulle aika luontevaa lähteä käsittelemään kokemuksellisuutta nimenomaan **kokemuksellisen (taide) oppimisen** näkökulmasta. Kokemuksellinen oppiminen nostaa kehollisuuden oppimisen keskiöön. Se on tuttua kaikille ihmisille: sitä tapahtuu jokapäiväisissä arkisissa tilanteissa, usein huomaamatta, kun ihminen on kehollisesti läsnä maailmassa ja peilaa kokemuksiaan omaan elämänhistoriaansa. Esimerkiksi lasten oppiminen perustuu luonnostaan kokemukselliselle oppimiselle (Snellman, 2018, s. 96). Marjo Räsänen (2000) kokemukselliseen taideoppimisen mallissa oppiminen onkin laajamuotoista toimintaa, jossa kokemuksellisuus ja käsitteellisyys vuorottelevat. Kokemuksellisia aineksia ovat aistilliset, emotionaaliset, intuitiiviset, elämäkokemuksiin liittyvät asiat, sekä taiteellinen toiminta. Käsitteellisiä ovat analyttiset tulkinnat, taidehistoriasta ja estetiikasta kumpua tieto. Käsitteellistämisen ja kokemuksellisen välillä tapahtuu jatkuvaa vuorovaikutusta: esimerkiksi sanallistamisen kyvyn kehittyminen herkistää havaintoa. (Räsänen, 2000, s. 14)

Useimmat haastateltavien opetusprojekteista edustivat tulkintani mukaan kokonaisvaltaista ja kokemuksellista taideoppimista. Pinja kuvaili, että hänen opetuskokonaisuutensa oli nimensä mukaisesti kokonaisuus: mitään osaa ei olisi voinut jättää pois, sillä fyysisestä taidetyöskentelystä, abstraktimmasta ohjelmoinnista ja tekemisen yhteydessä käydystä pohdinnasta muodostui taiteellista toimintaa sisältävä kokonaisvaltainen tapahtuma. Yhdenkin osan poisjättäminen olisi muuttanut koko oppimistapahtuman luonnetta. Kun kysyin haastattelussa Pinjalta, miten hän erittelisi opetuskokonaisuutensa ohjelmoinnillista sisältöä ja taidekasvatuksellista sisältöä, Pinja ajatellut, että osat olisivat erotettavissa toisistaan.

”No en mä oikeestaan erittelis. Mä aattelen, että ne on sellainen käsi kädessä menevä paketti, että jos sä et ois tehnyt niitä osia käsin, niin et sä voi ohjelmoida niitä. Ja jos et sä ohjelmoi niitä menemään digitaalisesti näitesti siellä, niin ei se lopputuotos synny siinä. Ehkä siinä oli, et aluks tehtiin se niinku käsin, se tehtäväpuoli, ja sit taas oli se digitaalinen puoli, mut kyllän se digitaalisuus sit, kun siitä lopputuloksesta tehtiin videoanimaatio, niin onhan sekin tavallaan kuviksen tuotos.” (Pinja)

Joannan tehtävissä ja Pietarin opetuskokeiluissa oppilaiden oma kokeileminen, tekeminen ja tuottaminen yhdistyi käsitteelliseen tietoon taidehistoriasta - kokemus,

käytäntö ja teoria siis vuorottelivat. Iiron kokeiluista hyvin monet olivat jo lähtökoh-
taisesti hyvinkin kokemuksellista käsillä tekemistä, kokeilua, ja keskustelujen kautta
kokeiluihin usein liittyi käsitteellistämistä, asioiden pohtimista ja sanallistamista.

Kokemuksellisesta oppimisesta hyvä näkökulma on leikki ja leikin kautta
oppiminen. Leikkiessä oppiminen tapahtuu kuin luonnostaan, ja esimerkiksi erilaisia
rooleja on mahdollista kokeilla, ilman paineita. Erityisesti Iiro pohti, että hänelle oli
tärkeää, että leikillisuus on mukana hänen tekemissään opetuskokeiluissa:

*"Joo, kyllä se on ehdottomasti mulle tosi tärkeä lähtökohta, et sellainen leikillisuus
on mukana, kaikkes tekemisessä... Ja ehkä sen takia, se sellanen, tai kokeiluiks näit ihan
mielellään itekin sanon, et ei niistä itekään oo aina ihan varma, et mihin ne päätyy.
Pikemminkin koittaa aina luoda sellasta turvallista kokeilun kenttää niille oppilaillekin,
et ois mahis innostuu ja kokis, et kaikki voi siihen osallistuu ja kaikki voi niit omaksua."
(Iiro)*

Tunteet ovat erottamaton osa kokemuksellisuutta, ja haastateltavat toivatkin
esille monenlaisia tunteita, joita opetusprojektit olivat herättäneet heissä itsessään
ja oppilaissa. Suhde opeteltavaan asiaan määrittyy yhä uudelleen ja uudelleen
vuorovaikutuksessa ulkoiseen ja sisäiseen maailmaan, yksilön tunteisiin: tunteet
voivat parhaimmillaan edistää oppimista, esimerkiksi oppimisen iloa kokiessa, tai
vesittää sen, kuten pelon tai voimakkaan turhautumisen ilmaantuessa. Esimerkiksi
uuden välineen käytön opettellessa, kehon tulee oppia rentoutumaan uuden välineen
käsittelyssä, jotta oppiminen on mahdollista, (Snellman, 2018, s. 97), mikä tukee
edellisen luvun ajatusta luovuuden ja harjaantumisen välisestä yhteydestä.

Edellisessä tuottamista käsittelevässä luvussa kuvailin, miten turhautuminen
välineen harjaantumattomassa käsittelyssä saattaa pahimmillaan estää oppimisen.
Ohjelmoinnin palasteleminen pieniin osasiin, tarpeeksi helposti saavutettaviin
tavoitteisiin kuvataidetunnilla mahdollisti kuitenkin opetuksen järjestämisen niin,
että turhautumista pystyttiin parhaimmillaan välttämään. Turhautumista enemmän
Pietari ja Joanna korostivatkin haastattelussa onnistumisen kokemuksia ja oppimisen
iloa ohjelmoinnin opettelemisessa.

*"Mä huomasin, et monelle se, et kirjoittaa jotain, mikä näyttää eka oudolta, ja sit
painaa jotain, ja sieltä tulee joku juttu, mikä toimii, niin se saattoi olla pieni onnistumisen
kokemus sitten." (Pietari)*

*"Niin tietysti ne nyt oppii varmaan jonkun verran käyttämään sitä ohjelmaa, sitten
ne oppii ehkä ajattelemaan, et miten se koodaus toimii, jossei se oo niitä hirveesti ennen
koskettanu... ja sit mä aattelin, että ne voi saada siitä myös oppimisen iloa, kun ne
näkee, et hei, et mä saan tän jutun toimimaan." (Joanna)*

Onnistumisen kokemukset ovat hyvin tavoiteltavia oppimisessa, sillä ne vahvistavat
yleensä oppilaan kiinnostusta aiheeseen, ja voimistavat oppijan itsetuntoa. Oppimisen

kokemukselliseen luonteeseen liittyy aina haaste opettajalle, sillä kokemukset ovat hyvin yksilöllisiä, ja kumpuavat yksilöllisistä kokemuksista ja tulkinnoista. Iiro otti esiin, että oppilaiden koulukuviksessa saamia kokemuksia on kenties mahdotonta opettajana täysin hallita:

”mm.. Ja... Sit toivottavasti sellasta... oppivat kyvykkyyden kokemusta, tai kokemusta siitä, et joku juttu ei ehkä ookaan niin outo, kuin mitä se ens alkuun saattaa vaikuttaa, tai niin vaikee.. Voi toki käydä päinvastoin, et joku kokee, on epäonnistunut, ja kammoksuu entisestään... et se on kyllä missä tahansa taito- ja taideaineen sisällössä, et... saattaa tulla niitä ns. traumaattisia kokemuksia koulukuviksessa.” (Iiro)

Iiro pohti lisäksi, että kuvataiteen kontekstissa on kenties luontevampaa suhtautua ohjelmointiin kokemuksellisena tekemisenä kuin monessa muussa oppiaineessa. Kuvataiteessa on luonnollista lähestyä asioita monitulkintaisesti, monesta eri näkökulmasta, ja ottaa mukaan tunteisiin ja kokemuksiin liittyvää sisältöä.

”Niin, ja jos vielä kuviksen näkökulmasta mieltii, sitä ohjelmointia versus kässänkin näkökulmasta, niin monesti ehkä siinä kuviksessa on luontevampaa tuoda niitä sellasia kysymyksiä teknologian kokemisesta ja siitä, et miltä se saa asiat näyttämään, ja tuntumaan ja toimimaan.” (Iiro)

Kokemuksellisuus on tietyllä tapaa sisäänrakennettuna kuvataiteeseen, ja sitä kautta Iiro näki mahdollisuutena lähestyä myös ohjelmointia enemmän tunteiden kautta, tuntuman ja kokemuksen näkökulmasta. Kokemuksellisuuden kautta luonteeltaan hyvin abstraktista ohjelmoinnista on mahdollista tehdä kuvataidetunnilla käsin kosketeltavaa, ja keskustella myös tekemisen aikaansaamista tunteista, ja siitä, minkälaisia tunteita teknologia ja ohjelmointi herättävät.

Jakamisen ja yhteistyön ulottuvuuden olisi voinut melkein nostaa omaksi teemakseen, mutta niputan sen osaksi kokemuksellisuutta, sillä näkemykseni mukaan kokemuksellisuuteen liittyy vahvasti myös jaettu kokemus, muiden kanssa yhdessä tekeminen, ja sitä kautta oppiminen.

Monille, myös minulle pinttynyt mielikuva koodaamisesta on, että puhumattomat, antisosiaaliset koodarit ohjelmoivat salamyhkäisesti yksin kellarissa silmät kiiluen, näppäimistö sauhuten ja energiajuoma virraten. Haastattelujen valossa ohjelmointi ainakin kuvataiteen kontekstissa näyttäytyi kuitenkin päin vastoin varsin sosiaalisena toimintana. Kaikkien haastateltavien opetusprojekteissa oli mukana jollain tavalla yhteistyön ja jakamisen ulottuvuus: osalla haastatelluista, kuten Joannalla ja Iirolla, ohjelmointitehtävät saattoivat olla lähtökohtaisestikin yhdessä toteutettavia pari- tai ryhmätehtäviä, ja yksilötöinäkin toteutetuissa opetusprojekteissa oppilaat saattoivat pyytää apua opettajan lisäksi myös toisiltaan. Jakamista tapahtui usein viimeistään siinä vaiheessa, kun työt alkoivat olla valmiita: oppilaat usein esittelivät töitä

oma-aloitteisesti toisilleen, ja esimerkiksi Pinja kuvaili aluksi tehtävään kapinallisesti suhtautuneiden oppilaiden olleen lopulta ylpeitä töistään, ja halunneen näyttää niitä muille. Joanna kuvaili kokeneempien oppilaiden auttaneen muita:

”Kun ne tekee parityönä tän, niin ne auttaa toinen toisiansa siinä sitten. Ja sit se, joka on tehnyt... Monet on tehnyt alakoulun puolella enemmän ennen, niin ne on sellasia tutori-henkilöitä sit näis tilanteis.” (Joanna)

Iiro korosti, että hänen tekemistään opetuskokeiluista todella monet olivat lähtökohdaisesti yhteistyöprojekteja: oppilaat työskentelivät yhteisen teoksen tai työn äärellä, ja osallistuivat projekteihin omien vahvuuksiensa mukaan. Yhtenä esimerkkinä hyvin monipuolisesta yhteistyöstä nostan Iiron tekemän piirtorobotti-kokeilun.

♥ Piirtorobotti

Esimerkki yhteistyöstä

Iiro oli rakennuttanut 5.-6.-luokkalaisten kanssa valinnaisessa peliohjelmoinnissa ja robotiikassa robotin. Oppilaat olivat rakentamisen yhteydessä opetelleet, miten robotin sai mikrobotilla liikkumaan. Koulun erikoispäivänä Iiro ja yksi oppilaista pitivät ykkös-nelosluokkalaisten opetustuokion, missä he ohjelmoivat mikrobotin kääntymään, niin että napista a se kääntyi vasemmalle ja napista b oikealle, ja kun molemmat napit olivat painettuna, robotti kulki suoraan. Sitten he kiinnittivät tussin robottiin, ja antoivat robotin liikuttamisen alaluokkalaisten tehtäväksi.

”—ja annettiin näille pienemmille oppilaille tehtäväksi, et niiden pitää rakentaa ite tällainen aa bee-ohjelmointikieleen perustuva algoritmi, ja sit ne kirjoitti paperiin a, b, bbb, a, b, a + b, b, ab, +, a + b, ja niin es päin, ja sitten tää yks viitos-kutosluokkalainen, se valinnaiskurssilainen oli siin sit laitteen käyttäjänä ja pyöritteli sitä robottia siinä isolla paperilla. Ja sit siihen syntyi joku sattumanvarainen kuva -- sit just pohdittiin niitten oppilaitten kanssa, et kukas tässä nyt on se teoksen tekijä ja kuka se taiteilija on tässä... ja tällasia kysymyksiä.” (Iiro)

Iiro kertoi, että heillä oli lopuksi oppilaiden kanssa hyvää keskustelua siitä, kuka on teoksen taiteilija: onko se robotin tekijä, koodin kirjoittaja vai se, joka on ohjaillut robottia. Lisäksi hän mainitsi, että konetta käyttänyt oppilas sai kokemuksen asiantuntijuudesta, sillä oppilas oli ollut alusta asti mukana rakentamassa robottia, ymmärsi miten se toimi, ja pystyi tarvittaessa korjaamaan robotin, mikäli robotin piuhat lähtivät irti. Piirtorobotti-kokeilussa yhteistyötä olikin useammassa kohtaa ja monilla tavoilla toteutettuna: robotti oli rakennettu alun perin yhdessä, ja yhteistyötä laajennettiin vielä lisää tuomalla robotti alaluokkalaisten ohjelmoitavaksi. Ohjelmoidun robotin seuraaminen yhdessä oppilaiden kanssa oli yhteisöllinen ja yhteistyöstä kumpuava tapahtuma. Lopuksi oppilaat keskustelivat yhdessä opettajan kanssa kokeilun merkityksestä, ja oletettavasti rakensivat ymmärrystään taiteilijuudesta ja tekijyydestä yhteisen keskustelun kautta.

Yhteistyö digitaalisessa ajassa ei ole sidottu aikaan tai paikkaan. **FLOSS** (kirjainyhdistelmästä Free/Libre and Open Source Software) on näkökulma ohjelmointiin, joka korostaa avoimuutta, ihmisten välistä yhteistyötä ja jakamista verkon yli. Haastateltavien käyttämistä ohjelmointialustoista ja -ohjelmistoista Scratch, Arduino ja OpenProcessing ovat kaikki vapaita ohjelmistoja perustuvat avoimelle lähdekoodille, eli ne ovat lähtökohtaisestikin jakamiseen ja yhteistyöhön kannustavia alustoja. FSF:n (Free Software Foundation) määritelmän mukaan vapaan ohjelmiston tunnusmerkkeihin kuuluu, että ohjelmistoa voi käyttää mihin tahansa käyttötarkoitukseen; sitä saa tutkia, muokata ja käyttää vapaasti; ohjelmiston lähdekoodia saa kopioida ja jakaa muiden auttamiseksi; ohjelmiston muokattuja versioita saa jakaa muille niin että ne hyödyttävät koko yhteisöä (Dufva, 2018, s. 35). FLOSS:in voi nähdä paitsi vastalauseena kapitalistiselle omistajuudelle: avoimen ohjelmiston koodin omistajuus ei ole yksilöllä, vaan yhteisöllä. FLOSS-ideologiaan kuuluu, että jokaisella on oikeus pystyä näkemään ohjelman lähdekoodi, ja muokkaamaan sitä niin halutessaan, ja se siten nostaa esiin yhteiskunnallisia kysymyksiä paitsi ohjelmoinnin demokraattisuudesta, myös yksityisyydensuojasta: jos käyttäjä pysty tarkistamaan, minkälaista koodia hänen käyttämässään ohjelmistossa on takana, hänellä ei ole mitään mahdollisuuksia varmasti tietää, mitä ohjelmisto oikeasti tekee, ja minkälaista tietoa se kerää. (Dufva, 2018, s. 36).

Sekä Iiro että Pietari kertovat tehneensä avointa lähdekoodia ja avoimia ohjelmistoja näkyväksi opetuksessaan, ja kannustaneensa oppilaita tutkimaan muiden tekemiä koodeja, muokkaamaan ja oppimaan muilta. Pietari kuvaili ilmiötä yleistajuisesti seuraavasti:

”Jonkin verraan puhuttiin myös sitten tietenkin pelimaailmasta, että kun täällä OpenProcessingissäkin on se kiva, että ihmiset voi näit omii koodejaan jakaa... et voi olla vaikka peli, ja sitä pelii voi pelata, ja sitä voi myös muokata, jos haluu... et pääsee vähän oppimaan muilta, miten noita juttuja tehään...” (Pietari)

Kaiken kaikkiaan jakaminen ja yhteistyö nähtiin lähinnä positiivisten mielle yhtymien kautta: haastateltavat esittivät yhteistyön ja jakamisen lähtökohtaisesti hyvänä asiana, jopa työskentelyn yhtenä tavoitteena. Ainoana ”varjopuolena” yhteistyöstä ja jakamisesta Iiro otti esiin, että yhteistyö muiden kanssa ja koodien jakaminen vaikeutti lopputulosten arviointia.

”Vaik se tuntuu itellekin aika luontevalta käyttää ohjelmointia opetuksen sisältönä, niin jotenkin sen arvioiminen on hirmu vaikeeta... -- ku se ei oo muuta kuin tekemistä ja erehtymistä, tekemistä ja erehtymistä... ja sit otetaankin kaverilta toimiva koodi, ja huomataan, et noniin, nyt tää toimii -- Mikä on siis hyvä! Koska sitähan se ohjelmointi just on, et sit jaetaan koodeja, ja opitaan sellaseen lainaamiseen, leikkaamiseen ja liimaamiseen... Ja sit kun se ohjelmointi on nois omis projekteissa selvästi palvellut jotakin muuta tavoitetta, et saadaan joku laite toimimaan. Ja sit monesti nää on ollut

yhteisprojekteja, mitä ollaan tehty, et jokainen on niihin projekteihin osallistunut omien vahvuuksien mukaan.”(Iiro)

Iiron pohdinnoista on kuitenkin huomattavissa, että arviointia tärkeämmäksi hän nostaa sen, että oppilaat oppivat jakamisen kulttuuriin ja pääsevät käyttämään omia vahvuuksiaan yhteisprojekteja tehdessä. Ohjelmoinnin näkeminen lainaamisena, leikkaamisena ja jakamisena voikin siis parhaimmillaan kannustaa oppilaita yhteistyön tekemiseen, ja sitä kautta merkityksellisten kokemusten kartuttamiseen.

Lopuksi kuvailen vielä lyhyesti, minkälainen **kokemus** ohjelmoinnin yhdistäminen taidekasvatukseen oli haastattelujen perusteella. Haastattelujen pohjalta voin sanoa, että pääsääntöisesti se näyttäytyi kaikille haastateltaville positiivisena kokemuksena. Haastateltavat kokivat oppineensa paljon uutta opetusprojektiansa aikana, kaikki heistä kuvailivat ymmärryksensä ohjelmoinnista kasvaneen opettamisen myötä. Pietari lisäksi pohti, että hänelle entuudestaan hieman vieraat ohjelmointia sivuavat taiteen maailmat, kuten monille oppilaille läheinen pelimaailma, ovat avautuneet kokeilun myötä paremmin. Iirolla oli samankaltaisia ajatuksia, hän kuvaili oppineensa paljon oppilastuntemusta, erityisesti minkälaiset ohjelmointiin liittyvät aiheet ja kulttuurit koskettavat oppilaita. Joanna ilmaisi tuntevansa ammatillista ylpeyttä siitä, että oli ottanut ohjelmointia kuvataiteen opetussisällöksi.

*”Mutta kyllä sitä vähän henkseleitä paukutellen sitten laittaa, että voidaan tehdä myös tällaista siellä digikwiksessa tai seiskalla, että -- voidaan tehdä pientä koodausta.”
(Joanna)*

Kokemuksiin oli liittynyt myös haasteita. Ohjelmointisisällöt kuvataiteen tunnilla olivat monesti oppilaillekin vieraita, vaikka ohjelmointia onkin tuotu tiiviisti osaksi peruskoulun opetussuunnitelmaa. Erityisesti Pinja ja Pietari kuvailivat, että opetus oli todella usein hyvin tiivistä lähiopetusta, kun oppilaita piti auttaa hyvinkin yllättävien, pienten ongelmien kanssa. Kärsivällisyyttä vaadittiin niin oppilailta kuin opettajaltakin, sillä ohjelmointi on hyvin virheille altista toimintaa: pienenkin virheen etsimiseen saattoi mennä yllättävän paljon aikaa opettajalta, ja sillä aikaa apua tarvitsevien oppilaiden jono kasvoi entisestään. Pietari esimerkiksi pohti nauraen, että eivät ehkä välttämättä koronapandemian aikaisessa etäopetuksessa teettäisi ohjelmointitehtäviä oppilailleen juurikin lähiopetuksen vaatimuksen vuoksi.

”Mitähän mä sanoisin suurimpina haasteina... No kyl varmaan sen välineen alltius kaikille pienille virheille. Et jos joku pieni asia on väärin, niin mikään ei toimi, ja sit jos mikään ei toimi, niin sit se ei oo kivaa – ja sit jos ei oo kivaa, se virheen etsiminen voi olla aika vaikeeta.. Ja sitten se ei oo kovin intuitiivista, jos sitä ei osaa, vaan kaikki asiat pitäis hyvin tarkasti miettiä, et okei, tää menee tähän ja sit siit... Joo. Eli tarkkuus ja kärsivällisyyden vaatiminen.” (Pietari)

”Ties käytännössä, et miten ne tehdään, mut sit kun lähti oikeesti tekeen, kun ne oppilaat kompastuu kaikkeen maholliseen mihin vaan voi kompastua siinä tehdessä...” (Pinja)

Haasteista huolimatta kaikki haastateltavista ilmaisivat aikomuksensa jatkaa ohjelmoinnin ja taidekasvatuksen yhdistämisen parissa tavalla tai toisella. Osalla oli jo ajatuksia siitä, minkälaisia projekteja he voisi tehdä jatkossa. Iiro kertoi mitä luultavimmin jatkavansa leikillisen ja kokeellisen työskentelyn parissa, Pietari pohti mahdollisuutta ottaa enemmän ohjelmoinnin lukutaitoon viittaavaa sisältöä opetukseensa. Joanna toivoi, että voisi tehdä jotakin nykytaiteellisempaa ja ohjelmoinnillisesti haastavampaa, mutta hän mainitsi, että sitä varten täytyisi löytyä tarpeeksi helposti omaksuttava ohjelma tai alusta. Pinja pohti, että ei ehkä lähtisi tekemään oppilaidensa kanssa yhtä isoa kokonaisuutta uudestaan kylmiltään, vaan tekisi aluksi yksinkertaisempia johdattelevia harjoituksia oppilaiden kanssa, ennen isompaan projektiin siirtymistä.

”Kyllä mä varmaan voisin näitten kanssa jonkun tälläsen tehdä myös jatkossa, että... Oli opettava kokemus niille, oli opettava kokemus mulle, niin jäi ihan silleen hyvät fiilikset.” (Pinja)

Onnistumisia oli tapahtunut puolin ja toisin, niin oppilaille kuin opettajillakin, ja kaiken kaikkiaan yhteenvetona haastatteluista uskallan vetää yhteenvetona, että haastateltaville ohjelmoinnin yhdistäminen taidekasvatukseen oli näyttäytynyt ensisijaisesti positiivisena kokemuksena.

6. Loppupäätelmät

Yhteenvedo: minkälaista tietoa tutkielma tuotti?

Tutkin opinnäytteessäni, minkälaisia taidekasvatuksen ja ohjelmoinnin yhdistäviä kokeiluja ja opetusprojekteja on tehty uuden OPSin ilmestymisen jälkeen peruskouluissa. Minua kiinnosti myös, minkälaisia kokemuksia opetusprojekteja tehneillä opettajilla oli ohjelmoinnin ja taidekasvatuksen yhdistämisestä. Vastauksia tutkimuskysymykseeni lähdin etsimään teemahaastattelun keinoin: haastattelin neljää opettajaa, joista kolme oli kuvataideopettajia ja yksi luokanopettaja. Haastateltavilla oli kaikilla käytännön opetuskokemusta ohjelmointia taidekasvatukseen yhdistelevästä kuvataideopetuksesta. Kolme opettajista koki olevansa melko aloittelijoita ohjelmoinnin suhteen, yksi oli tehnyt ohjelmoinnin ja taidekasvatuksen yhdistäviä kokeiluja jo pidempään.

Kaikkia haastateltavia opettajia yhdisti kiinnostus ohjelmointia kohtaan ja he lähtökohtaisesti suhtautuivat positiivisesti ohjelmointia ja taidekasvatusta yhdistelevään opetukseen: ohjelmoinnin opettamista kouluissa kaikki opettajat pitivät ylipäätään hyvänä asiana. Haastateltavat näkivät erityisesti ohjelmoinnin lukutaidon merkityksellisenä taitona oppia koulussa. Kuvataide nähtiin hyvin monipuolisena aineena, johon voi yhdistää mitä tahansa, myös ohjelmointia. Kuvataiteen kontekstissa asioiden oppiminen tapahtuu moniulotteisesti ja kokemuksellisesti, itse tehden ja tuottaen, ja sitä kautta laajempaa ymmärrystä rakentaen. Itse ohjelmointitaitoa ei pidetty kuvataiteen kontekstissa erityisen tärkeänä: tärkeämpää kuin puhdas koodi oli se, että oppilaat saivat kokemuksen ohjelmoinnin monipuolisista mahdollisuuksista. Haastatteluissa ohjelmointi nähtiin ongelmanratkaisuna, mutta myös luovana toimintana, mutta haastatteluissa tuli esiin, että kuvataiteen kontekstissa ohjelmoinnilla voi tehdä muutakin kuin ”vain” ratkaista ongelmia. Haastattelujen perusteella ohjelmointi näyttäytyi välineenä toteuttaa taiteellisia ideoita. Ohjelmointiin tarkoitetut visuaaliset käyttöliittymät houkuttelivat helppoudellaan: käytetyistä ohjelmista ylivoimaisesti suosituin oli Scratch, mutta opetusprojekteissa oli mukana myös Microsoft PowerPointilla, Arduinolla ja OpenProcessingilla tehtyjä töitä. Scratchin käyttöä perusteltiin sen helppoudella, se oli helppo oppia opettajalle, ja myös usein se oli oppilaille valmiiksi tuttu.

Haastattelun teemoitteluosiossa (eli aineiston analyysissä) nostin esiin kolme erilaista näkökulmaa ohjelmoinnin ja kuvataiteen yhdistämiseen: ymmärtämisen, tuottamisen, ja kokemuksellisuuden ulottuvuuden. Ymmärtämisen teeman kautta käsittelin haastatteluissa nousseita ajatuksia siitä, minkälaista tiedollista ymmärtämistä

ohjelmoinnin ja taidekasvatuksen yhdistäminen voi tuottaa. Ohjelmoinnin lukutaito, ohjelmoinnillinen ajattelu sekä taiteellisen ajattelun ja ohjelmoinnillisen ajattelun suhde nousivat ymmärtämisen teemassa keskeisiksi aihealueiksi. Taiteellista ajattelua ja ohjelmoinnillista ajattelua nähtiin yhdistävän luova ajattelu ja ongelmanratkaisu: sillä erotuksella, että taiteellinen ajattelu lähestyy asioita holistisesti ja pyrkii laajentamaan näkökulmia, ja ohjelmoinnillinen ajattelu pyrkii yksinkertaistamaan asioita ja luomaan maailmasta yksinkertaistettuja malleja.

Tuottamisen teeman kautta erittelin haastatteluissa ilmi tulleita käytännön tekemiseen liittyviä kokemuksia. Taidekasvatuksessa moni ymmärtämiseen liittyvä asia kumpuaa itse käytännön tekemisestä, joten haastatteluista nostamani tuottamiseen liittyvät aihealueet tekemällä oppiminen, käsillä tekeminen, sekä luovuuden ja harjaantumisen (taidon) välinen suhde sivusivat toisinaan myös ymmärtämisen teemaa. Ohjelmointi nähtiin kuvataiteen kontekstissa haastattelujen perusteella yleensä ensisijaisesti taiteen tekemisen välineenä, ja esimerkiksi puhtaan ohjelmakoodin tuottaminen nähtiin hyvin toissijaisena asiana taidekasvatuksessa. Virheet ohjelmointikoodissa nähtiin taidekasvatuksellisesti jopa kiinnostavina mahdollisuuksina, jotka voivat synnyttää jotakin uutta ja jännittävää.

Kokemuksellisuuden teemassa nostin esiin haastateltavien ajatuksia opettamisesta kokonaisvaltaisena toimintana. Tunteiden merkitys kaikessa oppimisessa, mutta myös ohjelmoinnin ja taidekasvatuksen yhdistämisen äärellä olivat merkityksellisiä haastateltaville: opettajat halusivat tarjota onnistumisen kokemuksia ohjelmoinnista, mutta yksi haastateltavista pohti myös, että kokemusten hallitseminen on hyvin haastavaa, ellei jopa mahdotonta. Kokemuksellisuuden teemassa nostin esiin myös ohjelmoinnin ja taidekasvatuksen sosiaalisen ulottuvuuden: haastattelujen valossa ohjelmointi taidekasvatuksen kontekstissa näyttäytyi monessakin kokeilussa yhteistyöhön ja jakamiseen kannustavana toimintana.

Olin aidosti iloisesti yllättynyt, miten monipuolisia opetusprojekteja haastattelemani opettajat olivat tehneet: vain neljällä haastattelulla sain aineistoksi kuvaukset opetusprojekteista, jotka lähestyivät ohjelmoinnin ja taidekasvatuksen yhdistämistä sekä sisällöiltään että tekniikoiltaan hyvinkin monenlaisista näkökulmista. Ohjelmoinnillisista sisällöistä esittelemissäni opetusprojekteissa ilmeni mm. ohjelmoinnillisen ajattelun, ohjelmoinnin lukutaidon, ohjelmointitaidon, avoimen lähdekoodin ja useiden eri ohjelmointialustojen käyttöä. Taidekasvatuksellisista sisällöistä opetusprojekteissa oli hyvin laajasti erilaisia teemoja pelimaailmasta taidehistoriaan ja muotoiluun. Taiteen tekemisen tekniikoista mukana oli digitaalista piirtämistä ja kuvankäsittelyä, käsin piirtämistä ja maalaamista, rakentelua, installointia sekä luovaa ohjelmointia.

Tutkimieni opetusprojektien ilahduttavasta laajuudesta huolimatta opetus-

projektit eivät tuoneet esiin kaikkia ohjelmoinnin ja taidekasvatuksen yhdistämisen mahdollisuuksia, mitä nyt ei tietenkään voi edes olettaa neljän haastattelun perusteella. Ohjelmoinnin emansipatorinen puoli, miten ohjelmointi vaikuttaa yhteiskunnassa, jäi näiden opetuskokeilujen valossa melko vähäiselle huomiolle: ohjelmoinnin lukutaitoon viitattiin monesti, ja haastatelluilla oli paljon ajatuksia emansipatorisen näkökulman huomioimisesta taidekasvatuksen ja ohjelmoinnin äärellä, mutta yksikään opetusprojekti ei varsinaisesti suoraan käsitellyt tai tehnyt ohjelmoinnin yhteiskunnallisia vaikutuksia näkyväksi. Emansipatorisuus ilmeni kuitenkin joissakin opetusprojekteissa esimerkiksi oppilaiden ja opettajan välisten keskustelujen kautta, kun oppilaiden kanssa esimerkiksi mietittiin, missä kaikkialla ohjelmointia on. Lisäksi kaksi opettajista toivat esiin, että jo se, että on kokemus ohjelmoinnista tekijänä, voi auttaa helpommin ottamaan ohjelmoituja ympäristöjä omakseen. Jäin kuitenkin kaipaamaan taidekasvatuksellista projektia, joka olisi taidetyöskentelyn kautta tehnyt ohjelmoinnin yhteiskuntaa muokkaavan voiman jollakin tapaa selkeämmin ymmärrettäväksi.

Opetuskokeiluista yksikään ei myöskään käsitellyt esimerkiksi ohjelmointia sivuavia, tällä hetkellä ohjelmointimaailmassa kuumia aiheita kuten tekoälyjä tai koneoppimista, vaikka taidemaailmassa esimerkiksi tekoälyllä tuotettu taide onkin jo olemassaoleva ilmiö (vrt. Du Satoy, 2019, ss. 146–149). Toki koulumaailmassa tekoälyn ottaminen mukaan vaatii opettajalta jo huomattavasti laajempaa teknologista ymmärrystä kuin pelkästään yhden ohjelmointiympäristön hallintaa, mutta näkisin, että pienemmässä mittakaavassa esimerkiksi yläkouluissa voisi olla mahdollista toteuttaa esimerkiksi pienimuotoisia valmiita datakirjastoja käyttäviä taideprojekteja, tai tutkia aihetta kuvataiteellisen työskentelyn kautta.

Yksi näkökulma, jonka olisin mielelläni ottanut voimakkaammin mukaan opinnäytteessäni, oli leikillisuus ohjelmoinnin ja taidekasvatuksen yhdistämisessä. Jätin leikin kuitenkin pois, sillä haastatelluista vain Iiro kertoi, että hänelle leikkisyys oli hyvin tärkeä osa ohjelmoinnin opetusta, muiden haastatteluissa leikki ei tullut ollenkaan esiin keskusteluissa, enkä oikein saanut sitä istumaan mukaan muihin teemoihin. Tulevaisuudessa voisi olla kiinnostavaa lähestyä ohjelmoinnin ja taidekasvatuksen yhdistämistä nimenomaan leikin näkökulmasta, ja tutkia, minkälaisia avauksia leikin näkökulma toisi keskusteluun taiteesta ja ohjelmoinnista.

Olen tässä tutkielmassa esitellyt muutamia tapoja tuoda ohjelmoinnillisia sisältöjä osaksi kuvataidekasvatusta. Minusta kuitenkin tuntuu, että olen vasta raapaissut pintaa. Ohjelmoinnin ja taidekasvatuksen risteyskohdissa on varmasti vielä paljon sellaisia avauksia ja mahdollisuuksia, joita en ole tullut tätä tutkielmaa tehdessäni edes ajatelleeksi. Tutkielmani valossa vaikuttaisi siltä, että ohjelmoinnin ja kuvataidekasvatuksen yhdistämisellä on kuitenkin lukuisia mahdollisuuksia, joista varmasti

suuri osa jää vielä tutkimattomaksi maastoksi. Opettajan oma kiinnostus aiheeseen voi synnyttää hyvinkin monenlaisia opetuskokonaisuuksia, ja ohjelmoinnin tuominen kuvataidekasvatukseen voi avata kiinnostavia näkökulmia digitaalisesti rakennetun maailman ilmiöihin.

Tutkielman luotettavuudesta

Huomasin opinnäytettä tehdessäni, että tutkimuskysymykseni oli paljon laajempi kuin olin osannut etukäteen ajatellakaan. Jälkeenpäin olen ymmärtänyt, että aiheen laajuus tuli jo siitä, etten ollut rajannut esimerkiksi ohjelmointia koskemaan tiettyä osaa ohjelmoinnista, vaan olin jo alusta asti olin suhtautunut ohjelmointiin hyvin laajana digitaalisuutta määrittelevänä ilmiönä. Haastattelussa ilmi tulleiden näkökulmien laajuus tuotti minulle hankaluuksia erityisesti aineiston analyysiä tehdessä. Haastatteluja tehdessä itse tutkimuskysymykseen ei ollut aivan täysin muotoutunut mielessäni lopulliseen muotoonsa, joten en myöskään osannut kysyä tarpeeksi tarkentavia kysymyksiä erityisesti koskien opettajien kokemuksia opettamisesta: oma harjaantumattomuuteni haastattelijana luultavasti vaikutti asiaan myös. Selkeämmin rajattu tutkimuskysymys olisi voinut tehdä tutkielman etenemisestä suoraviivaisempaa ja helpompaa minulle itselleni, mutta toisaalta olen iloinen siitä, miten monipuolisia näkökulmia laaja ja ”löysähkö” tutkimuskysymys ja haastattelurunko lopulta tuotti.

Opinnäytteeni tutki aihetta yksittäisten tapausesimerkkien kautta, eikä se siten anna yleistettävää kuvaa ohjelmoinnin ja taidekasvatuksen yhdistämisen nykytilasta Suomessa. Tämän tutkielman perusteella en siten pysty ottamaan kantaa siihen, onko ohjelmoinnin ja taidekasvatuksen yhdistämisessä kyse laajemmasta ilmiöstä vai kenties vain pienestä kuriositeetista kuvataidekasvatuksen kentällä. Jatkotutkimuksena voisikin olla kiinnostavaa hyödyntää määrällisen tutkimuksen menetelmiä ja selvittää laajemmalla otoksella, kuinka moni kuvataideopettaja ylipäätään on ottanut ohjelmoinnillisia sisältöjä mukaan opetukseensa. Toisena huomiona tuon esiin, että kaikki tässä tutkielmassa haastatellut opettajat olivat myös lähtökohtaisesti kiinnostuneita ohjelmoinnista, ja he olivat oman ammatillisen kiinnostuksensa vuoksi halunneet ottaa ohjelmoinnillisia sisältöjä mukaan kuvataideopetukseensa. Voisikin olla mielekästä selvittää laajemmin kuvataideopettajien käsityksiä ohjelmoinnista ja tutkia, minkälaisena ohjelmointi näyttäytyy kuvataideopettajien mielikuvissa - myös sellaisten, jotka eivät ole ottaneet ohjelmointisisältöjä mukaan opetukseensa. Asennekartoitus voisi antaa kiinnostavaa tietoa siitä, minkälaisia syitä kuvataideopettajilla on ottaa tai olla ottamatta ohjelmoinnillisia sisältöjä mukaan opetukseensa. Myöskään tutkielmani ei kartoittanut varsinaisesti oppimista, vaan opettajien käsitystä oppimisesta - kun esimerkiksi kysyin haastateltavilla, mitä taidekasvatuksen

ja ohjelmoinnin yhdistämisellä voi oppia, keskityin opettajien käsityksistä kumpuaviin ajatuksiin oppimisesta, enkä varsinaiseen oppimiseen sinänsä. Tutkielmani lähestyikin oppimista käsityksistä ja hypoteeseista kumpuavalla tavalla. Todellisen oppimisen tutkimiseen täytyisi osallistaa myös oppilaita, ja jotenkin mitata heidän oppimistaan, mikä taas olisi lähtökohta aivan toisenlaiselle tutkimukselle.

Minulle itselleni opinnäytteen kirjoittaminen on ollut hyvin opettavainen prosessi. Minusta tuntuu, että kirjoittaminen on avannut minulle paljon uusia suuntia, joihin olisi ollut kiehtovaa lähteä syventymään tarkemmin. Kun lähdin tutkimaan aihetta, käsitykseni ohjelmoinnista, taidekasvatuksesta ja varsinkin niiden yhdistämisestä, olivat todella hataralla pohjalla. Minulla ei ollut mitään käsitystä siitä, minkälaista ohjelmoinnin hyödyntäminen kuvataidekasvatuksen kontekstissa voisi olla. Erityisesti opettajien haastattelut avasivat minulle kuitenkin aivan uuden maailman, ja olin iloisesti yllättyneet opettajien tekemistä opetusprojekteista ja niiden monipuolisuudesta. Haastateltavien ennakkoluulottomuus, uteliaisuus ja rohkea kokeilevuus ohjelmoinnillisten sisältöjen yhdistämiseen taidekasvatuksessa muuttivat akateemisen kiinnostukseni aidoksi innostukseksi aihetta kohtaan. Toivon, että opinnäytteeni voisi välittää edes hippusen siitä ilosta, ylpeydestä ja ammatillisesta kunnioituksesta, jota koin erityisesti haastatteluja analysoidessani haastateltavia kohtaan. Parhaassa mahdollisessa tapauksessa tutkielmani herättäisi tekstiäni lukevissa taidekasvattajissa kiinnostuksen ohjelmointiin ja sen tuomiin taidekasvatuksellisiin mahdollisuuksiin, ja rohkaisisi aiheesta kiinnostuneita opettajia tekemään omia kokeilujaan.

7. Lähdeluettelo

- Anttila, E. (2017). 4.2. Kokemuksellinen oppiminen. Teoksessa *Teatterikorkeakoulun julkaisusarja: Vsk. 58. Ihmis- ja oppimiskäsitykset taideopetuksessa*. Noudettu 2.7.2020, osoitteesta <https://disco.teak.fi/anttila/4-2-kokemuksellinen-oppiminen/>
- Biesta, G. (2017). *Letting Art Teach. Art Education 'after' Joseph Beuys*. Arnhem: ArtEZ Press.
- Du Satoy, M. (2019). *The Creativity Code: How AI is Learning to Write, Paint and Think*. Lontoo: 4th Estate.
- Dufva, T. (2017). Maker Movement. *Techne Series: Research in Sloyd Education and Crafts Science*, 24(2), 129–141.
- Dufva, T. (2018). *Art education in the post-digital era—Experiential construction of knowledge through creative coding*. Helsinki: Aalto-yliopisto. <http://urn.fi/URN:ISBN:978-952-60-7948-6>
- Dufva, T., & Dufva, M. (2016). Metaphors of code—Structuring and broadening the discussion on teaching children to code. *Thinking Skills and Creativity*, 22, 97–110. <https://doi.org/10.1016/j.tsc.2016.09.004>
- Eskola, J., & Vastamäki, J. (2015). Teemahaastattelu: Opit ja opetukset. Teoksessa *Valli, R. & Aaltola, J. (toim.). Ikkunoita tutkimusmetodeihin 1. Metodien valinta ja aineistonkeruu: Virikkeitä aloittelevalle tutkijalle*. (4. uud. ja täyd. p., ss. 27–44). Jyväskylä: PS-kustannus.
- Fagerlund, J. (2020). Ohjelmointi ja ohjelmoinnillinen ajattelu. Noudettu 27.3.2020, osoitteesta https://peda.net/p/jannefagerlund/ohjelmoinnillinen_ajattelu
- Hallamaa, J., Launis, V., Lötjönen, S., & Sorvali, I. (Toim.). (2006). *Etiikkaa ihmistieteille*. Helsinki: Suomalaisen Kirjallisuuden Seura.
- Heikkinen, S. (2016, syyskuuta 13). Tiedeykkönen: Koodi on kaikkialla - lyhyt johdatus ohjelmoinnin maailmaan. Kuunnelma sarjassa *Tiedeykkönen*. Noudettu 3.4.2020, osoitteesta <https://areena.yle.fi/audio/1-3639697>
- Hirsjärvi, S., & Hurme, H. (2008). *Tutkimushaastattelu. Teemahaastattelun teoria ja käytäntö*. Helsinki: Gaudeamus Helsinki University Press.
- hooks, b. (2007). *Vapauttava kasvatustiete*. Helsinki: Kansanvalistusseura.
- Horila, M., & Tammi, T. (2017). *OPS2016 ja ohjelmointi—Johdattelu ohjelmointiin alakoulussa*. 48.
- Hyvärinen, M. (2017). Haastattelun maailma. Teoksessa M. Hyvärinen, P. Nikander, & J. Ruusuvaara (Toim.), *Tutkimushaastattelun käsikirja*. (ss. 11–45). Tampere: Vastapaino.
- Jormanainen, I. (2018). *Ohjelmointi kouluissa. Trendit ja tulevaisuus*. Esitetty tilaisuudessa MAOL:n syyskoulutuspäivät 2018, Itä-Suomen Yliopisto. Noudettu 6.4.2020, osoitteesta <https://peda.net/joensuu/maol/ljnt/ijokjt:file/download/1be-08958be28970c92619403e544aeb6f25fcce4/Jormanainen-ohjelmointi-kouluissa-2018-10-06.pdf>

- Jungner, M. (2015). *Otetaan digiloikka! Suomi digikehityksen kärkeen*. Esitetty tilaisuudessa Elinkeinoelämän keskusliitto EK. Noudettu 18.2.2020, osoitteesta https://ek.fi/wp-content/uploads/Otetaan_digiloikka_net.pdf
- Kekäleinen, O. (2015). *Onko automatisointiajattelu paras suomennos käsitteestä ”computational thinking”?* (Nro 15; ss. 27–29). Tampereen yliopisto. Noudettu 6.4.2020, osoitteesta: https://trepo.tuni.fi/bitstream/handle/10024/97917/tuovi_%2013_2015.pdf?sequence=1#page=28
- Kiili, C. (2012). *Online Reading as an Individual and Social Practice*. Noudettu 4.8.2020, osoitteesta <https://jyx.jyu.fi/bitstream/handle/123456789/38394/1/978-951-39-4795-8.pdf>
- Knochel, A. D., & Patton, R. M. (2015). If Art Education Then Critical Digital Making: Computational Thinking and Creative Code. *Studies in Art Education; Reston*, 57(1), 21–38.
- Kuula, A. (2006). Yksityisyyden suoja tutkimuksessa. Teoksessa *Etiikkaa ihmistieteille. Toimittaneet Jaana Hallamaa, Veikko Launis, Salla Löytönen ja Irma Sorvali*. (ss. 124–140). Helsinki: Suomalaisen Kirjallisuuden Seura.
- Liukas, L., & Mykkänen, J. (2015). *Koodi2016: Ensiapua ohjelmoinnin opettamiseen peruskoulussa* (1. painos). Helsinki: Linda Liukas ja Juhani Mykkänen.
- Löytönen, T. (ei pvm.). Minäkö tutkija? Johdanto laadulliseen/postpositiiviseen tutkimukseen. Teatterikorkeakoulu. Noudettu 26.3.2020, osoitteesta <https://www.xip.fi/tutkija/0402b.htm>
- Mertala, P., Palsa, L., & Dufva, T. S. (2020). Monilukutaito koodin purkajana: Ehdotus laaja-alaiseksi ohjelmoinnin pedagogiikaksi. *Media & viestintä*, 43(1). <https://doi.org/10.23983/mv.91079>
- Opetushallitus. (2016). *Perusopetuksen opetussuunnitelman perusteet 2014*. Helsinki: Next Print Oy.
- Papert, S. (1985). *Mindstorms. Lapset, tietokoneet, ajattelemisen taito*. Helsinki: Kirjayhtymä.
- Pohjakallio, P. (2005). *Miksi kuvista?: Koulun kuvataideopetuksen muuttuvat perusteet*. Taideteollinen korkeakoulu.
- Rushkoff, D. (2010). *Program or Be Programmed. Ten Commands for a Digital Age*. New York: OR Books.
- Rybatzki-Tiensivu, I. (2017). *BLOB Taiteellisesta ajattelemisesta ohjelmoinnillisen ajattelun opetuksessa* (Aalto-yliopisto). Noudettu 27.1.2020, osoitteesta <https://aaltodoc.aalto.fi:443/handle/123456789/25052>
- Räsänen, M. (2000). *Sillanrakentajat: Kokemuksellinen taiteen ymmärtäminen*. Helsinki: Taideteollinen korkeakoulu.
- Saaranen-Kauppinen, A., & Puusniekka, A. (2006). KvaliMOTV - 5.6 Sosiaalinen konstruktionismi. Noudettu 26.3.2020, osoitteesta https://www.fsd.tuni.fi/menetelmaopetus/kvali/L5_6.html
- Saarela-Kinnunen, M., & Eskola, J. (2015). Tapaus ja tutkimus = tapaustutkimus? Teoksessa R. Valli & J. Aaltoja (Toim.), *Ikkunoita tutkimusmetodeihin 1. Metodien*

- valinta ja aineistonkeruu: Virikkeitä aloittelevalle tutkijalle.* (4. uud. ja täyd. p., ss. 180–203). Jyväskylä: PS-kustannus.
- Saarinen, V. (2017). *Ohjelmointi ja laaja-alaiset osaamisen tavoitteet yläkoulun kuvataiteen opetuksessa* (Aalto-yliopisto). Noudettu 27.1.2020, osoitteesta <https://aaltodoc.aalto.fi:443/handle/123456789/29498>
- Snellman, M. (2018). *Kaikuja pimeästä-hämärästä metsästä. Affekti nykytaiteen oppimisen äärellä ja subjektiviteetin ekologia(ssa)*. Aalto University.
- Tomperi, T. (2017). Kriittisen ajattelun opettaminen ja filosofia. *niin & näin*, 4, 95–112.
- Tujula, I. (2016). *Esteettinen teknologiakasvatus*. Noudettu 27.1.2020, osoitteesta <https://aaltodoc.aalto.fi:443/handle/123456789/22090>
- Ukkonen, R. (2019, maaliskuuta 17). Naisia rohkaistaan, kannustetaan ja houkuteltaan koodaamaan – ”Vielä voi vaikuttaa tulevaisuuteen” | Yle Uutiset | yle.fi [verkkouutiset]. Noudettu 27.3.2020, osoitteesta: <https://yle.fi/uutiset/3-10687050>
- Varto, J. (2006). Seuraava askel—Nykytaiteen viesti kuvataidekasvatukselle. Teoksessa K. Kettunen, M. Hiltunen, S. Laitinen, & M. Rastas (Toim.), *Kuvien keskellä. Kuvataideopettajaliitto 100 vuotta*. (ss. 149–157). Keuruu: Like.
- Varto, J. (2011). Taidepedagogiikan käytäntö, tiedonala ja tieteenala: Lyhyt katsaus lyhyen historian juoneen. Teoksessa E. Anttila (Toim.), *Taiteen jälki. Taidepedagogiikan polkuja ja risteyksiä* (ss. 17–34). Helsinki: Teatterikorkeakoulu.
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717–3725. <https://doi.org/10.1098/rsta.2008.0118>

8. Liitteet

Haastattelulupa

Haastattelulupa

Olet osallistumassa haastatteluun, joka on osa Viia Rannan Aalto-yliopistolle tekemää opinnäytetyötä. Opinnäyte kulkee työnimellä Ohjelmointia kuvataidekasvatuksessa, ja sen tarkoituksena on kartoittaa opettajien tekemiä ohjelmointia ja taidekasvatusta yhdisteleviä kokeiluja peruskouluissa, sekä opettajien kokemuksia näiden kokeilujen äärellä.

Haastattelu taltioidaan ääninauhoitteeksi, josta se litteroidaan teemoittain kirjalliseen muotoon. Videohaastattelun tapauksessa haastattelu tallennetaan kuvan kanssa.

Haastateltavat voivat halutessaan esiintyä haastatteluaineistossa anonymieinä haastateltavien henkilöllisyyden suojelemiseksi. Tällöin haastattelussa ei käytetä haastateltavien oikeita nimiä tai tuoda esiin muita tunnistetietoja, joista haastateltava voitaisiin tunnistaa.

Haastateltava voi kieltäytyä haastattelusta missä vaiheessa tahansa tai kieltää haastatteluaineiston osittaisen käytön missä tahansa haastattelun vaiheessa.

Annan luvan haastattelun nauhoittamisen
Annan luvan käyttää suoria lainauksia haastattelumateriaalista opinnäytetyössä
Haluan esiintyä aineistossa anonymieinä.

Osallistujan allekirjoitus

Päivämäärä & paikka

Kiitos yhteistyöstä!

Haastattelurunko

Perustiedot

- ammatti, koulutustausta, työhistoria
- Millainen työ nyt?

Suhde ohjelmointiin (ennen)

- digit. teknol. yleensä opetuks. ?
- kokemus/taitotas: ohjelmoi. ?
 - miten opetellut?
- termit: onko __ sopiva? (kokonaisuuksille)
- lähtökohta: mikä sai kiinnost.?

Opetuskokonaisuus: kuvaile (voi olla monta)

- yksi/useampi?
- missä? kenelle? milloin? (konteksti)
- laajuus/kesto?
- mitä...
 - a. mater.? b. ohjelm.? c. (taide)välineitä?
- yksin/yhdessä? ketkä?

MITÄ teit?

MIKSI? (tavoitteet)

Oppilaat/oppiminen

- Miten ... teht. vastaan?
- Taitotas: mikä...? kuvis? ohjelm.?
- miten onnistui.. oppim. näkökulm. MITÄ oppivat?
- yleisesti: mitä voi oppia?

Kuvis & ohjelmointi

- yhtäläisyydet?
- erot?
- yhdistämisen... haasteet . mahdollisuudet?

OPS (& kouluympäristö)

- ajatuksia, onko perusteita.. ohjelmoint..+OPS?
- vaikuttiko OPS?
- paikallisOPS?

Merkitys yhteiskunnassa

- Miten ohjelmointi näkyy?
- Ilmiöt
- Tulevaisuus? Mitä, tarvitaanko kuvis, ohjelmointi?

Suhde aiheeseen nyt

- Millainen kokemus? Ajatuksia, tunteita?
- Muuttuiko suhtautumisesi...? Jos/miten?
- Opitko..?
- Miten aiot jatkaa kokeiluja?

